A scalable heuristic for hybrid IGP/MPLS traffic engineering - Case study on an operational network

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

***Abstract*— In current IP networks, a classical way to achieve traffic engineering is to optimise the link metrics. This operation cannot be done too often and can affect the route of a lot of traffic. Multiprotocol Label Switching (MPLS) opens new possibilities to address the limitations of IP systems concerning traffic engineering thanks to explicit label-switched paths (LSPs). This paper proposes a new method based on simulated annealing meta-heuristic to compute a set of LSPs that optimise a given operational objective. The hybrid IGP/MPLS approach takes advantage of both IP and MPLS technologies and provides a flexible method to traffic engineer a network on a day to day basis. We illustrate the capabilities of our method with some simulations and a comparison with other techniques on an existing operational network. The results obtained by setting up a small number of LSPs are nearly optimal and better than by engineering the IGP weights. Moreover, although it could be combined with a static setting of the latter, SAMTE alone gives already the same results as this combination in much less CPU time, which thus allows an administrator to keep its initial and meaningful IGP metrics in his network.**

***Index Terms*— Traffic Engineering, MPLS, Simulated Annealing, TOTEM, hybrid IP/MPLS**

I. Introduction

Due to rapid growth of the Internet and the requirements for quality of service, ISPs must build scalable network architectures and need to better engineer their traffic. Traffic engineering (TE) is defined ([1]) as mapping traffic flows onto an existing physical network topology in the most effective way to accomplish desired operational objectives.

In an IP network, a classical and simple way for improving an operational objective is to change the link metric. By setting the link metric values appropriately, it is possible to adjust the routes and react to the current load situation (see [2] and [3]).

But this approach has several limitations. Firstly, due to the shortest path calculation, whenever two traffic flows, which are destined for the same egress node, cross each other's way, they are merged. So they use the same links, possibly causing congestion while other links are still only lightly utilised.

A second deficiency of IGP-based traffic engineering lies in the transient behaviour while changing the routing pattern from one metric setting to another. A recent study [4] investigates a better way to distributes the link state update to reduce the convergence time after a link failure or a metric change.

Multiprotocol Label Switching (MPLS) was developed to overcome the limitations of conventional routing protocols. MPLS allows the specification of explicit routes through the network, so-called Label Switched Paths (LSPs). Classical MPLS traffic engineering aims at finding a full mesh of LSPs, i.e. one between each nodes pair, to optimise a given objective.

The main drawback of the MPLS full mesh approach is the scalability. Indeed, with a network of 200 nodes, the full mesh contains 39800 LSPs. Another practical drawback for an operator is the transition between a pure IP routing and a full mesh of LSPs i.e. some operators are afraid of this "big bang" migration (non-incremental) that can reduce the stability of their current IP network.

Between the pure IP metric-based optimisation and the full mesh of LSPs, a hybrid IGP/MPLS traffic engineering approach can combine both the simplicity and robustness of IGP routing and the flexibility of MPLS. In this study, we propose a novel hybrid IGP/MPLS traffic engineering approach based on simulated annealing meta-heuristic to optimise a given operational objective. The basic idea of an hybrid approach is to use IP routing by default and add only a few LSPs to improve the utilisation of the resources. As the number of LSPs is a parameter, this approach provides a scalable solution that can be deployed incrementally on any operational network.

Moreover, our approach wants to avoid "yo-yo" networks in which too many re-optimisations produce instability in the network. Thanks to the use of MPLS, we better control which traffic will be affected by the addition or the change of an LSP. In this paper, we investigate only the optimisation of a single traffic matrix. But, this approach can be extended to take into account multiple traffic matrices.

In the literature, we see different approaches but few

S. Balon is a Research Fellow of the Belgian National Fund for the Scientific Research (F.N.R.S).

papers compare their methods with existings ones or with other approaches achieving the same goal. The traffic engineering toolbox TOTEM [5] aims at creating a repository of TE methods available for operators and for researchers. By integrating our method is this toolbox, we are able to compare our results with MPLS-based approach like [6] or IP-metric based approach like [2].

