

Measuring Internet Routing from the Most Valuable Points

Thomas Alfroy*, Thomas Holterbach*, Thomas Krenc†, KC Claffy†, Cristel Pelsser‡
 *University of Strasbourg, †CAIDA / UC San Diego, ‡UCLouvain

<https://bgproutes.io>

Abstract

While the increasing number of Vantage Points (VPs) in RIPE RIS and RouteViews improves our understanding of the Internet, the quadratically increasing volume of collected data poses a challenge to the scientific and operational use of the data. The design and implementation of BGP and BGP data collection systems lead to data archives with enormous redundancy, as there is substantial overlap in announced routes across many different VPs. Researchers thus often resort to arbitrary sampling of the data, which we demonstrate comes at a cost to the accuracy and coverage of previous works. The continued growth of the Internet, and of these collection systems, exacerbates this cost. The community needs a better approach to managing and using these data archives.

We propose MVP, a system that scores VPs according to their level of redundancy with other VPs, allowing more informed sampling of these data archives.

Our challenge is that the degree of redundancy between two updates depends on how we define *redundancy*, which in turn depends on the analysis objective. Our key contribution is a general framework and associated algorithms to assess redundancy between VP observations. We quantify the benefit of our approach for four canonical BGP routing analyses: AS relationship inference, AS rank computation, hijack detection, and routing detour detection. MVP improves the coverage or accuracy (or both) of all these analyses while processing the same volume of data.

1 Introduction

Routing information services such as RIPE RIS [38] and RouteViews (RV) [51] continuously collect and publish data from more than 2500 Vantage Points (VPs), each of which is a BGP router that exports its best routes to the collection platform. These data collection systems are critical to scientific as well as operational analyses of the global Internet infrastructure. But these systems face a cost-benefit trade-off [2]. The information-hiding character of BGP means that improving

the visibility of the Internet routing system requires cultivating many peering relationships with operators willing to contribute VPs to the platform. However, deployment of new VPs amplifies the data management requirements caused by the growth of the Internet itself: the number of unique IP prefixes (e.g., due to de-aggregation or new assignments) constantly grows [12], as well as the number of unique ASes and links between them. Even with a constant number of VPs, the volume of routing data inevitably increases, contributing to a quadratic increase of observed updates over time (Fig. 2a). The situation presents a challenge for users, who often cannot or do not want to process terabytes of (redundant) data. Users often resort to sampling the data in arbitrary ways, such as grabbing all VPs on a single collector.

We design and implement a framework to optimize the use of these data collection systems, which will also lower the barrier to their use in lower-resourced circumstances. Our design relies on the principle of *redundancy* in BGP data, a delicate concept since even two identical updates from two different VPs may not be redundant (depending on the use case). We take a deep dive into a context-specific framework for quantifying redundancy in BGP data, grounded in operational principles and research use cases. Our resulting system identifies a set of VPs whose exported routes collectively exhibit a low level of redundancy—enabling users to prioritize the processing of the most valuable BGP updates.

Contributions. We make the following contributions.

- We perform a comprehensive analysis based on simulations and a survey that demonstrates the cost-benefit tradeoff of setting up new VPs, and the value of strategically selecting them to analyze Internet routing. We show that current approaches used by researchers to select VPs are largely unoptimized, sacrificing coverage and accuracy of a wide range of measurement studies and tools (§2-§3).
- We characterize redundancy between updates collected by different VPs. We explore different definitions of redundancy and find that optimizing our algorithms for a given definition leads to a undesirable overfitting effect (§4-§5).

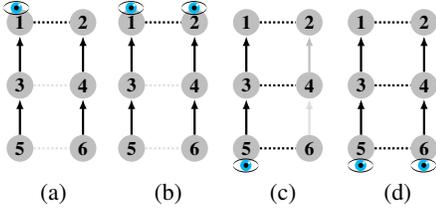


Figure 1: Combining local views can help to map the AS topology. Gray links are not visible from routes collected by VPs (👁).

- We design a system, MVP, that returns a list of the “most valuable” VPs, i.e., those that enable users to minimize data redundancy (regardless of how we define it) and prioritize valuable route updates. MVP relies on new data-driven algorithms that quantify redundancy between VPs based on the four main BGP attributes (time, prefix, AS path, and communities) while being robust against typical biases observed in the Internet routing ecosystem (§6-§7).
- We run MVP as a service at <https://bgproutes.io>. We benchmark MVP and show that it optimizes (without overfitting) the tradeoff between the volume of data used and its utility for many objectives (§8-§9).

Impact on scientific measurement studies. The value of MVP is its *wide* impact. Besides enabling a more systematic sampling of the RIS and RV data archives, it can consistently, and at no cost for users, improve the accuracy and coverage of measurement studies as well as monitoring tools fueled by BGP routes collected by RIS and RV. To measure the impact of MVP, we replicated the algorithms used in four studies/tools and used MVP to select the VPs from which they process BGP routes. In all four cases, using MVP improved the accuracy and coverage while processing the same data volume. We inferred more AS relationships (+15%), fixed errors in the AS rank dataset, observed more routing detours (+44%) while characterizing them more accurately, and inferred more forged-origin hijacks (+35%) with $\approx 4\times$ less incorrect inferences (i.e., false positives).

2 Background

RIPE’s Routing Information Service (RIS) [38] and RouteViews (RV) [51] are two widely-used platforms that collect BGP routes and make them available to the community. These platforms use BGP speakers (*a.k.a.* collectors) to peer with BGP routers in order to collect routes exported by those routers. We call **vantage points (VPs)** the BGP routers that export their routes to a collection platform. As of May 2023, 32% of the RIS and RV VPs [33, 42] are *full feeders*, i.e., they send a route for roughly all of the announced IP prefixes on the Internet ($\approx 941k$ prefixes [12]). A BGP route mainly

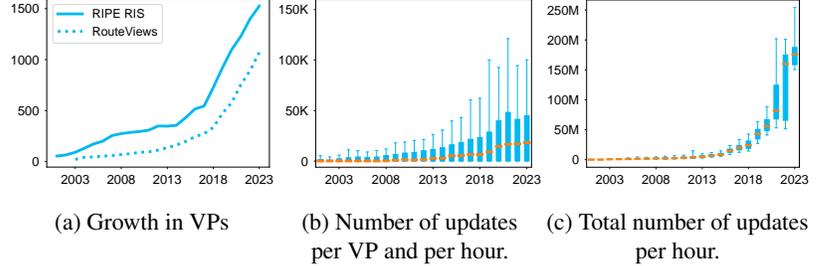


Figure 2: The number of VPs increases over time and so does the number of collected updates. Both RIS and RV are considered in Fig. 2b and 2c.

carries routing information in four of its attributes [36]: (i) the timestamp at which the route was received, (ii) the IP (v4 or v6) prefix that the route announces, (iii) the AS path used to reach that prefix, and (iv) a set of BGP communities. Among other uses, researchers leverage the timestamp to find transient paths [29], the prefix to detect hijacks [44], the AS paths to infer AS relationships [31], and the communities to measure unnecessary BGP traffic [28].

Each VP provides its *local view*, i.e., only the BGP routes it observes. Fig. 1 illustrates the effect of combining local views for inferring the AS topology from the AS paths in BGP routes. In Fig. 1, every AS runs a single BGP router, owns one prefix, and announces it in BGP. We configure routing policies based on the Gao-Rexford model [21], i.e., routing paths follow a valley-free pattern. Straight (resp. dashed) lines are customer-to-provider (resp. peer-to-peer) links. With the local view of ①, one can infer all the AS links but the two peering links ③—④ and ⑤—⑥ (Fig. 1a). Combining the local views of ① and ② does not help to discover more links (Fig. 1b). With the local view of ⑤, one can infer all the AS links but the two customer-to-provider links ②—④, ④—⑥ (Fig. 1c). Combining the local views of ⑤ and ⑥ enables discovery of the full topology (Fig. 1d). However, observe that this last scenario is unlikely in practice as the location of the VPs is skewed with many more VPs present in highly-connected or central (e.g., Tier1) ASes [45]. Observe also that VPs can have a redundant view over the AS topology, e.g., the two VPs in Fig. 1b observe the same set of links.

By May 2023, RIS had 1526 VPs and RV had 1071 VPs, and their number keeps increasing (Fig. 2a). Users can download BGP routes exported by these platforms at the granularity of the VP (with some limitations [41]) or the collector. Users can download a RIB dump, i.e., a snapshot of the BGP routes seen by a VP at a particular time, which (in Jan. 2023) yielded $\approx 941k$ routes for a full feeder. Alternatively, users may download every single BGP update observed by the VPs over time (e.g., using [36]), which currently results in $\approx 18K$ updates per hour (median in May 2023) for a single VP (Fig. 2b), and billions of updates per day for all RIS and RV VPs (Fig. 2c).

3 Problem

Deploying more VPs expands the visibility of the routing system (§3.1), but also increases collected data volumes raising barriers to its use (§3.2). We survey researchers and find that they resort to unoptimized sampling, which they acknowledge can negatively impact the quality of their results (§3.3).

3.1 More VPs improves data completeness

A tiny fraction (1.3%) of the 74k ASes participating in the global routing system [12] host a VP. This fraction remains low (8.4%) even when focusing on the 11441 transit ASes (i.e., those with at least one customer). While we cannot know how much additional topology we might observe from VPs that do not peer with the public collection systems, we can estimate this gap using simulations of topologies whose statistical parameters match those of the known global Internet.

Methodology. We created a mini-Internet with 600 ASes, each running a single BGP router. We generated the AS topology using the Hyperbolic Graph Generator [3]. We set the average node degree to 6.1, which results in a comparable degree of connectivity (*a.k.a.* Beta index) to the one observed in CAIDA’s AS relationship dataset from December 2022 [16], and use as the degree distribution a power law with exponent 2.1 (as in [3]). We defined the AS relationships as follows. The three ASes with the highest degree are Tier1 ASes and are fully meshed. ASes directly connected to a Tier1 are Tier2s. ASes directly connected to a Tier2 but not to a Tier1 are Tier3s, etc. Two connected ASes have a peer-to-peer (p2p) relationship if they are on the same level, and a customer-to-provider (c2p) relationship if not. The routing policies follow the Gao-Rexford model [21].

