International Journal of Computational Intelligence and Applications

A Multiple-objectives Evolutionary Perspective to Interdomain Traffic Engineering

Steve Uhlig*

Department of Computing Sciences and Engineering
Université catholique de Louvain, Louvain-la-neuve, 1348, Belgium
URL: `http://www.info.ucl.ac.be/~suh`
E-mail: suh@info.ucl.ac.be

Received (received date)
Revised (revised date)

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 potentially conflicting nature of the traffic engineering objectives. Furthermore, having to work on the parameter's space (BGP routing) of the real problem makes such techniques as evolutionary optimization very easy to use. We show the successful application of our algorithm to two practically relevant problems in interdomain traffic engineering.

Keywords: multiple-objectives evolutionary optimization; interdomain routing; 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 ASs. Autonomous systems are made of routers and links between routers that constitute the intradomain view of each AS. A router's purpose is to forward traffi c toward a destination in the Internet. 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. Typically, IGP routers know the whole path to reach any other host inside the AS. BGP routers on the other hand only know the next hop to reach a destination in the Internet.

The current interdomain routing protocol used in the Internet is BGP, that stands for border gateway protocol[13]. With BGP, an AS advertises to each neighbor AS all the destination networks (IP prefi xes) it can reach. Among the IP prefi xes that an AS advertises, some are internal prefi xes that are reachable within this AS (internal to this AS) and others are prefi xes that have been learned through its BGP neighbors. A key feature of BGP is

*Steve Uhlig is funded by the FNRS (Fonds National de la Recherche Scientifique, Belgium). This work was partially supported by the Waloon Government (DGTRE) within the TOTEM project.

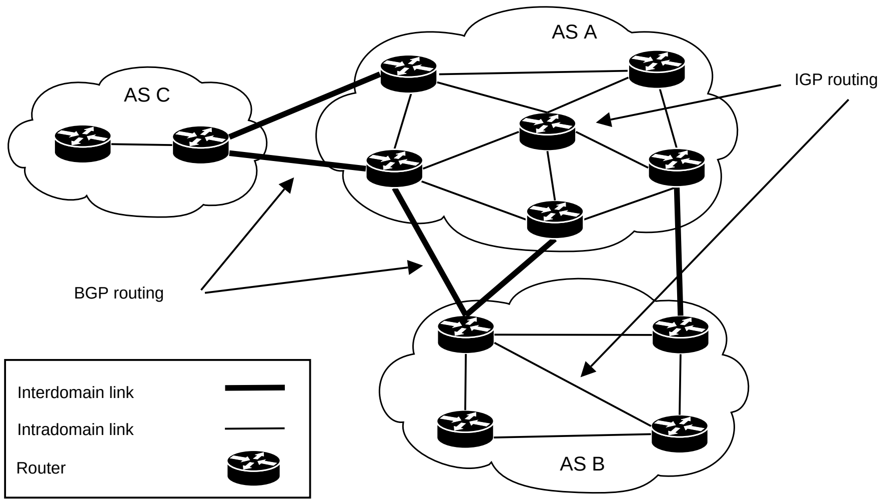

Fig. 1. Intradomain and interdomain views of the Internet

that it allows each network operator to define its routing policies. Those policies are implemented by using "filters"[9]. 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 ASs, and can also modify the attributes of the BGP routes on a per-AS basis so that some routes be preferred over others.

Fig. 1 shows a simplified Internet made of three ASs. Each AS has a particular intradomain topology the other ASs 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 knows the shortest path to reach any other router of the AS. On Fig. 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 each other 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 ASs crossed by its traffic to reach a destination AS.

Nowadays, more and more Internet Service Providers (ISP) rely on traffic engineering to modify the flow of the traffic inside their network[3]. Traffic engineering encompasses all techniques aimed at modifying the characteristics of the traffic, be it to change the load of the traffic among the network elements or to influence the very characteristics of the traffic. In practice, different traffic classes have different engineering requirements, so that traffic engineering techniques depend on the particular traffic class considered. In this paper, we restrict the focus on the engineering of best-effort traffic, for which no strict guarantees need to be enforced. While ISPs know their internal topology and techniques exist to tune the intradomain routing[8], most of them rely on manual configuration. At the interdomain level, traffic engineering is even more challenging[12]. 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 used by ISPs to automatically engineer the flow of their traffic with neighboring ASs. Having to do it manually may often lead to router misconfigurations[10] 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 successful applications of this algorithm.