In section II, we present related works dealing with hybrid IGP/MPLS TE approach. In section III, we describe the routing model we use to implement hybrid IGP/MPLS methods. Then in section IV, we describe our simulated annealing based heuristic and in section V, we illustrate this method by simulations and comparisons on an operational network.

II. Related work

To the best of our knowledge, there are only a few publications that consider hybrid IGP/MPLS scenarios for traffic engineering ([7] [8] [9] [10] [11]).

In [7], Ben-Ameur is one of the first to deal with an hybrid combination of IGP and MPLS. Ben-Ameur and al. explore the different routing strategies (single-path and multi-path) and their possible realization in an IP intra-domain network. The authors identified three models for combining IGP and MPLS: basic IGP shortcut, IGP shortcut and generalised IGP routing. Ben-Ameur and al. compare the performance and the complexity of different routing strategies in IP networks. The complexity of a routing pattern is defined as the number of MPLS tunnels (LSPs) needed. They propose two off-line traffic engineering methodologies based on mathematical programming for IP intra-domain networks: the first one is based on IGP/ MPLS architecture and the second one is based only on IGP routing using an optimized load balancing scheme. [7] introduces concepts and formulate problems but does not provide solutions to solve them efficiently. The mathematical programming approaches are very slow and only applicable to very small topologies.

The recent RFC3906 [11] describes a way of modifying the current Dijkstra implementation to take tunnels into account. The tunnel must be advertised in the IGP protocol with an associated metric. The metric can be absolutely fixed or be relative to the link metrics of the tunnel's path. This RFC supports multiple paths with possible traffic forwarded on shortest path and tunnels. This RFC does not provide a lot of details and many uncertainties remain.

In [8], Riedl takes another approach in which the IGP optimisation is performed first and a set of MPLS tunnels can be computed to improve the IGP solution. These two steps are completely separated. Riedl proposes a new heuristic based on simulated annealing to optimise the IGP metrics. This algorithms takes into account the original configuration and allows tradeoff considerations between routing optimality and adaptation impact. In a second step, the MPLS tunnels are computed using an mixed-integer programming (MIP) model. Riedl shows that a small number of LSPs decreases greatly the most loaded link and compares the combination of IGP optimisation with or without MPLS tunnels. [8] provides a simple heuristic for IGP metric optimisation but not for LSPs computations. Our solution computes quickly these LSPs based on a given set of IP metric.

In [9], Mulyana proposes a novel TE method based on genetic algorithms to optimise IGP/MPLS networks. This method can be considered as an off-line TE approach to handle long or medium-term traffic variations. In their approach the maximum number of hops as well as the maximum delay of an LSP and the maximum number of LSPs that can be installed in the network are treated as constraints. They apply the method to the German scientific network (G-WiN) with a randomly generated traffic matrix. They compare the results of the method for several hybrid routing schemes (presented in [7]) and pure IGP routing.

In [10], the authors investigate the effect of partial demand increase on the performance of the network and propose a simple policy scheme to decide whether re-optimization should be performed. Two re-optimization approaches based on plain local-search and simulated-annealing are presented. They apply their method for metric based traffic engineering scheme to the German scientific network (G-WiN) for which a traffic matrix and several traffic-increase patterns were randomly generated.

In opposition to slow mathematical programming approach, in this paper, we are interesting in designing an heuristic used to make a regular (few seconds) and offline re-optimisation of the network.

III. Problem definition

In a classical IP network, routing is very simple and uses the shortest path to route the demands. By adding tunnels (LSPs) in an classical IP network, the routing becomes more complex and different routing models can be defined (see [7] and [9]).

