A Multiple-objectives Evolutionary Perspective to Interdomain Traffic Engineering

Steve Uhlig*

Computer Science and Engineering Department
Université catholique de Louvain, Belgium
suh@info.ucl.ac.be

Abstract. We present an application of multiple-objectives evolutionary optimization to the problem of engineering the distribution of the interdomain traffic in the Internet. We show that this practical problem requires such a heuristic due to the potential conflicting nature of the traffic engineering objectives. Furthermore, having to work on the parameter's space of the real problem makes such techniques as evolutionary optimization very easy to use. We show the successful application of our algorithm to two important problems in interdomain traffic engineering.

1 Introduction

The Internet routing system today is divided into two views: intradomain and interdomain. The interdomain Internet is made of autonomous systems (AS). Each autonomous system uses the interdomain routing protocol (BGP) to exchange reachability information with its neighbor ASes. Autonomous systems are made of routers and links between routers that constitute the intradomain view of each AS. Routers in a given AS exchange intradomain routing information through an interior gateway protocol (IGP) that distributes the whole map of the intradomain network to all routers of the AS.
The current interdomain routing protocol used in the Internet is BGP, that stands for border gateway protocol [12]. With BGP, an AS advertises to each neighbor AS all the networks (IP prefixes) it can reach. Among the IP prefixes that an AS advertises, some are internal prefixes that are reachable within this AS (internal to this AS) and others are prefixes that have been learned through its BGP neighbors. A key feature of BGP is that it allows each network operator to define its routing policies. Those policies are implemented by using filters [8]. A BGP filter is a rule applied upon receiving a BGP route from a neighboring AS or before sending a BGP route to a neighboring AS. BGP filters can prevent some routes from being accepted from or announced to peer ASes, and can also modify the attributes of the BGP routes on a per-AS basis so that some routes be preferred over others.
Figure 1 shows a simplified Internet made of three ASes. Each AS has a particular intradomain topology the other ASes do not know about. Inside an AS, the intradomain routing protocol (IGP) distributes the whole map of the internal topology of the AS to the other routers of the AS so that each router of the AS

* This work was partially supported by the Waloon Government (DGTRE) within the TOTEM project.

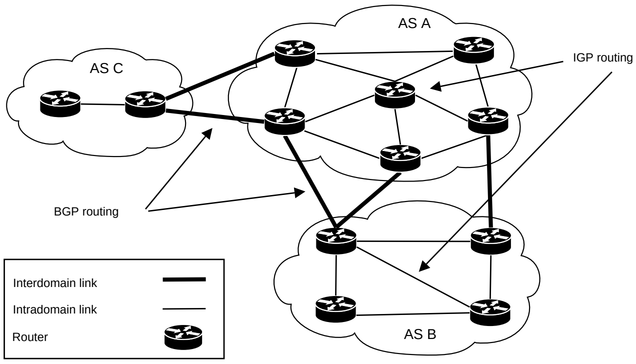

Fig. 1. Intradomain and interdomain views of the Internet

knows the shortest path to reach any other router of the AS. On Figure 1, AS A is directly connected to both AS B and AS C at the interdomain level, but AS B and AS C can only reach one another by crossing AS A. With the interdomain routing, neither AS B nor AS C knows the exact path followed by its traffic inside AS A. With BGP, an AS only knows the intermediate ASes crossed by its traffic to reach a destination AS. The path other ASes use to reach an AS is not known by the latter through BGP.

Nowadays, more and more Internet Service Providers (ISP) rely on traffic engineering to optimize the flow of the traffic inside their network [3]. While ISPs know their internal topology and techniques exist to tune the intradomain routing [7], most of them rely on manual tuning to do it. At the interdomain level, traffic engineering is even more challenging [11]. Operators change their routing policies and the attributes of their BGP routes on a manual basis, without a proper understanding of the implications of such changes on the flow of the traffic. Interdomain traffic engineering is important in practice for ISPs to be able to automatically engineer the flow of their traffic with neighboring ASes. Having to do it manually often may lead to router misconfigurations [9] that exacerbate the stability of interdomain routing.

In this paper, we present a multiple-objectives evolutionary algorithm especially designed to deal with interdomain traffic engineering with BGP and describe two successfull applications of this algorithm.

