Algorithms for the Inference of the Commercial Relationships between Autonomous Systems: Results Analysis and Model Validation *

Massimo Rimondini Maurizio Pizzonia Giuseppe Di Battista Maurizio Patrignani

Dipartimento di Informatica e Automazione – Università di Roma Tre

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

Abstract

The Internet has recently been object of several studies concerning its structural properties and the behavior of routing protocols. One of the most interesting challenges is the inference of commercial relationships between Autonomous Systems. This knowledge would provide a deeper insight into the laws governing routing processes, and would constitute a useful guideline for choosing connection strategies and device configurations. Several algorithms have been proposed for inferring the relationships on the basis of the routing data of the Border Gateway Protocol (BGP). This work aims at performing an analysis of the results produced by state-of-the-art algorithms, with the purpose of investigating the meaningfulness of such results. This is achieved by running the algorithms extensively on several BGP data sets and by observing how assigned relationships change. Two kinds of analysis are used for doing this: one considers the relationships assigned by the same algorithm on data sets relative to different time instants; the second takes into account the relationships assigned by different algorithms on the same data set. We define a methodology and implement some tools for performing the two kinds of analysis and apply the methodology to two well known algorithms, using publicly available data sets. What comes out is that the number of AS pairs whose relationship is steadily assigned never falls below 94% of the overall pairs, and that the solutions computed by the two algorithms overlap for more than 90% of the pairs. This is an evidence of the fact that the inference is well-founded, i.e., it is not heavily influenced by routing oscillations, and that following different approaches almost yields the same solution, which further validates its trustworthiness.

*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 Progetto ALINWEB: Algoritmica per Internet e per il Web, MIUR Programmi di Ricerca Scientifica di Rilevante Interesse Nazionale, and by The Multichannel Adaptive Information Systems (MAIS) Project, MIUR Fondo per gli Investimenti della Ricerca di Base.

1. Introduction

The Internet consists of millions of interconnected devices which form a giant network. Such network can be analyzed in order to discover properties that regulate its behavior, and this can be done at different levels of abstraction. We are interested in considering the Internet as an interconnection of Autonomous Systems. An *Autonomous System* (AS) is a large set of network devices which are under the control of a single administration and whose routing policies appear from the outside as coherently organized.

ASes exchange reachability information by using the Border Gateway Protocol (BGP) [16, 17]. A reachability message consists of a set of contiguous IP addresses, which is called *prefix*, and is associated with the sequence of ASes that it traversed, which is called *AS path*. Each AS may have several routers which run the BGP protocol, called *border routers*. In particular, the border routers of two ASes exchange routing information by establishing *BGP sessions*.

Hence, the Internet can be modelled as an undirected graph $G(V, E)$ which is made up as follows: V is the set of ASes; E contains an edge for each pair of vertices whose routers establish a BGP session. We call such graph the *AS graph*.

The administrative authorities controlling the ASes need to subscribe contracts for obtaining connectivity to the rest of the Internet. These contracts can be of different kinds; in general (and a bit roughly), we refer to them by using the name *commercial relationships* (see [12]).

The BGP protocol allows to impose limits on the spread of routing information. The configuration elements that allow to do this are called *policies*. In practice, commercial relationships are implemented using specific sets of policies. What follows is a description of the most common commercial relationships, together with the policies that usually implement them (see, e.g., [13, 14, 10]).

Customer-provider: AS_c is a customer of AS_p if AS_c pays AS_p for obtaining the connectivity to the rest of Internet. The policies that are used to export routing information are usually the following:

- AS_c exports to AS_p its own prefixes and the ones of its customers; it does not export prefixes coming from its peers or providers;
- AS_p exports to AS_c its own prefixes and those of its other customers, peers, and providers.

Peer-peer: Two ASes are peers if they mutually agree to exchange traffic between their customers, quite often free of charge. The policies that are used to export routing information are usually the following:

- each of the two ASes exports to the other its own prefixes and the ones of its customers; it does not export prefixes coming from its peers or providers;

Understanding the commercial relationships between ASes is useful for several reasons. New Internet Service Providers (ISPs) can exploit such knowledge in order to infer the relevance of the other ASes in the Internet and hence choose better which ASes should be preferred for establishing commercial relationships. Network administrators can obtain useful hints from such knowledge, because it can help them in avoiding configurations which induce BGP instabilities (see, e.g., [11]). People studying the Internet evolution can exploit the knowledge of the commercial relationships to better understand the laws that control the growth of the network.

Explictly asking the ISPs for the relationships they establish each other is practically impossible for several reasons:

- the number of ASes which we may need to query is incredibly large;
- such organizations are usually not willing to reveal information that are sensible for their core business;
- it is hard even to collect the needed contact information.

Hence, other procedures must be introduced, that do not involve the direct contact with the AS organization. In fact, several algorithms have been proposed in the literature to infer the commercial relationships between ASes, based on the observation "from the outside" of their routing behavior. Section 2 contains a brief survey of such algorithms. This paper focuses on the problem of analyzing their outputs. The main contributions of this work are the following:

- We introduce a methodology for comparing the results of inference algorithms that, based on the structure of the AS graph, aim at inferring the commercial relationships between ASes.
- We describe a publicly available suite of software tools that can be used to perform the above mentioned comparisons.
- The methodology and the tools are exploited to perform two main types of analysis:

  Stability, to determine which is the level of similarity of the relationships inferred by an algorithm when using snapshots taken at different times.

  Algorithm independence, to evaluate which is the level of similarity of the relationships inferred by different algorithms when using the same snapshot.

- The results of our analysis show a high stability and a fair independence of the computed commercial relationships from the algorithm being used. This seems to be an (indirect) evidence of the fact that the algorithms succeed in their target.

The rest of this paper is organized as follows. Section 3 describes more extensively the purposes of our analysis, illustrates the methodology for performing the comparisons, and shows the software tools that support it. Section 4 reports the results of the application of the methodology on different kinds of data. Section 5 summarizes our results and outlines open issues to work on in the future.

