Implementation of a traffic engineering technique that preserves IP Fast Reroute in COMET

Ho Trong Viet, Pierre Francois, Yves Deville and Olivier Bonaventure

Dept CSE, Université catholique de Louvain (UCL), Belgium.

Avec l'augmentation du traffic dans les réseaux IP et les contraintes en termes de disponibilité imposées par les clients, les opérateurs de grands réseaux doivent utilisent des techniques d'ingénierie de traffic et de reroutage local. Nous proposons une méthode d'ingénierie de traffic par optimisation des poids des liens dans un réseau OSPF qui minimise la charge des liens et maximise les possibilités de reroutage local avec la technique des Loop-Free-Alternates. Notre méthode est implémentée en COMET et son évaluation montre qu'elle est aussi efficace qu'une implémentation optimisée de la méthode de Fortz & Thorup.

Keywords: IP Routing, Traffic Engineering, Loop Free Alternates, Local search, Combinatorial Optimization.

1 Introduction

Enterprise and ISP networks have deployed IP Routing protocols such as OSPF and IS-IS for a long time. The use of the OSPF/IS-IS can provide a variety of practical advantages as described in [1]. However, the current network topology and capacity may seem insufficient to meet the increasing traffic demand [2]. Thus, an efficient use of available resources without changing the protocols becomes a critical need of many Internet Service Providers. By optimizing the configuration (Traffic Engineering) of link weights, operators [1, 2] can significantly improve the load of their links, which is known to be a NP-hard problem.

Another emerging issue for ISP networks is the fast restoration of IP services in the case of failures. Fast Reroute techniques have been initially proposed for MPLS networks. Recently, techniques have been proposed for pure IP networks [6]. The Loop-Free-Alternates (LFA) [7] is one of the most promising technique and is now supported by the major high-end router vendors. An LFA for a router A, for a destination d, is a neighbor C of A which does not use A to reach d, so that A can prepare itself to directly reroute the traffic destined to d along its link with C upon the failure of the link that A is using to reach d [7]. In a network using OSPF, the number of links that can be totally protected by using LFAs will depend on the setting of the link weights. Hence, a traffic engineering method relying on the setting of link weights may have an impact on the applicability of this Fast Reroute technique.

In this context, our objective is to provide a Traffic Engineering technique that takes the coverage of LFAs into account during the network resource usage optimization. Constraint Programming(CP) and Constraint-Based Local Search (CBLS) are well suited for solving such complex combinatorial problems. COMET is an object-oriented language with several innovative modeling and control abstractions for CP and CBLS. Especially in COMET, some classical problems can be modeled in only about a dozen lines of code [5]. So, we have chosen a local search algorithm implemented in COMET for solving our traffic engineering problem.

There are many previous works on optimizing OSPF weights, see [1,2] and the references therein, but to the best of our knowledge, neither of them has taken both optimizing OSPF weights and maximizing LFA coverage into account. The tabu search of Fortz et Thorup [2] seems to be one of the best local search techniques for solving this traffic engineering problem. In this paper, our works will be compared with the tabu search of Fortz et Thorup which is implemented in the IGP Weight Optimization module of the TOTEM - TOolbox for Traffic Engineering Methods [8].

This paper is organized as follows. We firstly present the study system and our proposed heuristic in Section 2. We introduce the algorithms based on local search and utilize a key sequence of metrics that permits to modify the link weights in each search step. Our evaluation is presented in Section 3 with an analysis of the experimental results, and we conclude the paper in Section 4.

2 Problem formulation

2.1 Study system

We consider our network as a directed graph $G = (N, A)$, in which the nodes represent the routers and the arcs represent the links. Each link (s,t) has a bandwidth BW[s,t]. We call IGP the matrix of the OSPF link weights and TD the matrix of the traffic demands, $TD[s,t]$ tells us how much traffic router s wants to send to router t. From TD and IGP, one can find the load of each link by distributing, for each pair of node (s,t), $TD[s,t]$ amount of traffic over the shortest paths from s to t in the graph described by IGP, using the Equal Cost Multi-Path technique (ECMP).
Definition of an optimal $IGP*$ assignment for the matrix of link weights IGP
We call L the matrix of the actual load on each link (s,t):

$$L[s,t] = \sum flow\ over\ link\ (s,t)$$

We define the utilization of a link (s,t) : $U[s,t] = L[s,t]/BW[s,t]$. This means that the load is kept within the capacity if U[s,t] ≤ 1, otherwise the link (s,t) is overloaded.
Suppose U_{max} = max{U[s,t] $\|$ $\forall$ link (s,t)}. Our objective is to find the optimal assignment $IGP*$ minimizing the value of U_{max} and maximizing the LFA coverage (the number of links that can be protected by LFAs). In this paper, we only consider per-link LFAs (see [7]).

