Traffic Engineering with Joint Link Weight and Segment Optimization

Mahmoud Parham
University of Vienna
Faculty of Computer Science
Vienna, Austria
mahmoud.parham@univie.ac.at

Thomas Fenz
University of Vienna
Faculty of Computer Science
Vienna, Austria
thomas.fenz@univie.ac.at

Nikolaus Süss
University of Vienna
Faculty of Computer Science
Vienna, Austria
nikolaus.suess@univie.ac.at

Klaus-Tycho Foerster
TU Dortmund
Dortmund, Germany
klaus-tycho.foerster@tu-dortmund.de

Stefan Schmid
TU Berlin, University of Vienna, and Fraunhofer SIT
Germany and Austria
stefan.schmid@tu-berlin.de

ABSTRACT

Most ISPs use sophisticated traffic engineering strategies based on link weight optimizations to efficiently provision their backbone network and to serve intra-domain traffic. While traditionally, traffic is split among the shortest weighted paths using ECMP, recently, an additional dimension for optimization arose in the context of segment routing: traffic can be steered away from congested shortest paths by inserting intermediate destinations, so-called *waypoints*.

This paper investigates the benefits of jointly optimizing the link weights and waypoints for traffic engineering both analytically and empirically. In particular, we formulate the joint optimization problem and formally quantify the benefits of joint optimizations over separate link-weights and waypoints optimizations, using a rigorous analysis. We also present an efficient joint optimization algorithm and evaluate its performance in realistic and synthetic scenarios.

CCS CONCEPTS

• **Networks** → **Traffic engineering algorithms**.

KEYWORDS

traffic engineering, network algorithms, segment routing

ACM Reference Format:
Mahmoud Parham, Thomas Fenz, Nikolaus Süss, Klaus-Tycho Foerster, and Stefan Schmid. 2021. Traffic Engineering with Joint Link Weight and Segment Optimization. In *The 17th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '21), December 7–10, 2021, Virtual Event, Germany*. ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3485983.3494846

1 INTRODUCTION

Traffic engineering (TE) is a fundamental task in communication networks. To optimally use their infrastructure and avoid congestion, Internet Service Providers (ISPs) employ sophisticated algorithms to steer intra-domain traffic through their network. Many innovations in networking over the last years were at least partially motivated by the desire to improve traffic engineering [1].

Traditionally, traffic routes can be influenced only fairly indirectly by adapting *link weights*: Routing is based on the Equal-Cost-MultiPath (ECMP) protocol, in which flows are split at nodes where several outgoing links are on shortest paths to the destination, using per-flow static hashing. Thus, by changing link weights, shortest paths can be adjusted accordingly. While several clever algorithms are known today to optimize such link weights [2], such strategies provide relatively limited control over the paths taken by flows.

Segment routing (SR) [3–6] has recently introduced a powerful opportunity to optimize traffic engineering along an additional dimension: by specifying one or multiple waypoints in the packet header, traffic can be routed around potentially congested links. In particular, given a waypoint w, traffic from a source s to a destination d, is not anymore restricted to a shortest st-path, and it is routed along a shortest path from s to w (the first segment) and then from w to t (the second segment). Segment routing hence provides two knobs for traffic engineering: the link weights in the network and the sequence of waypoints to be visited along the way.

The benefits of segment routing have been demonstrated empirically in many scenarios and have received significant attention for traffic engineering [6–10]. [1] However, relatively little is known today about the fundamental algorithmic problems arising from optimizing link weights and segments jointly. This paper studies the benefits of traffic engineering mechanisms for jointly optimizing link weights and waypoints. In particular, we aim at an analytical understanding of the improvements possible by joint optimization compared to optimizing link weights and waypoints independently. For example, we will show that the usefulness of waypoints critically depends on the given weight setting and inappropriate weight settings can render waypoint optimization ineffective. Accordingly, we study algorithms for joint optimization.

[1]See [6] for a recent survey on segment routing.

1.1 Background

Traffic engineering objectives typically revolve around network utilization, which is also the focus of this paper. In particular, we are interested in the maximum link utilization (MLU): the ratio obtained by dividing the load of a link by its capacity is called *link utilization*, and MLU is simply the largest such ratio over all links.

When flows are constrained to shortest paths (OSPF) and even-splits (ECMP), the maximum link utilization MLU can be significantly larger than the optimal utilization feasible without these constraints by a factor linear in the number of nodes [11].

Traffic Engineering (TE) under OSPF and ECMP. The open shortest path protocol (OSPF) [12] is a routing protocol for the IP layer, widely used within a single domain (e.g., an ISP, an enterprise, or a datacenter). Under OSPF, a packet is always routed along the shortest path to its destination. Hence, links weights (a.k.a. link costs) determine the actual route taken by a packet, and one can influence the distribution of traffic's load across a network by tuning these weights. Under OSPF, there might be multiple (shortest path) next-hops available from a node, which is an opportunity to split the traffic over multiple paths and prevent congestion. OSPF is often installed together with Equal-Cost-MultiPath (ECMP) routing [13], the local strategy of splitting the traffic evenly between all shortest paths available at a node. We assume a fine-grained splitting model, where flows split at the packet level [14].

TE via Link-Weight Setting. Tuning link weights for congestion control is a common TE technique, where the objective is often to optimize (a function of) network resources. Link weights (as positive real numbers) determine whether a link is on the shortest path to a node or not, and they are computed offline by network operators. Link weights are often chosen to induced shortest paths that do not congest any link beyond a tolerable factor of its capacity. This practice is known as *link weight optimization (LWO)*, which is generally NP-hard even for constant-factor approximation and the case of a single source-destination pair [11, 15]. Henceforth, in practice, link weights are computed using heuristics. One such heuristic (recommended by Cisco [16]) is to assign a weight to each link proportional to the inverse of its capacity.

TE via Waypoint Setting. Given a set of demands, setting link weights alone is not always sufficient to achieve a balanced load over all links, as will also show in this paper (§3). An attractive solution enabled by segment routing is to insert intermediate destinations and force traffic flow to reach them in a specific order before arriving at its final destination. We refer to this technique as *waypoint routing*, and we refer to the problem of finding appropriate waypoints as *waypoint optimization (WPO)*. Informally, given a demand matrix, for each demand, WPO decides whether to insert waypoints for that demand and if so, it determines which of the nodes to use as waypoints for this particular demand.

TE via Joint Weights and Waypoint Setting. This paper is interested in the optimization problem obtained by combining the two previous dimensions, link weight and waypoint optimizations. The motivation behind this approach is to alleviate shortcomings with LWO and WPO by joining their benefits; 1) LWO cannot optimize link weights for each demand independently of the other demands, while waypoints can be applied to each demand separately. 2) The effectiveness of WPO depends on the given link weights.

As we will show, WPO may perform poorly when link weights are set arbitrarily or even when they are given by standard settings commonly used in practice. We consider three such weight settings: uniform weights, the inverse of capacities, and optimal weights.

1.2 Contributions

We present analytical and empirical insights into the algorithmic opportunities of jointly optimizing link weights and waypoints for traffic engineering. We make the following contributions:

The Optimality Gap. We show that the joint optimization of weights and waypoints is provably competitive against the separate optimizations (§3). We formally define a notion of competitiveness between the two TE strategies, referring to it as the *optimality gap*. We advocate the effectiveness of the joint optimization by comparing its network utilization to the utilization under link weight and waypoint optimizations separately. We show that their joint optimization can improve network utilization by a factor in $\Omega(n \log n)$, where n is the number of nodes (Theorem 3.15).

We provide an upper bound for the gap in $O(n \log n)$ (§4), which implies our LB is tight on general networks (i.e., general graphs and capacities) and single source-target demands. We prove that the gap does not exist under uniform capacities and single source-target demands (Theorem 4.2). Additionally, the gap does grow by n, if the source and target are sparsely connected, e.g., when they are connected by a constant number of paths with disjoint capacities (Theorem 4.3).

Approximation Algorithm for Weight Optimization. As part of our gap analysis, we present an algorithm (§5) that computes a link weight setting that minimizes the maximum link utilization approximately. To the best of our knowledge, this is the first polynomial-time approximation algorithm for this problem on general directed networks (although single source-target demands), with a provable factor that depends on the number of nodes (in contrast to [17] which restricts capacities to a finite set of integers and the factor depends on the size of this set). Our approximation factor also serves as an upper bound for the extend of MLU improvement achievable in the joint optimization when compared to separate single optimizations (Corollary 4.4). Moreover, it shows that our example captures the worst case, i.e., the gap is tight.

Heuristic Algorithm. Since the general problem is NP-hard, in §6, we present a heuristic that extends the local search algorithm introduced in [11] by combining it with a greedy waypoint setting algorithm. We evaluate the quality of our algorithm on a variety of real-world topologies.

Empirical Gap Observation. We provide a mixed-integer linear program (MILP) formulation of the joint optimization problem available in [18]. We use the formulation to demonstrate the TE gap on small examples. For large networks, we run our heuristic and compare the resulting MLU to that of standard weight settings and computed by local search [11].

Artifacts. To contribute to the research community, ensure reproducibility, and support future research in this area, we release all our experimental artifacts and implementations as open source together with this paper [18].

1.3 Related Work

Traffic engineering is an evergreen topic in networking. Besides adjusting link weights in IP networks, traffic engineering can also be performed using MPLS [19] or centralized controllers as in software-defined networking [20]. We employ segment routing [3], which is a relatively recent TE approach.

Fortz and Thorup in [11] showed that OSPF with ECMP can result in an MLU that is larger than the optimal feasible MLU under arbitrary flows by a factor in $\Omega(n)$ where n is the number of nodes. They also prove the NP-hardness of LWO and present a "local search" heuristic. We first show a linear gap between LWO and Joint (§3) using a similar network construction, and then we present a stronger construction (§3.5) that improves the gap by a logarithmic factor. Generally, it is known that weight setting is NP-hard to approximate within any constant factor even with single source-target demands [15, 17]. We complement this inapproximability result with a polynomial-time approximation algorithm and a provable approximation guarantee.

Chiesa et al. [15] provided impossibilities for several weight optimization problems. In particular, they showed that LWO is NP-hard on general topologies and even on the special class of hypercubes. They presented one positive result on special topologies known as "folded Clos networks": an ad-hoc algorithm producing weights and an optimal congestion (as in OPT). When capacities are restricted to a finite set of integers, Pióro et al. [17] show a constant factor approximation for the maximum LWO, while our approximation works with arbitrary real capacities.

Segment routing [3] for traffic engineering has been considered in, e.g., [7, 8, 21, 22], see [6] for a recent survey on SR. Moreno et al. [22] using linear programming and heuristics showed that a very limited number of stacked labels suffice to exploit the benefits of segment routing successfully. Using a worst-case construction, we demonstrate that SR cannot benefit TE under inappropriate weight settings, even with an arbitrarily large number of segments. Aubry et al. [8, 23] lay the algorithmic foundations of segment routing, considering different and related applications, however, primarily focusing on the waypoint optimization. However, we are not aware of any analytical quantification of the benefits of joint waypoint and weight optimization with segment routing.

2 MODEL AND PROBLEM DEFINITION

We next define our key terminology and notations formally.

Network Instance. We model a network as a tuple $\mathcal{N} = (V, E, c)$, where V is a set of $n := |V|$ vertices (nodes or routers), E is the set of directed links (communication links) connecting nodes, and the mapping $c : E \mapsto \mathbb{R}^+$ assigns to each link $\ell \in E$ a capacity $c_\ell > 0$.

A *flow* is an assignment $f : E \mapsto \mathbb{R}^+$ that respects flow conservation constraints. The *load* on a link ℓ denoted by $f_\ell \geq 0$ is the amount of the (total) flow assigned to this link. Whenever the source and target of a flow f is clear from the context, $|f|$ denotes the total size of the flow emitting (entering) the source (the target). A *demand list* is a multiset denoted by $\mathcal{D}$ and consists of tuples $(s, t, d) \in \mathcal{D}$, where $s, t \in V$ are the *source* and *target* (destination) nodes of the demand, and $d \in \mathbb{R}^+$ is its size (i.e., required bandwidth). A *weight setting* $w : E \mapsto \mathbb{R}^+$ assigns a positive real $w_\ell > 0$ to each link $\ell \in E$. A *waypoint* is a node assigned to a demand, and serves as an intermediate destination for the flow of that demand. That is, the flow of that demand must reach the waypoint prior to reaching its final destination. A *waypoint setting* denoted by $\pi : V^W \mapsto \mathcal{D}$ assigns up to W waypoint nodes to each demand, where W is a given parameter.

