On Suitability of Euclidean Embedding for Host-Based Network Coordinate Systems

Sanghwan Lee, Zhi-Li Zhang, *Member, IEEE*, Sambit Sahu, *Member, IEEE*, and Debanjan Saha, *Fellow, IEEE*

***Abstract*—In this paper, we investigate the suitability of embedding Internet hosts into a Euclidean space given their pairwise distances (as measured by round-trip time). Using the classical scaling and matrix perturbation theories, we first establish the (sum of the) magnitude of *negative* eigenvalues of the (doubly centered, squared) distance matrix as a measure of suitability of Euclidean embedding. We then show that the distance matrix among Internet hosts contains negative eigenvalues of *large magnitude,* implying that embedding the Internet hosts in a Euclidean space would incur relatively large errors. Motivated by earlier studies, we demonstrate that the inaccuracy of Euclidean embedding is caused by a large degree of *triangle inequality violation* (TIV) in the Internet distances, which leads to negative eigenvalues of large magnitude. Moreover, we show that the TIVs are likely to occur *locally;* hence the distances among these close-by hosts cannot be estimated accurately using a *global* Euclidean embedding. In addition, increasing the dimension of embedding does not reduce the embedding errors. Based on these insights, we propose a new hybrid model for embedding the network nodes using only a two-dimensional Euclidean coordinate system and small *error adjustment terms*. We show that the accuracy of the proposed embedding technique is as good as, if not better than, that of a seven-dimensional Euclidean embedding.**

***Index Terms*—Euclidean embedding, suitability, triangle inequality.**

I. Introduction

ESTIMATING distance (e.g., as measured by round-trip time or latency) between two hosts (referred as nodes hereafter) on the Internet in an accurate and scalable manner is crucial to many networked applications, especially to many emerging overlay and peer-to-peer applications. One promising approach is the *coordinate (or Euclidean embedding) based network distance estimation* because of its simplicity and scalability. The basic idea is to embed the Internet nodes in a Euclidean space with an appropriately chosen dimension based on the pairwise distance matrix. The idea was first proposed by Ng *et al.* [2]. Their scheme, called Global Network Positioning (GNP), employs the least square multidimensional scaling technique to construct a low-dimensional Euclidean coordinate system and approximate the network distance between any two nodes by the Euclidean distance between their respective coordinates. To improve the scalability of GNP, [3] and [4] propose more efficient coordinate computation schemes using principal component analysis. Both schemes are in a sense centralized. Methods for distributed construction of Euclidean coordinate systems have been developed in [5] and [6]. In addition, [5] proposes to use height vector to account for the effect of access links, which are common to all the paths from a host to the others.

While most studies have focused on improving the accuracy and usability of the coordinate-based distance estimation systems, others have demonstrated the potential limitations of such schemes. For example, [7] shows that the amount of the triangle inequality violations (TIVs) among the Internet hosts is nonnegligible and illustrates how the routing policy produces TIVs in the real Internet. They *conjecture* that TIVs make Euclidean embedding of network distances less accurate. Reference [8] proposes new metrics such as relative rank loss to evaluate the performance and show that such schemes tend to perform poorly under these new metrics. A brief survey of various embedding techniques is found in [8]. In addition, [9] claims that the coordinate-based systems are in general inaccurate and incomplete, and therefore proposes a lightweight *active* measurement scheme for finding the closest node and other related applications.

In spite of the aforementioned research on the coordinate-based network distance estimation schemes, regardless of whether they advocate or question the idea, no attempt has been made to systematically understand the *structural* properties of Euclidean embedding of Internet nodes based on their pairwise distances: what contributes to the estimation errors? Can such errors be reduced by increasing the dimensionality of embedding? More fundamentally, how do we quantify the suitability of Euclidean embedding? We believe that such a systematic understanding is crucial for charting future research directions in developing more accurate, efficient, and scalable network distance estimation techniques. This paper is a first attempt in reaching such an understanding and proposes a simple new *hybrid* model that combines global Euclidean embedding with local non-Euclidean error adjustment for more accurate and scalable network distance estimation.

Manuscript received February 24, 2008; revised September 28, 2008; approved by IEEE/ACM Transactions on Networking Editor A. Feldmann. First published September 11, 2009; current version published February 18, 2010. This work was supported in part by Research Program 2008 of Kookmin University, Korea; by the National Science Foundation under Grants ITR-0085824, CNS-0435444, CNS-0626812, and CNS-0626808; and by an IBM Faculty Partnership Award. An earlier version of this paper was presented at ACM Sigmetrics/Performance 2006 [1].

S. Lee is with the School of Computer Science, Kookmin University, Seoul 136-782, Korea (e-mail: sanghwan@kookmin.ac.kr).

Z.-L. Zhang is with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: zhzhang@cs.umn.edu).

S. Sahu and D. Saha are with the IBM T. J. Watson Research Center, Hawthorne, NY 10532 USA (e-mail: sambits@us.ibm.com; dsaha@us.ibm.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2009.2023322

The contributions of this paper are summarized as follows. First, by applying the classical scaling and matrix perturbation theory, we establish the (sum of the) magnitude of *negative* eigenvalues of an appropriately transformed, squared distance matrix as a measure of suitability of Euclidean embedding. In particular, the existence of negative eigenvalues with large magnitude indicates that the set of nodes cannot be embedded well in a Euclidean space with small absolute errors.

Secondly, using data from real Internet measurement, we show that the distance matrix of Internet nodes indeed contains negative eigenvalues of large magnitude. Furthermore, we establish a connection between the degree of TIVs in the Internet distances to the magnitude of negative eigenvalues and demonstrate that the inaccuracy of Euclidean embedding is caused by a large degree of TIVs in the network distances, which leads to negative eigenvalues of large magnitude. We also show that TIVs cause the embedding schemes to be suboptimal in that the sum of estimation errors from a host is either positive or negative (far from zero), which means that the estimations are biased.

Thirdly, we show that a majority of TIVs occur due to the nodes that are close by. By clustering nodes based on their distances, we find that while the distances between the nodes in the different clusters (the *intercluster* node distances) can be fairly well approximated by the Euclidean distance function, the *intracluster* node distances are significantly more *non-Euclidean,* as manifested by a much higher degree of TIVs and the existence of negative eigenvalues with considerably larger magnitude. Based on these results, we conclude that estimating network distances using coordinates of hosts embedded in a *global* Euclidean space is rather inadequate for close-by nodes.

As the last (but not the least) contribution of this paper, we develop a new hybrid model for embedding the network nodes: in addition to a low-dimensional Euclidean embedding (which provides a good approximation to the intercluster node distances), we introduce a locally determined (*nonmetric*) adjustment term to account for the non-Euclidean effect within the clusters. The proposed hybrid model is mathematically proved to always reduce the estimation errors in terms of *stress* (a standard metric for fitness of embedding). In addition, this model can be used in conjunction with any Euclidean embedding scheme.

The remainder of this paper is organized as follows. In Section II, we provide a mathematical formulation for embedding nodes in a Euclidean space based on their distances and apply the classical scaling and matrix perturbation theories to establish the magnitude of negative eigenvalues as a measure for suitability of Euclidean embedding. In Section III, we analyze the suitability of Euclidean embedding of network distances and investigate the relationship between triangle inequality violations and the accuracy. Section IV shows the accuracy of various Euclidean embedding schemes over various real measurement data sets. We show the clustering effects on the accuracy in Section V. We describe the new hybrid model for the network distance mapping in Section VI and conclude this paper in Section VII.

