Leveraging duality for the Gamma-robust Segment Routing Traffic Engineering Problem

Hugo Callebaut[a,c,*], Jérôme De Boeck[b], Bernard Fortz[b,a,c]

[a]*Départment d'Informatique, Université libre de Bruxelles, 1050 Brussels, Belgium*
[b]*HEC Liège, Management school of the University of Liège, 4000 Liège, Belgium*
[c]*INOCS, INRIA, 59650 Villeneuve d'Ascq, France*

Abstract

Segment Routing (SR), a modern network architecture enhancing traffic engineering, offers flexibility by allowing traffic to be routed through intermediate nodes or links. This paper addresses the challenge of optimizing routing in uncertain traffic distribution scenarios. Rather than relying on a single traffic matrix for optimization, we take a unique approach, considering an infinite set of matrices defined by linear constraints. Our goal is to optimize routing strategies under the worst-case scenario within this set. Through novel formulations, we achieve significant improvements in optimization speed compared to traditional methods that explore all extreme points or use iterative constraint generation for these matrices. This research contributes to the field by enhancing the robustness of SR-based traffic engineering, an area previously explored using different methodologies.

Keywords: Segment Routing, Traffic Engineering, Robust Optimization, Mixed Integer Linear Problem.

1. Introduction

Segment Routing (SR) introduces a flexible framework for routing in IP networks, addressing the limitations of conventional routing protocols. SR is implemented over existing routing protocols, such as OSPF or IS-IS, and provides mechanisms to route traffic through intermediate nodes and links, known as node segments and adjacency segments. It can be realized using different network technologies, including MPLS and IPv6.

The Segment Routing Traffic Engineering Problem (SRTEP) builds on these underlying protocols, assuming fixed link weights. In the SRTEP, only the SR-paths (i.e., the detours taken by traffic) can be adjusted, and there can only be one unique SR-path between each ordered pair of nodes. The goal of SRTEP is typically to minimize the maximum link utilization (MLU) based on a given traffic matrix (TM) [1].

This paper focuses on robust optimization within the SRTEP. Unlike traditional approaches that optimize based on a single traffic matrix, we address an infinite set of matrices defined by linear constraints. Our objective is to identify routing strategies that are robust to the worst-case scenario within this set.

We build on prior work by introducing new linear formulations inspired by the gamma-robustness framework of Bertsimas and Sim [2]. These formulations are evaluated against

*Corresponding author

Email addresses: `hugo.callebaut@ulb.be` (Hugo Callebaut), `jerome.deboeck@uliege.be` (Jérôme De Boeck), `benard.fortz@uliege.be` (Bernard Fortz)

methods such as exhaustive enumeration and constraint generation. Our findings show that our approach provides significant speed improvements, making it a practical and efficient solution for SR-based robust optimization.

Related work: Bhatia et al. [3] present an exact optimization model for the Segment Routing Traffic Engineering Problem (SRTEP) through a path formulation, which can be adapted to account for multiple Traffic Matrices (TMs). They also introduce an oblivious formulation for demand uncertainty and an online routing approach. Callebaut et al. [4] introduce preprocessing techniques discarding potential detours from an optimal SR in path formulations, significantly reducing the size of such models. Jadin et al. [5] tackle the scalability issue by utilizing a column generation approach.

Gay et al. [6] present SRLS, a heuristic designed for fast adaptation to network changes, ensuring rapid responses to traffic variations. Schüller et al. [7] propose a failure-resilient Mixed Integer Linear Programming (MILP) model. Parham et al. [8] explore joint optimization of segment routing paths and shortest-path routing (SPR) weights, combining these two mechanisms for more efficient traffic engineering.

De Boeck et al. [9] introduce a new online segment routing approach under demand uncertainty, distinguishing itself from worst-case approaches and emphasizing the necessity for adaptive optimization strategies in segment routing networks. In addition, comprehensive surveys by Ventre et al. [10] and Wu and Cui [11] offer a broad overview of segment routing methodologies.

Beyond segment routing, other traffic engineering approaches also address the challenge of optimizing network performance over multiple traffic matrices. Kulfi [12] introduces Semi-Oblivious Traffic Engineering (SOTE), combining oblivious routing with dynamic rate adaptation to improve performance under uncertain demands. Adaptive Robust Traffic Engineering in SDN [13] uses the Clustered Robust Routing (CRR) algorithm to reduce reconstructions by clustering traffic matrices based on routing similarities, maintaining performance while minimizing reconfigurations.

L-balanced Weight Settings in OSPF/IS-IS [14] explores weight-setting techniques that balance traffic efficiently across multiple TMs, showing competitive results in handling network hotspots. Finally, Evolutionary Computation for Robust TE [15] uses evolutionary algorithms to optimize routing setups under various traffic matrices and topologies, offering a flexible approach to robust traffic engineering.

2. Base model

In the regular SRTEP, we are given a capacitated weighted graph (V, A) with vertex set V and directed arcs $a \in A$ where each arc has a capacity c_a. A Traffic Matrix $\mathbf{D}$ is also given and $\mathbf{D}_{(s,t)}$ $\forall (s,t) \in \mathbf{D}$ indicates the traffic flowing from node s to t.

We can furthermore compute all possible SR-paths. In general, enumerating all possible paths is impracticable, but by limiting the number of segments and using preprocessing techniques described in [4], the number of paths can be manageable. The set of segment paths of at most k segments between two nodes s and t is denoted by $\mathcal{P}^k_{(s,t)}$. Finally, because the underlying network protocol is defined (we assume here OSPF with Equal Cost Multi-Path), we can also compute the ratio of flow going on a link a when using path $p \in \mathcal{P}^k_{(s,t)}$ which is denoted as f^p_a.

By introducing two new variables, α which is the maximum utilization and x_p a binary variable indicating if a SR-path p is used or not, we can obtain the following mixed integer problem (MIP).

$$\min \quad \alpha \tag{1}$$

$$\text{s.t.} \quad \sum_{p \in \mathcal{P}^k_{(s,t)}} x_p = 1 \qquad \forall (s,t) \in \mathbf{D} \tag{2}$$

$$\sum_{(s,t) \in \mathbf{D}} \mathbf{D}_{(s,t)} \sum_{p \in \mathcal{P}^k_{(s,t)}} f^p_a x_p \leq c_a \alpha \qquad \forall a \in A \tag{3}$$

$$x_p \in \{0,1\} \qquad \forall p \in \mathcal{P}^k_{(s,t)}, (s,t) \in \mathbf{D} \tag{4}$$

$$\alpha \in \mathbb{R} \tag{5}$$

In (1) we minimise the maximum link utilization. In (2), we ensure that exactly one path is chosen between each ordered pair of nodes. In the left part of (3), we compute the total load on an arc a by multiplying for each demand $(s,t) \in \mathbf{D}$ the demand $\mathbf{D}_{(s,t)}$ by the ratio flowing on arc a when using path p and 1 if path p is used or 0 if p is not used. Suppose we divide this by c_a on the right side of the equation. In that case, we obtain the link utilization of a, and the inequality then ensures that α is indeed equal to the maximum link utilization. Constraints (4) and (5) define the domain of variables.