2. Inferring Commercial Relationships between Autonomous Systems

In the Introduction we summarized the reasons why inference algorithms for computing the commercial relationships between ASes are needed.

Several algorithms [10, 18, 8, 9] have been proposed in the literature for doing this. Usually, they take as input a list of AS paths and produce as output a relationship assignment. The list of AS paths is obtained from one or more *telnet looking glass* servers, which essentially are routers that can be queried from a remote location. The `show ip bgp` command asks a looking glass to dump its Routing Information Base (essentially, its internal routing table). A list of telnet looking glass servers can be found at [4]. The obtained AS paths are then merged into a graph. Observe that the vertices of this graph are the ASes, and the edges correspond to adjacencies in the paths. In turn, an adjacency (AS_1, AS_2) is evidence of a BGP session between a router of AS_1 and a router of AS_2. Hence, the obtained graph is an AS graph.

An inference algorithm produces a *relationship assignment* on an AS graph G; that is, it labels the edges of G with the relationship occurring between its terminal nodes. The relationship assignment corresponds to a partial orientation of the AS graph G; it can be assumed that an undirected edge corresponds to a peering relationship, while a directed edge is oriented from customer to provider. Therefore, we will use the terms *orientation* and *relationship assignment* as synonyms.

Now consider an oriented AS graph. In [10] has been first observed that, if the relationships are actually implemented using the policies listed in the Introduction, then all the AS paths should have no *valleys*. Such property is called the *valley-free* property:

Property 2.1 (valley-free) *Given an oriented AS graph G, an AS path $p = AS_1, AS_2, \ldots, AS_n$ is* valley-free *(or* valid*) if one of the following conditions holds:*

1. *p is a sequence $AS_1, \ldots, AS_i, 1 \leq i \leq n$ of customer-provider edges, followed by a sequence $AS_i, \ldots, AS_n$ of provider-customer edges;*

2. *p is a sequence $AS_1, \ldots, AS_i, 1 \leq i < n$ of customer-provider edges, followed by the peer-peer edge AS_i, AS_{i+1}, followed by a sequence $AS_{i+1}, \ldots, AS_n$ of provider-customer edges.*

In other words, p is *valley-free* with respect to a given orientation if provider-customer edges are always followed by provider-customer edges and peer-peer edges are always followed by provider-customer edges.

All the algorithms which are described here are based on the assumption that realistic routing policies would lead to AS paths all satisfying the valley-free property.

Figure 1 shows some examples of invalid paths. Consider, for example, the path (a), which traverses the provider-customer edge (Pr_1, C) and then the customer-provider edge (C, Pr_2). Such a configuration implies that customer C exports prefixes coming from Pr_2 to Pr_1 or vice versa. This would result in customer C offering a transit service between Pr_1 and Pr_2, which is unrealistic. A similar argument holds for the paths (b), (c) and (d).

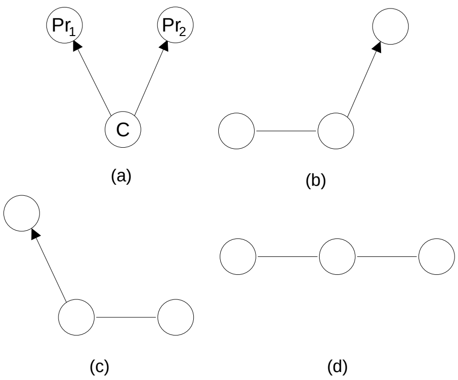

Figure 1. Examples of invalid paths

The valley-free property inspired the formulation of a combinatorial problem, which has been first presented in [18].

Problem 2.1 (ToR) *Given an AS graph G and a set of AS paths P, find an orientation (relationship assignment) of some of the edges of G which minimizes the number of invalid paths in P.*

It has been proved [8, 9] that ToR is NP-complete. Therefore, inference algorithms are either based on heuristics [10, 18] or on less constrained versions of the same problem [8, 9].

The first heuristic that has been proposed is the one by Lixin Gao [10]. This algorithm starts by computing the degree (number of adjacent ASes) of each AS and uses it to infer transit relationships. Finally, it assigns customer-provider relationships according to the inferred transit relationships. A refined version of this algorithm allows to also assign peer-peer relationships.

The second algorithm is the one by Agarwal et al. [18] (called *SARK* in the following). This algorithm considers routing data obtained from various looking glasses, which in the paper [18] are called *vantage points*. For each vantage point, the algorithm assigns a rank to the ASes. Then, it assigns the relationships on the basis of the ranks. In particular, if two adjacent ASes have a different rank, the one with lower rank is considered as a customer and the other one as a provider. Given a certain edge e, each vantage point could assign a different relationship to e, from its specific point of view. The algorithm actually assigns to e the relationship which is proposed by the highest number of vantage points.

The most recent algorithm (called *DPP* in the following) is the one which has been proposed by Di Battista et al. in [8] (a similar approach has been proposed in [9]). This algorithm is based on a reduction of the ToR problem to the well known problem of the satisfiability of propositional formulae (SAT). In particular, the algorithm looks for a solution of the ToR problem with no invalid paths, which reduces the computational complexity. This solution is essentially computed by defining an instance of the SAT problem and solving it. Starting from the input set P, a maximal subset of valid paths can then be computed by using some heuristics.

3. Methods and Tools

An important issue that is left open by the algorithms presented in Section 2 is to understand whether their approach yields results that are of practical interest.

One possibility could be to evaluate the proposed techniques against their ability of enforcing the valley-free property on the AS graph. However, this may not be enough. For example, suppose the input to our inference algorithm is just one AS path. Many solutions exist that make such path valley-free (see, e.g., Figure 2), but, presumably, only one of them is correct.

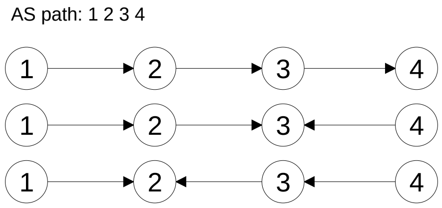

Figure 2. Examples of orientations which make an AS path valid.