The remainder of the paper is structured as follows. Section 2 introduces the different objectives important in the context of interdomain traffic engineering. Section 3 discusses the choice of the optimization method. Section 4 describes our multiple-objectives evolutionary algorithm. Section 5 discusses the practical issues of sampling a non-dominated front. Then, section 6 provides two applications of our algorithm to problems in interdomain traffic engineering.

2 Problem statement

Interdomain traffic engineering consists in modifying the flow of the traffic exchanged with neighboring ASes. The desirable objectives to be dealt with when designing an interdomain traffic engineering technique encompass:

1. minimizing the burden on the interdomain routing protocol required to implement the traffic engineering,
2. optimizing one or several objectives defined on the traffic exchanged with other ASes or on the distribution of the traffic inside the AS.

The first objective concerns interdomain routing. Given that there are many remote networks with which an AS exchanges traffic on timescales of hours to days [13], an interdomain traffic engineering technique should ideally minimize the number of reachable networks that need to be influenced. As the number of influenced networks corresponds to the number of the BGP routing changes that will be implemented, any interdomain traffic engineering technique should try to minimize the burden placed on BGP.

The second objective deals explicitly with the flow of the traffic, as it consists of a set of objectives defined on the interdomain traffic. As different ASes have different engineering needs, the traffic engineering objectives that an AS may want to optimize will depend on its size and the type of business it focuses on. Small ASes typically pay providers to have Internet connectivity. The price for this connectivity can be large, and minimizing the cost of their traffic is thus relevant especially if they have multiple connections to the Internet. Larger ASes on the other hand do not have to pay providers but need to carefully distribute the load of their traffic inside their network. For that purpose, one way is to tune their intradomain routing [7]. However, tuning the intradomain routing not only changes the distribution of the flow of the traffic inside the AS, but also their traffic demand, i.e. how traffic enters and leaves the network [1]. To control how traffic enters and leaves the network, large ISPs need to tweak the BGP routing. Large providers also often rely on "hot-potato routing", that consist in using the exit point inside the network that is closest to the ingress point where the traffic has been received. Hot-potato routing however does not lead to a balance distribution of the traffic among the exit points, so that traffic engineering objectives are often conflicting in practice.

In the context of interdomain traffic engineering, the problem of optimizing any traffic objective is always conflicting with the objective of minimizing the impact on BGP, as changing the flow of the traffic always requires to tweak BGP routing. Furthermore, traffic engineering objectives that are only concerned with the traffic can also be conflicting between one another. This is why a multiple-objectives algorithm is necessary, to sample the trade-offs among the possible solutions to the interdomain traffic engineering problem.

In the remainder of this paper, we distinguish between the *traffic objectives* that are purely concerned with the traffic and the *BGP routing objective* that is only concerned with the changes made to the BGP routing.

3 Motivations for evolutionary optimization

The traffic engineering objectives discussed in the previous section cannot be compared, i.e. an improvement in one of the objectives cannot be measured against an improvement in another objective. Optimizing a single composite objective that weights all these objectives is thus useless for practical purposes as a network operator would like to have the best solution in terms of all the objectives at the same time. The interdomain traffic engineering problem is thus intrinsically a multiple-objectives optimization problem. For such problems, evolutionary algorithms are a well-known technique capable to find a non-dominated front in a single run [4,5]. Recall that a *front* is a set of solutions and that a solution is said *non-dominated* if no other solution of the set is better in terms of all the considered objectives at the same time. Additionally, relying on the "evolutionary" paradigm allows to leverage the mechanisms of population-based search and selection among individuals.

The main reasons that motivate our choice of the evolutionary paradigm to tackle our problem are the following. The first reason is interdomain routing. Our aim is to be as close as possible to the way BGP works in practice. Thus, we do not want to simplify the way BGP chooses the best route towards a particular destination because it is most critical for practical interdomain traffic engineering. The complexity of BGP makes it very difficult to model. The second reason is that the traffic objectives need not be linear, convex, piecewise convex,... Interdomain traffic engineering objectives can be complex, non-linear, based on statistics,... Hence we consider that having to rely on strict assumptions concerning the traffic objectives would be too limiting.

