Traffic Engineering in the presence of Tunneling and Diversity Constraints: Formulation and Lagrangean Decomposition Approach

Shekhar Srivastava, Balaji Krithikaivasan, Deep Medhi *, Michał Pióro [a]

University of Missouri–Kansas City, Kansas City, MO 64110 USA
[a]*Warsaw University of technology, Poland and Lund University, Sweden*

We present a mixed integer linear programming (MIP) formulation for a traffic engineering problem where we address the restricted number of active tunnels on each link in the presence of diversity requirements. The design has significance in networks such as multi-protocol-label-switching (MPLS)-based networks where such conditions can be applicable. For large networks, we present the solution approach using Lagrangean decomposition algorithm by considering the dual problem and using subgradient optimization. The dual subproblems are shown to have integral solutions for continuous relaxed versions. Such a construction gives us the benefit of solving the MIP problem with a series of continuous problems. We present numerical results for fairly large networks generated randomly using a topology and traffic generator. We also present computational results to show the influence of tunnel restriction and diversity parameter on traffic engineering of a network.

Key words: Heterogeneous Routers, Traffic Engineering, Label Switched Paths, Lagrangean Decomposition.

1 Introduction

Traffic Engineering is becoming an increasingly important consideration for the backbone communication networks. For example, Multi-Protocol Label Switching (MPLS) provides one of the means for achieving traffic engineering in a network [1]. MPLS framework enables exploiting of the benefits of Constraint-Based Routing and Network Controls. An important feature of MPLS is its capability to set up multiple label switched paths (LSP) between source and destinations and enable load balancing of traffic between multiple LSPs ("tunnels"). As degree of load balancing supported (number of paths activated) by a router is increased, better performance can be achieved. But such a performance benefit comes at a complexity cost.

Each LSP setup requires a label on each intermediate node which is used for switching the input traffic to the destined output port. Hence setting up each new LSP brings additional label to each intermediate node. To route each packet, a label switched router (LSR) has to search through the Label Swapping table to find the matching label and the port to get the output label and the port. It then appends the output label to the packet and sends the packet to the output port. Hence each activated LSP leads to more labels at the LSR, thereby requiring more processing to forward each packet.

* Corresponding Author: dmedhi@umkc.edu

In this paper, we present a mixed integer linear optimization formulation of a traffic engineering problem where we have captured restriction on tunnels for a router in terms of number of LSPs that can be supported on a specific link in the presence of multi-service traffic classes; further, we put restriction on demand flow on a path by introducing diversity constraints. Typically, diversity constraint is introduced to provide some level of survivability, in case one of the active LSPs is affected due to a link failure (see, for a different example, [2]). While we use MPLS and LSPs to explain the problem, the model can be applicable in other traffic engineering frameworks where the restriction on the number of tunnels is an issue.

The rest of the paper is organized as follows. In section 2, we present the description, parameters and the formulation as an MIP problem. In section 3, we present a decomposition algorithm to solve the MIP problem with a series of continuous problems. In section 4, we present results for small and large networks (experimental and randomly generated).

2 Formulation

We consider an aggregated-flow based network, where traffic data (packets) arriving to a source for a specific destination needs to be sent over one of the active LSPs between the source and the destination. Traffic data belongs to one of the service classes and hence can only be sent on the LSPs of its service class. Each service class maintains its own set of LSPs between source and destinations. The LSPs are assumed not to be shared between service classes since each service class can have its own stringent end-to-end requirement. The total LSPs chosen to be activated across the network are such that the total number of LSPs flowing through each link are restricted. The formulation maximizes the weighted carried flow across the network while honoring the active LSP constraint and Link Capacity constraint on every link. We first describe the notation:

$\mathcal{N}$:	Set of nodes in the Network	$\mathcal{K}$:	Set of nodegroups with traffic
$\mathcal{L}$:	Set of links	C_ℓ:	Capacity of link ℓ
$\mathcal{S}_k$:	Set of service classes at nodegroup k	T_ℓ:	Maximum number of tunnels allowed on link ℓ
ξ_k^s:	Unit revenue of service class s and nodegroup k	d_k^s:	Traffic demand for service class s and nodepair k
ε:	Minimum threshold on flow on a path	θ_k^s:	Maximum allowed fraction of demand flow for service class s and nodepair k

We are given the following information: $\mathcal{N}$, $\mathcal{K}$, $\mathcal{L}$, C_ℓ, T_ℓ, $\mathcal{S}_k$, ε, θ_k^s, d_k^s and ξ_k^s. We initially generate a set of candidate paths for each service class and demand pair. We assume that a path generator (such as the k-shortest path algorithm) is used to generate the set of paths, $\mathcal{P}_k^s$.