2.2 The system

As mentioned above, we have implemented the LSA4IGPWO (Local Search Algorithms for the IGPWO problem) based on tabu search on COMET. Our COMET program contains only 759 lines of code. Each one in the four tasks : Computation of link load, Find All Pairs Shortest Path, Computation of LFAs, and Local Search is written in about 100 lines of code.
In Algorithm 1, we described the simple pseudo-code of our local search algorithms. The 4 steps of LSA4IGPWO,

Algorithm 1 Pseudo-code for LSA4IGPWO

```
1:  generate initial configurations
2:  while not termination-criteria do
3:      select a link (s,t)
4:      IGP[s,t] ⇐ IGP[s,t] ± Δ
5:      evaluate IGP
6:      if f(IGP) < f(IGP∗) then
7:          IGP∗ ⇐ IGP
8:      end if
9:      if expired_time then
10:         generate IGP
11:     end if
12: end while
```

according to the pseudo-code, are:

- Line 1 : An initial configuration of link weights IGP is generated by choosing alternately one of three options : the initial link weights, and the ones obtained with the following heuristics:
  - Heuristic 1 : The weight of a link (i,j) is proportional to the sum of the incoming traffic demands of j divided by bandwidth of (i,j) : $IGP[i,j] \sim total\_Incoming[j]/Bandwidth[i,j]$.
  - Heuristic 2 : The weight of a link (i,j) is proportional to the bandwidth of (i,j) divided by the sum of the outgoing traffic demands of j : $IGP[i,j] \sim Bandwidth[i,j]/total\_Outgoing[j]$.
- Line 2 : For this problem, we use execution time and number of search iterations as the $termination-criteria$.

Iteratively perform the following steps :

- Lines 3 – 4 : Choose a neighbor (the link whose weight will be modified in the next step) :
  1. Cost function : we use the cost function Φ of Fortz [2] to evaluate the cost of each link.
  2. Choose the link $a(s,t)$ that has the maximal value of cost $\Phi(a)$.

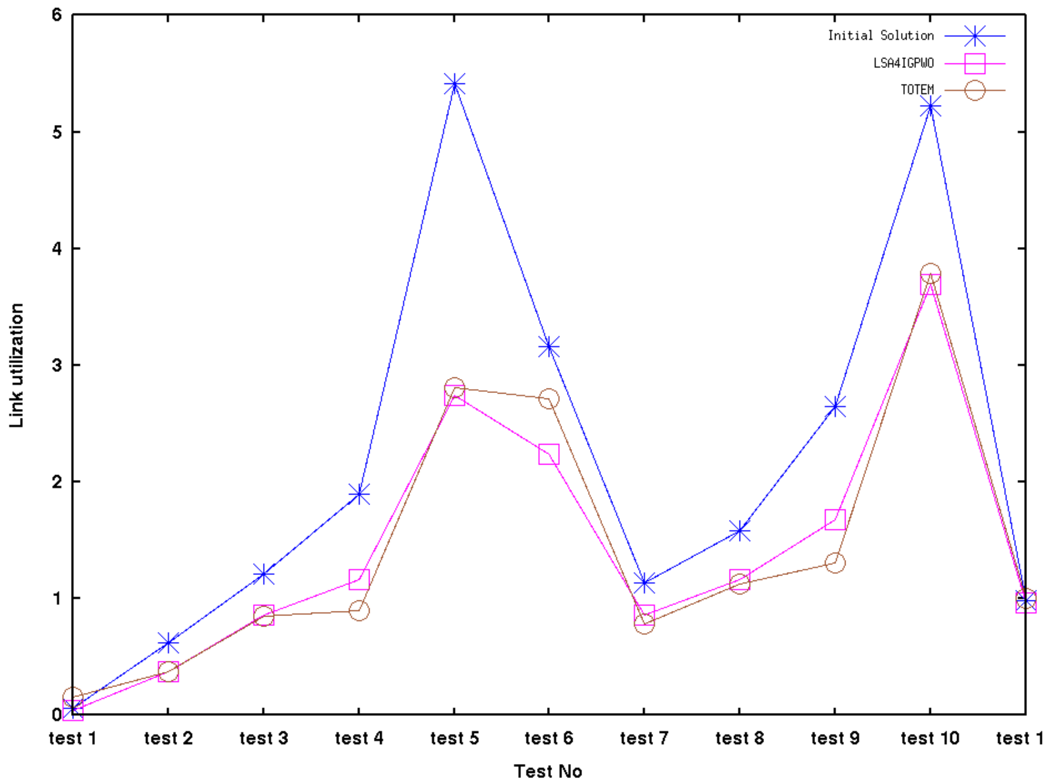

Fig. 1: Comparison of U_{max}

3. At this substep, we use the Randomized Selector of COMET to determine whether to increase or to decrease the weight of a link with the probability 0.9 for increase and 0.1 for decrease.
   (a) Increase: Suppose w_1 is the initial weight of (s,t) and a destination d that is reached via (s,t). We replace the weight of (s,t) with a random weight in the Reroute Metrics Sequence (RMS) for (s,t), as defined in [6]. The Key Metrics Sequence (KMS) for (s,t) is the sequence of increasing weights for (s,t) :$\{w_1, w_2, .., w_t\}$ so that each time we replace the weight of (s,t) with a different weight in KMS, it will force at least one router r to use an additional equal cost path not via (s,t) towards d. RMS is the sequence $\{w_1, w_1 + 1, w_2, w_2 + 1, .., w_t\}$.
   (b) Decrease: Considering a destination d that is reached via (s,t) by a router k and d has the maximal incoming traffic demand among the destinations that are reached via (s,t). In this case, by decreasing the cost of the second path from k to d that does not pass via (s,t), we split the traffic flow on several paths by using ECMP to lighten the load of (s,t).
4. We freeze the changed link during the next 10 search iterations.