4 Search procedure

Depending on the relationships between the traffic objectives which might be conflicting, harmonious or neutral [10], the search on the non-dominated front should have to be different. Recall that we do not know beforehand the relationship between the traffic objectives. This means that our search method must be as lightly biased as possible towards any of the traffic objectives to sample in the best possible manner the search space. Because sampling the whole search space would make the search space grow very large, we decided that the heuristic would iterate over the BGP routing changes by trying to add one BGP routing change at each generation of the algorithm. Doing this puts additional pressure on the population by forcing improvements in the traffic engineering objectives to have as few BGP route changes as possible early on during the optimization.

Figure 2 provides a pseudo-code description of the search procedure. The principle of the search is as follows. At the first generation, we start with a population of individuals initialized at the default solution found by BGP routing. Hence at generation zero all individuals have the same values of the traffic objectives and contain no BGP routing change. At each generation, we use a random local search aimed at improving the current population by applying an additional BGP routing change. Each individual of the population is non-dominated with respect to the other members of the population for what concerns the traffic objectives. In addition, the current population is always made of individuals having the same number of BGP routing changes. At each generation, we parse the whole population and for each individual we try to apply an additional randomly chosen BGP routing change. Whenever a BGP routing change provides improvement with respect to at least one of the traffic objectives, we accept this improved individual and put it in the set of accepted individuals. We iterate this procedure until we find a target number of improved individuals or stop when we have performed a target number of tries (the variable `iter`). Note that the pseudo-code given at Figure 2 concerns only one generation, and that the purpose of variable `iter` is not to count the generations but to ensure that the search will not loop indefinitely during the current generation.

```
accepted = 0
iter = 0
while ((accepted < MAXPOP) AND (iter == MAXITER)){
  foreach individual k {
      // trying a random BGP route change
      filter.prefix = rand_int_uniform(1,MAXPOP)
      filter.exit = rand_int_uniform(1,NUM_EXIT_POINTS)
      // if effect of filter is improvement accept it
      if (improved(k,filter)){
          accept(k,filter)
          // update counter for accepted improved individuals
          accepted++
      } // end if
  } // end foreach individual
  // update iteration counter
  iter++
} // end while
```

Fig. 2. Pseudo-code of search procedure for a single generation.

5 Sampling the non-dominated front

The previous section described the procedure to search for BGP routings changes that improve the individuals of the previous population with respect to any of the traffic objectives. These improved individuals however are not non-dominated. Some of them can be dominated since we did not check for non-domination when accepting an improved individual. Improvement was sufficient to accept an individual. The next step is to check for non-domination on this population of improved individuals to obtain a non-dominated front. For that purpose, we rely on the fast non-domination check procedure introduced in [6]. This procedure has time complexity $0(MN^2)$ where M is the number of objectives and N the size of the population. We do not describe this procedure in details but refer to [6] for the original idea and to [5] for a detailed explanation. Let us only mention the main points here. Let P denote the set of non-dominated individuals found so far at the current generation. P is initialized with anyone of the individuals among the accepted ones. Then try to add individuals from the set of accepted ones one at a time in the following way:

- temporarily add individual k to P
- compare k with all other individuals p of P:
  - if k dominates any individual p, delete p from P
  - else if k is dominated by other members of P remove k from P

This procedure ensures that only non-dominated individuals are left in P. The number of domination checks is in the order of $0(N^2)$ while for each domination check M comparisons are necessary (one for each objective). The time complexity is thus $0(MN^2)$.

Having found the non-dominated front for a given number of BGP routing changes, we are left with selecting the individuals of the population for the next generation. Actually, the number of non-dominated individuals from the set of improved ones is due to be smaller than the size of the population we use during the search process (MAXPOP). To constitute the population for the next generation, we have to decide how many individuals in the next population each non-dominated solution will produce. Because non-dominated individuals are not comparable between one another, we must choose a criterion that will produce MAXPOP individuals from the set of non-dominated ones. On the one hand, we would like to include at least every non-dominated individual in the population. On the other hand, depending on the way the accepted solutions are spread over the non-dominated front, we must sample differently different regions of the front for a given number of BGP routing changes. This notion of sampling the non-dominated front is close to an idea of distance between neighboring individuals in the objective space. Maintaining diversity on the non-dominated front requires that individuals whose neighbors are farther apart be preferred over non-dominated individuals whose neighbors are close. The rationale behind this is that less crowded regions should require more individuals to be correctly explored than regions having more non-dominated individuals. The computation of the crowding distance for each individual is done according to [5] pp. 248. First the non-dominated individuals are sorted according to each objective. Then the individuals having the smallest and largest value for any objective are given a crowding distance d^m of ∞ to ensure that they will be selected in the population. For each objective m, the crowding distance of any individual i, $1 \leq i \leq (|P| - 2)$, is given by

