How Well do Traffic Engineering Objective Functions Meet TE Requirements?

Simon Balon*, Fabian Skivée, and Guy Leduc

Research Unit in Networking,
EECS Department- University of Liège,
Institut Montefiore, B28 - B-4000 Liège - Belgium
{balon, skivee, leduc}@run.montefiore.ulg.ac.be

Abstract. We compare and evaluate how well-known and novel network-wide objective functions for Traffic Engineering (TE) algorithms fulfil TE requirements. To compare the objective functions we model the TE problem as a linear program and solve it to optimality, thus finding for each objective function the best possible target of any heuristic TE algorithm. We show that all the objective functions are not equivalent and some are far better than others. Considering the preferences a network operator may have, we show which objective functions are adequate or not.

1 Introduction

We consider the traffic engineering routing problem. Given the topology of the network to be engineered and an estimate of the traffic matrix to be routed on it, the problem is to find a routing scheme that optimises the network, with the joint goal of good user performance and efficient use of network resources. The way classical algorithms fulfil this objective is not clear. Indeed, many algorithms try to optimise their home-made objective functions which are said (but not proven) to reflect traffic engineering objectives. The foundations of all these objective functions are related, but could lead to quite different results, as we see in our simulations.

Some in-depth reflection and comparison studies of traffic engineering objective functions are needed. In many research papers, the quality of a new traffic engineering algorithm is evaluated regarding one specific objective function. If the algorithm obtains a good score, it is considered as good. But this is only meaningful if the objective function really reflects the traffic engineering goals. Furthermore, when analysing published papers it is difficult to figure out if the merits of a given TE method is due to its objective function or its heuristic algorithm. To fill this gap, we provide an independent comparison of many objective functions found in the literature.

To compare all the different objective functions, we will minimise (or maximise) each of these functions on the same topology and traffic matrix and analyse

---

if the routing scheme we obtain really reflects general Traffic Engineering goals. One important point is that we have used some real topologies and real traffic matrices to run our tests, which is not the case of many research papers. The use of real data provides a real case study and an objective basis for comparison.

Section 2 presents Traffic Engineering goals and requirements. Section 3 introduces existing TE algorithms and related objective functions. We discuss the foundation of these objective functions and why they were introduced. In section 4, we construct LP (Linear Programming) models of these objective functions. These models are used to compare all the presented functions on different networks. Then we analyse the results of simulations, highlighting the merits and/or shortcomings of each objective function. Finally, section 5 concludes the paper.

2   Traffic Engineering Objectives

A network is modeled as a directed graph, $G = (N, A)$ whose nodes and arcs represent routers and links. Each arc has a capacity c_a. Traffic on the network is represented by a traffic matrix D that with every pair (s, t) of nodes associates the value of the traffic demand, i.e. the traffic that flows from node s to node t.

Basically, the graph G and the traffic matrix D are the inputs of the problem. A traffic engineering algorithm has to find *good* paths between each pair of source and destination nodes to route corresponding traffic flow. The definition of *good* paths is related to what we want to optimise on the network. Generally, a good set of paths will be one that optimises a pre-defined objective function.

Once the paths are chosen, we can associate with each arc a load l_a, which is the total load on the arc, i.e. the sum over all demands of the amount of traffic sent over a. The utilisation of a link a is $u_a = l_a/c_a$. The available bandwidth on link a is $ABW_a = c_a - l_a$.

Finally, we define θ_{st} as the maximum flow that can be sent from node s to node t in the residual network, i.e. when the whole traffic matrix is routed on the network.

2.1   Discussion on TE Objectives

Typically, on-line algorithms have different objectives than off-line ones. On-line schemes usually try to minimise the probability of blocking future requests, while off-line ones try to minimise the load or the utilisation of the links, or try to maximise available bandwidth. To some extent, minimising the link utilisation (which is a relative measure) tends to maximise the available bandwidth (which is an absolute measure) on the links, thus also reducing the blocking probability of future requests. Clearly, these objectives are closely related, but no solid basis exists to choose one among all.

We will consider TE metrics at three different levels, which are a link, an OD pair[1] and the network. We will present and justify the foundation of the TE metrics at each level. We will differentiate metrics whose goal is to improve the

---

[1] OD stands for Origin Destination.

quality of the network given the present traffic (e.g. minimise the delay) from metrics whose goal is to maximise the acceptance of future traffic on the residual network (e.g. maximise residual max-flow).

