Objective Functions for Optimization of Resilient and Non-Resilient IP Routing

Matthias Hartmann, David Hock, Michael Menth, Christian Schwartz
University of Würzburg, Institute of Computer Science
Am Hubland, D-97074 Würzburg, Germany
Email: {hartmann,hock,menth,schwartz}@informatik.uni-wuerzburg.de

***Abstract*—Intradomain routing in IP networks follows least-cost paths according to administrative link costs. Routing optimization modifies these values to minimize an objective function for a network with given link capacities and traffic matrix. An example for an objective function is the maximum utilization of all links under failure-free conditions or also after rerouting in case of network failures. Many papers have provided heuristic algorithms for routing optimization using different objective functions, but the investigation and comparison of various objective functions has not attracted much attention so far.**

In this work we present several objective functions for resilient IP routing. We also propose a new combined optimization approach which can simultaneously optimize different objective functions with almost no additional computation effort and describe new techniques to minimize overall computation time. The different objective functions and combinations thereof are then analyzed and compared experimentally.

I. Introduction

Intradomain routing in IP networks follows least-cost paths according to administrative link costs. Distributed routing algorithms disseminate the cost values for all links throughout the network so that routers have a consistent view on the topology and the link costs. Based on this information, they calculate least-cost paths and install appropriate entries in the forwarding tables. When links or nodes fail in an IP network, this information is propagated to all nodes so that they can recalculate the least-cost paths and modify their forwarding tables accordingly. Thus, after some time, traffic can be routed again in the remaining topology. This property makes IP networks very robust against failures because correct routes between two nodes are installed as long as they still have a physical connection.

Traffic engineering in IP networks is rather difficult, even when considering routing only under failure-free conditions, because the path layout can be controlled only indirectly by modifying the administrative link costs. For a network with given link capacities and traffic matrix, the link cost settings can be optimized by minimizing a certain objective function like, e.g., the maximum utilization of all links in the network. This optimization problem has been extensively studied in the past and proven to be NP-complete [1]. Therefore, many heuristic algorithms have been proposed which use objective functions that are based on the utilization of the links in the network. The most prominent functions are the maximum link utilization and a function proposed by Fortz in [2] that takes the load of all links into account. There are also other objective functions and some of them have been compared in [3] in a more general context than IP routing.

This work is funded by Deutsche Forschungsgemeinschaft (DFG) under grant TR257/23-2. The authors alone are responsible for the content of the paper.

We consider traffic engineering for resilient IP networks where expected link loads should be low or well balanced even in case of certain outages. Traffic should be carried on appropriate paths under failure-free conditions and also in certain failure scenarios. Therefore, optimization of resilient routing requires objective functions reflecting the quality of the path layout under failure-free conditions and after rerouting in the failure scenarios of interest. So far, only a few studies have tackled optimization of resilient IP routing. Most of them use the maximum link utilization as objective function or a specific failure-comprising extension of the function proposed by Fortz [4].

In this work, we propose additional objective functions for optimization of resilient IP routing that intuitively extend Fortz's objective function to include failures. We investigate whether the objective functions are equivalent and lead to the same optimization results. We examine link utilizations and path lengths in differently optimized IP routings and measure the impact of the considered objective functions on the runtime of our heuristic algorithm. Furthermore, we propose enhanced techniques to minimize computation time of objective functions and present an extension to our heuristic that allows the combined optimization of different objective functions.

The paper is structured as follows. Section II gives an overview of related work. Section III introduces various objective functions for resilient and non-resilient routing optimization. In Section IV we review our optimization approach and propose the new combined optimization and computation speedup techniques. Section V investigates various objective functions for optimization of resilient and non-resilient IP routing. Section VI summarizes this work and draws conclusions.

II. Related Work

We briefly review existing work regarding the optimization of IP routing with and without resilience requirements.

©IEEE, International Workshop on Design of Reliable Communication Networks (DRCN), Washington, D.C., USA, October 2009

A. Optimization of Non-Resilient IP Routing

The problem of IP routing optimization without resilience requirements is NP-complete [1], [5]. Some papers solve the problem by (integer) linear programs [1], [6]–[14]. Since the search space is rather large, others prefer fast heuristics and use local search techniques [2], genetic algorithms [15]–[21], simulated annealing, or other heuristics [22], [23]. Riedl et al. consider non-additive link costs and thereby increase the solution space [24]. Xu et al. propose a new link state routing protocol PEFT; it is based on link costs and can achieve optimal traffic engineering [25].

The papers use different objective functions. The mostly applied objective function is the maximum utilization of all links [11], [16], [18], [20], [24], [25]. A similar objective is the maximization of unused bandwidth [22]. Others combine the maximum link utilization and some other performance objective [6], [9]. In contrast, Fortz et al. propose a more complex objective function where the utilization of each link in the network contributes to the target value using a continuous, piece-wise linear, monotonically increasing penalty function [2]. This function has been adopted by many other researchers [15], [23], [25].

B. Optimization of Resilient IP Routing

Optimization of resilient IP routing improves the path layout for the failure-free scenario and for a set of protected failure cases. First solutions to that problem were presented in [4], [26], [27] for single link failures. The presented algorithms use a local search technique combined with a tabu list or a hash function to mark already visited solutions. To escape from local minima, [4] sets some link weights to random values. To speed up the algorithm, [26] investigates only a random fraction of possible neighboring configurations (link cost settings) while [27], [28] benefit from a heuristic choosing appropriate neighboring configurations that are next visited. To accelerate the computation speed, [29] evaluates the objective function only for a reduced set of critical links instead of the entire set of protected failure scenarios. Recently, we have presented another heuristic for the optimization of resilient IP routing based on the idea of threshold accepting [30]. In [31] the efficiency of heuristic methods based on tabu search and steepest ascent were compared with bounds provided by mixed integer programs. Fortz et al. also propose to modify a few link costs to improve the current load situation when a link fails or when the traffic matrix changes. Thus, they tackle this problem by configuring new link costs rather than to find link cost settings that perform well for a given set of conditions [32], [33].