$$d_i^m = \left| \frac{f_{i+1}^m - f_{i-1}^m}{f_{max}^m - f_{min}^m} \right| \tag{1}$$

where f_i^m denotes the value of individual i for objective m, f_{max}^m (respectively f_{min}^m) denotes the maximum (respectively minimum) of the objective value m among individuals of the set P of non-dominated individuals. The global crowding distance for all objectives is the sum of the crowding distance for all objectives. For our two objectives, this crowding distance represents half the perimeter of the box in which individual i is enclosed by its direct neighbors in the objective space.

6 Simulations

In this section we use the previously described algorithm to two practically relevant instances of interdomain traffic engineering.

6.1 Outbound interdomain traffic engineering for a stub AS

Most of the ASes in the Internet do not provide transit service, i.e. either the source or the destination of the traffic is located inside their network. These ASes are called *stubs*. As more than half of the stub ASes have several connections to the Internet [2], these stubs may want to evenly distribute the load of the traffic among their Internet links. As stub ASes must pay for their Internet connection, the economical cost of these connections can become significant for the AS. However, the way providers bill stub ASes for their traffic often depends on different timescales. Most billing schemes rely on the following procedure: 1) collect samples of the traffic volume every t minutes (5 and 15 minutes are common); 2) combine these t minutes samples into one combined sample; 3) at the end of a billing cycle, compute the 95^{th} (or another) percentile of the combined samples; 4) this number corresponds to the bandwidth L which will be used for the price. The most common billing schemes in use today by ISPs are the following:

- percentile-based : x \$ per y Mbps (n^{th} percentile) with a commitment of c Mbps. The price per Mbps can be different for the commitment and for the traffic above the commitment (also called "burstable").
- average-based : same as previous but using an average instead of a percentile.
- volume-based : x \$ per y bytes.
- destination-based : x \$ per Mbps for "local" traffic (national for instance) and y \$ per Mbps for "non-local" traffic (international for instance).
- max-based : flat rate based on the maximum available bandwidth, independent of how many bits are used.

The actual billing cost of the traffic hence depends both on the short-term traffic dynamics on each Internet connection and the long-term traffic volume exchanged with providers. We thus evaluate in this section the problem of optimizing the cost of the traffic of a stub AS while balancing the short-term (10 minutes intervals) load of the traffic over the available providers, with as few BGP routing changes as possible.

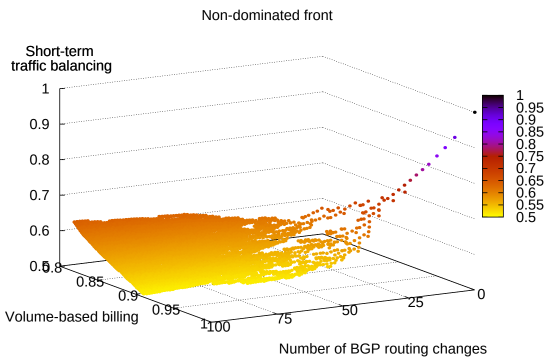

Fig. 3. Daily volume-based billing and short-term traffic balancing.