TE Instance. The input to all our TE problems is a *traffic engineering instance (TE-instance)*, denoted by the tuple $\mathcal{I} = (\mathcal{N}, \mathcal{D}, \omega)$, where $\mathcal{N}$ is the given network, $\mathcal{D}$ is the given demand list with total demand size $D := \sum_{(s,t,d)\in\mathcal{D}} d$, and ω is the given weight setting.

Using these notations, we formally describe the TE objective and optimization problems considered in this paper.

Maximum Link Utilization (MLU). Given a network $\mathcal{N}$ and a flow assignment f, the *utilization* of any link ℓ under the flow f is the ratio f_ℓ / c_ℓ. The *network utilization* is the maximum link utilization under f, that is, $MLU(\mathcal{N}, f) := \max_{\ell\in E} f_\ell / c_\ell$.

Even-Split Flow (ES-Flow). The aggregate flow that enters a node v and is destined to a node t may *split* over (a subset of) outgoing links of v. An *even-split* flow (ES-flow) either does not split at v or it splits evenly at this node, i.e., it may not split arbitrarily as in OPT.

ECMP Flow . If an even-split flow exiting a node v is constrained to traverse only the outgoing links of v that are on a shortest path to t, then we refer to it as an *ECMP-flow.*

2.1 Problem Definition

The minimal input common to all our optimization problems consists of a network $\mathcal{N}$ with n nodes and a demand set $\mathcal{D}$. We refer to a demand list where all demands share the same source-target pair (s, t) as a *single source-target* demand list. Next, we formally define each problem and its additional inputs separately.

Link-Weight Optimization (LWO). Given $(\mathcal{N}, \mathcal{D})$ as an input, the LWO problem computes a link weight setting such that the induced ECMP-flow minimizes MLU.

Waypoint Optimization (WPO). The input is $(\mathcal{N}, \mathcal{D}, w)$, where w is a weight setting. WPO selects an ordered set of up to W waypoints for each demand in $\mathcal{D}$. The ECMP-flow must reach each waypoint in the given order before finishing at t. WPO chooses these waypoints so that the MLU under the total flow is minimized.

Joint Link-Weight and Waypoint Optimization (Joint). Given the input $(\mathcal{N}, \mathcal{D})$, Joint computes a link-weight setting and a sequence of up to W waypoints for each demand such that MLU is minimized when flows are routed through shortest paths between consecutive waypoints.

The MLU for an instance $\mathcal{I}$ is denoted by LWO($\mathcal{I}$), WPO($\mathcal{I}$), and Joint($\mathcal{I}$), respectively, under an optimal weight setting, an optimal waypoint setting, and an optimal joint weight and waypoint setting. We omit $\mathcal{I}$ wherever it is clear from the context.

Note that in comparison to OPT, in the three latter problems, a flow must follow shortest path links, and it must split equally over all outgoing links that belong to a shortest path. Hence, LWO is equivalent to Joint when $W = 0$. By definition, if Joint and WPO are constrained to at most W waypoints (per demand), then

$$\text{OPT} \leq \text{Joint} \leq \min\{\text{LWO}, \text{WPO}\}. \tag{2.1}$$

Clarifying terminology on Joint. We emphasize that Joint is not equivalent to applying LWO and WPO separately (e.g., sequentially), and it generalizes both of these optimizations, and hence, it is stronger. Joint is specifically the optimization problem of minimizing MLU over the Cartesian product of all link-weight settings and all waypoint settings. However, our heuristic approximates Joint using a sequential optimization.

The Optimal Flow (OPT). OPT is the multi-commodity flow problem with the objective of minimizing the MLU subject to link capacities (formulated in [18]). Note that OPT has no routing restrictions; it may assign a positive flow to any link, and it may split a flow arbitrarily at any node. We denote the optimal MLU of a TE-instance $\mathcal{I}$ under the optimal flow by OPT$(\mathcal{I})$. Let f^* be a maximum (s,t)-flow. Observe that OPT $\geq D/|f^*|$.

Acyclic Maximum Flow. An *acyclic maximum* (s,t)*-flow* from a source s to a target t always exists: 1) take any maximum flow, 2) find a cycle and a link with the smallest flow value on this cycle, 3) subtract this value from the flow of every link of the cycle. 4) repeat from step 2 until the flow is acyclic. It is easy to see that the new flow has the same size and the algorithm terminates in polynomial time.

3 OPTIMALITY GAPS

In this section, we investigate the question of how competitive Joint is compared to LWO and WPO? To this end, we present several network instances where Joint yields a value for MLU noticeably smaller than the optimal values obtained from LWO or WPO separately. This will immediately imply that applying Joint is necessary in order to utilize the best of weights and waypoints. We assume single source-target demands throughout this section. We will study gaps (as ratios) between the optimal MLU feasible in Joint and the optimal values from LWO and WPO separately.

Definition 3.1. Given a network instance $\mathcal{I}$, the *optimality gap* between Joint and each of the two problems is defined as ratios:

$$R_{\text{LWO}}(\mathcal{I}) := \frac{\text{LWO}(\mathcal{I})}{\text{Joint}(\mathcal{I})}, \text{ and } \qquad R_{\text{WPO}}(\mathcal{I}) := \frac{\text{WPO}(\mathcal{I})}{\text{Joint}(\mathcal{I})}.$$

The ratio R_{WPO} depends on the given link-weight setting. We restrict the given weight setting to the special cases that are often used in practice. LWO does not take any weight setting as input.

Definition 3.2. We refer to any of the following weight settings as a *standard weight setting*.

- *Unit weights*: the weight 1 set for every link.
- *Inverse of capacities*: the weight of each link equals the reciprocal of its capacity (*capacity*$^{-1}$).
- Optimal weights: weight settings optimal for LWO.

We refer to general weight settings as *arbitrary*.

Definition 3.3. We define the *TE gap* as the worst-case ratio between the best MLU obtainable from individual optimizations and the one from Joint. Formally,

$$R^* := \max_{\mathcal{I}} \min\{R_{\text{LWO}}(\mathcal{I}), R_{\text{WPO}}(\mathcal{I})\}.$$

We may drop the instance $\mathcal{I}$ where it is irrelevant or clear from the context. Note that (2.1) implies $R_{\text{LWO}}, R_{\text{WPO}}, R^* \geq 1$.

Table 1: TE gaps for single source-target demands

Weights	Capacities	TE-Gaps	
Lower Bounds (Cor. 3.16)		$2 \leq W \in O(1)$	$W = 1$
arbitrary	arbitrary	$\Omega(n \log n)$	$\Omega(n)$
uniform	arbitrary	$\Omega(n \log n)$	$\Omega(n)$
capacity^{-1}	arbitrary	$\Omega(n \log n)$	$\Omega(n)$
optimal	arbitrary	$\Omega(n \log n)$	$\Omega(n)$
Upper bounds		$R^* \leq R_{\text{LWO}}$	
optimal	uniform	1, Theorem 4.2	
Thm. 4.3	arbitrary	$\lvert E\rvert$, Theorem 4.3	
optimal	arbitrary	$n \log n$, Corollary 4.4	

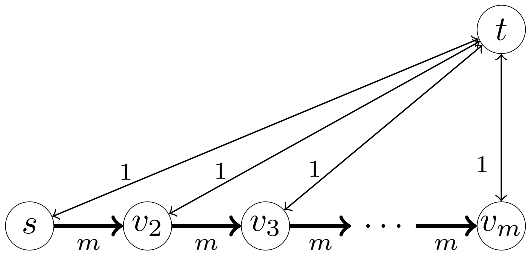


Figure 1: See TE-Instance 1. There are $m = n - 1$ unit-size demands from s to t. The optimal flow and Joint are able to split one unit size flow away from the thick path at each node v_i. This is not possible under shortest path routing which causes a large gap in $\Omega(n)$ between Joint and LWO.

Intuitively, a lower bound for the gap R^* indicates how far Joint can lower the MLU over the best of the other two optimizations. That is, a larger lower bound (i.e., a larger gap) emphasizes the necessity of applying Joint over separate optimizations.

Table 1 summarizes our findings for the gap in two major cases: i) when Joint is restricted to $W = 1$ waypoint per demand (Theorem 3.4), and ii) when Joint is restricted to 2 waypoints per demand (Corollary 3.16). Note that WPO is granted the extra privilege of using more waypoints as long as W is a constant.

The remainder of this section is devoted to showing the following statement.

Theorem 3.4. *There exist network instances with n nodes and demand lists that admits a TE gap $R^* \in \Omega(n)$ (Definition 3.3), when Joint and WPO are restricted to at most $W = 1$ waypoint per demand.*

We present a network instance that we use to prove Theorem 3.4. Similar examples are presented in [17] and [2] to show the gap between the optimal flow and the LWO for single demand cases. Here we adopt a similar example as in [17] and [11] and show that the linear bound holds for our gap ratio as well.

TE-Instance 1 (Figure 1). *Consider the network in Figure 1. It consists of n nodes $\{v_i, 1 \leq i \leq m\} \cup \{s,t\}$ where $s = v_1$, arcs $(v_i, v_{i+1}), 1 \leq i < m$ each with capacity m, and arcs $(v_i, t), 1 \leq i \leq m$*

each with capacity 1. *There are $m = n - 1$ unit-size demands from s to t. The optimal flow routes each demand via one of the paths with capacity 1, e.g., it may route the ith demand through s, v_i, t, which yields the optimal utilization OPT* = 1.

Next, we observe that in this instance the optimal MLU is feasible in Joint using only one waypoint per demand.

Lemma 3.5. *Instance 1 admits Joint = OPT = 1 using up to one waypoint for each demand.*

Proof. We observe that the optimal flow is feasible also for Joint using the following waypoint and weight setting.
i) Set the node v_i as a waypoint for the ith demand. ii) Set the weight m for every link $(v_i, t), (t, v_i), 1 \le i \le m$, and iii) set the weight 1 for the other (horizontal) links. The flow of the ith demand runs through the unique shortest path to v_i, that is $s, v_2, \dots, v_i$, which has the cost i. The (unique) shortest path from v_i to t has the cost m and it consists of the link (v_i, t). Therefore, Joint assigns the (unit-size) flow of the ith demand to the unit-capacity path $s, v_2, \dots, v_i, t$. This flow assignment is identical to that of optimal flow and therefore Joint = OPT = 1. □

3.1 Optimizing with Link Weights

Optimizing (only) weights may lead to an MLU significantly larger than the optimal feasible MLU in Joint. We show that applying only LWO may result in an MLU that is worse than the optimal from Joint by a factor in $\Omega(n)$.

Consider Instance 1 (Figure 1). Any optimal even-split flow splits evenly at a node $v_i, i < m$. Assume w.l.o.g. that the optimal flow splits evenly at $v_1 = s$. As a result, half of the flow runs through the link (s, t) having the capacity 1, that is, the load $m/2$ on this link. Observe that splitting at the latter nodes $v_i, i \ge 2$ does not improve the MLU. We set the weight 2 for the link (s, t) and the weight 1 for every other link. This weight setting realizes the optimal even-splitting flow that splits only over the two shortest paths s, t and s, v_2, t. We conclude this case by comparing the MLU in LWO and Joint using Lemma 3.5 in the following statement.

Lemma 3.6. *The optimal link weight setting for TE-Instance* 1 *yield a maximum link utilization LWO* $\ge (n-1)/2$.

Proof. We observe that the size maximum feasible even split (s, t)-flow is 2 units which implies the claim for the total demand size $D = m = n - 1$. From v_m to t, only 1 unit of ES-flow is feasible. We reason that the size of the maximum ES-flow feasible via each $v_i, i < m$, is 2. From v_{m-1} to t, 2 units of ES-flow splits into two unit size parts, one part traverses the path v_{m-1}, t and the other part reaches t through the node v_m. In general, at $v_i, i < m$, only 2 units can be delivered via the neighbor v_{i+1}, i.e., if the flow does not split at v_i. If the flow splits at v_i then each of the two paths delivers 1 units. Therefore, the size of the maximum the ES-flow at $v_i, i < m$ is 2, whether it splits into two or not. □

3.2 Optimizing with Waypoints

Waypoint optimization on top of arbitrary or standard weight settings is not always competitive to Joint. That is, applying only WPO may result in an MLU worse than the optimal of Joint by a factor in $\Omega(n)$.

Lemma 3.7. *Given an arbitrary or a standard weight setting (Definition 3.2), there exists a network instance of n nodes on which the optimal waypoint setting from WPO subject to $W = 1$ waypoint (per demand) yields a maximum link utilization WPO* $\ge (n-1)/3$.