Optimization of resilient IP routing requires different objective functions than optimization of IP routing for the failure-free scenario. The maximum utilization of all links in all protected failure scenarios has been minimized in [30]. The authors of [27] use a combination of the maximum link utilization in the failure-free scenario and the maximum link utilization in all protected failure scenarios as objective function that needs to be minimized. In [28] they include another performance metric motivated by service level agreements. The authors of [31] maximize the minimum unused capacity on the links in all protected failure scenarios. Fortz et al. extend their objective function previously proposed for the failure-free case by computing its value for the failure-free case and for all protected failure scenarios; their new objective function for resilient routing consists of half the value for the failure-free scenario and half the average of the value of all failure scenarios [4]. Sridharan et al. [29] use a generalized weighted average of these values. Taking the maximum over the values that are obtained by Fortz's function for all failure scenarios has been mentioned by Yuan [26]. In [30] and [34] we consider more complex optimization goals that take into account other technological constraints.

III. Objective Functions for Routing Optimization

We briefly introduce our nomenclature and present various objective functions for resilient and non-resilient routing optimization.

A. Nomenclature

We model a network topology by a graph consisting of a set[1] of nodes (vertices) $\mathcal{V}$ and a set of directed links (edges) $\mathcal{E}$. We describe a failure scenario $s \subseteq (\mathcal{V} \cup \mathcal{E})$ by the set of failed elements. The failure-free scenario is denoted by $s = \emptyset$, and $\mathcal{S}$ describes the set of all considered scenarios. In the remainder of this work, we usually consider the failure free scenario and all single link failures, i.e., $\mathcal{S} = \{\emptyset\} \cup \{\{l\} : l \in \mathcal{E}\}$.

We represent bandwidths and administrative link costs of all links by the vectors[2] $\mathbf{c}$ and $\mathbf{k}$. The link costs are integers between $k_{\min} = 1$ and $k_{\max}$, thus, they are taken from a vector space with $(k_{\max})^{|\mathcal{E}|}$ elements. The default metric uses uniform link costs and it is denoted by $\mathbf{k} = \mathbf{1}$. The routing in IP networks depends on the administrative link costs $\mathbf{k}$ and the specific set s of failed elements. The function $u_s^{\mathbf{k}}(l, v, w)$ indicates the fraction of traffic from v to w that is carried over link l in failure scenario s when link costs $\mathbf{k}$ apply. This description models both single-path and multipath routing.

The indexed components $\mathbf{D}(v, w)$ of traffic matrix $\mathbf{D}$ provide the expected traffic between nodes v and $w \in \mathcal{V}$. The utilization $\rho(\mathbf{k}, l, s)$ of a link l in a failure scenario s, the maximum utilization $\rho_{\mathcal{S}}^{\max}(\mathbf{k}, l)$ of link l in all failure scenarios $s \in \mathcal{S}$, and the maximum utilization $\rho_{\mathcal{S},\mathcal{E}}^{\max}(\mathbf{k})$ of all links $l \in \mathcal{E}$ in all failure scenarios $s \in \mathcal{S}$ are calculated for any link cost vector $\mathbf{k}$ by

$$\rho(\mathbf{k}, l, s) = \left(\sum_{v,w \in \mathcal{V}} u_s^{\mathbf{k}}(l, v, w) \cdot \mathbf{D}(v, w) \right) / c(l) \quad (1)$$

$$\rho_{\mathcal{S}}^{\max}(\mathbf{k}, l) = \max_{s \in \mathcal{S}} (\rho(\mathbf{k}, l, s)) \quad (2)$$

$$\rho_{\mathcal{S},\mathcal{E}}^{\max}(\mathbf{k}) = \max_{l \in \mathcal{E}} (\rho_{\mathcal{S}}^{\max}(\mathbf{k}, l)) \quad (3)$$

[1] Calligraphic letters $\mathcal{X}$ denote sets and the operator $|\mathcal{X}|$ indicates the cardinality of a set.

[2] A link-specific property x can be denoted in a compact way by a vector $\mathbf{x}$ that is printed boldface. The indexed components of a vector are denoted by $\mathbf{x}(l)$ with $l \in \mathcal{E}$.

Fortz et al. [2] explain that it is cheap to send traffic over links with small utilization, but gets increasingly more expensive when a link utilization approaches its capacity. Therefore, they define a function ϕ that depends on the link utilization and penalizes high values. The function ϕ (cf. Figure 1) is continuous and piecewise linear because some integer linear program problem solvers require these properties for their optimization. Furthermore, it is monotonically increasing (concave) in order to favor short paths in the network (cf. Section V-C2) while mitigating high link utilizations. It is defined by $\phi(0) = 0$ and its derivative:

$$\phi'(x) = \begin{cases} 1 & \text{for} \quad 0 \le x < 1/3 \\ 3 & \text{for} \quad 1/3 \le x < 2/3 \\ 10 & \text{for} \quad 2/3 \le x < 9/10 \\ 70 & \text{for} \quad 9/10 \le x < 1 \\ 500 & \text{for} \quad 1 \le x < 11/10 \\ 5000 & \text{for} \quad 11/10 \le x < \infty \end{cases} \tag{4}$$

The general Fortz function for a given scenario $s \in S$ is shown in Equation (5). In contrast to the original definition in [2] we normalized the sum by the number of links.

$$F(s) = \frac{1}{|\mathcal{E}|} \sum_{l \in \mathcal{E}} \phi(\rho(\mathbf{k}, l, s)) \tag{5}$$

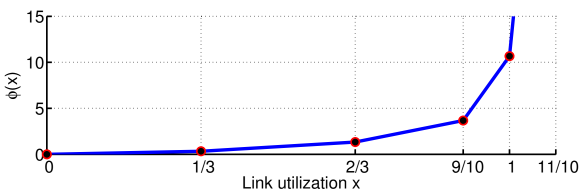


Fig. 1. Fortz's utilization-dependent penalty function ϕ.

B. Objective Functions for Non-Resilient IP Routing

All objective functions calculate a benchmark value for the routing, based on the current link costs $\mathbf{k}$. For explanations, we denote a generic objective function with Ψ, and if the dependency on a special $\mathbf{k}_{\text{spec}}$ is important, we write $\Psi(\mathbf{k}_{\text{spec}})$.

For routing optimization without resilience requirements, the following two objective functions are mainly used.

$U_\emptyset^{\max}$: The maximum link utilization, which has been used, e.g., in [30], reflects only the utilization of the most loaded link. $U_\emptyset^{\max} = \max_{l \in \mathcal{E}}(\rho(\mathbf{k}, l, \emptyset))$

$F_\emptyset$: This is the general Fortz function applied to failure free routing: $F_\emptyset = F(\emptyset)$. In contrast to $U_\emptyset^{\max}$, it reflects the load level of all links.

C. Objective Functions for Resilient IP Routing

For resilient routing optimization, the objective functions $U_\emptyset^{\max}$ and $F_\emptyset$ need to be extended to reflect the link utilizations in all protected scenarios $s \in \mathcal{S}$.