The remainder of the paper is structured as follows. Section 2 introduces the main objectives 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 as considered in this paper consists in modifying the flow of the traffic exchanged with neighboring ASs. The objectives are the following:

(1) minimize the burden on the interdomain routing protocol required to implement the traffic engineering,
(2) optimize one or several objectives defined on the traffic exchanged with other ASs or on the distribution of the traffic inside the AS.

The first objective concerns interdomain routing. There are many remote networks with which an AS exchanges traffic on timescales of hours to days[14,16]. 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, an 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 ASs 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 ASs typically pay providers for their Internet connectivity. The price of this connectivity can be high, and minimizing the cost of their traffic is thus relevant especially if they have multiple connections to the Internet. Large ASs on the other hand do not have to pay providers but need to carefully distribute the load of the traffic inside their network. For that purpose, one way is to tune their intradomain routing[8]. However, tuning the intradomain routing not only changes the distribution of the flow of the traffic inside the AS, but also 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 consists in using the exit point inside the network that is closest in terms of the IGP routing metric to the ingress point where the traffic has been received. Hot-potato routing however does not lead to a balanced distribution of the traffic among the exit points, so that traffic engineering objectives can be 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. A multiple-objectives algorithm is then 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.

BGP tweaking basics ASs can be roughly classified in two types: transit or stub. *Stub* ASs contain hosts that produce or consume network traffic. These domains do not carry traffic that is not produced by or destined to their hosts. *Transit* ASs interconnect different ASs together and carry traffic that is produced by and/or destined to external ASs. Tweaking BGP to modify the flow of the outgoing traffic requires knowledge of how BGP decides which route to use to reach a destination[12]. For the sake of simplicity, we only describe in this section how to tweak BGP routes in the case of stub ASs and for outgoing traffic. For further details about BGP tweaking see[12].

Assume a stub AS having one link per provider. It receives a BGP routing table from each of its providers. To control the flow of its interdomain traffic, it can rely on the information found in these BGP routing tables. BGP is a path vector routing protocol. Each BGP router sends BGP advertisements to its peers. A BGP advertisement sent by an AS means that the AS that advertises the route agrees to forward IP packets to the destinations corresponding to this route. In addition, the `AS-path` attribute[13] contained in this route advertisement also tells through which ASs the IP packets will transit to attain their destination. This information allows the stub AS to reconstruct the AS-level topology for outgoing traffic by relying on these BGP routing tables.

Fig. 2 provides an example topology with a stub AS connected to three providers. Each provider advertises one route towards each of the three considered destinations. In this figure, an arrow from AS X to AS Y indicates that the pattern “X Y” appears in the AS path for some destination. In addition, we use three different line styles to identify each provider’s routes, so that if part of the AS path of some routes that our AS received through several providers are identical, then we shall have as many arrows as there are routes from these different providers. On Fig. 2, the links between two ASs learned through routes advertised by provider 1 are represented by continuous arrows, while the ones from provider 2 are represented with medium dashed arrows and those from provider 3 with fine dashed arrows.

With the BGP routes received from each provider, our AS knows through which providers it can reach a particular destination. For the example provided on Fig. 2, all three destinations can be reached through any of the three providers. Assume that each of these three destinations represents one third of the outgoing traffic for our AS. We could

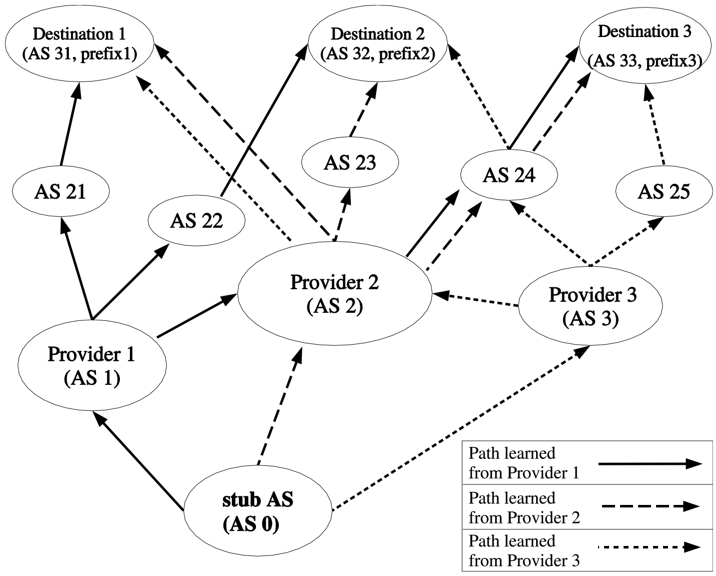

Fig. 2. Example AS-level topology.

choose to use one different provider for each destination so that its outgoing traffic is well balanced.

