On Improving the Accuracy of OSPF Traffic Engineering[*]

Gábor Rétvári, József J. Bíró, and Tibor Cinkler

High Speed Networks Laboratory,
Department of Telecommunications and Media Informatics,
Budapest University of Technology and Economics,
H-1117, Magyar Tudósok körútja 2., Budapest, Hungary
`{retvari, biro, cinkler}@tmit.bme.hu`

Abstract. The conventional forwarding rule used by IP networks is to always choose the path with the shortest length – in terms of administrative link weights assigned to the links – to forward traffic. Lately, it has been proposed to use shortest-path-first routing to implement Traffic Engineering in IP networks, promising with a big boost in the profitability of the legacy network infrastructure. The idea is to set the link weights so that the shortest paths, and the traffic thereof, follow the paths designated by the operator. Unfortunately, traditional methods to calculate the link weights usually produce a bunch of superfluous shortest paths, often leading to congestion along the unconsidered paths. In this paper, we introduce and develop novel methods to increase the accuracy of this process and, by means of extensive simulations, we show that our proposed solution produces remarkably high quality link weights.

Keywords: OSPF, traffic engineering, linear programming, shortest paths.

1   Introduction

OSPF Traffic Engineering (OSPF TE) exploits the potential of the Internet network infrastructure to implement economic and efficient traffic management right at the IP level. IP routers traditionally forward traffic along the shortest path(s) towards the destination, where the path length is computed in terms of an administrative weight associated with network links, and load-balancing is achieved by the optional Equal-Cost-MultiPath (ECMP) technique, that allows the traffic to be split roughly evenly amongst equal cost shortest paths. OSPF TE basically means the careful manipulation of OSPF link weights aiming towards balanced traffic distribution and reduced congestion [1], [2], [3].

The process model of OSPF TE is as follows. A dedicated TE network component participates in the signaling of the Open Shortest Path First (OSPF, [4]) or Intermediate-System-to-Intermediate System (IS-IS) routing protocol. Based

---

[*] This work has been done as a part of the European sixth framework research project IP NOBEL (www.ist-nobel.org).

on the routing information gathered from the network it becomes possible to compute OSPF link weights, so that the resultant shortest paths manifest some sophisticated TE goal. After the link weights are distributed back to the routers, the traffic in the network will follow the paths assigned by the traffic engineer, leading to, hopefully, more optimal network utilization and better user experience. And all this happens without modifying the basic operation of OSPF/IS-IS in any regards. This is in sharp contrast to the conventional models for traffic engineering, where TE functionality is delegated to a dedicated connection-oriented infrastructure, that has to be purchased, operated and maintained separately from the IP layer.

A fundamental restriction of OSPF TE is that all traffic must follow the shortest paths in the network. Only if a path set can be represented as a set of shortest paths, that is, positive, integer-valued link weights exist over which the paths are all shortest paths, it can be used in conjunction with OSPF TE. Notably, this limitation does not turn out to be overly restrictive, because any optional path set is either shortest path representable by itself, or otherwise, by eliminating redundant loops, it can be reduced to a shortest path representable one. Moreover, the reduced path set is not only capable to satisfy the same bandwidth requirements as the original path set, but it is also strictly shorter in terms of the overall number of edges traversed. This ground-breaking result, which is due to Wang *et al.* [5], immediately catapulted OSPF TE into the focus of interests, since it suggests that the range of potential path assignment strategies compatible with OSPF routing is much wider than anyone would have expected previously.

OSPF TE is generally NP-hard [1]. Thus, it is common to subdivide the process into two, mostly independent phases. In the first phase, a set of "good" paths is assigned to each ingress-egress router pair in the network (these will be referred to as *sessions* hereafter), and then, these paths are mapped to shortest paths. In a predecessor of this paper, [6], we pointed out that it is the first phase that hides the origin of exponential complexity of OSPF TE. Thus, to select the path set it is plausible to invoke some quick heuristics, such as for example the widest-shortest-path algorithm [7], which, in some way or another, promise with improving the performance of OSPF routing. What remains to be done is to map this designated path set to shortest paths with the greatest accuracy that is achievable.

Unfortunately, it has gone mostly unnoticed that this process is highly prone to ambiguity, and the resultant set of shortest paths usually contains a plethora of additional (and, unfortunately, completely superfluous) paths besides the ones designated in the first phase. Therefore, the carefully selected and fine-tuned traffic engineered path set often deteriorates into a bunch of overlapping and interfering shortest paths. In [6] we showed that, due to the adverse interference caused by traffic routed to those additional paths, the useful throughput might fall to an arbitrary small fraction of the optimal throughput realizable by OSPF TE. Thus, it is essential to devise methods to restrict the number of paths in

a shortest path representation to the fewest possible. This is precisely the main purpose of the research work presented in this paper.

