On Shortest Path Representation

Gábor Rétvári, *Member, IEEE*, József J. Bíró, *Member, IEEE*, and Tibor Cinkler, *Member, IEEE*

***Abstract*—Lately, it has been proposed to use shortest path first routing to implement Traffic Engineering in IP networks. The idea is to set the link weights so that the shortest paths, and the traffic thereof, follow the paths designated by the operator. Clearly, only certain *shortest path representable* path sets can be used in this setting, that is, paths which become shortest paths over some appropriately chosen positive, integer-valued link weights. Our main objective in this paper is to distill and unify the theory of shortest path representability under the umbrella of a novel flow-theoretic framework. In the first part of the paper, we introduce our framework and state a descriptive necessary and sufficient condition to characterize shortest path representable paths. Unfortunately, traditional methods to calculate the corresponding link weights usually produce a bunch of superfluous shortest paths, often leading to congestion along the unconsidered paths. Thus, the second part of the paper is devoted to reducing the number of paths in a representation to the bare minimum. To the best of our knowledge, this is the first time that an algorithm is proposed, which is not only able to find a minimal representation in polynomial time, but also assures link weight integrality. Moreover, we give a necessary and sufficient condition to the existence of a one-to-one mapping between a path set and its shortest path representation. However, as revealed by our simulation studies, this condition seems overly restrictive and instead, minimal representations prove much more beneficial.**

***Index Terms*—Linear programming, shortest path routing, traffic engineering.**

I. Introduction

TRAFFIC Engineering (TE, [1]) is concerned with the performance optimization of operational networks. Its main objective is to reduce the congestion and improve resource utilization across the network through carefully managing the traffic distribution on network links. In the traditional approach, the so-called *Overlay Model* [2], TE is delegated to a dedicated connection-oriented infrastructure, like MultiProtocol Label Switching (MPLS) or Asynchronous Transfer Mode (ATM). Tunnels are set up to carefully map traffic to the physical topology and a full-meshed virtual IP network is *overlayed* on top of the tunnels. While this method allows service providers to fine-tune the distribution of traffic in the network, it also raises certain scalability and network management issues [2].

Drifting more and more into the focus of the research community recently there is an alternative approach, the so-called *Peer Model*. In this model, TE is implemented 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. If multiple shortest paths exist, the optional Equal-Cost-MultiPath (ECMP) load balancing technique allows to split traffic roughly evenly amongst them. In the Peer Model, a suitable Traffic Engineer participates in the signaling of the Open Shortest Path First (OSPF, [3]) or Intermediate-System-to-Intermediate System (IS-IS, [4]) routing protocol. Based on the routing information gathered from the network, link weights are computed such that the resultant shortest paths manifest some sophisticated TE goal, and these link weights are then distributed back to the routers. Henceforward, the basic operation of OSPF/IS-IS remains the same but the traffic in the network will follow the paths assigned by the traffic engineer. Since this process basically boils down to the careful manipulation of OSPF link weights aiming towards balanced traffic distribution and reduced congestion, it is usually referred to as *OSPF Traffic Engineering* [5] in the literature.

On one hand, OSPF TE eliminates the connection-oriented layer all together from the network architecture leading to a more economical approach. On the other hand, OSPF TE is fundamentally restricted, because it dictates that all traffic must follow the shortest paths in the network. A crucial question yet remains to be answered is how stringent the limitations of the shortest path forwarding paradigm in effect turn out. Should it impose too much restriction in choosing paths for traffic instances in the network, OSPF TE would be doomed to remain yet another unfulfilled promise. If, in contrast, the shortest path forwarding paradigm proves itself flexible enough to accommodate a wide range of path assignment strategies, then OSPF TE may easily become the tool for "poor man's traffic engineering." OSPF routers are omnipresent and OSPF TE can readily be deployed without even the slightest modification of the legacy network infrastructure.

This paper is dedicated to an essential problem that inherently underlies OSPF TE: shortest path representation. A path set is said to be shortest path representable, if there exist positive link weights based on which it becomes a set of shortest paths. We assume that the TE algorithm has *a priori* knowledge on the set of source-destination pairs. We further assume that the set of paths to be assigned for the source-destination pairs is also given, but we are not concerned with the question of how to efficiently select these designated paths (see [6] for a discussion on the complexity of this problem). This approach allows us to maintain a sharp focus on the central problem of this paper: how to map a given set of paths to shortest paths as precisely as possible.

Manuscript received March 4, 2005; revised May 13, 2006; approved by IEEE/ACM Transactions on Networking Editor T. Griffin.

The authors are with the High Speed Networks Laboratory, Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, H-1117, Magyar Tudósok körútja 2., Budapest, Hungary (e-mail: retvari@tmit.bme.hu; cinkler@tmit.bme.hu; biro@tmit.bme.hu).

Digital Object Identifier 10.1109/TNET.2007.900708

A. Related Works

The foundations of OSPF TE are laid down by [7]. In an unpublished work [8] the same authors show that it is NP-hard to compute link weights, which assure the optimal distribution of the resultant shortest paths (see [9] for a different treatment). The proposed local search heuristic algorithm achieves near-optimal routing in some cases. For further discussion the reader is referred to [5], [10] and [11]. Another approach to OSPF TE is the so-called two-phase method [9]. In the first phase, the traffic engineer determines an optimal traffic distribution in the network by assigning paths for the traffic instances. In the second phase, the task is to find positive, integer-valued OSPF link weights, over which these paths are all shortest paths. Obviously, setting these weights in the OSPF routers yields the required routing pattern that was determined by the traffic engineer in the first phase. The emergent *shortest path representation problem* stands as the question of main interest in this paper.

An interesting approach to solve the shortest path representation problem was given in [12]. Thinking of the problem as the inverse of the traditional shortest path problem allowed Faragó *et al.* to invoke the elaborated toolset of inverse optimization (see [13] and references therein) to tackle it. Their most important result is that the inverse shortest path routing problem is—in theory—tractable in polynomial time. Another intriguing treatment is presented in [14]. The unique power of their algebraic description permits the authors to provide a thorough characterization of path selection algorithms that produce loop-free paths and prove that a number of popular routing algorithms, such as for example the widest-shortest-path algorithm [15], are amongst them. Nevertheless, it seems that the most promising approach to tackle the shortest path representation problem is a flow-theoretic one. Notably, the set of designated paths can be turned into path-flows, and the problem can be stated as a special case of the path-flow formulation of multicommodity flow problems. The authors in [9] take just this route. This work seems to be the first one to anticipate the deep embedment of this problem in the theory of linear programming. A good summary of the related results can be found in [16].

Although several different frameworks for shortest path representation have appeared in the literature recently [12], [16], [17], the first in-depth exposition of the concept is due to Wang *et al.* [18]. Building on an arc-flow formulation, first they argue that a number of traffic engineering optimization criteria—like for instance minimizing the cost of the routing or minimizing the maximal link utilization—give rise to a linear programming formulation that, by nature, produces shortest paths. This finding leads Wang *et al.* to the main result of their paper: a path set is either shortest path representable, or otherwise it contains certain directed cycles they call *loops*. Eliminating these loops in turn gives a shortest path representable path set. This work was the last one in a sequence of papers that helped to disprove the common misbelief that shortest path first routing is, by nature, useless to traffic engineering. It also induced a number of derivatives, see for example [19]. Moreover, this work is the main driving force behind our unified framework presented below. However, our framework differs significantly from that of Wang *et al.* and the other ones available in the literature. Once, our framework is not confined to find one feasible shortest path representation, but it instead goes for finding the most precise representation possible. We take special care to assure that the proposed algorithms for computing the offending link weights always retain polynomial tractability. This assures modest execution time and predictable storage requirements. Finally, our framework assures that the emergent link weights are always integer-valued—an important built-in requirement of routing protocols of our days.

B. Mathematical Model

Given a graph that describes a real physical entity, such as a telecommunications network, and some *designated paths* selected carefully in advance, the task is to 1) decide whether or not there exist positive-valued link weights over which the designated paths become shortest paths and 2) to actually compute such link weights. The mathematical model to cope with this problem is as follows.

Let $G(V,E)$ be a simple directed graph, formed by the set of nodes V $(|V|=n)$ and the set of edges E $(|E|=m)$. An $s \to d$ $(s \in V,\, d \in V,\, s \neq d)$ path P of length L_P is defined by its consecutive edges: $P := \{(v_i, v_{i+1}) \in E : i = 1 \ldots L_P, v_1 = s, v_{L_P+1} = d\}$. We assume that a path does not contain cycles and repeated edges (however, the proposed framework does not impose this restriction). Alternatively, one can employ a vector-representation of paths, which will be referred to as the *support*. The support of P is a column m-vector $\sup(P)$, such that the component corresponding to link (i,j) is 1 if $(i,j) \in P$ and zero otherwise. One might also think of P as a special subgraph $G_P(V,P)$ of $G(V,E)$.

In shortest path first routing there is an additive, positive valued weight w_{ij} associated with each network link $(i,j) \in E$, which may represent a real physical quantity (e.g., the delay on the link) or some other administrative policy set by the network operator. For a path P, the weight of P over the link weights $\mathcal{W} = \{w_{ij} : (i,j) \in E\}$ is defined as $W(P) = \sum_{(i,j)\in P} w_{ij}$. To simplify the notation, we gather w_{ij}s into a row m-vector w. The entry of w in the position corresponding to link (i,j) is w_{ij}. In vector notation, the weight of a path P can be expressed as the scalar product of w and the support of P, say, p, i.e., $W(P) = wp$.