$U_\mathcal{S}^{\max}$: We extend objective function $U_\emptyset^{\max}$ for resilient routing by defining $U_\mathcal{S}^{\max} = \rho_{\mathcal{S},\mathcal{E}}^{\max}(\mathbf{k})$. We used this function in [30]. A similar function has been used by Nucci et al. [27]. They composed a weighted average of the failure free performance and the maximum utilization over all failures: $U_\mathcal{S}^{\text{weighted}} = (1-w) \cdot U_\emptyset^{\max} + w \cdot U_\mathcal{S}^{\max}$.

$F_\mathcal{S}^{\text{weighted}}$: Fortz et al. [4] calculate a weighted average and give equal importance to the failure-free scenario as to all other protected scenarios together. Hence, we get $F_\mathcal{S}^{\text{weighted}} = \frac{1}{2} \cdot F_\emptyset + \frac{1}{2} \cdot \frac{1}{|\mathcal{S}|-1} \cdot \sum_{s \in \mathcal{S}, s \neq \emptyset}(F(s))$.
We also examined the simple equal-weighted average $F_\mathcal{S}^{\text{avg}} = \frac{1}{|\mathcal{S}|} \cdot \sum_{s \in \mathcal{S}}(F(s))$. It produces similar results as $F_\mathcal{S}^{\text{weighted}}$, but takes longer to calculate. Thus, only the interesting computation time results are presented for this objective function.

$F_\mathcal{S}^{\text{max,out}}$: Another option is to take the maximum $F(s)$ of all considered failure scenarios $s \in \mathcal{S}$, which results in $F_\mathcal{S}^{\text{max,out}} = \max_{s \in \mathcal{S}}(F(s))$. A function based on this is used, e.g., in [26].

$F_\mathcal{S}^{\text{max,in}}$: We extend the objective function $F_\emptyset$ for resilience purposes by substituting the utilization $\rho(\mathbf{k}, l, \emptyset)$ of a link in the failure-free scenario by its maximum utilization $\rho_\mathcal{S}^{\max}(\mathbf{k}, l)$ in all considered failure scenarios $s \in \mathcal{S}$. This results in: $F_\mathcal{S}^{\text{max,in}} = \frac{1}{|\mathcal{E}|} \cdot \sum_{l \in \mathcal{E}} \phi(\rho_\mathcal{S}^{\max}(\mathbf{k}, l))$.

IV. Heuristic Optimizer

To compare the quality of different objective functions, we built a heuristic routing optimizer, based on previous work [30]. We present a new optimized objective calculation process which lowers computation times drastically. We also developed a new combined optimization technique, which can improve two different objective functions simultaneously.

A. Heuristic Optimizer

Finding the optimal routing solution for a given network and traffic matrix is an NP-complete problem. The search space of possible link cost settings is very large and enumeration of all settings is impossible even for very small networks. Thus, a heuristic is required for the optimization process. We implemented threshold accepting (TA), a probabilistic heuristic that tries to minimize the objective functions in the network.

TA starts with a random initialization of all link costs $\mathbf{k}$ with values between 1 and $k_{\max}$. At each iteration step, the algorithm randomly selects a (random) number of links whose link costs are (randomly) changed. This new link cost setting $\mathbf{k}_{\text{new}}$ results in a new routing, and thus, a new objective function value Ψ_{new}. If Ψ_{new} is better than ever before, the algorithm stores this value in Ψ_{best} and also stores $\mathbf{k}_{\text{best}}$ to return it later as final result when no better value is found until then. If Ψ_{new} is not worse than a threshold T above the current best value ($\Psi_{\text{new}} \le \Psi_{\text{best}} + T$), the link costs $\mathbf{k}_{\text{new}}$ are accepted as starting point for the next iteration. The threshold is introduced to increase the chance to escape from one of the numerous local minima and find the global minimum. If the new value is above the threshold, the next iteration starts with the previous $\mathbf{k}$. Finally, if no new Ψ_{best} is found after $i_{\max}$ iterations, Ψ_{best} and $\mathbf{k}_{\text{best}}$ are returned as result. The algorithm can then be restarted with a different seed, resulting in a different initialization and other random neighbors so that new areas of the link cost search space are explored.

B. Combined Optimization

Some objective functions only consider specific attributes of a path layout. For example, $U_{\emptyset}^{\max}$ only looks at the link with the highest utilization. All other links are not regarded at all, but routing on them can still be improved using another objective function. Thus, we extended the TA heuristic to perform combined optimization of two objective functions Ψ_1 and Ψ_2. The primary objective function Ψ_1 is minimized as before. When a new cost setting $\mathbf{k}_{\text{new}}$ results in $\Psi_1(\mathbf{k}_{\text{new}}) = \Psi_1(\mathbf{k}_{\text{best}})$, it is only accepted as new best when $\Psi_2(\mathbf{k}_{\text{new}}) < \Psi_2(\mathbf{k}_{\text{best}})$. When $\Psi_1(\mathbf{k}_{\text{new}}) < \Psi_1(\mathbf{k}_{\text{best}})$, then it is accepted (and $\mathbf{k}_{\text{best}}$ is set to $\mathbf{k}_{\text{new}}$) regardless of the performance of the secondary objective function. Thus, Ψ_2 can also increase because more importance is given to Ψ_1.

Note that the computational overhead of the combined optimization is very low since Ψ_2 must only be calculated if $\Psi_1(\mathbf{k}_{\text{new}}) = \Psi_1(\mathbf{k}_{\text{best}})$. Also, the link utilizations are already calculated for Ψ_1 so that no new routing calculation must be performed. Figure 2 presents a typical combined optimization run. It shows the values of two different optimization functions. The upper line represents the current best value of the primary objective function $\Psi_1(\mathbf{k}_{\text{best}})$ which only decreases. The bottom points are the current values of the secondary objective function $\Psi_2(\mathbf{k}_{\text{best}})$.

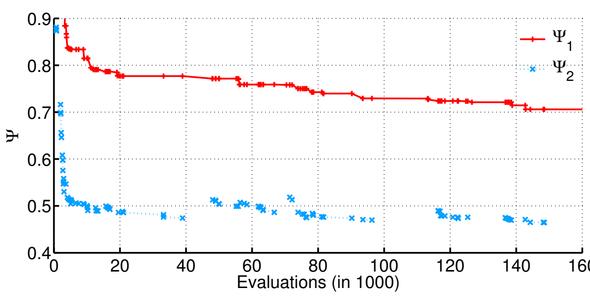


Fig. 2. Evolution of two objective values Ψ_1 and Ψ_2 during a combined optimization run.