At the *link level*, we should minimise delay and utilisation. We should also maximise the available bandwidth on this link (which corresponds to the notion of residual max-flow for a link). The delay of a link is composed of three components: the propagation delay which is a constant value, the transmission delay (inversely proportional to the link capacity) and the queueing delay which increases with the link load. If we take the delay to be the average delay of an M/M/1 queue, the mean queueing + transmission delay of link a is given by $Delay_a = \frac{mean\_packet\_size}{c_a - l_a}$. For a M/M/1 queue, all the percentiles/quantiles are also proportional to this value. On high capacity links, this delay is significant only if the link load is approaching the link capacity. Figure 1 summarizes the relations between link parameters.
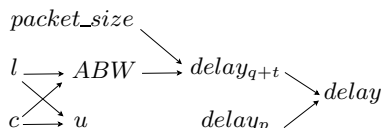

Fig. 1. Link parameters

At the level of an *OD pair* of nodes, we should minimise the path delay, i.e. the sum of the delays of all the links on the path. Minimising this delay can increase the quality of service perceived by the users of the network. We should also minimise the maximal link utilisation on the corresponding path. Indeed, the maximal link utilisation has a particular meaning. For example a maximal link utilisation (u_{max}) of 50% means that we can double all the traffic before having a link fully loaded (if we keep the same routing scheme), while a value of 20% means that we can multiply all the traffic by 5. In fact this value ($\frac{1}{u_{max}}$) is a lower bound because a change in the routing scheme may allow increasing this value. Finally, the residual max-flow between an OD pair of nodes should be maximised. Indeed, this value represents the maximal size of a future request that can be routed on the network between these nodes.

We have now some ideas of TE metrics to be optimised for a link or for an OD pair. But to be really useful in TE algorithms we have to generalise these concepts to the *whole network*. There are many ways to proceed. For example, considering link utilisations, one can minimise the maximal link utilisation (u_{max}) as for the OD level[2]. But the minimisation of the maximal link utilisation works poorly in some cases. Indeed if there is a real bottleneck in the network, i.e. a link whose utilisation cannot be decreased by changing the routing scheme, it is important

---

[2] We can prove that the routing scheme that achieves the minimal value of maximum link utilisation also provides the optimal value concerning the factor by which it is possible to multiply the current traffic matrix. In this case, this factor can be computed as $\frac{1}{u_{max}}$.

to minimise also the utilisation of other links. One way to proceed can be to minimise the mean link utilisation. Considering the delay to be minimised on the whole network, we can compute the mean link delay (each link being weighted by its load or not) or the mean path delay (each path being weighted by its corresponding traffic). The unweighted mean path delay seems less relevant to us. The following demonstration shows that the weighted mean path delay is equivalent to the weighted mean link delay. This demonstration highlights that it is possible to compute the mean path delay without path information. This is an important result from a computational point of view. Indeed it is less complex to compute a sum over all links than a sum over all paths of all possible OD pairs of nodes.

$$
\begin{aligned}
MeanDelay &= \tfrac{1}{\sum_{(s,t)} D(s,t)} \sum_{(s,t)} D(s,t) \sum_{a \in \mathcal{P}(s,t)} delay_a \\
&= \tfrac{1}{AllTr} \sum_{(s,t)} D(s,t) \sum_{a \in \mathcal{P}(s,t)} delay_a \\
&= \tfrac{1}{AllTr} \sum_{(s,t)} D(s,t) \sum_{a \in A} \delta_{a \in \mathcal{P}(s,t)} delay_a \\
&= \tfrac{1}{AllTr} \sum_{a \in A} delay_a (\sum_{(s,t)} D(s,t) \delta_{a \in \mathcal{P}(s,t)}) \\
&= \tfrac{1}{AllTr} \sum_{a \in A} l_a \times delay_a
\end{aligned}
$$

$\mathcal{P}(s,t)$ denotes the path from s to t[3], $\delta_{a \in \mathcal{P}(s,t)}$ is equal to one if link a belongs to $\mathcal{P}(s,t)$ and 0 otherwise. $AllTr$ denotes the sum of all traffics of the network ($\sum_{(s,t)} D(s,t)$). It is a constant for a given problem.