Proof. We show the claim in several cases of the given weight setting using TE-Instance 1 (Figure 1) denoted by $\mathcal{I}_1$.

Arbitrary Weights. Consider the weight setting for $\mathcal{I}_1$ that assigns the weight $\epsilon = 1/3$ for all links connected to t, i.e. to links $(v_i, t), (t, v_i)$ for every $1 \le i \le m$, and it assigns the weight 1 to every other link. Under this weight setting, the shortest path from s to t is the link (s, t). The shortest path from s to any node v_i is s, t, v_i. Therefore, with any choice of waypoints, the flow of every demand uses the link (s, t), which leads to WPO = $m = n - 1$.

Uniform Weights. Assume every link has the weight 1 in $\mathcal{I}_1$. Consider any assignment of zero or one waypoint to each demand. The flow that has v_3 as waypoint must split evenly over the two shortest paths s, v_2, v_3 and s, t, v_3 with equal costs of 2. All flows without a waypoint use the path s, t. Each flow using the node $v_i, i \ge 4$ as waypoint takes the shortest path s, t, v_i. Thus, an optimal waypoint setting may distributes the total load into at most three parts which assigns at least 1/3 of the load to the link (s, t) and therefore WPO $\ge m/3 = (n-1)/3$.

Inverse of Capacities. In this case, the weight of each link equals the reciprocal of its capacity. We transform the network instance into a new instance $\mathcal{I}_1'$ with $n = 2m + 1$ nodes (same demands). We replace each of the links (s, v_2) and (v_2, v_3) with m unit-capacity paths of two links as follows:

(1) Add $2m$ new nodes $u^1, \dots, u^m$ and $z^1, \dots, z^m$.
(2) Replace (s, v_2) with m paths where the jth path consists of unit-capacity links $\{(s, u^j), (u^j, z^j), (z^j, v_3)\}$ for $1 \le j \le m$.
(3) Set capacity of every new link to 1.

Consider the weight setting for $\mathcal{I}_1'$ that sets the weight of each link to the inverse of its capacity. Every link $(v_i, v_{i+1}), i \in \{3, \dots, m\}$, has the weight $1/m$, and every other links has the weight 1. For $i \ge 2$, any path $s, u^j, z^j, v_2, \dots, v_i$ has a cost at least 3. Then the shortest path from s to any node $v_i, i \ge 2$ is s, t, v_i of cost 2. Therefore, with any waypoint setting in WPO, the link (s, t) having capacity 1 receives the entire load and WPO$(\mathcal{I}_1') = m = (n-1)/2$.

Optimal LWO Weights. We show even when the given weight setting is optimal for LWO, the MLU obtained from WPO can be significantly larger than that of Joint.

By Lemma 3.7, the MLU under an optimal weight setting is is LWO $\ge m/2$, as it splits the flow into two parts at a node v_i. Consider the weight setting that assigns the weight 2 to the link (s, t) and the weight 1 to every other link. Under this weight setting, the flow of size m splits evenly at $v_1 = s$ into two parts of size $m/2$, overloading the link (s, t) by the factor $m/2$. Hence, LWO = $m/2$ and the weight setting is optimal.

We show that using a single waypoint per demand under the optimal weight setting does not improve the MLU beyond an additive factor. Assume w.l.o.g. that the optimal waypoint setting assigns v_i as the waypoint for the ith demand. The shortest paths to t and v_2 are respectively s, t and s, v_2, which are used by the first two demands without splitting. The third demand is destined to its waypoint v_3 and splits at s, since both paths s, t, v_3 and s, v_2, v_3

have the same cost of 3. The shortest path to v_i for every $i \geq 4$ is uniquely s, t, v_i. Thus, the link (s, t) receives a flow larger than $m-4$ and hence WPO $> m - 4 = n - 5$ in the optimal weight setting. □

3.3 Bounding the TE Gap

We can now show the TE gap claimed in Theorem 3.4.

Proof of Theorem 3.4. We show the claim for TE-Instance 1 illustrated in Figure 1. First, we derive the individual gap ratios of Definition 3.1. From Lemma 3.5, we have Joint = 1. Lemma 3.6 implies $R_{\text{LWO}} = \text{LWO}/1 \geq (n-1)/2$. Due to Lemma 3.7, under arbitrary and standard weight settings (Definition 3.2) the instance admits a gap $R_{\text{WPO}} = \text{WPO}/\text{Joint} = \text{WPO}/1 \geq (n-1)/3$, when both Joint and WPO are constrained with $W = 1$. Hence, by Definition 3.3, we conclude the claimed TE gap $R^* \geq (n-1)/3 \in \Omega(n)$. □

3.4 The Special Case of Uniform Capacities

So far we considered instances with arbitrary (i.e., non-uniform) capacities. We will show later (in Theorem 4.2) that gaps larger than 1 do not exist in the special case of uniform capacities and single source-target demands. In particular, we show when all link capacities are equal and all demands share the same source-target pair then LWO = Joint = OPT, which implies $R_{\text{LWO}} = 1$. The gap R_{LWO} can be larger than 1 when there are demands with arbitrary source-target pairs.

Theorem 3.8. *Under uniform capacities and arbitrary source-target pairs, there exists an instance with n nodes where the MLU is in $\Omega(n)$ under either of LWO or WPO, when the latter is restricted to one waypoint and standard/arbitrary weight settings.*

Proof. Consider the TE-Instance 1 (Figure 1) and denote it by $\mathcal{I} := ((G, c), \mathcal{D})$. We construct a new instance $\mathcal{I}' := ((G, c'), \mathcal{D}')$, where c' is the uniform capacity assignment that assigns the capacity m to every link, and $\mathcal{D}'$ is a demand list generated as follows. i) Initialize $\mathcal{D}' = \mathcal{D}$. ii) $\forall \ell = (u, v) \in G, c(\ell) < m : \mathcal{D}' = \mathcal{D}' \cup \{(u, v, m - c(\ell))\}$. That is, $\mathcal{D}'$ includes all demands in $\mathcal{D}$ and an additional demand between the endpoints of each link in $\mathcal{I}$ with capacity less than m. It is not difficult to see that under all link-weight settings that we use in our lower bound analysis, the (unique) shortest path between the endpoints of a link is the link itself. Therefore, under both LWO and WPO, each demand in $\mathcal{D}' \setminus \mathcal{D}$ is routed along the link that connects its endpoints. Hence, once we deduce the capacities occupied by these additional demands, the residual capacities are exactly those assigned by c (as in Figure 1). Thus, our lower bounds hold for $\mathcal{I}'$ via an analysis similar to that of Theorem 3.4. □

3.5 Amplifying the TE Gap

In this section, we introduce an instance that admits a TE gap $R^* = \Omega(n \log n)$ (Theorem 3.15). First, we introduce and analyze an instance that offers a logarithmic gap, and later, we combine it with TE-Instance 1 to obtain a new instance that admits larger gap.

TE-Instance 2 (Figure 2a). *In Figure 2a, there are $m = n - 2$ demands from s to t, where demand sizes constitute the harmonic series $H_m := 1, 1/2, \ldots, 1/m$. The size of the maximum (s,t)-flow through this network is $\sum_{1 \leq k \leq m} 1/k \approx \ln(m)$.*

Next, we analyze the optimal even-split flow for Instance 2.

Lemma 3.9. *Let f be any maximum even-split (s, t)-flow on Instance 2. The flow splits evenly over a subset of (s, t)-paths with capacities that form a prefix of H_m.*

Proof. Let f be the maximum ES-flow and P be the subset of paths s, w_j, t used by f. Equivalently to the statement of the claim, P consists of paths with consecutive capacities (in H_m) at least $1/j^*$ for some $1 \leq j^* \leq m$. Then, in a maximum ES-flow, the path $p^* \in P$ with the smallest capacity $1/j^*$ is saturated, the load on each of the paths in P is $1/j^*$, and the size of the ES-flow is $|f| = |P| \cdot 1/j^*$. Assume for contradiction that this is not the case, that is, for some $k < j^*$, P does not contain a path p with a larger capacity $1/k > 1/j^*$. Consider the set $P' = \{p\} \cup P \setminus \{p^*\}$. Let f' denote the ES-flow that splits evenly over all paths in P'. The smallest capacity in P' is at least $1/(j^* - 1)$. Since $|P'| = |P|$, we obtain $|f'| = |P'| \cdot 1/(j^* - 1) = |P| \cdot 1/(j^* - 1) > |f|$, which contradicts f being a maximum ES-flow. □

Lemma 3.10. *In the network Instance 2, the size of the maximum even-split (s, t)-flow is 1.*

Proof. By Lemma 3.9, for some $1 \leq j^* \leq m$, the maximum ES-flow f splits over a subset of paths s, w_j, t, $1 \leq j \leq j^*$ with capacities that constitute the first j^* numbers in H_m. The smallest capacity of these paths is $1/j^*$ which yields $|f| = j^* \cdot 1/j^* = 1$. □

Next, we introduce a network instance that has Instance 2 as a substructure, and admits a gap $R_{\text{LWO}} \in \Omega(n \log n)$.

TE-Instance 3 (Figure 2b). *Consider the network in Figure 2b with n nodes and the parameter $m = n/2$. Capacities of links (v_i, w_1), $\ldots, (v_i, w_m)$ form the harmonic series $H_m = \{1, 1/2, \ldots, 1/m\}$. Equivalently, every link connected to w_j, $1 \leq j \leq m$, has the same capacity $1/j$. All links (v_i, v_{i+1}), (w_i, w_{i+1}) have the capacity D. There are m^2 demands from $s = v_1$ to $t = w_m$. Demand sizes can be partitioned into m subsets, where each subset constitutes the Harmonic series H_m. Hence, the total demand size is $D = m \cdot \sum_{1 \leq k \leq m} 1/k \approx m \cdot \ln(m)$.*

Next, we show that for TE-Instance 3, two waypoints are sufficient to obtain a joint waypoint and weight setting that admits an ES-flow with *MLU* = 1.

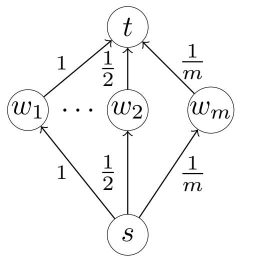
Lemma 3.11. *In TE-Instance 3 (Figure 2b), using only two waypoints per demand, Joint achieves a utilization Joint = 1.*

Lemma 3.12. *TE-Instance 3 admits a gap $R_{LWO} \in \Omega(n \log n)$ when Joint is constrained with $W \geq 2$ waypoints per demand.*

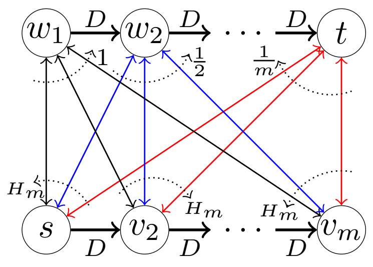
The proof of lemmas 3.11 and 3.12 can be found in Appendix A.

We now present another instance similar to TE-Instance 3 that differs only in link capacities. We will use it to lower bound the gap R_{WPO} (Definition 3.1).

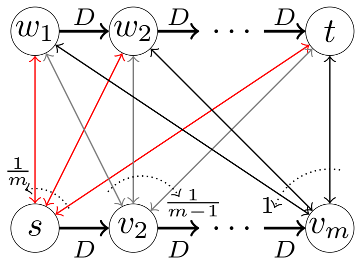
TE-Instance 4 (Figure 2c). *Consider the network in Figure 2c with n nodes and the parameter $m = n/2$. Each link (v_i, w_j), $1 \leq i, j \leq m$, has the capacity $1/(m-i+1)$. All links (v_i, v_{i+1}), (w_i, w_{i+1}) have the same capacity $D \geq m \ln m$, where D is the total demand size. The demand list identical to that of Instance 3.*



(a) See TE-Instance 2. The gap R_{LWO} is in $\Omega(\ln n)$ given $m = n - 2$ demands from s to t with harmonic sizes.



(b) See TE-Instance 3. The gap R_{LWO} is in $\Omega(n \log n)$ under m^2 demands with sizes that constitues m harmonic sets H_m, $m = n/2$, with total demand size D.



(c) See TE-Instance 4. Demands are identical to 2b. The gap R_{WPO} is in $\Omega(n \log n)$ which holds for any constant number of waypoints in WPO.