C. Improved Computation of Objective Functions

The most time-consuming step in the optimization algorithm is the calculation of the objective function Ψ. Dijkstra's algorithm must be called $|\mathcal{V}|$-times to calculate the routing towards each node $v \in \mathcal{V}$. Then, link utilizations must be calculated from these shortest paths. These complex calculations have to be repeated for every protected scenario $s \in \mathcal{S}$.

Sridharan et al. [29] accelerate the evaluation of the objective function by considering only a set of critical failures instead of the entire set of protected failure scenarios $\mathcal{S}$. This requires careful selection of the included failures. The critical scenarios can also change when routing is changed. Thus, the optimization could run in a wrong direction until a new set of critical links is chosen.

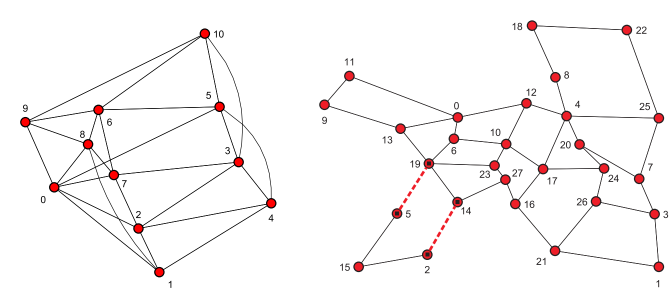
Our algorithm has three speedup improvements. First, it takes the previous Ψ and the current threshold T as input when calculating Ψ_{new}. In each iteration we iterate linearly over a list of all failure scenarios and incrementally calculate Ψ_{new}. The calculation can then be stopped as soon as it is clear that Ψ_{new} will be larger than $\Psi + T$. Second, we check which failure scenario had the largest impact on Ψ_{new} and move this scenario to the front of the scenario list. Then, subsequent calculations of Ψ start with the evaluation of previous worst-case scenarios, and in many cases, the calculation can be aborted early. The third improvement is incremental routing calculation, where only the failure-free routing is calculated completely. In each failure scenario, only the flows that used a failed network element need to be recalculated, while the remaining routing stays the same. All these features together lead to a very fast heuristic optimization and allow us to run a multitude of experiments.

V. Results

We present the networks under study and look at the average duration required by our heuristic to compute different objective functions. Then, we compare properties of optimized link cost settings achieved through different objective functions. We first consider non-resilient IP routing since it is easier to analyze and then resilient IP routing.

A. Experiment Setup

We use well-known realistic networks for our experiments: COST239 [35], GEANT [36], Labnet03 [37], and NOBEL [38]. For compactness sake we present here mainly the results for COST239 and some results for NOBEL (see Figures 3(a) and (b)) since the results from the other networks do not yield additional insights. As benchmark for the optimization we use the unoptimized hop count (HC) routing where each link cost is set to 1 ($\mathbf{k} = \mathbf{1}$). The resulting paths are shortest paths with respect to hop count. The equal-cost multipath (ECMP) routing option is used when multiple equal-cost paths exist. For the optimization and analysis of resilient routing we consider all single link failures. We use population-based traffic matrices [37] and scale them so that the maximum link utilization reaches 100% in the worst link failure scenario for HC routing.



(a) COST239 network with 11 nodes and 26 links.

(b) NOBEL network with 28 nodes and 41 links.

Fig. 3. Networks under study.

We run our TA heuristic for routing optimization with a small threshold $T = 0.001$, a maximum link cost $k_{\max} = 10$, and a high number of unsuccessful iterations $i_{\max} = 100,000$. Most optimization runs tested between 200,000 and 350,000 different link cost settings. We have optimized every network for every objective function during 24 hours, i.e., when a run was finished, the heuristic was restarted with a new seed. The number of performed optimization runs is in the magnitude of about 1000 and strongly depends on the used objective functions and whether resilience was needed or not.

B. Average Evaluation Time of Objective Functions

The optimization step requires the evaluation of the objective function for each considered link cost setting $\mathbf{k_{new}}$. That includes the calculation of the routing and the link utilization for each protected failure scenario $s \in \mathcal{S}$. Thus, optimizing resilient routing is usually more complex than only optimizing the failure-free scenario.

Section IV-C describes methods for computation speedups and here we show their effectiveness depending on the objective function. When calculations are aborted early and worst scenarios are moved to the front of the list, on average only a few scenarios must be evaluated during the search for a new Ψ_{best}. In addition, incremental routing calculation speeds up the computation of the routing in failure scenarios. Table I lists average computation times for the (possibly early aborted) evaluation of objective functions during a regular optimization run. It also shows the average number of evaluated scenarios.

TABLE I
AVERAGE COMPUTATION EFFORT FOR DIFFERENT OBJECTIVE FUNCTIONS.

Network	$U_\emptyset^{\max}$	$F_\emptyset$	$U_\mathcal{S}^{\max}$	$F_\mathcal{S}^{\text{avg}}$	$F_\mathcal{S}^{\text{weighted}}$	$F_\mathcal{S}^{\text{max,out}}$	$F_\mathcal{S}^{\text{max,in}}$
COST239 $\lvert\mathcal{S}\rvert = 27$	0.30ms / 1	0.29ms / 1	0.35ms / 1.88	0.46ms / 7.12	0.42ms / 5.69	0.33ms / 1.30	0.38ms / 2.86
NOBEL $\lvert\mathcal{S}\rvert = 42$	2.47ms / 1	2.47ms / 1	3.50ms / 3.29	8.00ms / 21.20	5.71ms / 12.36	2.97ms / 1.57	3.96ms / 4.20

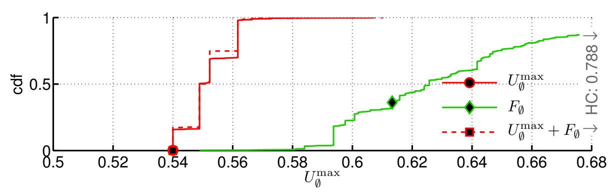
Especially the computation of objective functions that include a maximum can be aborted early. For $U_\mathcal{S}^{\max}$, on average only 1.88 out of 27 scenarios are evaluated in the COST239 network and in the NOBEL network 3.29 out of 42. All protected scenarios $s \in \mathcal{S}$ need to be computed only when $\Psi(\mathbf{k}_{\text{new}})$ is close to the current best result.