- Lines 5-8 : Decide whether to store the current solution : we first consider the improvement of link utilization (U_{max} decreases). Secondly, we consider the improvement of LFA coverage (LFA coverage increases).
- Lines 9-11 : Diversification : we restart periodically the search process with one of the three initial configurations after each third of execution time or number of iterations.

3 Experiments and Results

In this research, we have performed 11 tests for LSA4IGPWO. We use two sources for topologies: the Abilene Network [4] and the network topologies (link bandwidth, IGP weights) and traffic demands generated by IGEN [3]. We use two different traffic demands for Abilene (test 1 and test 2). The rest of the tests (test 3 - test 11) come from iGEN. With iGEN we can easily generate the network topologies and the traffic demands for large networks. The largest considered network has 100 nodes and 406 links. iGEN is configured as follows. Each POP is similar to a Sprint POP [3].

Tab. 1: Table (Comparison of LFA Coverage and execution time)

Test No	Num. Links	LFA Coverage		Execution time (s)	
		LSA4IGPWO	TOTEM	LSA4IGPWO	TOTEM
1	30	16 (53%)	10 (33%)	40	26
2	30	14 (46%)	10 (33%)	52	26
3	80	80 (100%)	80 (100%)	24	24
4	122	122 (100%)	120 (98%)	24	24
5	164	164 (100%)	125 (76%)	59	28
6	202	202 (100%)	146 (72%)	41	62
7	202	202 (100%)	160 (79%)	55	72
8	280	280 (100%)	174 (62%)	227	261
9	322	322 (100%)	172 (53%)	280	406
10	360	360 (100%)	231 (64%)	507	619
11	406	406 (100%)	220 (54%)	491	957

The POPs are selected by using the k-medoids method and a Delaunay triangulation is used to create the backbone network. The IGP weights are set in function of the geographical distance. Finally, we use a uniform traffic matrix. Our topologies are available from URL: `http://inl.info.ucl.ac.be/softwares/tests-lsa4igpwo`.
We compare LSA4IGPWO with the tabu search of Fortz et Thorup which is implemented in C in the IGP Weight Optimization module of TOTEM. Figure 1 shows U_{max} for the different topologies and Table 1 provides the execution time on an Intel T2300 running at 1.66GHz with 1.5GBytes of RAM and the number of links protected by LFA. Regarding Figure 1, we observe that the two programs provide similar solutions from a maximum utilization viewpoint (U_{max}). Both of them found a significant decrease of the link utilization value U_{max}. From a performance viewpoint, LSA4IGPWO was slightly faster than TOTEM particularly for the large topologies (test 8 – test11).
When comparing LSA4IGPWO with TOTEM for the LFA coverage, regarding Table 1, we can generally state that LSA4IGPWO is better than TOTEM for most instances (10 out of 11 tests). Especially with 9 tests generated by IGEN, our solutions improve U_{max} and have a LFA coverage of 100%.

4 Conclusion

In this paper, we have first shown that COMET provides a flexible and efficient environment for solving a traffic engineering problem. It turns out that IGPWO can be performed on network configuration by ensuring a good LFA coverage, without sacrificing the optimality of network resource usage. Our further work is to apply COMET to other more complex traffic engineering problems.

5 Acknowledgement

Ho Trong Viet is supported by the FRIA (Fonds pour la formation à la Recherche dans l'Industrie et dans l'Agriculture, Belgium). P. Francois is supported by the FNRS (Fonds National de la Recherche Scientifique, Belgium).

6 REFERENCES

[1] Bernard Fortz, Jenifer Rexford et Mikkel Thorup. Traffic Engineering with Traditional IP Routing Protocols. IEEE Communication magazine, 2002.
[2] Bernard Fortz et Mikkel Thorup. Internet Traffic Engineering By Optimizing OSPF Weights. IEEE INFOCOM, 2000.
[3] Bruno Quoitin. IGEN. 2005. http://www.info.ucl.ac.be/ bqu/igen/.
[4] Internet2 Community. 1996. http://www.internet2.edu/network/.
[5] Pascal Van Hentenryck et Laurent Michel. Constraint-based Local Search. MIT Press, 2005.
[6] Pierre Francois, Mike Shand, Olivier Bonaventure. Disruption Free Topology Reconfiguration in OSPF Networks. IEEE INFOCOM 2007.
[7] Atlas, and al. IP Fast Reroute: Loop-Free Alternates. RFC5286, September 2008.
[8] S. Balon et G. Monfort. TOTEM-TOolbox for Traffic Engineering Methods. 2003. http://totem.run.montefiore.ulg.ac.be/.