Abstract—It is of great significance for network operators and researchers to obtain accurate AS-level traceroute paths, for which mapping IP addresses to correct AS numbers is critical. Thus, there have been a lot of efforts to improve the original IP-to-AS mapping table, which was extracted from BGP routing tables. One of these efforts is called pair matching, which refines the original mapping table by maximizing the number of matched pairs of traceroute and BGP AS paths. However, the existing pair-matching-based methods refine the original IP-to-AS mapping table only with the prefix granularity, i.e., IP addresses in the same /24 prefix are mapped to the same AS or the same set of ASes, which does not fit reality. In this paper, we attempt to refine the IP-to-AS mapping table with the IP address granularity, i.e., allowing IP addresses in the same prefix to be mapped to different ASes. The results show that our fine-grained method can produce a more accurate IP-to-AS mapping table. In addition, this paper also provides a better understanding for the pair-matching-based methods.

Keywords- IP-to-AS; Mapping; AS-level traceroute

I. INTRODUCTION

Tracing AS-level paths accurately may be one of the most desired techniques for network operators to debug networking failures and for researchers to measure the Internet’s topology. One obstacle to achieving accurate tracing is the IP-to-AS mapping, i.e., how to map an IP (Internet Protocol) address on an interface of a router to the AS (Autonomous System) where the router belongs. The original IP-to-AS mapping table, which was obtained by taking the pairs of original ASes and prefixes in BGP (Border Gateway Protocol) routing tables, has been proven to be error-prone [1]. There are mainly two kinds of approaches to improving the accuracy of the original mapping table: (a) alias-resolution-based method, i.e., inferring the AS that owns the router (e.g. by majority voting) after discovering the multiple IP addresses on the same router [2,3,4,5,6,7]; (b) pair-matching-based method, i.e., refining the IP-to-AS mapping table by maximizing the number of matched pairs of traceroute-derived AS paths and BGP AS paths from the same sources to the same destinations [8]. The former has received more interests and efforts than the latter. Recently, we have utilized the pair-matching-based method to quantify the pitfalls of using traceroute in AS adjacency inference [9,10], which suggests the existence of space for improving the 7-year-old work [8].

This paper focuses on the pair-matching-based method. Recall the existing best pair-matching-based method [8], in which an iterative scheme has been devised to reassign /24 prefixes to AS numbers by maximizing the number of matched path pairs. The granularity for reassigning and mapping is /24 prefix, i.e., that IP addresses in the same /24 prefix are mapped to the same AS or the same set of ASes. The prefix granularity is too coarse to be true in some cases. For example, a border router may reply to a traceroute probe using one of its interfaces whose IP address is shared and announced by one of its neighboring ASes. Moreover, as each global routable IP address should be configured to only one router, the IP-to-AS mapping should be determined without ambiguity. However in [8]’s method, one prefix in an Internet eXchange Point (IXP) has to be mapped to multiple ASes to enable path pairs matched; therefore mapping with ambiguity cannot determine exactly which AS is using which IP address.

Naturally, a simple idea of making the granularity of mapping finer emerges. As a follow-up to the previously mentioned work [8], we attempt to refine the IP-to-AS mapping table with the IP-address granularity, i.e., allowing IP addresses in the same prefix to be mapped to different ASes. Although the idea is simple, the improvement is great. Our fine-grained method can avoid the complicated threshold setting in [8] and achieve 9%~17% higher match ratio of path pairs in the refining datasets, which highlights the potential of pair-matching-based methods. Moreover, in terms of validation using some IXP information, our fine-grained method has a much higher accuracy than the other pair-matching-based methods and the alias-resolution-based method. Moreover, this paper also provides a better understanding for the pair-matching-based methods. We conduct comprehensive experiments and comparisons, and argue that the pair-matching-based method with the IP-address granularity is a powerful tool, but the small volume of path pair data restricts the accuracy of the refined IP-to-AS mapping table. Thus, we call for more data of path pairs to build a more accurate IP-to-AS mapping table.