3. Methodology

In the real world, data is often uncertain. Assuming we know a single, perfect Traffic Matrix is elusive. One way to solve this problem is using stochastic optimization, but we focus here on robust optimization (RO). Here, the aim is to optimize for the best performance in the worst conditions, offering a practical solution to the challenge of dealing with uncertain data.

The worst-case(s) still need to be defined and to do this, we start from an initial TM that represents the expected traffic, and we add some uncertainty to this matrix. A common approach would be to add some uncertainty to each demand of the TM, but this is very conservative. If we, for example, added 10% inaccuracy for each demand, the worst-case scenario (which is the one we optimize in RO) would be the scenario where all demands are at 110% of their original capacity. One specificity of using the minimization of the maximum link utilization as objective function is that in such a case, the resulting optimal paths would not change, and the objective, on the other hand, would be multiplied by 110%.

This can be seen easily from the model in Section 2. If all demands are multiplied by a value x only equation (3) changes, and by multiplying α with the same value, an optimal solution for the original problem would stay optimal for the new problem. This shows that the MLU objective function is quite robust.

A drawback of robust optimisation is that it is generally considered too conservative as illustrated by De Boeck et al. [9] who compare robust and stochastic approaches. In our previous example, even if there was a 10% inaccuracy on the expected demands, optimising with respect to the case where all demands take their worst value is too conservative as it is very unlikely that all demands take their worst values simultaneously.

To handle this issue, a parameter Γ was introduced by Bertsimas and Sim [2], which corresponds roughly to the number of measures that can be wrong. Concretely $\tilde{\mathbf{D}}_{(s,t)}$ is the random variable corresponding to the demand between nodes s and t. This variable take values in $[\mathbf{D}_{(s,t)} - e_{st}, \mathbf{D}_{(s,t)} + e_{st}]$ where e_{st} is a parameter for each demand corresponding to how much inaccuracy we allow. In our example, if Γ is an integer, we allow up to Γ demands to differ from $\mathbf{D}$. All demands except for Γ are then equal to $\mathbf{D}_{(s,t)}$ and Γ demands are equal to $\mathbf{D}_{(s,t)}+e_{st}$ since we want to optimise against the worst case scenario.

The case where Γ is not an integer is an extension of the previous case, here $\lfloor\Gamma\rfloor$ demands take values in $[\mathbf{D}(s,t)-e_{st}, \mathbf{D}(s,t)+e_{st}]$ just as previously and one variable takes a value between in $[\mathbf{D}(s,t)-e_{st}(\Gamma-\lfloor\Gamma\rfloor), \mathbf{D}(s,t)+e_{st}(\Gamma-\lfloor\Gamma\rfloor)]$. If, for example, Γ is equal to 2.8, then two variables take the value $\mathbf{D}(s,t)+e_{st}$ and one variable takes the value $\mathbf{D}(s,t)+0.8*e_{st}$.

For ease of understanding, we assume in the remainder of the paper that Γ is integral, but the following proof can be adapted to allow for fractional values, and the resulting model will be the same.

3.1. Dual approach

We now show how to create the new robust model, based upon the model of Section 2, that incorporates the parameter Γ. To do this, we introduce a new set S_a, corresponding to the set of demands that are allowed to deviate from the original TM for each edge and obtain the following model.

$$\min \quad \alpha \tag{6}$$

$$\text{s.t.} \quad \sum_{p\in\mathcal{P}^k_{(s,t)}} x_p = 1 \qquad \forall (s,t)\in\mathbf{D} \tag{7}$$

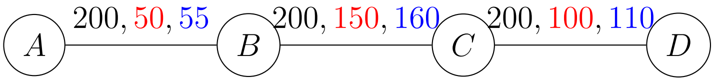
$$\sum_{(s,t)\in\mathbf{D}} \mathbf{D}_{(s,t)} \sum_{p\in\mathcal{P}^k_{(s,t)}} f^p_a x_p + \max_{\{S_a|S_a\subseteq\mathbf{D},|S_a|\leq\Gamma\}} \left\{ \sum_{(s,t)\in S_a} e_{st} \sum_{p\in\mathcal{P}^k_{(s,t)}} f^p_a x_p \right\} \leq c_a\alpha \qquad \forall a\in A \tag{8}$$

$$x_p\in\{0,1\} \qquad \forall p\in\mathcal{P}^k_{(s,t)}, (s,t)\in\mathbf{D} \tag{9}$$

$$\alpha\in\mathbb{R} \tag{10}$$

The approach consisting of adding a second maximization problem inside a constraint is regularly used to write robust counterparts of LPs. With this new model, we force, on each edge, that Γ demands deviate from their original value such that the maximum link utilization on each edge is maximal w.r.t. our robust model. One should note that this is slightly different from our original problem because the maximisation of S_a occurs for each constraint of type (8) which means that for each link the Γ demands for which the quantity $\sum_{(s,t)\in S_a} e_{st}\sum_{p\in\mathcal{P}^k_{(s,t)}} f^p_a x_p$ is maximal is allowed to deviate from $\mathbf{D}(s,t)$. At the same time, we would want the Γ demands that change to be the same on the whole network. This modification to our program also implies that the flows are modified. We show this in Figure 1.

Because Γ is equal to 1, only one demand should be different from $\mathbf{D}$, in this case, $\mathbf{D}_{(B,D)}$ should be equal to 110 because this is the worst case scenario. In the robust counterpart, on the other hand, we see that on edge AB, the value of $\mathbf{D}_{(A,C)}$ is also changed to 55, meaning that two demands are different from $\mathbf{D}(s,t)$. These two demands are not different "at the same time," and because we use the MLU objective function, this does not change the optimal paths nor the value of the objective function. We will prove this later.



$\mathbf{D}_{(A,C)}$	50
$\mathbf{D}_{(B,D)}$	100
e_{AC}	5
e_{BD}	10
Γ	1

Figure 1: Network on the left with parameters on the rights forming the SRTEP. Black numbers on the edges are the capacity, in red the flow of the non-robust optimisation problem, and in blue the flow of the robust counterpart. The robust counterpart of the SRTEP for this network does not conserve the flow.