Considering max-flows (θ), it is possible to maximise the minimal residual max-flow. But as for the maximal link utilisation, the minimal max-flow can be blocked by a set of bottleneck links. So we should also maximise the sum of all max flows (instead of the min value). We could also associate with each max-flow a weight related to its corresponding traffic demand.

As it could be interesting to maximise the sum of residual max-flows, it could be interesting to maximise the sum of the available bandwidths over all the links of the network. However, we can notice that it is equivalent to minimising the sum of the loads over all the links of the network. Indeed, $max \sum_{a \in A} ABW_a = max \sum_{a \in A} (c_a - l_a) = max(\sum_{a \in A} c_a - \sum_{a \in A} l_a) \equiv min \sum_{a \in A} l_a$, as the sum of all the capacities of the network is invariant.

Table 1 presents a summary of TE metrics introduced in this section.

Table 1. TE metrics summary

	Metric characterising good current state	Metric characterising likely good future
Link$_{(a)}$	$Delay_a$	u_a, ABW_a
Path$_{(s,t)}$	$\sum_{a \in \mathcal{P}(s,t)} Delay_a$	θ_{st}, $max_{a \in \mathcal{P}(s,t)} u_a$
Network	$\frac{\sum_{a \in A} Delay_a}{|A|}$, $\frac{\sum_{a \in A} l_a \times Delay_a}{AllTr}$	$min_{(s,t)} \theta_{st}$, $max_{a \in A} u_a$ $\sum_{(s,t)} \theta_{st}$

---

[3] Here, we assume that there is only one path used from s to t, but the demonstration can be easily generalised if there are multiple paths.

2.2   How to Measure the Quality of a Solution?

In this section we present the TE metrics we will use to evaluate the quality of the routing solutions in the simulation section. As presented in the preceding subsection, it is clear that the maximum link utilisation (u_{max}) is a good TE metric. In addition, the mean link utilisation (u_{mean}), the 10th percentile (u_{per10}), the minimal available bandwidth (ABW_{min}) and the mean load (l_{mean}) will be used. u_{per10} is defined so that 10% of the links have a utilisation over u_{per10}. We think that the weighted mean queueing + transmission delay of the network ($delay_{mean} = \frac{1}{AllTraffic} \sum_{a \in A} l_a \times \frac{packet\_size}{c_a - l_a}$) is also an important variable. We will also consider the minimum max flow ($\theta_{min} = Min_{(s,t)}\theta_{st}$) and total max flow ($\theta_{tot} = \sum_{(s,t)} \theta_{st}$) of the residual topology. The total max flow gives an idea of the throughput, i.e. which amount of traffic can be accepted on the residual network. This is not exactly the amount of bandwidth that can be routed on the residual network because all max-flows are computed independently of each other and thus all the flows are not in competition for the residual bandwidth. But this can still give a good idea of the residual throughput[4].

3   Presentation of Different Objective Functions

3.1   Fortz

In [1], B. Fortz et al. try to find an optimal set of IGP weights such that classical shortest path first algorithms taking these modified metrics in consideration lead to a good routing scheme. A cost is associated with each link of the network. This cost (ϕ_a) is a convex piecewise linear function of the link load. The objective function they try to minimise is the sum over all links of this cost ($\phi = \sum_{a \in A} \phi_a$). We will later refer to this objective function as *Fortz*. We have noticed that this function, though empirical, could be seen as a linear approximation of $\frac{l_a}{1-u_a}$. At low link utilisation, $1 - u_a \approx 1$ and $Fortz \approx min \sum_{a \in A} l_a$, while at high utilisations, $\frac{1}{1-u_a}$ becomes significant, leading to a load balancing policy (avoiding links with high utilisation). There is no OD pair consideration in this objective function. Many papers have reused this objective function.

3.2   MIRA

In [2], Kodialam et al. introduce the concept of minimum interference routing. They propose an objective function which is a weighted sum of the maxflows over all possible source-destination pairs on the residual topology. Their online algorithm, called MIRA, is a heuristic that tries to maximise this objective function. Formally, the objective function to be maximised is $\sum_{(s,t)} \alpha_{st}\theta_{st}$, where

---

[4] The amount of traffic that can be routed on the residual network is in fact the sum over all links of the available bandwidth. Indeed one obvious (and degenerated) solution to the max throughput problem is to associate traffic only with the pairs of nodes that are located at the extremities of a link. We can associate with these pairs the available bandwidth on the corresponding link.