In Section 2, after briefly reviewing the mathematical model, we show an illustrative example, that exhibits all the shortcomings of the traditional OSPF TE methodology. Then, in Section 3, we overview the theory related to shortest path representability and in Section 4, we introduce some new concepts with the aim to restrict the number of superfluous shortest paths to the bare minimum. We also give a polynomial time algorithm, which, according to the simulation results presented in Section 5, proves itself remarkably useful in practice. Finally, in Section 6, we draw the conclusions of our work.

Due to space limitations, in the sequel we shall confine ourselves to the most basic theory. For an in depth exposition the reader is referred to [8].
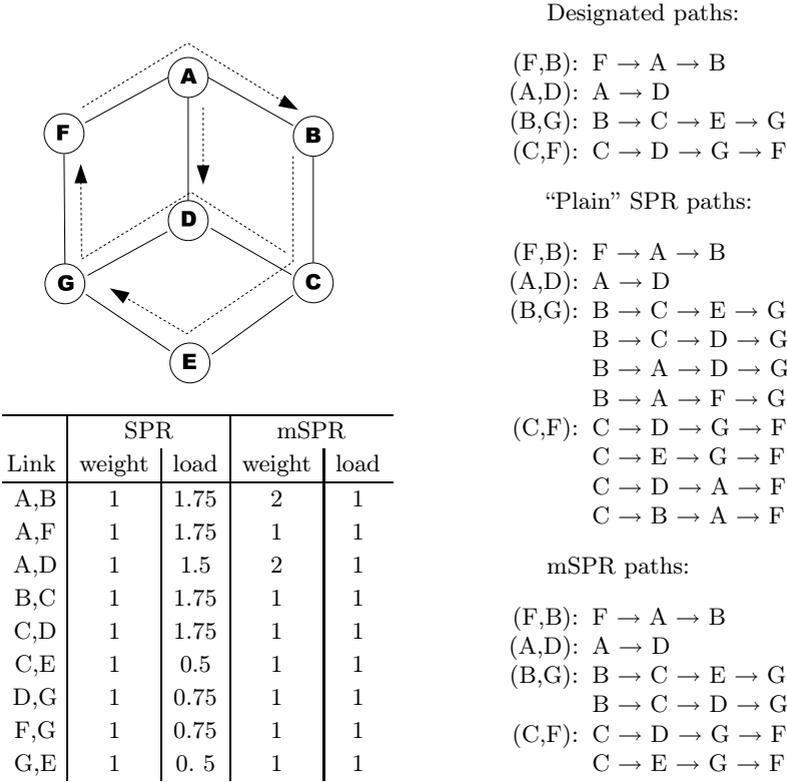
2 An Illustrative Example

In this Section, first we introduce the relevant notation. Let the network be described by the directed graph $G(V, E)$, formed by the set of nodes V ($|V| = n$) and the set of edges E ($|E| = m$). Let N be the corresponding node-arc incidence matrix. Furthermore, suppose that we are given a set of source-destination pairs, or sessions, $(s_k, d_k) : k \in \mathcal{K}$ and, provisioned between these source-destination pairs, some set of *designated paths* $\mathcal{P}_k$. A path in $\mathcal{P}_k$ is single $s_k \to d_k$ ($s_k \in V$, $d_k \in V, k \in \mathcal{K}$) path, say, P, given by its consecutive edges: $P := \{(v_i, v_{i+1}) \in E : i = 1 \ldots L_P, v_1 = s_k, v_{L_P+1} = d_k\}$, where L_P denotes the length of P. In vector-form, the *support* of P is a column m-vector p, such that the component corresponding to link (i, j) is 1 if $(i, j) \in P$ and zero otherwise. We generally assume that paths do not contain loops, and that all d_ks are distinct, which assures that IP maintains a separate entry in the routing table for each session. Let $t_k = |\mathcal{P}_k|$ denote the number of designated paths for session k.

The collection of all the designated paths is given as $\mathcal{P} = \cup_{k \in \mathcal{K}} \mathcal{P}_k$, whose support is $p = \sum_{P_k \in \mathcal{P}} p^k$ (here, p^ks are the supports of the $\mathcal{P}_k$ sets). We say that a path set $\mathcal{P}'$ is equivalent to another path set $\mathcal{P}''$, that is, $\mathcal{P}' \equiv \mathcal{P}''$ if $E(\mathcal{P}') = E(\mathcal{P}'')$, where $E(\mathcal{P}) = \{(i, j) \in E : \exists P \in \mathcal{P}, \text{ so that } (i, j) \in P\}$. Similarly, $\mathcal{P}' \subseteq \mathcal{P}''$ if $E(\mathcal{P}') \subseteq E(\mathcal{P}'')$.