When evaluating an objective function that calculates an average, e.g. $F_\mathcal{S}^{\text{avg}}$, more scenarios need to be computed before a worse $\mathbf{k}_{\text{new}}$ can be rejected. As each scenario contributes only a small share to the average value, many scenarios need to be computed before the currently computed average value can exceed a reference value. The evaluation of $F_\mathcal{S}^{\text{avg}}$ in the COST239 network was aborted on average after the computation of 7.12 out of 27 scenarios. In the NOBEL network, even 21.20 out of 42 scenarios were computed on average. The evaluation of $F_\mathcal{S}^{\text{weighted}}$ can be aborted earlier because the failure-free scenario is always computed first and contributes half to the weighted average. Therefore, compared to $F_\mathcal{S}^{\text{avg}}$, for $F_\mathcal{S}^{\text{weighted}}$ fewer other scenarios need to be computed until the interim weighted average exceeds a reference value. The average calculation times in milliseconds show the effect of the incremental routing calculation. Even if many scenarios have to be computed, the additional overhead is low. For example, the evaluation of the single failure-free scenario in $U_\emptyset^{\max}$ takes 2.47ms in the NOBEL network. Evaluating 21.2 scenarios instead of a single one for the calculation of $F_\mathcal{S}^{\text{avg}}$ takes only 8ms which is just 3.24 times longer.

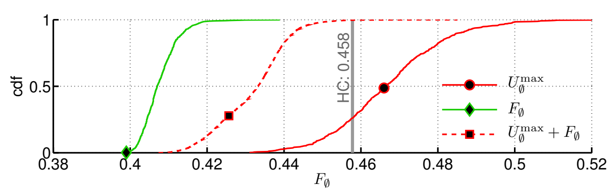
C. Optimization of Non-Resilient IP Routing

We investigate whether the optimization with the objective functions $U_\emptyset^{\max}$ and $F_\emptyset$ leads to link cost settings that are also good in the view of the other. Furthermore, we study average path lengths and link utilizations caused by the different objective functions. Besides, we illustrate the impact of combined optimization using two different objective functions $U_\emptyset^{\max} + F_\emptyset$.

1) Mutual Optimization: We would like to know whether link cost settings optimized with one objective function lead also to good values when evaluated by other objective functions. Therefore, we have evaluated the $U_\emptyset^{\max}$ and $F_\emptyset$ values for all link cost settings optimized with $U_\emptyset^{\max}$, $F_\emptyset$, and $U_\emptyset^{\max} + F_\emptyset$. Figure 4(a) shows the cumulative distribution function (CDF) of the $U_\emptyset^{\max}$ values for the optimized link cost settings in the COST239 network. Link cost settings optimized with $U_\emptyset^{\max}$ or $U_\emptyset^{\max} + F_\emptyset$ mostly have $F_\emptyset$ values between 0.54 and 0.57. This is a significant improvement since the $U_\emptyset^{\max}$ value for HC routing is 0.79. The distinct steps in the CDF imply that there are many link cost settings which lead to the same $U_\emptyset^{\max}$ value. The $U_\emptyset^{\max}$ values for the link cost settings optimized with $F_\emptyset$ are significantly larger and are spread over a wide utilization range. Nevertheless, they are clearly smaller than the $U_\emptyset^{\max}$ value of HC routing. The best link cost setting for $F_\emptyset$ has a relatively large $U_\emptyset^{\max}$ value of 0.613.



(a) CDFs of $U_\emptyset^{\max}$ values.



(b) CDFs of $F_\emptyset$ values.

Fig. 4. CDFs of objective functions for all link cost settings optimized for non-resilient IP routing (lines: optimization objective).

Figure 4(b) presents the CDFs of Fortz's original objective function $F_\emptyset$ for all optimized link cost settings. All CDFs have a continuous shape which means that only very few optimized link cost settings have the same $F_\emptyset$ value, regardless of the objective function used for their optimization. The $F_\emptyset$ values for link cost settings optimized with $F_\emptyset$ are significantly smaller than those for link cost settings optimized with $U_\emptyset^{\max}$

only. About 75% of the latter are even worse than the $F_\emptyset$ value for HC routing. However, the link cost settings of the combined optimization with $U_\emptyset^{\max} + F_\emptyset$ lead to significantly improved $F_\emptyset$ values. The best link cost settings according to their own optimization criterion are identified by a small marker in Figure 4. Their $F_\emptyset$ values are rather high, and the one for the link cost setting with the least $U_\emptyset^{\max}$ value is larger than the $F_\emptyset$ value for HC routing. Thus, link cost settings with good $U_\emptyset^{\max}$ values can produce rather moderate $F_\emptyset$ values.

We briefly recapitulate. The combined optimization leads to link cost settings whose $U_\emptyset^{\max}$ is as good as those of link cost settings optimized with $U_\emptyset^{\max}$ only, but it produces significantly better $F_\emptyset$ values. Since optimization with $U_\emptyset^{\max}$ yields many best link cost settings, a secondary objective function can effectively be pursued by combined optimization. This does not work for $F_\emptyset$ because any change to the routing is likely to modify the $F_\emptyset$ value so that it is hard to find link cost settings with the same $F_\emptyset$ values but better $U_\emptyset^{\max}$ values. Therefore, we did not show results for optimizing first $F_\emptyset$ and then $U_\emptyset^{\max}$. Furthermore, link cost settings optimized with $F_\emptyset$ lead to large $U_\emptyset^{\max}$ values and vice-versa. Hence, $U_\emptyset^{\max}$ and $F_\emptyset$ do not optimize each other.

2) Average Path Lengths: Figure 5 shows CDFs of the average path lengths for IP routing with optimized link cost settings. HC routing leads to the shortest possible average path length of 1.564. Routing optimization with objective function $U_\emptyset^{\max}$ is likely to move traffic away from shortest paths when they have a large utilization otherwise. This leads to a longer average path length.

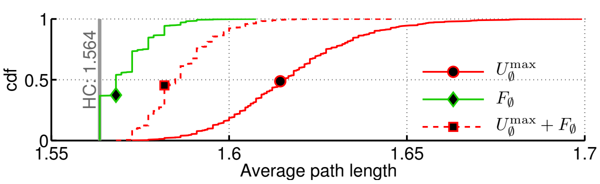


Fig. 5. CDFs of average path lengths for all link cost settings optimized for non-resilient IP routing (lines: optimization objective).