[11] is a two-page short paper in the early stage of this work. In this version, we more comprehensively evaluate and analyze the pair-matching-based methods. And we give some validations in this version. The rest of this paper is organized as follows. We introduce the related work in Section II. We introduce the work of [8] in Section III. We describe our fine-grained method of refining the IP-to-AS mapping table in Section IV. We introduce how we collected data in Section V. We comprehensively evaluate and analyze the pair-matching-based methods in terms of the match ratio in Section VI. We give some validations for our fine-grained method in Section VII. We conclude this paper in Section VIII.
II. RELATED WORK

There are two kinds of work that is relative to refine the IP-to-AS mapping table: the alias-resolution-based work and the pair-matching-based work.

The alias-resolution-based work is as follows. The alias-resolution-based method first groups IP addresses that belong to the same IP-to-AS mapping table. Then, it maps the routers to the ASes [7]. There has been related work with alias resolution since 2004.

The pair-matching-based work is as follows. Y. Hyun et al. [12] pointed out that IXP was the main cause of mismatches of traceroute-BGP path pairs. Mao et al. [1] comprehensively analyzed the causes of mismatches of traceroute-BGP path pairs and designed a heuristic method to refine the optimal IP-to-AS mapping table. [1]’s method is labor-intensive, so Mao et al. designed a systematic method to refine the original mapping table in [8].

The alias-resolution-based method has received lots of interests. However, the pair-matching-based method has not been improved yet since 2004. Recently, we used the pair-matching-based method to quantify the pitfalls using traceroute in [9,10]. We wish for this research to improve the accuracy of the IP-to-AS mapping table further. The existing best pair-matching-based method of refining the IP-to-AS mapping table was devised by Mao et al. [8], but its /24 prefix granularity does not fit reality. Thus a simple idea is: we should use IP address granularity instead of prefix granularity to refine the original IP-to-AS mapping table. Although the idea is simple, it can achieve great improvement, which will be discussed in the later sections.

III. BACKGROUND

The underlying idea of refining the IP-to-AS mapping table based on matching path pairs is: it assumes that the forwarding path is consistent with the BGP AS path, and modifies the IP-to-AS mapping table by maximizing the number of matched path pairs. It is NP-hard to maximize the number of matched path pairs, so heuristic methods are devised to achieve as many matched path pairs as possible. In this section, we will introduce a pair-matching-based work [8] by Mao et al. Two important contributions are made in [8] as follows. One contribution is to devise a dynamic programming algorithm for computing the optimal matching for a path pair, as introduced in Subsection A. The other contribution is to devise a method for refining the IP-to-AS mapping table according to the information provided by the optimal matchings. We call this refining method, devised by [8], the prefix-granularity method with moderate ambiguity (PGMMA), as introduced in Subsection B. PGMMA is the existing best pair-matching based method for refining the IP-to-AS mapping table.

A. The Dynamic Programming Algorithm of Computing the Optimal Matching

A dynamic programming algorithm is devised by Mao et al. to compute the optimal matching for a path pair. The dynamic algorithm for computing the optimal matching can provide modification information for the IP-to-AS refining methods. A matching for a path pair is a mapping function from IP addresses of traceroute to ASes of the corresponding BGP AS path. The optimal matching for a path pair is the matching that has the minimum number of errors. Given a IP-to-AS mapping table M and a path pair (s, t), where s is the traceroute IP and t is the BGP AS path, the number of errors of a matching is equal to the number of IP addresses on the traceroute path whose mappings in M are different from their matched ASes on the BGP AS path plus the number of ASes on the BGP AS path that IP addresses fail to be matched to. The number of errors of a path pair is equal to the number of errors of the optimal matching of this path pair. Because of space restriction, we refer readers to [8] for the details of the dynamic programming algorithm for computing the optimal matching. Here we use a figure example to illustrate the optimal matching as shown in Fig. 1. The traceroute path is (IP1, IP2, IP3, IP4) and the BGP AS path is (AS1, AS2, AS3). M(IP) denotes the set of ASes that the IP address is mapped to in the IP-to-AS mapping table M. The lines without arrows denote the optimal matching of this path pair. The number of errors of this optimal matching is 1. There is no any other matchings whose errors are less than 1.