Let w_{ij} be an additive, positive valued weight associated with each network link (i, j). Gather w_{ij}s into a row m-vector w. The length of a path P (of support p) over the link weights $\mathcal{W} = \{w_{ij}\}$ is defined as $W(P) = wp$. The set of shortest paths over $\mathcal{W}$ is denoted by $\mathcal{P}(\mathcal{W})$. In the remainder of this paper, link weights will be sought for in the form $w = \xi + \omega$, $\omega \geq 0$, where ξ is an strictly positive m-vector introduced with the sole purpose of separating w away from zero.

Now, we move on to introduce an illustrative example to demonstrate the main points of the paper. Suppose that we are given the network[1] depicted in Fig. 1, which we adopted from [5]. All edge capacities equal to 1. Furthermore,

---

[1] Note that, for the sake of simplicity, we shall use an undirected network in this example. However, the theory will be formulated for directed networks later on, which models the real case more thoroughly.

Designated paths:

(F,B): F → A → B
(A,D): A → D
(B,G): B → C → E → G
(C,F): C → D → G → F

"Plain" SPR paths:

(F,B): F → A → B
(A,D): A → D
(B,G): B → C → E → G
       B → C → D → G
       B → A → D → G
       B → A → F → G
(C,F): C → D → G → F
       C → E → G → F
       C → D → A → F
       C → B → A → F

mSPR paths:

(F,B): F → A → B
(A,D): A → D
(B,G): B → C → E → G
       B → C → D → G
(C,F): C → D → G → F
       C → E → G → F

	SPR		mSPR	
Link	weight	load	weight	load
A,B	1	1.75	2	1
A,F	1	1.75	1	1
A,D	1	1.5	2	1
B,C	1	1.75	1	1
C,D	1	1.75	1	1
C,E	1	0.5	1	1
D,G	1	0.75	1	1
F,G	1	0.75	1	1
G,E	1	0. 5	1	1

Fig. 1. Sample network topology, the set of designated paths and shortest paths in different representations and a table summarizing the weight and the emergent load of each link, assuming ECMP load-balancing

we are given 4 source-destination pairs, (namely, (F, B), (A, D), (B, G) and (C, F)) between which a set of paths, each of capacity 1, is assigned as indicated in the figure. The paths were provisioned as to assure that all links are filled to capacity. Our task is then to achieve, by the careful setting of the link weights, that all the shortest paths follow the designated paths.

Our first observation is that, by coincidence, all the designated paths are actually minimum-hop paths (so they traverse the least hops possible). Therefore, setting all the link weights uniformly to 1 assures that all the designated paths become shortest paths. This setting obviously conforms to the following conventional definition of shortest path representability [5]:

Definition 1. *A path set $\mathcal{P}$ is shortest path representable (SPR), if there exists a positive weight setting $\mathcal{W}$, such that $\mathcal{P} \subseteq \mathcal{P}(\mathcal{W})$.*

We call the attention of the reader to a subtlety in the definition. Namely, we do not require the equivalence of $\mathcal{P}$ and its shortest path representation $\mathcal{P}(\mathcal{W})$.

We only require $\mathcal{P}$ to be a subset of $\mathcal{P}(\mathcal{W})$, and quite often this is precisely the case. In our example, the "plain" shortest path representation contains significantly more paths than $\mathcal{P}$. For instance, in the case of session (B, G) not just the designated path, but also three other paths have become shortest paths. This, according to the ECMP load-balancing scheme, implies that the traffic of session (B, G) will be distributed evenly to the shortest paths, and the additional traffic directed to the superfluous paths will substantially overload some of their links. To avoid this, it is crucial to eliminate as many superfluous paths from the representation as possible. Perhaps, the most straightforward strengthening of Definition 1 would be the following:

Definition 2. *A path set $\mathcal{P}$ is perfectly shortest path representable (pSPR), if there exists a positive weight setting $\mathcal{W}$, such that $\mathcal{P} \equiv \mathcal{P}(\mathcal{W})$.*

Unfortunately, very often one can not achieve the total equivalence of the designated path set and the representation. Instead, the best one can hope for is to reduce the number of paths in the representation to the bare minimum by dropping the most paths possible. In other words, a minimal shortest path representation $\mathcal{P}_{min}$ is constituted of only those paths, which participate in *all* the shortest path representations.