Now, assuming that the model from equations (6) to (10) does indeed solve our original problem, we adapt the reformulation proposed by Bertsimas and Sim for Γ-robustness [2] to remove the maximization constraint from the model. If we focus on the maximisation part of constraint (8), given a vector x^*, a solution of the problem giving a boolean value to each path, we define:

$$\beta_a(x^*) = \max_{\{S_a | S_a \subseteq \mathbf{D}, |S_a| \leq \Gamma\}} \left\{ \sum_{(s,t)\in S_a} e_{st} \sum_{p\in\mathcal{P}^k_{(s,t)}} f^p_a x^*_p \right\} \leq c_a \alpha \qquad \forall a \in A \qquad (11)$$

This is equal to:

$$\beta_a(x^*) = \quad \max \quad \sum_{(s,t)\in\mathbf{D}} e_{st} z_{st} \sum_{p\in\mathcal{P}^k_{(s,t)}} f^p_a x^*_p \qquad (12)$$

$$\text{s.t.} \quad \sum_{(s,t)\in\mathbf{D}} z_{st} \leq \Gamma \qquad (13)$$

$$0 \leq z_{st} \leq 1 \qquad \forall (s,t) \in \mathbf{D} \qquad (14)$$

Clearly, the optimal solution of the above problem has Γ z_{st} variables set to 1 which is equivalent to the selection of subset $\{S_a | S_a \subseteq \mathbf{D}, |S_a| \leq \Gamma\}$ with corresponding cost function $\sum_{(s,t)\in S_a} e_{st} \sum_{p\in\mathcal{P}^k_{(s,t)}} f^p_a x^*_p$.

Now let us take the dual of the above problem;

$$\min \quad \Gamma\delta_a + \sum_{(s,t)\in\mathbf{D}} \lambda_{ast} \qquad (15)$$

$$\text{s.t.} \quad \delta_a + \lambda_{ast} \geq e_{st} \sum_{p\in\mathcal{P}^k_{(s,t)}} f^p_a x^*_p \qquad \forall (s,t) \in \mathbf{D} \qquad (16)$$

$$\delta_a \geq 0 \qquad (17)$$

$$\lambda_{ast} \geq 0 \qquad \forall (s,t) \in \mathbf{D} \qquad (18)$$

In this problem, we introduce one new variable δ_a and $|\mathbf{D}|$ variables λ_{ast} for each edge a. We also need to add $|\mathbf{D}| \times |A|$ new constraints, but most of them have no impact because the $e_{st} \sum_{p\in\mathcal{P}^k_{(s,t)}} f^p_a$ part is zero and if that is the case the constraint states that two positive variables must be greater than 0 which is always true.

Finally, we can replace the maximization problem in (8) by this dual problem. We can do this because of the strong duality theorem. Indeed, this theorem implies that the MLU (α) will be at least as big as the real α since the result by which we replace the maximisation is at least as big as the maximisation and at optimality it will have the exact same value. The final model we obtain is the following.

$$\begin{aligned}
\min \quad & \alpha && && (19)\\
\text{s.t.} \quad & \sum_{p \in \mathcal{P}^k_{(s,t)}} x_p = 1 && \forall (s,t) \in \mathbf{D} && (20)\\
& \sum_{(s,t)\in \mathbf{D}} \mathbf{D}_{(s,t)} \sum_{p \in \mathcal{P}^k_{(s,t)}} f^p_a x_p + \Gamma \delta_a + \sum_{(s,t)\in \mathbf{D}} \lambda_{ast} \leq c_a \alpha && \forall a \in A && (21)\\
& \delta_a + \lambda_{ast} \geq e_{st} \sum_{p \in \mathcal{P}^k_{(s,t)}} f^p_a x_p && \forall (s,t) \in \mathbf{D},\ a \in A && (22)\\
& x_p \in \{0,1\} && \forall p \in \mathcal{P}^k_{(s,t)}, (s,t) \in \mathbf{D} && (23)\\
& \delta_a \geq 0 && \forall a \in A && (24)\\
& \lambda_{ast} \geq 0 && \forall (s,t) \in \mathbf{D},\ a \in A && (25)
\end{aligned}$$

We will now prove that the solution given by this model in which the error for Γ demands is maximized at each arc is the same as the one given by the problem where the error for Γ demands is maximized over the whole network (i.e., the same demands on each arc). Let us take an optimal solution to the original robust problem and assume the MLU is equal to $\tilde{\alpha}$. This means that on all possible TMs where at most Γ demands deviate from their original value, using the paths from the solution, the MLU will always be smaller or equal to $\tilde{\alpha}$. If we now use the optimal paths found for $\tilde{\alpha}$ in our modified robust problem and assume that the obtained MLU α^* is strictly greater than $\tilde{\alpha}$, then there is an arc on which by increasing at most Γ demands, the MLU is strictly greater than $\tilde{\alpha}$. We could then simply use the traffic matrix where these demands are increased in the original model, and we would obtain a link utilization on an edge $> \tilde{\alpha}$, which is not possible since that was our optimal solution.

The other way round, we can also see that it is impossible for $\tilde{\alpha}$ to be greater than α^*, proving that both values should always be equal. This result depends on the fact that we use the MLU as objective function. It does not hold for general linear objectives.

3.2. Adversarial approach

Because the previous approach does not generally work with any objective function, we also present an adversarial approach that should work in the general case.

The LP from section 2 can be adapted to have a robust routing where we optimise with respect to a set of demand matrices instead of a unique TM. To this end, we only have to copy constraint (3) for each demand matrix. This would then become :

$$\sum_{(s,t)\in \mathbf{D}} \mathbf{D}_{(s,t)} \sum_{p \in \mathcal{P}^k_{(s,t)}} f^p_a x_p \leq c_a \alpha \qquad \forall a \in A, \mathbf{D} \in \mathcal{D} \qquad (26)$$

assuming that the set of demand matrices is $\mathcal{D}$.

If we use a set of TMs defined by linear constraints to define all the matrices we want to be protected against (which is the case in Γ optimisation), then we could simply enumerate all the extreme points of this set and giving them as input to our program would provide us with an optimal solution. An extreme point refers to a vertex of the feasible region, which is defined by the intersection of the linear constraints. In our case, an extreme point is a traffic matrix where exactly Γ demands take their worst value while the other demands remain unchanged.

Although the example provided here focuses on minimizing Maximum Link Utilization (MLU), the adversarial approach is flexible and can accommodate other objective functions. For example, instead of minimizing MLU, we could aim to minimize the total delay experienced by all traffic flows in the network [16] or use the sum of piecewise linear functions of link utilization [17]. However, the dual approach is not applicable with these objective functions, as the proof at the end of Section 3.1 does not hold. For a comparison of possible objective functions for traffic engineering problems, we refer to [18].

Another method consists of starting with one or multiple extreme points, solving the problem and then generate iteratively a new matrix that gives a new worst-case scenario and then solve the problem again with the previous matrices and the new worst-case matrix.

When the new worst-case matrix generated does not worsen the optimal solution anymore, we then have found an optimal solution and can stop. Because we will generate an extreme point of the demand set at each iteration and because there is only a finite number of extreme points, there can only be a finite number of iterations to this method.

Generating a new worst-case matrix is very simple in our case. Since all SR-paths are known for a solution, we can access the flow ratio on each arc for each demand. To create a new worst-case matrix, we then simply need to multiply these ratios for each arc by the corresponding error e_{st}, we then sort them and take the Γ largest values. We recompute the new utilization on each arc, assuming that these Γ demands take their worst possible value. A new worst-case TM can then be created by taking the Γ chosen demands that take their worst value on the link with the highest utilization.