II. Euclidean Embedding and Classical Scaling

In this section, we present a general formulation of the problem of embedding a set of points (nodes) into a r-dimensional Euclidean space given the pairwise distance between any two nodes. In particular, using results from classical scaling and matrix perturbation theories, we establish the (sum of the) magnitude of negative values of an appropriately transformed, squared distance matrix of the nodes as a measure for the *suitability* of Euclidean embedding.

A. Classical Scaling

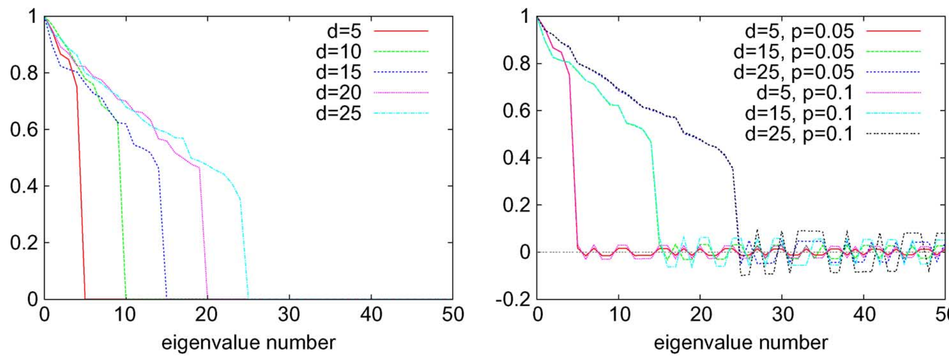
Given only the $n \times n$, *symmetric* distance matrix $D = [d_{ij}]$ of a set of n points from some arbitrary space, where d_{ij} is the *distance*[1] between two points $\mathbf{x}_i$ and $\mathbf{x}_j$, $1 \leq i, j \leq n$, we are interested in the following problem: can we embed the n points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ in an r-dimensional space for some integer $r \geq 1$ with *reasonably good accuracy?* To address this question, we need to first determine what is the appropriate dimension r to be used for embedding; given r thus determined, we then need to map each point $\mathbf{x}_i$ into a point $\tilde{\mathbf{x}}_i = (\tilde{x}_{i1}, \ldots, \tilde{x}_{ir})$ in the r-dimensional Euclidean space to minimize the overall error of embedding with respect to certain criterion of accuracy.

Before we address this problem, we first ask a more basic question. Suppose that the n points are actually from an r-dimensional Euclidean space, given *only* their distance matrix $D = [d_{ij}]$: is it possible to find out the original dimension r and recover their original coordinates in the r-dimensional space? Fortunately, this question is already answered by the theory of classical scaling [10]. Let $D^{(2)} = [d_{ij}^2]$ be the matrix of squared distances of the points. Define $B_D := -(1/2)JD^{(2)}J$, where $J = I - n^{-1}\mathbf{1}\mathbf{1}^T$, I is the unit matrix and $\mathbf{1}$ is an n-dimensional column vector whose entries are all one. J is called a centering matrix, as multiplying J to a matrix produces a matrix that has zero mean columns and rows. Hence B_D is a doubly centered version of $D^{(2)}$. A result from the classical scaling theory gives us the following theorem.

Theorem 1: If a set of n points $\{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}\}$ are from an r-dimensional Euclidean space, then B_D is semidefinite with exactly r *positive* eigenvalues (and all other eigenvalues are zero). Furthermore, let the *eigendecomposition* of B_D be given by $B_D = Q\Lambda Q^T = Q\Lambda^{1/2}(Q\Lambda^{1/2})^T$, where $\Lambda = [\lambda_i]$ is a diagonal matrix whose diagonal consists of the eigenvalues of B_D in decreasing order. Denote the diagonal matrix of the first r *positive* eigenvalues by Λ_+ and Q_+ the first r columns of Q. Then the coordinates of the n points are given by the $n \times r$ coordinate matrix $Y = Q_+\Lambda_+^{1/2}$. In particular, Y is a translation and rotation of the original coordinate matrix X of the n points.

Hence the above theorem shows that if n points are from a Euclidean space, then we can determine precisely the original dimension and recover their coordinates (up to a translation and rotation). The *contrapositive* of the above theorem states that if B_D is not semidefinite, i.e., it has *negative* eigenvalues, then the n points are *not* originally from an Euclidean space. A natural question then arises: *does the negative eigenvalues of B_D tell*

[1]We assume that the distance function $d(\cdot, \cdot)$ satisfies $d(x, x) = 0$ and $d(x, y) = d(y, x)$ (symmetry) but may violate the *triangle inequality* $d(x, z) \leq d(x, y) + d(y, z)$; hence d may not be *metric*.



(a) (b)

Fig. 1. Scree plots of the eigenvalues on data sets. Random points are generated in d-dimensional Euclidean space. The noise is computed as $dnoise_{xy} = d_{xy} + d_{xy} \times f$, where the noise factor f is uniformly randomly selected from a range of [0,p). $p = 0.05$ and $p = 0.1$ are used. (a) Random points from Euclidean space. (b) Random points with noise.

us how well a set of n *points can be embedded in a Euclidean space?* In other words, can they provide an appropriate measure for *suitability* of Euclidean embedding? We formalize this question as follows. Suppose the n points are from an r-dimensional Euclidean space but the actual distance $\tilde{d}_{ij}$ between two points $\mathbf{x_i}$ and $\mathbf{x_j}$ is "distorted" slightly from their Euclidean distance d_{ij}, e.g., due to measurement errors. Hence, intuitively, if the total error is small, we should be able to embed the n points into an r-dimensional Euclidean space with small errors. Using the matrix perturbation theory, in the following we show that *in such a case, the (doubly centered) squared distance matrix must have small negative eigenvalues*.

Formally, we assume that $\tilde{d}_{ij}^2 = d_{ij}^2 + e_{ij}$, where $|e_{ij}| \leq \epsilon/n$ for some $\epsilon > 0$. Hence $\tilde{D}^2 := [\tilde{d}_{ij}^2] = D^{(2)} + E$, where $E := [e_{ij}]$. A frequently used matrix norm is the *Frobenius norm* $\|E\|_F := \sqrt{\sum_i \sum_j |e_{ij}|^2} \leq \epsilon$. Then $B_{\tilde{D}} := -(1/2)J\tilde{D}^{(2)}J = B_D + B_E$, where $B_E := -(1/2)JEJ$. It can be shown that $\|B_E\|_F \leq \epsilon$. For $i = 1, 2, \ldots, n$, let $\tilde{\lambda}_i$ and λ_i be the ith eigenvalue of $B_{\tilde{D}}$ and B_D, respectively, where $\tilde{\lambda}_1 \geq \cdots \geq \tilde{\lambda}_n$ and $\lambda_1 \geq \cdots \geq \lambda_n$. Then the Wiedlandt–Hoffman theorem [11] states that $\sum_{i=1}^n (\tilde{\lambda}_i - \lambda_i)^2 \leq \|B_E\|_F^2$. Since $\lambda_i \geq 0$, we have

$$\sum_{\{i:\tilde{\lambda}_i<0\}} |\tilde{\lambda}_i|^2 \leq \sum_{\{i:\tilde{\lambda}_i<0\}} (-\tilde{\lambda}_i + \lambda_i)^2 \leq \sum_{i=1}^n (\tilde{\lambda}_i - \lambda_i)^2 \leq \|B_E\|_F^2 \leq \epsilon^2.$$

Hence the sum of the squared absolute values of the *negative* eigenvalues is bounded by the squared Frobenius norm of the (doubly centered) error matrix $\|B_E\|_F^2$, which is the sum of the (doubly centered) squared errors. In particular, the absolute value of any negative eigenvalue $|\tilde{\lambda}_i|$ is bounded by $\|B_E\|_F$. Hence if the total error (as reflected by $\|B_E\|_F^2$) is small and bounded by ϵ, then the negative eigenvalues of $\tilde{B}_D$ are also small and their magnitude is bounded by ϵ. Hence the *magnitude* of negative eigenvalues (and their sum) provides a measure of the *suitability* of Euclidean embedding: if a set of n points can be well- approximated by a Euclidean space with an appropriate dimension, then their associated doubly centered squared distance matrix only has negative eigenvalues of small magnitude, if any. On the other hand, the contrapositive of the above proposition leads to the following observation.