Imagine an additive objective function similar to $F_\emptyset$ whose utilization-dependent penalty function is purely linear. Its lowest value is achieved if traffic is carried on shortest paths, but it does not offer incentives to move traffic away from paths with high utilization. The utilization-dependent penalty in Fortz's objective function $F_\emptyset$ has different slopes. Therefore, its objective value increases if traffic is not carried over shortest paths, unless the utilizations of some links are thereby reduced under a lower threshold of the penalty function. Therefore, link cost settings optimized with $F_\emptyset$ lead to shorter path lengths compared to those optimized with $U_\emptyset^{\max}$. The combined optimization with $U_\emptyset^{\max} + F_\emptyset$ significantly reduces the average path length compared to optimization with $U_\emptyset^{\max}$ only.

3) Link Utilizations: Figure 6 shows the complementary CDF (CCDF) of the link utilizations for the link cost settings with the best objective function values in the COST239 network. The logarithmic y-axis makes differences in the high utilization range more visible. HC routing leads to rather high utilization values on a few links. We use it as reference for comparison. The link cost setting optimized with $F_\emptyset$ reduces the high utilization values and increases the load on lightly utilized links. The link cost setting optimized with $U_\emptyset^{\max}$ minimizes the maximum link utilization dramatically but also increases the load on many other lightly utilized links. The link cost setting obtained from the combined optimization $U_\emptyset^{\max} + F_\emptyset$ limits the maximum link utilization to the same value as the link cost setting optimized with $U_\emptyset^{\max}$ but it increases the load of fewer links.

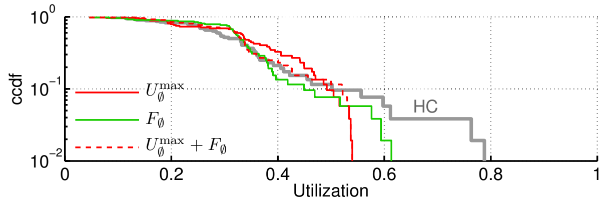


Fig. 6. CCDF of the link utilization for the link cost settings with the best objective function values.

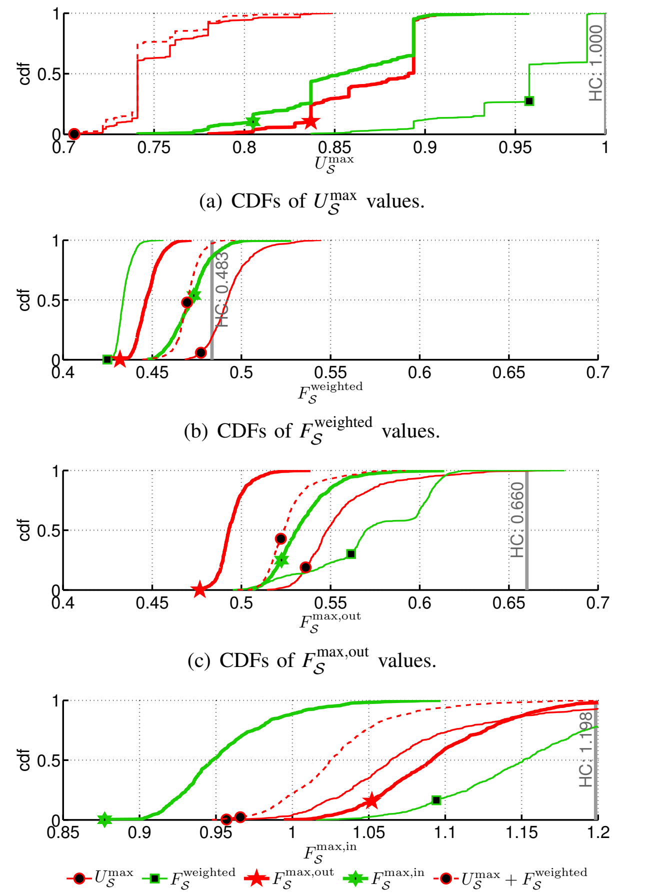
D. Optimization of Resilient IP Routing

We consider the potential of mutual optimization of objective functions for resilient IP routing optimization and report on path lengths and maximum link utilizations. The NOBEL network has extreme bottlenecks in failure cases. We show that combined optimization is particularly useful for resilient IP routing optimization in such cases.

1) Mutual Optimization: Similar to Figures 4(a) and (b), Figures 7(a)–(d) show the CDFs of specific objective functions for link cost settings that were optimized with different objective functions in the COST239 network. The objective function values for the best link cost settings as well as those for HC routing are also indicated.

We observe that the four considered objective functions produce values in different ranges. $U_S^{\max}$ yields maximum utilization values and leads, therefore, to different numbers than the other functions whose values are derived from the utilization-dependent penalty function in Figure 1. The values obtained for F_S^{weighted} are larger than those for the failure-agnostic $F_\emptyset$ in Figure 4(b), followed by $F_S^{\max,\text{out}}$ and $F_S^{\max,\text{in}}$. Due to failures they use larger utilization values as arguments and also various maximization operations lead to larger values. However, it does not make sense to compare values from different objective functions with each other.

The smallest values for a specific objective function Ψ are achieved by link cost settings that were optimized for Ψ. This is the same phenomenon as in Section V-C1 and not a surprising result. The figure also shows that the link cost settings with the best optimized objective values lead only to moderate other objective values. For all objective functions except for F_S^{weighted}, routing optimization achieves a significant improvement compared to HC routing. Compared to $U_S^{\max}$, the combined optimization $U_S^{\max} + F_S^{\text{weighted}}$ improves the values for all considered objective functions. Link cost settings optimized with F_S^{weighted} lead to the worst values for all other objective functions. However, this latter finding depends on the network under study.



(a) CDFs of $U_S^{\max}$ values.

(b) CDFs of F_S^{weighted} values.

(c) CDFs of $F_S^{\text{max,out}}$ values.

(d) CDFs of $F_S^{\text{max,in}}$ values.

Fig. 7. CDFs of objective functions for all link cost settings optimized for resilient IP routing (lines: optimization objective).

2) Average Path Lengths: We report essentially the same findings for average path lengths for optimized resilient IP routing as for optimized non-resilient IP routing in Section V-C2. HC routing produces the shortest paths but link cost settings optimized with F_S^{weighted} also lead to quite short paths. Link cost settings optimized with $U_S^{\max}$ lead to significantly longer average path lengths, but combined optimization with $U_S^{\max} + F_S^{\text{weighted}}$ reduces these values. The relation of the average path lengths of $F_S^{\text{max,out}}$ and $F_S^{\text{max,in}}$ to those of the other objective function depends on the network.