To create an initial worst-case matrix, we will take the Γ demands with the highest error e_{st}, and these demands will be the ones that deviate from their original value.

4. Results

The experiments were conducted in two stages using different sets of topologies from the Repetita dataset [19]. In the first set of tests, 134 topologies were used, with up to 30 nodes and 98 edges. Each topology was associated with 5 traffic matrices for a total of 670 instances. These tests were executed on a machine equipped with 62 GB of RAM and 8 cores, each running at 2.60 GHz.

In a second set of experiments, we expanded the testing to 81 larger topologies, with up to 68 nodes and 226 edges. For these tests, we used a machine with 110 GB of RAM, while keeping the same 8 cores at 2.60 GHz. Unlike the previous tests, only one traffic matrix per topology was evaluated. In both cases the traffic matrices were generated such that an optimal general routing (in the non robust case) would result in a maximum link utilization of 90%. Results for these tests are available in Appendix A.

The choice of using 8 cores was informed by preliminary results, which showed that for similar LPs, using 8 cores provided faster results compared to 4 or 16 cores. According to Gurobi's documentation, using fewer cores can sometimes reduce the overhead of data exchange between cores, which may explain why 8 cores yielded better performance in these tests.

During the solving of the linear programs (LPs), two main issues were encountered. The first issue related to Gurobi's default `optimalityTol` parameter, which in some cases,

Name	α (NR)	α (R)	$\hat{\alpha}$ (NR)	$\hat{\alpha}$ (R)	t (s) (NR)	t (s) (R)
Renater2008 $\Gamma = 2$	0.900	0.992	1.317	1.069	0.86	617.36
Renater2008 $\Gamma = 5$	0.900	0.961	1.466	1.174	0.86	989.66

Table 1: Example showing the impact of robustness on the topology Renater2008 with different values of Γ

particularly with the robust dual approach, resulted in solutions considered optimal but deviating up to 25% from the actual optimal solution. This was caused by the introduction of small variables (δ_a and λ_{ast}) in the robust formulation, which led to inaccurate reduced costs. By reducing the `optimalityTol` parameter, we were able to achieve more accurate solutions.

The second issue arose in the iterative constraint generation approach. When adding new constraints to an existing model, previously added constraints were sometimes violated after solving the updated model. This was resolved by rebuilding the model from scratch at each iteration, which did not result in slower performance compared to incrementally updating the model.

4.1. Maximum utilization gain

In this section, we analyse the gains and losses in maximum utilization we obtain when using a robust model compared to a non-robust model in the average and worst cases. To do this, we first need to create the robust cases towards which we want to protect. Here, we assume that up to Γ demands may double. Let us now first explain an example in Table 1 on the topology Renater2008 with 33 nodes and 86 edges. This example is available in Tables A.5 and A.6 from Appendix A.

The explanation for each column is the following:

- α (NR): Is the optimal load on the average case using SR (i.e. the original TM using SR-paths optimised for that matrix).
- α (R): Is the load on the average case (i.e. original TM) using SR-paths optimised for the robust case.
- $\hat{\alpha}$ (NR): Is the load in the worst case scenario according to the robustness used (i.e. Γ equal to 2 or 5 depending on the row and where Γ demands may up to double) and using SR-paths optimised for the average (i.e. original) TM.
- $\hat{\alpha}$ (R): Is the optimal load in the worst case scenario using SR.
- t (s) (NR): Is the solve time in seconds for the non-robust case (i.e. α (NR) column)
- t (s) (R): Is the solve time in seconds for the robust case (i.e. $\hat{\alpha}$ (R) column)

We mention that the "α (R)" and "$\hat{\alpha}$ (NR)" columns do not have computation times since there is no optimization involved. It simply amounts to computing the load in a network where the SR-paths are already given. Columns "α (NR)" and "t (s) (NR)" are also exactly the same in both rows since these values are independent of the value of Γ used.

We can then compute the difference between "α (R)" and "α (NR)", 9.2% in the case where $\Gamma = 2$. The interpretation of this value is the loss in maximum link utilization we have using the optimal robust paths instead of the optimal average paths when the traffic

	Gamma=1	Gamma=2	Gamma=3
Average α (NR)	0.959		
Average α (R)	1.046	1.025	1.025
Average $\hat{\alpha}$ (NR)	1.359	1.478	1.550
Average $\hat{\alpha}$ (R)	1.257	1.352	1.420
Loss in average case	0.087	0.066	0.066
Gain in worst case	0.102	0.126	0.130

Table 2: Averages of gains and losses using robust or non-robust optimization on the average and worst case traffic matrices

is equal to the original traffic matrix. In the same way we can compute the difference between "$\hat{\alpha}$ (R)" and "$\hat{\alpha}$ (NR)" which is this time 24.8%. The interpretation for this value is then the gain in maximum link utilization when using paths optimised for the robust problem instead of the average traffic matrix when the worst case scenario traffic matrix occurs.

The results on the small topologies (less than 30 nodes) for values of Γ equal to 1, 2 and 3 are shown in Table 2 the average is computed over all 670 instances (i.e. 134 topologies times 5 TMs per topology). Interestingly, we see that when $\Gamma = 1$, we have the highest loss on the average TM. When using values of 2 or 3 for Γ, we can see that the gains and losses are approximately the same, though the maximum utilization in the worst-case scenario differs significantly.

These results highlight the importance of choosing an appropriate value for Γ. Selecting Γ as large as the number of demands would not lead to any improvements, as the minimization of maximum link utilization is already relatively robust, as explained in Section 3. Furthermore, although $\Gamma = 1$ may seem less robust and performs worse in the average case, it performs best in cases where only one demand may double. Since the purpose of Γ-robustness is to balance robustness, selecting a higher value for Γ may be unnecessary if the scenario of multiple demand increases is unlikely. Thus, if the worst-case scenario is improbable, it would not be beneficial to increase Γ unnecessarily.

We provide the same analysis this time for the larger topologies (30 to 68 nodes) in Table 3. Although not all instances could be solved to optimality within the 1-hour computation limit, the results obtained upon timeout were sufficiently close to optimal and have been retained for analysis. This time the values of Γ have been set to 2 and 5 as it makes sense for larger topologies to use larger values of Γ.

We see here that in the case where Γ is equal to 2, the results with respect to gains and losses in maximum utilization are approximately equivalent. We can this time also see that a higher value of Γ results in bigger differences when comparing the different routings on the non-robust and robust TMs which is what one would expect as long as Γ is smaller than the number of demands divided by 2. This is because if Γ is equal to the number of demands, all demands would double their value and an optimal routing for the robust case would be optimal for the non-robust case.