Now, suppose that we are given a set of source-destination pairs (or *sessions*) $(s_k, d_k) : k \in \mathcal{K}$ and, for each $k \in \mathcal{K}$ a set of $s_k \to d_k$ paths $\mathcal{P}_k$. Note that we require that all d_ks are distinct, which assures that a separate entry is maintained for each session in the IP routing table (see [19] for a discussion on how to adapt a network configuration to this requirement). Then, the support of $\mathcal{P}_k$ is defined as $\sup(\mathcal{P}_k) = \sum_{P \in \mathcal{P}_k} \sup(P)$. In another interpretation, $\mathcal{P}_k$ is simply a union of the subgraphs corresponding to the paths in it, that is, $G_{\mathcal{P}_k}(V, E(\mathcal{P}_k)) : E(\mathcal{P}_k) = \cup_{P\in\mathcal{P}_k} P$. This interpretation allows us to easily compare two path sets: we say that a set of $s_k \to d_k$ paths $\mathcal{P}_k$ is equivalent to another set of $s_k \to d_k$ paths $\mathcal{Q}_k$, that is, $\mathcal{P}_k \equiv \mathcal{Q}_k$, if $E(\mathcal{P}_k) = E(\mathcal{Q}_k)$. Similarly, $\mathcal{Q}_k$ is broader than, or equals $\mathcal{P}_k$, that is, $\mathcal{P}_k \preceq \mathcal{Q}_k$, if $E(\mathcal{P}_k) \subseteq E(\mathcal{Q}_k)$. For a weight set $\mathcal{W}$, we denote the set of $s_k \to d_k$ shortest paths over $\mathcal{W}$ by $\mathcal{P}_k(\mathcal{W})$.

Next, we extend the notation to path sets that contain paths for multiple different source-destination pairs. Let $\mathcal{P} = \cup_{k\in\mathcal{K}} \mathcal{P}_k$, respectively $\mathcal{Q} = \cup_{k \in \mathcal{K}} \mathcal{Q}_k$, be two sets of paths with respect to some set of sessions $\mathcal{K}$. Then, $\mathcal{P} \equiv \mathcal{Q}$ if $\forall k \in \mathcal{K} : \mathcal{P}_k \equiv \mathcal{Q}_k$ and similarly, $\mathcal{P} \preceq \mathcal{Q}$, if $\forall k \in \mathcal{K} : \mathcal{P}_k \preceq \mathcal{Q}_k$. Additionally, for a weight set $\mathcal{W}$ the set of shortest paths over $\mathcal{W}$ with respect to the source-destination pairs $(s_k, d_k) : k \in \mathcal{K}$ is denoted by $\mathcal{P}(\mathcal{W})$.

Lastly, we introduce the notion of path-graphs, which capture some (but not all) important structural properties of a path set. The path-graph $G_{\mathcal{P}}$ is defined as the subgraph of G spanned by the nodes V of G and the edges that show up in at least one path in $\mathcal{P} : G_{\mathcal{P}}(V, E(\mathcal{P})) : E(\mathcal{P}) = \cup_{k \in \mathcal{K}} E(\mathcal{P}_k)$. Throughout this paper, we shall usually restrict the network G to $G_{\mathcal{P}}$ without causing any loss of generality. We shall assume that the weight of the links outside of the path-graph is eventually set to a suitably large value, assuring that all shortest paths circumvent these links.

C. Problem Formulation

The focal problem we investigate in this paper is, given a set of paths $\mathcal{P}$, to compute a link weight set $\mathcal{W}$ as to assure that all designated paths are shortest paths over $\mathcal{W}$. In this case, we say that $\mathcal{P}(\mathcal{W})$ is a *shortest path representation* (SPR) of $\mathcal{P}$ and $\mathcal{W}$ *implements* a shortest path representation of $\mathcal{P}$ or $\mathcal{W}$ is an SPR link weight with respect to $\mathcal{P}$. Furthermore, $\mathcal{P}$ is shortest path representable (again SPR) if such weight set exists. More formally:

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

Consider the sample network depicted in Fig. 1, which we adopted from [18]. In this setting,[1] there are three source-destination pairs: (A, G), (B, D) and (C, B). We are given the path set marked by dashed black arrows in the figure and the task is to either compute a proper SPR weight set or conclude that no such weights exist. It is relatively easy to see that the latter case holds. Observing that subpaths of shortest paths must again be shortest paths, we have that $W((A,F),(F,G)) < W((A,F),(F,G),(G,D))$ since former is a subpath of the latter, $W((A,F),(F,G),(G,D)) \leq W((A,D))$ since both are required to be shortest paths and thus $W((A,D)) < W((A,D),(D,C),(C,E),(E,G)) \leq W((A,F),(F,G))$ is a contradiction. Nevertheless, the decision is not so trivial for more complex path sets, therefore, the first part of the paper, Section II, is devoted to this problem. We define a universal framework and then we state a sufficient and necessary condition to shortest path representability. This theorem has broad implications in the field of OSPF TE, as revealed in the rest of Section II. We also enumerate the most appealing techniques to compute SPR link weights.

It is timely to call the attention of the reader to a subtlety in Definition 1, which will play a crucial role in the sequel. Namely, we do not require the equivalence of the designated path set $\mathcal{P}$ and its shortest path representation $\mathcal{P}(\mathcal{W})$. We only demand $\mathcal{P}$ to be a subset of $\mathcal{P}(\mathcal{W})$. This weak definition allows $\mathcal{P}(\mathcal{W})$ to contain more paths than $\mathcal{P}$, and in fact, this is usually

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

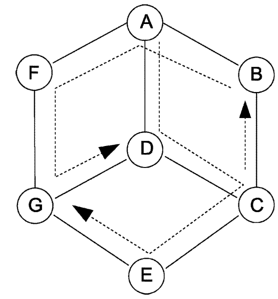


Fig. 1. Sample topology and path set.

the case. In [6] we pointed out that traffic routed to the additional paths that were not taken into account in the course of network dimensioning may introduce adverse interference. Therefore, it is essential to reduce the number of extraneous paths in the representation to the minimum. Such *minimal representations* constitute the topic of the second part of the paper, Section III. We provide a thorough characterization of the paths in the minimal representation and give a polynomial time algorithm to compute the corresponding link weights. Finally, we investigate when a one-to-one mapping between a path set and its shortest path representation exists. While there have already been some sporadic attempts in the literature (see [9] for an approach based on successive linear programming and [17] for some necessary conditions), as far as we know, this is the first time that a concise necessary and sufficient condition is given.

Real-life routing protocols impose a number of restrictions on the range link weights can take their values from. Link weights must be positive to avoid routing loops (oddly enough, IS-IS allows for zero-valued link weights). Furthermore, routing protocols can not cope with fractional link weights. For example, OSPF allocates 16 bits to store the weight and handles this quantity as an unsigned integer. Thus the need to restrict the computed weight set to integral values naturally arises. In order to simplify the development, for the most part of the paper we relax this integrality requirement, which might easily turn out to be disastrous in the long run. The workhorse in computing SPR weights is linear programming, for which general polynomial time algorithms exist. However, at the point when one imposes integrality criteria on some variables, the well-conditioned linear programming problem turns into a mixed integer linear programming problem, which is in general in *co-NP*. Therefore, our result that an integral SPR weight set can always be found (provided it exists) in polynomial time may have substantial practical relevance. This is the topic of main interest in Section IV.

Having laid down the essential theoretical foundations, in Section V we present the results of some related simulation studies to compare different concepts of SPR. Finally, in Section VI, we summarize our contributions.

Throughout the paper we shall repeatedly point out the deep embedment of our focal problem into the theory of network flows and linear programming. Where appropriate, we shall call the attention of the reader to the broader interpretations and generalizations of our results reaching far beyond the scope of mere OSPF TE. Most of the proofs are indispensable to get a good understanding of the mathematical backgrounds and concise enough to be presented in line with the text.

II. Shortest Path Representability

In this Section a unified framework for shortest path representability is presented. Suppose that we are given a *single* source node s and a *single* destination node d and some set of $s \rightarrow d$ paths $\mathcal{P}$ provisioned between s and d. In the broad literature dealing with shortest path problems of similar kind (see for example [20], [21]), a common practice is to associate distance labels, or *node potentials* with the nodes of the graph and describe the shortest paths in terms of the node potentials. Hence, for any node $v \in V$ the node potential π_v is a label, which signifies an upper bound on the shortest distance from the source node s to v over $\mathcal{W}$.

Proposition 1 (Shortest Path Optimality Conditions [20]): π_v and w_{ij} together satisfy

$$\forall (i,j) \in E : \pi_j - \pi_i \leq w_{ij}. \tag{1}$$

Furthermore, a path P is a shortest path over $\mathcal{W}$ if and only if (1) is satisfied with strict equality at all links of P:

$$\forall (i,j) \in P : \pi_j - \pi_i = w_{ij}.$$

Now, one can directly apply Proposition 1 and conclude that $\mathcal{W} = \{w_{ij} : (i,j) \in E\}$ represents the designated paths $\mathcal{P}$ as shortest paths if and only if there exist proper node potentials π_v, which solve the following linear system:

$$\pi_j - \pi_i = w_{ij} \quad \forall P \in \mathcal{P}, \forall (i,j) \in P \tag{2}$$
$$\pi_j - \pi_i \leq w_{ij} \quad \forall (i,j) \in E \tag{3}$$
$$w_{ij} > 0 \quad \forall (i,j) \in E. \tag{4}$$

In the usual setting, we are given the weight set $\mathcal{W}$ and the task is to compute the shortest paths $\mathcal{P}$ (or at least one such path). This can easily be solved by for instance Dijkstra's algorithm in strictly polynomial time. However, in the case of OSPF TE the task is just the inverse: given a set of designated paths $\mathcal{P}$, compute an appropriate weight set $\mathcal{W}$, which satisfies (2)–(4).