In [10] the results of the inference are validated against a sample of the relationships taken from reality. However, this approach can be only performed in the small, since it cannot scale to the entire Internet. Hence, we believe that further investigation is needed in order to validate the inference results. One possible approach is to identify other features that a reasonably good inference should have.

We think that a reliable inference result should have the following two features:

Stability. Internet is constantly evolving under the pressure of social and economic forces, but this evolution is slow and never affected by many changes on the short period. On the contrary, the technical aspects of the network superimpose to the above slow change the changes produced by the routing algorithms, mainly when technical failures happens. This evolution usually suffers of bursts of many events on the short period. The results of an inference should ideally be mostly affected by the first kind of changes.

Independence from the algorithm. Computing the types of the relationships implies using an inference algorithm to do it. This, in turn, implies that the obtained relationships might be bound to choices which are specific to the chosen algorithm. This is obviously undesirable, since only one choice of the relationships can be considered correct. The ideal inference result should be algorithm independent, in the sense that it is the same (or, at least, very similar) for all "good" inference algorithms.

This constitutes a first possible set of features to test the validity of the different inference procedures known in literature. The aim is to exploit them in order to show that considering the valley-free property (or whichever else) alone is sufficient to understand whether the computed assignment is acceptable.

3.1. Methodology

This section describes our methodology, based on the following two kinds of analysis:

Stability analysis: compares the inference results obtained from a single algorithm run on inputs taken at different time instants;

Algorithm independence analysis: compares the inference results obtained using different algorithms on the same input.

We formally define the concept of relationship assignment as follows. Given a set of AS paths, a corresponding AS graph $G(V, E)$ is naturally induced (see Section 2). Such set of AS paths is given as input to an inference algorithm which returns a relationship assignment on G.

Let $R : E \rightarrow V \cup \{peering, unknown\}$ be a function giving the relationship assigned to each edge $e = (AS_1, AS_2)$ of G, in the following way:

$$R(e) \quad = \quad \begin{cases} AS_2 & \text{if } AS_2 \text{ is a provider of } AS_1 \\ AS_1 & \text{if } AS_1 \text{ is a provider of } AS_2 \\ peering & \text{if } AS_1 \text{ and } AS_2 \text{ are peers} \\ unknown & \text{if no relationship is known.} \end{cases}$$

In the following sections we identify the measures, based on $R(\cdot)$, used to analyze the inferred relationship assignments and we describe a software suite that makes it easy to apply our methodology.

3.2. Measures

We introduce measures which can be used to compare two distinct relationship assignments that we call $R_A(\cdot)$ and $R_B(\cdot)$, defined on the same graph $G(V, E)$. Namely, we investigate the differences between the assignments by identifying several sets of edges, each of them isolating a particular kind of difference, and measuring their cardinality.

We define the following subsets of E (see Figure 3):

$$\begin{aligned} Both &= \{e \in E | R_A(e) \neq unknown \wedge \\ &\quad R_B(e) \neq unknown\} \\ OnlyInA &= \{e \in E | R_A(e) \neq unknown \wedge \\ &\quad R_B(e) = unknown\} \\ OnlyInB &= \{e \in E | R_A(e) = unknown \wedge \\ &\quad R_B(e) \neq unknown\} \end{aligned}$$

The set $Both$ contains those edges which have successfully been assigned a relationship by both the inferences; $OnlyInA$ and $OnlyInB$ contain edges for which the assignment has successfully been inferred in only one case.

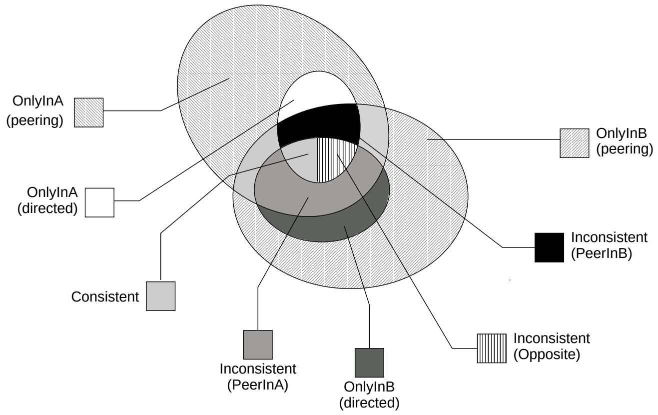

Figure 3. The sets of edges we consider for evaluating differences between orientations. This diagram is to be intended in a universe which contains the edges of G.

The set $Both$ is further partitioned into the following two subsets:

$$\begin{aligned} Consistent &= \{e \in Both \mid R_A(e) = R_B(e)\} \\ Inconsistent &= \{e \in Both \mid R_A(e) \neq R_B(e)\} \end{aligned}$$

The set $Inconsistent$ can be further investigated in order to identify the kind of difference occurring between the two assignments. Thus, it is possible to partition it into the following subsets:

$$\begin{aligned} Opposite &= \{e \in Inconsistent \mid \\ & \quad R_A(e) \neq peer \ \wedge \ R_B(e) \neq peer\} \\ PeerInA &= \{e \in Inconsistent \mid \\ & \quad R_A(e) = peer \ \wedge \ R_B(e) \neq peer\} \\ PeerInB &= \{e \in Inconsistent \mid \\ & \quad R_A(e) \neq peer \ \wedge \ R_B(e) = peer\} \end{aligned}$$

The set $Opposite$, as the name itself suggests, is the set of edges that have been assigned the opposite relationship in the two graphs (that is, they appear as customer-provider edges in one graph and as provider-customer edges in the other, or vice versa).

The above sets have been defined to compare two different assignments performed on the same graph. We now define measures to characterize how the same assignment algorithm works on many graphs distributed over time.

Consider a sequence of sets of AS paths that are obtained by consecutively probing the network over time and suppose to run a single inference algorithm on each set. Such sequence of sets induces a sequence of AS graphs: $G_1(V_1, E_1), \ldots, G_n(V_n, E_n)$.

