Towards Optimal Prepending for Incoming Traffic Engineering *

G. Di Battista M. Patrignani M. Pizzonia M. Rimondini

Dipartimento di Informatica e Automazione – Università degli Studi Roma Tre

E-mail: {gdb,patrigna,pizzonia,rimondin}@dia.uniroma3.it

Abstract

One of the main concerns for an Internet Service Provider (ISP) is to optimize the distribution of network traffic among its upstream providers, for example to balance bandwidth usage, or to distribute link costs evenly. While outgoing traffic can be easily controlled, influencing the volumes of incoming traffic is more challenging.

An effective and widely used technique to influence the distribution of incoming traffic is AS-path prepending, which consists in artificially inflating the length of the AS-path of BGP announcements. Since shorter AS-paths are often preferred, this can force incoming traffic to use different links. ISPs usually search for the optimal amount of prepending on a trial-and-error basis.

In this work, we formulate the problem of finding the optimal amount of prepending as an Integer Linear Programming problem, which permits to consider several optimality criteria and to embody many constraints. We also show how efficient algorithms for the problem can be devised by considering it from a Computational Geometry perspective. We believe that, under reasonable assumptions, this theoretic approach can have interesting practical impacts.

1. Introduction

An Internet Service Provider (ISP) interacts with the rest of the Internet using the Border Gateway Protocol (BGP). For each of its IP prefixes P, it "announces" P to its neighboring ISPs. In turn, such neighbors "pass" the announcement of P to their neighbors, etc. At each step, every ISP receiving an instance of the announcement checks it against its policies, compares it against other instances of the same announcement received from other neighbors, selects the best according to the BGP metrics, and sends to its neighbors only the selected one.

In this propagation process the ISP originating P progressively "loses the control" of what happens. Such a control is quite strong in the interaction with the immediate upstreams. It is regulated by a contract and enforced with several BGP features, like *communities*, *prepending*, etc. However, from the second-third step what will happen to the announcement of P is, up to a large extent, uncontrolled by the original ISP. Since the traffic takes the opposite direction with respect to the one of the announcements, not controlling the propagation of the announcements means not controlling the incoming traffic flows.

Up to now, this topic attracted limited research interest for several reasons:

1. Up to a few years ago, ISPs at the lower level of the hierarchy often had one upstream only, and in this case it is hard to influence the propagation of the announcements.

2. A stable routing was often considered much more important than an optimal traffic flow and the attention was often focused on stability rather than on efficiency.

3. For an ISP the knowledge of the rest of the Internet was quite poor.

Such obstacles are now less relevant than in the past. Most ISPs are multi-homed. Stability is always a problem but also the competition is and it is crucial to offer better services at a lower cost. Many resources are available to explore the structure of the Internet [9, 11, 2, 6].

Even a limited control on the propagation of the announcements could be used for:

1. Balancing the incoming traffic from the upstream providers, to improve performance or to shape the traffic according to the cost of the links.

2. Letting a large portion of the incoming traffic to use a specific transit Autonomous System (AS) that is known to be reliable and/or with high bandwidth availability.

*Work partially supported by European Commission - Fet Open project COSIN - COevolution and Self-organisation In dynamical Networks - IST-2001-33555, by European Commission - Fet Open project DELIS - Dynamically Evolving Large Scale Information Systems - Contract no 001907, by "Project ALGO-NEXT: Algorithms for the Next Generation Internet and Web: Methodologies, Design, and Experiments", MIUR Programmi di Ricerca Scientifica di Rilevante Interesse Nazionale, and by "The Multichannel Adaptive Information Systems (MAIS) Project", MIUR–FIRB.

3. Improving the distribution of the internal traffic flows of an ISP.

An announcement of a prefix P is equipped with an *AS-path*, which is the list of the ASes traversed by the announcement. At the beginning, the AS-path consists of just the *origin* AS. When propagating an announcement, each AS inserts (prepends) in the AS-path its identifier as the first element. The length of the AS-path is one of the main factors used to select the best path to reach P. *Prepending* is a technique that deliberately "inflates" the length of an AS-path. When an ISP does prepending it inserts its identifier several times (more than once) into the AS-path (see Fig. 1).

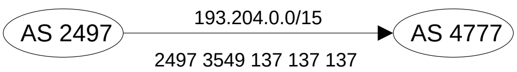

Figure 1. An example of usage of prepending. AS 2497 sends AS 4777 an announcement of prefix 193.204.0.0/15, which is originated by AS 137. From the AS path, it is possible to see that AS 137 announces its prefix with prepending (137 137 137), while the other ASes do not apply prepending.

We give a contribution to the problem of engineering the incoming traffic flows of an AS proposing two, in our opinion, promising approaches for the problem of choosing the optimal prepending. First, we present the current state of the art for this problem (Section 2). Then, we show how to formulate the problem with Integer Linear Programming techniques (Section 3). This allows to consider several optimality criteria and to embody many constraints. We also show how efficient algorithms for the problem can be devised by considering it from a Computational Geometry perspective (Section 4). We then argue about the practical impact of these research directions (Section 6).

2. Previous Work

The following brief survey on the state of the art in this field puts in evidence that only a few works address the optimal prepending problem directly.