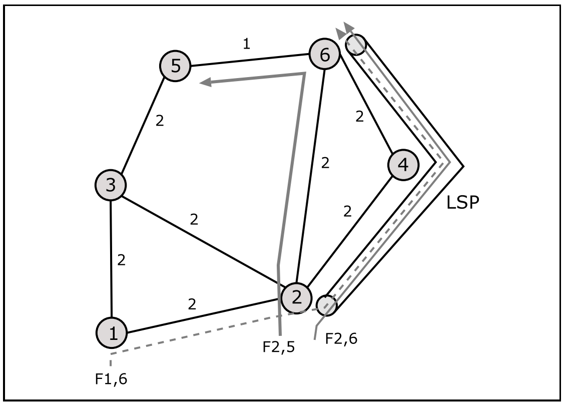


Fig. 1. *Shortest path in the Basic IGP shortcut*

In this paper, we choose to investigate only the Basic IGP shortcut model (BIS) (proposed in [9]) that provides a simple and scalable model. This model can be implemented easily in real routers and seems to allow enough flexibility to improve the considered traffic engineering objectives substantially.

Before explaining this model, we need to define some concepts. In this study, we deal with intra domain traffic engineering in which traffic enters the network at the ingress and leaves the network at the egress. Traffic is identified by an origin/destination (OD) pair or equivalently by an ingress/egress pair. The traffic matrix represents the traffic from all ingress to all egress nodes.

Suppose an LSP from node A to node B, in the BIS model, all the packets arriving at node A with destination B will be forwarded in the LSP. This model is the most simple and can be easily implemented in real networks. A simple lookup in the BGP table gives the next-hop for any prefix. If an LSP exists to this next-hop, it will be used to forward the traffic. The LSP appears like a virtual interface in the forwarding table.

The other models differs by the way they choose the packets to forward in the LSPs. In this paper, we do not have space to explain and compare to the other models. For more information, see [7] and [9].

Model 1: IGP/MPLS hybrid model

---

indices

- $d = 1, 2, ..., D$ demands
- $p = 1, 2, ..., P$ candidate paths for LSPs
- $e = 1, 2, ..., E$ links

parameter

- $\delta_{dp}(x)$ $= 1$ if demand d uses LSP p and 0, otherwise
- $\tau_{pe}(x)$ $= 1$ if LSP p contains link e and 0, otherwise
- γ_{de} $= 1$ if demand d uses link e without crossing an LSP and 0, otherwise
- h_d volume of demand d
- c_e capacity of link e
- K maximum number of LSPs in the solution

variables

- x_p binary variable set to 1 if LSP p is chosen for the solution

constraints

$$\sum_p x_p \leq K$$

$$l_e = \sum_d \gamma_{de} h_d + \sum_p \sum_d \delta_{dp} \tau_{pe} x_p \quad \text{for} \quad e = 1, 2, ..., E$$

$$l_e \leq c_e, \quad e = 1, 2, ..., E$$

With this formulation, δ_{dp} and γ_{de} depend on the routing model used. l_e is computed as the sum of the IGP traffic and the LSP traffic. The problem is then to find a set of LSPs that optimise a given operational objective. In this paper, we choose to study two kinds of objectives.

The first is to minimize f_{ML}: the load of the most loaded link where $\rho_e = \frac{l_e}{c_e}$ is the relative load on link e.

$$f_{\mathrm{ML}}: \quad \max\{\rho_e \mid \forall e\}$$

The second objective is to minimise f_{LB} : an hybrid function of load balancing and traffic minimisation

$$f_{\mathrm{LB}}: \quad \sum_e (\rho_e - \overline{\rho_e})^2 + \alpha \sum_e \rho_e^2$$

$$\text{with} \quad \overline{\rho_e} = \frac{1}{|E|} \sum_e \rho_e$$

This function 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 weighting factor α allows us to give more importance to one aspect or the other. For futher details on this function see [6].

IV. Heuristic description

Our solution uses a simulated annealing meta-heuristic to compute a good solution in reasonable time. The intuitive idea of the heuristic is to test different combinations of tunnels and select a better solution in the neighbourhood of the current solution. The algorithm uses a precomputed candidate path list that contains all the allowed tunnels. We compute the candidate path list as follows: for each source/destination pair we choose the P shortest paths of maximum H hops.