Theorem 1: If the doubly centered squared distance matrix of a set of n points has negative eigenvalues of *large* magnitude, then the set of n points cannot be embedded into a Euclidean space with a small total error (as measured by $\|B_E\|_F$). In other words, they are less amenable to Euclidean embedding.

In this derivation, we use *total* error ϵ. However, the total error can be from only a few distance estimations so that eigenvalue analysis can wrongfully conclude that the Euclidean embedding is not good for this distance matrix. Actually, the meaning of *good fitting* depends on the objectives of the embedding. Typical objective functions usually try to minimize the total sum of *squared* absolute errors or relative errors. In such a case, even if only a few distances happen to have really high error terms, the errors are distributed to a large number of points because these objective functions tend to prefer many small errors rather than a few large errors. As a consequence, when the total error is high (regardless of whether it is from a few sources or many sources), the embedding is difficult to find the original positions of the points in the Euclidean space. So the eigenvalue analysis is useful to measure the suitability of the Euclidean embedding computed by the embedding schemes of which objective functions are to minimize the total (sum of squared) error.

B. Illustration

We now generate some synthetic data to demonstrate how classical scaling can precisely determine the original dimensionality of data points that are from a Euclidean space. First, we generate 360 random points in a unit hyper cube with different dimensions and compute the corresponding distance matrix for each data set. Fig. 1(a) shows the *scree plot* of the eigenvalues obtained using classical scaling. The eigenvalues are normalized by the largest value (this will be the same for the rest of thise paper). We see from Fig. 1(a) that the eigenvalues vanish right after the dimensionality of the underlying Euclidean space where the data points are from, providing an unambiguous cutoff to uncover the original dimensionality. We now illustrate what happens when distances among data points

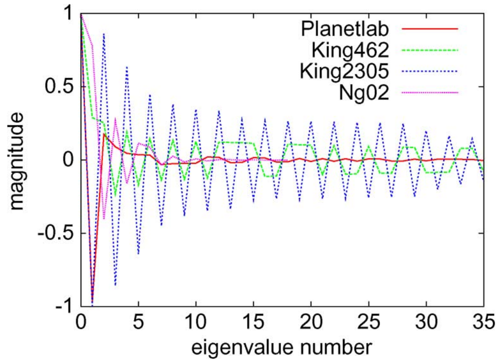


Fig. 2. The eigenvalue scree plot of network distance matrices.

TABLE I
DATA SETS USED IN THIS PAPER. THE NUMBER OF NODES IS CHOSEN TO MAKE THE MATRIX COMPLETE AND SQUARE

Data Set	Nodes	Date
King462[2]	462	8/9/2004
King2305 ([12])	2305	2004
PlanetLab[3]	148	9/30/2005
Ng02[4]	19	May 2001

are not precisely Euclidean (e.g., due to measurement errors). We add noise to the synthetically generated Euclidean data sets as follows: the noise component in the data is $d \times (1 + f)$, where d is the original Euclidean distance and f is a randomly selected number from $(-p, p)$. We use $p = 0.05$ and $p = 0.1$ for the illustration below. We observe in Fig. 1(b) that the first r eigenvalues are positive and are nearly the same as in the case without noise, where r represents the actual dimension of the data set. Beyond these eigenvalues, we observe only small negative eigenvalues. As the noise increases, the magnitudes of negative eigenvalues increase slightly. It is clear that as the data set deviates from Euclidean more, the magnitudes of the negative eigenvalues become larger.

III. SUITABILITY OF EUCLIDEAN EMBEDDING

To understand the suitability of Euclidean embedding of network distances, in this section, we perform eigenvalue analysis of the distance matrices and investigate how the TIVs affect the accuracy of the embedding, and thus the suitability of Euclidean embedding for a wide range of data sets.

To be specific, we apply eigenvalue analysis to show that the (doubly centered, squared) distance matrices of the data sets contain negative eigenvalues of relatively large magnitude. We then attribute existence of the negative eigenvalues of relative large magnitude to the large amount of triangle inequality violations existing in the data sets by showing i) embedding a subset of nodes without triangle inequality violations in a Euclidean space produces higher accuracy, and the associated distance matrix also contains only negative eigenvalues of much smaller magnitude; and ii) by increasing the degree of TIVs in a subset of nodes of the *same* size, the performance of Euclidean embedding degrades and the magnitude of the negative eigenvalues also increases.

We use four different data sets, which we refer to as *King462, King2305,* and *PlanetLab,* and *Ng02,* as listed in Table I. The King462 data set is derived from the data set used by Dabek *et al.* after removing the partial measurements to derive a 462 $\times$ 462 complete and square distance matrix among 462 hosts from the original 2000 DNS server measurements. Using the same refinement over the data set used in [12], we derive the King2305 data set, which is a 2305 $\times$ 2305 complete and square distance matrix. PlanetLab is derived from the distances measured among the PlanetLab nodes on September 30, 2005. We chose the minimum of the 96 measurement (one measurement per 15 min) data points for each measurement between node pairs. After removing the hosts that have missing distance information, we obtain a 148 $\times$ 148 distance matrix among 148 nodes. The Ng02 data set is obtained, which contains a 19 $\times$ 19 distance matrix. Even though the number of hosts is small in this data set, we have chosen this data set in order to compare with the results in other papers.

A. Eigenvalue Analysis

First, we perform eigenvalue analysis of the doubly centered, squared distance matrix $B_D = -JD^{(2)}J$. Fig. 2 shows the scree plot of the resulting eigenvalues, normalized by the eigenvalue of the largest magnitude $|\lambda_1|$, in decreasing order in the magnitude of the eigenvalues. We see that each of the data sets has one or more negative eigenvalues of relatively large magnitude that are at least about 20% (up to 100%) of $|\lambda_1|$, and the negative eigenvalue of largest magnitude is among the second and fourth largest in terms of magnitude. This suggests that the network distances are somewhat less suitable for Euclidean embedding. Hence it is expected that embedding the nodes in a Euclidean space would produce considerable amount of errors.

B. TIV Analysis

Motivated by earlier studies (e.g., [7]), which show that there is a significant amount of TIVs in the Internet distance measurement and attribute such TIVs to Internet routing policies,[5] here we investigate how the amount of TIVs in the data sets affects the suitability and accuracy of Euclidean embedding of network distances. In particular, we establish a strong correlation between the amount of TIVs and the magnitude of negative eigenvalues of the associated distance matrix. First, we analyze the amount of TIVs in the four data sets. For each dataset, we take a triple of nodes and check whether they violate triangle inequality. We then compute the fraction of such TIVs over all possible triples. Table II shows the results for the four data sets. We see that the fraction of TIVs in the King2305 data set is about 0.23, while for the other three data sets, it is around 0.12. Hence the triangle inequality violations are fairly prevalent in the data sets.