An introduction to the basic principles of traffic engineering is made by Awduche et al. in [12]. This work mainly focuses on intra-domain traffic engineering, but it also presents some considerations about inter-domain routing.

Feamster et al. propose in [15] some objectives and guidelines for inter-domain traffic engineering using BGP. They show how data from BGP tables and from NetFlow [4] archives can be used to predict traffic flow changes, to limit the influence of neighboring ASes on the routing choices, and to reduce the overhead of routing changes.

Some BGP-based techniques for traffic engineering are presented in [21] by Quoitin et al. Such a work describes how to control both the incoming and the outgoing traffic of an ISP, but does not present an experimental study.

Another description of BGP-based techniques for traffic engineering is made by Swinnen et al. in [22]. It also contains an experimental study of the impact of AS-path prepending on incoming traffic volumes. In their study, they use the Javasim [7] event-driven simulator for running a BGP model over a topology built with the topology generator Brite [3]. They show that the distribution of inter-domain paths is actually affected by AS-path prepending.

Chang and Lo [13] approach the problem of finding the optimal prepending by using two kinds of measurements on the network. They collect NetFlow [4] data (passive measurements) and probe the network with *ping* packets in order to discover the upstream ISPs' routing policies with respect to the AS-path length (active measurements). Both kinds of measurements are used to predict the impact of prepending variations on network routing. They also test their methodology on a dual-homed AS. This approach, although effective, does not efficiently scale-up with the number of upstreams. The same paper shows that prepending-based techniques favorably compare with alternative methods.

Other contributions focus on the impact of routing policies on the length of the AS-paths [23, 16].

Further, it is worth mentioning that several "route control" tools are available to automatically perform traffic engineering [8, 5, 1]. These solutions consist of devices that analyze live data flows and, if needed, adjust the network configuration according to user defined policies (application priorities, expected network behavior, performance, etc.). Some route control tools also claim to perform incoming traffic optimization by tuning BGP announcements of local prefixes [8]. In our opinion, such tools would benefit from theoretically sound formulations of incoming traffic engineering via prepending.

3. Model and Integer Linear Programming Formulation

In this section, we first define a model to describe the prepending choices of an ISP and their effects on Internet routing. Second, we describe an Integer Linear Programming (ILP) formulation framework that can be used as a starting point to compute the optimal prepending, and discuss several alternative ILP objective functions.

Let A be the set of all the ASes. Consider a specific AS t, called *target*, announcing a prefix P to its peers. We call *upstreams* such peers and denote them by U ($U \subset A$). We assume they are numbered $1, \ldots, |U|$.

Each AS a (but t) receives one or more announcements about P and chooses one, based on the length of the AS-path associated with them.

We define a partition of the ASes of $A - \{t\}$ into sets $A_0, A_1, \ldots, A_{|U|}$, where A_i is the set of the ASes that reach P through upstream i. Set A_0 is used to denote the ASes that have two (or more) shortest paths to reach P. More formally, if a has exactly one shortest AS-path to reach P, we say that a belongs to A_i, where $i \in U$ is the last AS occurring in the AS-path before t. If a has two or more shortest paths to reach P, each using a different upstream link of t, we say that a belongs to A_0. Essentially, A_0 contains the ASes whose choice cannot be predicted by simply looking at the length of the AS-path. Fig. 2.(a) shows a simple network illustrating the model. Nodes represent ASes and edges represent peerings. We assume that policies allow announcements to traverse the network without constraints. Further, we assume that ASes choose the best announcement concerning P on the basis of the sole AS-path length. We refer to a simple model where attributes such as Local-Preference, MED, etc. are not used.

By using prepending, AS t can try to affect the way in which the other ASes reach P. AS t uses a prepending $w_i \geq 1$ when announcing P to AS i, $i \in U$ if it inserts its identifier in the AS-path w_i times. Consider again the network of Fig. 2.(a). Suppose that t wants to decrease the number of ASes that reach P through AS 2 (i.e., it wants to decrease the size of A_2). It can apply prepending $w_1 = 1$, $w_2 = 3$, $w_3 = 1$, and $w_4 = 1$. The new sets A_i are shown in Fig. 2.(b).

An administrator of t could use prepending for different purposes.

- EQUAL-CARDINALITY. The first possibility is to balance as much as possible the cardinalities of the sets A_i. This, to a first approximation, corresponds to balancing traffic to t coming from its upstream links.

- EQUAL-LOAD. Another possibility is to take into account the amount of traffic sent by each AS to t, so that the load on each upstream link is balanced.

- SHAPE-BANDWIDTH. Bandwidth requirements can be considered as well. The aim then becomes to compute a prepending assignment such that the load on the upstream links conforms to bandwidth availability.

- EQUAL-COST. It is also possible to introduce a cost model for the upstream links, which takes into account their usage. This leads to search for the prepending assignment that ensures the best cost sharing.

- EQUAL-COST-THRESHOLD. The cost model can be refined to introduce a fixed base cost and a threshold, so that additional charges are only applied when the threshold is crossed. ISPs often apply this kind of charging in their contracts.