Figure 2: Each instance has n nodes. Arc labels represent link capacities. Bi-directed arcs represent two directed links in the opposite directions with equal capacities. All demands are between $s = v_1$ and t. The parameter m is in $\Omega(n)$. In each instance, JOINT = OPT = 1 using up to two waypoints per demand. In 2b, OPT and JOINT separate one harmonic subset of demands away from the thick path at each node v_i causing a large gap between LWO and JOINT, while in 2c, they separate a subset of equal-size demands away from the thick path at each v_i causing large gap between WPO and JOINT. In short, OPT and JOINT are capable of fine-grained control over individual paths unlike LWO and WPO.

Similarly to Lemma 3.11, we show that for TE-Instance 4, two waypoints are sufficient to obtain a joint waypoint and weight setting that admits an ES-flow with $MLU = 1$.

LEMMA 3.13. *In TE-Instance 4 (Figure 2c), using two waypoints (per demand), JOINT achieves the utilization JOINT = 1.*

The proof of Lemma 3.13 can be found in Appendix A.

In the following lemma, we lower-bound the gap R_{WPO} when the input weight setting is arbitrary or the standard weight setting from Definition 3.2.

LEMMA 3.14. *Let $\mathcal{I}_3$ denote TE-Instance 3 (Figure 2b), and $\mathcal{I}_4$ denote TE-Instance 4 (Figure 2c). Assume the number of waypoints per demand for JOINT is constrained to $W \geq 2$, and for WPO, it is constrained to $W = \lceil c \cdot n \rceil$, $0 < c < 1/3$.*

(i) If the given weight setting is arbitrary, uniform, or the inverse of link capacities, then the instance $\mathcal{I}_4$ admits a gap

$$R_{WPO}(\mathcal{I}_4) \in \Omega((n \log n)/W).$$

(ii) If the given weight setting is the optimal from LWO then $\mathcal{I}_3$ admits a gap $R_{WPO}(\mathcal{I}_3) \in \Omega((n \log n)/W)$.

The proof of Lemma 3.14 can be found in Appendix A.

TE-INSTANCE 5. *Let $\mathcal{N}_3 = (G_3, c_3)$ denote the network instance 3 and (s_3, t_3) denote its source-target pair. Let $\mathcal{N}_4 = (G_4, c_4)$ denote instance 4 and (s_4, t_4) denote its source-target pair. We define the TE instance $(\mathcal{N}_5, \mathcal{D})$ where*

$$\mathcal{N}_5 = (G_3 \cup G_4 \cup \{(t_3, s_4)\}, c_3 \cup c_4 \cup \{(t_3, s_4) \mapsto D\})$$

is the concatenation of the two networks $\mathcal{N}_3$ and $\mathcal{N}_4$ with $n = 4m + 2$ nodes. The source node in $\mathcal{N}_5$ is s_3 and the target node is t_4. The link (t_3, s_4) with capacity D connects the target of $\mathcal{N}_3$ to the source of $\mathcal{N}_4$, where D is the total demand size. The demand list is identical to that of instances 3 and 4. That is, $\mathcal{D}$ consists of m^2 demands with sizes that can be partitioned into m identical harmonic sets H_m.

We aggregate all our optimality gaps into the (joint) TE gap R^* from Definition 3.3.

THEOREM 3.15. *TE-Instance 5 admits a gap $R^* \in \Omega(n \log n/W)$ if*

(1) WPO is constrained to $W \leq c \cdot n$ waypoints for $c \leq 1/3$,
(2) JOINT is constrained to 2 waypoints, and
(3) the given weight setting input to WPO is arbitrary, or it is the standard setting from Definition 3.2.

The proof is in the Appendix A. We unify the lower bounds from Theorems 3.4 and 3.15 (restricting $W \leq 2$) as follows:

COROLLARY 3.16. *When WPO is constrained to a constant number of waypoints (per demand) at least W, and JOINT is constrained to at most W waypoints, the TE gap (Definition 3.3) satisfies $R^* \in \Omega(n)$ for $W = 1$, and $R^* \in \Omega(n \log n)$ for $W = 2$.*

4 UPPER BOUNDING THE GAP

In this section, we compare optimal values in JOINT and LWO by studying the worst case of the ratio LWO/JOINT and its upper bounds. This ratio is at least 1 since any feasible solution to LWO is feasible also in JOINT for $W = 0$ (no waypoints). We consider only single source-target demands in this section.

First, we restate a helper lemma from [15]. Given a directed acyclic graph (DAG) G, it ensures a weight setting on G such that the ECMP-flow under this weight setting assigns positive flows to all links of G, i.e., every link is on a shortest path to the target node.

LEMMA 4.1 (LEMMA 2.5 IN [15]). *Assume a given DAG G containing nodes s and t without any incoming link to s and outgoing link from t. There exists a weight setting for G such that the DAG of the induced ECMP-flow is identical to G.*

4.1 Uniform Capacities

We show that with uniform link capacities, applying LWO is sufficient to obtain the optimal MLU (Theorem 4.2). This immediately concludes the TE gap R^* = LWO/JOINT = 1.

THEOREM 4.2. *Given a network and a list of demands $\mathcal{D}$ between the same source and target nodes, if the capacity of all links are equal (i.e. uniform) then LWO = OPT.*

PROOF. Assume all links have the capacity C. We denote the capacity of a minimum cut between s and t by $cut(s,t)$. Thus, OPT $\geq \mathcal{D}/cut(s,t)$. By Menger's theorem [24], there exists a set of link-disjoint paths (corresponding to an acyclic maximum flow) denoted by $\mathcal{P}$ from s to t such that $C \cdot |\mathcal{P}| = cut(s,t)$. Therefore, OPT $\geq D/(C \cdot |\mathcal{P}|)$. We refer to paths in $\mathcal{P}$ as *basic* paths.

We construct a ES-flow that is feasible for LWO, then we show that the MLU under this flow is optimal, that is, LWO = OPT. Consider the DAG $G := \cup_{P \in \mathcal{P}} P$ obtained by taking the union of all basic paths in $\mathcal{P}$. By applying Lemma 4.1 to G, we obtain link weights such that every path in $\mathcal{P}$ is a shortest path to t. We set the weight of every other link $\mathcal{N} \setminus G$ to a number sufficiently large in order to ensure only links of basic paths are on a shortest path to t. Let $\mathcal{F}$ be the ES-(s,t)-flow in G (splitting evenly over links of basic paths).

Next, we show that $\mathcal{F}$ assigns the same flow size $D/|\mathcal{P}|$ to every link that belongs to any shortest path to t. The aggregate ES-flow at s splits evenly over all the outgoing links of s that are on a basic path (i.e. links in G). Hence, the flow on each of these outgoing links is of size $\mathcal{D}/|\mathcal{P}|$. We show by induction that the flow on every other link is $D/|\mathcal{P}|$ as well. Consider the topologically sorted ordering of nodes $v_1 = s, \ldots, v_n = t$ in G. Assume the load on each outgoing links of every node $v_j, j < i < n$ is $\mathcal{D}/|\mathcal{P}|$. That is, the load on each link incoming to v_i is $D/|\mathcal{P}|$. We show the same flow size is assigned the outgoing links of v_i. The number of incoming links equals the number of outgoing links at v, as otherwise some basic paths share the same (incoming/outgoing) link of v_i, which contradicts the assumption they are link-disjoint. Therefore, each outgoing link carries a flow of size $D/|\mathcal{P}|$ concluding our claim.

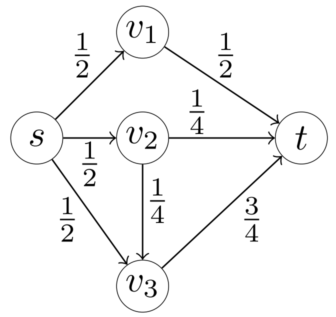
Thus, every link that belongs to a basic path (i.e. to G) carries a portion of the aggregate flow of the size $D/|\mathcal{P}|$, which implies the maximum utilization in $\mathcal{F}$ is $(D/|\mathcal{P}|)/C = D/(C \cdot |\mathcal{P}|) =$ OPT. □

Consider any maximum (s,t)-flow f^* and its decomposition into set of paths $\mathcal{P}$ s.t. $\sum_{P \in \mathcal{P}} c(p) = |f^*|$ (by *flow decomposition* theorem in [25]). We provide upper bounds for the TE gap in the number of paths in a flow decomposition and also in the number of links.

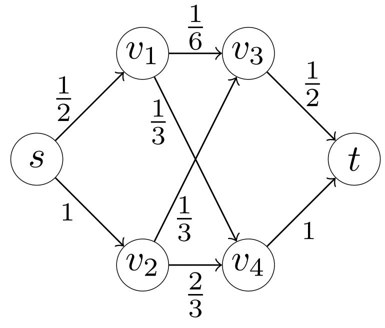
THEOREM 4.3. *Given the flow decomposition $\mathcal{P}$ of any maximum (s,t)-flow, the optimality gap for an optimal weight setting satisfies $R_{LWO} \leq |\mathcal{P}| \leq |E|$.*

PROOF. Let $p_{max} \in \mathcal{P}$ be the path with the largest capacity in the flow decomposition $\mathcal{P}$ and let $c(p_{max})$ be its capacity, i.e., the capacity of the weakest link on p_{max}. Set the weight of every link in p_{max} to 1 and the weight of every other link to n. Therefore, p_{max} is the unique shortest (s,t)-path in $\mathcal{N}$. The MLU under this weight setting is $D/c(p_{max})$ and

$$\begin{aligned} \text{LWO} &\leq \frac{D}{c(p_{max})} = \frac{|\mathcal{P}| \cdot D}{|\mathcal{P}| \cdot c(p_{max})} \leq \frac{|\mathcal{P}| \cdot c(p_{max})}{\sum_{P \in \mathcal{P}} c(p)} \\ &= \frac{|\mathcal{P}| \cdot D}{|f^*|} \leq |\mathcal{P}| \cdot \text{OPT}(\mathcal{P}). \end{aligned}$$



(a)
$ec(t) = \infty$
$ec(v_1) = \frac{1}{2}$
$ec(v_2) = 2 \times \frac{1}{4}$
$ec(v_3) = \frac{3}{4}$
$ec((s,v_1)) = ec(v_1) = \frac{1}{2}$
$ec((s,v_2)) = ec(v_2) = \frac{1}{2}$
$ec((s,v_3)) = ec(v_3) = \frac{3}{4}$
$ec(s) = 3 \times ec((s,v_1)) = \frac{3}{2}$
$|f^*| = \frac{3}{2} = ec(s)$



(b)
$ec(t) = \infty$
$ec(v_3) = \frac{1}{2}$
$ec(v_4) = 1$
$ec(v_1) = 2 \times \frac{1}{6} = \frac{1}{3}$
$ec(v_2) = 2 \times \frac{1}{3} = \frac{2}{3}$
$ec((s,v_1)) = ec(v_1) = \frac{1}{3}$
$ec((s,v_2)) = ec(v_2) = \frac{2}{3}$
$ec(s) = 2 \times ec((s,v_1)) = \frac{2}{3}$
$|f^*| = \frac{3}{2} = 2.25 \times ec_t(s)$

Figure 3: Two examples illustrating the effective capacity of nodes and links from Definition 5.1. The label of each link represents its capacity which for simplicity is chosen to be also its usable capacity, i.e., the value assigned by a maximum (s,t)-flow f^*. In (3a), the effective capacity of the node s equals its usable capacity. In (3b), the effective capacity of s is less than half of its usable capacity.

It is known from the flow decomposition theorem that $|\mathcal{P}| \leq |E|$. Thus, we conclude

$$R_{\text{LWO}} = \frac{\text{LWO}}{\text{JOINT}} \leq \frac{\text{LWO}}{\text{OPT}} \leq |\mathcal{P}| \leq |E|. \quad \square$$

Note that Theorem 4.3 refines Theorem 3.15 for sparse networks, as it eliminates the logarithmic factor when $|E| \in O(n)$.

4.2 General Networks

For networks on arbitrary graphs, arbitrary capacities, and single source-target demands, we show that the gap ratio is bounded above by a factor in $O(n \log n)$ (Corollary 4.4). This immediately implies our lower bound in Theorem 3.15 is asymptotically tight.

In Section 5, we introduce the algorithm LWO-APX that approximates the optimal weight setting within a factor in $O(n \log n)$ of the optimal setting (Theorem 5.4). This bound implies

$$R_{\text{LWO}} = \frac{\text{LWO}}{\text{JOINT}} \leq \frac{\text{LWO}}{\text{OPT}} \leq \frac{\text{LWO-APX}}{\text{OPT}} \in O(n \log n).$$

Hence, the following corollary follows immediately.