Let $\widetilde{G} = (\widetilde{V}, \widetilde{E})$ be an AS graph which is defined as follows: $\widetilde{E}$ is the set of edges which appear in at least one of the G_i, that is, $\widetilde{E} = \bigcup_{i=1}^{n} E_i$; $\widetilde{V}$ is naturally induced by $\widetilde{E}$.[1] The inference on the i-th data set results in a relationship assignment $R_i(\cdot)$ which is defined on E_i. However, to make the notation easier, in the following we assume $R_i(\cdot)$ being defined on $\widetilde{E}$, having value $unknown$ for each edge in $\widetilde{E} \setminus E_i$.

The edges of $\widetilde{G}$ are labelled with values which summarize the history of the relationships assigned to them. In particular, we associate the following values to each $e \in \widetilde{E}$:

$$\begin{aligned} Occurrences(e) &= \left|\left\{i \mid e \in E_i\right\}\right| \\ Assignments(e) &= \left|\left\{i \mid R_i(e) \neq unknown\right\}\right| \\ Changes(e) &= \Big|\Big\{i \mid i \in \{1, \ldots, n-1\} \ \wedge \\ & \qquad R_i(e) \neq R_{i+1}(e)\Big\}\Big| \end{aligned}$$

The value $Occurrences(e)$ corresponds to the number of graphs $H_i (i = 1, \ldots, n)$ the edge e appears in; $Assignments(e)$ is the number of graphs in which edge e has been assigned a relationship; $Changes(e)$ is the number of times $R(e)$ changes its value in the sequence $R_1(e), \ldots, R_n(e)$.

3.3. Software Tools

We developed a suite of software tools to compute the measures defined in Section 3.2. The suite is called TORQUE (Type Of Relationship QUality Evaluation) and it is available in [3].

Computing such measures requires processing both the data sets provided as input to the inference algorithm and the corresponding inference results; the tools of the suite handle both kinds of information. The overall architecture of the suite is shown in Figure 4, where the different kinds of data are shown as square boxes while the tools are shown as rounded boxes. Data flow is shown using arrows.

Data sets come, usually, in the form of `show ip bgp` dumps (called *BGP dumps* in Figure 4), while inference algorithms work on files containing plain *AS path lists*. For this reason we implemented the *AS path extractor* tool. We payed special attention to implement various kinds of processing on the extracted AS paths (like removal of prepending, AS sets, cycling paths, duplicates, etc.). In this way the semantic of such processing results clear, and we believe this can help in establishing a unified approach for this kind of data analysis.

[1] Note that the graphs $G_1, \ldots, G_n$ are connected; therefore, no isolated nodes are omitted in $\widetilde{V}$.

The *graph generator* tool merges the information of an AS path list, of a BGP dump and of the relationship assignment computed by an inference algorithm, obtaining a *labelled AS graph* where each edge of the graph is associated with the values of the function $R(\cdot)$ (see Section 3.2). The tool allows us to obtain a graph representing the inference results with great flexibility.

The *graph compare* tool allows us to compare two or more labelled AS graphs. When two labelled AS graphs are given as input it can compute the cardinalities of the following sets: *Both*, *OnlyInA*, *OnlyInB*, *Consistent*, *Inconsistent*, *Opposite*, *PeerInA*, and *PeerInB*. In this case the output is a textual report. When more than two labelled AS graphs are given as input it can compute, for each edge, the following measures: *Occurrences*, *Assignments*, and *Changes*. In this case the output is an AS graph which is labelled with such measures. To produce various kinds of reports the *graph view* tool may be used.

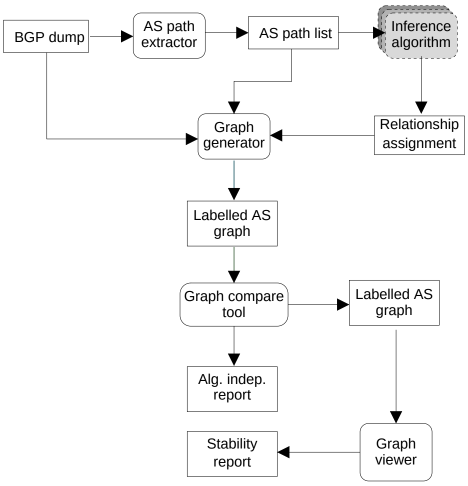

Figure 4. Typical usage of the TORQUE software suite for computing the metrics shown in Section 3.2. Square boxes represent data; rounded boxes represent tools.

4. Experiments

In this Section we present the results we obtained by applying the methodology to several data sets. In the following, we will use the term *snapshot* to denote both a list of AS paths and the AS graph which it induces. The specific meaning depends on the context.

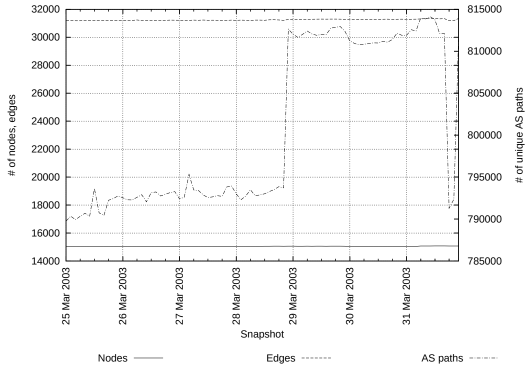

Figure 5. Dimensions of the snapshots in *RV1*.

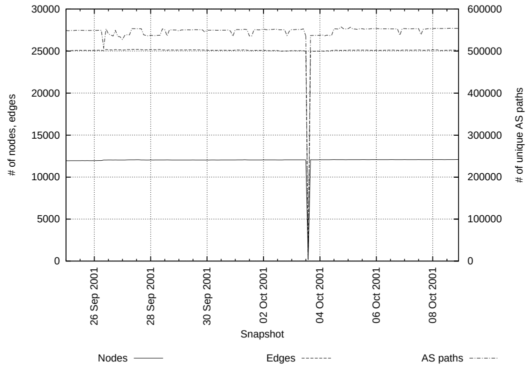

Figure 6. Dimensions of the snapshots in *RV2*.