The simulated annealing meta-heuristic is based on an analogy taken from metallurgy. To grow a metal, you start by heating a row of materials to a molten state. You then reduce the temperature of this metal melt until the metal structure is frozen in. If the cooling is done too quickly, some irregularities are locked in the metal and the trapped energy level is much higher than in perfectly structured metal.

An optimisation problem can be solved by a similar method. We choose a solution in the neighbourhood of the current solution. If the new solution is better, we accept it, otherwise we accept it only with a probability function of the temperature that decreases during the execution of the algorithm. With this method, we allow large movements in the solution space when the temperature is high and reduce this movement by reducing the temperature. This heuristic avoid the algorithm to be blocked in a local optimum.

Our heuristic (described in Algorithm 1) starts with an initial temperature of T_0 and keeps this temperature during a whole plateau of size L. The decreasing of the temperature is given by the cooling ratio α. The stop condition is defined as "stop if less than E2 moves are accepted in the last K2 plateaus".

Two other problem-specific components must be defined: the initial solution and the neighbourhood. The initial solution is generated by selecting K tunnels at random in the candidate path list. A neighbourhood is defined as "two solutions are neighbours if they differ only by one LSP". The neighbourhood function replaces one LSP of the solution by another LSP from the candidate path list. A summary of the parameters is given in Table I.

Algorithm 1: Simulated Annealing

```
/* x_0, x* and x are the initial, the best
   and the current solution */
/* F(x) is the evaluation function */
/* move(x) return a neighbour of x */
x* ← x_0; x ← x_0; T ← T_0;
while not stopCondition do
    nbIter ← 0;
    while nbIter < L do
        x' ← move(x) ;
        if F(x') < F(x) then
            x ← x' ;
            if F(x') < F(x*) then
                x* ← x'
            end
        else
            ΔF ← F(x') − F(x) ;
            p_k ← e^(−ΔF/T) ;
            if random() < p_k then
                x ← x' ;
            end
        end
        nbIter ← nbIter +1;
    end
    T ← αT;
end
```

The performance bottleneck of the algorithm is the evaluation of the score function F. For each new solution to evaluate, we need to compute the load on each link. The link load depends of the path that each flow will use. A naïve approach is to recompute the path for all the source/destination pairs for each new solution. Another approach is to recompute only the path for the pair that can be affected by the modification taken on a solution (add/remove LSP). For this purpose, we store a matrix (called PUN: Pair Using Node) that stores for each node all the path that contain this node. So, when we compute a neighbour of a solution (i.e change an LSP), we recompute the path for all pairs that use the ingress of the removed LSP and of the new LSP. This optimisation improves hugely the execution time.

V. Simulation on an operational network

In this section, we present simulations on a real operational network composed of about 20 routers and 40 links. To build a realistic traffic matrix on this network, we collect the netflow data on each interface and we aggregate them to build a traffic matrix. The traffic matrix used in these simulations is measured on the 18 January 2005 between 12h00 and 14h00.

In the simulation, we compute the bandwidth consumption of each link according to the traffic matrix. We used LSPs with zero provisioning bandwidth and take into account the real bandwidth of the flows routed in each LSPs. All the simulations are done with ECMP disabled and with a candidate path list generated with P=5 and H=7. The simulations are done on a laptop with a 1.3Ghz centrino processor and 512 Mb of RAM. All the source codes for the simulations are available in the TOTEM toolbox[1].

The first question is "How does the number of LSPs affect the objective?". Figure 2 shows the influence of the number of LSPs on the two objective functions. We can see that three LSPs are enough to reduce the maximum link load from 70.7% to 42.1% with the f_{ML} objective function. On the other hand, if we prefer to load balance the network with the f_{LB} objective function, with 12 LSPs, the most loaded link decreases from 70.7% to 42.5%.

Figure 3 shows a typical execution of the simulated annealing. This execution is done with 5 LSPs with the f_{ML} objective function. We can see the evolution of the best solution and the current solution. When the temperature decreases, less and less bad solutions (i.e., solutions that increase the objective), are chosen. Relatively soon, the best solution cannot be improved and the execution converges.