Let us assume for the sake of simplicity that there is only one next hop per provider although this is not always the case in practice. If the stub AS shown on Fig. 2 wants to achieve a good balance of its outgoing traffic, it can rely on the `local-pref` attribute to prefer some routes over others. Recall that we assume that each destination prefix receives one third of the traffic. Suppose that the restricted BGP decision process with only `local-pref` and the AS path length is considered. If the routes towards the three destinations have a default `local-pref` value of 100, then given the fact that the shortest AS path route would be chosen by BGP to reach a destination, the traffic towards destination 1 will be forwarded through provider 2 (AS path length of 2), the traffic towards destination 2 will be forwarded through providers 1, 2 or 3 (AS path length of 3), while the traffic towards destination 3 through providers 2 or 3 (AS path length of 3). This would mean that the routes for destinations 2 and 3 are non-deterministic *for this restricted BGP decision process* since there are several "best routes" among which BGP can choose. In practice, the BGP decision process ensures that only one route is used but this would not automatically lead to a good balance of the traffic. So a solution for evenly distributing the outbound traffic of the local AS is to put a higher value of the `local-pref` attribute in one of the routes learned from destinations 2 and 3 to ensure that only one best route remains and that the traffic balance be good among the three providers. For that purpose, our stub can attach a `local-pref` value of 110 to the route towards destination 2 learned from provider 1 and also attach a `local-pref` value of 110 to the route towards destination 3 learned from provider 3. This configuration would ensure that each provider gets one third of the outgoing traffic if the traffic is evenly distributed over the three prefixes.

Due to the way the BGP decision process chooses the best route towards a destination, there is ample choice concerning how to tweak the BGP decision process. In this paper we decided to tweak the value of the `local-pref` attribute only, the effect of a BGP route change is quite simple: a "BGP route change" is a pair <*prefix,provider*> that indicates that the traffic having the prefix *prefix* as destination will be forwarded through provider *provider* among the providers which advertised a BGP route towards this network. What we call a BGP route change in the remainder of this paper hence concerns a single BGP prefix. The actual effect of a BGP route change on the BGP routes is to force the value of the `local-pref` attribute of the BGP route towards prefix *prefix* which was learned via provider *provider* to be set with a value of 110 (higher preference). All other routes for prefix *prefix* (learned from other providers) have their `local-pref` attribute (re)set to the same value which will be strictly smaller than 110. Because our focus in this paper is on tweaking BGP in the most controllable way, we rely on changing the `local-pref` attribute. Working on the `local-pref` attribute ensures that the changes performed by the traffic engineering will overrule other aspects of the network configuration. Note that the MED attribute of BGP routes can also be used to control the outgoing traffic of stub ASs[15].

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 of finding a non-dominated front in a single run[4,5]. A *front* is a set of solutions. 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. More details about Pareto-optimality can be found below.

Now let us describe the main motivations for selecting the evolutionary paradigm to tackle our problem. 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 as it is most critical for practical interdomain traffic engineering. The complexity of BGP makes it very difficult to model[7]. The second reason is that interdomain traffic engineering objectives can be non-linear, based on statistics,… Hence we consider that having to rely on strict assumptions concerning the traffic objectives would be too limiting.

Pareto-optimality The concept of Pareto-optimality is related to the set of solutions whose components cannot be improved in terms of one objective without getting worse in at least one of the other components. More formally, a multiobjective search space is partially ordered in the sense that two solutions are related to each other in two possible ways : either one dominates or neither dominates. Consider the multiobjective minimization problem:

$$\begin{array}{ll} Minimize & y = f(x) = (f_1(x), ..., f_n(x)) \\ where & x = (x_1, ..., x_m) \in X \\ & y = (y_1, ..., y_n) \in Y \end{array}$$

and where x is called the decision vector, y the objective vector, X the parameter space and Y the objective space.