B. Prefix-granularity Method with Moderate Ambiguity (PGMMA)

PGMMA is the existing best pair-matching-based method for refining the IP-to-AS mapping table. PGMMA uses the granularity of /24 prefix to modify the IP-to-AS mapping table by exploiting the information from optimal matchings. One /24 prefix may be matched to different ASes in different optimal matchings, but not all matched ASes in optimal matchings should be assigned to the /24 prefix as mapping. This is because some of the matched ASes in optimal matchings might be noises. PGMMA devises a systematic method based on threshold control to eliminate those noise ASes from the matched ASes in optimal matchings.

That threshold control needs to be manually set and is complex. Because of space restriction, we refer readers to [8] for the details of PGMMA. For PGMMA, prefixes are split into /24 prefixes. Nevertheless, there are still many /24 prefixes that are mapped to multiple ASes. In [8], if a path pair can be matched by using any one AS of these multiple ASes, the path pair will be considered as match. However, the path pair is actually not an exact match, and we call it an ambiguous match, which will be described in detail in Section IV.

IV. METHODOLOGY

IP addresses in the same /24 prefix may belong to different ASes, which is due to causes, such as border-AS routers and IXPs. PGMMA maps such /24 prefixes to sets of ASes, which imposes some extent of ambiguity, and it may even fail to add legitimate ASes to mappings due to failing to satisfy the
A. Definitions of Exact Match, Ambiguous Match and Mismatch

In this subsection, we will introduce the definitions of exact match, ambiguous match and mismatch.

**Exact Match:** We say a path pair is an exact match if each IP address of the path pair is mapped to one unique AS and the number of errors of the optimal matching for this path pair is zero.

**Ambiguous Match:** We say a path pair is an ambiguous match if at least one IP address of the path pair is mapped to two or more ASes and the number of errors of the optimal matching for this path pair using one of the multiple mapped ASes is zero. Specifically, if a mapping of a prefix contains multiple ASes, we call the mapping an ambiguous mapping.

**Mismatch:** We say a path pair is a mismatch if the number of errors of the optimal matching for this path pair using any mapped ASes cannot be zero, i.e. the path pair does not satisfy the conditions of exact match and ambiguous match.

B. Two Extremes of PGMA

Ambiguous matches were considered as matches in [8], whereas ambiguous matches are not real matches. Obviously, the fraction of ambiguous matches depends on the thresholds. If the thresholds add more ASes to the mappings, the fraction of ambiguous matches will be larger, and the fraction of the sum of exact matches and ambiguous matches will also be larger, but the fraction of exact matches will be smaller. To comprehensively compare our method with PGMA, we introduce two extremes of PGMA. One extreme of PGMA is that each prefix is only mapped to one single AS that is most matched to by the dynamic programming algorithm among all the ASes. In this extreme, no ambiguous matches will be generated. We call this extreme of PGMA: prefix-granularity method without ambiguity (PGMA). The other extreme of PGMA is that each prefix is mapped to all its optimally matched ASes by the dynamic programming algorithm. In this extreme, there will be a large fraction of ambiguous matches, so we call this extreme of PGMA: prefix-granularity method with great ambiguity (PGMGA).