On Figure 3, we plot the non-dominated front found by the algorithm for a scenario of a stub AS having Internet connections with three different providers. On Figure 3, the stub AS tries to minimize the daily cost of its total traffic while evenly balancing the traffic over its three providers over 10 minutes time intervals. The grayscale palette located at the right of Figure 3 maps the z-value of the points to some color to ease the interpretation of the 3D plots. The point corresponding to the default BGP routing (upper right) has no BGP routing change and value of the two traffic objectives equal to 1 as these objectives were normalized with respect to their value under default BGP routing (no BGP routing change). Globally, two regions appear on Figure 3. The first region concerns point for the first few BGP routing changes (about 20). These points start at the top right of Figure 3 (default BGP solution) and converge to the front which constitutes the second region of the non-dominated front (bottom left). The second region of the non-dominated front indicates that the two traffic objectives are conflicting for more than 20 BGP routing changes. The conflicting nature of the objectives can

be seen by a relatively linear (slightly convex) trade-off between the two traffic objectives, for a given number of BGP routing changes. Finding a solution providing a smaller cost on the long-term for a given number of BGP routing changes requires to worsen the short-term objective value. In the same way, finding a solution providing a smaller value of the short-term objective function for a given number of BGP routing changes requires that one worsens the value of the long-term objective function.
Volume-based billing as used above is not the most realistic traffic billing scheme one can think of. Now, we use as the long-term traffic objective the 95^{th} percentile billing over 10 minutes time intervals. For the short-term traffic objective, we use the same traffic balancing objective as above. The non-dominated front for

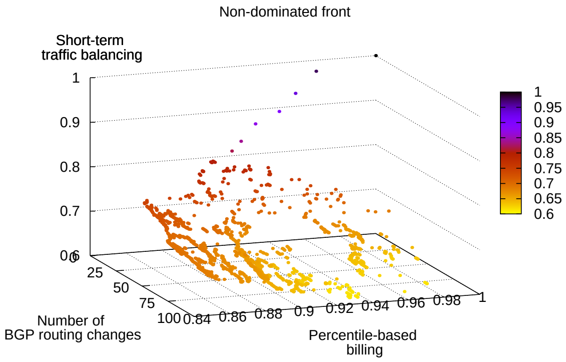

Fig. 4. Daily percentile-based billing and short-term traffic balancing.

the long-term percentile-based traffic objective is provided on Figure 4. Figure 4 shows no smooth non-dominated front even for a large number of BGP routing changes, in contrast to the results of the traffic cost objective above. The explanation for this phenomenon is the statistical nature of the percentile-based objective which largely depends on the short-term dynamics of the traffic. Indeed, the value of the 95^{th} percentile depends on the distribution of the values of the traffic for each provider and each short-term time interval. Changing the provider used to carry the traffic for some reachable network over the whole day has a non-trivial effect on the value of the percentile. A cost function as volume-based billing is insensitive to the short-term variability for some reachable network, in contrast to the percentile-based objective. A percentile-based cost function thus appears as a relatively difficult long-term traffic objective to optimize.

6.2 Outbound interdomain traffic engineering for a transit AS

A very different interdomain traffic engineering problem is the one of transit ASes. Contrary to stub ASes, transit ASes receive traffic at some ingress point of their network and forward it to another AS through some egress point of their network. In that case, not only is the balance of the traffic among the egress points important, but also the cost for the traffic to cross the internal topology of the transit AS. In this section, we show the results of a simulation where we optimized the balance of the outgoing traffic over the Internet connections of a transit AS while minimizing the cost of the traffic to cross its network (IGP cost), by relying on as few BGP routing changes as possible.

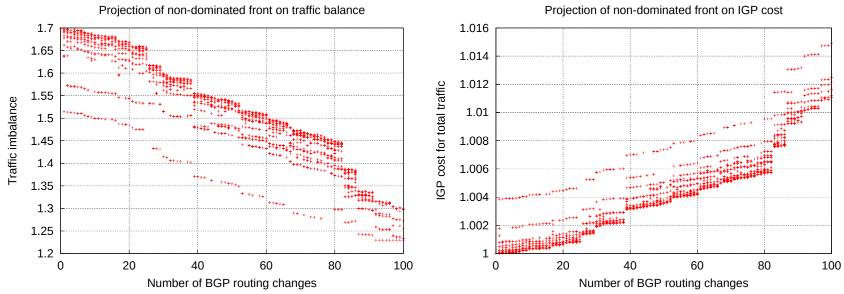

Fig. 5. Outbound traffic balancing and IGP cost minimization.