α_{st} is a weight associated with the ingress-egress pair (s, t). The weights associated with ingress-egress pairs are administrative weights that determine the relative importance of the ingress-egress pairs to the network administrator. Behind this objective function, the goal is to minimise the blocking probability of a future new request, without information about it. The idea is that if the maxflow between one source and one destination decreases, this means that the maximum request that can be accepted between these two nodes decreases as well. Thus, the $MIRA$ objective function is characterising likely good future. There is no embedded metric characterising good current state. We will see later in the simulations the implications of this fact.

3.3   Blanchy

In [3], Blanchy et al. present an online heuristic traffic engineering algorithm to optimise a load balancing objective function. The pure load balancing objective function is $\sum_{a \in A}(u_a - u_{mean})^2$ with $u_{mean} = \frac{1}{|A|} \sum_{a \in A} u_a$, the mean link utilisation in the network. *This function is the variance on the link utilisation and, as such, represents the deviation from the optimal load balancing situation.* To limit the length of the paths of a pure load balancing function, they add a "shortest path" term and arrive at the following objective function: $\sum_{a \in A}(u_a - u_{mean})^2 + \alpha \sum_{a \in A}(u_a)^2$. It is interesting because *the (weighted) combination of both terms will give more importance to the load-balancing term if the deviation is high enough to justify the detour, else it will let the "shortest path" term minimise the resources used. The weighted factor α allows to give more importance to one aspect or to the other.* This objective function does not directly include TE metrics we introduced in section 2.1. It does not include a delay contribution and there is no consideration about OD pairs. The traffic minimisation term tries to minimise the size of the paths.

3.4   Delay

In [4], Elwalid et al. associate a cost with each link. They try to minimise the total cost which is the sum over all links of the link cost. The cost of a link is a function of the link load. They assume that this function is convex. They say that a natural choice for the link cost is the delay so that their network-wide cost function is defined as $MeanDelay = \sum_{a \in A} \frac{1}{c_a - l_a}$. In section 2.1 we called this function the (unweighted) mean link delay, if we do not take the propagation delay into account. We introduce a new delay objective function (referred to as $WMeanDelay$) which is $\sum_{a \in A} \frac{l_a}{c_a - l_a}$, the weighted mean delay. Note that this objective function can also be formulated using only u_a as $WMeanDelay = \sum_{a \in A} \frac{u_a}{1 - u_a}$. These objective functions are metrics characterising good current state.

3.5   Degrande

In [5], Degrande et al. propose to maximise an objective function which is the sum of four terms: F(airness), T(hroughput), B(alance) and (network) U(tilisation).

A coefficient (named C_F, C_T, C_B or C_U) is associated to each term to give more influence to one or another. Fairness and Throughput are traffic oriented objectives while Balance and Utilisation are resource oriented objectives. Balance is defined as: $B = 1 - u_{max}$. Network utilisation is defined as $U = \sum_{a \in A} u_a$. We will not consider Fairness and Throughput in our formulation because it is not possible to express these in our LP formulation. The balance is a metric characterising likely good future. The utilisation term will minimise the size of the paths. There is no OD pair consideration and no delay contribution in this objective function. Some papers only try to minimise the maximum link utilisation. This is equivalent to $Degrande$ objective function where $C_B = 1$ and $C_U = 0$. We will refer to this objective function as u_{max}.

$Degrande$ objective function where $C_B = 0$ and $C_U = 1$ is a function which minimise $U = \sum_{a \in A} u_a$. This objective is also minimised by a classical SPF routing considering link weights equal to the inverse of their capacities. In fact, inverse capacity routing (recommended by CISCO) gives the optimal value of U. We will thus refer to this objective function as $InvCap$. We prove this by contradiction. If it is not the case, this means that there exists one flow for which the $InvCap$ path does not minimise its contribution to $\sum_{a \in A} u_a$. But its contribution to this sum is in fact the traffic on this flow multiplied by the sum of the inverse of the capacity of all the links of the path, which is minimised by $InvCap$ SPF.