C. IP-granularity Method (IGM)

In this subsection, we will introduce our improved method: the IP-granularity method (IGM). Our key insight is that the refining granularity should be the IP address granularity instead of the prefix granularity. Although this insight is simple, the improved performance is great as discussed in the latter sections. IGM uses the same dynamic programming algorithm as PGMA. We first define the score of an AS for a given IP address, which is: the number of path pairs in which the IP address is matched to the AS by the dynamic programming algorithm. The following two steps are repeated by IGM until the refined IP-to-AS mapping table keeps unchanged in a repeat process.

**Step 1:** Compute the optimal matchings for all the path pairs by using the dynamic programming algorithm. The IP-to-AS mapping table used by the dynamic programming algorithm is the refined IP-to-AS mapping table in the current iteration. The original IP-to-AS mapping table is provided as the initial IP-to-AS mapping table at the first iteration.

**Step 2:** For each IP address, the AS with the highest score is chosen. If the chosen AS is different from the one in the current IP-to-AS mapping, the mapping of the IP address is updated with the chosen AS. An IP address can be regarded as a /32 prefix, so updating the mapping of an IP address is equivalent to updating the mapping of the corresponding /32 prefix or creating one if lack of the /32 prefix mapping.

The pseudo code of IGM is as shown in Fig. 2. Now we describe the meanings of the variables in the pseudo code of IGM. We use Mat[IP] to denote the AS that the IP address is matched to by the dynamic programming algorithm, and use OM to denote the refined IP-to-AS mapping table, and use RM to denote the original IP-to-AS mapping table, and use pairs[i] denote the i-th path pair, where the number of path pairs is L. RM is initially equal to OM. We use IP/32 to denote the /32 prefix of the IP address. TM[IP/32] and OM[IP/32] hit the mapping of the IP/32 prefix in the mapping table. We use Count[IP][AS] to denote the number of path pairs, in which the IP is optimally matched to the AS by the dynamic programming algorithm. The initial value of Count[IP][AS] is zero.

The IP-to-AS mapping table refined by IGM contains two groups of mappings: (a) the IP-granularity mappings (i.e. mappings of /32 prefixes) created by IGM; (b) the original mappings. The IP-to-AS mapping table is looked up for a given IP address according to the longest prefix matching rule.
Obviously, the IP-granularity mappings created by IGM have the highest priority to be matched.

D. Comparative Analysis

PGMMA was the best pair-matching-based method before IGM was put forward by this paper. The fundamental difference between them is the granularity of modifying the IP-to-AS mapping. IGM uses IP address as granularity while PGMMA uses /24 prefix as granularity. In the aspect of running time, IGM and PGMMA are approximately equal. Both of them take less than ten hours to process the entire 1.4M path pairs on a 2.5Ghz processor. IGM is even a little faster than PGMMA. It is because IGM needs less iterations than PGMMA. In the aspect of the space cost, IGM and PGMMA are also approximately equal. The number of IP addresses is approximately as only twice as the number of /24 prefixes in the path pair data. It is because only a small number of IP addresses are used on routers for most /24 prefixes. In the aspect of simplicity of method, IGM is simple while PGMMA is complex as the complex thresholds must be manually set. In the aspect of accuracy, IGM can achieve greater accuracy compared with PGMMA as shown in the later sections.

V. DATA COLLECTION

The tasks of data collection are: (a) obtaining the original IP-to-AS mapping table, which is used as the initial refining point; (b) generating traceroute-BGP path pairs from traceroute probes and BGP routing tables.

A. Obtaining the Origin Mapping Table (OM)

We collected routing tables on 2010-04-22 from ten collectors of Routeviews [13] and RIPE [14]. We then extracted prefixes and origin ASes from routing entries. The original IP-to-AS mapping table was obtained by mapping prefixes to their corresponding origin ASes. In total, there are 371K prefixes in the original IP-to-AS mapping table, and 4000 out of the 371K prefixes are mapped to multiple ASes due to the multiple original ASes (MOAes).