Fig. 3 shows the proportion of observed AS links as a function of the number of ASes hosting a VP. We consider three VP deployment strategies: (i) *random*, which randomly deploys VPs across all the ASes; (ii) *distance-based*, which aims to maximize the AS-level distance between the deployed VPs; and (iii) *greedy specific*, which approximates the best case for topology discovery using a greedy approach. We ran every selection strategy twenty times (with different random seeds). We computed the proportion of observed links and show separately the p2p and c2p links in Fig. 3.

Conclusions. Although we take the results with a grain of salt because the topology differs (but exhibits similar patterns) from the visible portion of the actual (unknown) AS topology, we tentatively draw the following four conclusions.

- (I) As expected, for a given VP deployment strategy, more VPs often lead to more links observed; all links are observed only when all ASes host a VP.
- (II) P2p links are harder to observe than c2p links. We find that p2p links are more visible from VPs at the edge. This result is consistent with the fact that p2p links are generally

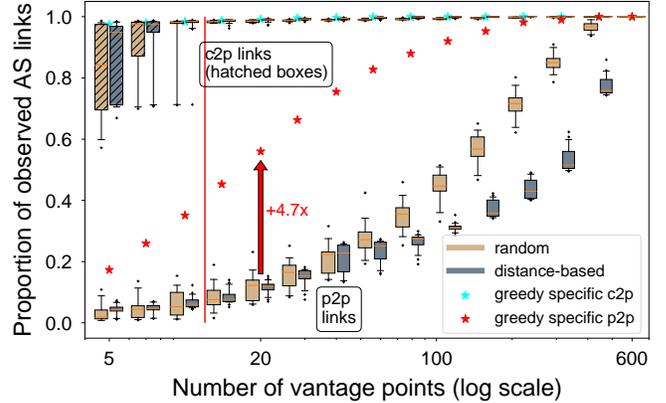


Figure 3: Simulations of a mini Internet with 600 ASes. We make two key observations: (i) deploying more VPs helps to reveal more AS links, and (ii) arbitrarily selecting VPs performs poorly compared to selecting them with *greedy specific* (a best-case approximation). The line in a box depicts the median value; the whiskers show the 5 and the 95th percentile.

not advertised upwards in the Internet hierarchy when routing policies follow the Gao-Rexford model.

(III) The *distance-based* deployment strategy performs poorly (even worse than random) because it overprioritizes isolated VPs at the edge over some other important VPs in the core.

(IV) When 1.3% of ASes host a VP (same proportion as current RIS and RV VPs), only $\approx 5\%$ of the p2p links are seen when using the *random* deployment strategy.

Confirmation with real (but private) data. We contacted a private BGP data provider (bgp.tools) that collects BGP routes from ≈ 1000 routers and compared the set of AS links observed from these private feeds against the set of AS links observed by RIS and RV VPs (in September 2023). We find that the private data provider saw 192k AS links that *none* of the RIS and RV VPs observed, and vice versa, RIS and RV VPs observed 401k links that the private data provider did not observe. In either case, the lack of VPs leads to missing routing information. We can thus expect—and hope for—the number of VPs to keep increasing.

3.2 BGP data management is challenging

Deploying more VPs generates more data as each of them collects BGP updates. Moreover, new IP prefixes advertised in BGP (see [12]) increase the volume of data collected by every VP as it triggers the propagation of new BGP routes that many VPs (e.g., the full feeders) observe and send to the collection platforms. The compound effect—more VPs (Fig. 2a) and more updates per VP (Fig. 2b)—yields a *quadratic* increase in updates reaching the collection platforms (Fig. 2c), which challenges both users and data providers [2]. Although several tools can speed up data processing [5, 7, 36], many

measurement studies and monitoring tools use only a sample of data collected by RIS and RV, either using only a subset of the VPs or a short time window, or both¹. While authors do not typically explain why they do not use all the data, the sampling suggests two (inter-related) explanations: authors believe the sample is representative and sufficiently complete; and/or the data volume is not worth trying to manage. We confirm these explanations with a survey that we conducted involving authors of eleven research papers.

Methodology of our survey. We classified eleven BGP-based studies from top conferences² into two categories based on how they used BGP data.³ Nine papers used **all** routes collected from a **subset** of the VPs (category C_1); six papers used a **short time frame** (C_2). For each paper, we asked authors questions regarding their use of BGP data: whether data volume limited their work, how and why they sampled BGP data sources, their understanding of the impact on the quality of their results, and if they would do things differently if they had more resources or time. We did not receive answers from the authors of three papers. Thus, we have seven respondents in C_1 and five in C_2 . We summarize the results here; details of the survey are in an appendix (§A).

The volume of BGP data to process is often a limiting factor. Seven (of eight) respondents found the BGP data expensive to process. For three respondents in C_1 , processing time motivated them to use only a subset of the VPs; three respondents in C_2 considered the processing time when choosing a measurement interval. Even a respondent who used a Spark cluster found it inhibitive time-consuming to process the BGP data.

Respondents in C_1 selected VPs in an unoptimized fashion. One respondent picked geographically distant BGP collectors. Our experiments (Fig. 3) and evaluation (§9) show that this strategy, while intuitive, often fails to optimize for any given metric (e.g., coverage). Other respondents said they chose VPs randomly, or those with the highest number of prefixes. Another responded to have unintentionally discarded some VPs, leaving an arbitrarily selected set in the study. Two respondents did not remember how they selected VPs.

3.3 Unoptimized sampling negatively impacts the quality of the results

We show the negative effects of an unoptimized sampling using our controlled simulations as well as our survey.

Selecting VPs arbitrarily performs poorly. Our mini-Internet simulation (§3.1) showed that arbitrary VP selection strategies perform significantly worse than *greedy specific* (a best-case approximation) when the goal is to map the AS topology. For instance, randomly selecting 20 VPs reveals 12% of the p2p links compared to 56% when selecting them

¹We purposively do not cite any paper to preserve the anonymity of the respondents of our survey.

²SIGCOMM, NSDI, S&P, USENIX Security, NDSS and IMC.

³A paper may be in both categories.

using *greedy specific*—a 4.7× improvement factor that we highlight in Fig. 3. Our evaluation reveals that this performance gap between using an arbitrary VPs selection strategy and a best-case approach also exists for various other metrics, e.g., hijacks or transient paths detection (§9).

Six respondents in C_1 acknowledged that using more VPs would improve the quality of their analysis. The last respondent was not sure, given the potential redundancy in the data sources (which he did not analyze). Two of the six believed it would not *significantly* change the conclusion of their measurement studies (e.g., one said that it could help to pinpoint corner cases). However, six of the seven authors in C_1 affirmed that they would have used more VPs if they had more resources and time.

All five respondents in C_2 said that extending the duration of their study would improve the quality of their results. One respondent thought the gain would not be significant; another said it could help detect rare routing events. All respondents in C_2 would have extended the duration of their observation window given more time and resources. We experimentally confirm in §9.2.3 that extending the timeframe of analysis improves the quality of its results with a case study on routing detour characterization [46].

4 Opportunity to Optimize Sampling

We propose a systematic framework to characterize *redundancy* across BGP routes collected by the VPs. We use the term *redundant* to refer to updates with similar (or identical, depending on the *redundancy* definition) attribute values (see definitions below). Thus, two redundant VPs, i.e., that observe redundant routes, likely provide similar views over routing events such as hijacks, traffic engineering, etc.

Methodology. We characterize redundancy between pairs of VPs by computing the proportion of redundant updates that they collect using three different, gradually stricter, definitions of update redundancy. We denote U_i the set of updates observed by VP i . Consider a BGP update $u_{t,p} \in U_i$ with t the time at which the route was observed and p its prefix.

Definition 1 (prefix based) The update $u_{t_1,p_1} \in U_1$ is *redundant* with the update $u_{t_2,p_2} \in U_2$ if:

- $|t_1 - t_2| < 5$ minutes, and $p_1 = p_2$.

We chose 5 minutes because it is an approximation of the BGP convergence time [29]. This first definition might be appropriate to map prefixes with their origin AS.

For our second definition, we denote $A_i(t,p)$ the set of AS links in the AS path of the most recent BGP route observed by VP i for prefix p at time t .

Definition 2 (prefix and as-path based) The update $u_{t_1,p_1} \in U_1$ is *redundant* with the update $u_{t_2,p_2} \in U_2$ if:

- $|t_1 - t_2| < 5$ minutes, and $p_1 = p_2$, and
- $A_1(t_1, p_1) \setminus A_1(t_1 - \epsilon, p_1) \subset A_2(t_2, p_2) \setminus A_2(t_2 - \epsilon, p_2)$.

The second condition checks whether the changes (operator \setminus) in the AS paths observed by VP 1 for a given prefix are included (operator \subset) in the set of changes observed by VP 2 for the same prefix. This second definition might be appropriate to detect new AS links or transient paths.

Our third definition follows the same approach but adds BGP communities. We denote $C_i(t, p)$ the set of community values of the most recent BGP route observed by VP i for prefix p and at time t .

Definition 3 (prefix, as-path, and community-based) The update $u_{t_1, p_1} \in U_1$ is redundant with update $u_{t_2, p_2} \in U_2$ if:

- $|t_1 - t_2| < 5$ minutes, and $p_1 = p_2$, and
- $A_1(t_1, p_1) \setminus A_1(t_1 - \epsilon, p_1) \subset A_2(t_2, p_2) \setminus A_2(t_2 - \epsilon, p_2)$, and
- $C_1(t_1, p_1) \setminus C_1(t_1 - \epsilon, p_1) \subset C_2(t_2, p_2) \setminus C_2(t_2 - \epsilon, p_2)$.

We note that Def. 2 and 3 are asymmetric because, given two set X and Y of objects of same type, $X \subset Y \not\Rightarrow Y \subset X$.