To investigate how the amount of TIVs affects the suitability and accuracy of Euclidean embedding—in particular, its impact on the magnitude of negative eigenvalues—we start with a subset of nodes without any triangle inequality violation (we

[2]http://pdos.lcs.mit.edu/p2psim/kingdata

[3]http://www.pdos.lcs.mit.edu/ strib/pl app/

[4]http://www-2.cs.cmu.edu/ eugeneng/ research/gnp/

[5]In particular, [7] shows that the Hot Potato Routing policy and the interplay between interdomain and introdomain routing can cause TIVs. It also shows that private peering between small ASs is another source of TIVs.

TABLE II
FRACTION OF TIVS OVER ALL TRIPLES OF NODES

Data Set	*Ng02*	*King2305*	*King462*	*Planetlab*
fraction	0.116	0.233	0.118	0.131

refer to such a subset of nodes as a *TIV-free* set). Ideally, we would like this subset to be as large as possible, namely, obtain the *maximal TIV-free (sub)set*. Unfortunately, finding the maximal TIV-free subset is NP-hard, as is stated in the following theorem (the proof of which is delegated to the Appendix).

Theorem 2: Finding the maximal TIV-free set problem is NP-complete.

Hence we have to resort heuristics to find a large TIV-free set. Here we describe three heuristic algorithms. The basic algorithm (referred to as *Algo 0*) is to randomly choose k nodes from a given set of n nodes and check whether any three nodes of these randomly selected k nodes violates the triangle inequality. If the triangle inequality is violated, the process is repeated again by randomly selecting another set of k nodes. If we find a TIV-free set of size k, we increase k by one and try again to attempt to find a larger set. Otherwise, the algorithm terminates after a prespecified number of failed tries and returns the TIV-free set of size $k-1$.

The second heuristic algorithm *(Algo 1)* is as follows. We start with a TIV-free set (initialized with two randomly selected nodes). From the remaining node set C (initially with $n-2$ nodes), we then randomly pick a new node and check to see whether it violates the triangle inequality with any two nodes in the existing TIV-free set. If yes, this node is removed from the remaining node set C. Otherwise, it is added to the TIV-free set (and removed from the remaining node set). The process is repeated until the remaining node set becomes empty.

The third heuristic algorithm *(Algo 2)* is slightly more sophisticated and works in a similar fashion as *Algo 1,* except that we do not choose nodes randomly for consideration. We start with an initial TIV-free set A of two nodes, where the two nodes are chosen such that the pair of nodes has the least number of TIVs with nodes in the remaining node set C. Given this pair of nodes, we remove all nodes in the remaining node set C that violate the triangle inequality with this pair of nodes. For each node c in C, we compute the number of nodes in C that violates triangle inequality with c and any two nodes in A. We pick the node c that has the smallest such number, add it to A, and remove it from C. We then purge all the nodes in C that violate the triangle inequality with c and any two nodes in A. We repeat the above process until C becomes empty.

For the data sets PlanetLab, King462, and King2305 (the Ng02 data set is not used since it is too small), the size of largest TIV-free sets found using the three heuristic algorithms is shown in Fig. 3. For each data set, Algo 0 only finds a TIV-free set of about ten nodes. Algo 2 finds the largest TIV-free sets for the King462 and King 2305 data sets, while Algo 1 finds the largest TIV-free set for the PlanetLab data set. For the following analysis, we use the largest TIV-free set found for each data set. Fig. 4(a) shows the scree plot of the eigenvalues for the associated (doubly centered, squared) distance matrix of the TIV-free node sets. We see that they all have only a small number of negative eigenvalues, and the magnitude of all the negative eigenvalues is also fairly small. Comparing with Fig. 2, either the number or the magnitude of negative eigenvalues is significantly reduced.

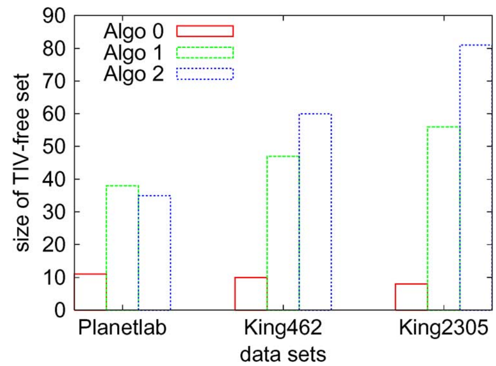


Fig. 3. Performance of the three heuristic algorithms.

The embedding accuracy of the TIV-free data sets is shown in Fig. 4(b). The relative errors, which are defined precisely in Section IV-A, are relatively small. For example, for the PlanetLab data set, in almost 98% of the cases, the relative errors are less than 0.2. We see that the Euclidean embedding of the TIV-free sets has a fairly good overall accuracy. However, Fig. 4(b) still shows nonnegligible errors for the TIV-free data sets. Since multidimensional scaling methods such as GNP can actually embed Euclidean data set without any error, this means that the errors of the TIV-free data set embedded by GNP are truly from the non-Euclidean characteristics of Internet routing. Actually, it is well known that the non-Euclidean *metric* space such as the shortest path routing is hard to embed into a low-dimensional Euclidean space without distortions or errors [13].

C. Correlation Between Negative Eigenvalues and Amount of TIVs

Next, we show how the amount of TIVs in a data set contributes to the magnitude of negative eigenvalues, and thereby the suitability and accuracy of Euclidean embedding. We use the King2305 data set as an example. The largest TIV-free set we found has 81 nodes. We fix the size of the node sets and randomly select *six* other node sets with exactly 81 nodes, but with *varying* amount of TIVs. The scree plots of the eigenvalues for the six node sets are shown in Fig. 5(a), and the cumulative relative error distributions of the corresponding Euclidean embedding are shown in Fig. 5(b). We see that with the increasing amount of TIVs, both the magnitude and number of negative eigenvalues increase. Not surprisingly, the overall accuracy of the Euclidean embedding degrades. In fact, we can mathematically establish a relation between the amount of TIVs and the sum of squared estimation errors as follows.

Lemma 1: If the distances t_a, t_b, t_c among three nodes violate the triangle inequality, i.e., $t_c > t_a + t_b$, the minimum squared estimation error of any metric (thus Euclidean) embedding of the three nodes is $(t_c - t_a - t_b)^2/3$.

Theorem 3: The sum of squared estimation errors of any Euclidean embedding of n nodes is at least $=$

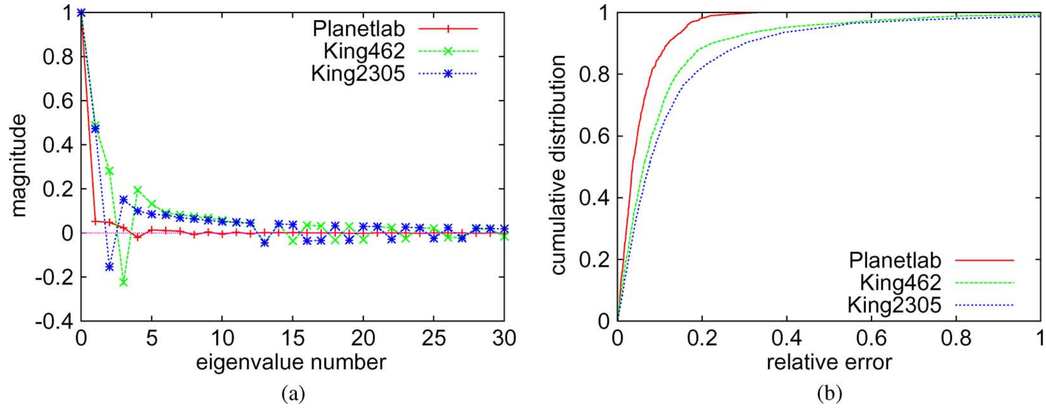


Fig. 4. Performance of embedding the TIV-free node sets using GNP. (a) Eigenvalues and (b) relative errors.