B. Generating Traceroute-BGP Path Pairs

To generate traceroute-BGP path pairs, we used the traceroute monitors and BGP monitors that were located in the same ASes. There are four such pairs of traceroute and BGP monitors in CAIDA [15], Routeviews [13] and RIPE [14]. We collected traceroute probes from the four traceroute monitors and BGP routing tables from the four BGP collectors on 2010-04-22. The procedure for generating traceroute-BGP path pairs is as follows.

Processing traceroute probes: We extracted traceroute paths and destination IP addresses from the traceroute probes.

Processing BGP routing tables: We extracted prefixes and BGP AS paths from the BGP routing tables. We then associated the prefixes with the corresponding BGP AS paths. We removed duplicate ASes of the BGP AS paths due to padding.

Generating traceroute-BGP path pairs: For each traceroute path, we looked up the longest matched prefix for the destination IP address of the traceroute path. The traceroute path and the BGP AS path that the longest matched prefix is associated with formed a traceroute-BGP path pair.

The summary of the generated traceroute-BGP path pairs is shown in Table I. We now describe the meaning of each column in Table I. The first column presents the collectors where traceroute probes are collected. The second column presents the collectors where BGP routing tables are collected. The third column presents the ASes where traceroute and BGP collectors are located. The fourth column presents the number of path pairs that are generated from the traceroute probes and the BGP routing tables. The last column presents the number of IP addresses that are contained in the path pairs.

VI. EVALUATIONS FROM MATCH RATIOS

We have four monitors of traceroute-BGP path pairs. We use three of the four as the refining dataset, which is used to refine the IP-to-AS mapping table, and use the last one as the test dataset, which is used to test the refined IP-to-AS mapping table. There are four combinations of refining and test datasets, which are shown in Table II. ‘Group ID’ column in Table II indicates the ID of the group of the refining and test dataset. We now give the definitions of match ratio, mismatch ratio, refining match ratio and test match ratio, which are terms that will be used often in the rest of this paper.

Match ratio: Match ratio refers to the ratio of matched path pairs to the total path pairs.

Mismatch ratio: Mismatch ratio refers to the ratio of mismatched path pairs to the total path pairs.

Refining match ratio: We run the IP-to-AS refining algorithm on the refining dataset, where the match ratio is called the refining match ratio.

Test match ratio: We apply the refined IP-to-AS mapping table to the test dataset, where the match ratio is called the test match ratio.

In Subsection A, we will evaluate the refining algorithms in terms of the refining match ratio. In Subsection B, we will evaluate the refining algorithms in terms of the test match ratio. In addition, we will investigate the limitations of IP granularity and prefix granularity. In Subsection C, we will investigate the potential to further improve the test match ratio for the refining methods. Based on our assumption that the forwarding path is consistent with the BGP AS path, the higher match ratio of path pairs the IP-to-AS mapping table can achieve, the more accurate the IP-to-AS mapping table is.
A. The Refining Match Ratios

In this subsection, the match and the mismatch refer to the ones in the refining datasets unless otherwise stated.

The original IP-to-AS mapping table can only make the path pairs of refining datasets achieve the match ratio of around 60%. We run the IP-to-AS refining methods on the refining datasets to refine the original IP-to-AS mapping table. The match ratios that the refined IP-to-AS mapping tables make the path pairs of the refining datasets achieve are shown in Fig. 3, the horizontal axis denotes the IP-to-AS refining methods, and the vertical axis denotes the ratio to total path pairs. We use different filling styles to distinguish exact matches and ambiguous matches. We find that IGM achieve a higher match ratio of around 13% than PGMMA, even if ambiguous matches of PGMMA are considered as matches. PGMGA has higher match ratio than IGM, but almost all the matches for PGMGA are ambiguous matches, which have little significance.

We then inspect the refining methods for the changes to the original IP-to-AS mapping table: IGM modifies around 21319 (7%) IP addresses’ mappings; PGMNA modifies around 4613 (3%) /24 prefixes’ mappings; PGMMA modifies around 4616 (3%) /24 prefixes’ mappings, and 915 (0.7%) out of them are ambiguous mappings; PGMGA modifies 8597 (6.6%) /24 prefixes’ mappings, and 8591 (6.6%) out of them are ambiguous mappings.