Redundant pairs of VPs exist. Fig. 4 (top row) shows the level of redundancy for the three definitions and between 100 VPs randomly selected and computed over the updates observed during two hours on August 1, 2022. Observe that we performed 30 random selections with different seeds and show the median case (in terms of redundant pairs of VPs). One cell in the matrix indicates the redundancy of the VP on the ordinate with the VP on the abscissa. We define the redundancy between VP 1 and VP 2 as the proportion of updates observed by VP 1 that are redundant with at least one update observed by VP 2. For better visibility, we show the most redundant VPs at the top of the figures.

Redundant pairs of VPs exist regardless of the redundancy definition used. Logically, the stricter the definition, the fewer redundant pairs of VPs. Fig. 4 (left) shows that the VPs can be highly redundant when they are selected randomly. For instance, with the loose Def. 1, we observe that 74 among the 100 randomly selected VPs have $>50\%$ of their updates that are redundant with the ones observed by two other VPs or more (23 for Def. 2 and 16 for Def. 3). We observe a similar redundancy level when considering only full feeders.

5 Main challenge: prevent overfitting

Our design objective is a general framework that can accommodate different definitions of redundancy in selecting the set of least redundant VPs. However, optimizing selection for one objective is likely to overfit, leading to poor performance for other objectives. Thus, while the three definitions in §4 enable illustrating the redundancy across current VPs, none of

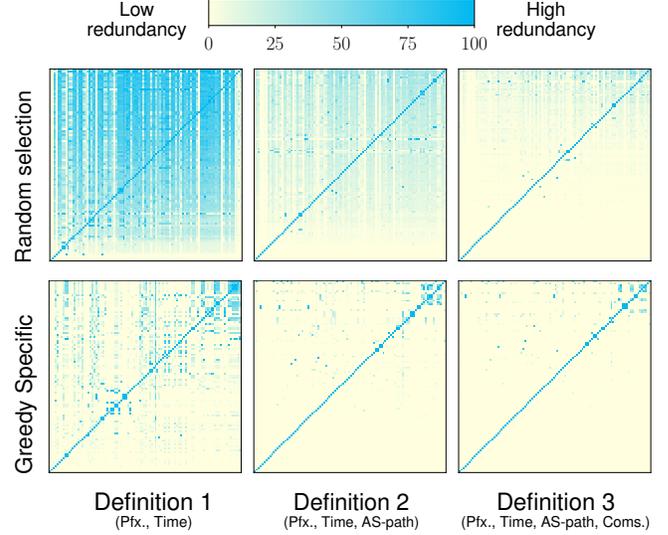


Figure 4: Redundancy among a subset of 100 existing VPs selected using two different techniques for three increasingly stricter redundancy definitions. Randomly selecting VPs (top row) returns significantly more pairs of redundant VPs.

them are used in the design of MVP. These definitions are too naive to accurately quantify redundancies between the VPs.

We explore this risk of overfitting to a particular objective using a VPs selection strategy optimized for one objective: minimizing redundancy. This selection strategy, which we name *greedy specific*, iteratively selects (in a *greedy* fashion) the VP that minimizes the proportion of redundant updates across all the updates collected by the selected VPs. We implement three versions of it, one for each redundancy definition used in §4. Thus, *greedy specific* approximates an optimal VP selection when the goal is to minimize redundancy between VPs according to a specific definition of redundancy.

Greedy specific limits redundancy. We select 100 VPs using *greedy specific*. Logically, the selected VPs are less redundant (see Fig. 4, bottom row) compared to the 100 VPs randomly selected. With the loose Def. 1, only 30 VPs have $>50\%$ of their updates redundant with ones observed by two other VPs or more. This number drops to 9 with Def. 2 and 5 with Def. 3. This result highlights that while VPs can be highly redundant, nonredundant pairs of VPs also exist.

Greedy specific overfits. *Greedy specific* overfits because it optimizes one particular objective. Thus, it works well for this objective but not for the others. We confirm this overfitting effect in §9 where we benchmark *greedy specific* against MVP on various objectives and show that it performs poorly on objectives that it does not optimize. Consequently, one would need to design a *greedy specific* VPs selection for every possible definition of data redundancy—which is unpractical given that there is an infinite number of definitions.

6 Methodology Overview

MVP samples BGP updates from RIS and RV at the VP granularity. Our method has four steps that we overview below.

Step 1 (§7.1): Select a large, unbiased set of BGP events that we use to gauge pairwise redundancy between VPs. MVP evaluates the redundancy between two VPs based on a carefully selected set of *non-global* BGP events (i.e., AS path changes). Global events are typically seen by all VPs and have the same impact on every VP view, rendering them less discriminating for this purpose. We stratify our selection of sampled events across space and time to avoid bias.

Step 2 (§7.2): Characterize how VPs experience the selected events. For every BGP event, MVP quantifies topological features [48] of the ASes involved as observed by each VP. These features embed information about the four attributes of a BGP update: time, prefix, AS path, and communities.

Step 3 (§7.3): Compute pairwise redundancy between VPs. MVP computes the pairwise Euclidean distance in a n -dimensional space, where n is the number of topological features times the number of events. VP pairs with similar feature values for many events are close in this space and thus likely redundant. MVP then computes the average Euclidean distances between each pair of VPs computed over different and nonoverlapping time periods.

Step 4 (§7.4): Sort and select the least redundant VPs. MVP relies on a greedy algorithm that considers both data *redundancy* and its *volume* to build a set of the most valuable VPs. MVP first adds the VP with the lowest average Euclidean distance to all other VPs, and then greedily adds the VP that balances minimal redundancy with already selected VPs and minimal additional data volume that the VP brings.

7 Methodology Details

In the following, we consider the set of VPs V that includes all VPs from RIS and RV. We compute the RIB of VP v at time t using its last RIB dump before t and subsequent updates until t . We use this RIB to construct and maintain the undirected weighted graph $G_v(t) = (N_v(t), E_v(t))$ from the AS paths of the best routes observed by v at time t , with $N_v(t)$ the set of nodes and $E_v(t) \in N_v(t) * N_v(t)$ the set of AS links. The edges are undirected because two identical paths in opposite directions should not appear as nonredundant. Each edge in $E_v(t)$ has a weight in \mathbb{Z}^+ which is the number of routes in the RIB that includes this edge in their AS path.

7.1 Select BGP events to assess redundancy

MVP uses local and partially visible new-AS-link events. MVP focuses on BGP events that trigger a new AS link to appear in the path to reach prefix p from different VPs. A new-AS-link event is a candidate event in C if at least two

ID	Name	# of ASes	Avg.degree	Description
1	Stub	63310	3	ASes without customer
2	Transit-1	10845	27	Transit ASes with a customer cone size lower than the average
3	Transit-2	704	267	Transit ASes \notin Transit-1
4	HyperGiant	15	1078	Top 15 as defined in [8]
5	Tier1	19	1817	Tier1 in the CAIDA dataset [16]

Table 1: MVP balances selected events across 5 AS types.

and fewer than half of the VPs begin to use the same new AS link to reach the same prefix within a 10-minute window (to accommodate typical BGP convergence and path exploration delays [29,35]). Since the aim of MVP is to find data unique to individual VPs, we exclude global events (i.e., seen by most VPs) to focus on local events.

MVP avoids biases across time and location. From candidate set C , MVP builds the final set of events \mathcal{E} by selecting 15 events in 500 different and nonoverlapping 10-minute time periods. Adding more periods does not affect significantly the results. MVP samples time periods randomly within a one-month timeframe to avoid mis-inferring one larger event (e.g., a route leak that continuously generates new links for multiple hours) as several smaller AS-link-level events. Inspired by previous approaches to mitigate the risk of over-sampling core or stub (edge) ASes [37,45], our approach classifies ASes into five categories (Table 1) and selects an equal number of new-AS-link events for every pair of AS categories. We distinguish two classes of transit providers by customer cone size (Transit-1 and -2) since they have different topological properties. If an AS belongs to more than one category, we classify it in the category with the highest ID. ASes classified in a lower row of Table 1 have a higher degree, and there are more low-degree ASes than high-degree ASes.

Fig. 5 shows the proportion of selected events for each of the 15 pairs of AS category (the matrixes are symmetric) and for 7500 events selected in January 2023 using two schemes: balanced and random. The random selection (Fig. 5b) selects many more events involving Transit-2 ASes (69%) than hypergiants (11%), while our balanced selection scheme mitigates biases by selecting the same number of links in every category (Fig. 5a). For each time period, MVP selects one event in each of the 15 pairs of AS, yielding $15 * 500 = 7500$ events ($|\mathcal{E}| = 7500$) for use in the next step.

7.2 Quantifying the observation of the VPs

MVP considers the four main BGP attributes. MVP computes topological features on the graphs $G_v(t)$ for all VPs. The combination of these topological features prevents overfitting as the graphs on which they are computed embed information about the four main BGP attributes (§2). More concretely, the graphs $G_v(t)$ embed information about (i) the

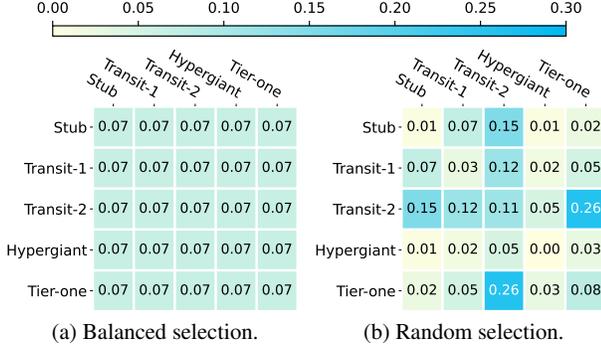


Figure 5: MVP selects the new-AS-link events using a balanced selection scheme that reduces bias (Fig. 5a vs. Fig. 5b). The x- and y-axis are the five categories of ASes (see Table 1).

time as the graph is updated over time, (ii) the AS path as it is used to build the AS graph, (iii) the prefixes as they are used to weight every edge on the graph, and (iv) the community values as they are strongly correlated with the AS path. We confirm this correlation by downloading the first RIBs of Jan. 2023 for all VPs and analyzing the correlation between the AS path and the set of BGP communities. We find that two identical AS paths share the exact same set of BGP communities in 93% of the cases. We thus do not embed more information about BGP communities because many of them encode local traffic engineering decisions [17] that could lead to MVP overfitting. We validate this design choice in §9.1.