Definition 3. *A weight set $\mathcal{W}_{min}$ implements a minimal shortest path representation (mSPR) of a path set $\mathcal{P}$, if for each weight set $\mathcal{W}$: $\mathcal{P} \subseteq \mathcal{P}(\mathcal{W}) \Rightarrow \mathcal{P}(\mathcal{W}_{min}) \subseteq \mathcal{P}(\mathcal{W})$. We denote $\mathcal{P}(\mathcal{W}_{min})$ as $\mathcal{P}_{min}$.*

In Fig. 1, we indicated a possible choice of weights that implements a minimal representation, and the set of shortest paths it induces. Observe that we still have superfluous shortest paths (exactly one for both (B,G) and (C,F)), but, interestingly, these paths can never be dropped from the shortest path representation. This is because, if we wanted to eliminate for example path B → C → D → G from the set of shortest paths of (B,G), we would need to increase the weight of either link (C,D) or (D,G). But in this case, the designated path C → D → G → F would cease to be a shortest path for session (C,F). Finally, it is noteworthy that using the minimal representation we could avoid to overload any of the links in the network.

   In the remaining part of this paper, we shall argue that the concept of minimal representations is a remarkably useful one. First, it manifests an interesting theoretical lower bound on narrowing a shortest path representation, and, as shall be shown, this lower bound is well-defined. As a corner case, it contains perfect representations. Furthermore, a minimal representation can always be obtained in polynomial time, though, it might pose significantly more computational burden than in general.

3   A Flow-Theoretic Approach

Below, we give a brief overview of the theory of Shortest Path Representability. The fundamental result that characterizes shortest path representable paths is as follows [5], [8]:

Proposition 1. *Let $\mathcal{P} = \cup_{k \in \mathcal{K}} \mathcal{P}_k$ be a set of paths for some set of sessions $\mathcal{K}$, and let p^k be the support of $\mathcal{P}_k$ for each $k \in \mathcal{K}$. Then, $\mathcal{P}$ is representable as shortest paths, if and only if the setting $x^k = p^k$ yields an optimal feasible solution to the* primal fundamental LP of $\mathcal{P}$, P-LP($\mathcal{P}$)*:*

$$\sum_{k \in \mathcal{K}} \xi p^k - \min \sum_{k \in \mathcal{K}} \xi x^k : \quad Nx^k = t^k \quad \forall k \in \mathcal{K} \tag{1}$$

$$\sum_{k \in \mathcal{K}} x^k \leq p \tag{2}$$

$$x^k \geq 0 \quad \forall k \in \mathcal{K} \tag{3}$$

where t^k, the vector of the number of designated paths for k, is defined as:

$$(t^k)_v = \begin{cases} -t_k & \text{if } v = s_k \\ t_k & \text{if } v = d_k \\ 0 & \text{otherwise} \end{cases}$$

In this case, the optimal objective is zero. If, in contrast, the optimal objective is positive, then $\mathcal{P}$ is not SPR.

The most important observation regarding the fundamental LP is that – apart from a constant term – it basically is a minimum cost multicommodity flow problem. Constraints (1) give the flow conservation constraints with respect to t_k. The bundle constraints (2) restrict the sum of the arc-flows to remain under $\mathcal{P}$'s support, p_{ij}, at every link. In fact, p acts as some sort of link capacity. Finally, arc-flows are required to be non-negative by (3). Under the hood, P-LP($\mathcal{P}$) can be interpreted as the task to reallocate the paths in the network, such that after the reallocation the number of paths placed on a link does not exceed the number of paths using that link in $\mathcal{P}$. If this can be done such that the length of the new path set (in terms of ξ) is less than that of $\mathcal{P}$, then the path set is not SPR, because it contains loops.

The proof of the Proposition proceeds as follows. Let π^k be a row n-vector for each $k \in \mathcal{K}$, which denotes the dual variables for constraints (1). Similarly, let ω (a row m-vector) be the dual for the constraints (2). Now, for the dual variables it holds that:

$$\forall k \in \mathcal{K}, \forall (i,j) \in E : \pi_j^k - \pi_i^k \geq \xi_{ij} + \omega_{ij}, \qquad \omega_{ij} \geq 0 . \tag{4}$$

Hence, we could interpret π^k as distance labels, or *node potentials*, so that π_v^k defines an upper bound on the shortest distance from the source node s_k to any node v over the positive-valued weight set $\mathcal{W} = \{\xi_{ij} + \omega_{ij}\}$. The *Shortest Path Optimality Conditions* [9] require that a path P_k is shortest path from s_k to d_k, if and only if (4) holds with strict equality at all $(i,j) \in P_k$:

$$p_{ij}^k > 0 \Rightarrow \pi_j^k - \pi_i^k = \xi_{ij} + \omega_{ij} . \tag{5}$$

But p_{ij}^k is the dual variable corresponding to the constraints (4). Therefore, it follows from complementary slackness that (5) holds true, if and only if $[p^k]$ is optimal to P-LP($\mathcal{P}$).