3) Maximum Link Utilization: With resilient routing, we consider the maximum utilization of each link over all failure scenarios. Again, we observe very similar phenomena as in Section V-C3. HC routing produces the largest maximum utilization values on a few links (cf. Figure 8). Link cost settings optimized with $U_S^{\max}$ lead to rather low maximum utilization values but some links that carried only little traffic with HC routing carry significantly more load. Link cost settings optimized with F_S^{weighted} also lead to large maximum link utilization values, but many links have a lower maximum utilization compared to link cost settings optimized with $U_S^{\max}$. Compared to optimization with $U_S^{\max}$, combined optimization with $U_S^{\max} + F_S^{\text{weighted}}$ yields the same upper bounds for maximum utilization values and lightly loaded links carry slightly less traffic. The relation of the CDFs of the maximum link utilization of $F_S^{\text{max,out}}$ and $F_S^{\text{max,in}}$ to those of the other objective function depends on the network.

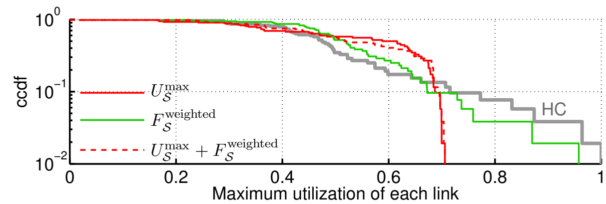


Fig. 8. CCDF of the maximum link utilization over all single link failure scenarios for the link cost settings with the best objective function values (COST239 network).

4) Effect of Combined Optimization in the Presence of Severe Bottlenecks: We consider now the NOBEL network. When one of the links $14 \leftrightarrow 2$ or $19 \leftrightarrow 5$ (highlighted in Figure 3(b)) fails, the other link has a high utilization of 91.45% since nodes 2, 5, and 15 are reachable from the rest of the network only via this link. This sets a lower bound for the maximum link utilization $U_S^{\max}$.

Our experiments with the NOBEL network (no figures provided) show that all link cost settings optimized with $U_S^{\max}$, $F_S^{\text{max,in}}$, and the combined optimization $U_S^{\max} + F_S^{\text{weighted}}$ find a routing that minimizes the maximum link utilization to 91.45%. In contrast, most link cost settings optimized with $F_S^{\text{max,out}}$ exceed this value and all link cost settings optimized with F_S^{weighted} are far from reaching this bound at all. Conversely, the link cost settings optimized with $U_S^{\max}$ have all significantly worse F_S^{weighted} values than HC routing.

TABLE II
BEST LINK COST SETTINGS IN THE NOBEL NETWORK.

Ψ	$U_S^{\max}$	F_S^{weighted}	$F_S^{\text{max,out}}$	$F_S^{\text{max,in}}$
Ψ		**0.3823**	**0.5200**	**1.2186**
$U_S^{\max} + \Psi$		0.3978	0.5364	1.3299
$U_S^{\max}$	**0.9145**	*0.4680*	0.6341	*1.5910*
Hop Count	1.0000	0.4213	0.6612	1.5328

The combined optimization achieves impressive improvements for any secondary Fortz-based objective function Ψ. The first row of the table shows the best values of 3 different optimizations using resilient Fortz-based functions, and the second row shows best values of 3 combined optimizations, using $U_S^{\max}$ together with a Fortz-based function. The columns specify which Fortz-based function was used. The lower part of Table II shows objective function values for a best result of $U_S^{\max}$ optimization and for HC routing. The best $U_S^{\max}$ link cost setting leads to Fortz-based objective values that are about as bad or even worse than those for HC routing. On the contrary, the combined optimizations using $U_S^{\max}$ and a Fortz-based objective function can almost close the gap to the best Fortz results, while still maintaining the same best $U_S^{\max}$ value of 0.9145.

VI. Summary and Conclusion

Non-resilient IP routing optimization is NP-complete and resilient IP routing optimization requires even more effort. Therefore, this problem is usually tackled by heuristics. In this paper we proposed new speedup techniques, new objective functions, and combined optimization of a primary and secondary objective function. The effectiveness of our proposed speedup techniques depends on the applied objective function. The considered objective functions improve the network in different ways. While some of them lower the maximum link utilization, others also try to minimize the path lengths. Routing optimized with one objective function does not need to be good in the light of another objective function. It may even be worse than hop count routing with respect to that objective function. Therefore, it is up to the traffic engineer to carefully decide which optimization goal is most important for him. Combined optimization effects that low maximum link utilizations can be achieved while keeping average path lengths short. It is applicable for optimization of both resilient and non-resilient IP routing and very effective if a few severe network-inherent bottlenecks prohibit an effective improvement of the routing.

Acknowledgments

The authors thank Bernard Fortz for the helpful input, Martin Knauer and Alexander Ruwe for their programming efforts, and Phuoc Tran-Gia for general support.

References

[1] M. Pióro, Á. Szentesi, J. Harmatos, A. Jüttner, P. Gajowniczek, and S. Kozdrowski, "On Open Shortest Path First Related Network Optimisation Problems," *Performance Evaluation*, vol. 48, pp. 201 – 223, 2002.

[2] B. Fortz and M. Thorup, "Internet Traffic Engineering by Optimizing OSPF Weights," in *IEEE Infocom*, Tel-Aviv, Israel, 2000, pp. 519–528.

[3] S. Balon, F. Skivée, and G. Leduc, "How Well Do Traffic Engineering Objective Functions Meet TE Requirements?" in *IFIP-TC6 Networking Conference (Networking)*, May 2006.

[4] B. Fortz and M. Thorup, "Robust Optimization of OSPF/IS-IS Weights," in *Int. Network Optimization Conference (INOC)*, Paris, France, Oct. 2003, pp. 225–230.

[5] ——, "Increasing Internet Capacity Using Local Search," *Computational Optimization and Applications*, vol. 29, no. 1, pp. 13–48, 2004.

[6] S. Köhler, D. Staehle, and U. Kohlhaas, "Optimization of IP Routing by Link Cost Specification," in 15th*ITC Specialist Seminar*, Würzburg, Jun. 2002, pp. 167 – 175.

[7] A. Sridharany, R. Guérin, and C. Diot, "Achieving Near-Optimal Traffic Engineering Solutions for Current OSPF/IS-IS Networks," in *IEEE Infocom*, San Francisco, CA, Apr. 2003.