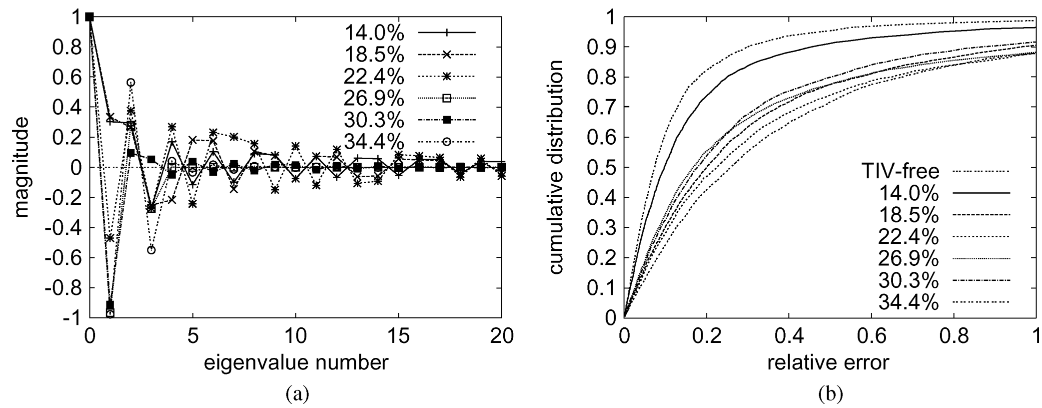


Fig. 5. Eigenvalue scree plots and cumulative distributions of relative errors of node sets with increasing fraction of TIVs. (a) Eigenvalues and (b) relative error.

$(1/(3(n-2)))\sum_{t \in V}(t_c - t_a - t_b)^2$, where V is the set of triples that violate triangle inequality, t_a, t_b, and t_c are the three distances of a triple $t \in V$, and $t_c > t_a + t_b$.

The proofs are delegated to the Appendix. Theorem 3 shows that as the amount of TIVs increases, the sum of the squared estimation errors also increases. A similar result can also be established for the sum of squared *relative* errors, the details of which are omitted here. As an aside, we note that this theorem holds not only for a Euclidean distance function but also for any *metric* distance function where the triangle inequality property is required. However, it should be noted that the lower bound computed in Theorem 3 is loose in some cases. For example, the lower bound for the TIV-free data set is zero, but the embedding has nonnegligible errors. Nonetheless, Theorem 3 sheds new light on the relationship between the accuracy and the amount of TIVs.

IV. Euclidean Embedding of Network Distances

In this section, we examine the accuracy of Euclidean embedding of network distances for a wide range of data sets. We consider five different metrics that we believe are useful for a variety of delay-sensitive applications.

A. Metrics for Goodness of Embedding

We consider four performance metrics—*stress*, *(cumulative) relative errors, relative rank loss* (RRL), and *closest neighbor loss* (CNL)—that have been introduced across various studies in the literature (e.g., [2]–[4] and [8]), as well as a new fifth metric, *skewness,* which we introduce in this paper to gauge whether an embedding is more likely to over- or underestimate the distances between the nodes. These five metrics are formally defined as follows.

- *Stress:* This is a standard metric to measure the overall fitness of embedding, originally known as *Stress-1* [10]

$$\text{Stress}-1 = \sigma_1 = \sqrt{\frac{\sum_{x,y}(d_{xy} - \hat{d}_{xy})^2}{\sum_{x,y} d_{xy}^2}} \tag{1}$$

  where d_{xy} is the actual distance between x and y and $\hat{d}_{xy}$ is the estimated one.
- *Relative error:* This metric is introduced in [2] and is defined as follows: for each pair of nodes x and y, the relative error in their distance embedding is given by $re_{xy} := |d_{xy} - \hat{d}_{xy}|/\min(d_{xy}, \hat{d}_{xy})$. Note that the denominator is the minimum of the actual distance and the estimated one.[6] The *cumulative distribution* of relative errors re_{xy} provides a measure of the overall fitness of the embedding.
- *Relative rank loss [8]:* RRL denotes the fraction of a pair of destinations for which their relative distance ordering, i.e., rank in the embedded space with respect to a source,

[6]In some literature, instead of $\min(d_{xy}, \hat{d}_{xy})$, d_{xy} is used. This usually produces smaller relative errors.

has changed compared to the actual distance measurement. For example, for a given source node, we take a pair of destinations and check which one is closer to the source in the real distances and the estimated distances. If the closest one is different, then the relative rank is defined to be lost. We compute the fraction of such relative rank losses for each source and plot the cumulative distribution of such rank losses among all sources as a measure of the overall fitness of the embedding.

- *Closest neighbor loss [8]:* For each source, we find the closest node in the original data set and the embedded space. If the two nodes are different, the closest neighbor is lost. The CNL metric is then defined as the fraction of sources that have the closest neighbor lost. As an extension to the original CNL metric in [8], we introduce a margin parameter δ: if the closest neighbor nodes in the original data set and the embedded space are different but the distance between the two nodes in the embedded space is within δ ms, we consider it as a nonloss; only if the distance between the two is more than δ ms do we consider it as a closest neighbor loss. Hence with $\delta = 0$, we have the original CNL. We expect that as δ increases, the CNL metric decreases.
- *Skewness:* We introduce a new metric *skewness* to gauge whether an embedding is more likely to over- or underestimate the distances between the nodes. For each node x, we define the *skewness* of embedding with respect to node x as follows: $s_x = \sum_{y \neq x}(d_{x,y} - \hat{d}_{x,y})/(n-1)$, where n is the total number of nodes. In other words, s_x is the average of the embedding errors between the real distances and the estimated distances between node x to all the other nodes. Clearly, when s_x is a large positive, the embedding method tends to underestimate the distances between node x to other nodes, and if it is a large negative, it tends to overestimate the distances between node x to other nodes. Note that $s_x = 0$ does not mean that there is no error but that the underestimates and overestimates are "balanced," i.e., canceled out. The distribution of skewnesses of all nodes then provides us with a measure of whether a given embedding method tends to under- or overestimate the real distances among the nodes.

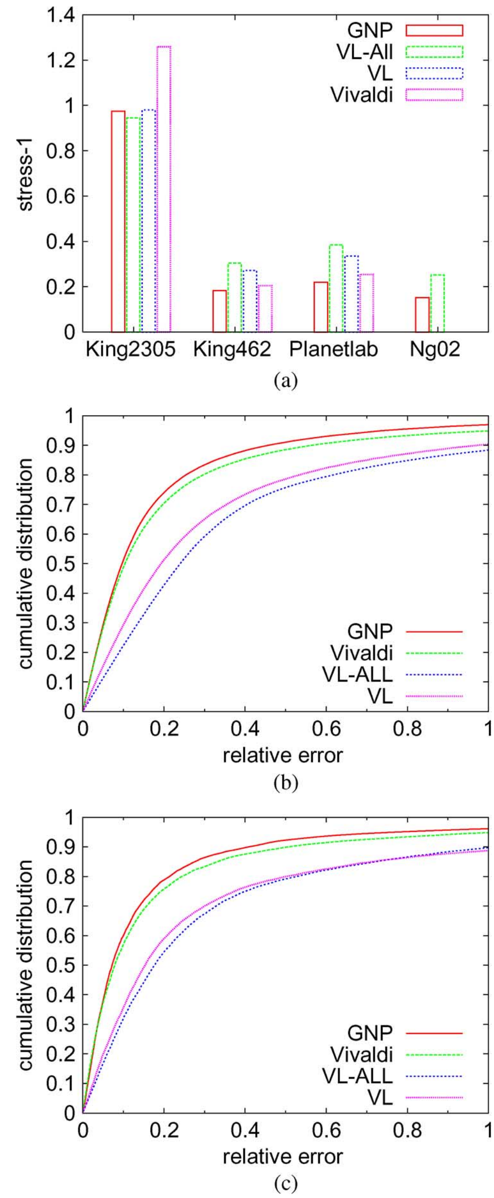


Fig. 6. Performance embedding. The number of dimensions is seven. (a) Stress, (b) relative error with King462, and (c) relative error with PlanetLab.