Type	Categorie	Name	Weighted	Index
Node-based	Centrality Metrics	Closeness centrality	✓	0
		Harmonic centrality	✓	1
	Neighborhood Richness	Average neighbor degree	✓	2
		Eccentricity	✓	3
	Topological Pattern	Number of Triangles	✗	4
	Clustering	✓	5	
Pair-based	Closeness Metrics	Jaccard	✗	6
		Adamic Adar	✗	7
		Preferential attachment	✗	8

Table 2: Node-based and pair-based features used by MVP.

MVP uses 15 diverse topological features (Table 2). MVP computes topological features (extracted from literature [20]) that are either node-based or link-based. Node-based features are computed for the two ends of a new AS link, while link-based are computed for the new AS link. MVP uses six node-based features that we classify into three categories. The first one quantifies how central and connected a node is in the graph; the second quantifies how connected are the neighboring nodes; and the third quantifies the topological patterns (e.g., triangles) that include the node. We classify the three pair-based features into a single category that measures how close two nodes are based on their neighboring nodes. Five

features rely on edge weights. We omit other topological features as they are redundant with the selected ones.

MVP computes the value of the features for each VP and selected event. Consider the event $e \in \mathcal{E}$ that is the appearance of the AS link (e_{AS1}, e_{AS2}) at time e_t , and the VP $v \in V$. Computation of the feature values depends on the feature type. We denote F_n (resp. F_p) the set of node-based (resp. pair-based) features and show how MVP computes the value of these two types of features for event e and VP v .

Node-based features: Consider feature $f_i \in F_n$ and $f_i(x, G_v(t))$ its value for node x on the graph $G_v(t)$, with i the feature index in Table 2. MVP computes the following 12-dimensional feature vector.

$$T_{node_based}(v, e) = [f_0(e_{AS1}, G_v(e_t)), f_0(e_{AS2}, G_v(e_t)), \dots, f_5(e_{AS1}, G_v(e_t)), f_5(e_{AS2}, G_v(e_t))]$$

Pair-based features: Consider feature $f_i \in F_p$ and $f_i(x_1, x_2, G_v(t))$ its value for the node pair (x_1, x_2) on the graph $G_v(t)$, with i the feature index in Table 2. MVP computes the following 3-dimensional feature vector.

$$T_{pair_based}(v, e) = [f_6(e_{AS1}, e_{AS2}, G_v(e_t)), \dots, f_8(e_{AS1}, e_{AS2}, G_v(e_t))]$$

The final feature vector used by MVP is $T(v, e)$, an 15-dimensional vector that is the concatenation (denoted \oplus) of the node- and pair-based features.

$$T(v, e) = T_{node_based}(v, e) \oplus T_{pair_based}(v, e)$$

7.3 Redundancy scoring

MVP computes pairwise redundancy between VPs in the following four steps.

Step 1: Concatenate the feature vectors. MVP first concatenates the computed topological feature vectors (15 features) for all the events selected in the same time period (15 events). We denote \mathcal{E}_p the events selected in the p -th time period ($|\mathcal{E}_p| = 15$), with $0 \leq p < 500$, and denote $e_{p,i} \in \mathcal{E}_p$ the i -th selected event in the p -th time period. $F(v, p)$ is the concatenated feature vector for VP v and the events \mathcal{E}_p , which has $15 * 15 = 225$ dimensions and which MVP calculates as:

$$F(v, p) = T(v, e_{p,0}) \oplus T(v, e_{p,1}) \oplus \dots \oplus T(v, e_{p,14})$$

Step 2: Normalize concatenated feature vectors. MVP normalizes the data for each time period using the feature matrix $\mathcal{M}(p)$ that includes the concatenated feature vectors for all VPs (rows) and events (columns) in period p .

$$\mathcal{M}(p) = \begin{bmatrix} F(v_0, p) \\ \dots \\ F(v_{|V|}, p) \end{bmatrix}$$

MVP normalizes (operation ∇) the matrix $\mathcal{M}(p)$ column-wise using a standard scaler that transforms every column such that its average is zero and its standard deviation is one.

Step 3: Compute Euclidean distance between VPs. MVP uses the normalized matrix $\nabla(\mathcal{M}(p))$ to compute the Euclidean distance between every pair of VPs and for all events in the time period p (operation \diamond). We denote $\nabla(\mathcal{M}(p))_x$ the x -th row in the matrix $\nabla(\mathcal{M}(p))$ and $\nabla(\mathcal{M}(p))_{x,i}$ its value at index i (i.e., the i -th column). We define the Euclidean distance between the n -th VP v_n and the m -th VP v_m over the selected events in the time period p as follows.

$$\diamond(v_n, v_m, p) = \sum_{i=0}^{225} (\nabla(\mathcal{M}(p))_{n,i} - \nabla(\mathcal{M}(p))_{m,i})^2$$

Step 4: Compute the average distance over all time periods.

The redundancy score $\mathcal{R}(v_n, v_m)$ between two VPs v_n and v_m relates to the normalized average Euclidean distance between them over the 500 time periods, computed as:

$$\mathcal{R}(v_n, v_m) = 1 - \prod_{p=0}^{500} \left(\left(\sum_{p=0}^{500} \diamond(v_n, v_m, p) \right) * \frac{1}{500} \right)$$

The operator \prod applies a min-max scaler so that scores are between 0 and 1, with 1 meaning the most redundant pair of VPs and 0 the less redundant pair of VPs. MVP thus computes and returns a redundancy score for every pair of VPs.

7.4 Generating a set of VPs

We now explain how MVP generates a set of VPs \mathcal{O} that minimizes the proportion of redundant information collected. MVP initializes the set \mathcal{O} with the most redundant VP, i.e., the one with the lowest sum of Euclidean distances to all the other VPs. This design choice allows the redundant part of the BGP data (e.g., c2p links) to be visible by the first selected VP. At every following iteration, MVP builds a candidate set of VPs \mathcal{K} that contains the unselected VPs exhibiting the lowest maximum redundancy score. The maximum redundancy score P measures the maximum redundancy between a VP v and the set of VPs \mathcal{O} and is defined as follows.

$$P(\mathcal{O}, v) = \max(\mathcal{R}(v, v_i), \forall v_i \in \mathcal{O})$$

MVP adds in \mathcal{K} the $\alpha = 25\%$ of the nonselected VPs that exhibit the lowest maximum redundancy score.

MVP then adds in set \mathcal{O} the VP that is in the candidate set \mathcal{K} and that collects the lowest volume of data compared to the other VPs in \mathcal{K} . MVP estimates the volume of data collected by the VPs by counting the number of updates that they received over 365 one-hour periods, one randomly selected in each day of the year to align with the yearly update rate of MVP (§8). The α parameter allows tuning redundancy and volume knobs: a low α prioritizes low redundancy while a

higher α prioritizes low resulting data volume. We found that $\alpha = 25\%$ performs well in practical scenarios (we tested a range from 10% to 50%).

8 System functionalities

MVP runs on a commodity server. Upon launch, it collects BGP routes from RIS and RV using BGPStream [36] and computes the redundancy between every pair of VPs at a yearly granularity, which is enough given that redundancies between VPs remain stable over time (see §9.3). MVP then takes as input a *year* and a *volume of data* and returns a set of VPs that generates a volume of data lower than the volume specified as input. MVP returns the redundancy scores calculated for every pair of VPs. Thus, users have the option to compute their own set of complementary VPs based on these redundancy scores and some additional constraints that they might have. This is useful when users want to include (or exclude) some VPs (regardless of how redundant they are), which will result in another set of VPs rather than the default set provided by MVP. For instance, when trying to detect new peering, a user may want to take some VPs at an IXP in addition to some VPs selected by MVP.

MVP runs at <https://bgproutes.io>, allowing users to get a list of VPs or the redundancy scores without computational expenses. We implemented three versions of MVP, one for IPv4 routes (MVP^{v4}), one for IPv6 routes (MVP_{v6}) and one that considers both IPv4 and IPv6 routes (MVP_{v6}^{v4}). The three versions use the same methodology (described in §7) to compute redundancy and generate a set of VPs.

9 Evaluation

We show that MVP improves the trade-off between the volume of data collected and the routing information inferred compared to current VPs selection strategies in five use cases for which we have ground truth (§9.1). We then show that MVP would improve coverage and accuracy of previous studies for which ground truth is unknown (§9.2). Finally, we show that the key design choices of MVP are sound (§9.3).

9.1 Benchmarking MVP

We benchmark MVP against three baselines per use case.