4.2. Time analysis

In this section, we analyse the time taken for each method to find the optimal solution, we only analyse the small instances as for the larger ones, many instances could not be solved within the given time limit using the dual approach. Using the other approaches, even the small instances could be very challenging. We still assume that there can be

	Gamma=2	Gamma=5
Average α (NR)	0.906	
Average α (R)	0.970	0.979
Average $\hat{\alpha}$ (NR)	1.208	1.354
Average $\hat{\alpha}$ (R)	1.091	1.208
Loss in average case	0.064	0.073
Gain in worst case	0.117	0.146

Table 3: Averages of gains and losses using robust or non-robust optimization on the average and worst case traffic matrices

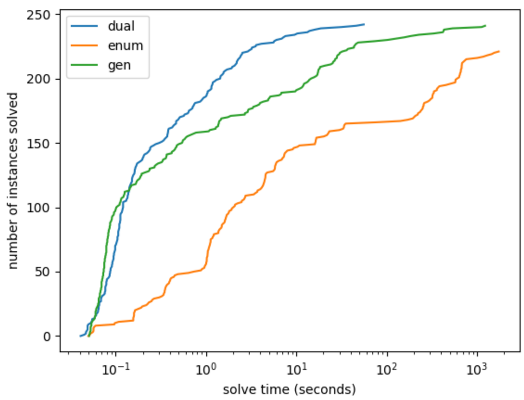


Figure 2: Number of instances solved within a time limit on the 243 smallest instances

up to Γ demands that can double, and we take $\Gamma = 2$ unless specified otherwise. In Figure 2, we first show the computation time to solve the 243 smallest instances using the different methods. We clearly see that the brute force enumeration performs very poorly, as expected, and does not even manage to solve all instances. The constraint generation approach is better than expected but still much worse than the dual model. Given 24 hours of computation total time and 30 minutes per instance, the enumeration approach only managed to solve 231 instances optimally, the constraint generation approach solved 303 instances optimally, and the dual approach solved 621 instances optimally. This shows that the dual approach is much better than the other two methods. This is even more obvious, knowing that all instances were ordered in increasing size, meaning that the 318 more instances that the dual approach solved were much more challenging than the previous ones.

Since the dual formulation seems to be the best one by far, we continue our comparisons using only this method. In Figure 3, we compare the solving time between the non-robust and robust approaches. Here, we can see that, as expected, adding robustness to the problem makes it much harder to solve. We see that apart from the last few topologies, within a certain time limit, we manage to solve over twice as many instances without adding robustness to our problem, note that this is using only $\Gamma = 2$. We could also analyze the impact of the parameter Γ on the solving time.

We show this impact in Figure 4. Here, we can see that despite the size of our model not changing, the complexity of the problem increases as the computation time increases with bigger values of Γ.

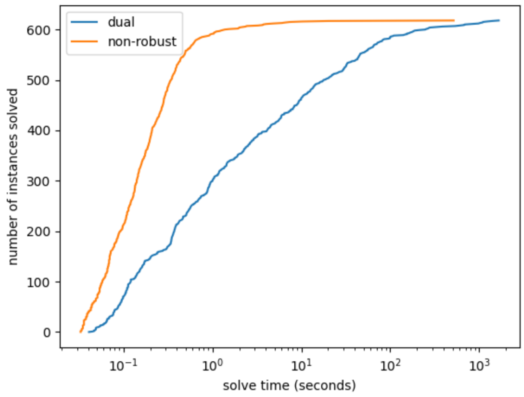


Figure 3: Comparison in solving time between robust and non-robust approach

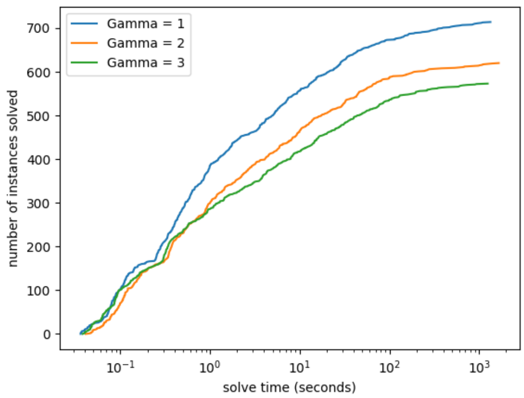


Figure 4: Impact of Γ on the solving time

5. Conclusion and future work

In this article, we analysed different methods for solving the gamma-robust SRTEP. We provided a dual formulation, slightly different from the original problem, for which an optimal solution also is an optimal solution of our original problem.

This formulation unfortunately only works when using the maximum utilization objective function. We then compared our dual model to models exhaustively enumerating all possible TMs and generating a new worst TM at each iteration with respect to the robust parameter.

The dual model outperformed both models by far, but an advantage of the other models is that they can accept any linear objective function.

We analysed the possible gains and losses on the average and worst-case TMs using the robust paths and, surprisingly, saw that only allowing one demand to deviate from the original TM produced the worst results on the average TM.

We also analysed the computation time needed for the robust dual model compared to a non-robust model and noticed that over twice as many instances could be solved within a certain time using the non-robust model compared to the robust model where up to two demands can deviate.

Finally, we also analysed the influence of the parameter Γ on the solving time. Despite all models having exactly the same size regardless of the Γ value, we could see that higher values lead to increasingly difficult problems.

Moving forward we plan on exploring the constraint generating method even more. Currently our approach involves solving the integer program optimally to produce a single new worst-case traffic matrix in each iteration. However we envision enhancing our methodology by generating multiple matrices at each step. Another way to speed up the constraint generation would be to initially only solve the relaxation of the problem, in the later stages only will the integer program be solved optimally to generate constraints that the relaxation did not generate and finally find an optimal solution to the integer problem.

Acknowledgements

Computational resources have been provided by the Consortium des Équipements de Calcul Intensif (CÉCI), funded by the Fonds de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under Grant No. 2.5020.11 and by the Walloon Region.

References

[1] R. Hartert, S. Vissicchio, P. Schaus, O. Bonaventure, C. Filsfils, T. Telkamp, P. François, A declarative and expressive approach to control forwarding paths in carrier-grade networks, SIGCOMM (2015).

[2] D. Bertsimas, M. Sim, Robust discrete optimization and network flows, Mathematical Programming 98 (2003) 49–71.

[3] R. Bhatia, F. Hao, M. Kodialam, T. Lakshman, Optimized network traffic engineering using segment routing, IEEE Conference on Computer Communications (INFOCOM) (2015) 657–665.

[4] H. Callebaut, J. De Boeck, B. Fortz, Preprocessing for segment routing optimization, Networks 82 (4) (2023) 459–478. doi:10.1002/net.22165.

[5] M. Jadin, F. Aubry, P. Schaus, O. Bonaventure, Cg4sr: Near optimal traffic engineering for segment routing with column generation, IEEE INFOCOM 2019 - IEEE Conference on Computer Communications (2019) 1333–1341.

[6] S. Gay, R. Hartert, S. Vissicchio, Expect the unexpected: Sub-second optimization for segment routing, in: IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, 2017, pp. 1–9. doi:10.1109/INFOCOM.2017.8056971.