COROLLARY 4.4. *If all demands share the same source-target pair and JOINT can assign an arbitrary number of waypoints to any demand (i.e., $W = n$), then $R^* \in O(n \log n)$.*

5 APPROXIMATING OPTIMAL WEIGHTS

We introduce an approximation algorithm for LWO (Algorithm 1) that computes a weight setting for instances where all demands are defined between the same source-target pair (s, t). We fix an acyclic maximum (s, t)-flow denoted by f^*. The capacity of a link may be arbitrarily larger than $|f^*|$ (the size of the flow). To obtain minimal capacities without decreasing the flow size, we redefine the capacity of each link ℓ as $c^*(\ell) = f^*(\ell)$, i.e., the fraction of the flow through ℓ, and we refer to it as *usable capacity.* Next, we refine the notion of usable capacity and define a notion of capacity for nodes and links that is minimal for maximal ES-(s, t)-flows.

Effective Capacity. Informally, the *effective capacity (e-capacity)* of a node v with respect to the target node t, denoted by $ec_t(v)$, is the size of the maximal ES-flow from v to t subject to c^*. Since the flow splits evenly at v, an outgoing link of v having the smallest e-capacity determines the size of the maximal ES-flow feasible from v. The e-capacity of a node v is equal to the smallest e-capacity of links in $out_G(v) = \{\ell \mid \ell = (v, *)\}$ times the *out-degree* of v denoted by $\delta_G(v) = |out_G(v)|$. The e-capacity of a link $\ell = (*, u)$ is bounded by the e-capacity of u and $c^*(\ell)$, i.e., the size of the maximum (s, t)-flow through ℓ subject to the original capacities c. We define the notation ec_t formally as follows.

Definition 5.1. Assume a network $N = (G, c)$, source-target pair (s, t), and an acyclic maximum (s, t)-flow f^* are given. Let G^* be the acyclic subgraph of G consisting of links with a positive flow under f^*. Let $c^*(\ell) = f^*(\ell)$ be the usable capacity for each link ℓ. We define the effective capacity of all nodes and links w.r.t t inductively. We define $ec_t(t) = \infty$. Assume the effective capacity of all outgoing links of a node $v \neq t$ is already determined. Then

- $ec_t(v) = \delta_{G^*}(v) \times \min_{\ell'=(v,*)} ec_t(\ell')$,
- $\forall \ell = (*, v) : ec_t(\ell) = \min\{c^*(\ell), ec_t(v)\}$.

We use $ec(.)$ in place of $ec_t(.)$ when t is clear from the context. Definition 5.1 is also an algorithm that computes the size of a ES-flow from every node to the target node t. This algorithm is embedded in Algorithm 1, which computes an approximately maximum ES-flow. Figure 3 illustrates the distinction between the usable capacity of nodes and links, i.e., the size of the maximum (s, t)-flow feasible through them, and their e-capacity. In both examples, the e-capacity of each link connected to t equals its usable capacity. The e-capacity of the remaining links and nodes are obtained in a backward traversal starting from t. Observe that in Figure 3b, there are two choices at v_1: splitting the flow evenly over its two outgoing links yields the same e-capacity as not splitting and using only the link with larger e-capacity. Therefore in such cases, we break ties arbitrarily, e.g., by always splitting. In contrast, if we split the flow evenly at v_2 then the size of the maximum ES-flow is limited to $2 \times 1/4 = 1/2$, while not splitting allows for a larger ES-flow of size $2/3$ from v_2. Algorithm 1 selects a subset of outgoing links at every node such that splitting the flow evenly over these links yields an approximately optimal ES-flow.

Now we describe Algorithm 1 (LWO-APX), which employs Definition 5.1 for computing an approximately optimal weight setting. The algorithm works in two stages. The first stage (Lines 5–10) computes an ES-flow with a size at least $\approx |f^*|/n \log n$. It begins with computing an acyclic maximum (s, t)-flow and its corresponding DAG G^*. Then each iteration i removes certain outgoing links of v_i from G^* to optimize the ES-flow from v_i. Specifically, for each node v_i in the reverse topological ordering of nodes of G^*, the algorithm at Line 7 selects a subset of outgoing neighbors such that the effective capacity of v_i is (locally) maximized. Then it removes the remaining outgoing links of v_i from G^*. Note that at each iteration, the effective capacity of all outgoing links has been determined in previous iterations. After all these iterations, G^* reduces to a possibly sparser DAG where the size of the ES-flow on this DAG is within our approximation guarantee. The second stage (Line 11) produces a weight setting that realizes the ES-flow computed in the first stage.

Algorithm 1: Algorithm LWO-APX

```
   Input  : directed network $\mathcal{N}$, nodes $s$ and $t$
   Output : weight setting $E \mapsto \mathbb{R}^+$
   // initialization
1  Let $G^*$ denote the subgraph of $G$ consisting of links $\ell$ with
     positive flow $c^*(\ell) = f^*(\ell) > 0$ in an acyclic maximum
     $(s, t)$-flow $f^*$.
2  Let $v_1, \dots, v_n$, where $v_1 = t, v_n = s$, be nodes of $G^*$ sorted in
     the reverse topological ordering, and let $\delta_i = \delta_{G^*}(v_i)$
     denote the number of $v_i$'s outgoing links in $G^*$.
3  Initialize $ec : V \mapsto \mathbb{R}^+$ for assigning effective capacities.
   // maximizing the e-capacity of $s$
4  Set $ec(t) = \infty$ ;                    // the e-capacity of $t$
5  for $i = 2$ to $i = n$ do                 // for each node $v_i \neq t$
6      Let $\ell_1, \dots, \ell_{\delta_i}; \ell_j \in out(v_i)$ be the outgoing links of $v_i$
         sorted s.t. $ec(\ell_1) \geq \cdots \geq ec(\ell_{\delta_i})$.
7      Let $j^* = \arg\max_{1 \leq j \leq \delta_i} \{j \cdot ec(\ell_j)\}$.
8      $ec(v_i) = j^* \cdot ec(\ell_{j^*})$ ;          // e-capacity of $v_i$
9      $\forall \ell \in in(v_i) : ec(\ell) = \min\{c^*(\ell), ec(v_i)\}$
       //removing links of low e-capacity
10     $G^* = G^* \setminus \{\ell_{j^*+1}, \dots, \ell_{\delta_i}\}$
11 return the weight setting for $G^*$ from Lemma 4.1.
```

5.1 Analysis

We first show that at the end of the ith iteration, the e-capacity of v_i may be smaller than the sum of children's e-capacity by at most a logarithmic factor. Equivalently, the ith iteration reduces the size of the maximal ES-flow from v_i to t by a logarithmic factor of its out-degree $\delta_{G^*}(v_i)$. Intuitively, the algorithm takes the optimal flow (with optimal flow-splits) and in $|V|$ iterations transforms it into a flow that splits evenly everywhere. Each iteration yields an intermediate flow with one more even-splitting node than the previous flow; this is the reason behind the capacity reduction. The worst-case of this reduction occurs when e-capacities of the outgoing links of v_i constitute a harmonic series. The following lemma and its proof formalize the idea.

Lemma 5.2. *Consider the DAG G^* of the maximum (s, t)-flow (Line 1.1). The total effective capacity of v_i's outgoing links in G^* is at most a logarithmic factor larger than v_i's effective capacity. That is, $\sum_{\ell=(v_i,*)} ec(\ell) \leq \lceil \ln(\delta_{G^*}(v_i)) \rceil \cdot ec(v_i)$.*

Proof. Consider the outgoing links of v_i in the non-decreasing order of their effective capacities at line 1.6. The algorithm at line 1.7 selects the first j^* values in this ordering s.t. $ec(v_i) = j^* \cdot ec(\ell_{j^*})$ is maximized. Thus, for each outgoing link at ℓ_j, we have $j \cdot ec(\ell_j) \leq j^* \cdot ec(\ell_{j^*})$ and

$$ec(\ell_j) \leq (1/j) \cdot j^* \cdot ec(\ell_{j^*}) = (1/j) \cdot ec(v_i). \tag{5.1}$$

The total effective capacity of all outgoing links of v_i is

$$\sum_{1 \leq j \leq \delta_{G^*}(v_i)} ec(\ell_j) \leq ec(v_i) \cdot \sum_{1 \leq j \leq \delta_{G^*}(v_i)} (1/j) \leq ec(v_i) \cdot \lceil \ln(\delta_{G^*}(v_i)) \rceil,$$

which follows from (5.1) and the fact $\sum_{1 \leq y \leq z} \approx \ln(z)$. □

The effective capacity of the source node determines the MLU under the weight setting computed by Algorithm 1. We show that the effective capacity of s may be smaller than its usable capacity, i.e. the size of the maximum (s, t)-flow, by at most the factor $n \log n$. That is, we show $c(s) \leq n \log n \cdot ec(s)$, which implies the approximation factor $n \log n$. Recall that $ec(u)$, the effective capacity at the node u, is the size of the maximum ES-flow from u to t. Next, we show that $ec(s)$ may be smaller than the usable capacity from s to t at most by the factor that depends on the number of nodes n.

Lemma 5.3. *Let Δ^* denote the largest splitting factor of the maximum (s, t)-flow f^* in $\mathcal{N}$. Then,*

$$|f^*| \leq n \cdot \lceil \ln(\Delta^*) \rceil \cdot ec_t(s).$$

Proof Idea. Recall that each iteration (Line 5) determines the e-capacity of a node v_i, which by Lemma 5.2 may be smaller than the ec of v_i's children by a logarithmic factor of their number. We interpret each iteration (Line 5) as a process that reduces the usable capacity at s. Consider the reverse of this process where we undo the ith iteration by switching from even-split to optimal split at v_i. Before the reverse process, v_i splits its flow evenly its usable capacity is $ec(v_i)$. Afterward, v_i splits optimally and its usable capacity is $\sum_{u \in out(v_i)} ec(u) \geq ec(v_i)$. We upper bound the impact of each reversed iteration on s (for each v_i) in a formal proof in Appendix A. We conclude the approximation factor of LWO-APX as follows.

Theorem 5.4. *The weight setting from Algorithm 1 induces a maximum ECMP-flow smaller than the maximum (s, t)-flow (i.e. OPT) by at most a factor in $O(n \log n)$.*

Proof. Assume a given TE-instance $\mathcal{I} = (\mathcal{N}, \mathcal{D}, w^*)$ comprising a network $\mathcal{N}$, a single source-target demand list $\mathcal{D}$, and a weight setting w^* from Algorithm 1.

Let Δ^* be the largest out-degree in $\mathcal{N}$. Let LWO-APX$(\mathcal{I})$ denote the MLU under the ECMP-flow induced by w^*. Recall that OPT$(\mathcal{I}) = D/|f^*|$ is the MLU under the optimal (arbitrary split) flow, where $|f^*|$ is the size of the maximum (s, t)-flow.

By the definition of effective capacity, the MLU under w^* is LWO-APX$(\mathcal{I}) = D/ec(s)$. Then from Lemma 5.3 and the fact that $\Delta^* < n$, we obtain LWO-APX$(\mathcal{I}) = D/ec(s) \leq$

$$\frac{D}{|f^*|/(n\lceil \ln(\Delta^*) \rceil)} = \frac{D \cdot n\lceil \ln(n) \rceil}{|f^*|} = n \cdot \lceil \ln(n) \rceil \cdot \text{OPT}(\mathcal{I}),$$

which concludes the claim. □

Algorithm 2: JOINT-Heur

```
  Input  : network $\mathcal{N}$, demand list $\mathcal{D}$
  Output : weight and waypoint settings
1 Run HeurOSPF [11] and let $\omega$ be the weight setting result.
2 Run GreedyWPO using $\omega$ to obtain a waypoint setting $\pi$.
3 Replace each demand $\psi = (s, t, d) \in \mathcal{D}$ with two new
    demands: $(s, \pi_\psi, d)$ and $(\pi_\psi, t, d)$, and let $\mathcal{D}'$ be the new
    demand list.
4 Run HeurOSPF on $\mathcal{D}'$ to obtain a new weight setting $\omega'$.
5 return $\omega'$ and $\pi$.
```

Algorithm 3: Waypoint Selection (GreedyWPO)

```
  input  : network instance $(\mathcal{N}, \mathcal{D})$, weight setting $\omega$
  output : waypoint setting $\pi : \mathcal{D} \mapsto V$
1 Let $U_{min}$ be the MLU of the ECMP-flow for $\mathcal{D}$ induced by $\omega$.
2 for each demand $\psi = (s, t, d) \in \mathcal{D}$ in the descending order of
    demand sizes do
3     for each node $w \in V$ do // improving MLU greedily
4         Let $\mathcal{D}' := \mathcal{D} \setminus \{\psi\} \cup \{(s, w, d), (w, t, d)\}$.
5         Let $U$ be the MLU under $\mathcal{D}'$ and induced by $\omega$.
6         If $U < U_{\min}$ then set $\pi_\psi = w$ and $U_{\min} = U$.
7 return $\pi$.
```

6 ALGORITHM FOR JOINT

As the Joint problem is NP-hard, we propose a polynomial-time algorithm that employs an adaptation of the local search algorithm of Fortz and Thorup [11] as a subroutine (referred as HeurOSPF). Given a general demand list (arbitrary source-target pairs) and a general network instance $\mathcal{I} = (\mathcal{N}, \mathcal{D})$, Algorithm 2 computes a weight setting and a waypoint setting by running the two optimizations separately and iteratively. While it is not exactly a joint optimization, it is a first attempt to approximate Joint. Our experiments show that even such a simplistic heuristic can improve the utilization beyond what is feasible with weight optimization.