Let $|P_k^s|$ be the number of candidate paths generated for service class $s \in \mathcal{S}_k$ of demand $k \in \mathcal{K}$. We now introduce the *fractional flow variable* x_{km}^s associated with the path m for service class $s \in \mathcal{S}_k$ of request $k \in \mathcal{K}$ which takes a value between 0 and θ_k^s as the fraction of demand d_k^s allocated to m^{th} path. Here, θ_k^s restricts the maximum amount of flow that can be allocated to any one path and is referred to as diversity constraint. Diversity constraints are useful to incorporate a certain degree of fractional survivability for a service class. As discussed earlier, due to capacity limitation, it is quite possible that a demand may not be routed (while proper network design would try to avoid such situations by over-engineering; from a traffic engineering modeling standpoint, it is necessary to incorporate this variable to avoid infeasibility of the problem). To consider this aspect, we have the following constraint

$$\sum_{m \in \mathcal{P}_k^s} x_{km}^s \;\leq\; 1.0 \quad s \in \mathcal{S}_k, k \in \mathcal{K} \tag{1}$$

$$0 \leq x_{km}^s \leq \theta_k^s \quad m \in \mathcal{P}_k^s, s \in \mathcal{S}_k, k \in \mathcal{K}. \tag{2}$$

To address the fraction of flow of a service class of the demand on each link (for each path), we now introduce the indicator notation to map between the demand, the service class and the link, as they relate to paths as follows:

$$\delta_{km}^{s\ell} = \begin{cases} 1 & \text{if path } m \text{ for service class } s \text{ of nodepair } k \text{ uses link } \ell \\ 0 & \text{otherwise.} \end{cases}$$

Thus, the bandwidth needed on any link ℓ (denoted by F_ℓ) to carry flow for different service classes and demands can now be captured by the amount

$$F_\ell = \sum_{k \in \mathcal{K}} \sum_{s \in \mathcal{S}_k} \sum_{m \in \mathcal{P}_k^s} \delta_{km}^{s\ell} d_k^s x_{km}^s.$$

Since each link ℓ has capacity C_ℓ, we thus have the following constraints for each link $\ell \in \mathcal{L}$:

$$\sum_{k \in \mathcal{K}} \sum_{s \in \mathcal{S}_k} \sum_{m \in \mathcal{P}_k^s} d_k^s \delta_{km}^{s\ell} x_{km}^s \leq C_\ell \quad \ell \in \mathcal{L}. \tag{3}$$

Next, we consider the number of active LSPs sharing a link. Since we want that the number of active tunnels on any link ℓ should be less that T_ℓ. We first need to have a variable that captures if any given path is being used or not. The value take by such a variable depends on the value taken by the corresponding flow variable. Hence, we define w_{km}^s as the (binary) tunnel activity variable, which is 1 if a path is being used to route flows and 0 otherwise. Such a functionality can be achieved by incorporating following two constraints:

$$\varepsilon\, w_{km}^s \leq d_k^s x_{km}^s \quad m \in \mathcal{P}_k^s, s \in \mathcal{S}_k, k \in \mathcal{K} \tag{4}$$

$$x_{km}^s \leq w_{km}^s \quad m \in \mathcal{P}_k^s, s \in \mathcal{S}_k, k \in \mathcal{K}. \tag{5}$$

These constraints incorporate and force dependencies between variables, $\mathbf{x}$ and $\mathbf{w}$. When x is 0, constraint (4) forces w to be 0. On the other hand, when w is 0, constraint (5) forces x to be 0. Note the use of the threshold parameter, ε, to limit the generation of tunnels of very low bandwidth. We force tunnel constraint using

$$\sum_{k \in \mathcal{K}} \sum_{s \in \mathcal{S}_k} \sum_{m \in \mathcal{P}_k^s} \delta_{km}^{s\ell}\, w_{km}^s \leq T_\ell \quad \ell \in \mathcal{L} \tag{6}$$

$$w_{km}^s \in \{0, 1\} \qquad m \in \mathcal{P}_k^s, s \in \mathcal{S}_k, k \in \mathcal{K}. \tag{7}$$

The objective function maximizes the total weighted flow accepted by the network. Thus, the traffic engineering problem (**P**) can be formulated as

$$\max_{\{\mathbf{x},\mathbf{w}\}} \sum_{k\in\mathcal{K}} \sum_{s\in\mathcal{S}_k} \xi_k^s \sum_{m\in\mathcal{P}_k^s} d_k^s x_{km}^s \tag{8}$$

subject to constraints (1- 7).