3.6   Summary

Clearly, all the presented objective functions are related, while quite different. A first difference is that some of them use only absolute values of the load l (like $MIRA$), some only relative values u (like $Blanchy$, $WMeanDelay$, or $Degrande$) and finally some use both (like $Fortz$, or $MeanDelay$).

Table 2 presents a summary of all the presented objective functions. $MIRA$'s function is used with $\alpha_{st} = 1, \forall (s,t)$. For $Blanchy$, we have to fix the α parameter. For $Degrande$, we have to fix C'_B and C'_U. In the table, we have added the cost function called $MinHop$. This function simply minimises the total load over all

Table 2. Summary of objective functions

	Score Function (to be minimised)
Fortz	$\sum_{a \in A} \phi_a$
MIRA	$-\sum_{(s,t)} \theta_{st}$
Blanchy	$\sum_{a \in A} (u_a - u_{mean})^2 + \alpha \sum_{a \in A} (u_a)^2$
MeanDelay	$\sum_{a \in A} \frac{1}{c_a - l_a}$
WMeanDelay	$\sum_{a \in A} \frac{l_a}{c_a - l_a}$
InvCap	$\sum_{a \in A} u_a$
u_{max}	u_{max}
Degrande	$C_B . u_{max} + C_U . \sum_{a \in A} u_a$
MinHop	$\sum_{a \in A} l_a$

the links of the networks ($\sum_{a \in A} l_a$). Following the same development as for $InvCap$, this function is minimised by a SPF routing considering a weight of 1 for each link (what we call a min hop routing).

We can point out that at low load, $1 - u \approx 1$ and $c - l \approx c$ and thus $Fortz \approx MinHop$ while $WMeanDelay \approx InvCap$.

4   Simulations

In order to compare all the objective functions, we will model the traffic engineering routing problem as a linear program (LP) and solve it to optimality for all the presented objective functions. In this formulation, all the flows can be arbitrarily split. Obviously, this cannot be really implemented in a network, but can be approached with MPLS routing and to some extent with ECMP. This assumption allows us to formulate the problem as a linear program (which is easy to solve to optimality) instead of a mixed integer program (which cannot be solved to optimality in a reasonable time). The LP formulation will be used to solve the routing problem to optimality and compare the solutions obtained for every objective function. We will not write the formulation of all the objective functions, because this would take too much space, but we explain clearly how they can be reproduced. We have used an LP node-link formulation (as in [1]). $Fortz$ is expressed in [1]. For $MIRA$, we use a classical $max\ flow$ formulation for each pair of nodes. For $Blanchy$, the square function is approximated by its linear approximation in the range $[-1, 1]$. $MeanDelay$ and $WMeanDelay$ are approximated by convex piecewise linear functions. $Degrande$ and $MinHop$ are linear so they can be expressed easily, without modification. We will not present $MeanDelay$ in our result tables because we have noticed similar results than $WMeanDelay$ (noted $Delay$ in the tables). For $Degrande$ function, we use $C'_B = 10^3$ and $C'_U = 1$ (as in [5]) and for $Blanchy$, $\alpha = 3$ (which seems to provide good results).

4.1   Simulation Description

We made our simulations on three different networks. The first topology was generated in the TOTEM toolbox [6] using Waxman's method [7]. This topology is composed of 25 nodes and 50 full-duplex links. We set the value for parameters α and β to 0.15 and 0.2. We have generated a random traffic matrix for this topology. The second topology is an operational network. This operational network is composed of about 20 routers and 40 bidirectional links. To build a realistic traffic matrix, we have collected netflow data on each interface of the network and aggregated this information to build a traffic matrix (the procedure to generate traffic matrices from netflow traces is described in [8]). The last topology is the US research network (Abilene). It is composed of 11 nodes and 14 bidirectionnal links of 10 Gbps each. As for the operational network, we have used netflow data measured on the network to build a realistic traffic matrix. We have run our simulation on two traffic matrices per topology: the actual one (TM) and the double of it (2TM) (where each OD component has been multiplied by 2).