7 EMPIRICAL GAP OBSERVATIONS

In §3, we provided bounds on worst cases of the gap (ratio) between Joint and LWO. In this section, we observe the gap on a collection of real network topologies with real and synthetic traffic demands [18].

Test Environment. All simulations were executed on an HP DL380 G9 with 2x Intel Xeons E5-2697V3 SR1XF with 2.6 GHz, 14 cores each, and a total of 128 GB DDR4 RAM. The host machine was running Ubuntu 18.04.4 LTS. We implemented the proposed algorithms in Python (3.7.10) [26] leveraging the libraries NetworkX (2.5.1) [27], NetworKit (8.1) [28] Numpy (1.20.3) [29], and SciPy (1.6.3) [30]. To solve the MIPs/LPs we used Gurobi (9.1.2) [31].

Data Sources. For our simulations, we used real-world traffic data and topology from SNDLib [32–34], which is provided in a standardized XML file format. SNDLib provides a large set of data for the Abilene, Géant and Germany50 topologies consisting of traffic matrices in various granularity levels. Our second real-world source is TopologyZoo [35, 36] which provides only topology data

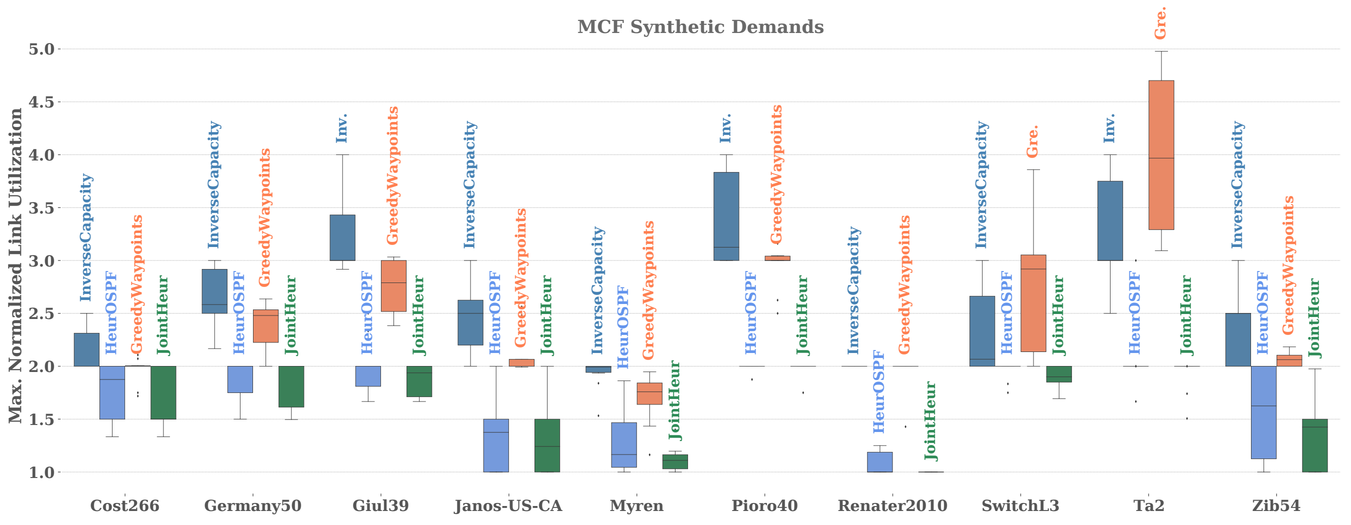


Figure 4: MLU statistics from different algorithms on the 10 largest capacitated non-tree topologies from TopologyZoo and SNDLib. Demands are generated using the MCF Synthetic method.

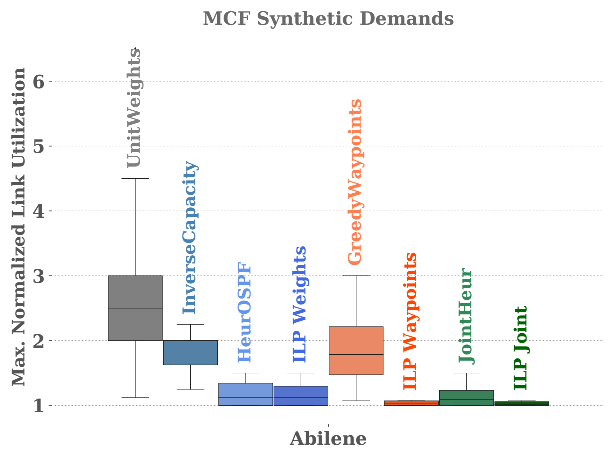


Figure 5: Comparing MILP Results to Heuristics.

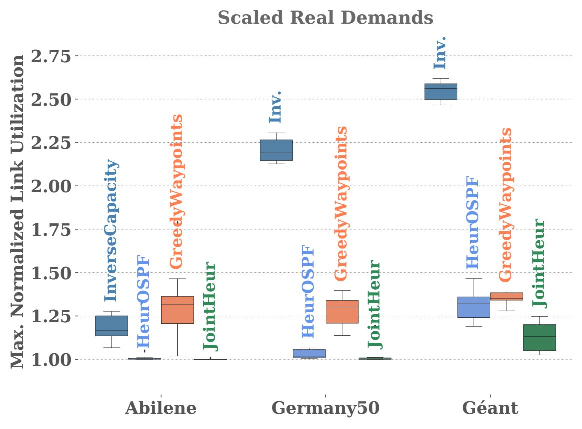


Figure 6: MLU under real demands and SNDLib topologies.

in GraphML format. We selected topologies with capacity information and used the latest version of a topology if multiple ones exist. We synthesize traffic demands using the maximal concurrent multi-commodity flow (MCF) [18] as follows.

Demand generation. Due to the proprietary nature of ISPs and backbones we could not procure real traffic data. We extracted the traffic data from research repositories and synthesized traffic for the evaluation of arbitrary topologies.

MCF Synthetic Demands. In all our experiments with non-real demands, we generated demands using the maximal multi-commodity flow (MCF) formulated in [18]. We randomly select 20% of connection pairs and scale the demand between those as such that the objective MLU computed by the MCF routing is 1. The resulting demands are further split into smaller sub-demands so that each source-target pair has multiple equal-size flows. The number of flows per pair scales with the topology size, or precisely, by $|E|/4$.

7.1 Comparing the different Algorithms

We now present our empirical results on the previously described real-world and synthetic data. Each algorithm is evaluated with 10 sets of demands on each of the selected topologies.

Small Networks. The MILP formulation of JOINT is presented in [18]. Due to its complexity, we computed the optimal solutions using the MILP solver only on Abilene with 12 nodes and 30 links. We derive formulations for LWO and LWO from the JOINT's MILP separately: for LWO, we simply set $W = 0$; for WPO, given a weight setting ω', we add one constraint for each link ℓ: $\omega_\ell = \omega'(\ell)$. We obtained optimal solutions from each MILP on small examples and results are depicted in Figure 5. The average MLU for WPO, LWO, and JOINT respectively is 1.17, 1.04, and 1.03.

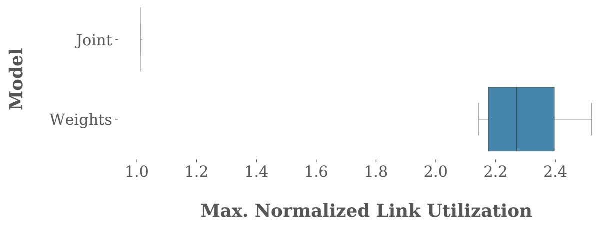


Figure 7: Nanonet Experiments: Joint vs. LWO

Large Networks In Figure 4, we evaluated the heuristic algorithms on all topologies from TopologyZoo and SNDLib, which provide complete link capacity information. The average MLU of the naive algorithm InverseCapacity is 2.74 which improves to 1.65 using HeurOSPF (from [11]) and further to 1.58 using our JOINT-Heur. The running time of our JOINT-Heur is longer than that of HeurOSPF by up up 3min in the largest topology with $|V| = 65$ nodes. The improvements from applying a second weight optimization (steps 3 and 4) is negligible. The first two stages of JOINT-Heur (steps 1 and 2) are already sufficient for most of evaluated instances, and the plots are obtained only from the first two steps.

Real Demands. In Figure 6, we evaluated the gap reduction on real demands from SNDLib. Note that for real traffic demands, all connection pairs are active, though a huge skew can be observed. The average MLU of 1.11 using the best LWO heuristic, i.e. HeurOSPF, improves to 1.05 using our joint optimization.

Discussion. In all of the experiments, the MCF is $MLU = 1$ since we scale all demand sizes. Figure 5 shows that the Joint MILP achieves an average MLU of 1.03, but its complexity prevents employing it on larger network instances. Our JOINT-Heur improves the average MLU on nearly all evaluated topologies compared to the LWO heuristic HeurOSPF. Particularly, the instances using real demands (Figure 6) indicate the potential of optimization utilizing waypoints on top of conventional LWO techniques. Here we obtained an average MLU of 1.05. The overhead execution time due to waypoint setting is negligible compared to the execution time of HeurOSPF.

7.2 Nanonet Implementation

To further verify our model, we performed a small investigation in Nanonet [37], a virtualized network environment conceptually based off Mininet. Nanonet simulates network nodes by creating network namespaces in the Linux kernel and (virtual) links between them. Shortest path routes are calculated directly by Nanonet, which also supports ECMP. To achieve better splitting, Layer-4 hash was used by setting `net.ipv6.fib_multipath_hash_policy=1` on all (virtual) nodes. In our experiments, we use the TE-Instance 1 with optimal solutions for LWO and Joint and measure the link utilization to obtain the MLU. For the Joint experiment, additional routes, possibly including a segment, are added to the sources. As perfect splitting is not the case due to the hash functions even with an L4 hash, we add four additional (pseudo) nodes, each for one flow. The flows all start at the same time and run for 300 seconds. For the throughput evaluation we use `nuttcp` 6.1.2, with 32 streams per source and limit the bandwidth to the total demand size. Thus, for Weights, the rate is set to 40 MBit/s with 32 parallel streams, and in case of Joint the rate is set to 4 times 10 MBit/s with 32 parallel streams each. The results of 10 executions of each test are depicted in Figure 7. The results at Joint are relatively constant and match the expected normalized value of 1 closely, with some deviation due to, e.g., Neighbor Discovery Protocol packets. However, Weight has a wider range beyond the expected MLU of 2, caused by the hash function, where the flows are not split perfectly across the equal cost routes. More precisely, Joint in our 10 test executions has MLU results of approximately ≈ 1.0138, while Weight has a range from ≈ 2.1439 to ≈ 2.5219, with a median of ≈ 2.2704. For the latter, parallel equal cost routes have a roughly correspondingly reduced link utilization. We hence conclude that the expected results are closely matched in Nanonet, with some deviation caused by the L4 hash function.

8 CONCLUSION

Motivated by the traffic engineering flexibilities introduced by segment routing architectures, we studied analytically and empirically the benefit of jointly optimizing link weights and waypoints (mathematically modeled as an optimality gap) to improve network utilization. We showed that already in relatively simple settings, the joint optimization can help significantly compared to individual optimizations: the achievable maximum link utilization can be improved by a factor linear in the number of nodes n, respectively even $\Omega(n \log n)$ for dense networks. We gave an $O(n \log n)$ approximation algorithm for link-weight optimization in a single source-target setting, which also serves as an upper bound for our optimality gap, rendering the gap asymptotically tight.