Use cases. We evaluate MVP on five different use cases that we carefully picked such that each BGP attribute is useful for at least one of them. For instance, the *time* is useful to detect transient events (use case *I*); the *prefix* is useful to detect Multiple Origin ASes (MOAS) prefixes (use case *II*); the *AS path* is useful to map the Internet topology (use case *III*); and the *community values* are useful to detect traffic engineering (use case *IV*) and unnecessary updates (use case *V*). Our goal is to demonstrate that MVP does not overfit on some particular

Use case	Objective	Naives baselines			Greedy specifics use cases (§9.1)					Greedy specifics Def. (§4)		
		Random	AS-distance	unbiased [45]	<i>I</i>	<i>II</i>	<i>III</i>	<i>IV</i>	<i>V</i>	Def. 1	Def. 2	Def. 3
Transient path detection (<i>I</i>)	50 %	1.55	1.76	1.82	0.70	2.99	3.29	3.82	2.89	1.96	2.12	1.69
	70 %	1.38	1.62	1.53	0.76	3.24	3.51	3.42	3.09	1.56	1.56	1.78
	90 %	1.13	1.17	1.21	0.75	1.66	1.67	1.66	1.66	1.33	1.15	1.59
MOAS detection (<i>II</i>)	50 %	2.35	3.38	3.41	2.31	0.98	1.80	2.83	1.53	3.39	2.85	3.98
	70 %	2.18	3.44	3.38	2.56	0.85	1.79	2.30	1.83	3.02	2.66	3.67
	90 %	1.98	2.69	3.06	2.37	1.04	2.31	2.82	2.56	2.46	2.19	3.31
AS topology mapping (<i>III</i>)	50 %	2.59	2.97	2.43	1.58	1.29	0.71	1.53	1.94	2.27	2.18	3.35
	70 %	2.06	2.29	2.13	1.33	1.22	0.64	1.29	1.37	2.14	1.64	2.28
	90 %	1.72	1.88	1.80	1.30	1.18	0.77	1.23	1.27	1.73	1.64	1.80
Traffic engineering detection (<i>IV</i>)	50 %	4.59	4.74	4.82	3.76	2.67	2.34	0.47	3.33	3.95	3.34	4.76
	70 %	2.71	3.37	3.52	3.04	1.86	2.89	0.41	1.85	4.34	2.02	3.51
	90 %	1.55	1.70	1.95	1.61	1.54	1.52	0.32	1.46	1.88	1.33	1.89
Unnecessary updates detection (<i>V</i>)	50 %	1.72	2.89	2.41	2.19	2.10	2.63	2.20	0.38	2.43	2.92	2.59
	70 %	1.30	2.04	1.91	1.43	1.53	1.50	1.90	0.39	1.51	1.63	2.02
	90 %	1.01	1.36	1.39	1.17	1.18	1.14	1.16	0.50	1.09	1.38	1.35

Table 3: Data reduction factors with MVP^{v4} compared to several baselines for five use cases. MVP outperforms every baseline for all five use cases. Unlike *greedy specifics*, MVP greatly avoids overfitting.

use cases or BGP attributes. For each use case, we process the updates collected during 100 one-hour periods (randomly selected in May 2023) and benchmark MVP on a set of events found. We thus have ground truth. We briefly describe below each use case along with our experimental settings.

I Transient paths detection. Transient paths are BGP routes visible for less than five minutes, a typical BGP convergence delay [29], and which can be attributed to e.g., path exploration [35]. We focus on 200 randomly selected transient path events for every one-hour period, making a total of $100 * 200 = 20000$ events used.

II MOAS prefixes detection. MOAS prefixes are announced by multiple distinct ASes [44], which can be caused by legitimate [53] or malicious [13, 40, 49] actions. We focus on 200 MOAS randomly selected events for every one-hour period, making a total of $100 * 200 = 20000$ MOAS events used.

III AS topology mapping. This is useful for e.g., inferring BGP policies [31] or AS paths [32]. For each VP, we process the first RIB dump of May 2023 as well as the updates collected during the 100 one-hour periods and focus on all distinct AS links observed.

IV Traffic engineering detection. We focus on action communities i.e., those associated with traffic engineering actions [50]. For every one-hour time period, we focus on 80 updates for which a path change coincides with the appearance of an action community, making a total of $100 * 80 = 8000$ path changes used.

V Unnecessary Updates detection. An unnecessary update is a BGP update that only signals a change in the community

values but not in the AS path [28]. We consider 200 unnecessary updates randomly picked within each one-hour period, making a total of $100 * 200 = 20000$ events used.

Baselines. We benchmarked MVP against three naive baselines commonly used in practice (§3.2): (*i*) *random* selection of VPs, which results in a skewed set of VPs as they exhibit biases [45]; (*ii*) *AS-distance*, i.e., select the first VP randomly and the following ones to maximize the AS-level distance between selected VPs; and (*iii*) *unbiased*, i.e., start with all VPs and iteratively remove the one that most increases the bias on the set of remaining VPs. We measure the bias using the definition in [45].

We compare MVP against the three *greedy specific* VPs selection strategies optimized for Def. 1, 2, and 3 (§4). Additionally, we compare MVP against five other *greedy specifics*, one optimized for each of the five use cases described above. Unlike the *greedy specifics* described in §5, these five *greedy specifics* optimize the trade-off between the volume of the data and its capacity to achieve a particular objective. For instance, when the objective is to map the AS topology (use case *III*), *greedy specific* iteratively selects the VP that best improves the trade-off between the number of discovered AS links and the volume of processed data.

Reduction factor definition. We define the *reduction factor* to capture how much MVP reduces the number of BGP updates required to fulfill a particular objective. More precisely, assume an objective O and a baseline B . We iteratively build a set of VPs using baseline B . At every iteration, we download all the updates that the newly selected VP observes

during 100 one-hour periods randomly selected in May 2023. We stop iterating when all updates collected by the selected VPs enable the data to meet O . Similarly, we build another set of VPs using MVP and stop selecting new VPs (see §7.4) when the selected ones meet O . The *reduction factor* is the ratio between the number of updates processed with B and with MVP. More formally, the *reduction factor* is $\frac{|U_B^O|}{|U_{MVP}^O|}$ with $|U_B^O|$ and $|U_{MVP}^O|$ the number of updates processed to fulfill objective O with baseline B and MVP respectively. A *reduction factor* = 2 means that we can fulfill objective O with half as many updates when using MVP compared to when using baseline B . More generally, a *reduction factor* > 1 means that we can fulfill the same objective with less data when using MVP compared to when using B .

Benchmark results. Table 3 summarizes our results. For each use case, we focus on three objectives: mapping $X\%$ of the AS topology (use case III) or detecting $X\%$ of the events (use case I, II, IV, and V), with X equal to 50, 70, or 90. Here, we focus on the performance of MVP^{v4}. MVP_{v6} and MVP^{v4}_{v6} yield comparable performance (see §B).

Takeaway #1: MVP outperforms every naive baseline for every use case, i.e., the *reduction factor* is always above one. For instance, we detect 90% of the MOAS events with 3.06× less data (the *reduction factor* is 3.06) when using MVP compared to selecting the VPs using the *unbiased* baseline. This means that MVP only needs 32% of the updates required by the *unbiased* baseline to fulfill the objective. Comparably to what we observe in our mini-Internet simulations (§3), the *random* baseline performs better than *AS-distance*.

Takeaway #2: We can see that MVP generalizes whereas greedy specific overfits. In fact, for a particular use case, MVP is less performant than the *greedy specific* strategy optimized for this use case. For any other use case, MVP performs better than the *greedy specific*s not optimized for that use case. These results demonstrate that the *greedy specific* strategies overfit. They are also not practical as they need ground truth.

9.2 Impact on previous works

We show that MVP would improve the outcome of three measurement studies and tools that are fueled by the BGP data from RIS and RV (and for which there is no ground truth).

9.2.1 Inference of AS properties

We show that MVP improves AS relationship inferences (a popular research problem [18, 22, 27, 31]) and AS ranking [9]. **MVP helps to infer +15% more AS relationships.** We replicate the methodology proposed in [31] that relies on public BGP data from RIS and RV to infer AS relationships and build the widely-used CAIDA AS-relationship dataset [16]. We compute the number of inferred AS relationships for every month in 2023 when using the 648 VPs that CAIDA uses

to build its dataset (In January 2023) and when using VPs selected by MVP. We ensure that the VPs selected with MVP generate the same volume of data as the 648 used by CAIDA so that any performance gap can confidently be attributed to MVP. We find that the VPs selected by MVP enable consistent (from Jan. 2023 to Aug. 2023) inference of ≈90k additional AS relationships (≈+17%) while missing only ≈11k AS relationships (≈2.2%) present in the original dataset. Thus, the tradeoff is largely in favor of using MVP (≈+15% overall).

We also replicated the AS relationship validation algorithm used in [31] (which relies on the IRR and RIR data) and found that the true positive rate (the metric used in [31]) remains identical (97%). Thus, MVP significantly improves coverage without processing more data or losing accuracy.

MVP prevents flawed inferences in the ASRank dataset. We replicate the methodology used by ASRank [9] to compute the AS Customer Cone Sizes (CCS). We find that the CCS changes for 1067 ASes when using MVP and manually investigated two cases of substantial changes:

Case I⁴: AS132337 has a CCS of 1 in the original dataset and a CSS of 18k when using MVP, making it the 15th AS highest ranked by CCS. We contacted AS132337 who confirmed that it has 18k customers. MVP correctly ranks AS132337 because it selects the unique VP that sees it as a transit AS.

Case II⁵: AS24745 is the route server of Balcan-IX and has a CSS of 16 in the original ASRank dataset. However, we manually checked its participants and found that the 16 customers are misclassified and actually peer through AS24745. With MVP, the CSS of AS24745 is 1 and these errors are avoided.

In both cases, MVP enables more accurate inferences of CCSs because it collects more diverse AS paths. Thus, we can confidently say that MVP would prevent many flawed inferences likely present in the dataset provided by ASRank.

9.2.2 Detection of forged-origin hijacks

We show that MVP improves forged-origin hijack detection, which is the goal of many systems that use BGP routes from RIS and RV [1, 11, 15, 26, 44]. Forged-origin hijacks are a type of BGP hijack where the attacker prepends the valid origin to the AS path to make the hijacked route appear legitimate.

MVP improves the accuracy of forged-origin hijack inferences. We replicate the algorithm of DFOH [26] that uses routes collected by 287 RIS and RV VPs to infer forged-origin hijacks. We implement two versions of DFOH, one called DFOH_{MVP} which uses a set of VPs selected with MVP, and another one called DFOH_R that uses a random set of VPs. In both versions, we ensure that the volume of data collected is identical to the one used in [26]. As DFOH relies on probabilistic inference, we measure the performance of DFOH_{MVP} and DFOH_R in terms of True Positive Rate (TPR) and False Positive Rate (FPR). We obtain an approximation of ground

⁴<https://asrank.caida.org/asns?asn=132337&type=search>

⁵<https://asrank.caida.org/asns?asn=24745&type=search>

Experiment	Duration	# of VPs	# of processed Updates	# of Detours
Original paper	1 Month	All VPs	≈61B	174k
Random selection	2 Months	624 (median)	≈61B	165k (median)
	4 Months	313 (median)	≈61B	171k (median)
MVP selection	2 Months	413	≈61B	250K
	4 Months	220	≈61B	263k

Table 4: Using fewer VPs selected by MVP enables a longer study that detects more detours with the same volume of data.

truth (needed to compute the TPR and FPR) by implementing a third version of DFOH, called $DFOH_{ALL}$ that uses all VPs from RIS and RV. Observe that $DFOH_{ALL}$ is an approximation of ground truth because incorrect inferences are still possible even if all VPs are used. We restrict our analysis to one month (Jan. 2022) because $DFOH_{ALL}$ is resource-hungry as it uses all VPs. We find that $DFOH_{MVP}$ uncovers 947 suspicious cases against only 700 for $DFOH_R$. $DFOH_{MVP}$ outperforms $DFOH_R$ for both the TPR and the FPR: It has a TPR of 85.7% (against 61.1% for $DFOH_R$) and a FPR of 14.4% (against 60.1% for $DFOH_R$)—a $\approx 4\times$ better precision.