Observe that the diversity constraint can play a crucial role in the throughput achieved. For smaller values of θ it forces the use of multiple tunnels and hence increases the conflict between various nodepairs. In such a scenario, restriction on active number of tunnels can result in serious performance degradation. We will discuss more on the role of diversity constraint in section 4. The problem is a Mixed Integer Linear Program with number of variables being $\sum_{k\in\mathcal{K}}\sum_{s\in\mathcal{S}_k}|\mathcal{P}_k|$ (**w**, binary) and $\sum_{k\in\mathcal{K}}\sum_{s\in\mathcal{S}_k}|\mathcal{P}_k|$ (**x**, continuous). The number of constraints required to be satisfied are $\sum_{k\in\mathcal{K}}\sum_{k\in\mathcal{S}_k} 4\ |\mathcal{P}_k|$ (Constraints (4), (5) and (2)) + $\sum_{k\in\mathcal{K}}|\mathcal{P}_k|$ (Constraint (1)) + 2 $|\mathcal{L}|$ (Constraint (3) and (6)). Since the MIP contains overwhelming number of binary and continuous variables along with high number of constraints, using direct solution methods they could only be solved for small networks.

3 Decomposition algorithm

In this section, we describe a decomposition algorithm for the problem (**P**) using Lagrangean relaxation with duality and subgradient optimization [3]. In our case, we consider the Lagrangean relaxation by observing that the constraints (4) and (5) are coupling constraints between variables x and w. Hence we take Lagrangean relaxation around these two constraints, so that the remaining constraints get decoupled from the objective function. In the process, we'll show that one of the subproblems with integrality constraints can be solvable by relaxing the integrality requirement. First we start with the Lagrangean

$$L(\mathbf{x},\mathbf{w};\mathbf{u},\mathbf{v}) = \sum_{k\in\mathcal{K}} \sum_{s\in\mathcal{S}_k} \sum_{m\in\mathcal{P}_k^s} [\xi_k^s d_k^s x_{km}^s + u_{km}^s(d_k^s x_{km}^s - \varepsilon w_{km}^s) + v_{km}^s(w_{km}^s - x_{km}^s)]. \tag{9}$$

Rearranging, we get

$$L(\mathbf{x},\mathbf{w};\mathbf{u},\mathbf{v}) = \sum_{k\in\mathcal{K}} \sum_{s\in\mathcal{S}_k} \sum_{m\in\mathcal{P}_k^s} [(\xi_k^s d_k^s + u_{km}^s d_k^s - v_{km}^s) x_{km}^s + (v_{km}^s - \varepsilon u_{km}^s) w_{km}^s]. \tag{10}$$

This can be written as $L(\mathbf{x},\mathbf{w};\mathbf{u},\mathbf{v}) = L_x(\mathbf{x};\mathbf{u},\mathbf{v}) + L_w(\mathbf{w};\mathbf{u},\mathbf{v})$. The dual problem (D) is

$$s_D = \min_{\{\mathbf{u},\mathbf{v}\geq 0\}} \quad g(\mathbf{u},\mathbf{v}) = \min_{\{\mathbf{u},\mathbf{v}\geq 0\}} g_x(\mathbf{u},\mathbf{v}) + \min_{\{\mathbf{u},\mathbf{v}\geq 0\}} g_w(\mathbf{u},\mathbf{v}). \tag{11}$$

Note that for a given $\mathbf{u}$ and $\mathbf{v}$, the Lagrangean L is separable in $\mathbf{x}$ and $\mathbf{w}$ reduces to solving two independent subproblems where

$$g_x(\mathbf{u},\mathbf{v}) = \max_{\{\mathbf{x}\}} L_x(\mathbf{x};\mathbf{u},\mathbf{v}) = \max_{\{\mathbf{x}\}} \sum_{k\in\mathcal{K}} \sum_{s\in\mathcal{S}_k} \sum_{m\in\mathcal{P}_k^s} (\xi_k^s d_k^s + u_{km}^s d_k^s - v_{km}^s) x_{km}^s \tag{12}$$

Step 1	:	Generate $\mathcal{P}_k^s$ for all $s \in \mathcal{S}_k, k \in \mathcal{K}$
Step 2	:	Solve $g_x(\mathbf{u}, \mathbf{v})$ and $g_w(\mathbf{u}, \mathbf{v})$ and derive $\mathbf{x}$ and $\mathbf{w}$
Step 3	:	Solve for s_D
Step 4	:	Check for convergence, if not converged: go to Step 3

Table-1: Algorithmic Steps

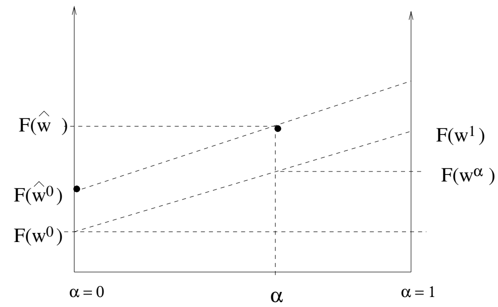

Fig. 1. Diagram with location of points

subject to constraints (1), (3) and (2) and

$$\begin{aligned} g_w(\mathbf{u}, \mathbf{v}) &= \max_{\{\mathbf{w}\}} L_w(\mathbf{w}; \mathbf{u}, \mathbf{v}) \\ &= \max_{\{\mathbf{w}\}} \sum_{k \in \mathcal{K}} \sum_{s \in \mathcal{S}_k} \sum_{m \in \mathcal{P}_k^s} (v_{km}^s - \varepsilon u_{km}^s) w_{km}^s \end{aligned} \tag{13}$$

subject to constraints (6) and (7). Thus, to solve the original problem (**P**), we use the algorithmic steps shown in Table-1.