In general, the forwarding path is consistent with the BGP AS path, but why cannot the refined IP-to-AS mapping tables make the path pairs of the refining datasets achieve the match ratio of 100%? We analyze the reasons as follows.

The mismatch ratio of around 3% using the IP-to-AS mapping table refined by IGM: Around 50% of mismatched pairs that IGM cannot correct have three or more errors, so they are obviously inconsistent. It implies that around 3% of path pairs, which are mismatches, may be truly inconsistent due to the restriction of collecting data and routing instability.

The mismatch ratio of around 23% using the IP-to-AS mapping table refined by PGMNA: Except around 3% of path pairs, which are mismatches, are due to the inconsistency of path pairs; the last 20% of path pairs, which are mismatches, are due to the bind flaw of prefix, i.e., mapping IP addresses in the same prefix to the same AS. Maybe some IP addresses in the same prefix should be mapped to other different ASes.

The mismatch ratio of around 16% using the IP-to-AS mapping table refined by PGMMA: PGMMA permits prefixes to be mapped to multiple ASes to alleviate the bind flaw of prefix, so compared with PGMNA, PGMMA can lower the mismatch ratio by around 7%, but reversely it will reduce the exact match ratio by around 9%.

No mismatched pairs using the IP-to-AS mapping table refined by PGMGA: PGMGA makes prefixes to be mapped to all possible ASes (i.e. all the ASes that the IP addresses in the prefixes are optimally matched to by the dynamic programming algorithm), so PGMGA can achieve the match ratio of 100%, but almost all the matches of PGMGA are ambiguous matches, which have little significance in practice.

From Fig. 3, we find that PGMMA can only achieve the match ratio of 80%~87%, which is far from the match ratio of 95% [8] referred to. To find the reason why there is such a big difference, we ran PGMMA and IGM on the four separate datasets to observe the match ratios that PGMMA and IGM can achieve. The results are shown in Table III. The number in each cell indicates the percentage of matched path pairs of the total path pairs. To be identical with [8], we also consider ambiguous matches as matches, but we mark the percentage of ambiguously matched path pairs of the total path pairs in the parentheses. We find that IGM stably achieves the match ratio of more than 96% on the four separate datasets, whereas PGMMA achieves the match ratio of 96% on the dataset of AS4538, but it only achieves the match ratio of 76% on the dataset of AS6939. Why do the results of PGMMA have such a difference in the two datasets of AS4538 and AS6939? It is because, in the dataset of AS6939, there are more cases that the IP addresses in the same /24 prefixes should be mapped to many other different ASes than those of AS4538. Because of the bind flaw of prefix, PGMMA cannot correct such cases well. Thus, whether PGMMA can achieve a high match ratio on the refining dataset depends on the characteristic of the dataset.

Of course, PGMMA can raise the match ratio by adjusting the threshold. By raising the threshold, the match ratio will decrease, but the exact match ratio will increase. Conversely, by lowering the threshold, the match ratio will increase, but the exact match ratio will decrease. It is noted that matches for PGMMA include exact matches and ambiguous matches. An important threshold parameter of PGMMA is denoted by $r$, which adjusts mappings as follows. If the proportion of an AS that a prefix $p$ is matched to by the dynamic programming algorithm is larger than $r$ and $|M(p)|>1$, where $|M(p)|$ denotes the number of mappings of the prefix $p$, the AS will be added to the mappings of the prefix $p$. The threshold parameter $r$ is set to be 0.1 in [8]. To raise the match ratio of PGMMA, we can lower the threshold as follows. We remove the condition of $|M(p)|>1$ and decrease the threshold $r$ of PGMMA from 0.1 to 0.001. Taking the second refining dataset for example, as shown in Fig. 4, the horizontal axis denotes the threshold $r$ of PGMMA. It is noted that the horizontal axis is arranged in the inverted sequence. The vertical axis denotes the percentage over the total path pairs. We find that if the threshold is lowered, the increasing of the match ratio is much lesser than
the decreasing of the exact match ratio. Thus, it is not worth
continuing to lower the threshold to raise the match ratio. The
results of the other three refining datasets show the same
findings.