[7] T. Schüller, N. Aschenbruck, M. Chimani, M. Horneffer, Failure resiliency with only a few tunnels – enabling segment routing for traffic engineering, IEEE/ACM Transactions on Networking 29 (1) (2021) 262–274. doi:10.1109/TNET.2020.3030543.

[8] M. Parham, T. Fenz, N. Süss, K. Foerster, S. Schmid, Traffic engineering with joint link weight and segment optimization, in: Proceedings of the 17th International Conference on Emerging Networking EXperiments and Technologies, CoNEXT '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 313–327. doi:10.1145/3485983.3494846.

[9] J. De Boeck, B. Fortz, S. Schmid, The case for stochastic online segment routing under demand uncertainty, in: 2023 IFIP Networking Conference (IFIP Networking), 2023, pp. 1–8. doi:10.23919/IFIPNetworking57963.2023.10186361.

[10] P. Ventre, S. Salsano, M. Polverini, A. Cianfrani, A. Abdelsalam, C. Filsfils, P. Camarillo, F. Clad, Segment routing: A comprehensive survey of research activities, standardization efforts, and implementation results, IEEE Communications Surveys & Tutorials PP (2020) 1–1. doi:10.1109/COMST.2020.3036826.

[11] D. Wu, L. Cui, A comprehensive survey on segment routing traffic engineering, Digital Communications and Networks (2022). doi:10.1016/j.dcan.2022.02.006.

[12] P. Kumar, Y. Yuan, C. Yu, N. Foster, R. Kleinberg, R. Soulé, Kulfi: Robust traffic engineering using semi-oblivious routing (2016). arXiv:1603.01203.
URL https://arxiv.org/abs/1603.01203

[13] D. Sanvito, I. Filippini, A. Capone, S. Paris, J. Leguay, Adaptive robust traffic engineering in software defined networks, in: 2018 IFIP Networking Conference (IFIP Networking) and Workshops, 2018, pp. 145–153. doi:10.23919/IFIPNetworking.2018.8696406.

[14] H. Abrahamsson, M. Bjorkman, Robust traffic engineering using l-balanced weight-settings in ospf/is-is, in: 2009 Sixth International Conference on Broadband Communications, Networks, and Systems, 2009, pp. 1–8. doi:10.4108/ICST.BROADNETS2009.7184.

[15] V. Pereira, M. Rocha, P. Cortez, M. Rio, P. Sousa, A framework for robust traffic engineering using evolutionary computation, in: G. Doyen, M. Waldburger, P. Čeleda, A. Sperotto, B. Stiller (Eds.), Emerging Management Mechanisms for the Future Internet, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 1–12.

[16] M. Gerla, L. Kleinrock, Communication nets: stochastic message flow and delay 25 (1) (1977) 48–60.

[17] B. Fortz, M. Thorup, Increasing internet capacity using local search, Computational Optimization and Applications 29 (1) (2004) 13–48.

[18] S. Balon, F. Skivée, G. Leduc, How well do traffic engineering objective functions meet te requirements?, Vol. 3976, 2006, pp. 75–86. doi:10.1007/11753810_7.

[19] S. Gay, P. Schaus, S. Vissicchio, Repetita: Repeatable experiments for performance evaluation of traffic-engineering algorithms, arXiv:1710.08665v1 (2017).

Appendix A. Appendix: Detailed Results

In this appendix, we present the detailed results of our experiments focusing on the larger topologies (30-68) nodes. These topologies represent more complex and computationally significant cases, providing a clearer evaluation of the robustness and scalability of our approach. Although similar experiments were conducted on smaller topologies (4–30 nodes), their results follow the same trends as those observed for the larger topologies and are thus omitted here for brevity.

Table A.4 is a list of all topologies used with the number of nodes, edges and non-dominated paths. These are the paths that are effectively used in the linear problem [4].

Tables A.5 and A.6 present the results obtained on the larger topologies for values of Γ respectively equal to 2 and 5. The columns represent the following data:

- α (NR): Is the optimal load on the average case using SR (i.e. the original TM using SR-paths optimised for that matrix).
- α (R): Is the load on the average case (i.e. original TM) using SR-paths optimised for the robust case.
- $\hat{\alpha}$ (NR): Is the load in the worst case scenario according to the robustness used (i.e. Γ equal to 2 or 5 depending on the table and where Γ demands may up to double) and using SR-paths optimised for the average (i.e. original) TM.
- $\hat{\alpha}$ (R): Is the optimal load in the worst case scenario using SR.
- t (s) (NR): Is the solve time in seconds for the non-robust case (i.e. α (NR) column)
- t (s) (R): Is the solve time in seconds for the robust case (i.e. $\hat{\alpha}$ (R) column)

We mention that the "α (R)" and "$\hat{\alpha}$ (NR)" columns do not have computation times since there is no optimization involved. It simply amounts to computing the load in a network where the SR-paths are already given. Columns "α (NR)" and "t (s) (NR)" are also exactly the same in both tables A.5 and A.6 since these values are independent of the value of Γ used. Finally if a computation time is greater than 3600 seconds, this means that an optimal value was likely not found as this was the limit set per instance. The obtained results are likely still close to optimality and are therefore left in the tables.

Name	Nodes	Edges	Non-Dominated paths

Arnes	34	94	7408
AsnetAm	65	158	28354
Bellcanada	48	130	29046
Bellsouth	51	132	31368
BeyondTheNetwork	53	130	30172
Bics	33	96	14316
BtLatinAmerica	45	100	7650
Canerie	32	82	8198
Cernet	41	118	5853
Cesnet200511	39	88	7478
Cesnet200603	39	88	7478
Cesnet200706	44	102	9514
Cesnet201006	52	126	18822
Chinanet	42	132	25231
Cudi	51	104	4678
Cwix	36	82	8922
DeutscheTelekom	30	110	12378
Dfn	58	174	64559
Digex	31	76	6828
Ernet	30	64	3770
Evolink	37	90	8738
Forthnet	62	124	5220
Garr200902	54	142	18770
Garr200908	54	136	18410
Garr200909	55	138	19035
Garr200912	54	136	18509
Garr201001	54	136	20215
Garr201003	54	142	18903
Garr201004	54	142	18903
Garr201005	55	144	19445
Garr201007	55	148	19610
Garr201008	55	148	19610
Garr201010	56	150	20232
Garr201012	56	150	21418
Garr201101	56	152	22148
Garr201102	57	154	23054
Garr201103	58	162	24099
Garr201104	59	166	26387
Garr201105	59	168	25387
Garr201107	59	170	25446
Garr201108	59	170	25446
Garr201109	59	172	26109
Garr201110	59	174	25149
Garr201111	60	174	23968
Garr201112	61	178	24876
Garr201201	61	178	25413