4.1. Data Sets

We consider snapshots provided by two different sources.

The first one is the site [1] describing the work by Agarwal et al., which provides several snapshots. Each of them collects paths simultaneously taken from various BGP looking glasses. We consider the snapshots listed in Table 1. We address this data set using the name *MVP*.

The second source is the Oregon Route Views Archive [5], which provides BGP RIB dumps of a router having peering sessions with about 50 ASes at the time of the experiments. The dumps are collected every two hours. We consider the snapshots corresponding to the time intervals in Table 2. Since we want to test the stability of the inference results, we choose the snapshots in the following way: the first set, denoted with *RV1*, is a period during which only a few ASes changed the commercial agreements; the second one, *RV2*, is an interval during which

SNAPSHOTS IN THE MVP DATA SET.

Date	Used looking glasses	AS graph		Unique AS paths
		Vertices	Edges	
18 Apr 2001	1, 1740, 3549, 3582, 3967, 4197, 7018, 8220, 8709	10909	23817	511200
29 Jan, 04 Feb 2002	1, 3549, 3582, 3967, 4197, 7018, 8220, 8709	12708	27555	722481
06 Apr 2002	1, 1838, 3549, 3582, 3967, 4197, 5511, 7018, 8220, 8709, 15290	13079	28309	942382
29 Jul 2002	1, 1838, 3257, 3549, 3582, 3967, 4197, 5511, 7018, 8220, 8709, 15290	13705	29073	948720
09 Aug 2002	1, 1838, 3257, 3549, 3582, 4197, 5511, 7018, 8220, 15290	13754	29009	894396
19 Oct 2002	1, 1838, 3582, 3967, 5511, 7018, 8220, 15290	14113	29422	881836
29 Oct 2003	1, 50, 210, 553, 852, 1838, 3257, 3549, 3582, 3741, 3967, 4197, 5388, 5511, 6395, 6539, 6893, 7018, 8220, 8709, 8843, 9328, 15290	16420	37470	1143373
13 Nov 2003	1, 50, 210, 553, 852, 1838, 3257, 3549, 3582, 3741, 3967, 4197, 5388, 5511, 6395, 6539, 6893, 7018, 8220, 8709, 8843, 9328, 15290	16461	37406	1157802
28 Nov 2003	1, 50, 210, 553, 852, 1838, 3257, 3549, 3582, 3741, 3967, 4197, 5388, 5511, 6395, 6539, 6893, 7018, 8220, 8709, 8843, 9328, 15290	16316	31419	247691
12 Dec 2003	1, 50, 210, 553, 852, 1838, 3257, 3549, 3582, 3741, 3967, 4197, 5388, 5511, 6395, 6539, 6893, 7018, 8220, 8709, 8843, 9328, 15290	16585	37790	1216534
29 Dec 2003	1, 50, 210, 553, 852, 1838, 3257, 3549, 3582, 3741, 3967, 4197, 5388, 5511, 6395, 6539, 6893, 7018, 8220, 8709, 8843, 9328, 15290	16728	38162	1164370
13 Jan 2004	1, 50, 210, 553, 852, 1838, 3257, 3549, 3582, 3741, 3967, 4197, 5388, 5511, 6395, 6539, 6893, 7018, 8220, 8709, 8843, 9328, 15290	16762	38205	1264677

Table 1. The snapshots taken from [1].

	Start	End	# of snapshots
RV1	25 Mar 2003 00:00	31 Mar 2003 22:00	84
RV2	25 Sep 2001 00:00	08 Oct 2001 22:00	168

Table 2. The snapshots taken from [5].

ANALYSIS OF THE RESULTS OF THE DPP ALGORITHM OVER CONSECUTIVE SNAPSHOTS.

	Pair of consecutive input snapshots (A, B)		
	18 Apr 2001 (A) *29 Jan, 04 Feb 2002 (B)*	*29 Jan, 04 Feb 2002 (A)* *06 Apr 2002 (B)*	*06 Apr 2002 (A)* *29 Jul 2002 (B)*
OnlyInA	7	1	2
OnlyInB	18	9	1
Both	15919	24743	23638
Consistent	15118 (95%)	24112 (97%)	23044 (97%)
Inconsistent	801 (5%)	631 (3%)	594 (3%)
	29 Jul 2002 (A) *09 Aug 2002 (B)*	*09 Aug 2002 (A)* *19 Oct 2002 (B)*	*29 Oct 2003 (A)* *13 Nov 2003 (B)*
OnlyInA	5	3	4
OnlyInB	1	2	8
Both	28039	25458	35751
Consistent	27719 (99%)	24916 (98%)	35211 (98%)
Inconsistent	320 (1%)	542 (2%)	540 (2%)
	13 Nov 2003 (A) *28 Nov 2003 (B)*	*28 Nov 2003 (A)* *12 Dec 2003 (B)*	*12 Dec 2003 (A)* *29 Dec 2003 (B)*
OnlyInA	4	4	7
OnlyInB	2	2	2
Both	30654	31034	37038
Consistent	30120 (98%)	30509 (98%)	36653 (99%)
Inconsistent	534 (2%)	525 (2%)	385 (1%)
	29 Dec 2003 (A) *13 Jan 2004 (B)*		
OnlyInA	9		
OnlyInB	6		
Both	36865		
Consistent	36268 (98%)		
Inconsistent	597 (2%)		

Table 3. Data set *MVP*. Measurements performed using the DPP algorithm. The values represent the cardinalities of the sets in the leftmost column. Percentages are relative to the value of $Both$.

several commercial changes took place. We obtained information about commercial agreements from [6] and [2], and used them to identify the two intervals. We consider an assignment stable when the relationships remain almost the same over time. Figures 5 and 6 show the dimensions of the data sets *RV1* and *RV2*. The set *RV2* contains an incomplete snapshot, which is due to some failure in the Route Views collection process. The snapshot is the one of 03 Oct 2001 at 14:00.