4.2   Results

We have to keep in mind that we made some linear approximation of some objective functions. The original function could give slightly different results in some cases. Also, some (non-linear) objective functions give different results depending on the load of the links. So, the particular traffic matrices and networks on which we made the tests can have its influence as well. Notice that $InvCap$ and $MinHop$ objective functions do not provide exactly the same routing scheme than classical shortest path first algorithm with inverse capacity or unitary metrics. Indeed, our LP model of these objective functions allows extensive and non-equal flow splitting (which is not the case in classical OSPF or ISIS implementations). So our results may present better solutions than the ones obtained by shortest path first algorithms. We have also noticed a negative point for some objective functions: multiple routing schemes achieve the optimal objective function value (especially for u_{max}). By default, the LP solver returns one of these solutions, at random. As we did not want random values in our tables, we have added a small delay contribution to these objective functions ($MIRA$, $InvCap$, u_{max}, $Degrande$ and $MinHop$) so that the LP solver returns a "good" solution from the set of possible equivalent routing schemes. So we should keep in mind that these objective functions could lead to worse results than the ones presented in this section if we do not add this delay contribution and if we do not allow arbitrary flow splitting.

Table 3. Results on network of 25 nodes (Waxman topology). The table contains absolute optimal values (in bold, green, without parentheses), or relative non-optimal values (between parentheses) with respect to the optimal one. The values that are less than 10% from the optimal value are bold. Finally the values that are 2 times worse than the optimal one are in italic and red. For each metric, we present the values for the actual traffic matrix (TM) and for the doubled traffic matrix (2TM).

Objective function	u_{max} %		u_{per10} %		u_{mean} %		ABW_{min} Mbps		l_{mean} Mbps		θ_{tot} Mbps	
	TM	2TM	TM	2TM	TM	2TM	TM	2TM	TM	2TM	TM	2TM
$Fortz$	(1.14)	(1.14)	(1.28)	(1.33)	(1.26)	(1.21)	(0.67)	*(0.55)*	**(1.03)**	**(1.05)**	**(0.97)**	**(0.95)**
$MIRA$	*100*	*100*	(1.40)	(1.48)	(1.17)	(1.16)	*0.0*	*0.0*	(1.15)	(1.10)	**6504**	**5012**
$Blanchy$	(1.22)	(1.23)	**26.0**	**50.0**	(1.13)	(1.12)	(0.88)	**531**	(1.12)	(1.11)	**(0.96)**	**(0.94)**
$Delay$	(1.20)	**(1.08)**	(1.17)	(1.20)	**(1.04)**	(1.11)	**882.0**	**(0.95)**	(1.16)	(1.11)	**(0.97)**	**(0.95)**
$InvCap$	*(2.07)*	*100*	(1.55)	(1.61)	**15.7**	**31.5**	**882.0**	*0.0*	(1.21)	(1.20)	**(0.98)**	**(0.96)**
u_{max}	**34.9**	**69.7**	(1.15)	(1.20)	**(1.07)**	(1.12)	(0.74)	*(0.57)*	(1.17)	(1.11)	**(0.97)**	**(0.95)**
$Degrande$	**34.9**	**69.7**	(1.35)	(1.39)	**(1.05)**	**(1.05)**	(0.74)	*(0.57)*	(1.19)	(1.18)	**(0.97)**	**(0.95)**
$MinHop$	*100*	*100*	(1.29)	(1.43)	(1.27)	(1.25)	*0.0*	*0.0*	**781**	**1578**	**(0.97)**	**(0.95)**

Tables 3 and 4 give the values of the TE metrics at the optimum for each objective function on Waxman and the operational networks. We do not present results on Abilene network due to lack of space. We have removed the θ_{min} metric from the tables because all the objective functions obtained the optimal

value. We have also removed the $delay_{mean}$ metric because corresponding values were very small, or infinite (when $ABW_{min} = 0$). Indeed, we do not take the propagation delay into account and the link capacities are huge. This implies that all the delay values are almost equivalent, because negligible when compared to the propagation delays. Although all the delay values (except infinite values, of course) are tiny, we can point out that the *Delay* objective function gives good results for all the TE metrics on all the topologies. This is because the delay objective embeds most TE concerns (load, utilisation, available bandwidth) and even though the queuing delays are most often negligible, they become non-linearly sufficiently high when the load approaches the capacity to enforce load balancing.