A decision vector $x_1 \in X$ is said to dominate another decision vector $x_2 \in X$ ($x_1 \prec x_2$), iff

$$\forall i \in 1, .., n : f_i(x_1) \leq f_i(x_2) \quad \wedge$$
$$\exists j \in 1, .., n : f_j(x_1) < f_j(x_2).$$

Domination is an important notion because it determines the result of the comparison of two decision vectors. A decision vector x is said Pareto-optimal *iff* x is non-dominated regarding X, i.e. $\nexists x' \in X : x' \prec x$.

A Pareto-optimal decision vector cannot be improved in any objective without degrading at least one of the other objectives. These are global optimal points. In our context however, we are not interested in global optima but optimal points in some neighborhood of some of the objectives. More precisely, we aim at finding the Pareto-optimal points with respect to the traffic objectives and having a distance of at most ϵ BGP configuration changes compared to this default BGP routing solution. Hence we do not search for globally Pareto-optimal decision vectors but locally Pareto-optimal decision vectors:

Consider a set of decision vectors $X' \subseteq X$.

1. The set X' is denoted as a local Pareto-optimal set *iff*
$\forall x' \in X' : \nexists x \in X : x \prec x' \wedge ||x - x'|| < \epsilon \wedge ||f(x) - f(x')|| < \delta$
where $||.||$ denotes a distance metric, $\epsilon > 0$ and $\delta > 0$.
2. The set X' is called a global Pareto-optimal set *iff*
$\forall x' \in X' : \nexists x \in X : x \prec x'$.

Note that a global Pareto-optimal set does not necessarily contain all Pareto-optimal decision vectors.

4. Search procedure

Depending on the relationships between the traffic objectives which might be conflicting, harmonious or neutral[11], 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.

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

Fig. 3. Pseudo-code of search procedure for a single generation.

Fig. 3 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 Fig. 3 concerns only one generation. The purpose of variable `iter` is not to count the generations but to ensure that the search will not loop indefinitely during a given 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. Some of these improved individuals 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[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 the original NSGA-II paper[6] for the original idea and to Deb's book[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 has to be smaller than the size of the population we use during the search process (MAXPOP), unless the front is almost continuous and easy to sample. To build 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 among themselves, 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 the procedure described in Deb's book[5] pp. 248. First the non-dominated individuals are sorted with respect 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 simply the sum of the crowding distances over each objective. 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 solve two practical problems of interdomain traffi c engineering.

6.1. *Outbound interdomain traffic engineering for a stub AS*

Most of the ASs in the Internet do not provide transit service, i.e. either the source or the destination of the traffi c is located inside their network. These ASs are called *stubs*. As more than half of the stub ASs have several connections to the Internet [2], these stubs may want to evenly distribute the load of the traffi c among their Internet links. As stub ASs must pay for their Internet connection, the economical cost of these connections can become signifi cant for the AS. However, the way providers bill stub ASs for their traffi c often depends on different timescales. In order to have an idea of the pricing schemes currently used for billing the traffi c, we asked how providers are billed for interdomain capacity on the NANOG mailing list (`nanog@merit.edu`, the North American Network Operators' Group). Here we provide a summary of the answers we received.

Most billing schemes rely on the following procedure:

(1) collect samples of the traffi c 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.

Then, once the statistics have been computed, ISPs use one of the following billing schemes:

- 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 traffi c 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” traffi c (national for instance) and y \$ per Mbps for “non-local” traffi c (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 traffi c hence depends both on the short-term traffi c dynamics on each Internet connection and the long-term traffi c volume exchanged with providers. We thus evaluate in this section the problem of optimizing the cost of the traffi c of a stub AS while balancing the short-term (10 minutes intervals) load of the traffi c over the available providers, with as few BGP routing changes as possible. Note that in this section traffi c balancing objectives (short-term or long-term) are measured in terms of the maximum amount of traffi c carried by any provider over the considered time intervals.