Now, we make the following revision to (2)–(4). First, we describe the system in terms of the support of $\mathcal{P}$. Recall that $p = \sup(\mathcal{P})$ counts the number of paths in $\mathcal{P}$ that use a particular link (i,j). Thus, $p_{ij} = 0$ implies that (i,j) is not used by any of the designated paths, while $p_{ij} > 0$ indicates that (i,j) is used by at least one path in $\mathcal{P}$. Therefore, the shortest path optimality condition $\pi_j - \pi_i \leq w_{ij}$ must hold with strict equality at such (i,j) links. We also associate a slack variable v_{ij} with each inequality (3). Observe that $p_{ij} > 0$ induces that $v_{ij} = 0$, which yields $v_{ij}p_{ij} = 0$ for each link (i,j). Finally, we observe that w_{ij} is required to be strictly positive. Such strict inequalities tend to be notoriously hard to consider in linear feasibility problems, because they restrict the search into the closure of the feasible region. We tackle this difficulty by associating a strictly positive *initial cost* ξ_{ij} with each link, which separates w_{ij} away from zero. Thus, we shall seek the link weights in the form $w_{ij} = \xi_{ij} + \omega_{ij}$ with $\omega_{ij} \geq 0$. Note that the choice of the initial costs is completely optional, the only requirement is that ξ_{ij} must be positive (see later). We shall use the setting $\xi_{ij} = 1$ for each link throughout this paper.

Hence, the *SPR problem* can be posed as, given a set of designated paths $\mathcal{P}$ (with support p_{ij}), compute link weights w_{ij} and node potentials π_v, such that

$$v_{ij}p_{ij} = 0 \quad \forall (i,j) \in E \tag{5}$$
$$\pi_j - \pi_i + v_{ij} = \xi_{ij} + \omega_{ij} \quad \forall (i,j) \in E \tag{6}$$
$$\omega_{ij} \geq 0, \ v_{ij} \geq 0 \quad \forall (i,j) \in E. \tag{7}$$

Observation 1: If ω_{ij} solves the SPR problem (5)–(7) for some set of $s \rightarrow d$ paths $\mathcal{P}$, then the weight set $\mathcal{W} = \{\xi_{ij} + \omega_{ij}\}$ is strictly positive and represents $\mathcal{P}$ as shortest paths. If, on the other hand, the system does not have a feasible solution for any choice of ξ, then $\mathcal{P}$ is not shortest path representable.

In the foregoing discussions, it will be more convenient to use a vector formulation of the SPR problem. Let N denote the node-arc incidence matrix. Each row in this matrix corresponds to a node and each column corresponds to an arc. Each column has exactly two non-zero entries. The column corresponding to arc (i,j) has a -1 in the row i and a $+1$ in the row j and a zero corresponding to all other rows. Let $p = \sup(\mathcal{P})$. Furthermore, gather the node potentials into a n-dimensional row vector π, and link weights, slack variables and initial costs into m-dimensional row vectors ω, v and ξ, respectively. Then the SPR problem can be written as

$$vp = 0 \tag{8}$$
$$\pi N + v = \xi + \omega \tag{9}$$
$$\omega \geq 0, v \geq 0. \tag{10}$$

In practice, usually we are given multiple source-destination pairs $(s_k, d_k) : k \in \mathcal{K}$, and between each one of these source-destination pairs some set of paths $\mathcal{P}_k$ (with support p^k) is provisioned. Let $\mathcal{P} = \cup_{k \in \mathcal{K}} \mathcal{P}_k$, let $K = |\mathcal{K}|$ denote the number of sessions and $t_k = |\mathcal{P}_k|$ denote the number of paths for session k. We extend (8)–(10) to this setting by assigning a separate node potential vector π^k and a separate vector of slack-variables v^k for each session $k \in \mathcal{K}$. Then, the full-fledged multicommodity formulation of the SPR problem can be stated as

$$v^k p^k = 0 \quad \forall k \in \mathcal{K} \tag{11}$$
$$\pi^k N + v^k = \xi + \omega \quad \forall k \in \mathcal{K} \tag{12}$$
$$\omega \geq 0, \ v^k \geq 0. \tag{13}$$

Note that Observation 1 readily generalizes to the multicommodity SPR problem.

The most familiar approach to solving linear feasibility problems of similar kind is to convert them to a linear program (LP) and use some sophisticated LP solver to obtain a solution. Notably, the SPR problem lends itself perfectly to such a conversion. Consider the following LP formulation:

$$\min \sum_{k \in \mathcal{K}} v^k p^k \tag{14}$$

$$\pi^k N + v^k = \xi + \omega \quad \forall k \in \mathcal{K} \tag{15}$$
$$\omega \geq 0,\ v^k \geq 0. \tag{16}$$

It is not particularly hard to deduce that a path set $\mathcal{P}$ is shortest path representable, if and only if the optimal objective value of the above LP is zero. If the objective is zero, then condition (11) is immediately satisfied (recall that both $p^k \geq 0$ and $v^k \geq 0$). Together with the constraints (15), (16) this yields a feasible solution to the SPR problem. Otherwise, if the optimal objective is greater than zero (it can not be less), then for some session k and for some link $(i,j) : v_{ij}^k p_{ij}^k > 0$. This implies that the shortest path optimality conditions do not hold for some path in $\mathcal{P}_k$, so $\mathcal{P}$ is not shortest path representable (at least, for the present choice of ξ), because a valid SPR weight set would induce a super-optimal solution to the above LP. However, not just that the LP (14)–(16) generates the SPR link weights (provided that such weights exist), but the reverse is also true: any optional set of link weights, which happens to implement a shortest path representation of $\mathcal{P}$ immediately gives an optimal feasible solution to the above problem. Since this LP is so closely coupled with the shortest path representability of $\mathcal{P}$, we call it the *dual formulation of the fundamental LP of* $\mathcal{P}$: D-LP($\mathcal{P}$). Without causing too much notational abuse, the solutions of D-LP($\mathcal{P}$) $\pi^1,\ldots,\pi^K, v^1,\ldots,v^K,\omega$ will be abbreviated as $[\pi^k, v^k, \omega]$ throughout this paper.

Next, we show that the primal formulation of the fundamental LP is basically a multicommodity network flow problem. Let $-y^k$ be an m-dimensional column-vector of the dual variables corresponding to the constraints (15). Using this notation, we obtain the following dual of (14)–(16), the so-called *Multicommodity Improvement Problem*:

$$-\min \sum_{k\in\mathcal{K}} \xi y^k \tag{17}$$
$$Ny^k = 0 \quad \forall k \in \mathcal{K} \tag{18}$$
$$\sum_{k\in\mathcal{K}} y^k \leq 0 \tag{19}$$
$$y^k \geq -p^k \quad \forall k \in \mathcal{K}. \tag{20}$$

Curiously, the dual variables y_{ij}^k behave just like arc-flows in conventional multicommodity flow problems. This relation becomes even more obvious if we substitute $x^k = y^k + p^k$ into the improvement problem. First, we observe that p^k defines a valid flow, where the supply/demand coincides with the number of designated paths for session k, t_k. Let t^k be an n-vector that describes the demands

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

Thus, $Np^k = t^k$ and the substitution $x^k = y^k + p^k$ to constraint (18) yields that $Nx^k = N(y^k + p^k) = Ny^k + Np^k = t^k$. Letting $p = \sup(\mathcal{P}) = \sum_{k\in\mathcal{K}} p^k$ and further applying the substitution to the remaining constraints yields the following *primal formulation of the fundamental LP*, P-LP($\mathcal{P}$):

$$\sum_{k\in\mathcal{K}} \xi p^k - \min \sum_{k\in\mathcal{K}} \xi x^k \tag{21}$$
$$Nx^k = t^k \quad \forall k \in \mathcal{K} \tag{22}$$
$$\sum_{k\in\mathcal{K}} x^k \leq p \tag{23}$$
$$x^k \geq 0 \quad \forall k \in \mathcal{K}. \tag{24}$$

Interestingly, the primal fundamental LP is basically a minimum cost multicommodity flow problem (apart from a constant term in the objective function). Observe that constraints (22) give the *flow conservation* constraints with respect to the demands t_k. The so-called *bundle constraints* (23) restrict the sum of the arc-flows for each session to remain under p_{ij} on each link. In fact, p_{ij} acts as some sorts of link capacity. Finally, arc-flows are required to be non-negative by (24). We have seen that the prerequisite of the shortest path representability of a path set is that the optimal objective of the dual fundamental LP must be zero. By the strong duality-theorem of linear programming, we have that in this case the optimal solution of the primal must also be zero. Observe also that this property does not depend on the actual choice of ξ: if the objective function value becomes zero for some choice $\xi = \xi_1$, then it is zero for any other strictly positive initial cost vector ξ_2 (this can be seen by factoring out ξ from (21)). This leads to following important result:

Theorem 1: Let $\mathcal{P}$ be some set of designated paths for some set of sessions $\mathcal{K}$. Then, $\mathcal{P}$ is representable as shortest paths, if and only if $p^k = \sup(\mathcal{P}_k)$ gives an optimal feasible solution to the primal fundamental LP of $\mathcal{P}$. In this case, the optimal objective is zero. If, in contrast, the optimal objective is positive, then $\mathcal{P}$ is not SPR.

In the remaining part of the paper, we shall refer to a solution $x^1,\ldots,x^K$ of P-LP($\mathcal{P}$) as $[x^k]$ for short.