Note that the solution to the problem described in Section 2 consists of solving three smaller problems namely, $g_x(\mathbf{u}, \mathbf{v})$, $g_w(\mathbf{u}, \mathbf{v})$ and s_D. Observe that problem $g_x(\mathbf{u}, \mathbf{v})$ is a linear continuous programming problem which can be efficiently solved using Simplex method for fairly large number of variables. Thus, we discuss below how to solve $g_w(\mathbf{u}, \mathbf{v})$.

3.1 Solving $g_w(\mathbf{u}, \mathbf{v})$

Problem $g_w(\mathbf{u}, \mathbf{v})$ is a special case of Multiple Knapsack Problem (MKP). MKP is known to be NP complete. Thus, solving the General MKP problem by direct methods imposes a severe constraint on the scalability of the solution approach. In our case, the volume of each item is equal to 1 or 0 in all the knapsacks and size of each knapsack is integral, making this special case effectively solvable by faster approaches. Below we shall show that there always exists an integral solution for the LP version of MKP. Moreover, at least one feasible optimum solution of the relaxed $g_w(\mathbf{u}, \mathbf{v})$ is integral and hence the use of the Simplex algorithm will give us the integral solutions. $RMKP$ (Relaxed MKP) can be written as:

$$\bar{F} = \max_{\{\mathbf{w}\}} F(\mathbf{w}) = \max_{\{\mathbf{w}\}} \left\{ \sum_{j=1}^{n} c_j w_j \,\middle|\, \sum_{j=1}^{n} \mathbf{A}_j w_j \leq \mathbf{b}, 0 \leq w_j \leq 1 (j = 1, 2, \ldots, n) \right\}. \tag{14}$$

where $\mathbf{A}$ is a m x n matrix with non-binary entries ($\mathbf{A}_j$ is the j^{th} column of $\mathbf{A}$ capturing the a_{ij}'s, amount of resources required by j^{th} object in i^{th} sack), $\mathbf{w}$ is an n-vector of variables (w_1, w_2,..., w_n), $\mathbf{b}$ a m-vector with nonnegative integer components. The Dynamic Programming framework [4] is used to solve the RMKP. It is sufficient to consider the family of problems

$RMKP_k(\mathbf{E})$:

$$\begin{aligned} F_k(\mathbf{b}') &= \max_{\{\mathbf{w}\}} F_k(\mathbf{b}', \mathbf{w}) \\ &= \max_{\{\mathbf{w}\}} \left\{ \sum_{j=k}^{n} c_j w_j \,\middle|\, \sum_{j=k}^{n} \mathbf{A}_j w_j \le \mathbf{b} - \mathbf{b}', 0 \le w_j \le 1 (j = k, k+1, \ldots, n) \right\}. \end{aligned} \tag{15}$$

for k varying from 1 to n, and where $\mathbf{b}'$ is a state vector of dimension m, $\mathbf{b}' = (b'_1, b'_2, \ldots, b'_m)$, every component b'_i is continuous and can take values in [0, b_i]. The recurrence relation between the values of $F_k(\mathbf{b}')$ reads:

$$F_k(\mathbf{b}') = \max_{0 \le w_k \le 1} \{c_k w_k + F_{k+1}(\mathbf{b}' + \mathbf{A}_k w_k)\} \tag{16}$$

for k decreasing from n to 1 and $\bar{F} = F_1(\mathbf{0})$.

Theorem 1 *Relaxed Multiple Knapsack Problem (RMKP) with $a_{ij} = \{0, 1\}$ and integer b has at least one binary solution in the set of all optimal solutions.*

PROOF. We prove the theorem by mathematical induction using the Dynamic Programming approach. We intend to prove that for each, $k = n, n-1, \ldots, 1$ and each integer vector $\mathbf{b}' \le \mathbf{b}$, we have an optimal binary solution of $RMKP_k(\mathbf{b}')$ with $w_j \in \{0, 1\}$ for $j = k, k+1, \ldots, n$.

The initial inductive assumption is satisfied as the solution of $RMKP_n(\mathbf{b}')$ is binary, since

$$F_n(\mathbf{b}') = \max_{0 \le w_n \le 1} c_n w_n = \begin{cases} c_n \ (w_n = 1) & \text{if } \mathbf{b} - \mathbf{b}' \ge \mathbf{A}_n \\ 0 \quad (w_n = 0) & \text{otherwise.} \end{cases} \tag{17}$$

Note that for integral $\mathbf{b}$ and $\mathbf{b}'$, $\mathbf{b}' \le \mathbf{b}$, as an object either fills totally or not at all. Hence for $k = n$, variable w_n takes either the value 0 or 1.