We now propose an ILP formulation of the problem of finding the "best" prepending. Observe that a naïve approach to find the optimal prepending could consist of trying every possible combination of prepending amounts and choosing the combination that minimizes a specific objective function. This could require a great number of attempts, and for each attempt a considerable amount of time. Actually, after a fault, the network is known to converge within few minutes [18, 17]. However, many consecutive routing updates can easily trigger route flap damping [20, 24]. Experimental settings which need to repeatedly send updates at fixed rates adopt time intervals of 2 hours [19]. Furthermore, ISPs may deprecate a great amount of configuration changes on the part of their peers. This is why we consider approaches that can achieve optimality with a limited number of attempts.

The ILP formulation we propose is as follows:

- Constants d_{ai} are an input of the problem, and the ISP t is supposed to know them (we shall see in Section 6 which is the practical impact of this assumption). They represent the length of the shortest AS-path from a to i, when P is announced to i only.

- w_j represents the amount of prepending through upstream j.

- Variables c_{ai} are used to model how the ASes choose to reach prefix P. Namely, c_{ai} is 1 if AS a uses upstream link i to reach P. 0 otherwise.

- We consider the following generic objective function, that will be refined later on.

$$\min f(c_{ai}) \tag{3.1}$$

There are two main types of constraints: CHOICE-CONSTRAINTS and TIE-CONSTRAINTS.

CHOICE-CONSTRAINTS (3.2) model the choice of an AS that has only one shortest AS-path to reach P. In particular, if AS a chooses upstream link i ($c_{ai} = 1$), then the corresponding AS-path is the shortest one and there is no other path with the same length involving a different upstream link:

$\forall a \in A - \{t\}, i \in U :$

$$c_{ai} = 1 \Rightarrow \forall j \in U - \{i\} : w_j + d_{aj} > w_i + d_{ai}$$

In order to write this implication in the form of a set of linear constraints, we introduce a constant M. We choose

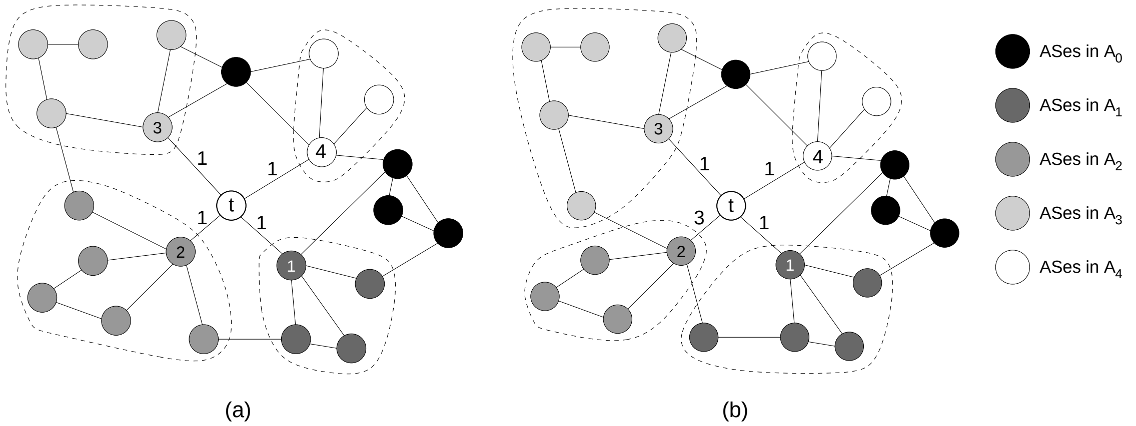

Figure 2. (a) AS t is a *target* AS, ASes $1, \ldots, 4$ are its upstreams. The figure shows the sets $A_0, \ldots, A_4$. (b) The effect of prepending. Edge labels represent prepending amounts. The figure shows the sets $A_0, \ldots, A_4$ after changing the prepending.

M large enough to ensure that constraints 3.2 are satisfied whenever $c_{ai} = 0$.

$\forall a \in A - \{t\}, \forall i, j \in U, i \neq j$:

$$w_j + d_{aj} > w_i + d_{ai} + (c_{ai} - 1)M \qquad (3.2)$$

TIE-CONSTRAINTS (3.3, 3.4, and 3.5) model the case in which AS a knows (at least) two shortest paths to reach P through two different upstreams i and j. When this happens, e_{aij} is 1. Otherwise, e_{aij} is 0. Obviously, the two paths through i and j have the same length:

$\forall a \in A - \{t\}, \forall i, j \in U, i \neq j$:

$$e_{aij} = 1 \Rightarrow w_i + d_{ai} = w_j + d_{aj}$$

and their length is that of a shortest path:

$\forall a \in A - \{t\}, \forall i, j \in U, i \neq j$:

$$e_{aij} = 1 \Rightarrow \forall k \in U : w_i + d_{ai} \leq w_k + d_{ak}$$

Again, we use a constant M that is large enough to satisfy constraints 3.3, 3.4, and 3.5 when $e_{aij} = 0$.

$\forall a \in A - \{t\}, \forall i, j \in U, i \neq j$:

$$\begin{aligned} w_i + d_{ai} &\geq w_j + d_{aj} + (e_{aij} - 1)\, M & (3.3) \\ w_j + d_{aj} &\geq w_i + d_{ai} + (e_{aij} - 1)\, M & (3.4) \end{aligned}$$

$\forall a \in A - \{t\}, \forall i, j, k \in U, i \neq j$:

$$w_i + d_{ai} + (e_{aij} - 1)\, M \leq w_k + d_{ak} \qquad (3.5)$$

Constraint 3.6 is introduced to ensure that each AS a either belongs to one A_i or belongs to A_0. Note that, if $a \in A_0$, there can be more than one e_{aij} that is set to 1. Observe that constraints 3.2, 3.3, and 3.4 prevent the two situations from happening simultaneously.