We start our analysis with table 3, which presents results for the topology generated using Waxman's model. We can see that all the objective functions are not equivalent. $MinHop$ is given for comparison purposes (it gives the lowest achievable value for l_{mean}) but is clearly not a good objective function on its own. Indeed, it leads to a high value of u_{max} which is a very important concern. The lowest achievable value for u_{max} is given by the u_{max} function which only optimises this variable. The lowest achievable value of the u_{mean} variable is given by $InvCap$. This function is not very good on its own because it leads in this case to a high u_{max} value. The combined $Degrande$ is a very good objective function on this topology. Indeed, it gives nearly optimal values for all the metrics. $Blanchy$, $Fortz$ and $Delay$ are quite good. We notice also that $MIRA$ is good except for u_{max} which is 100% (and thus $ABW_{min} = 0$ and $delay_{mean}$ is infinite). We analyse this fact as follows. $MIRA$ is based on max-flows (and only on max-flows). Suppose that we have two routes in the network for a particular OD pair of nodes. The value of the residual max-flow will be the same if we route all the traffic on one route or if we route half of it on each route. This is the cause of the bad load balancing policy and the high value of u_{max} given by $MIRA$.

To better discriminate the *Degrande*, *Delay* and *Blanchy* functions we we can analyse the results corresponding to the doubled traffic matrix. In this case,

Table 4. Results on the operational network. See the legend of table 3 to understand these values.

Objective function	u_{max} %		u_{per10} %		u_{mean} %		ABW_{min} Mbps		l_{mean} Mbps		θ_{tot} Gbps	
	TM	2TM	TM	2TM	TM	2TM	TM	2TM	TM	2TM	TM	2TM
$Fortz$	(1.18)	(1.13)	(1.63)	(1.17)	(1.17)	(1.14)	(0.89)	(0.56)	**(1.00)**	**(1.04)**	(0.99)	(0.98)
$MIRA$	(1.41)	*100*	(1.63)	(1.65)	**(1.07)**	**(1.09)**	(0.75)	*0.0*	(1.03)	(1.05)	4331	4027
$Blanchy$	(1.16)	(1.15)	**14.2**	**28.5**	**(1.07)**	(1.11)	**(0.90)**	*(0.50)*	(1.24)	(1.23)	(0.99)	(0.97)
$Delay$	**(1.04)**	**(1.02)**	(1.32)	(1.21)	**(1.01)**	**(1.02)**	**(0.97)**	**(0.92)**	(1.15)	(1.17)	(0.99)	(0.99)
$InvCap$	(1.18)	**(1.09)**	(1.51)	(1.49)	**6.9**	**13.8**	(0.89)	(0.69)	(1.19)	(1.19)	(0.99)	(0.98)
u_{max}	**38.4**	**76.9**	(1.56)	(1.21)	**6.9**	**(1.01)**	**95.7**	**36.0**	(1.20)	(1.15)	(0.99)	(0.99)
$Degrande$	**38.4**	**76.9**	(1.51)	(1.49)	**6.9**	**13.8**	**95.7**	**36.0**	(1.19)	(1.19)	(0.99)	(0.98)
$MinHop$	(1.36)	*100*	(1.76)	(1.60)	(1.16)	(1.20)	(0.78)	*0.0*	**262**	**525**	(0.99)	(0.98)

Blanchy has a quite high value of u_{max}, while *InvCap* leads to a fully loaded link ($u_{max} = 100\%$). *Degrande* and *Delay* are in this case the best objective functions. *Fortz* is also quite good in this situation.

On table 4 we can see the results for the operational topology. *Blanchy* obtains good values for all the metrics and the best value of u_{per10}. *MIRA* logically gives the optimum for the θ_{tot} variable, which is its objective function. We remark that many other objective functions give values close to this optimal θ_{tot} value. On the operational network, we consider that the best compromise is *Degrande* because it gives almost optimal values for all the variables except u_{per10}. Both *Delay* and *Blanchy* are quite good and give better results for u_{per10}. *Fortz* improves l_{mean} at the expense of all the other variables. *MIRA* and *MinHop* give high values regarding u_{max}.

We have noticed on the Abilene network that there is less variation between the values of our metrics. But we have still pointed out the performance of *Delay* and *Degrande* which are the best objective functions of these simulations.