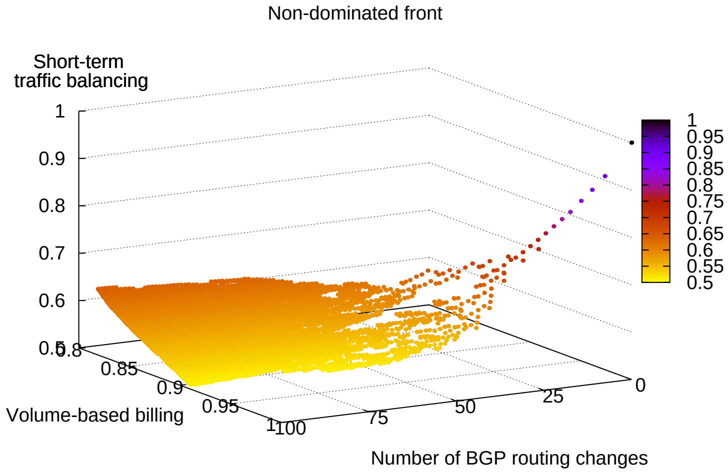

Fig. 4. Daily volume-based billing and short-term traffic balancing.

On Fig. 4, 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 Fig. 4, the stub AS tries to minimize the daily cost of its total traffi c while evenly balancing the traffi c over its three providers over 10 minutes time intervals. The grayscale palette located at the right of Fig. 4 and Fig. 5 maps the z-value of the points to some tone to ease the interpretation of the plots. The point corresponding to the default BGP routing (upper right of Fig. 4) has no BGP routing change, its values of the two traffi c objectives equal to 1 as we normalized the traffi c objectives with respect to their value under no BGP routing change. Globally, two regions appear on Fig. 4. The fi rst region concerns points for the fi rst few BGP routing changes (about 20). These points start at the top right of Fig. 4 and converge to the front that makes the second region of the non-dominated front (bottom left). The second region of the 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.

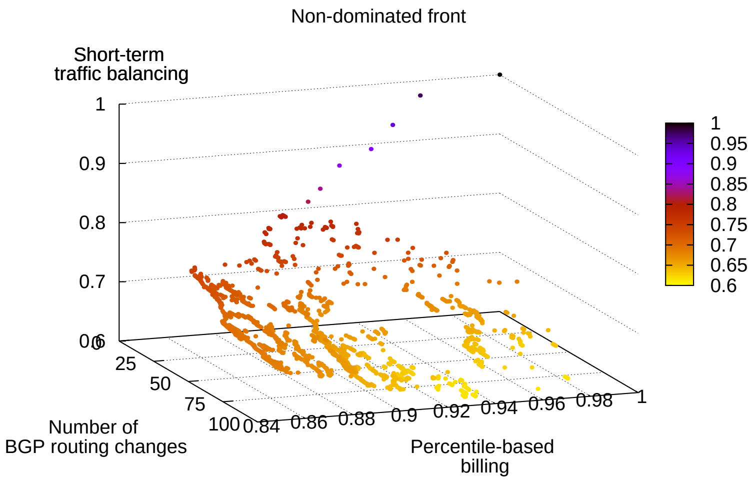

Fig. 5. Daily percentile-based billing and short-term traffic balancing.

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 the long-term percentile-based traffic objective is provided on Fig. 5. Fig. 5 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 like 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.

The last scenario we evaluate in this section is the same as the previous concerning the chosen providers, but the long-term objective is traffic balancing. Traffic balancing for the two traffic objectives seems intuitively harmonious, in that improving the long-term balance should improve the short-term one on average. In the same way, one might think that improving the short-term balance is likely to improve the long-term balance. In practice, the validity of the previous intuitions depends to a large extent on the dynamics of the traffic over the short term. Fig. 6 provides the non-dominated front for the two traffic

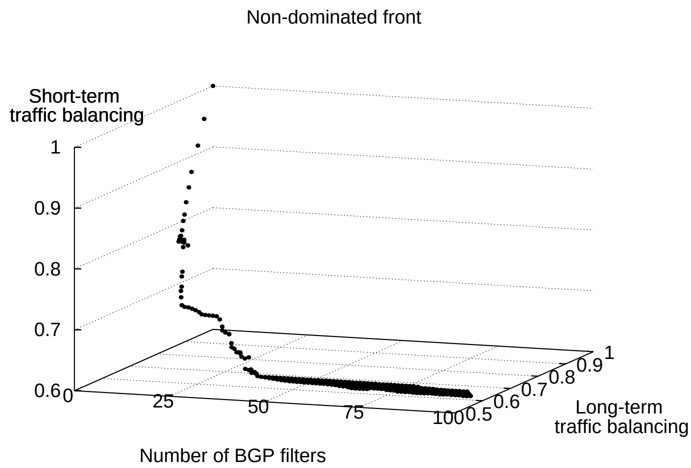

Fig. 6. Long-term and short-term traffic balancing objectives.

balancing objectives and confirms our intuition that indeed the long-term and short-term load-balancing objectives are harmonious. Fig. 6 shows that very few points found by the search are non-dominated, meaning that at each generation very few solutions dominate all the other. Once again, a large gain is achieved through the first 20 BGP routing changes. The next 20 BGP routing changes then still provide a non-negligible gain in terms of both traffic objectives but after 40 BGP routing changes the search improves very slowly.