$\forall a \in A - \{t\}$:

$$\sum_{i \in U} c_{ai} + \sum_{i,j \in U, i \neq j} e_{aij} > 0 \qquad (3.6)$$

Constraints 3.7 to 3.10 are used to define the domains of the variables.

$$\begin{aligned} \forall a \in A - \{t\}, i \in U \quad &: \quad c_{ai} \in \{0, 1\} & (3.7) \\ \forall a \in A - \{t\}, & & \\ \forall i, j \in U, i \neq j \quad &: \quad e_{aij} \in \{0, 1\} & (3.8) \\ \forall i \in U \quad &: \quad w_i \in \mathbb{N}, & (3.9) \\ & \quad\;\; w_i > 0 & (3.10) \end{aligned}$$

Let $n = |A|$ and $m = |U|$. The size of the problem is dominated by constraints 3.5, which give raise to $(n-1)m^2(m-1)$ inequalities.

Objective function 3.1 can be used to implement several requirements, as follows. Standard operations research techniques can be used to plug all the following objective functions in the ILP.

EQUAL-CARDINALITY.

$$f(c_{ai}) = \max_{i,j \in U} \left(\sum_{a \in A - \{t\}} c_{ai} - \sum_{a \in A - \{t\}} c_{aj} \right) \qquad (3.11)$$

EQUAL-LOAD. Let l_a, $a \in A - \{t\}$ be the amount of traffic that AS a sends to P.

$$f(c_{ai}) = \max_{i,j \in U} \left(\sum_{a \in A-\{t\}} c_{ai} l_a - \sum_{a \in A-\{t\}} c_{aj} l_a \right) \tag{3.12}$$

SHAPE-BANDWIDTH. Let b_i, $i \in U$ be the available bandwidth on upstream link i.

$$f(c_{ai}) = \max_{i \in U} \left| \sum_{a \in A-\{t\}} c_{ai} l_a - b_i \right| \tag{3.13}$$

Minimizing 3.13 corresponds to limiting traffic bursts and, at the same time, maximizing link usage.

EQUAL-COST. Let cost_i be the cost that AS t has to pay for the upstream link i.

$$f(c_{ai}) = \max_{i,j \in U} (\text{cost}_i - \text{cost}_j) \tag{3.14}$$

Function 3.14 can be used in conjunction with different cost models (i.e., different definitions of *cost*$_i$). For example, let *unit*$_i$ be the cost for the unit of traffic that flows through upstream link i. The following constraint defines a simple linear cost model:

$$\forall i \in U : \text{cost}_i = \text{unit}_i \sum_{a \in A-\{t\}} c_{ai} l_a \tag{3.15}$$

EQUAL-COST-THRESHOLD. A more realistic cost model that can be used with objective function 3.14 is the one in which upstream links have a fixed base cost *base*$_i$ and traffic exceeding a threshold *thresh*$_i$ is charged *unit*$_i$ per unit. $\forall i \in U$:

$$\text{cost}_i = \max \left\{ \sum_{a \in A-\{t\}} c_{ai} l_a - \text{thresh}_i, 0 \right\} \text{unit}_i + \text{base}_i$$

3.1. Dealing with Ties

As already discussed, given a prepending assignment, an AS a can have two or more shortest AS-paths to reach P, each using a different upstream link of t. In this case, we say that there is a *tie* at AS a. The AS-path chosen by a to reach P cannot be predicted on the basis of the sole AS-path length. Since ties correspond to unpredictable choices, we would like not to introduce ties at all.

Observe that a prepending assignment that does not give rise to ties always exists (just set $w_1 = 1$ and $w_i = n$, $i = 2, \ldots, m$), but it is possible to show that such an assignment can be arbitrarily bad.

As an example, consider the network in Fig. 3 and suppose to rule out those prepending assignments that introduce ties. In this network, there are two AS-paths between AS 1 and AS 2: one of odd length (1 3 4 5 2) and one of even length (1 6 7 2). Every prepending assignment such that $|w_1 - w_2| = 2k$, $k = 0, 1$ introduces a tie at one of the ASes in the path 1 3 4 5 2. On the other hand, every prepending assignment such that $|w_1 - w_2| = 2k + 1$, $k = 0, 1$ introduces a tie at one of the ASes in the path 1 6 7 2. Therefore, ties are only avoided by using prepending amounts such that $|w_1 - w_2| \geq 4$. In this way, all the ASes would fall into either A_1 or A_2 and, for example, the objective function EQUAL-CARDINALITY would assume high (i.e., bad) values, which is undesirable.

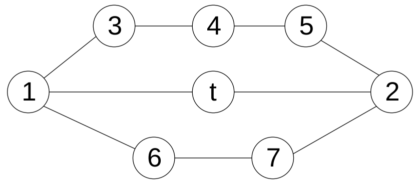

Figure 3. A network for which ties can only be avoided with trivial prepending assignments (e.g., $w_1 = 5$ and $w_2 = 1$ or $w_1 = 1$ and $w_2 = 5$).