Now assume that for all integer vectors $\mathbf{b}'$ such that $\mathbf{b}' \le \mathbf{b}$, there exists a binary optimal solution for $RMKP_{k+1}(\mathbf{b}')$. We now need to show that $RMKP_k(\mathbf{b}')$ has a binary solution for each $\mathbf{b}' \le \mathbf{b}$. Let $(w^0_{k+1}, w^0_k, \ldots, w^0_n)$ be the optimal solution for $RMKP_{k+1}(\mathbf{b}')$ and $(w^1_{k+1}, w^1_k, \ldots, w^1_n)$ for $RMKP_{k+1}(\mathbf{b}' + \mathbf{A}_k)$, and consider the two feasible binary solutions of $RMKP_k(\mathbf{b}')$:

$$\mathbf{w}^0 = (0, w^0_{k+1}, w^0_{k+2}, \ldots, w^0_n) \quad : \quad \text{with } F(\mathbf{w}^0) = F_{k+1}(\mathbf{b}')$$

$$\mathbf{w}^1 = (1, w^1_{k+1}, w^1_{k+2}, \ldots, w^1_n) \quad : \quad \text{with } F(\mathbf{w}^1) = c_k + F_{k+1}(\mathbf{b}' + \mathbf{A}_k),$$

$w^0_i, w^1_i \in \{0, 1\}$ for $i = k+1, k+2, \ldots, n$. Let

$$\begin{aligned} \mathcal{O}_1 &= \{w_j | \ w^0_j = 0 \text{ and } w^1_j = 1, j = k+1, k+2, \ldots, n\} \\ \mathcal{O}_2 &= \{w_j | \ w^0_j = 1 \text{ and } w^1_j = 0, j = k+1, k+2, \ldots, n\} \\ \mathcal{O}_3 &= \{w_j | \ w^0_j = w^1_j, j = k+1, k+2, \ldots, n\}. \end{aligned}$$

Consider a set of feasible solutions of $RMKP_k(\mathbf{b}')$ defined as $\mathbf{w}^\alpha = \mathbf{w}^0 + \alpha(\mathbf{w}^1 - \mathbf{w}^0)$. The solution $\mathbf{w}^\alpha$ consists of fractional values for some of its components and can be written in the following form:

$$w_j^\alpha = w_j^0 + \alpha \;\; j \in \mathcal{O}_1, \qquad w_j^\alpha = w_j^0 - \alpha \;\; j \in \mathcal{O}_2 \text{ and } \qquad w_j^\alpha = w_j^0 \;\; j \in \mathcal{O}_3. \tag{18}$$

Observe that $\mathbf{w}^\alpha$ is a feasible point since by construction $\mathbf{w}^\alpha$, is a convex combination of $\mathbf{w}^0$ and $\mathbf{w}^1$, since $\mathbf{w}^\alpha = (1-\alpha)\mathbf{w}^0 + \alpha\mathbf{w}^1$.
We define $F(\mathbf{w}^\alpha) = c_k\alpha + F_{k+1}(\mathbf{b}' + \mathbf{A}_k\alpha)$ which is the revenue when the knapsack contains exactly a fraction α of object k and some amounts of objects $\{k+1, k+2, \ldots, n\}$ (fractional are possible) which are required to be chosen to fill optimally a knapsack smaller by $\mathbf{A}_k\alpha$ in volume.
We show that the problem $RMKP_k(\alpha\mathbf{A}_k)$ for all $0 \le \alpha \le 1$, the optimal revenue is $F(\mathbf{w}^\alpha)$= $F_{k+1}(\mathbf{b}') + \sum_{j=k}^n c_j\alpha\,(\mathbf{w}^1 - \mathbf{w}^0)$, and thus, $\mathbf{w}^\alpha$ is the optimal and binary solution of $F_k(\mathbf{b}')$.
We show this by contradiction. Suppose that there exists $\hat{\mathbf{w}} = (\alpha, \hat{w}_{k+1}, \hat{w}_{k+2}, \ldots, \hat{w}_n)$ such that $F(\hat{\mathbf{w}}) = F(\mathbf{w}^\alpha) + \gamma$ and $\gamma > 0$. Consider a feasible solution $\hat{\mathbf{w}}^0$ = $\hat{\mathbf{w}}$ - $(\mathbf{w}^\alpha - \mathbf{w}^0)$ of $RMKP_k(0)$ (cf. Figure 1). Because the considered problem is linear, we have

$$F(\hat{\mathbf{w}}^0) = F(\hat{\mathbf{w}}) - (F(\mathbf{w}^\alpha) - F(\mathbf{w}^0)) \tag{19}$$