Geant2009	34	104	16240
Geant2012	40	122	15589
Globenet	67	226	84198
Grnet	37	94	4523
GtsCzechRepublic	32	66	4006
GtsHungary	30	62	2874
GtsPoland	33	74	8576
GtsSlovakia	35	74	5082
HiberniaGlobal	55	162	60773
Internode	66	156	15760
Intranetwork	39	106	14186
IowaStatewideFiberMap	33	82	10448
Iris	51	128	33076
KentmanJan2011	38	78	2216
LambdaNet	42	92	8952
Litnet	43	86	3256
Missouri	67	166	79040
Myren	37	80	4238
Ntelos	47	122	30480
Palmetto	45	140	34786
PionierL1	36	82	9982
PionierL3	38	104	11592
Renater2004	30	72	5572
Renater2008	33	86	10390
Renater2010	43	112	25136
Rnp	31	68	3868
Roedunet	42	100	2425
RoedunetFibre	48	104	15398
Sanet	43	90	7912
Surfnet	50	146	40869
SwitchL3	42	126	19084
Tinet	53	178	68784
UsSignal	61	158	72078
Uunet	49	168	53396
Xspedius	34	98	17510

Table A.4: Summary of the topologies' sizes

Name	α (NR)	α (R)	$\hat{\alpha}$ (NR)	$\hat{\alpha}$ (R)	t (s) (NR)	t (s) (R)
Arnes	0.903	1.078	1.429	1.203	0.53	118.86
AsnetAm	0.900	0.900	1.062	1.062	1.08	15.78
Bellcanada	0.900	0.900	0.933	0.917	2.14	1935.93
Bellsouth	0.900	0.944	1.012	1.012	1.44	16.44
BeyondTheNetwork	0.900	0.962	1.135	1.011	2.67	3604.38
Bics	0.900	0.912	1.022	0.942	0.98	546.18

BtLatinAmerica	0.900	0.900	0.942	0.942	0.42	3.76
Canerie	0.900	0.995	1.071	0.978	0.60	3603.41
Cernet	0.940	1.049	1.284	1.248	0.41	15.85
Cesnet200511	0.900	0.998	1.175	1.072	0.41	86.38
Cesnet200603	0.900	1.087	1.346	1.175	0.44	299.44
Cesnet200706	0.900	1.015	1.307	1.039	0.46	693.00
Cesnet201006	0.900	1.054	1.368	1.091	1.00	542.11
Chinanet	0.900	1.132	1.708	1.517	4.98	1342.94
Cudi	0.900	0.900	1.094	1.094	0.19	2.95
Cwix	0.899	0.992	1.264	1.039	0.69	3601.91
DeutscheTelekom	0.900	0.967	1.190	0.996	0.82	3602.30
Dfn	0.900	0.930	1.142	1.006	5.87	3606.38
Digex	0.899	0.899	0.963	0.931	0.54	107.23
Ernet	0.900	0.900	1.148	1.148	0.23	1.49
Evolink	0.900	0.900	1.070	0.985	0.49	16.50
Forthnet	0.900	0.900	1.116	1.146	0.27	3.83
Garr200902	0.900	0.989	1.317	1.270	1.41	40.86
Garr200908	0.900	0.974	1.604	1.156	1.10	3603.21
Garr200909	0.957	1.134	1.467	1.296	1.49	3603.37
Garr200912	0.901	1.119	1.445	1.150	2.07	3603.33
Garr201001	0.964	1.101	1.597	1.199	1.35	1855.07
Garr201003	0.900	1.075	1.502	1.130	1.49	3603.45
Garr201004	0.928	1.009	1.448	1.248	1.34	290.25
Garr201005	0.900	1.004	1.292	1.294	1.03	81.64
Garr201007	0.900	0.938	1.386	1.194	0.87	24.75
Garr201008	0.900	1.126	1.380	1.178	1.23	1505.19
Garr201010	0.900	1.037	1.665	1.192	1.61	3604.41
Garr201012	0.900	0.929	1.144	1.167	1.25	120.25
Garr201101	0.900	1.074	1.290	1.153	1.80	2561.20
Garr201102	0.900	0.900	1.146	1.140	0.98	10.28
Garr201103	0.900	0.900	1.090	1.090	1.08	9.35
Garr201104	0.900	0.990	1.238	1.102	1.13	13.66
Garr201105	0.900	1.087	1.242	1.126	1.13	2587.96
Garr201107	0.900	0.962	1.343	1.155	1.05	64.01
Garr201108	0.900	0.957	1.253	1.118	1.14	70.87
Garr201109	0.900	1.010	1.217	1.080	1.24	699.77
Garr201110	0.900	1.019	1.365	1.233	1.18	68.69
Garr201111	0.900	0.981	1.279	1.230	1.31	72.72
Garr201112	0.900	0.901	1.183	1.042	1.13	95.98
Garr201201	0.900	1.018	1.255	1.047	1.63	1231.63
Geant2009	0.900	0.938	1.312	1.008	1.52	3602.60
Geant2012	0.900	0.960	1.213	1.015	1.21	3602.63
Globenet	0.900	0.916	0.956	0.932	5.24	3612.85
Grnet	0.900	0.900	1.248	1.175	0.25	3.33
GtsCzechRepublic	0.900	0.916	1.067	0.977	0.41	28.19
GtsHungary	0.900	0.927	1.282	1.137	0.20	5.36

GtsPoland	0.900	0.906	1.012	0.969	0.79	3604.16
GtsSlovakia	0.900	0.900	1.326	1.141	0.32	5.21
HiberniaGlobal	0.900	0.906	0.927	0.918	4.90	3607.54
Internode	0.990	1.039	1.174	1.073	0.52	360.83
Intranetwork	0.900	0.900	0.935	0.935	1.00	13.52
IowaStatewideFiberMap	0.900	0.923	1.048	0.976	1.11	3602.22
Iris	0.900	0.900	0.942	0.922	2.29	771.06
KentmanJan2011	0.900	0.900	1.298	1.298	0.17	1.47
LambdaNet	0.900	0.920	0.984	0.946	0.98	301.87
Litnet	0.900	1.009	1.335	1.226	0.17	4.63
Missouri	0.900	0.902	0.925	0.917	4.92	3611.27
Myren	0.900	1.239	1.686	1.616	0.35	8.32
Ntelos	0.900	0.900	0.928	0.928	2.78	740.43
Palmetto	0.900	0.901	0.961	0.930	2.24	1228.94
PionierL1	0.900	0.920	0.972	0.944	0.72	3602.69
PionierL3	0.901	0.971	1.234	1.029	1.30	3602.74
Renater2004	0.951	1.147	1.556	1.270	0.57	217.62
Renater2008	0.900	0.992	1.317	1.069	0.86	617.36
Renater2010	0.900	0.917	1.076	0.962	2.73	3604.33
Rnp	0.900	0.900	0.994	0.994	0.25	2.59
Roedunet	1.041	1.041	1.795	1.633	0.17	2.12
RoedunetFibre	0.900	0.900	0.949	0.949	0.91	24.57
Sanet	0.900	0.945	1.014	0.970	0.97	594.83
Surfnet	0.900	0.902	1.038	0.969	3.90	393.91
SwitchL3	0.939	0.954	1.265	1.197	0.91	25.03
Tinet	0.900	0.908	0.971	0.941	8.47	1492.72
UsSignal	0.900	0.907	0.953	0.929	12.44	3609.73
Uunet	0.899	0.915	1.173	0.965	4.91	924.93
Xspedius	0.900	0.908	1.067	0.952	1.31	3602.23

Table A.5: Summary of results across topologies using $\Gamma = 2$. Explanation of the columns is provided at the beginning of the appendix.