All the data sets have been processed in order to extract AS paths from BGP dumps. We always consider lists of *unique* AS paths; i.e., for AS paths occurring more than once in a single data set only one instance is kept.

4.2. Stability Analysis

Our stability analysis considers all the data sets described above. In particular, we run the DPP algorithm (see Section 2) on all the sets *MVP*, *RV1*, and *RV2*, and we consider the results of the SARK algorithm on the *MVP* data set published on [1]. Stability is evaluated as follows.

For the *MVP* data we consider pairs of consecutive snapshots and compare the inferences computed by the same algorithm (DPP or SARK) on such pairs (with the exception of the pair 19 Oct 2002, 29 Oct 2003). In particular, for each pair we compute the cardinalities of the sets defined in Section 3.2, where $R_A(\cdot)$ and $R_B(\cdot)$ are defined on the intersection of the AS graphs of the two snap-

ANALYSIS OF THE RESULTS OF THE SARK ALGORITHM OVER CONSECUTIVE SNAPSHOTS.

	Pair of consecutive input snapshots (A, B)		
	18 Apr 2001 (A) *29 Jan, 04 Feb 2002 (B)*	*29 Jan, 04 Feb 2002 (A)* *06 Apr 2002 (B)*	*06 Apr 2002 (A)* *29 Jul 2002 (B)*
OnlyInA	86	109	96
OnlyInB	77	111	87
Both	15749	24472	23407
Consistent	15056 (96%)	23713 (97%)	22517 (96%)
Inconsistent	693 (5%)	759 (3%)	890 (3%)
Opposite	55 (8%)	33 (4%)	55 (6%)
PeerInA	350 (51%)	383 (50%)	358 (40%)
PeerInB	288 (41%)	343 (46%)	477 (54%)
	29 Jul 2002 (A) *09 Aug 2002 (B)*	*09 Aug 2002 (A)* *19 Oct 2002 (B)*	*29 Oct 2003 (A)* *13 Nov 2003 (B)*
OnlyInA	80	111	103
OnlyInB	94	89	157
Both	27789	25224	34542
Consistent	27343 (98%)	24425 (97%)	33812 (98%)
Inconsistent	446 (2%)	799 (3%)	730 (2%)
Opposite	14 (3%)	44 (6%)	21 (3%)
PeerInA	257 (57%)	367 (46%)	376 (52%)
PeerInB	175 (40%)	388 (48%)	333 (45%)
	13 Nov 2003 (A) *28 Nov 2003 (B)*	*28 Nov 2003 (A)* *12 Dec 2003 (B)*	*12 Dec 2003 (A)* *29 Dec 2003 (B)*
OnlyInA	613	64	157
OnlyInB	58	599	168
Both	29099	29482	35739
Consistent	28748 (99%)	29146 (99%)	35100 (98%)
Inconsistent	351 (1%)	336 (1%)	639 (2%)
Opposite	8 (2%)	2 (1%)	10 (2%)
PeerInA	237 (68%)	103 (31%)	346 (54%)
PeerInB	106 (30%)	231 (68%)	283 (44%)
	29 Dec 2003 (A) *13 Jan 2004 (B)*		
OnlyInA	111		
OnlyInB	177		
Both	35642		
Consistent	35172 (99%)		
Inconsistent	470 (1%)		
Opposite	14 (3%)		
PeerInA	233 (50%)		
PeerInB	223 (47%)		

Table 4. Data set *MVP*. Measurements performed using the SARK algorithm The values represent the cardinalities of the sets in the leftmost column. Percentages in the rows $Consistent$, $Inconsistent$ are relative to the value of $Both$; percentages in the other rows are relative to the value of $Inconsistent$.

shots. The results are shown in Tables 3 for the DPP algorithm and 4 for the SARK algorithm. The percent values for the rows $Consistent$ and $Inconsistent$ are relative to the value of $Both$; the percent values for the rows $Opposite$, $PeerInA$, and $PeerInB$ are relative to the value of $Inconsistent$. In Table 3 the cardinalities concerning peering relationships are omitted since the DPP algorithm never assigns peering relationships. All the comparisons show that at least 95% of the assigned relationships are consistent for consecutive data sets.

For the *RV1* and *RV2* data sets we consider the relationship assignments computed using the DPP algorithm and, for each of the two data sets, we evaluate the historical measures defined in Section 3.2. The results are shown in Figures 7 to 12. Table 5 reports the dimensions of the graph $\widetilde{G}$.

Figure 7 shows a distribution that helps us in evaluating the goodness of the inference as far as edge coverage is concerned. An edge e may appear in a number $Occurrences(e) \leq n$ of snapshots and be assigned relationship in $Assignments(e) \leq Occurrences(e)$ of them. On the X axis we show the percentage $\frac{Assignments(e)}{Occurrences(e)} \times 100$, and on the Y axis we show the number of edges that have such percentage. Figure 10 shows the same for the data set *RV2*. There are more than 30000 edges for *RV1* and more than 25000 for *RV2* to which the inference assigned a relationship in all the snapshots. That is, for at least 99% of the edges of $\widetilde{G}$ the assignment always succeeds.

Figure 8 shows the evolution over time of the fraction of the edges of each snapshot which have an assigned relationship. Figure 11 shows the same for the *RV2* data set. The values are fairly constant around 99% for every snapshot. The spike of value 100% on 03 Oct 2001 at 14:00 for *RV2* is due to the presence of an incomplete BGP dump in the Oregon Route Views Archive (see Figure 6).

Figure 9 shows a distribution that allows to understand how stable is the relationship assignment for each edge of $\widetilde{G}$. Each edge e may change its assignment a number $Changes(e)$ of times over the observation period. On the X axis we show the number of assignment changes for each edge, and on the Y axis we show the number of edges that have such changes. Figure 12 shows the same distribution for the *RV2* data set. The number of edges for which the assignment never changes is around 30000 for *RV1* and around 25000 for *RV2*. This means that the percentage of the edges of $\widetilde{G}$ which never change the assignment is always above 94%. Interestingly, Figures 9 and 12 put in evidence a scale free behavior (see, e.g., [7]) of the distribution of the values of $Changes(e)$. As far as we know, this is the first time that a scale free distribution is observed in this type of phenomenon.