We also studied the impact of the simulated annealing parameters: T_0, L, α, $E2$, $K2$ and do we not see a big influence on the quality of the solution. The following empirical rules give good results: T_0 can be taken for obtaining 50% of accepted moves in the first plateau, the length of a plateau L can be taken between 0.1 and 1 of the size of the neighbourhood. We have tested 576 sets of parameters with $T_0 \in [0.01, 0.03]$, $L \in [1259, 12590]$, $\alpha \in [0.8, 0.975]$, $E2 \in [2, 12]$ and $K2 \in [2, 6]$. For the f_{ML} objective with 4 LSPs, these parameters give solutions with max loaded link between 42.1% to 45.5% in a time between 130 ms and 9780 ms. A good set of parameters is ($T_0 = 0.023$, $L = 2500$, $\alpha = 0.9$, $E2 = 5$, $K2 = 4$). In 5 executions, it gives a mean of 42.3% in the maximum loaded link in 1030 ms.

Table II compares our approach with other propositions. The colums correspond respectively to the name of the method, the number of LSPs required, the maximum loaded link, the 10th percentile[2] of the links load, the mean of the links load, the standard deviation and finally, the CPU time to execute the method.

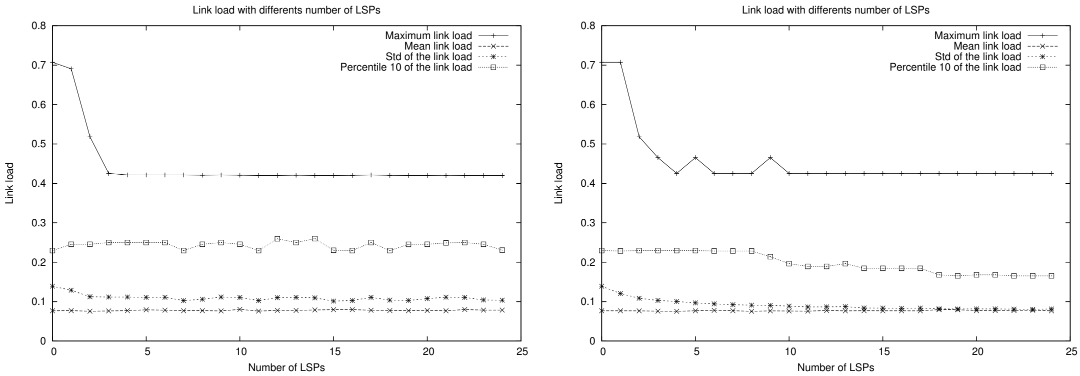
[1] http://totem.run.montefiore.ulg.ac.be/

[2] The N percentile gives the load of the link for which N% of the links of the network are more loaded than this link.

P	Maximum number of shortest paths by demand in the candidate path list
H	Maximum number of hops of a path of the candidate path list
K	Number of LSPs in the solution
T_0	Initial temperature of the simulated annealing
L	Size of a plateau in the simulated annealing
α	Cooling ratio
$E2$ & $K2$	Stopping condition: less than $E2$ accepted moves in the $K2$ last plateaus

TABLE I

SUMMARY OF ALL THE PARAMETERS



(a) (b)

Fig. 2. *Influence of the number of LSPs on (a) f_{ML} and (b) f_{BL} objective functions*