and hence we have $F(\hat{\mathbf{w}}^0) = F(\mathbf{w}^0) + \gamma$. This is a contradiction, since we have assumed that $F(\mathbf{w}^0)$ is the optimal solution among the all possible values of $w_{k+1}^0, w_{k+2}^0, \ldots, w_n^0$. Hence the values of $F(\mathbf{w}^\alpha)$, $0 < \alpha < 1$ lie on the line segment joining points $F(\mathbf{w}^0)$ and $F(\mathbf{w}^1)$. Due to optimality of all $F(\mathbf{w}^\alpha)$, the solution set always contains either $\mathbf{w}^1$ $(F_k(\mathbf{E}) = c_k + F_{k+1}(\mathbf{E} + \mathbf{A}_k))$ or $\mathbf{w}^0$ $(F_k(\mathbf{E}) = F_{k+1}(\mathbf{E}))$. ■

Remark 2 An intuitive way to see the solution is to consider each constraint $\sum_{j=1}^n a_{ij}w_j \le b_i$ for $i = 1, \ldots, m$ as a hyperplane being placed in the n dimensional space. Since all $a_{ij} \in \{0,1\}$, these hyperplanes intersect with the axes and with each other at points of type $w = (w_j, j = 1, \ldots, n)$ only, where $w_j \in \{0,1\}$. Hence all the extreme points (points of intersection of n planes) have binary values. And for the same reason, we can effectively find solutions using simplex approach which checks the extreme points only.

3.2 Solving the master dual: s_D

Observe that the problem s_D is an unconstrained optimization problem with variables $\mathbf{u}$ and $\mathbf{v}$. The function to be minimized is nonsmooth, we use subgradient approach [3] to solve the dual problem s_D. This method iterates on the dual variables $\mathbf{u}$ and $\mathbf{v}$. Thus given the value of $\mathbf{u}$ and $\mathbf{v}$, once the solutions to the subproblems $g_x(\mathbf{u},\mathbf{v})$ and $g_w(\mathbf{u},\mathbf{v})$ are obtained, a dual subgradient, $\boldsymbol{\pi}_{(u)} = (\pi_{km}^{s(u)})$ and $\boldsymbol{\pi}_{(v)} = (\pi_{km}^{s(v)})$, for $g(.)$ can be computed using subgradient methods.

$$\begin{aligned} \pi_{km}^{s(u)} &= (x_{km}^s d_k^s - \varepsilon w_{km}^s) \qquad && m \in \mathcal{P}_k^s, s \in \mathcal{S}_k, k \in \mathcal{K} \\ \pi_{km}^{s(v)} &= (w_{km}^s - x_{km}^s) \qquad && m \in \mathcal{P}_k^s, s \in \mathcal{S}_k, k \in \mathcal{K}. \end{aligned} \tag{20}$$

Then dual multipliers, $\mathbf{u}$ and $\mathbf{v}$ are updated using

$$u_{km}^s = \max\{0, u_{km}^s - \lambda_u\pi_{km}^{s(u)}\}, \quad v_{km}^s = \max\{0, v_{km}^s - \lambda_v\pi_{km}^{s(v)}\}. \tag{21}$$

We have used the subgradient method with relaxation to minimize the non-smooth dual function. Hence the sizes, λ_u and λ_v, are given by

$$\lambda_u = \rho \frac{g(\mathbf{u},\mathbf{v}) - g^{\#}}{||\boldsymbol{\pi}_{(u)}||^2}, \qquad \lambda_v = \rho \frac{g(\mathbf{u},\mathbf{v}) - g^{\#}}{||\boldsymbol{\pi}_{(v)}||^2} \tag{22}$$

where, $g^{\#}$ is the relaxed primal value. We take $\rho = 2.0$ and half it, if the solution value does not change for consecutive 40 iterations. We put the maximum iteration bound as 1000 and if reached accept the maximum revenue solution among the already encountered ones as the optimal solution.

4 Results and Discussion

We have implemented our method in C^{++} using CPLEX callable libraries to solve subproblems. The aim of this section is three-fold: (a) to show the convergence behavior depending on whether tunneling or capacity constraint is dominant, (b) to show the interaction between tunneling and diversity constraints, (c) to demonstrate the effectiveness of the decomposition algorithm.

We start with the first aim. Consider experimental networks shown in Figures $2-5$. For experimental networks EN-I, EN-II and EN-III, capacities of links are given in Figures $2-5$ and demands are presented in Tables $1-3$. For EN-IV, we consider capacity of 10×622 Mbps for each link and assume demands between all pairs of nodes of value 1000 Mbits and three service classes are considered here.

In order to understand the convergence behavior based on the domination of tunneling or capacity constraint, we construct two scenarios. For scenario I, we increase the capacity of each link to 10 times it present value (10×622) Mbps and allowed nuber of tunnels are maintained at their present value (5) and is referred to as 'Tunnel Constrained Scenario'(TCS). In scenario II, we make allowed number of tunnels on each link to be 5 times its present value (25) and hence is referred to as 'Capacity Constrained Scenario'(CCS).