	RV1	*RV2*
$\widetilde{V}$	15150	12317
$\widetilde{E}$	32534	27490

Table 5. Main features of the graph $\widetilde{G}$.

4.3. Independence from the Algorithm

Algorithm independence is evaluated in the following way.

We compare the relationships inferred by two different algorithms on the same snapshot G, and use the cardinalities of the sets defined in Section 3.2 to estimate their level of similarity. The compared assignments, $R_A(\cdot)$ and $R_B(\cdot)$, are both defined on the same graph G.

We perform this comparison for each of the snapshots listed in Table 1, and compare the results of the DPP algorithm against those of the SARK algorithm.

The results are shown in Table 6. The edges in $Consistent$ are always more than 90% of those in $Both$, which means that the two algorithms almost produce the same solution. Inconsistencies are equally shared between $Opposite$ and $PeerInB$.

5. Conclusions and Open Problems

In this paper we analyze the results produced by state-of-the-art algorithms for the inference of the commercial relationships between Autonomous Systems. We perform two kinds of analysis: the first evaluates the degree of independence of the inferred relationships from routing oscillations; the second kind of analysis is aimed at determining whether the results are independent from the specific algorithm being used. We introduce a methodology to perform the analyses and implement a software toolkit [3] which supports the methodology. We apply the methodology to two well known algorithms [8] and [18], using publicly available data sets from [1] and [5].

What we find is that the algorithms produce highly stable results; in particular, for both the algorithms, using data taken from [1], the percentage of AS pairs that have the same assigned relationship when using the same algorithm on two time adjacent snapshots is above 95%; for the algorithm [8] and data taken from [5] this percentage slightly reduces to 94%.

The results also show that the algorithms [8] and [18] almost produce the same assignments: among the AS pairs to which both the algorithms assigned a relationship, more than 90% have the same assignment.

Such conclusions make us think that the valley-free approach, on which inference algorithms are based, leads to reliable results.

Results of the DPP algorithm against those of the SARK algorithm.

	Snapshot			
	18 Apr 2001	*29 Jan, 04 Feb 2002*	*06 Apr 2002*	*29 Jul 2002*
OnlyInA	192	213	197	200
OnlyInB	39	21	6	6
Both	23584	27317	28106	28866
Consistent	21487 (91%)	24991 (91%)	25631 (91%)	26261 (91%)
Inconsistent	2097 (9%)	2326 (9%)	2475 (9%)	2605 (9%)
Opposite	993 (47%)	1047 (45%)	1217 (49%)	1176 (45%)
PeerInB	1104 (53%)	1279 (55%)	1258 (51%)	1429 (55%)
	09 Aug 2002	*19 Oct 2002*	*29 Oct 2003*	*13 Nov 2003*
OnlyInA	179	196	1205	1173
OnlyInB	9	14	28	25
Both	28819	29212	36235	36208
Consistent	26348 (91%)	26659 (91%)	32683 (90%)	32651 (90%)
Inconsistent	2471 (9%)	2553 (9%)	3552 (10%)	3557 (10%)
Opposite	1165 (47%)	1247 (49%)	1689 (48%)	1738 (49%)
PeerInB	1306 (53%)	1306 (51%)	1863 (52%)	1819 (51%)
	28 Nov 2003	*12 Dec 2003*	*29 Dec 2003*	*13 Jan 2004*
OnlyInA	1548	1367	1194	1118
OnlyInB	5	12	37	22
Both	29866	36409	36930	37063
Consistent	28332 (95%)	32754 (90%)	33321 (90%)	33557 (91%)
Inconsistent	1534 (5%)	3655 (10%)	3609 (10%)	3506 (9%)
Opposite	829 (54%)	1783 (49%)	1795 (50%)	1711 (49%)
PeerInB	705 (46%)	1872 (51%)	1814 (50%)	1795 (51%)

Table 6. Data set *MVP*. Comparison between the DPP and the SARK algorithms. For each column, (A) corresponds to the results of the DPP algorithm; (B) corresponds to those of the SARK algorithm. The values represent the cardinalities of the sets in the leftmost column. The row $PeerInA$ has been skipped because the DPP algorithm does not infer peering relationships.

There are also several interesting topics which we consider relevant and worth being further analyzed. What follows is a limited list:

- We would like to exploit the methodology to investigate the results obtained by the algorithm in [10].
- It would be interesting to better characterize the space of the solutions of the relationship assignment problem. Up to now, the research focused on the identification of just one solution, but it would be important to have a complete view of the degrees of freedom of the problem.
- Other papers tackled the problem of studying the commercial relationships from a game-theoretic point of view [15]. It would be interesting to examine possible contact points between such papers and the algorithms for inferring commercial relationships.

References

[1] Characterizing the internet hierarchy from multiple vantage points. `http://www.cs.berkeley.edu/~sagarwal/research/BGP-hierarchy/`.

[2] ISP Business: Archives. `http://isp-planet.com/`.

[3] TORQUE: Type Of Relationship Quality Evaluation. `http://www.dia.uniroma3.it/~compunet/torque`.

[4] Traceroute.org list of looking glasses. `http://www.traceroute.org`.