[8] C. Lopes and A. de Sousa, "Heuristics for the MPLS Network Design with Single Path Minimum Weight Routing," in *International Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks (HETNET)*, 2005.

[9] Y. Wang, Z. Wang, and L. Zhang, "Internet Traffic Engineering without Full Mesh Overlaying," pp. 565 – 571, 2001.

[10] K. G. Ramakrishnan and M. A. Rodrigues, "Optimal Routing in Shortest-Path Data Networks," *AT&T Bell Laboratories Technical Journal*, pp. 117–138, Spring 2001.

[11] M. P. Petterson, R. Szymankek, and K. Kuchcinski, "A CP-LP Hybrid Method for Unique Shortest Path Routing Optimization," in *Int. Network Optimization Conference (INOC)*, 2007.

[12] A. Tomaszewski, M. Pióro, M. Dzida, and M. Zagożdżon, "Optimization of Administrative Weights in IP Networks Using the Branch-and-Cut Approach," in *Int. Network Optimization Conference (INOC)*, 2005.

[13] A. Tomaszwski, M. Pióro, M. Dzida, M. Mycek, and M. Zagożdżon, "Valid Inequalities for a Shortest-Path Routing Optimization Problem," in *Int. Network Optimization Conference (INOC)*, Spa, Belgium, 2007.

[14] A. Bley, "An Integer Programming Algorithm for Routing Optimization in IP Networks," in 16th *Annual European Symposium on Algorithms (ESA)*, Karlsruhe, Germany, 2008, pp. 198–209.

[15] M. Ericsson, M. Resende, and P. Pardalos, "A Genetic Algorithm for the Weight Setting Problem in OSPF Routing," *Journal of Combinatorial Optimization*, vol. 6, pp. 299–333, 2002.

[16] E. Mulyana and U. Killat, "An Alternative Genetic Algorithm to Optimize OSPF Weights," in 15th*ITC Specialist Seminar*, Würzburg Germany, Jul. 2002.

[17] ——, "A Hybrid Genetic Algorithm Approach for OSPF Weight Setting Problem," in 2nd*Polish-German Teletraffic Symposium (PGTS)*, Gdansk Poland, Sep. 2002.

[18] A. Riedl, "A Hybrid Genetic Algorithm for Routing Optimization in IP Networks Utilizing Bandwidth and Delay Metrics," in *IEEE Workshop on IP Operations and Management (IPOM)*, Dallas, TX, Oct. 2002.

[19] ——, "Optimized Routing Adaptation in IP Networks Utilizing OSPF and MPLS," in *IEEE International Conference on Communications (ICC)*, Anchorage, May 2003.

[20] C. Reichert and T. Magedanz, "A Fast Heuristic for Genetic Algorithms in Link Weight Optimization," in 5th*International Workshop on Quality of future Internet Services (QofIS)*, Barcelona, Spain, Sep. 2004, pp. 144–153.

[21] N. Wang, K.-H. Ho, , and G. Pavlou, "Adaptive Multi-Topology IGP Based Traffic Engineering with Near-Optimal Network Performance," in *IFIP-TC6 Networking Conference (Networking)*, Singapore, May 2008.

[22] J. Harmatos, "A Heuristic Algorithm for Solving the Static Weight Optimisation Problem in OSPF," in *IEEE Globecom*, Nov. 2001.

[23] S. Balon and G. Leduc, "Combined Intra- and Inter-domain Traffic Engineering using Hot-Potato Aware Link Weights Optimization," in *ACM SIGMETRICS (short paper)*, 2008.

[24] A. Riedl and D. A. Schupke, "Routing Optimization in IP Networks Utilizing Additive and Concave Link Metrics," *IEEE/ACM Transactions on Networking*, vol. 15, no. 5, pp. 1136–1148, Oct. 2007.

[25] D. Xu, M. Chiang, and J. Rexford, "Link-State Routing with Hop-by-Hop Forwarding can Achieve Optimal Traffic Engineering," in *IEEE Infocom*, Phoenix, AZ, Apr. 2008.

[26] D. Yuan, "A Bi-Criteria Optimization Approach for Robust OSPF Routing," in 3rd*IEEE Workshop on IP Operations and Management (IPOM)*, Kansas City, MO, Oct. 2003, pp. 91 – 98.

[27] A. Nucci, B. Schroeder, S. Bhattacharyya, N. Taft, and C. Diot, "IGP Link Weight Assignment for Transient Link Failures," in 18th *International Teletraffic Congress (ITC)*, Berlin, Sep. 2003.

[28] A. Nucci, S. Bhattacharyya, N. Taft, and C. Diot, "IGP Link Weight Assignment for Operational Tier-1 Backbones," *IEEE/ACM Transactions on Networking*, vol. 15, no. 4, pp. 789–802, Aug. 2007.

[29] A. Sridharan and R. Guerin, "Making IGP Routing Robust to Link Failures," in *IFIP-TC6 Networking Conference (Networking)*, Ontario, Canada, May 2005.

[30] M. Menth, M. Hartmann, and R. Martin, "Robust IP Link Costs for Multilayer Resilience," in *IFIP-TC6 Networking Conference (Networking)*, Atlanta, GA, USA, May 2007.

[31] M. Dzida, M. Zagożdżon, and M. Pióro, "Optimization of Resilient IP Networks with Shortest Path Routing," in *Int. Workshop on the Design of Reliable Communication Networks (DRCN)*, La Rochelle, France, 2007.

[32] B. Fortz, J. Rexford, and M. Thorup, "Traffic Engineering with Traditional IP Routing Protocols," *IEEE Communications Magazine*, vol. 40, no. 10, pp. 118–124, 2002.

[33] B. Fortz and M. Thorup, "Optimizing OSPF/IS-IS Weights in a Changing World," *IEEE Journal on Selected Areas in Communications*, vol. 20, pp. 756 – 767, May 2002.

[34] M. Menth and M. Hartmann, "Threshold Configuration and Routing Optimization for PCN-Based Resilient Admission Control," *accepted for Computer Networks*, 2009.

[35] P. Batchelor et al., "Ultra High Capacity Optical Transmission Networks. Final Report of Action COST 239," Jan. 1999.

[36] "The GEANT website," http://www.geant.net/, 2008.

[37] M. Menth, "Efficient Admission Control and Routing in Resilient Communication Networks," PhD thesis, University of Würzburg, Faculty of Computer Science, July 2004.

[38] "SNDlib 1.0 – Survivable Network Design Data Library," http://sndlib.zib.de, 2009.