In this section, we limited the problem to static optimization of a stub AS outgoing traffic, i.e. the optimization does not care about re-optimizing the BGP routing over time. We dealt with the dynamic optimization of the outgoing traffic of stub ASs in another paper[15].

6.2. *Outbound interdomain traffic engineering for a transit AS*

Interdomain traffic engineering in the case of transit ASs is different to the one of stubs. Contrary to stub ASs, transit ASs receive traffic at their ingress points and forward it to another AS through some egress point. 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 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 metric), by relying on as few BGP routing changes as possible.

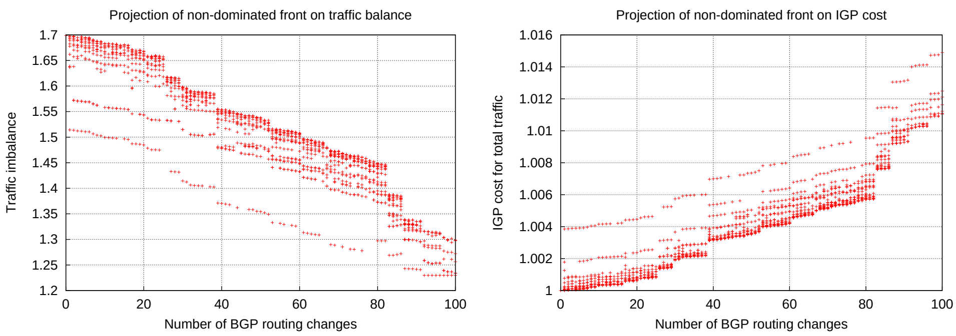

Fig. 7. Outbound traffic balancing and IGP cost minimization.

The two parts of Fig. 7 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 Fig. 7) and IGP cost (right of Fig. 7). The left part of Fig. 7 shows that for the particular scenario we used, the default traffic imbalance among the Internet connections is of about 1.7 (no 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 (BGP) does not balance the outbound traffic of a transit AS very well, hence interdomain traffic engineering is desirable for such networks.

The traffic optimization starts at no 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. Fig. 7 shows that adding BGP routing changes allows to improve the traffic balance but this also increases the cost of the traffic to cross the network. On the simulations of Fig. 7, 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 already 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 Fig. 7 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 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 to two instances of interdomain traffic engineering. 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. N. Feamster, J. Winick, and J. Rexford. A model of BGP routing for network engineering. In *Proc. of ACM SIGMETRICS*, June 2004.
8. B. Fortz, J. Rexford, and M. Thorup. Traffic engineering with traditional IP routing protocols. *IEEE Communications Magazine*, October 2002.
9. B. Halabi. *Internet Routing Architectures*. Cisco Press, 1997.
10. R. Mahajan, D. Wetherall, and T. Anderson. Understanding BGP Misconfigurations. In *Proc. ACM SIGCOMM '02*, September 2002.
11. R. Purshouse and P. Fleming. Conflict, Harmony, and Independence: Relationships in Multi-criterion Optimisation. In *Proc. of EMO2003*, April 2003.
12. B. Quoitin, S. Uhlig, C. Pelsser, L. Swinnen, and O. Bonaventure. Interdomain traffic engineering with BGP. *IEEE Communications Magazine, Internet Technology Series*, May 2003.
13. J. Stewart. *BGP4 : interdomain routing in the Internet*. Addison Wesley, 1999.
14. S. Uhlig and O. Bonaventure. Implications of interdomain traffic characteristics on traffic engineering. *European Transactions on Telecommunications*, January 2002.
15. S. Uhlig and O. Bonaventure. Designing BGP-based outbound traffic engineering techniques for stub ASes. *ACM SIGCOMM Computer Communication Review*, 34(5), 2004.
16. 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.