To evaluate the gap empirically, we presented both a mixed-integer linear program formulation and an efficient heuristic for joint waypoint and weight optimization. Our heuristic is scalable and provides fair utilization benefits over prior work, evaluated on various real-world topologies. It would be interesting to see an analysis for heuristics such as ours that combine the two optimization strategies sequentially. We leave open questions: how well a sequential approach can approximate the optimal Joint solution in practical instances? How may iterations and how many waypoints would be sufficient to achieve the best outcome within a reasonable runtime? We also validate our model setting in a proof-of-concept experiment in a Mininet-like environment.

Overall, we see our work as a first step towards unifying the two TE strategies, and we believe that it opens several interesting avenues for future work. We assumed static traffic demands and focused on the worst-case analysis of the loss if the two optimizations are not performed jointly. It would be interesting to explore TE algorithms that react to shifts in the traffic demand and account for reconfiguration costs.

ACKNOWLEDGMENTS

This research received funding from the Austrian Science Fund (FWF), grant I 4800-N (ADVISE), a joint D-A-CH project with DFG in Germany. We would also like to thank our shepherd Sanjay Rao and the reviewers for their valuable feedback.

REFERENCES

[1] Nick Feamster, Jennifer Rexford, and Ellen Zegura. The road to sdn: an intellectual history of programmable networks. *ACM SIGCOMM Computer Communication Review*, 44(2):87–98, 2014.

[2] Bernard Fortz and Mikkel Thorup. Internet traffic engineering by optimizing OSPF weights. In *INFOCOM*, pages 519–528. IEEE Computer Society, 2000.

[3] Clarence Filsfils, Stefano Previdi, Bruno Decraene, Stephane Litkowski, and Rob Shakir. Segment routing architecture. In *IETF Internet-Draft*, 2017.

[4] Clarence Filsfils, Pierre Francois, Stefano Previdi, Bruno Decraene, Stephane Litkowski, Martin Horneffer, Igor Milojevic, Rob Shakir, Saku Ytti, Wim Henderickx, Jeff Tantsura, Sriganesh Kini, and Edward Crabbe. Segment routing architecture. In *Segment Routing Use Cases, IETF Internet-Draft*, 2014.

[5] Clarence Filsfils, Stefano Previdi, John Leddy, S. Matsushima, and D. Voyer. IPv6 Segment Routing Header (SRH). Internet-Draft draft-ietf-6man-segment-routing-header-14, Internet Engineering Task Force, June 2018. Work in Progress.

[6] Pier Luigi Ventre, Stefano Salsano, Marco Polverini, Antonio Cianfrani, Ahmed Abdelsalam, Clarence Filsfils, Pablo Camarillo, and François Clad. Segment routing: A comprehensive survey of research activities, standardization efforts, and implementation results. *IEEE Commun. Surv. Tutorials*, 23(1):182–221, 2021.

[7] Renaud Hartert. *Fast and scalable optimization for segment routing.* PhD thesis, UCLouvain, 2018.

[8] FranÁois Aubry. *Models and Algorithms for Network Optimization with Segment Routing.* PhD thesis, UCLouvain, 2020.

[9] A. Sgambelluri, F. Paolucci, A. Giorgetti, F. Cugini, and P. Castoldi. Experimental demonstration of segment routing. *Journal of Lightwave Technology*, 34(1):205–212, 2016.

[10] Randeep Bhatia, Fang Hao, Murali Kodialam, and TV Lakshman. Optimized network traffic engineering using segment routing. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 657–665. IEEE, 2015.

[11] Bernard Fortz and Mikkel Thorup. Increasing Internet Capacity Using Local Search. *Computational Optimization and Applications*, 29(1):13–48, 2004.

[12] J. Moy. Ospf version 2. Technical report, April 1998.

[13] C. Hopps. Analysis of an equal-cost multi-path algorithm. rfc 2992. Technical report, April 2000.

[14] Advait Abhay Dixit, Pawan Prakash, and Ramana Rao Kompella. On the efficacy of fine-grained traffic splitting protocolsin data center networks. In *SIGCOMM*, pages 430–431. ACM, 2011.

[15] Marco Chiesa, Guy Kindler, and Michael Schapira. Traffic engineering with equal-cost-multipath: An algorithmic perspective. *IEEE/ACM Transactions on Networking*, 25(2):779–792, 2017.

[16] Cisco. Configuring ospf. Technical report, April 1997.

[17] Michal Pióro, Áron Szentesi, János Harmatos, Alpár Jüttner, Piotr Gajowniczek, and Stanislaw Kozdrowski. On open shortest path first related network optimisation problems. *Perform. Evaluation*, 48(1/4):201–223, 2002.

[18] Thomas Fenz, Klaus-Tycho Förster, Mahmoud Parham, Stefan Schmid, and Nikolaus Süß. Traffic engineering with joint link weight and segment optimization. September 2021. https://whatif-tools.net/segment-routing.

[19] Xipeng Xiao, Alan Hannan, Brook Bailey, and Lionel M. Ni. Traffic engineering with MPLS in the internet. *IEEE Netw.*, 14(2):28–33, 2000.

[20] Sugam Agarwal, Murali S. Kodialam, and T. V. Lakshman. Traffic engineering in software defined networks. In *INFOCOM*, pages 2211–2219. IEEE, 2013.

[21] Randeep Bhatia, Fang Hao, Murali S. Kodialam, and T. V. Lakshman. Optimized network traffic engineering using segment routing. In *INFOCOM*, pages 657–665. IEEE, 2015.

[22] Eduardo Moreno, Alejandra Beghelli, and Filippo Cugini. Traffic engineering in segment routing networks. *Computer Networks*, 114:23–31, 2017.

[23] François Aubry, Stefano Vissicchio, Olivier Bonaventure, and Yves Deville. Robustly disjoint paths with segment routing. In *CoNEXT*, pages 204–216. ACM, 2018.

[24] Frank Göring. Short proof of menger's theorem. *Discret. Math.*, 219(1-3):295–296, 2000.

[25] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications.* 1993.

[26] Python 3.7.10, Last Accessed: 2021-06-28. https://www.python.org/downloads/release/python-3710/.

[27] Networkx 2.5.1, Last Accessed: 2021-06-28. networkx.github.io/documentation/networkx-2.5/.

[28] Networkit 8.1, Last Accessed: 2021-06-28. https://networkit.github.io/.

[29] Numpy 1.20.3, Last Accessed: 2021-06-28. https://numpy.org/devdocs/release/1.20.3-notes.html.

[30] Scipy 1.6.3, Last Accessed: 2021-06-28. https://docs.scipy.org/doc/scipy/reference/release.1.6.3.html.

[31] Gurobi optimizer 9.1.2, Last Accessed: 2021-06-28. https://support.gurobi.com/hc/en-us/articles/360060235871-Gurobi-9-1-2-released.

[32] Sebastian Orlowski, Roland Wessäly, Michal Pióro, and Artur Tomaszewski. Sndlib 1.0 - survivable network design library. *Networks*, 55(3):276–286, 2010.

[33] Sebastian Orlowski, Roland Wessäly, Michal Pióro, and Artur Tomaszewski. Sndlib 1.0—survivable network design library. *Networks: An International Journal*, 55(3):276–286, 2010.

[34] Sndlib, Last Accessed: 2021-01-05. http://sndlib.zib.de/home.action?show=/docu.formats.gml.action.

[35] Simon Knight, Hung X. Nguyen, Nick Falkner, Rhys Alistair Bowden, and Matthew Roughan. The internet topology zoo. *IEEE J. Sel. Areas Commun.*, 29(9):1765–1775, 2011.

[36] Topology zoo, Last Accessed: 2021-01-05. http://www.topology-zoo.org/dataset.html.

[37] David LeBrun. Virtual networks testing framework (nanonet), Last Accessed: 2021-06-28. https://github.com/segment-routing/nanonet.

A DEFERRED PROOFS

Proof of Lemma 3.11. We set the weight m to each pair of links (v_i, w_j) and (w_j, v_i), $1 \leq i, j \leq m$, and the weight 1 to every other link. We assign two waypoints to each demand $(s, t, 1/j) \in \mathcal{D}$ such that its flow is routed along a link (v_i, w_j) which has the matching capacity $1/j$. Specifically, for the ith demand of size $1/j$, we set v_i, w_j as waypoints, to be visited in the same order. Under these weights, the only shortest path to v_i is $s, v_2, \ldots, v_i$ with a cost at most m, as any alternative path has a weight larger than $2m$. The shortest path between each pair of waypoints v_i and w_j is the link between them. Each node w_j receives one flow of size $1/j$ via each incoming link, which is then rerouted along the only shortest path to t of capacity D. Thus, all capacities are respected and the MLU under this weight and waypoint setting is 1. □

Proof of Lemma 3.12. We show the maximum ES-flow from s to t is 2. Let $\mathcal{N}$ denote the network in Figure 2b (Instance 2) and $\mathcal{N}'$ denote the network in Figure 2a. Consider the sub-network $\mathcal{N}_i \subset \mathcal{N}$ induced by the node set $\{v_i\} \cup \{w_j\}_j$ in $\mathcal{N}$. Recall that each link (v_i, w_j), $1 \leq j \leq m$, has the capacity $1/j \in H_m$, which is also the size of the maximum ES-flow feasible through the link. Observe that from the perspective of v_i and the source node s in $\mathcal{N}'$, networks $\mathcal{N}_i$ and $\mathcal{N}'$ are indistinguishable. Then, applying Lemma 3.10 to $\mathcal{N}_i$ implies the size of every maximum even-split (v_i, t)-flow f_i in $\mathcal{N}_i$ is $|f_i| = 1$. Then, the size of the maximum ES-flow from v_m to t is 1. However, this is not the case for $v_{i'}$, $i' < m$, because of the additional link $(v_{i'}, v_{i'+1})$.

We show that at maximum, two units of ES-flow can be delivered from each $v_{i'}$ including $v_1 = s$. From v_{m-1}, the sub-network $\mathcal{N}_{m-1}$ can deliver 1 unit of ES-flow to t. Another unit of ES-flow can be delivered via the link (v_{m-1}, v_m). Therefore, the size of the maximum even-split (v_{m-1}, t)-flow is 2, which is also the maximum feasible quantity via the link (v_{m-2}, v_{m-1}). At the node v_{m-2}, the sub-network $\mathcal{N}_{m-2}$ can deliver 1 unit in addition to 2 units via the link (v_{m-2}, v_{m-1}). Since the flow splits evenly, the size of the maximum (v_{m-2}, t)-ES-flow is 2. Repeating a similar argument for nodes $v_{m-3}, \ldots, v_1$ implies the size 2 for the maximum (s, t)-ES-flow. Hence, satisfying all demands requires an ES-flow larger than the maximum (feasible) ES-flow by the factor $(m \ln m)/2$. Applying Lemma 3.11 yields $R_{\text{LWO}} = \text{LWO}/1 = (m \ln m)/2 \in \Omega(n \log n)$. □

Proof of Lemma 3.13. We set the weight m to each pair of links (v_i, w_j) and (w_j, v_i), $1 \leq i, j \leq m$, and the weight 1 to every other link. We assign two waypoints to each demand, such that its flow is routed along the link (v_i, w_j) that has a matching capacity. Specifically, for the jth demand of size $1/(m - i + 1)$, we set nodes v_i, w_j as waypoints to be visited in the same order. Under these weights, the only shortest path to v_i is $s, v_2, \ldots, v_i$ with a cost at most m, as any alternative path has a weight larger than $2m$. The shortest path between each pair of waypoints v_i and w_j is the link between them. Each node w_j receives m flows of harmonic sizes, which is then routed along the unique shortest path to t with capacity D, i.e., $w_j, \ldots, w_m$. Thus, no link is overloaded and the MLU is 1 under this weight and waypoint settings. □

Proof of Lemma 3.14. In each of the given weight setting cases, we show that WPO cannot utilize a large part of the available capacity (unless $W \geq m$), while Joint can utilize all of this capacity using only two waypoints and an appropriate weight setting. The basic idea in all cases is as follows. Limited to $W = c \cdot n$ waypoints for a constant c. e.g. $c = 1/3$, WPO distributes the load D (i.e. total demand size) over $\approx n/3$ of nodes v_i (Figure 2b and 2c). Hence, WPO manages to distribute the load over a number of parts $W = n/3$, each part having a size $\approx D/W$, implying the claim.