Consider that a configuration similar to the one in Fig. 3 is likely to appear in real world instances. In conclusion, looking for solutions that do not introduce ties can lead to very poor prepending assignments.

This is the reason why objective functions do not consider ASes in A_0. However, once an optimal prepending assignment has been found by using the ILP, the number of ASes in A_0 provides an estimate of the quality of the assignment itself: assignments with a low number of ties should be preferred. All the ASes in A_0 may then be arbitrarily assigned to one of the $A_1, \ldots, A_{|U|}$ (i.e., the corresponding c_{ai} can be set to 1). By doing so, it is possible to explore the range of values that objective functions can assume, thus deriving bounds on the quality of the assignment.

3.2 Considering Multiple Prefixes

The model we have introduced assumes that AS t announces a single prefix P. However, an AS typically announces many prefixes to its neighbors. Both the model and the ILP formulation can be modified to consider multiple prefixes. Since different policies can be applied for each prefix, the input values d_{ai} depend on the specific prefix being considered.

In particular, suppose that t announces p prefixes $P_1, P_2, \ldots, P_p$. We can introduce vectors $\mathbf{d}_{ai}$, such that the k-th component ($\mathbf{d}_{ai}[k]$) of vector $\mathbf{d}_{ai}$ is the length of the shortest path from a to i when t announces P_k to i only.

Similarly, also variables w_i, c_{ai}, e_{aij} can be replaced by vectors, and the sets A_i can be organized in a vector as well. When using objective functions EQUAL-LOAD, the quantities l_a become vectors too.

Constraints 3.2 to 3.10 must be written for each prefix P_k, and they must use the values $\mathbf{w}_i[k]$, $\mathbf{d}_{ai}[k]$, $\mathbf{c}_{ai}[k]$, and $\mathbf{e}_{aij}[k]$.

Objective functions should be modified to consider all the prefixes. For example, function EQUAL-CARDINALITY (3.11) can be replaced with the following:

$$f(\mathbf{c}_{ai}) = \max_{i,j \in U} \left(\sum_{a \in A-\{t\}}^{1 \leq k \leq p} \mathbf{c}_{ai}[k] - \sum_{a \in A-\{t\}}^{1 \leq k \leq p} \mathbf{c}_{aj}[k] \right)$$

Also, the cost models should be rewritten. For example, the linear cost model 3.15 can be replaced by the following:

$$\forall i \in U : \text{cost}_i = \text{unit}_i \sum_{a \in A-\{t\}}^{1 \leq k \leq p} \mathbf{c}_{ai}[k] \mathbf{l}_a[k]$$

4. A Computational Geometry Approach

We now show how computational geometry ingredients can lead to efficient algorithms for the optimal prepending problem. For simplicity, we focus on the case in which t has 3 upstreams (i.e., $m = 3$). Our considerations can be generalized to a greater number of upstreams.

For any choice of prepending, it is possible to map each AS $a \in A - \{t\}$ to a point in a 3-dimensional Euclidean space, parametrically with respect to the amount of prepending. In particular, if we consider a coordinate system $OX_1X_2X_3$, such a point has coordinates $[x_1, x_2, x_3]^T$, where $x_i = d_{ai} + w_i$, $i = 1, 2, 3$.

This space can be partitioned into regions so that all the points falling in the same region correspond to ASes in the same $A_i, i = 1, 2, 3$. Points belonging to A_0 fall on the boundary between regions. Changing the prepending results in translating all the points or, equivalently, the coordinate system. In this way, points can shift from one region to another.

More precisely, the correspondence among the regions and the sets A_i, $i = 1, 2, 3$ is as follows:

$$\begin{aligned} A_1 \quad &\leftrightarrow \quad \begin{cases} x_1 < x_2 \\ x_1 < x_3 \end{cases} \\ A_2 \quad &\leftrightarrow \quad \begin{cases} x_2 < x_1 \\ x_2 < x_3 \end{cases} \\ A_3 \quad &\leftrightarrow \quad \begin{cases} x_3 < x_1 \\ x_3 < x_2 \end{cases} \end{aligned} \tag{4.16}$$

Each pair of regions corresponding to (A_i, A_j), $i, j = 1, 2, 3, i \neq j$ is separated by a boundary B_{ij}:

$$\begin{aligned} B_{12} \quad &: \quad \begin{cases} x_1 = x_2 \\ x_3 \geq x_1 \end{cases} \\ B_{23} \quad &: \quad \begin{cases} x_2 = x_3 \\ x_1 \geq x_2 \end{cases} \\ B_{13} \quad &: \quad \begin{cases} x_1 = x_3 \\ x_2 \geq x_1 \end{cases} \end{aligned} \tag{4.17}$$

The union of B_{12}, B_{13}, and B_{23} corresponds to the set A_0. The intersection of B_{12}, B_{13}, and B_{23} defines a straight line $r : x_1 = x_2 = x_3$.

The following property holds.

Property 1 $\forall k > 0$, $\forall w_i$, $i = 1, 2, 3$*: the sets* A_j, $j = 0, 1, 2, 3$*, do not change when the prepending is set to* $w'_i = w_i + k$*.*

Property 1 can be proved by showing that using prepending $w'_i = w_i + k$ is equivalent to translating points in the same direction as r.