Method	#LSP	Max in %	Per10[2] in %	Mean in %	Std in %	CPU time
MCNF	506	41.9	-	-	-	> 2 days
SPF-ActualMetrics	0	70.7	23.0	7.1	11.5	0
SPF-InvCap	0	46.3	22.3	6.9	9.6	0
IGP-WO	0	45.1	22.2	7.2	9.6	315 s
DAMOTE$_{\alpha=2}$	506	41.9	16.1	8.5	7.5	2.5 s
SPF-ActualMetrics + SAMTE$_{f_{ML}}$	4	42.1	24.5	7.5	10.7	1.0 s
SPF-ActualMetrics + SAMTE$_{f_{LB}}$	12	42.5	18.9	7.8	8.4	3.1 s
SPF-ActualMetrics + SAMTE$_{f_{LB}}$	23	42.5	16.5	7.8	8.0	13.8 s
IGP-WO	0	45.2	22.8	7.8	9.9	315 s
IGP-WO + SAMTE$_{f_{ML}}$	2	42.0	22.8	7.8	9.9	315.5 s
IGP-WO + SAMTE$_{f_{LB}}$	12	42.2	18.4	7.6	8.2	320.8 s
IGP-WO + SAMTE$_{f_{LB}}$	20	42.2	17.0	7.9	7.9	327.7 s

TABLE II

COMPARISON BETWEEN SAMTE, OTHER TE APPROACHES AND THE COMBINATION OF METRIC-BASED OPTIMISATION WITH SAMTE

MultiCommodity Network Flow[2] (MCNF) gives the lower bound on the maximum loaded link obtained without flow spliting. The second line (SPF-ActualMetrics) of the Table II gives results with the real metrics of the operational network. These metrics take several factors into account, such as bandwidth and delay. IGP-WO is a method proposed by Fortz in [2] that optimises the IGP metric to reduce a piecewize linear function. This function penalises the most loaded links. We configured IGP-WO to compute IGP metrics between [0,20] in 500 iterations. With IGP-WO, the most loaded link is reduced in comparison with the classical CISCO recommendation (SPF-InvCap in Table II) to take the inverse of the capacity as metric. SAMTE applied on the real IGP metric of the operational network with f_{ML} objective achieves better results with only 4 LSPs in 1.0 second (SPF-ActualMetrics + SAMTE$_{f_{ML}}$ in Table II).

We use DAMOTE [6] to compute an LSP between each source/destination pair to route the demand. We configure DAMOTE to use the f_{LB} score function with $\alpha = 2$. DAMOTE gives a better result with a most loaded link of

[2]MCNF gives a lower bound for the max load. Complete MCNF took too much time to produce the solution (more than two days). In this paper, we have relaxed some constraints not related to the bottleneck link. This simplest (and thus faster) problem gives a lower bound which is less or equal to the actual lower bound of the initial problem. But, we have noticed that there exists a solution (DAMOTE) for which the max load is equal to this value. Thus, we can say that this value is the optimal solution of the MCNF problem as well.

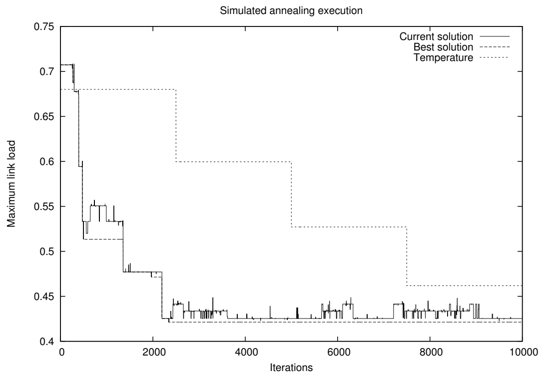


Fig. 3. *Typical execution of the simulated annealing with the f_{ML} objective function with five LSPs*

41.9% but requires a full mesh of 506 LSPs. Our method configured with the f_{LB} objective function gives very good result with 12 LSPs.

We can also combine a metric-based optimisation with SAMTE. Thanks to the integration of all these tools in the TOTEM toolbox [5], we can first optimise the metrics using the IGP-WO method and then execute SAMTE with the f_{ML} objective function. With just two LSPs, SAMTE can reduce the maximum link load from 45.2% to 42.1% (see the third part of Table II). Perhaps the gain is not enough to create the two LSPs but the major advantage is that SAMTE can be used to improve other kinds of objectives using the flexibility of MPLS. Indeed, if we use SAMTE with the f_{LB} objective, with 12 LSPs, we can reduce the standard deviation from 9.9% to 8.2% and the 10th percentile[2] from 22.8% to 18.4%. With 20 LSPs, the improvement is clear with a standard deviation reduced to 7.9 and a 10th percentile to 17.0%.