A. Remarks on Theorem 1

The relation between linear programming and inverse shortest path problems have already been pointed out. Most notably, in [12] the authors show a SPR method based on the path-flow formulation of network flow problems. However, the relatively huge size of the resultant LP makes this formulation unattractive for practical purposes. Additionally, the relation of multicommodity flow problems and the SPR problem is also reported in [18]. What sets apart our development from previous results is the universal treatment as revealed below in more detail.

Under the hood, the primal fundamental LP 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 the original path set. If the reallocation can be done so that the aggregate length of the new path set (in terms of ξ) is less than that of $\mathcal{P}$, then the path set is not SPR. This sheds more light on the notion of the multicommodity improvement problem as well. Since, owing to the constraint $Ny^k = 0$, the arc-flows of the improvement problem y^k form flow circulations (i.e., unions of cycles), the improvement problem in fact asks for negative cost flow circulations, over which the initial feasible solution $x^k = p^k$ of the primal fundamental LP can be *improved*. If no negative cost flow circulations can be found, then $y^k = 0$ is optimal solution to the improvement problem, and hence, $x^k = p^k$ is an optimal solution to P-LP($\mathcal{P}$). Readers familiar with network flow theory may recognize an interesting extension of the single commodity *Negative Cycle Optimality Conditions* [20] to multicommodity flows here.

The significance of Theorem 1 is that it transforms the SPR problem from a mostly unknown generic linear feasibility problem into a convex cost multicommodity flow problem. Multicommodity flow problems have constituted one of the most researched fields for at least 40 years now to the point that even enormously large problem instances have become effectively solvable with reasonable computational efforts [22]. Exploiting the special structure of the constraint set, various decomposition techniques were developed (see [20], and references therein), such as the column-generation and basis-partitioning methods [21], to yield precise primal and dual solutions. Nevertheless, for moderate sized problems even a generic LP solver, such as the simplex algorithm provides a viable option.

Finally, it is important to mention that it is not always necessary to solve the SPR problem to optimality. One might trade-off the precision of the representation for the implied computational requirements by using some multicommodity flow approximation technique [23]. In [24] we give a solution technique based on the Lagrangian-relaxation of the fundamental LP, which promises to improve a potentially non-SPR path set towards shortest path representability, while also delivering suitable link weights at the same time. Our experiments suggest that one can obtain a reasonable SPR weight set in a few dozen iterations by solving nothing more than simplistic shortest path problems.

B. Consequences of Theorem 1

Theorem 1 tells whether or not a set of designated paths is representable as shortest paths and states that this is an inherent property of the path set. Shortest path representability is therefore tightly coupled with the structural properties of a path set $\mathcal{P}$, and hence, its path-graph $G_{\mathcal{P}}$ (recall that the path-graph corresponds to the subgraph spanned by the edges in $\mathcal{P}$). Consequently, the path-graph $G_{\mathcal{P}}$ was used previously to make some interesting observations regarding the shortest path representability of a path set $\mathcal{P}$. For example, in [6] the following was proposed as a quick test for assessing the shortest path representability of a path set:

Proposition 2: A path set $\mathcal{P}$ is shortest path representable if the path graph $G_{\mathcal{P}}$ induced by $\mathcal{P}$ is acyclic (i.e., it does not contain any directed cycles).

This can easily be seen as a consequence of Theorem 1, since, in an acyclic graph constraint (19) always holds with strict equality (otherwise, $\sum_{k\in\mathcal{K}} y^k$ would define a directed cycle). Hence, the objective function value is always zero, which equals to asserting that the path set is SPR (see [6] for an alternative proof based on topological ordering). Note that a set of paths may very well be SPR even if its path-graph contains directed cycles. Therefore, the above condition is obviously not a necessary one.

Furthermore, Theorem 1 may also help to get a better understanding of some prior results in the field of OSPF TE. For example, in light of the theorem we can give a new, more concise proof to the following result of Wang *et al.* [18]:

Proposition 3: Let $\mathcal{P}$ be a set of paths for some set of sessions $\mathcal{K}$. Now, either $\mathcal{P}$ is shortest path representable or otherwise, there exists a modified path set $\mathcal{P}'$, such that:

(i) $\mathcal{P}'$ is SPR;
(ii) $\mathcal{P}'$ is strictly shorter (in terms of the number of edges traversed) than $\mathcal{P}$;
(iii) $\mathcal{P}'$ can be obtained from $\mathcal{P}$ by eliminating some redundant multicommodity flow circulations;
(iv) the eliminated multicommodity flow circulations add up to directed cycles;
(v) $\mathcal{P}'$ has at least the same capacity (in terms of bottleneck bandwidth) as $\mathcal{P}$.

Proof: It is easy to see that the feasible region of the fundamental LP is non-empty, since at least $[p^k]$ defined by $\mathcal{P}$ is a feasible solution. Now, either $[p^k]$ is optimal (in which case $\mathcal{P}$ is SPR) or not. In the latter case, consider a primal optimal feasible solution $[x^k]$ to P-LP($\mathcal{P}$) and let the corresponding dual optimal solution be $[\pi^k, v^k, \omega]$. By the Flow Decomposition Principle [20], the optimal arc-flows x^k for some session k can be decomposed into at most t_k path-flows. Let P be such a path, i.e., $\forall(i,j) \in P : x_{ij}^k > 0$. But P is a shortest path over the weight set $\xi+\omega$, since, by complementary slackness, the corresponding slack variables v_{ij}^k are zero. That is, $\forall(i,j) \in P : \pi_j^k - \pi_i^k = \xi_{ij}+\omega_{ij}$ and this just corresponds to the assertion of the Shortest Path Optimality Conditions in Proposition 1. Therefore, $[x^k]$ defines a modified path set $\mathcal{P}'$, which is, in contrast to $\mathcal{P}$, shortest path representable. This proves (i). Since $\mathcal{P}$ is not SPR, but the modified path set defined by $[x^k]$ is, we have that the objective function value of P-LP($\mathcal{P}$) is positive, so

$$\sum_{k\in\mathcal{K}} \xi x^k < \sum_{k\in\mathcal{K}} \xi p^k. \tag{25}$$

Letting $\xi_{ij} = 1$ for each link $(i,j) \in E$, we have that (25) in fact measures the difference of the lengths of $\mathcal{P}$ and $\mathcal{P}'$, which proves (ii). It has already been pointed out that if $\mathcal{P}$ is not SPR, then $y^k = x^k - p^k$ defines flow circulations, which proves (iii). Any cycles add up to cycles, therefore aggregating $[y^k]$ into a "mass-flow" z again yields flow circulations. Furthermore, z is directed, since, by (19): $z = \sum_{k\in\mathcal{K}} y^k \leq 0$. Therefore, all links are traversed in the reverse direction by the cycles in z, which proves (iv). Finally, we observe that $\mathcal{P}'$ contains the same number of paths, t_k, for each session. If the designated path set, for example, contains 3 paths for some session, then $\mathcal{P}'$ again contains 3 paths, (yet, the modified paths may overlap). This is because $\mathcal{P}'$ also satisfies the flow conservation constraints (22) with respect to t_k just like $\mathcal{P}$. This brings us to a broader interpretation of t_k: if t_k denotes the amount of "demand" associated with each session, then $\mathcal{P}'$ is able to tolerate the same demand as $\mathcal{P}$, which proves (v). ■

The most notable finding of Proposition 3 is that a non-SPR path set can always be improved into an SPR one, which is on one hand shorter and, on the other hand, capable of carrying the same amount of bandwidth as the original paths. Thus, any arbitrary set of designated paths is either immediately SPR or otherwise, it can be substituted by a SPR path set, which lies entirely within the original paths, provides the same capacity

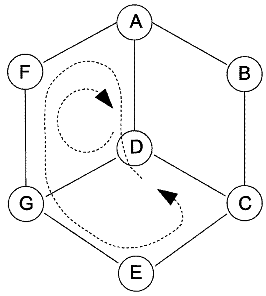


Fig. 2. A feasible solution to the improvement problem.

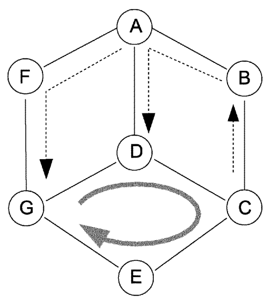


Fig. 3. A modified path set that is shortest path representable.

but uses less links. This finding points out the power of shortest path forwarding and OSPF TE.

To demonstrate the above discussion, we revisit our previous counter-example in Fig. 1. A set of paths is not representable as shortest paths, if there exists a specially structured set of multicommodity flow circulations (marked by dashed arrows in Fig. 2), which solve the improvement problem. Starting from the "bad" paths in Fig. 1 and instantiating one unit of flow along the flow circulations marked by dashed black arrows in Fig. 2 (which equals to sending -1 amount of flow in the reverse direction as suggested by a negative solution of the improvement problem), one obtains the optimal path set of Fig. 3. In the resultant setting, it is straightforward to set all link weights to one, which obviously reproduces the new path set (and, quite regrettably, some additional paths too).

Furthermore, the result that the improvement always comes in the form of directed mass-flow circulations deserves some special attention here. For example, aggregating the redundant multicommodity flow circulations of Fig. 2 into a mass-flow yields the directed cycle that is marked by a thick grey arrow in Fig. 3. Such directed cycles are generally called *loops* in the shortest path first routing terminology and regularly network operators go to any lengths to avoid them. In some respect, the multicommodity improvement problem can be thought of as the explicit flow-theoretic declaration that the path-set is loop-free. The mere fact that the dual problem rising from this declaration immediately delivers the required SPR weights might be perceived as an unanticipated beauty of network flow theory by some.