B. The Test Match Ratios

In this subsection, the match and the mismatch refer to the
ones in the test datasets unless otherwise stated.

In the previous subsection, we have refined the IP-to-AS
mapping tables using the IP-to-AS refining methods. We apply
the refined IP-to-AS mapping tables to the corresponding test
datasets. The match ratios that the refined IP-to-AS mapping
tables make the path pairs of the test datasets achieve are as
shown in Fig. 5, the horizontal axis denotes the IP-to-AS
refining methods, and the vertical axis denotes the match ratio
to total path pairs. We use different filling styles to distinguish
exact matches and ambiguous matches. We find that the IP-to-
AS mapping tables refined by the refining methods all improve
the match ratios compared to the original mapping table (OM).
The match ratio of IGM is higher than the ones of PGMNA
and PGMMA. Although the match ratio of PGMGA is higher
than the one of IGM, almost all the matches of PGMGA are
ambiguous matches, which are of little significance in practice.
Why are the match ratios in the former two test datasets (they
are AS4538 and AS7660) much higher than the ones in the
latter two test datasets, as shown in Fig. 5? It is because
AS4538 is the customer of AS7660, and there is great
similarity between the two datasets, so AS4538 and AS7660
are mutually refined and tested.

However, what do the mismatches, as shown in Fig. 5, stem
from? There are two possible situations of mismatches: (a) path
pairs are originally matched but are changed to be mismatched
using the refined mapping table, i.e., the match-mismatch path
pairs in Fig. 6; (b) path pairs are originally mismatched and are
still mismatched using the refined mapping table, i.e., the
mismatch-mismatch path pairs in Fig. 6. To know which
situation is dominant, we inspect the path pairs for the
transitions between matches and mismatches, and divide path
pairs into four parts, as shown in Fig. 6. In Fig. 6, the match or
mismatch before the hyphen means that a path pair is a match
or mismatch using the original mapping table. Similarly,
the match or mismatch after the hyphen means that a path pair
is a match or mismatch using the refined mapping table. From
Fig. 6, we find that for each of all the above refining methods,
there are a very small proportion of match-mismatch path pairs.
It implies that these refining methods all modify mappings to
the direction of matches, and the mismatches in Fig. 5 are
mainly from the second situation instead of the first situation.

C. Potential to Improve the Test Match Ratio

In this subsection, we inspect the refining methods for the
change of the test match ratio by increasing the volume of the
refining data gradually in order to cover more IP addresses in
the test dataset. We used the data of AS2152 as the test dataset,
and added the data of AS7660, AS4538, AS6939 and AS2152
one by one to the refining dataset, and observed the change of
the test match ratio. The results are shown in Fig. 7. The
horizontal axis denotes the proportion of IP addresses in the
test dataset that also appear in the refining dataset. The
vertical axis denotes the test match ratio. We find that the test
match ratio of IGM greatly increases as the proportion of IP
addresses that also appear in the refining dataset. In contrast,
the test match ratios of PGMMA and PGMNA increase
slightly even if the ambiguous matches of PGMMA are
considered as matches. IGM has greater potential to improve
the accuracy of the IP-to-AS mapping table than PGMMA and
PGMNA. Thus, we call for more data of path pairs to build a
more accurate IP-to-AS mapping table using IGM.
The key point of validating the refined IP-to-AS mapping table is to obtain IP-to-AS truths. Internal IP-to-AS truths are hard to be obtained due to the lack of access permissions. Fortunately, PeeringDB [16], which is a public website where volunteers maintain some information about IXP, provides the corresponding ASes for some IP addresses. Although these correspondences between IP addresses and ASes may not be completely true due to information being not up-to-date, we can approximately consider these correspondences to be our IP-to-AS truths, which can be used to validate one part of the refined IP-to-AS mapping table. In this section, the refined IP-to-AS mapping tables were obtained by running the refining methods on the data of all the four monitors. IGM achieved the match ratio of 97%, PGMNA achieved the match ratio of 82%, and PGMMA achieved the match ratio of 76%, and PGMMA achieved the match ratio of 3%~6% in the test datasets than PGMMA. Moreover, in terms of validation using the IXP information, IGM has a much higher accuracy than the other pair-matching-based methods and the alias-resolution-based method. In addition, we provide a better understanding for the pair-matching-based methods through comprehensive experiments and analysis.