One last important point is the fact that at low load, we can see that *Fortz* is approaching the optimal value of l_{mean}, the objective of *MinHop*, while *Delay* is approaching the optimal value of u_{mean}, the objective of *InvCap*. This confirms the approximation we made in section 3.6.

Table 5. Metrics At Low Load (LL) and High Load (HL)

Objective Function	u_{max}		u_{per10}		u_{mean}		ABW_{min}		l_{mean}		θ_{tot}	
	LL	HL	LL	HL	LL	HL	LL	HL	LL	HL	LL	HL
Fortz	√	√	√	√	√	√	±	±	√	√	√	√
MIRA	•	•	•	•	√	√	•	•	√	√	√√	√√
Blanchy	√	•	√	√	√	√	√	•	√	√	√	√
Delay	√	√	√	√	√	√	√	√	√	√	√	√
InvCap	•	•	√	±	√√	√√	±	•	√	√	√	√
Degrande	√√	√√	√	√	√	√	√	√	√	√	√	√

To conclude this section, we analyse table 5 which presents the good ($\sqrt{}$) and bad ($\bullet$) metrics for each objective function at low and high load[5]. On this table, we see that *Fortz*, *Delay* and *Degrande* are the best because these have no red point.

5   Conclusion

In this paper, we have shown how well-known network-wide objective functions reflect requirements for Traffic Engineering. As our results reflect, they are not equivalent. We have shown the power of some functions and the weaknesses of others. We have outlined that, although the transmission + queueing delay is

---

[5] In this table, $\sqrt{}\sqrt{}$ is used to denote the optimal value and $\pm$ to denote a value which is not bad, but which is not as good as $\sqrt{}$ values.

often negligible, choosing this delay as objective function gives good results for almost all TE metrics. It is not that surprising considering that almost all TE link metrics feed into the delay (see figure 1).

The best objective functions are *Delay* and *Degrande* on the tested topologies. We have a preference for *Delay* because it does not need any configuration or parameter. *Fortz* is quite good also in all the situations, while having performance somewhat under *Delay* and *Degrande*. *Blanchy* has good results also, except for highly loaded networks. *MIRA* gives good solutions concerning the total residual max flow, but this function gives bad results concerning the maximal link utilisation.

This study provides an objective basis to select an objective function when designing a new Traffic Engineering routing algorithm. It may also be useful to revisit existing TE algorithms to make them work with the objective functions that best match the various TE concerns we have studied. Furthermore, while this study has been performed for packet switched networks, the objective functions and TE metrics used (see table 5) are also valid in circuit switched networks.

Acknowledgments

References

1. B. Fortz and M. Thorup. Internet Traffic Engineering by Optimizing OSPF Weights. In *Proc. of IEEE INFOCOM*, pages 519–528, 2000.
2. M. S. Kodialam and T. V. Lakshman. Minimum interference routing with applications to MPLS traffic engineering. In *Proc. of IEEE INFOCOM*, pages 884–893, 2000.
3. F. Blanchy, L. Mélon, and G. Leduc. An efficient decentralized on-line traffic engineering algorithm for MPLS networks. *Proc. of 18th ITC*, pages 451–460, 2003.
4. A. Elwalid, C. Jin, S. H. Low, and Indra Widjaja. MATE: MPLS adaptive traffic engineering. In *Proc. of IEEE INFOCOM*, pages 1300–1309, 2001.
5. N. Degrande, G. Van Hoey, P. de La Vallée-Poussin, and S. Van den Busch. Inter-area traffic engineering in a differentiated services network. *J. Networks Syst. Manage.*, 11(4), 2003.
6. G. Leduc, H. Abrahamsson, S. Balon, S. Bessler, M. D'Arienzo, O. Delcourt, J. Domingo-Pascual, S. Cerav-Erbas, I. Gojmerac, X. Masip, A. Pescaph, B. Quoitin, S.F. Romano, E. Salvatori, F. Skivée, H.T. Tran, S. Uhlig, and H. Ümit. An Open Source Traffic Engineering Toolbox. To appear in Computer Communications, 2006.
7. B.M. Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1671–1622, Dec 1988.
8. S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon. Providing public intradomain traffic matrices to the research community. *SIGCOMM Comput. Commun. Rev.*, 36(1):83–86, 2006.