B. Performance of Euclidean Embedding

We apply the three most commonly used embedding methods proposed in the literature—GNP [2], Virtual Landmark (VL) [3], [4], and Vivaldi [5]—to the four data sets and compute their corresponding embedding errors as measured using the aforementioned five metrics. Following the results in [2]–[4], we choose seven as the dimension of Euclidean embedding for the three embedding methods: GNP, VL, and Vivaldi. More specifically, for the GNP and VL embedding methods, we use 20 landmarks randomly selected from the data set for computing the seven-dimensional Euclidean embedding. For the Vivaldi embedding methods, for each node, 20 neighbors are randomly selected and used for computing the seven-dimensional Euclidean embedding *plus the height vector*.[7] For the purpose of comparison and to eliminate the effect of landmark selection in the Virtual Landmark embedding method, we also use *all the nodes* as landmarks to compute the seven-dimensional Euclidean embedding[8]—this is referred to as *VL-ALL* in the figures that follow.

Fig. 6(a) shows the resulting overall stress of embedding using the four embedding methods GNP, VL, VL-ALL, and Vivaldi. Except for the King2305 data set, we see that the overall stress ranges from 0.2 to 0.5, which indicates that on the average, the estimations deviate from the original distances from 20% to 50%. For the King2305 data set, the overall stress is much larger (above 0.9) for all three methods. This is possibly due to the fact that in the King2305 data set, there are quite a few links with more than 90 s round-trip time, which may

[7]We used the Vivaldi implementation from http://www.eecs.harvard.edu/~syrah/nc/.

[8]Using all the nodes as landmarks (or "neighbors" in the case of Vivaldi) is only computationally feasible for the Virtual Landmark embedding method, *not* for GNP and Vivaldi.

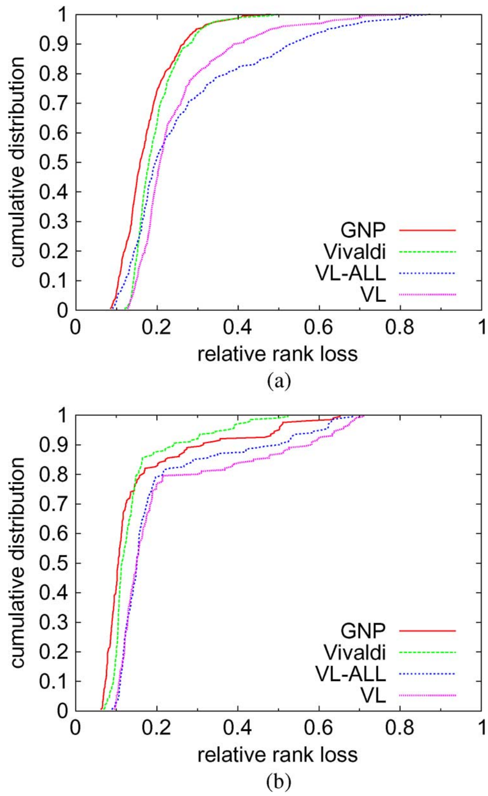


Fig. 7. The cumulative distribution of relative rank loss. (a) Relative rank loss with King462 and (b) relative rank loss with PlanetLab.

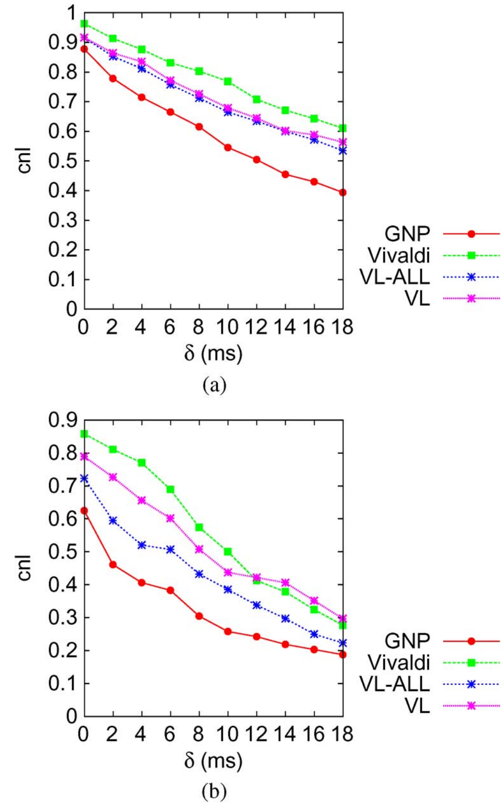


Fig. 8. Closest neighbor loss. Closest neighbor loss with (a) King462 and (b) PlanetLab.

produce many outliers that can significantly affect the overall stress—this is a major shortcoming of the stress metric (1). Note that as the Ng02 data set has only 19 nodes, the results for VL and Vivaldi are not available since there are fewer nodes than the required number of landmarks and neighbors (20 nodes). Fig. 6(b) and (c) shows the cumulative distributions of relative errors using GNP, VL, VL-ALL, and Vivaldi for the data sets King462 and PlanetLab, respectively. We see that all the embedding methods produce a relative error less than 0.5 for more than 75% (up to 90% in the case of GNP and Vivaldi) of the estimates.

Fig. 7(a) and (b) shows the cumulative distributions of relative rank losses using GNP, VL, VL-ALL, and Vivaldi for the data sets King462 and PlanetLab, respectively. We see that for all four methods, more than 70% of the sources have a fraction of relative rank losses less than 0.3–in other words, with respect to these sources, fewer than 30% of destination pairs have a different rank order in the embedded space from that in the original data set.

In terms of the CNL metric, from Fig. 8(a) and (b), we see that as the margin parameter δ increases, the fraction of CNLs improves for all embedding methods: embedding the PlanetLab nodes in a Euclidean space using GNP, 60% to 70% of the sources have a different closest neighbor node in the embedded space (i.e., when $\delta = 0$), but for only about 20% of the sources, the closest neighbor node in the original data set is more than 15 ms (i.e., $\delta = 15$ ms) away from the closest neighbor node in the embedded space. The CNLs of VL, VL-ALL, and Vivaldi lie above that of GNP.

Lastly, to compare the skewness of the embedding methods, we use the King462 data set as a representative example. Fig. 9 shows the results for three embedding methods GNP, VL-ALL, and Vivaldi (to avoid cluttering, we did not include the result for VL in the figure, as it has slightly worse performance than that of VL-ALL). To plot the distribution of the skewness s_x, we use bins of size of 10 ms that are centered at multiple of 10 ms, such as $[-15, -5)$, $[-5, 5)$, and $[5, 15)$. As can be seen in the figure, the GNP and Vivaldi embedding methods have the best performance, as the highest fraction of skewness values of the nodes falls within $[-5, 5)$, and the majority of the values are within the interval $[-5, 15)$. In contrast, however, the skewness values of VL-ALL are more widely spread, indicating that for a fraction of nodes, it tends to either under- or overestimate their distances to other nodes.