Name	α (NR)	α (R)	$\hat{\alpha}$ (NR)	$\hat{\alpha}$ (R)	t (s) (NR)	t (s) (R)
Arnes	0.903	1.085	1.628	1.421	0.53	163.67
AsnetAm	0.900	0.900	1.162	1.162	1.08	12.55
Bellcanada	0.900	0.900	0.980	0.941	2.14	2675.28
Bellsouth	0.900	1.029	1.120	1.120	1.44	23.45
BeyondTheNetwork	0.900	0.962	1.259	1.111	2.67	3603.88
Bics	0.900	0.906	1.184	0.999	0.98	477.91
BtLatinAmerica	0.900	0.900	0.986	0.987	0.42	3.87
Canerie	0.900	0.993	1.245	1.078	0.60	3602.97
Cernet	0.940	1.083	1.534	1.470	0.41	12.93
Cesnet200511	0.900	0.974	1.396	1.225	0.41	3601.85

Cesnet200603	0.900	1.034	1.519	1.331	0.44	3602.23
Cesnet200706	0.900	1.081	1.481	1.227	0.46	1331.42
Cesnet201006	0.900	0.931	1.505	1.232	1.00	3603.35
Chinanet	0.900	0.989	1.756	1.517	4.98	3003.65
Cudi	0.900	0.900	1.249	1.249	0.19	3.35
Cwix	0.899	1.060	1.378	1.159	0.69	3602.46
DeutscheTelekom	0.900	1.015	1.399	1.103	0.82	3602.42
Dfn	0.900	0.993	1.283	1.093	5.87	3607.02
Digex	0.899	0.901	1.033	0.969	0.54	1496.03
Ernet	0.900	0.900	1.341	1.341	0.23	1.64
Evolink	0.900	0.996	1.207	1.071	0.49	93.76
Forthnet	0.900	0.900	1.209	1.267	0.27	3.23
Garr200902	0.900	0.973	1.677	1.393	1.41	3603.39
Garr200908	0.900	1.006	1.655	1.323	1.10	3603.13
Garr200909	0.957	1.085	1.753	1.520	1.49	3603.20
Garr200912	0.901	0.977	1.701	1.356	2.07	3603.52
Garr201001	0.964	1.084	1.685	1.364	1.35	3603.47
Garr201003	0.900	0.980	1.678	1.354	1.49	3603.47
Garr201004	0.928	1.034	1.605	1.414	1.34	3603.38
Garr201005	0.900	0.979	1.474	1.277	1.03	3603.74
Garr201007	0.900	0.964	1.540	1.287	0.87	27.36
Garr201008	0.900	0.935	1.521	1.356	1.23	3603.50
Garr201010	0.900	1.230	1.750	1.390	1.61	3603.79
Garr201012	0.900	1.035	1.418	1.219	1.25	3603.66
Garr201101	0.900	0.988	1.527	1.354	1.80	3603.54
Garr201102	0.900	0.900	1.387	1.333	0.98	9.12
Garr201103	0.900	0.900	1.267	1.267	1.08	8.84
Garr201104	0.900	0.962	1.403	1.301	1.13	12.23
Garr201105	0.900	1.025	1.499	1.271	1.13	3604.43
Garr201107	0.900	1.077	1.447	1.320	1.05	117.47
Garr201108	0.900	1.075	1.407	1.297	1.14	42.46
Garr201109	0.900	0.953	1.369	1.241	1.24	3604.58
Garr201110	0.900	0.975	1.531	1.322	1.18	3604.35
Garr201111	0.900	1.239	1.401	1.353	1.31	3605.02
Garr201112	0.900	0.917	1.362	1.178	1.13	484.95
Garr201201	0.900	0.995	1.449	1.210	1.63	3604.59
Geant2009	0.900	0.945	1.597	1.146	1.52	3602.36
Geant2012	0.900	1.020	1.347	1.155	1.21	3602.10
Globenet	0.900	0.941	1.030	0.970	5.24	3611.91
Grnet	0.900	0.900	1.458	1.335	0.25	2.57
GtsCzechRepublic	0.900	0.928	1.218	1.077	0.41	175.11
GtsHungary	0.900	1.000	1.469	1.322	0.20	6.32
GtsPoland	0.900	0.912	1.135	1.041	0.79	3601.91
GtsSlovakia	0.900	0.900	1.469	1.254	0.32	4.61
HiberniaGlobal	0.900	0.911	0.962	0.951	4.90	3607.41
Internode	0.990	1.099	1.316	1.160	0.52	3603.81

Intranetwork	0.900	0.900	0.980	0.980	1.00	8.66
IowaStatewideFiberMap	0.900	0.950	1.203	1.076	1.11	3601.77
Iris	0.900	0.900	0.982	0.947	2.29	519.05
KentmanJan2011	0.900	0.900	1.475	1.475	0.17	1.56
LambdaNet	0.900	0.913	1.059	0.997	0.98	564.67
Litnet	0.900	0.977	1.587	1.447	0.17	27.21
Missouri	0.900	0.902	0.958	0.935	4.92	3610.59
Myren	0.900	1.049	1.737	1.677	0.35	51.15
Ntelos	0.900	0.900	0.963	0.963	2.78	529.41
Palmetto	0.900	0.913	1.044	0.972	2.24	271.82
PionierL1	0.900	0.943	1.060	1.006	0.72	3602.53
PionierL3	0.901	1.013	1.408	1.177	1.30	3602.66
Renater2004	0.951	1.206	1.701	1.459	0.57	3601.31
Renater2008	0.900	0.961	1.466	1.174	0.86	989.66
Renater2010	0.900	0.939	1.200	1.032	2.73	3603.74
Rnp	0.900	0.900	1.127	1.089	0.25	2.27
Roedunet	1.041	1.041	1.844	1.832	0.17	2.03
RoedunetFibre	0.900	0.900	0.997	0.997	0.91	16.65
Sanet	0.900	0.970	1.147	1.040	0.97	3602.38
Surfnet	0.900	0.900	1.181	1.041	3.90	3605.05
SwitchL3	0.939	0.939	1.460	1.324	0.91	14.72
Tinet	0.900	0.909	1.034	1.004	8.47	3607.81
UsSignal	0.900	0.909	1.005	0.968	12.44	3609.80
Uunet	0.899	0.955	1.453	1.051	4.91	3606.51
Xspedius	0.900	0.904	1.220	1.018	1.31	3602.50

Table A.6: Summary of results across topologies using $\Gamma = 5$. Explanation of the columns is provided at the beginning of the appendix.