$DFOH_R$ misses suspicious cases that $DFOH_{MVP}$ does not. We manually investigated, using public peering databases (e.g., PeeringDB) some of the suspicious cases inferred by $DFOH_{MVP}$ and not by $DFOH_R$. We find cases that appear particularly suspicious (thus useful for operators) and describe two of them below (also found by the original DFOH).

Case I⁶: On Jan. 1, 2022, AS267548, a small Peruvian AS, appears between Sprint, a Tier1 AS, and AS199524, a large content provider. However, AS267548 is not supposed to provide transit between these two ASes.

Case II⁷: On Jan. 6, 2022, AS9269, an ISP based in Hong Kong appears directly connected with AS268568, a Brazilian ISP. These two ASes do not share any IXP and are not supposed to peer directly.

These two cases show that MVP enables the detection of additional potential routing attacks versus not using it.

9.2.3 Characterizing international routing detours

We focus on a study that uses all VPs to characterize international routing detours over one month [46]. International detours occur when two ASes in the same country are reachable through an AS in another country, which can lead to extra forwarding delays. We show that by using fewer VPs selected by MVP, we can lengthen the duration of the study to find more detours without processing more data.

⁶http://dfoh.uclouvain.be/cases/2022-01-01_1239_267548

⁷http://dfoh.uclouvain.be/cases/2022-01-06_9269_268568

MVP helps to detect +44% more routing detours. We replicate the methodology used in [46] to detect routing detours except that (i) we use a set of VPs selected using MVP that generates $\alpha\times$ less data compared to using them all, with $\alpha = 2$ and $\alpha = 4$, and (ii) we run the analysis over two months when $\alpha = 2$ and four months when $\alpha = 4$. Thus, the overall volume of data collected remains similar ($\approx 61B$ RIB entries), regardless of α . Table 4 shows the number of routing detours detected in May 2023 (and until June and August 2023 when $\alpha = 2$ and 4, respectively). We detect 250k detours over two months ($\alpha = 2$) when using 413 VPs selected by MVP—a +44% increase compared to using all VPs during one month as in [46]. When $\alpha = 4$, we use 220 VPs selected by MVP on four months and find 263k detours—better than using them all on one month.

We explored the trade-off between the number of VPs and the duration of the study using a *random* VPs selection strategy. We detected 165k detours when using ≈ 624 random VPs and running the analysis over two months (we tested the *random* selection with 50 seeds and report the median in Table 4). This is fewer than when we replicated the original experiment, which demonstrates that optimized VP selection enables discovering more routing detours.

MVP enables improved characterization of routing detours.

We replicate the methodology used in [46] to rank countries based on their number of detours, and ASes based on how often they originate a detoured path. We find differences when using MVP, including two interesting cases:

Case I: Using MVP (with $\alpha = 2$), we discover 33k (+68%) additional detours traversing the US and 22k (+37%) traversing Russia compared to when using the settings in [46]. These additional detours rank the US as the #1 country with the highest number of routing detours and Russia as #2, whereas with the settings in [46] Russia is ranked #1 and the US #2.

Case II: Using MVP (with $\alpha = 2$) enables detecting 720 (+83%) additional routing detours involving AS262503 compared to when using the settings of [46]. This changes rankings: AS262503 became #1 vs. #7 with the settings in [46].

As our rankings are based on the highest number of routing detours compared to [46], we can confidently say MVP improves the characterization of international routing detours.

9.3 Soundness of design choices

We show that our three key design choices – yearly update frequency of redundancy scores, balanced sampling, and topological feature selection – are sound.

MVP’s redundancy scores are sufficiently stable over time that annual recomputation is sufficient

We ran MVP every six months, starting in January 2023 and then going backward until January 2018 (i.e., a total of ten independent runs). We limit the scope of this experiment to 100 randomly selected VPs to limit the computational resources required. Logically,

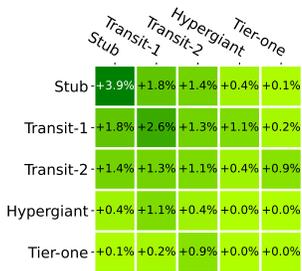


Figure 6: MVP enables mapping more links than rMVP.

we find that the redundancy score differences increase as the time interval between two runs of MVP increases. However, these differences are low. The median difference between the scores of two runs of MVP separated by one year is only 0.021 (which corresponds to a difference of 9%), and it increases to 0.171 (i.e., a difference of 23%) when the two runs are separated by four years. We thus configure MVP to recompute redundancy scores and update its set of selected VPs on a yearly basis (see §8)—a good trade-off between computational cost and performance.

The balanced sampling avoids biases in the collected data.

We implement rMVP, a modified version of MVP where new-AS-links events used to compute redundancy scores across VPs are sampled randomly, i.e., using the distribution depicted in Fig. 5a (instead of using the balanced sampling in §7.1). We compare the performance of MVP and rMVP on AS topology mapping. Note that we observe similar results for other use cases. We map the AS topology for May 2023 (following the methodology in §9.2.1) using both MVP and rMVP and with the same volume of data in either case. Fig. 6 depicts the proportion of additional AS links that we can map when using MVP compared to when using rMVP for every new-AS-link category. MVP always yields better or identical performance than rMVP. The highest difference is when mapping stub-to-stub links (+3.9%) or Transit1-to-Transit1 links (+2.6%). These two link categories are underrepresented when using a random sampling (see Fig. 5a), demonstrating that our balanced sampling scheme mitigates biases.

Every feature category is useful. We implement MVP $\setminus\{f_i, \dots, f_j\}$, a modified version of MVP where we omit features $\{f_i, \dots, f_j\}$ when computing redundancy scores, with $i \dots j$ the feature indexes in Table 2. We use four different versions of MVP $\setminus\{\dots\}$, each omitting a different feature category. We show the *reduction factor* of MVP over each MVP $\setminus\{\dots\}$ for use cases I, II, III, IV, and V in §9.1, with the objective of detecting 70% of the events or mapping 70% of the AS-level topology. Regardless of which feature category is omitted, MVP performs better (i.e., the *reduction factor* is above 1). We conclude that every feature category is valuable.

	$\setminus\{0,1\}$	$\setminus\{2,3\}$	$\setminus\{4,5\}$	$\setminus\{6,7,8\}$
I	1.04	1.05	1.07	1.06
II	1.17	1.02	1.07	1.34
III	1.09	1.11	1.09	1.12
IV	1.32	1.25	1.15	1.16
V	1.61	1.59	1.71	1.62

Table 5: Omitting one feature category reduces the performance of MVP for every use case.

10 Related work

Redundancy and bias between the VPs. Chen et al. showed that VPs observe identical (redundant) AS links and that it is possible to reduce the number of VPs while providing similar measurement power [10]. However, they only focus on one objective (observing AS links) whereas MVP works for *any* objective. Previous works reported that the VPs are biased (in terms of location, network size, etc.) [14, 45, 47]. MVP is data-driven and does not consider these biases as we show that an *unbiased* selection strategy performs poorly (§9.1).

Strategies to select VPs. Prior works demonstrated that carefully selecting VPs increases the utility of the data [52], and proposed a greedy selection strategy that performs better than other naive approaches [34, 52]. However, their selection strategy optimizes one objective (discovering AS links) and thus lacks generality (§9.1). Recent works also study the impact of the VP selection on the discovered IP space and AS links [30].

Placement of the VPs. Gregori et al. proposed a methodology that finds a relevant placement for a new VP [23]. Roughan et al. estimated that 700 strategically positioned VPs were enough to monitor the Internet topology [43]. Finally, Cittadini et al. demonstrated the marginal utility of adding new VPs at the core of the Internet [14].

Strategies to select active measurement probes. Active measurement platforms (e.g., RIPE Atlas) also generate a large volume of data and several data-driven approaches for probe selection exist [4, 6, 25]. Unlike MVP, these approaches optimize the probe selection for specific use cases.

Uses of topological features. Previous works computed topological features on the AS topology to detect routing anomalies [19, 24, 26].

11 Conclusion

We uncovered redundancy in the BGP routes exported by the RIS and RV VPs and identified this redundancy as an opportunity to optimize the use of these data collection systems. We presented MVP, a system that samples BGP data at the VP granularity, enabling users to improve the coverage and accuracy of their studies without processing more data.

The principles that MVP embodies can also lead to a better understanding of the structure of the global Internet as well as how to optimize the measurement and analysis of its routing system. For instance, our redundancy scores could lead to more strategic approaches to gathering and retaining BGP data, e.g., RIS and RV could deprioritize VPs which are overwhelmingly redundant with many others, on a more scientific basis. Finally, our approach can be adapted to active measurement platforms (e.g., Atlas [39]) to reach the same objective of extensive coverage with reduced redundant data.

12 Acknowledgements

This work was supported by the ArtIC project (grant ANR-20-THIA-0006-01), Région Grand Est, Inria Nancy-Grand Est, IHU of Strasbourg, University of Strasbourg, University of Haute-Alsace, the RIPE NCC Community Projects Fund, NSF CNS-2120399 and NSF OAC-2131987. Views are those of the authors and do not represent the endorsements of the funding agencies.