V. Local Non-Euclidean Effect

In this section, we dissect the data set further to find out whether the inherent clustering structure of Internet hosts contributes to the errors in the Euclidean embedding—in particular, what kinds of nodes are likely to contribute to the higher degree of TIVs, and whether increasing the dimension of the embedding helps improve the embedding performance.

The hosts in the Internet are clustered due to many factors such as geographical location, network topology, and routing policies. This clustering causes many hosts to have short distances among themselves and far longer distances to other hosts. To investigate the effect of host clustering on embedding accuracy, we first identify clusters within the network distances. For this, we apply the spectral clustering algorithm [14] to the

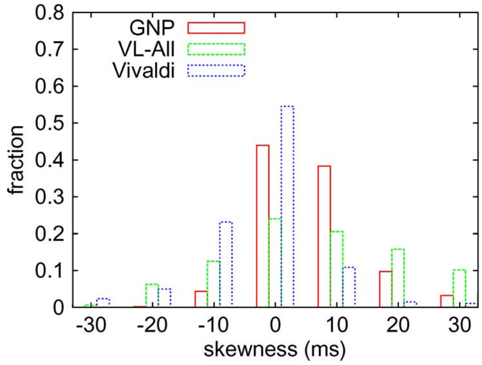


Fig. 9. Distributions of the skewness s_x under GNP, VL-ALL, and Vivaldi for the King462 data set.

King462 data set with the outliers[9] removed. In this experiment, 28 nodes out of 462 are removed. The algorithm[10] obtains *four* clusters for the King462 data set. We use a *grayscale* plot to show the existence of the clusters in the King462 data set with the outliers removed.

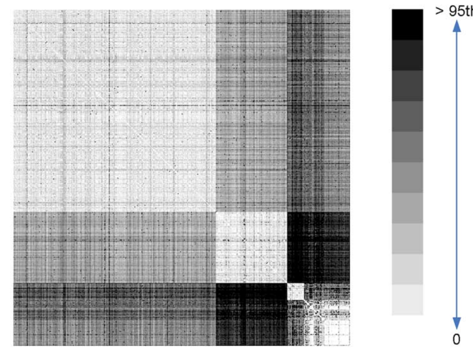
The square image in Fig. 10 is a graphical representation of the King462 distance matrix. In the square image, the vertical axis represents the source nodes and the horizontal axis represents the target nodes. The nodes are sorted by their clusters in such a way that the nodes in cluster 1 appear first, followed by the nodes in cluster 2, and so on. The nodes in the vertical and horizontal axes are in the same order. Each point (x, y) represents the distance between the corresponding two nodes x and y.

The distance is represented in a grayscale: white represents distance zero and black represents a distance larger than the ninety-fifth percentile. The interval between zero and the ninety-fifth percentile distance is divided into ten grayscales (with a total of 11 grayscales), with increasing darkness from white to black (beyond the ninety-fifth percentile distance). We can clearly see the *four* white blocks on the diagonal, each of which represents the distance matrix of each cluster. The table in Fig. 10 shows the median distances between the nodes within and across the *four* clusters in a numeric form. As is expected, the intracluster median distances shown in the diagonal entries of the table are much smaller than the intercluster median distances.

To illustrate the characteristics of the individual clusters, in Fig. 11, we show the eigenvalue scree plots of the distance matrices obtained from the four clusters in the King462 data set. Compared to Fig. 2, we see that the magnitudes of the negative eigenvalues of the clusters are larger than those of the whole data set. The "non-Euclideanness" amplifies within each cluster. It suggests that the intracluster distances are much harder to embed into the Euclidean space. This can be easily observed by looking at the relative errors of the embedding. Fig. 12 shows

[9]Outliers are defined as those nodes whose distance to the eighth nearest nodes is larger than a threshold. The reason to choose the eighth node is that we want the node to have at least a certain number of neighbors (in this paper, the number is eight) within the threshold.

[10]The algorithm takes as input a parameter K, the number of clusters, and produces *up to* K as a result. We have experimented with $K = 3$ to 7, and the algorithm in general produces three to four "relatively big" clusters for the three data sets King462, King2305, and PlanetLab.



	c1	c2	c3	c4
c1	62.099	154.287	254.469	212.538
c2	154.287	60.681	376.146	321.508
c3	254.469	376.146	61.194	238.938
c4	212.538	321.508	238.938	61.950

Fig. 10. Distances between each pair of nodes in King462 data set after removing outliers. White represents distance zero and black represents ninety-fifth percentile or higher distances. Median distances (in milliseconds) among the nodes of the intra- and interclusters are shown in the table.

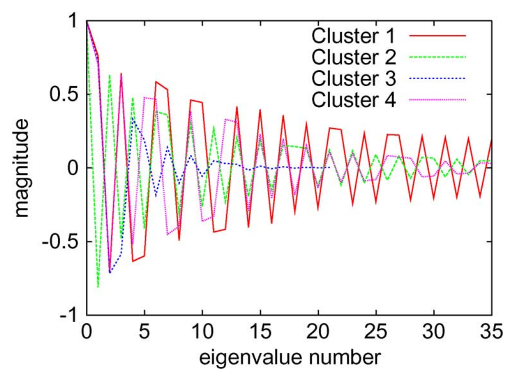


Fig. 11. Scree plot of the eigenvalues of CS on the four clusters of the King462 data set after removing 28 outliers : Cluster 1 (261 nodes), Cluster 2 (92 nodes), Cluster 3 (22 nodes), and Cluster 4 (59 nodes).

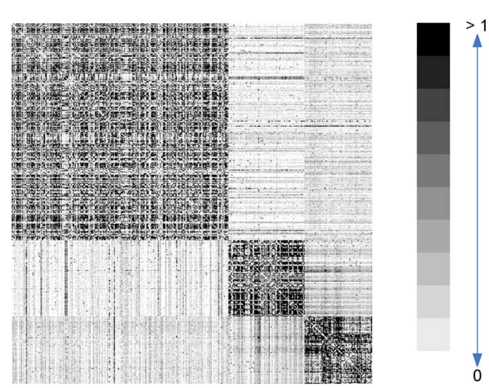


Fig. 12. Relative errors between each pair of nodes in King462 data set without outliers. White represents relative error zero and black represents relative error one or larger. Virtual Landmark method with seven dimensions is used.

the relative errors in a grayscale matrix for the King462 data set, where VL-ALL is used for the embedding. The pure black color represents the relative error of 1.0 or larger, and ten grayscales are used for relative errors between zero and one. We see that

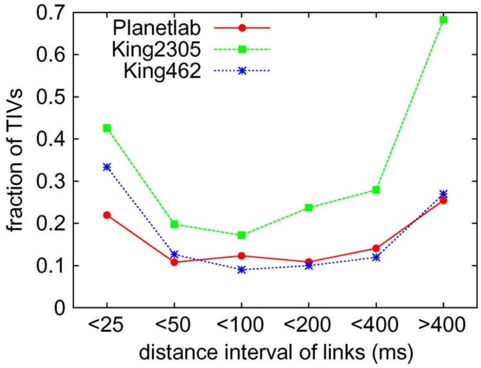


Fig. 13. Average fraction of TIVs at each distance interval.