The two parts of Figure 5 provide the projection of the non-dominated front found by the algorithm on the two traffic objectives: traffic balance over the Internet connections (left of Figure 5) and IGP cost (right of Figure 5). The left part of Figure 5 shows that for the particular scenario we used, the default traffic imbalance among the Internet connections is of about 1.7 (O BGP routing change). By traffic imbalance, we mean the maximum amount of traffic carried through an Internet connection divided by the average traffic carried by all Internet connections. This shows that by default the interdomain routing protocol does not balance the outbound traffic of a transit AS well, hence interdomain traffic engineering is desirable for such networks.

The traffic optimization starts at 0 BGP routing change, with a traffic imbalance of about 1.7 and an IGP cost of 1. We normalized the IGP cost so that the sum of the amount of traffic multiplied by its IGP cost to cross the network under default BGP routing adds to one. Figure 5 then shows that adding BGP routing changes is able to improve the traffic balance but this also increases the cost of the traffic to cross the network. On the simulations of Figure 5, the algorithm is able to improve the traffic balance while not increasing very much the IGP cost. This is possible because in our simulation the initial solution is closest possible from the optimal traffic distribution in terms of the IGP cost, while very far from optimal in terms of the traffic balance. The algorithm hence does not have too much trouble to find BGP routing changes that improve the traffic balance while not increasing too much the IGP cost. The graphs of Figure 5 however show discontinuities in the non-dominated front, indicating that the solutions do not form a well-spread surface. This not well-looking non-dominated front might be either due to the nature of the objectives of to the considered problem. This asks for further work to improve the sampling of the non-dominated front.

7 Conclusions

In this paper we have presented an application of multiple-objectives evolutionary optimization to interdomain traffic engineering in the Internet. We have shown that the problem is intrinsically a multiple-objectives one where the different objectives cannot be compared to one another. The potentially conflicting nature of some of the objectives also make evolutionary-based heuristics suited to the problem. We have then presented the successful application of our algorithm on two instances of interdomain traffic engineering in the Internet. The first problem instance we tackled was of minimizing the daily billing cost of the outbound traffic of a stub AS while evenly balancing the outbound traffic over its Internet connections on the short-term. The second problem instance consisted in balancing the outbound traffic of a transit AS over its Internet connections while minimizing the cost of the traffic to cross its internal topology.

References

1. S. Agarwal, C. Chuah, S. Bhattacharyya, and C. Diot. The Impact of BGP Dynamics on Intra-Domain Traffic. In *Proc. of ACM SIGMETRICS*, June 2004.
2. S. Agarwal, C. Chuah, and R. Katz. OPCA: Robust Interdomain Policy Routing and Traffic Control. In *Proc. of IEEE OPENARCH*, April 2003.
3. D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao. Overview and Principles of Internet Traffic Engineering. Internet Engineering Task Force, RFC3272, May 2002.
4. C. A. Coello Coello. An updated survey of GA-based multiobjective optimization techniques. *ACM Computing Surveys*, 32(2):109–143, 2000.
5. K. Deb. *Multi-objective Optimization using Evolutionary Algorithms*. Wiley Interscience series in systems and optimization, 2001.
6. K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In *Proc. of PPSN'00*, 2000.
7. B. Fortz, J. Rexford, and M. Thorup. Traffic engineering with traditional IP routing protocols. *IEEE Communications Magazine*, October 2002.
8. B. Halabi. *Internet Routing Architectures*. Cisco Press, 1997.
9. R. Mahajan, D. Wetherall, and T. Anderson. Understanding BGP Misconfigurations. In *Proc. ACM SIGCOMM '02*, September 2002.
10. R. Purshouse and P. Fleming. Conflict, Harmony, and Independence: Relationships in Multi-criterion Optimisation. In *Proc. of EMO2003*, April 2003.
11. B. Quoitin, S. Uhlig, C. Pelsser, L. Swinnen, and O. Bonaventure. Interdomain traffic engineering with BGP. *IEEE Communications Magazine, Internet Technology Series*, May 2003.
12. J. Stewart. *BGP4 : interdomain routing in the Internet*. Addison Wesley, 1999.
13. S. Uhlig, V. Magnin, O. Bonaventure, C. Rapier, and L. Deri. Implications of the topological properties of internet traffic on traffic engineering. In *Proc. of ACM SAC'04*, March 2004.