References

- [1] Global Routing Intelligence Platform. <https://grip.inetintel.cc.gatech.edu/>.
- [2] Emile Aben. Route Collection at the RIPE NCC - Where are we and where should we go? 2020. <https://labs.ripe.net/author/emileaben/>.
- [3] Rodrigo Aldecoa, Chiara Orsini, and Dmitri Krioukov. Hyperbolic graph generator. In *Computer Physics Communications*, 2015.
- [4] Malte Apple, Emile Aben, and Romain Fontugne. Metis: Better Atlas Vantage Point Selection for Everyone. In *TMA '22*, 2022.
- [5] Lorenzo Ariemma, Mariano Scazzariello, and Tommaso Caiazzì. MRT#: a Fast Multi-Threaded MRT Parser. In *IFIP/IEEE IM '21*, 2021.
- [6] Vaibhav Bajpai, Steffie Jacob Eravuchira, Jürgen Schönwälder, Robert Kistelegi, and Emile Aben. Vantage point selection for IPv6 measurements: Benefits and limitations of RIPE Atlas tags. In *2017 IFIP/IEEE IM*, 2017.
- [7] BGPKIT. BGPKIT. 2022. <https://blog.bgpk.it/>.
- [8] Timm Böttger, Félix Cuadrado, and Steve Uhlig. Looking for hypergiants in PeeringDB. In *SIGCOMM CCR*, 2018.
- [9] CAIDA. As rank. 2023. <https://asrank.caida.org/>.
- [10] Kai Chen, Chengchen Hu, Wenwen Zhang, Yan Chen, and Bin Liu. On the eyeshots of BGP Vantage Points. In *GLOBECOM '09*, 2009.
- [11] Shinyoung Cho, Romain Fontugne, Kenjiro Cho, Alberto Dainotti, and Phillipa Gill. BGP hijacking classification. In *TMA'19*, 2019.
- [12] CIDR. CIDR REPORT. 2023. <https://www.cidr-report.org/as2.0/>.
- [13] CitizenLab. A case study of the China Telecom incident. 2012. <https://citizenlab.ca/2012/12/>.
- [14] Luca Cittadini, Stefano Vissicchio, and Benoit Donnet. On the quality of BGP route collectors for iBGP policy inference. In *IFIP '14*, 2014.
- [15] Avichai Cohen, Yossi Gilad, Amir Herzberg, and Michael Schapira. Jumpstarting BGP security with Path-End Validation. In *SIGCOMM'16*, 2016.
- [16] University San Diego. The CAIDA AS Relationships Dataset, 2022. 2022. <https://www.caida.org/catalog/datasets/as-relationships/>.
- [17] Benoit Donnet and Olivier Bonaventure. On BGP communities. In *SIGCOMM CCR*, 2008.
- [18] Guoyao Feng, Srinivasan Seshan, and Peter Steenkiste. UNARI: An Uncertainty-Aware Approach to AS Relationships Inference. In *CoNEXT '19*, 2019.
- [19] Romain Fontugne, Anant Shah, and Emile Aben. AS Hegemony: A Robust Metric for AS Centrality. In *SIGCOMM'17*, 2017.
- [20] Linton C. Freeman. Centrality in social networks conceptual clarification. *Social Networks*, 1978.
- [21] Lixin Gao and Jennifer Rexford. Stable Internet Routing without Global Coordination. In *SIGMETRICS '00*, 2000.
- [22] Vasileios Giotsas, Matthew Luckie, Bradley Huffaker, and kc claffy. Inferring Complex AS Relationships. In *IMC '14*, 2014.
- [23] Enrico Gregori, Alessandro Improta, Luciano Lenzini, Lorenzo Rossi, and Luca Sani. On the Incompleteness of the AS-Level Graph: A Novel Methodology for BGP Route Collector Placement. In *IMC '12*, 2012.
- [24] Kevin Hoarau, Pierre Ugo Tournoux, and Tahiry Razafindralambo. BGNN: Detection of BGP Anomalies Using Graph Neural Networks. In *ISCC '22*, 2022.
- [25] Thomas Holterbach, Emile Aben, Cristel Pelsser, Randy Bush, and Laurent Vanbever. Measurement Vantage Point Selection Using A Similarity Metric. In *ANRW '17*, 2017.
- [26] Thomas Holterbach, Thomas Alfroy, Amreesh Phokeer, Alberto Dainotti, and Cristel Pelsser. A System to Detect Forged-Origin BGP Hijacks. In *NSDI'24*, 2023.
- [27] Zitong Jin, Xingang Shi, Yan Yang, Xia Yin, Zhiliang Wang, and Jianping Wu. TopoScope: Recover AS Relationships From Fragmentary Observations. In *IMC '20*, 2020.

- [28] Thomas Krenc, Robert Beverly, and Georgios Smaragdakis. Keep Your Communities Clean: Exploring the Routing Message Impact of BGP Communities. In *CoNEXT '20*, 2020.
- [29] Craig Labovitz, Abha Ahuja, Abhijit Bose, and Farnam Jahanian. Delayed Internet Routing Convergence. In *SIGCOMM CCR*, 2000.
- [30] Franziska Lichtblau. From the Edge to the Core : towards informed Vantage Point selection for Internet measurement studies. In *Ph.D. Thesis*, 2021.
- [31] Matthew Luckie, Bradley Huffaker, Amogh Dhamdhare, Vasileios Giotsas, and kc claffy. AS Relationships, Customer Cones, and Validation. In *IMC '13*, 2013.
- [32] Z. Morley Mao, Lili Qiu, Jia Wang, and Yin Zhang. On AS-Level Path Inference. In *SIGMETRICS '05*, 2005.
- [33] University of Oregon. Route Views Peers list. 2023. <http://www.routeviews.org/peers/peering-status.html>.
- [34] Ricardo Oliveira, Mohit Lad, Beichuan Zhang, Dan Pei, Daniel Massey, and Lixia Zhang. Placing BGP monitors in the Internet. In *Technical No. UCLA, TR*, 2006.
- [35] Ricardo Oliveira, Beichuan Zhang, Dan Pei, Rafit Izhak-Ratzin, and Lixia Zhang. Quantifying path exploration in the internet. In *ACM IMC'06*, 2006.
- [36] Chiara Orsini, Alistair King, Danilo Giordano, Vasileios Giotsas, and Alberto Dainotti. BGPStream: A Software Framework for Live and Historical BGP Data Analysis. In *IMC '16*, 2016.
- [37] Lars Prehn and Anja Feldmann. How Biased is Our Validation (Data) for AS Relationships? In *IMC '21*, 2021.
- [38] RIPE. RIPE RIS Raw Data. 1. <https://www.ripe.net/data-tools/stats/ris/>.
- [39] RIPE. The RIPE Atlas measurement platform. 1. <https://atlas.ripe.net/>.
- [40] RIPE. YouTube Hijacking: A RIPE NCC RIS case study. 2018. <http://www.ripe.net/internet-coordination/news/industry-developments/>.
- [41] RIPE. Per-peer dump files. 2023. https://ris.ripe.net/docs/40_Prototypes/10_per_peer_dumps.html.
- [42] RIPE. RIPE RIS Peers list. 2023. <https://www.ris.ripe.net/peerlist/>.
- [43] Matthew Roughan, Simon Jonathan Tuke, and Olaf Maennel. Bigfoot, Sasquatch, the Yeti and other missing links: what we don't know about the AS graph. In *IMC '08*, 2008.
- [44] Pavlos Sermpezis, Vasileios Kotronis, Petros Gigis, Xenofontas Dimitropoulos, Jae Hyun Park, Danilo Calese, Alistair King, and Alberto Dainotti. ARTEMIS: Neutralizing BGP hijacking within a minute. In *ToN*, 2018.
- [45] Pavlos Sermpezis, Lars Prehn, Sofia Kostoglou, Marcel Flores, Athena Vakali, and Emile Aben. Bias in Internet Measurement Platforms. In *TMA'23*, 2023.
- [46] Anant Shah, Romain Fontugne, and Christos Papadopoulos. Towards Characterizing International Routing Detours. In *Asian Internet Engineering Conference '16*, 2016.
- [47] Y. Shavitt and U. Weinsberg. Quantifying the Importance of Vantage Points Distribution in Internet Topology Measurements. In *INFOCOM*, 2009.
- [48] Mattia Tantardini, Francesca Ieva, Lucia Tajoli, and Carlo Piccardi. Comparing methods for comparing networks. In *Nature*, 2019.
- [49] Ars Technica. Russian-controlled telecom hijacks financial services' internet traffic. 2017. <https://arstechnica.com/security/2017/04/>.
- [50] Krenc Thomas, Luckie Matthew, Marder Alexander, and kc Claffy. Coarse-grained Inference of BGP Community Intent. In *IMC'23*, 2023.
- [51] Oregon Univ. Route Views Project. 2021. www.routeviews.org/.
- [52] Ying Zhang, Zheng Zhang, Z. Morley Mao, Y. Charlie Hu, and Bruce M. Maggs. On the impact of route monitor selection. In *IMC '07*, 2007.
- [53] Xiaoliang Zhao, Dan Pei, Lan Wang, Dan Massey, Allison Mankin, S. Felix Wu, and Lixia Zhang. An Analysis of BGP Multiple Origin AS (MOAS) Conflicts. In *IMW '01*, 2001.

Appendix

A Survey

Detailed methodology. We selected eleven papers and classified them based on how authors collected the BGP data (categories C_1 and C_2). We then emailed the authors and asked them about their experience with using BGP routes from RIS and RV. We did not have answers for three papers.

We promised to share the answers of the participants in an anonymized fashion. Thus, we do not show parts of a few answers that would make de-anonymization possible. However, the missing parts never change the main message conveyed in the answers.

Detailed answers. Table 6 lists the questions we asked the participants of our survey along with their detailed answers. We color the answers based on whether they are in favor (green) of using a tool such as MVP or not (red). Neutral answers are colored in blue. The vast majority of the answers indicate that MVP would be beneficial for users and improve the quality of their measurement studies.