III. Toward More Precise Shortest Path Representations

So far, we have used a rather weak definition of shortest path representability. Namely, we have not required the equivalence of the designated path set $\mathcal{P}$ and its shortest path representation $\mathcal{P}(\mathcal{W})$. We have only demanded $\mathcal{P}$ to be a subset of $\mathcal{P}(\mathcal{W})$. In this section, we treat different "strengthenings" of the definition in order to obtain more and more accurate shortest path representations.

A weak definition of shortest path representability helped us to state the related theory in the most generic sense possible, however, it does not prove to be too prosperous in practice. To see why, consider the sample network of Fig. 4. All edge capacities equal 1 and we are given four source-destination pairs, (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. Again, the paths are all least-hop paths, so setting the link weights uniformly to 1 will safely reproduce the designated paths.

In this 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 at the branching nodes to the available shortest paths, and the additional traffic directed to the superfluous paths will substantially overload some of their links. In [6] we pointed out that the useful throughput of a network might decrease by as much as fifty percent due to the adverse interference caused by the extraneous shortest paths. To avoid this, it is crucial to eliminate as many extraneous paths from the representation as possible. Perhaps, the most straightforward strengthening of Definition 1 would be the following:

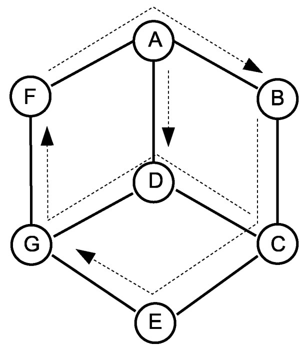
Definition 2: A path set $\mathcal{P}$ is perfectly shortest path representable (pSPR), if there exists a positive weight setting $\mathcal{W}$, such that $\mathcal{P} \equiv \mathcal{P}(\mathcal{W})$.

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 minimum by dropping as many extraneous paths as 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. 4, 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. If we wanted to eliminate, say, path $((B, C), (C, D), (D, G))$ 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 $((C, D), (D, G), (G, F))$ would cease to be a shortest path for session (C, F). It is noteworthy that using the minimal representation we could avoid to overload any of the links in the network.

In this section we shall argue that the concept of minimal representations is a remarkably useful one. First, it manifests an



(a)

	SPR		mSPR	
Link	weight	load	weight	load
(A,B)	1	1.83	2	1
(A,F)	1	1.75	1	1
(A,D)	1	1.42	2	1
(B,C)	1	0.83	1	1
(C,D)	1	0.58	1	1
(C,E)	1	0.58	1	1
(D,G)	1	0.66	1	1
(F,G)	1	0.75	1	1
(G,E)	1	0.58	1	1

(b)

Designated paths:
(F,B): ((F,A),(A,B))
(A,D): (A,D)
(B,G): ((B,C),(C,E),(E,G))
(C,F): ((C,D),(D,G),(G,F))

"Plain" SPR paths:
(F,B): ((F,A),(A,B))
(A,D): (A,D)
(B,G): ((B,C),(C,E),(E,G))
((B,C),(C,D),(D,G))
((B,A),(A,D),(D,G))
((B,A),(A,F),(F,G))
(C,F): ((C,D),(D,G),(G,F))
((C,E),(E,G),(G,F))
((C,D),(D,A),(A,F))
((C,B),(B,A),(A,F))

mSPR paths:
(F,B): ((F,A),(A,B))
(A,D): (A,D)
(B,G): ((B,C),(C,E),(E,G))
((B,C),(C,D),(D,G))
(C,F): ((C,D),(D,G),(G,F))
((C,E),(E,G),(G,F))

(c)

Fig. 4. A comparison of different concepts of shortest path representability. (a) Sample network topology. (b) Weight and load at each link assuming ECMP load-balancing. (c) Designated paths and shortest paths.

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.

A. Minimal Shortest Path Representations

As suggested by the previous discussions, computing the link weights that represent a set of paths as shortest paths amounts to solving a linear program, the fundamental LP. However, this may introduce a substantial number of extraneous paths into the set of shortest paths. Below, we show how to eliminate most of them to finally arrive to the minimal representation. But first, we need to state some technical Lemmata.

Lemma 1: Let $\mathcal{P}$ be SPR and let $[x^k]$ be a feasible solution of P-LP($\mathcal{P}$). Then, $[x^k]$ is also optimal.

Proof: Since $[x^k]$ is feasible in P-LP($\mathcal{P}$), it satisfies the bundle constraints (22). Furthermore, by $\xi > 0$ we have that

$$\xi \sum_{k \in \mathcal{K}} x^k \leq \xi p. \tag{26}$$

Now there are two cases. Either (26) holds with strict inequality, in which case $\xi p - \xi \sum_{k \in \mathcal{K}} x^k > 0$ implies that the objective function value of P-LP($\mathcal{P}$) is strictly positive, which, according to Theorem 1, contradicts the assumption that $\mathcal{P}$ is SPR. Otherwise, (26) holds with equality and, again by Theorem 1, $[x^k]$ is optimal. ■

What this Lemma insists is that if some path set is SPR, than the region of feasible solutions and the optimal solutions of the primal fundamental LP overlap. This suggests that the fundamental LP is indeed quite a special linear program. Next, consider the following application of the *Complementary Theorem of Linear Programming* [21] to the fundamental LP:

Lemma 2: Let $\mathcal{P}$ be an SPR path set. Now for some session $h \in \mathcal{K}$ and for some link $(i,j) \in E$, $x_{ij}^h = 0$ for all optimal feasible solutions of P-LP($\mathcal{P}$) if and only if for any $\kappa > 0$, there exists an optimal feasible solution to D-LP($\mathcal{P}$), such that $v_{ij}^h \geq \kappa$.

A detailed proof of the Lemma can be found in [25].

Armed with the Lemmata, we are in a position to give a thorough characterization of the paths in a minimal shortest path representation $\mathcal{P}_{min}$:

Theorem 2: Let $\mathcal{P}$ be a shortest path representable path set and let P be a $s_k \rightarrow d_k$ path for some session $k \in \mathcal{K}$. Then, the following statements are equivalent:

(i) $P \in \mathcal{P}_{\min}$;
(ii) there exists $[x^k]$ optimal feasible solution to P-LP($\mathcal{P}$), such that $\forall (i,j) \in P : x_{ij}^k > 0$;
(iii) for any optimal feasible solution $[\pi^k, v^k, \omega]$ of D-LP($\mathcal{P}$), $\forall (i,j) \in P : v_{ij}^k = 0$.

Proof: (i) ⇒ (ii): assume the contrary, that is, for some $(i,j) \in P : x_{ij}^k = 0$ for all optimal feasible solutions to P-LP($\mathcal{P}$). Then, by Lemma 2 we have that for some optimal feasible solution of D-LP($\mathcal{P}$) : $v_{ij}^k > 0$ (in fact, v_{ij}^k can be made arbitrarily large). However, from Proposition 1, we have that P is not a shortest path in this case, which contradicts the minimality of $\mathcal{P}_{\min}$.

(ii) ⇒ (iii) comes from the complementary slackness of x_{ij}^k and v_{ij}^k.

(iii) ⇒ (i): condition (iii) basically asserts that P is shortest path over *any* SPR weight set, and as such, it belongs to the minimal representation by definition. ■

Theorem 2 suggests that *the set of paths in the minimal representation is spanned by the alternative optimal solutions of the primal fundamental LP*. Suppose that, for some arbitrary $s_k \rightarrow d_k$ path P (designated or not), there exists an optimal feasible solution $[x^k]$ of P-LP($\mathcal{P}$), such that for each link (i,j) of $P : x^k_{ij} > 0$. Note that this equals to suitably relocating the paths in the designated path set, such that P is used by session k. In such cases, P will be a shortest path in *all* the representations, since, by complementary slackness, the corresponding slack variables v^k_{ij} are bound to zero in all dual optimal feasible solutions.

In our example of Fig. 4, the primal fundamental LP has two optimal feasible solutions (in fact, it has infinitely many, but these are all convex combinations of these two). The first optimal feasible solution $[x^k]$ corresponds to the case when one unit of flow is placed to the designated path of each one of the sessions. The other one, $[y^k]$, arises if the subpaths of the designated path of session (B,G) and (C,F) between nodes C and G are swapped, and one unit of flow is placed to the modified paths. Hence, we found an optimal feasible solution $[y^k]$ in which the flow y^k_{ij} corresponding to session (B,G) is strictly positive at each link of path $((B,C),(C,D),(D,G))$. The same applies to path $((C,E),(E,G),(G,F))$ and session (C,F). Therefore, these paths can never be eliminated, and thus, belong to the minimal shortest path representation. On the other hand, path $((B,A),(A,F),(F,G))$ can easily be dropped from the set of shortest paths of session (B,G), since, as easily seen, the primal fundamental LP has no optimal feasible solution placing flow on, say, link (A,F).

Our experiments with solving the fundamental LP (see Section V for the related simulation studies) suggest that usually there arises a large number of alternative optimal solutions, probably due to highly degenerate nature of the feasible region. Hence, the best one can hope for is that the minimal representation does not contain too many unintended paths, but there is no guarantee that a carefully selected designated path set will not deteriorate into a bunch of overlapping and interfering paths in the course of applying SPR. A minimal representation is an intrinsic property of a path set and there is *theoretically* no way to narrow the representation even further.