As a consequence of Property 1, studying the effect of prepending does not require considering all the combinations of prepending amounts. For example, it is possible to keep the prepending on an upstream i fixed, while only altering the others. Hence, the component x_i is fixed as well, which corresponds to projecting points on one of the coordinate planes. Consider that, if we fix a prepending amount, others may become negative while searching for optimal prepending. This can be amended by translating all the points in the same direction as r in order to move them to the first octant without altering the composition of sets A_i, $i = 0, 1, 2, 3$.

To exploit the symmetry of our construction, we project the points on the plane $H : x_1 + x_2 + x_3 = 0$, which passes through the origin O and whose normal is r. After projection, boundaries 4.17 become half lines. In order to easily figure out the disposition of the points and the shape of the regions after projection, we now consider a coordinate system $OX'_1X'_2$ such that axis X'_2 coincides with the projection of X_2 over H and axis X'_1 also belongs to H and is orthogonal to X_2.

Using standard linear algebra, it's easy to see that a point $\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T$ becomes

$$\begin{bmatrix} x'_1 \\ x'_2 \end{bmatrix} = \begin{bmatrix} \frac{x_1 - x_3}{\sqrt{2}} \\ \frac{-x_1 + 2x_2 - x_3}{\sqrt{6}} \end{bmatrix}$$

in the coordinate system $OX'_1X'_2$. Note that each point on H is the projection of a line parallel to r.

In Fig. 4, half lines X_1, X_2, X_3 represent the projection of the positive half of the of axes X_1, X_2, X_3, which partition the plane into three regions corresponding to A_1, A_2, A_3. Points falling over X_1, X_2, or X_3 correspond to A_0. Optimal prepending is obtained by placing the origin O so that one of the objective functions 3.11, 3.12, 3.13, 3.14 is minimized. Note that in 3.11, 3.12, 3.13, 3.14 the values of

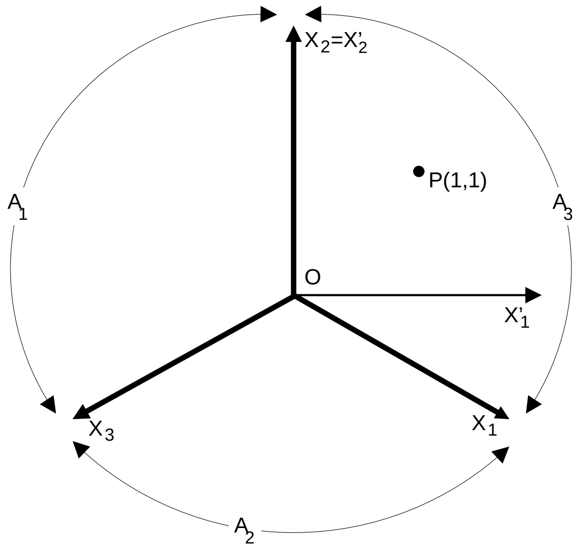

Figure 4. The projection on the plane H of the axes X_1, X_2, X_3 and the coordinate system $OX'_1X'_2$. Point P is the projection of the straight line $x_1 = t$, $x_2 = t+\frac{\sqrt{6}-\sqrt{2}}{2}$, $x_3 = t-\sqrt{2}$.

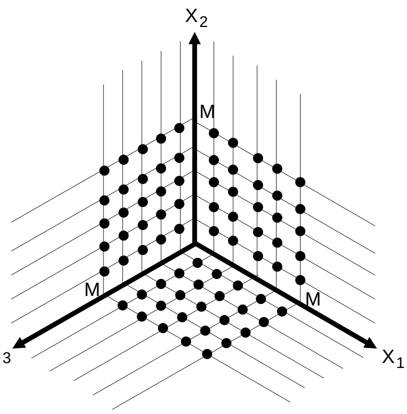

Figure 5. The points represent all the possible placements of origin O, corresponding to different prepending assignments. The drawing lies on the plane H. Finding optimal prepending requires placing origin O in $3M^2$ different points.

the c_{ai} can be easily computed on the basis of the disposition of the points on H.

Let $M = \max_{a \in A-\{t\}, i=1,2,3} d_{ai}$. The hexagon of side M lying on H and centered in O contains all the points. By projecting on H we need $3M^2$ attempts to find the position of the origin O that gives optimal prepending (see Fig. 5). If we consider any of the objective functions 3.11, 3.12, 3.13, 3.14, each attempt takes $O(n)$ time to compute the value of the function. In general, M is $O(n)$, and this corresponds to having $O(n^3)$ complexity.

The above considerations lead to the following theorem.

Theorem 1 *Given a network with n ASes and a target AS announcing a prefix to 3 upstream providers, the optimal amount of prepending to be used in the announcements can be found in $O(n^3)$ time.*

This holds for any of the objective functions EQUAL-CARDINALITY, EQUAL-LOAD, SHAPE-BANDWIDTH, EQUAL-COST, EQUAL-COST-THRESHOLD described in Section 3. This result can be generalized to an arbitrary number m of upstreams, thus leading to computational complexity $O(n^m)$.