We present results regarding the convergence properties for the experimental networks EN-I and EN-IV in plots presented in figures 6 to 9. Results for EN-II and EN-III were similar to the ones presented for EN-I and aren't shown here. The idea is to observe the convergence characteristics for these two cases. Observe that the convergence of the algorithm is good for small as well as large networks. For experiments in TCS, the current best solution has much lower value than the value of the relaxed problem. Such a behavior can be explained as follows: the relaxation is taken around the tunnel constraints and thus, forcing them to be honored which leads to much lower objective value. As for the experiments in CCS, the current best solution closely follows the relaxed primal value. The relaxed constraints are nearly obviated (allowed number of tunnels are fairly large). We can also observe that the objective value does not decrease consistently at each step of the iteration since the algorithm does not guarantee descent and therefore we needed to store the best possible solution over all iterations. With thee help of plots we can conclude that the convergence of the algorithm is good and that fairly large sized networks can be effectively solved using the decomposition algorithm.

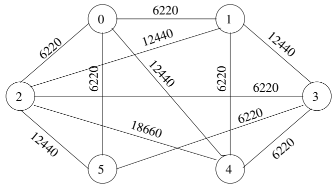

Fig. 2. EN I

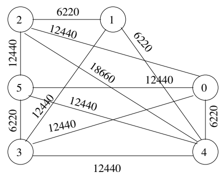

Fig. 3. EN II

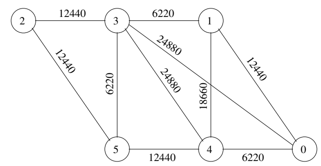

Fig. 4. EN III

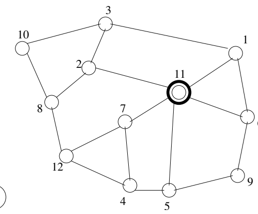

Fig. 5. EN IV

k	(S, D)	d_k^1	d_k^2	d_k^3
1	(0, 2)	2550	20224	2000
2	(0, 4)	5000	13755	2100
3	(0, 5)	9130	3105	1200
4	(2, 3)	7000	3654	6400
5	(2, 4)	8680	67689	4900
6	(2, 5)	52500	1204	1800
7	(3, 4)	7500	7590	500
8	(3, 5)	81900	2226	1200

Table 2.**Demand for EN-I**

k	(S, D)	d_k^1	d_k^2	d_k^3
1	(0, 1)	3750	24864	2000
2	(0, 3)	9430	38868	2400
3	(0, 4)	6080	6943	1500
4	(1, 3)	6000	3024	1800
5	(1, 4)	13600	12388	1000
6	(3, 4)	5580	34584	1800
7	(3, 5)	903	2618	1500
8	(4, 5)	2176	6422	2000

Table 3.**Demand for EN-II**

k	(S, D)	d_k^1	d_k^2	d_k^3
1	(0, 2)	14790	10044	2500
2	(0, 4)	6080	7614	1200
3	(0, 5)	220	2958	4800
4	(1, 2)	12690	1000	3500
5	(1, 4)	8000	12644	7200
6	(2, 4)	13950	41607	618
7	(2, 5)	2720	16405	2721
8	(4, 5)	5100	24864	300

Table 4.**Demand for EN-III**

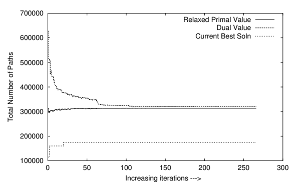

Fig. 6. EN-I: TCS

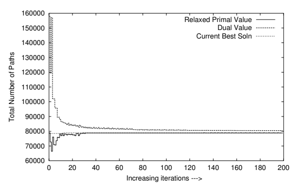

Fig. 7. EN-I: CCS

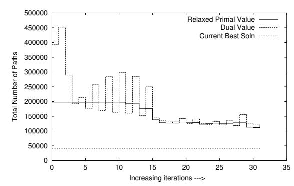

Fig. 8. EN-IV: TCS

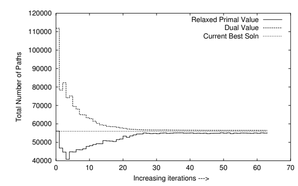

Fig. 9. EN-IV: CCS

No.	3-Tuple	Rel. Primal Value	Dual Value	Current Best Soln.	Iterations
1	(10,20,40)	42637.2	45340.2	19686.5	83
2	(30,90,120)	128522	133365	50752.5	85
3	(50,150,200)	167646	174177	53405.5	168

Table 5.**Results for Random Topologies**

We next present results demonstrating the interplay between the tunneling constraint and the diversity constraint. The capacities taken were the ones considered for tunnel constrained scenario. In figures 10, 11, 12 and 13, we can observe that allowed number of tunnels indeed impact the throughput of the system. For smaller networks the interplay is relatively less and we can reach the maximum throughput for relatively small number of tunnels (20−40) with a moderate value of θ(0.25). However, for large networks with every nodepair generating traffic of *multiple* classes (3 in our case), maintaining a decent value of θ might require a large number of tunnels to reach its maximum value.