Instead of characterizing the paths in $\mathcal{P}_{min}$, the following complementary formulation of Theorem 2 rather describes the links, which do not participate in *any* of the paths in $\mathcal{P}_{min}$:

Theorem 3: Let $\mathcal{P}$ be a shortest path representable path set. Then, for some session $k \in \mathcal{K}$ and for some link $(i,j) \in E$, the following statements are equivalent:

(i) there is no $s_k \rightarrow d_k$ path P in $\mathcal{P}_{\min}$ such that $(i,j) \in P$;

(ii) for all optimal feasible solutions $[x^k]$ to P-LP($\mathcal{P}$) : $x^k_{ij} = 0$;

(iii) for any $\kappa > 0$, there exists an optimal feasible solution $[\pi^k, v^k, \omega]$ of D-LP($\mathcal{P}$) such that $v^k_{ij} \geq \kappa$;

(iv) there is an *optimal ray* d in the set of optimal feasible solutions of D-LP($\mathcal{P}$), such that the entry in d corresponding to v^k_{ij} is strictly positive.

In the light of the previous discussions, the proof of the theorem should be fairly obvious to the reader. The reason why we are still formulating this result as a theorem is because it gives a useful idea not only to construct an algorithm to obtain a minimal representation (see Section III-C for the discussion of such an algorithm), but also to prove a number of interesting properties of perfect representations.

B. Perfect Shortest Path Representations

In a perfect shortest path representation, the set of shortest paths of some session is completely determined by its own designated paths, and no additional shortest paths show up owing to *other* sessions. Obviously, a prefect representation is what the network operator would like to achieve eventually, as in this case no interference can occur along superfluous shortest paths, which have not been considered upon dimensioning the network. By using a pSPR path set in which only one path is assigned to every session even the notorious limitations of the ECMP equal-splitting requirement can be overcome. Unfortunately, as discussed below and demonstrated later by simulations, a perfect representation is hardly achievable in the vast majority of practical cases.

Theorem 3 insists that if a path P does not belong to $\mathcal{P}_{\min}$ for some session k, then it must contain at least one link (i,j), so that there exists an optimal ray d with strictly positive surplus in the position corresponding to v^k_{ij}. Note that an optimal ray (or optimal direction) d is a vector, such that starting from an arbitrary optimal feasible solution one can move in the direction of d forever while still remaining within the region of optimal feasible solutions. Therefore, moving along d yields an infinite number of candidate SPR weight sets. When eventually all those slack variables are elevated from zero for which such optimal ray exists, then the current link weights implement a minimal representation of $\mathcal{P}$. If this minimal representation is identical to $\mathcal{P}$, then $\mathcal{P}$ is pSPR, which supplies the following necessary and sufficient condition:

Corollary 1: A set of paths $\mathcal{P}$ is perfectly shortest path representable (pSPR) if and only if there exists an optimal feasible solution $[\pi^k, v^k, \omega]$ to D-LP($\mathcal{P}$), such that for each $k \in \mathcal{K}$ and for each $(i,j) \in E \setminus E(\mathcal{P}_k) : v^k_{ij} > 0$.

As mentioned previously, all alternative optimal solutions of the primal fundamental LP induce a candidate path set in the representation. One gets the impression that if the designated path set $\mathcal{P}$ itself provides a *unique* optimal feasible solution of P-LP($\mathcal{P}$), then $\mathcal{P}$ must be perfectly representable. This is because the paths in $\mathcal{P}$ can not be relocated to other flow paths in this case. As it turns out, this idea is almost pertinent: the uniqueness of the optimal feasible solution of P-LP($\mathcal{P}$) is generally a *sufficient condition to perfect shortest path representability*, but only by imposing some modest restrictions to $\mathcal{P}$ does it become necessary as well.

Corollary 2: Let $\mathcal{P}$ be some set of paths for some set of sessions $\mathcal{K}$, and let $\sup(\mathcal{P}_k) = p^k$. Now, if $[p^k]$ is the unique optimal feasible solution of the primal fundamental LP of $\mathcal{P}$, then $\mathcal{P}$ is perfectly shortest path representable.

The proof of the Corollary is omitted for brevity. It is tempting to investigate, why the above condition does not prove to be necessary as well. Take for instance our previously discussed example of Fig. 4 and, as designated paths, assign

$$P = ((B,C),(C,D),(D,G))$$
$$Q = ((B,C),(C,E),(E,G))$$

for session (B, G) and

$$R = ((C, D), (D, G), (G, F))$$
$$S = ((C, E), (E, G), (G, F))$$

for session (C, F), respectively. Now, the designated path set $\mathcal{P} = \{P, Q, R, S\}$ is obviously pSPR, however, the corresponding primal fundamental LP possesses infinite number of optimal feasible solutions, two of which just happens to be extremal. One extreme point solution rises when all the 2 units of demand of session (B, G) is shifted to the flow path P, and all demand of (C, F) to path S. The other one comes by assigning Q exclusively to session (B, G) and R to session (C, F). In some respect, the existence of the two extreme point solutions can be attributed to the fact that both sessions can use 2 paths, and their flows can be swapped between these two paths. However, in the case of *single-path routing*, that is, if all sessions are restricted to use only one designated path, this swapping behavior is precluded. Thus, for single-path routing the uniqueness of the solution $[p^k]$ of P-LP$(\mathcal{P})$ is not only a sufficient, but also a necessary condition.

Corollary 3: Suppose that some set of designated path set $\mathcal{P}$ contains only one path for each distinct $(s_k, d_k) : k \in \mathcal{K}$. Now, $[p^k]$ is the unique optimal feasible solution of the primal fundamental LP of $\mathcal{P}$ if and only if $\mathcal{P}$ is perfectly shortest path representable.

Proof: Sufficiency has already been discussed, so only necessity is dealt with. First, suppose that $\mathcal{P}$ is pSPR and that the designated path set for all sessions k contains only one path P_k. Then, P_k does not contain cycles in the sense of Theorem 1. In other words, any cycle in G contains at least one link (i, j), for which $p^k_{ij} = 0$. Second, suppose that there exists an alternative optimal feasible solution to P-LP$(\mathcal{P})$, say, $[x^k]$, such that for some $k \in \mathcal{K} : x^k \neq p^k$. We observe that $x^k - p^k$ is a circulation. So, let C_k denote the links in this circulation: $C_k := \{(i, j) \in E : p^k_{ij} \neq x^k_{ij}\}$. Now, C_k contains at least one link (i, j), for which $p^k_{ij} = 0$, and by the non-negativity of x^k and the definition of C_k, $x^k_{ij} > p^k_{ij} = 0$. This also means that, by Theorem 2, for some path P defined by $x^k : P \in \mathcal{P}_{min}$. This either contradicts the assumption that $\mathcal{P}_k$ contains only one path for each session or that $\mathcal{P}$ is pSPR. So, for each $k \in \mathcal{K} : x^k = p^k$, which completes the proof. ■

C. The mSPR Algorithm

At this point of the development, we have all the crucial theoretical foundations (mostly supplied by Theorem 3) to construct a simple algorithm to improve an arbitrary SPR weight set until it implements a minimal representation.

To obtain a plain SPR weight set, one simply needs to solve the corresponding fundamental LP. Many of the solution techniques enumerated in Section II-A inherently produce basic feasible solutions. However, such basic feasible solutions, by nature, contain a large number of zero valued slack variables, which calls for the formation of superfluous paths (see the related simulation results in Section V). What one needs to do is to, starting from an optimal feasible solution, search an optimal ray d for each slack variable, for which $v^k_{ij} = 0$ holds. If such ray happens to exist, then moving along the ray separates v^k_{ij} away from zero yielding an alternative SPR weight set. However, any paths of session k, which traverse link (i, j) cease to be shortest paths according to Proposition 1, because now $v^k_{ij} > 0$. Furthermore, the above operation can not decrease the value of any of the slack variables, which assures that we do not create more superfluous paths by accidently decreasing a slack variable to zero.

The last question that remained to be answered is how to obtain the optimal rays. Suppose that we are in a position to calculate the optimal ray corresponding to some slack variable v^k_{ij}. If some designated path of session k traverses link (i, j), that is, $(i, j) \in E(\mathcal{P}_k)$, then $v^k_{ij} = 0$ for all optimal solutions of D-LP$(\mathcal{P})$, so we can move to the next slack variable. Otherwise, the question we are asking is, whether or not there exists a direction of the set of optimal feasible solutions of D-LP$(\mathcal{P})$, such that moving along the direction increases v^k_{ij}. Fortunately, linear programming *Parametric Analysis* is concerned with just this kind of questions.

First, add a constraint to the problem, which restricts the feasible region to the optimal region. For example, adding (11) explicitly requires that all slack variables corresponding to the designated paths remain bound to zero. Second, perturb the objective function vector by setting the coefficient of v^k_{ij} to some arbitrary positive value and all other coefficients to zero. Finally, set the direction of the optimization to maximization. Now, either the perturbed problem is bounded, in which case no appropriate directions exist for v^k_{ij}, or otherwise it is unbounded. Let the ray causing the unboundedness of the perturbed problem be d. Notably, d has strictly positive surplus in the position corresponding to v^k_{ij} (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. So, d is the ray we have been searching for, and moving along the ray will separate v^k_{ij} away from zero. Repeating this step for each slack variable yields the *mSPR algorithm* as described in Fig. 5.

Note that if the designated path set does not prove to be SPR, then one can restart the mSPR algorithm from the modified path set defined by the solution of D-LP$(\mathcal{P})$. This, as shown in Proposition 3, yields a modified path set that is guaranteed to be SPR and capable to serve the same demands as the designated path set. Applying this minor modification assures that the mSPR algorithm produces a reasonable output for any arbitrary designated path sets thrown at it as input.