5. Efficient Computational Geometry Algorithms

Theorem 1 proposes an $O(n^3)$ exhaustive algorithm to search for optimal prepending. One can argue that more clever techniques can lead to better results.

Exploring the prepending space by local search techniques is hard due to local minima. For example, consider a hill climbing search, in which it is possible to move from one prepending configuration by changing only one w_i, $i = 1, 2, 3$ of one unit. Fig. 6 shows a situation in which this kind of search is not effective for pursuing EQUAL-CARDINALITY.

An alternative approach consists in trying to avoid all the prepending assignments that lead to the same sets A_i, $i = 0, 1, 2, 3$. For this purpose, the plane H can be considered partitioned in *equivalence areas*, so that placing O in any of the points of an area leads to the same sets A_i. These areas are represented with different gray tones in Fig. 7.

Let R be the set of all equivalence areas and $n = |A|$. Suppose to add one point at a time. Each point introduces $(n-1)+2$ new equivalence areas. Hence, $|R|_n = |R|_{n-1} + n + 1$ and $|R|_1 = 3$. Therefore, we have that $|R|_n = \frac{n^2+3n+2}{2}$.

In real settings it is not unusual that a small set of ASes is responsible for a large amount of the ISP incoming traffic [22]. In such situations we can consider only those ASes that generate most of the traffic, and hence we can have $n << M$ (i.e., even with a small number of ASes we may have large distance values). In this case, exploiting equivalence areas can lead to a great speedup.

The set of all the equivalence areas is described by an arrangement of $3n$ half lines. In [14] is presented an algorithm for describing and enumerating the arrangement which takes $O(n^2)$ time, and this is shown to be optimal.

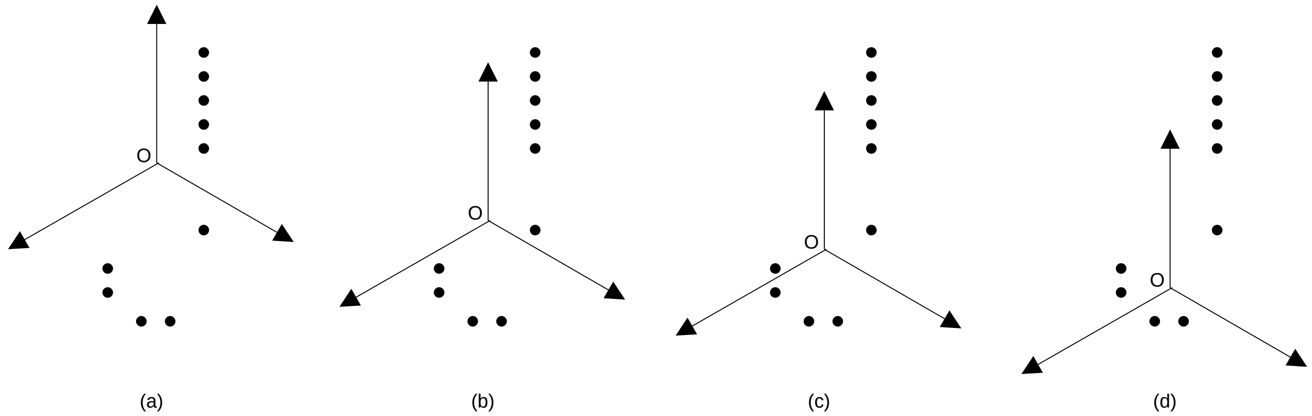

Figure 6. Using hill climbing search may not give optimal prepending. With respect to EQUAL-CARDINALITY**, we have: (a) a local minimum with $f(c_{ai}) = 5$; (b) by moving against hill climbing, we get $f(c_{ai}) = 6$; (c) $f(c_{ai}) = 5$; (d) $f(c_{ai}) = 4$.**

This result is also valid for an arbitrary dimension. That algorithm works with straight lines. However, considering the arrangement with straight lines instead of half lines does not increase the complexity of the arrangement. In fact, suppose to replace half lines with straight lines. Then, $|R|_n = 3n^2 + 2n + 1$, since $|R|_n = |R|_{n-1} + 6n - 1$ and $|R|_1 = 3$.

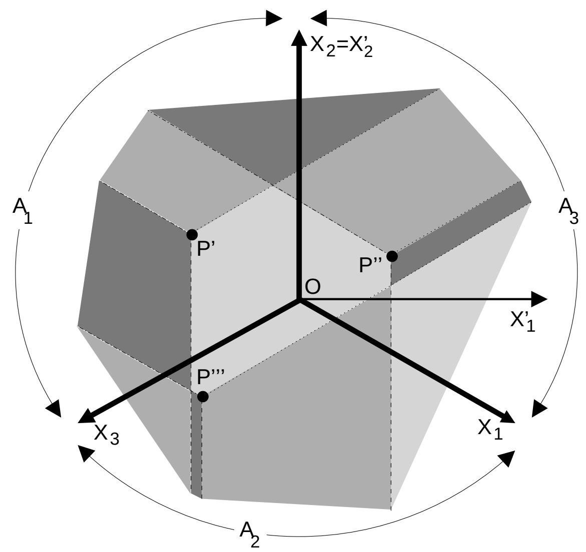

Figure 7. A configuration of points (P', P'', P''') after projection on H. The drawing lies on the plane H. Shaded areas are equivalence areas.