Formulating the SPR problem as a multicommodity flow problem provides a wealth of options to easily solve it [10]. Furthermore, Proposition 1 gives important insights into the very nature of the problem, which we shall exploit in the foregoing discussions in our quest for improving the accuracy of shortest path representations.

4   Minimum and Perfect Shortest Path Representations

In the previous Section, we have seen that in order to obtain the link weights that represent a set of paths $\mathcal{P}$ as shortest paths, one needs to solve the dual of P-LP($\mathcal{P}$). This yields the link weights in the form: $w_{ij} = \xi_{ij} + \omega_{ij}$. However, in the vast majority of the cases there arises a large number of alternative optimal solutions, both for the primal and the dual problem, probably due to highly degenerate nature of the feasible region. The following important result characterizes the set of paths in a shortest path representation, in terms of the different alternative optimal solutions of P-LP($\mathcal{P}$).

Theorem 1. *The set of paths in the minimal representation is spanned by the alternative optimal solutions of the primal fundamental LP.*

More formally, given a set of paths $\mathcal{P}$ and some $s_k \to d_k$ path P for some session $k \in \mathcal{K}$: $P \in \mathcal{P}_{\min}$ if and only if there exists some $[x^k]$ optimal feasible solution to P-LP(P), such that $\forall (i,j) \in P : x_{ij}^k > 0$.

Proof (Sketch). Due to the *Complementary Theorem of Linear Programming* [11, p. 310, Exercise 6.39], there exists some $[x^k]$ optimal feasible solution to P-LP($\mathcal{P}$), so that $\forall (i,j) \in P : x_{ij}^k > 0$, if and only if (4) holds with strict equality at all links of P in *all* the dual solutions. Noting that, for an SPR path set, the optimal region and the feasible region of the dual coincide, this precisely means that any such P will be a shortest path over any SPR link weights.   □

Regarding our example in Fig 1, the existence of additional paths in the minimal representation can be attributed to the fact that, for session (B,G) and (C,F), the subpaths of the designated paths between nodes C and G can be swapped, and both configurations supply a potential optimal feasible solution to the fundamental LP.

The significance of Theorem 1 is manifold. First, it implies that the concept of minimal representations is well-defined, because the set of alternative optimal feasible solutions of P-LP($\mathcal{P}$) is also well-defined. Therefore, the minimal representation is the intrinsic property of the path set itself, and there is *theoretically* no way to obtain a more precise shortest path representation. Another interesting corollary is that, for the single-path routing case, Theorem 1 gives a nice characterization of perfect representations:

Corollary 1. *Suppose that some path set $\mathcal{P}$ contains only one path for each distinct $(s_k, d_k) : k \in \mathcal{K}$. Now, $\mathcal{P}$ is perfectly shortest path representable, if and only if $[p^k]$ is the unique optimal feasible solution of P-LP($\mathcal{P}$).*

This result explains, why it is relatively rare for a path set to be pSPR: the support of $\mathcal{P}$ must be a unique optimizer of the corresponding fundamental LP, which, as it is usual with minimum cost multicommodity flow problems, is not a particularly frequent occurrence.

Finally, we construct an algorithm to search for the minimal representation. For this, first we introduce some more notation. Let $v^k \geq 0$ be a row m-vector of slack-variables for each session $k \in \mathcal{K}$, so that v^k is complementary to x^k in the fundamental LP. With this notation, we can write (4) as $\pi_j^k - \pi_i^k + v_{ij}^k = \xi_{ij} + \omega_{ij}$.