Interestingly, the mSPR algorithm is still a polynomial time algorithm, since the underlying solution technique remains to be linear programming. However, upgrading the definition form 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 $O(mK)$, which may be a tedious task in a large network with many sessions. However, we do not have to solve all problems from scratch: given an initial optimal feasible solution we can always start the parametric analysis from this solution, which, in the case of the two-phase simplex algorithm, eliminates all the tedious Phase 1 computations. Our experiments

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 $\mathcal{P}$.
ALGORITHM:

1) Solve D-LP($\mathcal{P}$). If the optimal objective is positive, then conclude that $\mathcal{P}$ is not SPR. Otherwise, let a feasible solution of D-LP($\mathcal{P}$) be $[\hat{\pi}^k, \hat{v}^k, \hat{\omega}]$.
2) For all $k \in \mathcal{K}$ and for all $(i,j) \in E \setminus E(\mathcal{P}_k)$ with $v_{ij}^k = 0$, construct the perturbed LP:

$$\begin{aligned} &\max v_{ij}^k \\ &\textstyle\sum_{k\in\mathcal{K}} v^k p^k = 0 \\ &\pi^k N + v^k = \xi + \omega \qquad \forall k \in \mathcal{K} \\ &\omega \geq 0, v^k \geq 0 \qquad \forall k \in \mathcal{K} \end{aligned}$$

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

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

3) Now, $\mathcal{W}_{\min} = \{\xi + \hat{\omega}\}$ implements a minimal shortest path representation of $\mathcal{P}$.

Fig. 5. The mSPR algorithm.

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.

IV. Link Weight Integrality

Continuing with our methodology to move from weaker definitions of shortest path representability to more and more stringent ones, in this section we impose yet another requirement on SPR link weights: integrality. In practice, the calculated link weights must always be integer-valued, but we have mostly relaxed this requirement up until now.

Since the basic feasible solutions of the minimum cost multicommodity flow problem are not guaranteed to be integral (in contrast with the single commodity case, for which they are), in many practically interesting cases solving the fundamental LP with a linear program solver yields fractional link weights [26]. The most plausible approach to overcome this difficulty would be to solve the fundamental LP as a mixed integer linear program (MIP) with the side-constraint that ω_{ij} are integral. Unfortunately, applying such a side-constraint generally transforms the (so far) polynomially tractable SPR problem into co-NP. This might prevent us from obtaining integral link weights with reasonable computational efforts in some cases. However, thanks to the special properties of the dual fundamental LP, we can always modify any SPR link weight set until it becomes integral while still retaining the polynomial tractability of our solution. In order to achieve this goal, we make the following seemingly self-evident observation:

Observation 2: Let $\mathcal{P}$ be a set of designated paths and suppose that the weight set $\mathcal{W} = \{w_{ij} : (i,j) \in E\}$ implements a shortest path representation of $\mathcal{P}$. Then, for any scalar $\lambda > 0$, the weight set $\mathcal{W}' = \{\lambda w_{ij} : (i,j) \in E\}$ is again a SPR weight set with respect to $\mathcal{P}$.

In words, one is free to multiple the weights in some SPR weight set with the same positive number, and the weight set still remains to be representing the very same path set. This is because for any two paths P_1 and P_2 with $W(P_1) < W(P_2)$, P_1 remains to be shorter over the modified weight set, since $wp_1 < wp_2 \Rightarrow (\lambda w)p_1 < (\lambda w)p_2$ for any $\lambda > 0$. Along the same lines, one can prove that equal-cost paths remain to be equal-cost. Hence, turning a fractional SPR weight set into an integral one is as easy as multiplying the weights with the *least common multiple* (l. c. m.) of their denominators. As a Theorem, we state the following:

Theorem 4: Let $\mathcal{P}$ be a path set. Then, either $\mathcal{P}$ is not SPR or otherwise, for any rational positive initial cost vector $\xi \in \mathbb{R}^m$ the weight set λw implements a minimal shortest path representation of $\mathcal{P}$, where $w = \xi + \omega$ is obtained by the mSPR algorithm and λ is the l. c. m. of the denominators in the elements of w. This can be done in polynomial time.

Proof: Since the constraint matrix, the objective function vector and the right-hand-side of D-LP($\mathcal{P}$) are rational (in fact, the first two are integral), any optimal basic feasible solution of D-LP($\mathcal{P}$) and as such, the output of the mSPR algorithm w is rational. Hence, λw defines a positive-valued, integral mSPR link weight set with λ being the l. c. m. of the denominators of the elements of w. Furthermore, since the mSPR algorithm is of polynomial complexity and Euclidean algorithm to find the l. c. m. is tractable in polynomial time too, so is the full-fledged integer mSPR problem. ■

We have already seen that one is completely free to choose the initial costs ξ. Theorem 4 in addition implies that it is not even necessary to choose ξ to be integral as long as it is positive. Moreover, any element of ξ can be made arbitrarily large and we can still find proper ω, which, together with ξ, implements a shortest path representation.

V. 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.[2] 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.[3] We used the random network topology generator, BRITE [27] 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 was provisioned randomly, by selecting two different nodes in the network with uniform probability. The capacity of the links (between 10 and 1024 units) was also uniformly distributed.

[2]http://www.gnu.org/software/glpk/glpk.html.

[3]See the Math::GLPK project page at http://qosip.tmit.bme.hu/~retvari/Math-GLPK.html.

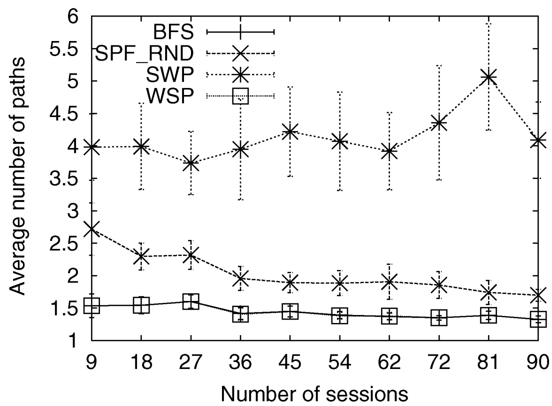


Fig. 6. Average number of paths in the plain shortest path representation.

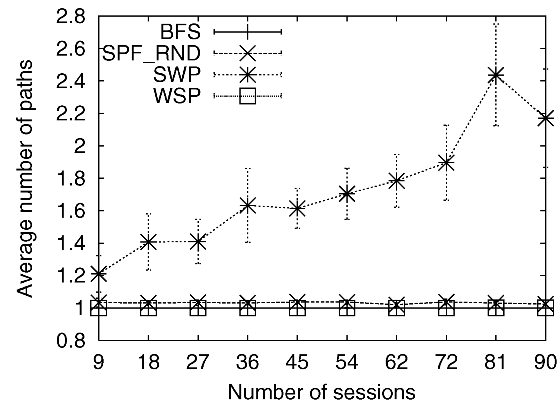


Fig. 7. Average number of paths in the *minimal* representation.

Selecting the designated paths to achieve optimal OSPF performance is NP-hard [6]. Therefore, we used quick heuristics to assign one single path to each session and observed the shortest path representation of the path sets obtained this way. The first heuristic is the breadth-first-search (BFS) algorithm, which manifests minimum hop-count routing in our simulations. Secondly, we experienced with shortest path routing over random weights chosen between 1 and 20 according to a uniform distribution (SPF_RND). This scenario represents the case when a network operator chooses the link weights randomly. Finally, we used the widest-shortest-path (WSP, [15]) and the shortest-widest-path (SWP, [28]) algorithms. Here, the length of the links was always 1 unit. Since SWP paths are not guaranteed to be SPR [14], we substituted the corresponding shortest path representation.

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. Our methodology was to generate 30 random graphs with increasing number of sessions, assign the designated paths, compute the shortest path representations and finally average the results. The figures highlight the interval estimate of the average at the level of significance of 95%. We repeated the simulations for random graph series of 35 and 45 nodes and the results were quite similar. Below, we present the results for networks of 45 nodes.

The average number of shortest paths per session in the plain SPR is depicted in Fig. 6. 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 produce shortest path representations with numerous extraneous paths. For the WSP and the BFS paths (which turned out to be fairly similar with respect to SPR) 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. We observed exceptional cases when the representation contained an astonishing number of 8 paths per session. 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 an algorithm that produces SPR paths, 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. 7, a minimal representation usually 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. 8, 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. One may argue that this notorious trait might be attributed to the intricate manner we obtained these paths (that is, by reducing a non-SPR path set to an SPR one). However, this reasoning does not seem to be pertinent, as, despite of the theoretical issues, SWP paths quite usually ended up being shortest path representable by themselves (in more than 20 out of 30 cases in our simulations regardless of K).

Finally, we compared how much real traffic could be served over the paths in the designated path set and its plain and minimal representations. For this, we computed the bottleneck bandwidth (that is, the smallest capacity along the links of the path) of every one of the paths in the path sets and depicted the average in Fig. 9. While this choice 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 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

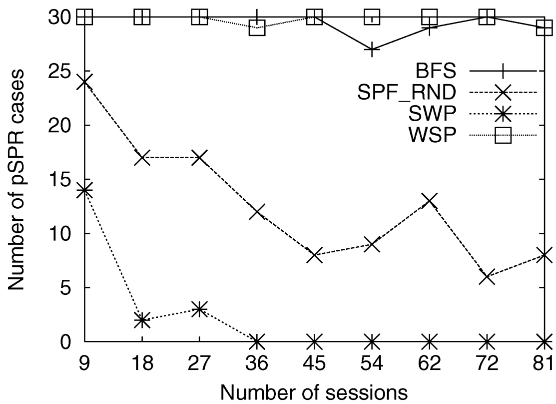


Fig. 8. Number of cases the minimal representation was perfect as well.