B Extended evaluation

In this section, we evaluate the performances of MVP_{v6}^{v4} (Table 7) and MVP_{v6} (Table 8) on the five use cases presented in §9.1, namely transient paths detection (*I*), MOAS detection (*II*), AS topology mapping (*III*), traffic engineering detection (*IV*), and unnecessary updates detection (*V*). Similarly to §9.1, we compare MVP_{v6} and MVP_{v6}^{v4} against the three naive baselines (random, AS-distance, and unbiased) as well as the eight *greedy specific* VPs selection strategies (three optimized for Def. 1, 2, and 3 and one optimized for each of the five use cases). We present the results in terms of data *reduction factor*, as defined in §9.1.

MVP_{v6}^{v4} and MVP_{v6} outperform the three naive baselines for every objective. For MVP_{v6}^{v4} , the *reduction factor* can be as high as 6.57 when trying to detect 50% of the traffic engineering paths while for MVP_{v6} it can be as high as 5.05 when trying to map 50% of the AS topology. On average, MVP_{v6}^{v4} only needs 41.6% of the data (*reduction factor* of 2.4) required by a naive baseline to meet the same objective while MVP_{v6} needs 44.5% (*reduction factor* of 2.26).

MVP_{v6}^{v4} and MVP_{v6} prevent overfitting. For the vast majority of the objectives, *greedy specific* performs better than MVP_{v6}^{v4} or MVP_{v6} only for the use cases for which it is optimized. There are a few cases where *greedy specific* performs better than MVP_{v6}^{v4} or MVP_{v6} for a use case that it does not optimize. For instance, MVP_{v6} needs to process 20% (*reduction factor* of 0.8) more data than *greedy specific* optimized for use case *I* to detect 90% of the MOAS (use case *II*). However, in the vast majority of the cases, both MVP_{v6}^{v4} and MVP_{v6} outperform the *greedy specifics*. For instance, MVP_{v6} only needs 26% (*reduction factor* of 3.74) of the volume required by the *greedy specific* optimized for use case *IV* to detect 90% of the MOAS (use case *II*). These results show that MVP does not overfit while *greedy specific* does.

Collection strategy	Questions asked	Collected answers
C₁: All routes and subset of VPs (seven papers)	Why did you use a subset of the VPs ?	To speed up data processing (x2) For disk space and time efficiency (x1) I thought the rest would be similar (x1) I did not manage to use them all (x2)
	How did you select your VPs ?	I took them randomly (x2) I do not remember (x2) It was arbitrary: my script partially failed (x1) I took geographically distant BGP collectors (x1) I did not manage to use VPs from one data provider (x1)
	Do you think more VPs would improve the quality of your results?	Yes (x4) Results would be similar, but it can help to find corner cases (x1) Yes, but not significantly (x1) I am not sure (x1)
	Would you have used more VPs if you could?	Yes (x4) Yes, I'd love to (x1) Definitely (x1) I am not sure, but I don't think so (x1)
C₂: Limited duration of experiment (five papers)	Was the processing time a factor that you considered when you decided on the duration of your measurement study?	Yes (x3)
	Do you think extending the duration of your measurement study would improve the quality of your results?	Yes (x2) Yes, especially for rare events (x1) Potentially (x1) Yes, but not significantly (x1)
	Would have extended the duration of your measurement study if you had more resources?	Yes (x2) Yes, but it depends on the time remaining before the deadline (x1) I think so, but also if I had more time before the deadline (x1)
All eight papers	Do you find the data from RIS and RouteViews expensive to process in terms of computational resources?	Yes (x1) Yes, CPU and storage (x2) Yes, the storage cost and the download cost are very large (x1) CPU is the main issue (x1) RIS data takes a lot of time to download, especially when we need data for multiple days (x1) Not the worst, but we definitely need a resourceful server if we want to catch some deadline (x1) We did that in a server so that was not a huge issue (x1) No (x1)
	Is there any additional challenge that you encountered when processing the BGP data from RIS and RouteViews?	Our team used Spark clusters and Python but it was too slow (x1) We had to download the data from all VPs as there is no optimal solution for selecting them, the storage overhead and time overhead were extremely high (x1) It'll be helpful to make processing faster and less resource-consuming (x1) Too many duplicate announcements make processing harder (x1) Variable sizes of update files exacerbate scheduling parallelization (x1) RIS took a lot longer than RouteViews (x1) We had issues when collecting updates in real-time (x1) We had to deal with bugs in BGPdump (x1) Broken data feeds and data cleanup is also an issue that we need to take care of (x1) Our study was done pre-BGPStream, which would have helped quite a bit already (x1)

Table 6: An exhaustive list of the questions asked to the participants of the survey along with their detailed answers. We color an answer in (bold) green if it (strongly) motivates the usage of a tool such as MVP. Blue answers are neutral, i.e., they do not motivate MVP but also do not disincentive it. Finally, (bold) red answers (strongly) disincentive the usage of a tool such as MVP.

Use case	Objective	Naives baselines			Greedy specifics use cases (§9.1)					Greedy specifics Def. (§4)		
		Random	AS-distance	unbiased	<i>I</i>	<i>II</i>	<i>III</i>	<i>IV</i>	<i>V</i>	Def. 1	Def. 2	Def. 3
Transient path detection (<i>I</i>)	50 %	1.32	1.87	1.94	0.61	1.19	1.11	1.36	1.21	2.08	2.24	1.79
	70 %	1.38	1.62	1.83	0.74	1.30	1.15	1.40	1.18	1.78	1.97	1.53
	90 %	1.16	1.42	1.40	0.71	1.39	1.74	1.69	1.21	1.34	1.36	1.40
MOAS detection (<i>II</i>)	50 %	1.93	3.38	4.03	1.95	0.78	1.41	2.05	1.37	3.34	3.21	2.88
	70 %	1.96	3.49	4.16	2.14	0.68	1.91	2.52	1.56	2.91	2.60	2.81
	90 %	1.16	1.69	2.07	1.52	0.69	1.68	1.87	1.53	1.31	1.25	1.40
AS topology mapping (<i>III</i>)	50 %	2.47	2.90	2.72	1.18	1.02	0.58	1.45	1.41	2.38	2.30	2.16
	70 %	2.27	2.52	2.29	1.26	1.14	0.68	1.25	1.19	2.03	1.71	2.03
	90 %	1.71	1.85	1.78	1.14	1.13	0.82	1.17	1.15	1.62	1.61	1.56
Traffic engineering detection (<i>IV</i>)	50 %	3.77	6.57	4.43	3.21	1.89	1.47	0.47	2.57	3.43	2.89	2.57
	70 %	2.34	3.17	2.56	2.20	1.60	1.93	0.35	2.06	1.97	2.01	2.05
	90 %	1.90	2.02	1.76	2.02	1.78	1.94	0.31	1.93	2.13	1.67	1.93
Unnecessary updates detection (<i>V</i>)	50 %	2.13	4.10	3.15	2.41	2.54	2.72	3.12	0.41	2.94	2.78	2.83
	70 %	1.27	1.95	1.80	1.47	1.12	1.28	1.59	0.35	1.71	1.81	1.57
	90 %	1.01	1.29	1.35	1.04	0.85	0.96	1.09	0.46	1.00	1.11	1.11

Table 7: Data reduction factor for MVP_{v6}^{v4} compared to several baselines for five use cases. MVP_{v6}^{v4} enables to detect 70% of the MOAS using only 28.6% (*reduction factor* of 3.49) of the volume required by the AS distance baseline to meet the same objective. The average reduction factor over all objectives and naive baselines is 2.25.

Use case	Objective	Naives baselines			Greedy specifics use cases (§9.1)					Greedy specifics Def. (§4)		
		Random	AS-distance	unbiased	<i>I</i>	<i>II</i>	<i>III</i>	<i>IV</i>	<i>V</i>	Def. 1	Def. 2	Def. 3
Transient path detection (<i>I</i>)	50 %	1.43	1.65	2.29	0.44	1.11	1.14	1.67	1.20	1.56	1.36	1.90
	70 %	1.71	1.84	2.00	0.64	1.59	1.96	2.88	2.25	1.75	1.86	1.67
	90 %	1.52	1.43	1.42	0.62	1.49	1.49	1.79	1.48	1.51	1.72	2.11
MOAS detection (<i>II</i>)	50 %	1.94	1.65	2.37	1.10	0.21	0.36	1.56	2.33	1.04	0.73	1.30
	70 %	4.24	1.70	3.25	1.26	0.51	1.05	4.98	3.38	4.20	3.71	4.13
	90 %	3.03	1.75	2.19	0.80	0.53	2.67	3.74	2.56	3.54	3.69	3.84
AS topology mapping (<i>III</i>)	50 %	4.45	3.68	5.05	1.49	0.72	0.54	2.41	3.03	1.92	1.65	3.29
	70 %	2.83	3.26	3.14	1.18	1.14	0.73	2.07	1.38	2.27	2.17	2.48
	90 %	1.86	2.00	1.99	1.10	1.12	0.86	1.25	1.30	1.56	1.70	2.02
Traffic engineering detection (<i>IV</i>)	50 %	2.27	1.68	1.34	2.68	0.51	0.58	0.12	1.89	0.75	0.95	0.53
	70 %	3.76	5.14	2.86	3.03	2.64	3.03	0.30	4.66	2.07	1.61	1.14
	90 %	1.29	1.36	1.18	1.49	1.49	1.49	0.65	1.49	0.88	0.56	1.19
Unnecessary updates detection (<i>V</i>)	50 %	1.45	2.19	2.63	1.57	2.94	1.97	3.21	0.22	2.44	2.70	1.95
	70 %	1.37	2.13	2.09	1.79	1.98	2.07	2.26	0.31	1.91	2.18	1.82
	90 %	1.23	1.46	1.58	1.34	1.39	1.46	1.69	0.50	1.83	1.49	1.45

Table 8: Data reduction factor for MVP_{v6} compared to several baselines for five use cases. MVP_{v6} enables to detect 90% of the MOAS using only 33% (*reduction factor* of 3.03) of the volume required by the random selection to meet the same objective. The average reduction factor over all objectives and naive baselines is 2.25.