<table>
<thead>
<tr>
<th></th>
<th>OM</th>
<th>IGM</th>
<th>PGMNA</th>
<th>PGMMA</th>
<th>Alias-resolution-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>4</td>
<td>569</td>
<td>24</td>
<td>39 (30)</td>
<td>86</td>
</tr>
<tr>
<td>False</td>
<td>629</td>
<td>64</td>
<td>609</td>
<td>594</td>
<td>288</td>
</tr>
</tbody>
</table>

VIII. VALIDATING REFINED MAPPINGS

The way of obtaining the alias-resolution-based IP-to-AS mapping is as follows: We first collected the IP-to-router topology derived from alias-resolution methods MIDAR [17] and iffinder [18] from CAIDA, and we removed the routers that have less than three interfaces. Finally we obtained 120221 groups of routers, which included 508690 IP addresses. We then collected the router-to-AS assignments derived from the method [7] from CAIDA. We then generated the alias-resolution-based IP-to-AS mapping table by combining the IP-to-router topology and router-to-AS assignments.

The results of validation are shown in Table IV. The ‘True’ in the row header denotes that a given case can pass the validation. The ‘False’ in the row header denotes that a given case cannot pass the validation. The column headers denote the respective IP-to-AS mapping tables. The number in each content cell denotes the number of cases that are hit by the corresponding column header and row header. There are 633 IP addresses, which appear both in the refining dataset and the dataset of IXP truths. There are 374 IP addresses, which appear both in the alias-resolution-based dataset and the dataset of IXP truths. For the original IP-to-AS mapping table (OM), only 4 out of the 633 cases can pass validation, which implies that the original mappings of IXP IP addresses are incorrect in general. For the IP-to-AS mapping table refined by IGM, 569 out of the 633 cases can pass validation. For the IP-to-AS mapping table refined by PGMNA, only 24 out of the 633 cases can pass validation. For the IP-to-AS mapping table refined by PGMMA, only 39 out of the 633 cases can pass validation.

VIII. CONCLUSION

PGMMA is too coarse to be true in some cases while IGM is fine-grained and can solve the problem well. Although the idea of IGM is simple, the improvement is great. In terms of the match ratio of path pairs, IGM can achieve a higher match ratio of 96%~97% in the refining datasets and a higher match ratio of 3%~6% in the test datasets than PGMMA. Moreover, in terms of validation using the IXP information, IGM has a much higher accuracy than the other pair-matching-based methods and the alias-resolution-based method. In addition, we provide a better understanding for the pair-matching-based methods through comprehensive experiments and analysis.

ACKNOWLEDGEMENT

This research is supported by the National High-tech R&D Program ("863" Program) of China (No.2013AA010605), the National Science Foundation of China (No.61073172), and National Key Basic Research Program ("973" Program) of China (No.2009CB320501). Jun Bi is the corresponding author.

REFERENCES

[16] IXP data: http://www.peeringdb.com
[18] Iffinder: http://www.caida.org/tools/measurement/ifinder