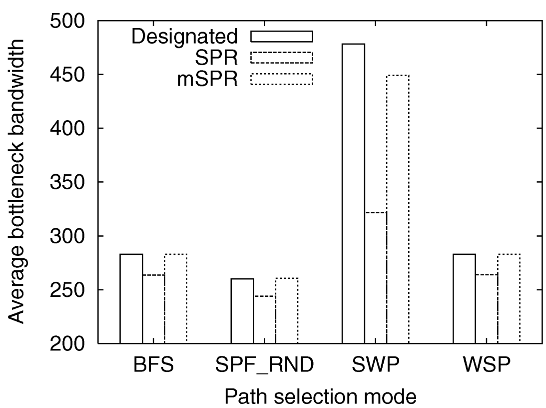


Fig. 9. Average bottleneck bandwidth of the paths.

routing supplies a really promising traffic engineering platform. Not just that SWP paths can be mapped quite accurately to shortest paths but, in addition, these paths usually provide an abundance of capacity at the same time.

VI. Contributions

In this paper, we investigated one of the most important problems concerning OSPF Traffic Engineering: the property of a path set that it can be mapped to shortest paths by positive link weights. The most important achievements can be highlighted as follows.

- Starting from the Shortest Path Optimality Conditions found in every textbook on network flow theory, through a sequence of easy steps we converted the SPR problem into one of the best-known type of linear optimization problems: a multicommodity flow problem. This allowed us to give efficient algorithms and to derive some interesting consequences.
- We gave a necessary and sufficient condition to shortest path representability and we traced back an important prior result of Wang *et al.* [18] to our theorem, which states that practically any path set useful for OSPF TE is shortest path representable.
- Realizing that a representation might contain extraneous shortest paths, ruining the performance of OSPF TE, we characterized all those paths that can be eliminated from the representation and all those paths that are not. We also gave a polynomial time algorithm to tackle the problem.
- We gave several necessary and sufficient conditions for a path set to be perfectly SPR, however, this concept turned out to be much less beneficial from a practical standpoint than minimal SPR.
- We defined the first ever provably polynomial time algorithm to compute integer-valued SPR link weights.

Throughout this paper, we have gradually moved from a weak SPR definition to more and more stringent ones, to eventually arrive to the conclusion that the problem is tractable in polynomial time in all contexts. We believe that the results presented in this paper supply both theoretical and empirical evidence to the potential of OSPF Traffic Engineering to become the tool for "poor man's traffic engineering."

Acknowledgment

The authors are grateful to Prof. P. A. Rubin for his invaluable help on the sci.op-research newsgroup and also to Prof. A. Recski for his insightful comments.

References

[1] D. Awduchen, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and Principles of Internet Traffic Engineering," RFC 3272, May 2002.
[2] Z. Wang*, Internet QoS: Architectures and Mechanisms for Quality of Service*. San Diego, CA: Academic Press, 2001.
[3] J. Moy, "OSPF Version 2," RFC 2328, Apr. 1998.
[4] *Intermediate System to Intermediate System Routing Information Exchange Protocol for Use in Conjunction With the Protocol for Providing the Connectionless-Mode Network Service (ISO 8473)*, ISO/IEC 10589:1992.
[5] B. Fortz, J. Rexford, and M. Thorup, "Traffic engineering with traditional IP routing protocols," *IEEE Commun. Mag.*, vol. 40, no. 10, pp. 118–124, Oct 2002.
[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: Proc. 3rd Int. IFIP-TC6 Networking Conf.*, Athens, Greece, May 2004, pp. 1180–1191.
[7] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing OSPF weights," in *Proc. IEEE INFOCOM*, 2000, pp. 519–528.
[8] B. Fortz and M. Thorup, "Increasing Internet capacity using local search," 2000 [Online]. Available: http://www.research.att.com/~mthorup/PAPERS/or_ospf.ps
[9] M. Pióro, A. Szentesi, J. Harmatos, A. Juttner, P. Gajowniczek, and S. Kozdrowski, "On open shortest path first related network optimisation problems," *Perform. Eval.*, vol. 48, pp. 201–223, May 2002.
[10] B. Fortz and M. Thorup, "Optimizing OSPF/IS-IS weights in a changing world," *IEEE J. Sel. Areas Commun.*, vol. 20, no. 5, pp. 756–767, May 2002.
[11] G. Rétvári and T. Cinkler, "Practical OSPF traffic engineering," *IEEE Commun. Lett.*, vol. 8, no. 11, pp. 689–691, Nov 2004.
[12] A. Farago, B. Szviatovszki, and A. Szentesi, "Inverse optimization in high-speed networks," *Discrete Appl. Math.*, vol. 129, pp. 83–98, Jun. 2003.
[13] R. K. Ahuja and J. B. Orlin, "Inverse optimization, part I: linear programming and general problem," MIT Loan School of Management, Cambridge, MA, Working paper, 1998.
[14] J. L. Sobrinho, "Algebra and algorithms for QoS path computation and hop-by-hop routing in the Internet," in *Proc. IEEE INFOCOM*, 2001, pp. 727–735.
[15] R. Guerin, A. Orda, and D. Williams, "QoS routing mechanisms and OSPF extensions," IETF RFC 2676, 1999.
[16] M. Pióro and D. Medhi*, Routing, Flow, and Capacity Design in Communication and Computer Networks*. San Diego, CA: Morgan Kaufmann, Nov 2004.
[17] W. Ben-Ameur and E. Gourdin, "Internet routing and related topology issues," *SIAM J. Discrete Math.*, vol. 17, no. 1, pp. 18–49, 2003.
[18] Z. Wang, Y. Wang, and L. Zhang, "Internet traffic engineering without full-mesh overlaying," in *Proc. IEEE INFOCOM*, 2001, vol. 1, pp. 565–571.

[19] A. Sridharan, C. Diot, and R. Guérin, "Achieving near-optimal traffic engineering solutions for current OSPF/IS-IS networks," in *Proc. IEEE INFOCOM*, 2003, vol. 2, pp. 1167–1177.
[20] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin*, Network Flows: Theory, Algorithms, and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
[21] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali*, Linear Programming and Network Flows*. New York: Wiley, 1990.
[22] J. L. Kennington, "A survey of linear cost multicommodity network flows," *Oper. Res.*, vol. 26, no. 2, pp. 209–236, 1978.
[23] T. Leighton, C. Stein, F. Makedon, É. Tardos, S. Plotkin, and S. Tragoudas, "Fast approximation algorithms for multicommodity flow problems," in *Proc. 23rd Annu. ACM Symp. Theory of Computing*, May 1991, pp. 101–111.
[24] G. Rétvári, J. J. Bíró, and T. Cinkler, "A novel Lagrangian relaxation to the minimum cost multicommodity flow problem and its application to OSPF traffic engineering," in *Proc. 9th IEEE Symp. Computers and Communications (ISCC'2004)*, Alexandria, Egypt, Jun. 2004, vol. 2, pp. 957–962.
[25] G. Rétvári, T. Cinkler, and J. J. Bíró, "Notes on Shortest Path Representation," Tech. Rep. 2005-02, Feb 2005 [Online]. Available: http://qosip.tmit.bme.hu/~retvari/publications/SPR_tech_rep.pdf
[26] J. Evans, "The simplex method for integral multicommodity networks," *Naval Research Logistics Quarterly*, vol. 25, pp. 31–37, Mar 1978.
[27] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: Universal Topology Generation From A User's Perspective," Comput. Sci. Dept., Boston Univ., Tech. Rep. 2001-003, 2001, 1.
[28] Z. Wang and J. Crowcroft, "Quality-of-service routing for supporting multimedia applications," *IEEE J. Sel. Areas Commun.*, vol. 14, no. 7, pp. 1228–1234, Jul. 1996.

Gábor Rétvári (M'03) received the M.Sc. and Ph.D. degrees in eletrical engineering from the Budapest University of Technology and Economics (BME), Budapest, Hungary, in 1999 and 2007, respectively.

He is now a Research Assistant at the High Speed Networks Laboratory, Department of Telecommunications and Media Informatics, BME. His research interests include QoS routing, Traffic Engineering and the networking applications of computational geometry and the mathematical theory of network flows. He is a Perl expert, maintaining numerous open-source scientific tools written in Perl, C, and Haskell.

József J. Bíró (M'95) received the M.Sc. and Ph.D. degrees in electrical engineering and computer science from the Budapest University of Technology and Economics (BUTE), Budapest, Hungary, in 1993 and 1998, respectively.

In 1996, he joined the Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics (BME), where he is currently an Associate Professor. His research interests include network performance modeling and evaluation, and the support of multimedia applications over heterogeneous networks.

Tibor Cinkler (M'95) received the M.Sc. and Ph.D. degrees in 1994 and 1999, respectively, from the Budapest University of Technology and Economics (BME), Budapest, Hungary, where he is currently an Associate Professor in the Department of Telecommunications and Media Informatics.

His research interests focus on optimization of routing, traffic engineering, design, configuration, dimensioning and resilience of IP, Ethernet, MPLS, ngSDH, OTN and particularly of heterogeneous GMPLS-controlled WDM-based multilayer networks. He is the author of over 130 refereed scientific publications and holds three patents. He has been involved in numerous related European and Hungarian projects including ACTS METON and DEMON; COST 266, 291, 293; IP NOBEL I and II and MUSE; NoE e-Photon/ONe and NoE e-Photon/ONe+; CELTIC PROMISE; NKFP, GVOP, and ETIK.

Dr. Cinkler is a member of ONDM, DRCN, BroadNets, AccessNets, IEEE ICC and Globecom, EUNICE, CHINACOM, Networks, WynSys, and ICTON Scientific and Program Committees. He has been a guest editor of a Feature Topic of the *IEEE Communications Magazine* and a reviewer for many journals.