6. Concluding Remarks

We focus on the problem of optimizing the distribution of the incoming traffic of an ISP. In particular, we introduce a model to compute the optimal prepending that the ISP should use in its BGP announcements. We propose several optimality criteria, and we show how to compute optimal prepending both by using an Integer Linear Programming formulation and by exploiting Computational Geometry techniques.

In our model, the lengths of the shortest AS-paths from each AS to each upstream are supposed to be known. However, we believe that this assumption does not prevent the approach to have a practical impact. First, observe that in order to compute such lengths the knowledge of the whole network is not required. On the contrary, it is possible to announce a prefix to one upstream at a time and to measure how it reaches the remote ASes.

Second, the amount of information needed in practical cases is far less than the theoretical bound. A few ASes can be responsible for a large amount of incoming traffic. Restricting to these “critical” ASes (which can be automatically identified by exploiting widely adopted tools such as NetFlow) would provide a reasonable solution with a limited amount of input data.

Finally, the AS-paths of the announcements reaching the remote ASes may be retrieved from several sources. The Oregon Route Views Project [11] and RIS [9] collectors cumulatively offer a view of hundreds of ASes. Also, traceroute servers and BGP looking glasses [10] provide a long list of ASes for which such information is available. As a last resort, for those critical ASes not covered by these sources, an assumption of symmetric routing may be tried, and the reverse AS-path may be considered.

Due to the assumptions above, and to behaviors that are not modeled (like local preferences, “prefer customer” policies etc.) the solution found may be not optimal, and local search, as described in [13], may be required to refine it.

It would be interesting to investigate the impact of prepending choices on the stability of the Internet. Suppose that the interest for inter-domain traffic engineering increases and suppose that several ISPs start performing aggressive routing control based on prepending. That is, suppose that ISPs systematically “play” the game of influencing routing using prepending. By applying game theory techniques, it is possible to study whether this game admits a (Nash) equilibrium.

References

[1] Adaptive Networking Software - RouteScience, inc. `http://www.routescience.com`.

[2] BGPlay - Università degli Studi Roma Tre. `http://www.ris.ripe.net/bgplay/`, `http://bgplay.routeviews.org/bgplay/`.

[3] Brite Internet Topology Generator - Boston University. `http://www.cs.bu.edu/brite/`.

[4] Cisco IOS Netflow. `http://www.cisco.com/warp/public/732/Tech/nmp/netflow/`.

[5] Flow Control Platform - Internap, Inc. `http://www.internap.com/products/FCP-overview.htm`.

[6] Hermes project - Università degli Studi Roma Tre. `http://tocai.dia.uniroma3.it/~hermes`.

[7] Javasim simulation environment - The Ohio State University. `http://www.j-sim.org/`.

[8] Peer Director - Radware, Inc. `http://www.radware.com/content/products/pd/default.asp`.

[9] Réseaux IP Européens Routing Information Service (RIPE RIS). `http://www.ripe.net/ris/index.html`.

[10] Traceroute and looking glass servers. `http://www.traceroute.org/`.

[11] University of Oregon RouteViews Project. `http://www.routeviews.org`.

[12] 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.

[13] Rocky K. C. Chang and Michael Lo. Inbound traffic engineering for multihomed ases using as path prepending. In *Proc. NOMS 2004*, 2004.

[14] Herbert Edelsbrunner. *Algorithms in combinatorial geometry*. Springer-Verlag New York, Inc., 1987.

[15] N. Feamster, J. Borkenhagen, and J. Rexford. Guidelines for interdomain traffic engineering. In *ACM SIGCOMM Computer Communications Review*, October 2003.

[16] Lixin Gao and Feng Wang. The extent of as path inflation by routing policies. In *Proc. IEEE Global Internet Symposium 2002*, 2002.

[17] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed internet routing convergence. In *Proceedings of SIGCOMM*, pages 175–187, Stockholm, Sweden, September 2000.

[18] C. Labovitz, C. Wattenhofer, S. Venkatachary, and A. Ahuja. The impact of internet policy and topology on delayed routing convergence. In *Proc. IEEE INFOCOM*, 2001.

[19] Z. Mao, R. Bush, T. G. Griffin, and M. Roughan. Bgp beacons. In *IMC ’03: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pages 1–14. ACM Press, 2003.

[20] Z. Mao, R. Govindan, G. Varghese, and R. Katz. Route flap damping exacerbates internet routing convergence, 2002.

[21] B. Quoitin, S. Uhlig, C. Pelsser, L. Swinnen, and O. Bonaventure. Interdomain traffic engineering with bgp. *IEEE Communications Magazine*, (Volume 41, Issue 5):122–128, May 2003.

[22] L. Swinnen, S. Tandel, S. Uhlig, B. Quoitin, and O. Bonaventure. An evaluation of bgp-based traffic engineering techniques. Technical Report Infonet-2002-10, 2003.

[23] H. Tangmunarunkit, R. Govindan, S. Shenker, and D. Estrin. The impact of routing policy on internet paths. In *Proc. IEEE INFOCOM*, pages 736–742, 2001.

[24] C. Villamizar, R. Chandra, and R. Govindan. BGP route flap damping. RFC 2439, IETF, November 1998.