Next, we give our formal proof based off the aforementioned idea. We observe the optimal MLU in WPO on the instance $\mathcal{I}_4$ (Figure 2c) and under each assumed weight setting separately.

Arbitrary Weights. For any $\epsilon < 1/2$, set the weight ϵ for links (s, w_1), (w_1, v_i) and (v_i, w_i), $1 \leq i \leq m$, and the weight 1 for every other link. Under this weight setting, the shortest path from s to every other node starts with the link (s, w_1). Therefore, in any waypoint setting, the link (s, w_1) receives the entire load D, and $\text{WPO} = D/(1/m) = D \cdot m \geq m^2 \ln m$.

Uniform Weights. Assume w.l.o.g. the weight of every link is 1. In this setting, for any node $v_i \notin \{s, v_2\}$, all links (s, w_j) are on shortest paths from s to v_i. However, by inserting waypoints, flows can be routed away from these links. The shortest path from v_i to v_{i+1} is the link (v_i, v_{i+1}). Hence, WPO may route a flow to a node $v_k, k \leq W + 1$, via the path $s, v_2, \ldots, v_k$, using $k - 1 \leq W$ waypoints: $v_2, \ldots, v_k$. From v_k, WPO may split flows optimally at this node using one additional waypoint which is a node w_j. The capacity of each link emanating from v_k is $1/(m - k + 1)$. Therefore the maximum (arbitrary-split) flow feasible from v_k to t is at most $m/(m - k + 1) < m/(m - W)$. We refer to the latter as the *available capacity* at v_k.

Hence, using up to $W = c \cdot n < m$ waypoints in an optimal waypoint setting for $c \leq 1/3$, WPO may distribute the set of flows (of total size $D = m \log m$) over nodes $v_1, \ldots, v_W$ proportionally to their available capacities in $1, \ldots, m/(m - W + 1)$, before forwarding it to t via the upper (horizontal) path. Therefore, the size of the maximum (s, t)-ES-flow via at most W waypoints (per flow) is at most $W \cdot m/(m - W + 1)$, and

$$\text{WPO} \geq \frac{m \cdot \ln m}{W \cdot m/(m - W + 1)} = \frac{(m - W + 1) \cdot \ln m}{W} \in \Omega(\frac{n \log n}{W}),$$

since $(m - W) \geq (n/2 - n/3) \in \Omega(n)$.

Inverse of Capacities. In this case, the weight of each link equals the reciprocal of its capacity. That is, the weight of every link (v_i, v_{i+1}) is $1/D$, and every other links has the weight 1. Similarly to the construction in Section 3.2, and replace the two links (s, v_2) and (v_2, v_3) with D (parallel) paths each comprising two new links with the capacity 1. As a result, the cost of the shortest path from s to $v_i, i \geq 3$ is larger than 2, which makes the path through w_j (of cost 2) a shortest path to nodes $v_i, i \geq 3$, overloading links (s, w_j). It is not difficult to see that an argument analogously as in the case of unit weights follows from here and that $R_{\text{WPO}} \in \Omega(n \log n/W)$.

Optimal Weights. We show Lemma 3.14.ii for TE-Instance $\mathcal{I}_3$. By Lemma 3.12, a maximum of 2 ES-flow units is feasible from any $v_i, i < m$ to t. We set link weights realizing such optimal ES-flow. Let $\varepsilon = 1/(2(m + 1))$. Consider the weight setting:

- the weight 2ε to the link (s, w_1),
- the weight ε to the link (v_2, w_1),
- the weight ε to links (v_i, v_{i+1}), (w_i, w_{i+1}), (w_1, v_i), $1 \leq i \leq m$,
- and the weight 1 (or larger) to every other link.

Under this setting, both of the unit-capacity paths $s, w_1, w_2, \dots, t$ and $s, v_2, w_1, w_2, \dots, t$ are shortest paths from s to t with the equal cost $(m+1)\cdot\varepsilon = 1/2$. Via these two paths, 2 units of flow can split equally at s and arrive at t without overloading any link, implying the weight setting is optimal for LWO (Lemma 3.12).

Observe that for any node $u \notin \{s, v_2, v_3\}$, the link (s, w_1) is always on a shortest path from s to u, and at least half of the flow from s to u traverses through this link. In general, for any node $u \notin \{v_i, v_{i+1}, v_{i+2}\}$, the link (v_i, w_1) is always on a shortest path from v_i to u, and at least half of a flow from v_i to u traverses through this link (the half flow is the case only for the flow to v_{i+3}).

Therefore, by setting nodes v_2 or v_3 as the first waypoint, WPO can distribute the entire load D equally on nodes $\{s, v_2, v_3\}$. Each additional waypoint increases extend of the reach two more v_i nodes. In general, using $k \le W$ waypoints $v_2, v_4, \dots, v_{2k}$ for a demand, its flow reaches the node v_{2k}, and then takes the link (v_{2k}, w_1) towards t. Hence, we can distribute the total load equally over $2k$ nodes $\{s, \dots, v_{2k}\}$, to be then delivered to t via the (upward) link connected to w_1. Thus, under an optimal waypoint setting, the aggregate flow splits into $2k+1$ equal-size subset of unit-size flows, and the load on each link (v_i, w_1) is $D/(2k+1) \ge D/(2W+1)$. Since the capacity of links connected to w_1 is 1, the maximum link utilization is at least $m\ln(m)/(2(W+1)) \in \Omega(n \log n/W)$, which concludes Lemma 3.14.ii.

Since Joint = 1 due to Lemma 3.13, in all the assumed weight settings, $R_{\text{WPO}} = \text{WPO}/1 \in \Omega(n\log n/W)$. □

Proof of Theorem 3.15. Let $\mathcal{I}_3$, $\mathcal{I}_4$ and $\mathcal{I}_5$ denote, respectively, TE-instances 3, 4 and 5. Let $\mathcal{N}_3$, $\mathcal{N}_4$ and $\mathcal{N}_5$ denote their respective network instances. Recall that $\mathcal{N}_5$ contains $\mathcal{N}_3$ and $\mathcal{N}_4$ as sub-networks. Any (s,t)-flow in $\mathcal{N}_5$ first traverses through the sub-network $\mathcal{N}_3$ and then through $\mathcal{N}_4$ before reaching its target t. Moreover, the flow can traverse only in one direction between the two sub-networks, that is, via the connecting link (t_3, s_4) (see Instance 5). Hence, the optimal weight setting for $\mathcal{I}_5$ consists of two separate optimal weight settings for the two sub-instances. Therefore, by Lemma 3.12, the gap R_{LWO} under the optimal weight setting is $R_{\text{LWO}}(\mathcal{I}_5) \ge$

$$\max\{R_{\text{LWO}}(\mathcal{I}_3), R_{\text{LWO}}(\mathcal{I}_4)\} \ge R_{\text{LWO}}(\mathcal{I}_3) \in \Omega(n\log n/W).$$

Similarly, for the gap R_{WPO} under an optimal waypoint setting, we have $R_{\text{WPO}}(\mathcal{I}_5) \ge \max\{R_{\text{WPO}}(\mathcal{I}_3), R_{\text{WPO}}(\mathcal{I}_4)\}$. If the given weight setting is arbitrary, uniform, or inverse of capacities, then by Lemma 3.14.i, $R_{\text{WPO}}(\mathcal{I}_5) \ge R_{\text{WPO}}(\mathcal{I}_4) \in \Omega(n\log n/W)$. Otherwise, the given weight setting is the optimal one from LWO, in which case Lemma 3.14.ii implies $R_{\text{WPO}}(\mathcal{I}_5) \ge R_{\text{WPO}}(\mathcal{I}_3) \in \Omega(n\log n/W)$. By Definition 3.3, the instance $\mathcal{I}_5$ implies the TE gap

$$R^* \ge \min\{R_{\text{LWO}}(\mathcal{I}_5), R_{\text{WPO}}(\mathcal{I}_5\} \in \Omega(n\log n/W). \quad □$$

Proof of Lemma 5.3. Consider all nodes sorted in the reverse topological ordering of the DAG G^* (Line 1.2). We define a succession of capacity assignments $c^1, \dots, c^n$ as follows. We let $c^n = c$ (the original capacity assignment). Under the ith capacity assignment for $2 \le i \le n$, the capacity of each link $\ell \in out(v_j)$, $1 \le j \le n$, is

$$c^i(\ell) = \begin{cases} \min_{\ell' \in out(v_j)} ec(\ell') & \text{for } j > i, \\ c(\ell) & \text{for } j \le i. \end{cases}$$

That is, if $j < i$ then the c^i capacity of every outgoing link of v_j equals the smallest e-capacity of these links. Else, $j \ge i$ and all these links have their original capacities. Let f^i be the maximum (s,t)-flow in the network (G^*, c^i). Note that f^i splits optimally at nodes $v_1, \dots, v_i$, and it splits evenly at nodes $v_{i+1}, \dots, v_n$. Hence, f^1 splits evenly at every node, which implies $|f^1| = ec(s)$. Moreover, f^n splits optimally at every node and $|f^n| = |f^*|$. Trivially, the maximum flow does not decrease when link capacities increase. Therefore, these flows satisfy

$$ec(s) = |f^1| \le \cdots \le |f^n| = |f^*|. \tag{A.1}$$

Next, we bound the increase in the maximum flow between consecutive flows f^i and f^{i-1}. Recall that every flow in $f^1, \dots, f^{i-1}$ splits evenly at v_i, and every flow in $f^i, \dots, f^n$ splits optimally at v_i. Intuitively, the increase of flow specified by $|f^i| - |f^{i-1}|$ is due to switching to the optimal splitting at v_i, i.e., due to the increase of the flow passing through v_i. For flows that split evenly at v_i, we have $f^j(v_i) \ge ec(v_i)$, $1 \le j \le i-1$, and for flows that split optimally at v_i, we have $f^j(v_i) \le \sum_{\ell \in out(v_i)} ec(\ell)$, $i \le j \le n$. Therefore,

$$|f^i| - |f^{i-1}| \le \sum_{\ell \in out(v_i)} ec(\ell) - ec(v_i) \le \lceil \ln(\Delta_i) \rceil \cdot ec(v_i).$$

If $ec(v_i) \le ec(s)$ then the increase in the maximum flow is

$$|f^i| - |f^{i-1}| \le \lceil \ln(\Delta_i) \rceil \cdot ec(s). \tag{A.2}$$

Else, $ec(v_i) > ec(s)$. Observe that in f^1, the portion of the maximum (s,t)-flow that passes through v_i, that is $f^1(v_i)$, satisfies

$$f^1(v_i) = \min\{ec_{v_i}(s), ec_t(v_i)\}. \tag{A.3}$$

That is, the maximum (s,t)-flow passing through v_i under c^i equals the smaller of the two effective capacities, one between s and v_i, and the other between v_i and t.

By definition, $f^1(v) \le |f^1| = ec_t(s)$, which extends (A.3) to

$$f^1(v_i) = \min\{ec_{v_i}(s), ec_t(v_i)\} \le ec_t(s). \tag{A.4}$$

The assumption $ec_t(v_i) > ec_t(s)$ simplifies (A.4) to

$$f^1(v_i) = ec_{v_i}(s) \le ec_t(s). \tag{A.5}$$

Since each flow $f^1, \dots, f^i$ splits evenly at every node $s, \dots, v_{i+1}$, they are constrained by the e-capacity between s and v_i, i.e., the portion of each such flow through v_i is at most $ec_{v_i}(s)$. That is, $f^j(v_i) \le ec_{v_i}(s)$ for $1 \le j \le i$. From (A.1) and (A.5), we obtain

$$ec_{v_i}(s) = f^1(v_i) \le \cdots \le f^i(v_i) \le ec_{v_i}(s),$$

which implies $f^1(v_i) = \cdots = f^i(v_i) = ec_{v_i}(s)$.

Therefore, in the case where $ec(v_i) > ec(s)$, we have

$$|f^i| - |f^{i-1}| = f^i(v_i) - f^{i-1}(v_i) = 0. \tag{A.6}$$

We sum up the increase between consecutive pairs of flows, and by applying (A.2) and (A.6), we obtain

$$\sum_{1<i\le n} (|f^i| - |f^{i-1}|) = |f^n| - |f^1| \le \sum_{1<i\le n} \lceil \ln(\Delta_i) \rceil \cdot ec(s)$$
$$\le n \cdot \lceil \ln(\Delta) \rceil \cdot ec(s), \tag{A.7}$$

which implies the claim since $|f^1| = ec(s)$ and $f^n = f^*$. □