VI. Conclusion

The hybrid combination of (the simplicity of) IGP routing and (the flexibility of) MPLS explicit routing gives interesting results. Our method, based on the simulated annealing meta-heuristic, can be used to select LSPs that optimise any given operational objective. An operator can choose the number of LSPs that he/she is ready to afford to engineer the network. This method is independent of the IGP metric configuration. An operator can keep his/her favourite metric configuration and set up only a few LSPs to improve any kind of objective. Moreover, with the computed LSPs, the problematic flows are identified. An administrator knows exactly which traffic flows are routed along the LSPs and keeps control on all the traffic paths.

Another flexible parameter is the candidate path list that makes the set of allowed LSPs explicit. This list can integrate different kinds of constraints like avoiding risky or costly links, etc.

We have shown results on an operational network with a real traffic matrix. The results obtained by setting up a small number of LSPs are nearly optimal and better than by engineering the IGP weights. Moreover, although it could be combined with a static setting of the latter, SAMTE alone gives already the same results as this combination in much less CPU time, which thus allows an administrator to keep its initial and meaningful IGP metrics in his network. Furthermore, recomputing the paths of the few LSPs when the traffic matrix changes turns out to be easy and not time consuming, which makes SAMTE a suitable adaptive method.

Finally, our method is integrated in the TOTEM toolbox [5] which makes it easily available to anyone, either to use it or to compare it with other approaches. In futher study, we will study SAMTE with other kinds of objectives such as the resilience or the minimization of the reoptimisation in case of traffic matrix change. We will also provide results on a large set of random topologies.

Acknowledgments

This work was partially supported by the Walloon Region (Belgium) in the framework of the WIST programme (TOTEM project) and by the E-NEXT European Network of Excellence.

References

[1] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao. Overview and principles of Internet traffic engineering. RFC 3272, Internet Engineering Task Force, May 2002.

[2] B. Fortz and M. Thorup. Optimizing OSPF/IS-IS weights in a changing world. *IEEE Journal in Selected Areas in Communications*, 20(4):756–767, 2002.

[3] U. Killat E. Mulyana. A hybrid genetic algorithm approach for OSPF weight setting problem. In *Second Polish-German Teletraffic Symposium PGTS*, 2002.

[4] P. François and O. Bonaventure. Avoiding transient loops during IGP convergence. In *IEEE INFOCOM*, 2005.

[5] 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. *Computer Communications*, 29(5):593–610, March 2006.

[6] F. Blanchy, L. Mélon, and G. Leduc. A Preemption-Aware Online Routing Algorithm for MPLS Networks. *Telecommunication Systems*, 24:187–206, 2-4, Oct.-Dec. 2003.

[7] W. Ben-Ameur, B. Liau, and N. Michel. Routing strategies for IP networks. *Telektronikk Magazine*, 2/3 2001.

[8] A. Riedl. Optimized routing adaptation in IP networks utilizing OSPF and MPLS. In *Proceedings of IEEE ICC*, May 2003.

[9] E. Mulyana and U. Killat. Optimization of IP networks in various hybrid IGP/MPLS routing schemes. In *Proceedings of the 3rd Polish-German Teletraffic Symposium PGTS*, pages 295–304, Dresden Germany, September 2004.

[10] E. Mulyana and U. Killat. Impact of partial demand increase on the performance of IP networks and re-optimization approaches. In *Proceedings of the 3rd Polish-German Teletraffic Symposium PGTS*, pages 275–284, Dresden Germany, September 2004.

[11] N. Shen and H. Smith. Calculating Interior Gateway Protocol (IGP) routes over traffic engineering tunnels. RFC 3906, IETF, October 2004.

[12] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. RFC 3031, Internet Engineering Task Force, January 2001.