Corollary 2. *For some $s_k \to d_k$ path P_k: $P_k \notin \mathcal{P}_{\min}$, if and only if there exists some optimal feasible solution of the dual of P-LP($\mathcal{P}$), so that $v_{ij}^k > 0$ for some $(i,j) \in P_k$.*

Proof. From Theorem 1, $P_k \notin \mathcal{P}_{\min}$ if and only if P_k traverses at least one link, say (i,j), so that $x_{ij}^k = 0$ for each optimal feasible solution of P-LP($\mathcal{P}$). But this, from the Complementary Theorem of Linear Programming, precisely means that there exists a dual optimal solution with $v_{ij}^k > 0$, so P_k is not a shortest path according to the Shortest Path Optimality Conditions. □

In [8], we show that $v_{ij}^k > 0$ also implies that there exists an *optimal ray d* in the set of optimal feasible solutions of the dual, such that the entry in d corresponding to v_{ij}^k is strictly positive. Now, our algorithm to search for the minimal representation is based on the idea that if we elevate as many slack variables v_{ij}^k from zero as possible, while simultaneously assuring that all the designated paths remain shortest paths, we eventually obtain a minimal representation. First, we add a constraint $\sum_{k \in \mathcal{K}} v^k p^k = 0$ to the dual problem, which explicitly requires that all slack variables corresponding to the designated paths are bound to zero. Hence, solving the problem will yield yet another SPR link weight set. Second, we perturb the objective function vector by setting the cost of some v_{ij}^k to an arbitrary positive value and all other costs to zero. Finally, we set the direction of the optimization to maximization. Now, either the perturbed LP is bounded, in which case no appropriate directions exist for v_{ij}^k, or otherwise it is unbounded. Let the ray causing the unboundedness be d. Notably, d has strictly positive surplus in the position corresponding to v_{ij}^k (otherwise, the problem might not have become unbounded) and it has non-negative surplus corresponding to all other slack variables due to the non-negativity constraint imposed on the slack variables. Hence, moving along d yields a new SPR weight set, but now v_{ij}^k is separated away from zero, so all paths of k traversing (i,j) cease to be shortest paths. Repeating this step for each slack variable yields the *mSPR algorithm*:

INPUT: A designated path set $\mathcal{P}$ and initial costs $\xi > 0$.
OUTPUT: Link weight set $\mathcal{W}_{\min}$ that implements a minimal shortest path representation of $\mathcal{P}$.
THE mSPR ALGORITHM:
    1. Solve the dual of P-LP($\mathcal{P}$). If the optimal objective is positive, then conclude that $\mathcal{P}$ is not SPR. Otherwise, let a feasible solution of the dual be $[\hat{\pi}^1, \ldots, \hat{\pi}^K, \hat{v}^1, \ldots, \hat{v}^K, \hat{\omega}]$.

2. For all $k \in \mathcal{K}$ and for all $(i, j) \in E \setminus E(\mathcal{P})$ with $v_{ij}^k = 0$, construct and solve the perturbed dual LP:

$$\max v_{ij}^k : \quad \sum_{k \in \mathcal{K}} v^k p^k = 0 \tag{6}$$

$$\pi^k N + v^k = \xi + \omega \qquad \forall k \in \mathcal{K} \tag{7}$$

$$\omega \geq 0, v^k \geq 0 \quad \forall k \in \mathcal{K} \tag{8}$$

If the perturbed problem is unbounded, then, for some optimal ray d causing the unboundedness:

$$[\hat{\pi}^1, \ldots, \hat{\pi}^K, \hat{v}^1, \ldots, \hat{v}^K, \hat{\omega}] \leftarrow [\hat{\pi}^1, \ldots, \hat{\pi}^K, \hat{v}^1, \ldots, \hat{v}^K, \hat{\omega}] + d .$$

3. Now, $\mathcal{W}_{\min} = \{\xi + \hat{\omega}\}$ implements a minimal shortest path representation of $\mathcal{P}$.
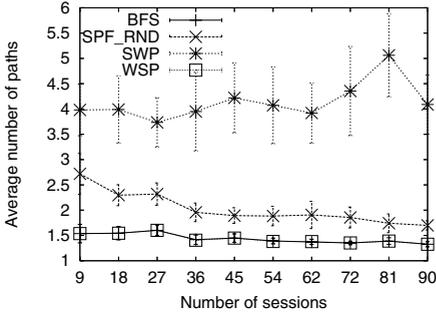
Interestingly, the mSPR algorithm is still a polynomial time algorithm, since the underlying solution technique remains to be linear programming. However, upgrading the definition from "plain" SPR to mSPR results in a significant complexity penalty: while computing an SPR weight set generally requires the solution of one multicommodity flow problem instance, mSPR requires $O(mK)$. Fortunately, we do not have to solve all problems from scratch: given an initial optimal feasible solution we can always start (6)-(8) from this solution, which, in the case of the two-phase simplex algorithm, eliminates all the tedious computations of the first phase. Our experiments suggest that obtaining the optimal rays is usually a matter of some few dozen simplex pivot operations. Furthermore, it is not necessary to compute a separate ray for each slack variable, because very often one ray increases multiple slack variables at once.
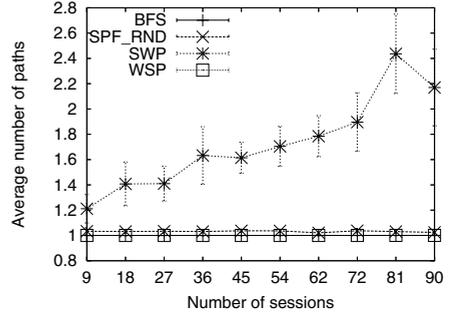
5 Simulation Studies