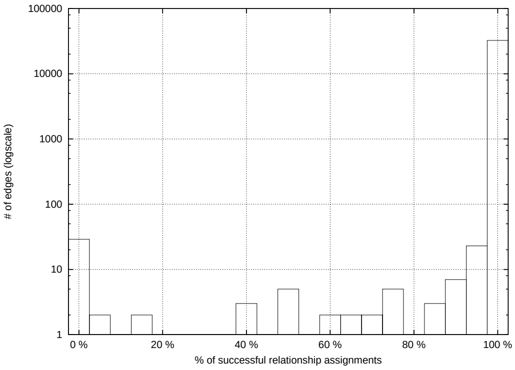

Figure 7. Data set $RV1$**. Distribution of the values of** $\frac{Assignments(e)}{Occurrences(e)} \times 100, e \in \widetilde{E}$**.**

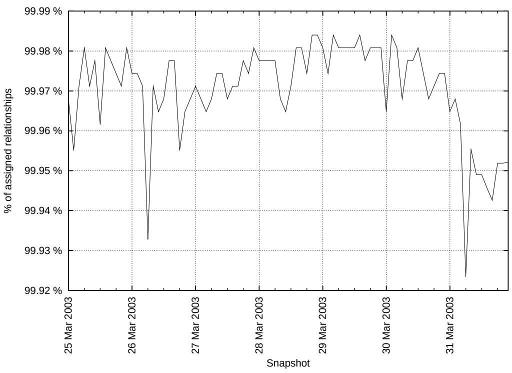

Figure 8. Data set $RV1$**. Evolution over time of the fraction of the edges that have been assigned a relationship.**

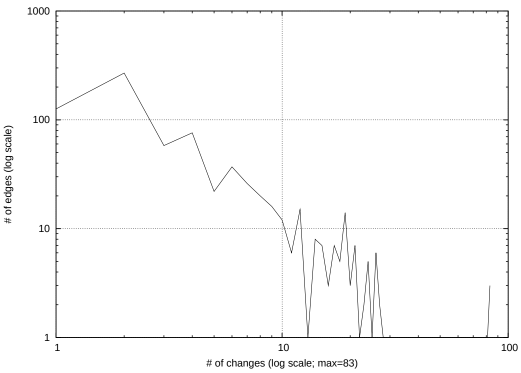

Figure 9. Data set $RV1$**. Distribution of the values of** $Changes(\cdot)$**.**

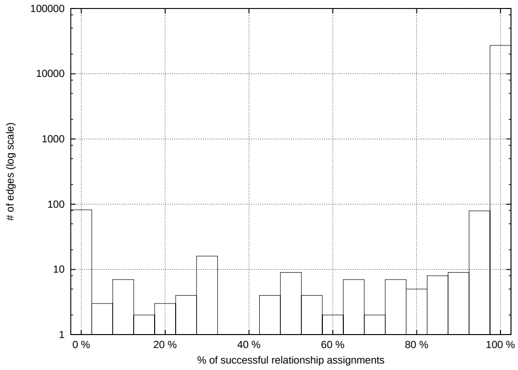

Figure 10. Data set $RV2$**. Distribution of the values of** $\frac{Assignments(e)}{Occurrences(e)} \times 100, e \in \widetilde{E}$**.**

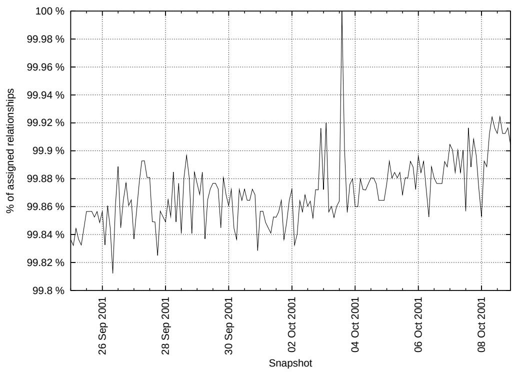

Figure 11. Data set $RV2$**. Evolution over time of the fraction of the edges that have been assigned a relationship.**

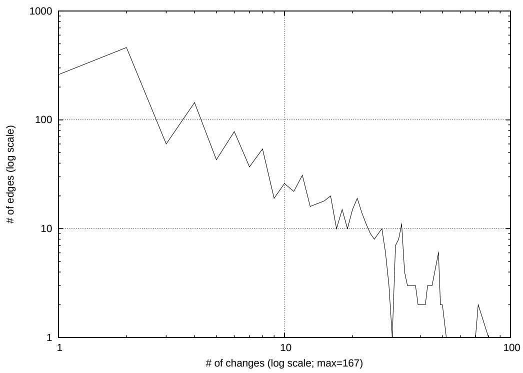

Figure 12. Data set $RV2$**. Distribution of the values of** $Changes(\cdot)$**.**

[5] University of Oregon RouteViews project.
`http://www.routeviews.org`.

[6] xSP Archives.
`http://www.internetnews.com`.

[7] Albert-László Barabási. *Linked: The New Science of Networks*. Perseus Publishing, 2002.

[8] Giuseppe Di Battista, Maurizio Patrignani, and Maurizio Pizzonia. Computing the types of the relationships between autonomous systems. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, volume 1, pages 156–165, March–April 2003.

[9] Thomas Erlebach, Alexander Hall, and Thomas Schank. Classifying customer-provider relationships in the internet. In *International Conference on Communications and Computer Networks (CCN 2002)*, pages 538–545, Cambride, USA, November 2002. IASTED.

[10] Lixin Gao. On inferring autonomous system relationships in the internet. *IEEE/ACM Transactions on Networking*, 9(6):733–745, Dec 2001.

[11] Lixin Gao, Timothy G. Griffin, and Jennifer Rexford. Inherently safe backup routing with BGP. In *IEEE INFOCOM 2001*, pages 547–556, Apr 2001.

[12] Geoff Huston. *ISP Survival Guide: Strtegies for Running a Competitive ISP*. Paperback, New York, NY, 1998.

[13] Geoff Huston. Interconnection, peering, and settlements – Part I. *Internet Protocol J.*, March 1999.

[14] Geoff Huston. Interconnection, peering, and settlements – Part II. *Internet Protocol J.*, June 1999.

[15] Christos H. Papadimitriou. Algorithms, games, and the internet. In *STOC 2001*, 2001.

[16] Y. Rekhter. A border gateway protocol 4 (BGP-4). IETF, RFC 1771.

[17] John W. Stewart. *BGP4: Inter-Domain Routing in the Internet*. Addison-Wesley, Reading, MA, 1999.

[18] Lakshminarayanan Subramanian, Sharad Agarwal, Jennifer Rexford, and R.H. Katz. Characterizing the internet hierarchy from multiple vantage points. In *Proc. IEEE INFOCOM 2002*, 2002.