For the third aim, we have generated various random topologies and class based traffic with the topology generator used in [5]. We represent the random topologies through a 3-tuple (n_n, n_l, n_d) where they stand for the number of nodes, number of links, total number of demand pairs, respectively (Table 5). The number of active tunnels are sampled as *uniform*($\frac{2\, n_d\ n_s}{n_l}$,

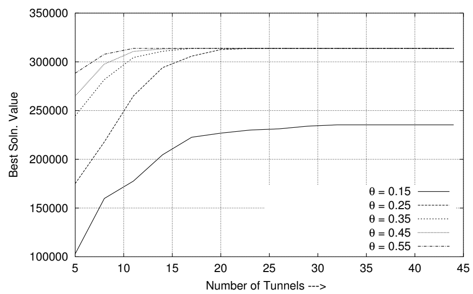

Fig. 10. EN-I:Tunnel *vs* Diversity Constraint

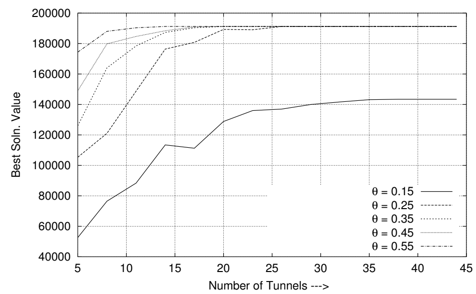

Fig. 11. EN-II:Tunnel *vs* Diversity Constraint

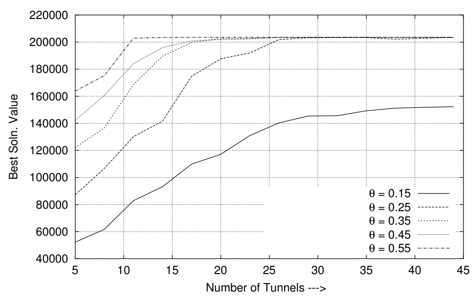

Fig. 12. EN-III:Tunnel *vs* Diversity Constraint

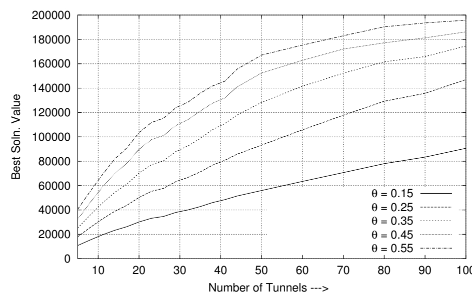

Fig. 13. EN-IV:Tunnel *vs* Diversity Constraint

$\frac{4\,n_d\;\;n_s}{n_l}$) for each link. We have set the diversity parameter at $\theta = 0.25$. We found the method to converge in a reasonable amount of iterations, and for the relaxed primal problem, the duality gap to be very minimal. Finally note that, all investigations conducted here assumed that ξ_k^s is 1.0 for all service classes and demands. By having different values of ξ_k^s, either based on service class or demand (or both), can alter results and convergence properties to a fairly high degree.

To summarize, we have presented a mixed integer linear optimization formulation for a traffic engineering problem where we address the restricted number of active tunnels on each link in the presence of diversity requirements. We have also developed an effective dual-based approach to solve the formulated problem. Through computational studies, we have observed that the increase in throughput is not linear with the increase in the maximum number of tunnels allowed (per link); in fact, we encounter diminishing returns. For over-provisioned networks, we observed that the tunnel constraint indeed becomes an issue and leads to low throughput. We also noted that when the diversity factor is increased beyond 0.5, impact on the obtained throughput is minimal for observed tunnel constraints (T_ℓ). Moreover, for large networks with reasonable diversity parameter (0.25), number of tunnels required to reach maximum throughput could become considerably high. Hence compromise needs to be made in choosing the diversity parameter and the active number of tunnels.

References

[1] B. Davie, Y. Rekhter, *MPLS: Technology and Applications*, Morgan Kaufmann Publishers, 2000.

[2] D. Medhi *A Unified Approach to Network Survivability for Teletraffic Networks: Models, Algorithms and Analysis*, IEEE Transaction on Communication, Vol 42, pp. 534-548, 1994.

[3] M. Held, P. Wolfe and H. Crowder *Validation of Sub-Gradient Optimization* Mathematical Programming, Vol 6, pp 62-88, 1974.

[4] M. Minoux *Mathematical Programming - Theory and Algorithms*, J. Wiley & Sons, 1986.

[5] B. Cotter *Design of Multi Layered Survivable Networks*, Doctoral Dissertation, School of Interdisciplinary Computing and Engineering, University of Missouri-Kansas City.