In this Section, we present the results of extensive simulation studies with the purpose of comparing different concepts of shortest path representability.

We chose to develop our SPR software toolkit in Perl, which – thanks to the unique flexibility and performance – provides an excellent platform to quickly prototype algorithms. For solving the fundamental LP we used the GNU Linear Programming Toolkit, GLPK [12]. Although GLPK does not support network programming, it is reliable, stable and, first and foremost, open source letting us to integrate the SPR toolkit very tightly into the simplex solver[2] We used the random network topology generator, BRITE [13] with the *router-level Waxman-model* ($\alpha = 0.15$, $\beta = 0.2$, $m = 3$) throughout the simulations. The source and destination node of the sessions and the capacity of the links (between 10 and 1024 units) were selected according to independent uniform distributions. Our methodology was to generate 30 random graphs with increasing number of

---

[2] See the `Math::GLPK` project page at `http://qosip.tmit.bme.hu/~retvari/` `Math-GLPK.html`.
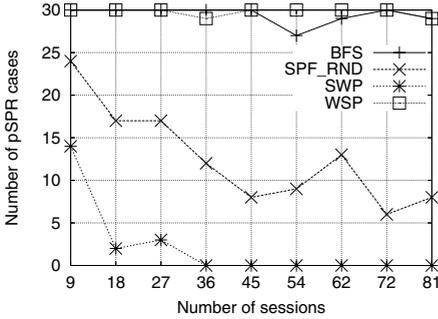
Fig. 2. Average number of paths in the SPR

Fig. 3. Average number of paths in the *minimal SPR*

sessions and average the results (the level of significance was chosen as 95 %). Below, we present the results for networks of 45 nodes.
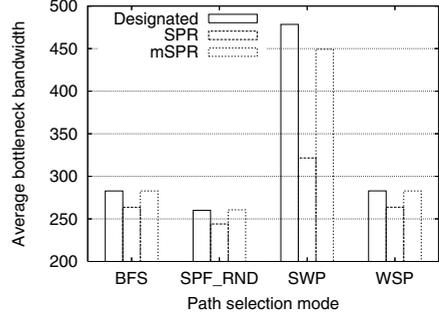
To select the designated paths, we used the breadth-first-search (BFS) algorithm (which manifests minimum hop-count routing), shortest path routing over random weights (SPF_RND) (which represents the case when a network operator chooses the link weights randomly) and the widest-shortest-path (WSP, [7]) and the shortest-widest-path (SWP, [14]) algorithms[3]. One may argue, why would anyone want to compute the shortest path representation of some paths, which are immediately shortest paths by themselves. For example, setting the weight of all links to 1 apparently reproduces BFS paths. The reason is that we want to observe, how many superfluous paths such a naive representation produces by comparing it with the corresponding minimal representation.

The average number of shortest paths per session in the plain SPR is depicted in Fig. 2. Note that the SPR link weights were generated by extreme point solutions of the dual fundamental LP. Our first observation is that such extreme point solutions, due to the relatively huge number of implicit zero-valued slack variables, produce low quality shortest path representations. For the WSP and the BFS paths the representation contains about one and a half times as much paths as the designated path set (which contains exactly one) almost irrespectively of the number of sessions. However, the representation of SPF_RND paths contains more than two paths in average, while this value is 4 for SWP. This suggests that a naive setting of the link weights can easily turn out to be adverse. Even if the designated paths were chosen by a SPR-compatible algorithm, such naive link weights usually only implement a superposition of a huge number random paths, and there are no appropriate mechanisms built into OSPF to select exactly the designated one from amongst them. One needs to carefully tweak the link weights to minimize the ambiguity, and this is exactly what the mSPR algorithm can do for us. As affirmed by Fig. 3, a minimal representation usually

---

[3] Since SWP paths are not guaranteed to be SPR [15], we always substituted the corresponding shortest path representation.

Fig. 4. Number of cases the minimal SPR was perfect

Fig. 5. Average bottleneck bandwidth of the paths

contains only a few superfluous paths up to the point that, except for SWP, it becomes almost perfect in most of the cases.

This observation is further confirmed by Fig. 4, which, as the function of the session number, shows the number of cases out of the total 30 simulations when the minimal representation turned out to be perfect as well. Observe that BFS and WSP paths are almost always pSPR. However, it seems that it is completely hopeless to expect a SWP path set to be pSPR, especially as the number of sessions grows close to the range of the number of nodes in the network.

Finally, we compared the average bottleneck bandwidth of the paths in the designated path set and its plain and minimal representations (see Fig. 5). While this choice obviously omits the interference amongst the sessions, the average bottleneck bandwidth is indeed a good measure of the transmission capacity that is made available by the network for the sessions. On the one hand, SWP is clearly superior in this regard by providing almost twice as much capacity as the other path selection schemes. On the other hand, sharpening the representation apparently improves the capacity of the paths in the representation (by one and a half times in the case of SWP). Our results indicate that in smaller networks the SWP algorithm combined with shortest path routing constitutes a really promising traffic engineering platform. Not just that SWP paths can be mapped quite accurately to shortest paths in this case but, in addition, these paths usually provide an abundance of capacity at the same time.

6   Conclusions

OSPF TE holds tremendous potential to optimize the performance of legacy IP networks. Wang *et al.* showed that whatever path set the traffic engineer assigns for the traffic instances, it is either immediately shortest path representable or otherwise, it can be reduced to a shortest path representable one. In this paper, however, we have shown that this result alone is not sufficient to warrant optimal performance of OSPF networks, because the process of mapping paths to shortest paths is highly prone to ambiguity. We have supplied both

theoretical and empirical evidence that the concept of minimal representations is a remarkably useful one, or at least, it is much more useful than plain or perfect representations. A minimal representation constitutes a theoretical upper bound on the achievable precision and, moreover, it also contains perfect and plain representations as corner cases. In addition, using he mSPR algorithm, a minimal representation can always be computed in polynomial time by solving a series of linear programs.

References

1. B. Fortz, J. Rexford, and M. Thorup, "Traffic engineering with traditional IP routing protocols," *IEEE Communications Magazine*, vol. 40, pp. 118–124, Oct 2002.
2. A. Sridharan, C. Diot, and R. Guérin, "Achieving near-optimal traffic engineering solutions for current OSPF/IS-IS networks," in *Proceedings of INFOCOM 2003*, vol. 2, pp. 1167–1177, March 2003.
3. G. Rétvári and T. Cinkler, "Practical OSPF traffic engineering," *IEEE Communications Letters*, vol. 8, pp. 689–691, Nov 2004.
4. J. Moy, "OSPF Version 2." RFC 2328, April 1998.
5. Y. Wang, Z. Wang, and L. Zhang, "Internet traffic engineering without full-mesh overlaying," in *Proceedings of INFOCOM 2001*, vol. 1, pp. 565–571, April 2001.
6. G. Rétvári, R. Szabó, and J. J. Bíró, "On the representability of arbitrary path sets as shortest paths: Theory, algorithms, and complexity," in *Lecture Notes in Computer Science: Proceedings of the Third International IFIP-TC6 Networking Conference, Athens, Greece*, pp. 1180–1191, May 2004.
7. R. Guerin, A. Orda, and D. Williams, "QoS routing mechanisms and OSPF extensions." IETF RFC 2676, 1999.
8. G. Rétvári, T. Cinkler, and J. J. Bíró, "Notes on shortest path representation," Tech. Rep. 2005-001, Feb 2005. available on-line: `http://qosip.tmit.bme.hu/~retvari/publications/SPR_tech_rep.pdf`.
9. R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications.* Prentice-Hall, Englewood Cliffs, NJ, 1993.
10. J. L. Kennington, "A survey of linear cost multicommodity network flows," *Operations Research*, vol. 26, no. 2, pp. 209–236, 1978.
11. M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, *Linear Programming and Network Flows.* John Wiley & Sons, January 1990.
12. The GLPK project: `http://www.gnu.org/software/glpk/glpk.html`.
13. A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: Universal topology generation from a user's perspective," Tech. Rep. 2001-003, Jan 2001.
14. Z. Wang and J. Crowcroft, "Quality-of-service routing for supporting multimedia applications," *IEEE Journal of Selected Areas in Communications*, vol. 14, no. 7, pp. 1228–1234, 1996.
15. J. L. Sobrinho, "Algebra and algorithms for QoS path computation and hop-by-hop routing in the Internet," in *INFOCOM*, pp. 727–735, 2001.