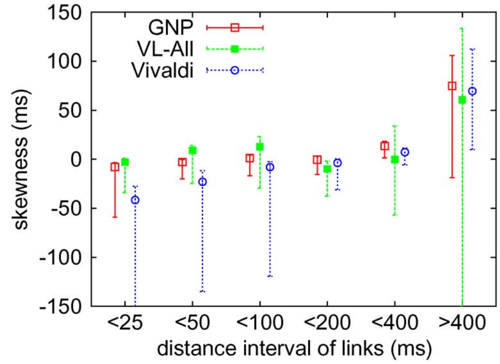


Fig. 14. The thirtieth, fiftieth, and seventieth percentiles of the skewness measures at each distance interval. The data set is King462.

the relative errors of the intracluster estimations are larger than those of intercluster estimations.

We next examine which nodes are more likely to contribute towards the TIVs. As we shall illustrate next, the high errors in the intracluster distance estimation and the large magnitudes of the negative eigenvalues can be explained by the varied number of TIVs over the different distances. Intuitively, a TIV is likely to occur if the distance between two nodes is very short or very large compared to the other two distances for a given triple of three nodes. Using this intuition, we proceed with our data analysis as follows: we divide the distances into six intervals: [0 ms, 25 ms), [25 ms, 50 ms), [50 ms, 100 ms), [100 ms, 200 ms), [200 ms, 400 ms), and [400 ms, ∞). We group all the pairs of nodes by their distance intervals. Then, for each pair of nodes, we compute the fraction of TIVs in conjunction with the rest of the nodes, i.e., we count how many nodes violate triangle inequality with the given pair. Lastly, we compute the average of the fractions of all the pairs in each interval. Fig. 13 shows the average fraction of TIVs in each distance interval. We observe that higher fractions of TIVs occur in the intervals [0, 25 ms) and [400, ∞) compared to other intervals. Since the fractions of pairs in [400, ∞) are quite small in all the data sets, reducing the errors in short distance estimations is thus much more crucial for the overall performance of embedding.

The above analysis illustrated that the distances among the intercluster nodes are more likely to be better approximated by their Euclidean coordinates, whereas Euclidean embedding of nodes within a cluster would likely provide a poor estimate of their distances. This seems to suggest that there is much stronger *local* "non-Euclidean effect" on the network distances. By local non-Euclidean effect, we mean that the embedding of local (short) distances into a Euclidean space is much harder than global (long) distances.

The "local non-Euclidean" effect can be also illustrated using the skewness metric. For each node, we compute its skewness to the nodes within the intervals mentioned in the previous sections. Then we plot the thirtieth percentile, the median, and the seventieth percentile of the skewness measures of all nodes using GNP, VL-ALL, and Vivaldi. As can be seen in Fig. 14, more skewnesses exist in the intervals [0, 25 ms) and [400, ∞) compared to other intervals. Furthermore, for short distances, the skewness measures are likely to be negative; and for large distances, the skewness measures are likely to be positive. Since each of the embedding methods tries to embed the nodes of TIVs in a Euclidean space by minimizing an error function, it is natural to lengthen the short distances and to shorten the long distances. This observation is also evident in the proof of Lemma 1 using the stress error function.

Furthermore, we have examined the impact of increasing dimension and using non-Euclidean distance functions such as the Minkowski p-norm on the accuracy of the estimation. Similar to the result in other papers such as [2], increasing dimension does not increase the accuracy, and the Minkowski p-norm does not help, either. We omit the result due to the page limitation.

VI. A Hybrid Model for Local Adjustment

The results from previous sections show that the existence of TIVs highly affects the accuracy of the Euclidean embedding (for that matter, any *metric* embedding). In particular, Euclidean embedding is fairly good at estimating network distances between nodes that are relatively far away (in different clusters), whereas it is rather poor at estimating local network distances (distance between nodes within a cluster). These observations inspire us to develop a hybrid embedding model, which incorporates a (non-Euclidean) localized adjustment term (LAT) into the distance estimation. We show that using only a two-dimensional Euclidean embedding plus the localized adjustment terms, we can obtain better performance than a pure Euclidean embedding with seven dimensions.

A. The Hybrid Model

The basic ideas behind our hybrid model are as follows: we first embed the network distances in a Euclidean space of d dimensions, and then for each node, we compute an adjustment term to account for the (local) non-Euclidean effect. Hence in our hybrid model, each node x has a d-dim Euclidean coordinate, $(x_1, x_2, \ldots, x_d)$, and a (non-Euclidean) adjustment e_x: we use $(x_1, x_2, \ldots, x_d; e_x)$ to denote the total "coordinate" of node x. The distance d_{xy} between two nodes x and y is then estimated by $\hat{d}_{xy} := d^E_{x,y} + e_x + e_y$, where $d^E_{x,y} = \sqrt{\sum_{k=1}^{d}(x_k - y_k)^2}$ is the Euclidean distance between x and y in the embedded d-dim Euclidean space. At first glance, e_x may look similar to the height vector in Vivaldi system [5], but actually it is quite different, as will be discussed later in this section. The key question in this model is how to define and determine e_x for each node

x. Ideally, we would like e_x to account for the "non-Euclidean" effect on the distance estimation errors to nodes within its own cluster. However, this requires us to know which cluster node x is in as well as the other nodes in its cluster. For simplicity, we derive e_x using all nodes as follows. We first compute ϵ_x, which minimizes the error function $E(x) = \sum_y (d_{xy} - (d^E_{xy} + \epsilon_x))^2$, where d_{xy} is the actual distance between x and y. It can be shown that the optimal ϵ_x is given by the average error in estimation

$$\epsilon_x = \frac{\sum_y (d_{xy} - d^E_{xy})}{n}. \tag{2}$$

We then set e_x to the half of ϵ_x, namely, $e_x = \epsilon_x/2$. In other words, $\hat{d}_{xy}$ can be rewritten as $d^E_{x,y} + (\epsilon_x + \epsilon_x)/2$. In short, we adjust the Euclidean estimation by the average of the two error terms of x and y. We have the following theorem that establishes the advantage of the hybrid model. The proof sketch is provided in the Appendix.

Theorem 4: The hybrid model using a d-dim Euclidean space and the adjustment term defined above reduces the squared stress of a pure d-dim Euclidean embedding by

$$\frac{4n\sum_x e_x^2 + 2n^2 \mathrm{Var}(e_x)}{\sum_{x,y} d^2_{xy}} \geqq 0$$

where $\mathrm{Var}(e_x) = \sum_x e_x^2/n - (\sum_x e_x/n)^2$.

Hence the larger the individual adjustment term $|e_x|$ (thus the average estimation error for each node x using the pure Euclidean embedding), the more performance gain the hybrid model attains. It should be noted that e_x can be positive or negative.[11]

In (2), e_x is determined by the measurement to all the other nodes in the system. In practice, however, this is not feasible nor scalable. Instead, we compute $\tilde{e}_x$ based on *sampled* measurements to a small number of randomly selected nodes. Let S denote the set of randomly sampled nodes. Then

$$\tilde{e}_x = \frac{\sum_{y \in S} (d_{xy} - d^E_{xy})}{2|S|}. \tag{3}$$

Hence, in practice, the hybrid model works as follows.

a) A number of landmarks are preselected and perform distance measurements among themselves to obtain a distance matrix. Using either Virtual Landmark or GNP, a d-dim Euclidean embedding of the landmarks is obtained and its coordinates determined.

b) Each node x measures its distance to the landmarks and computes its d-dim Euclidean coordinate $(x_1, x_2, \ldots, x_d)$; it then measures its distance to a small number of randomly selected nodes and computes $\tilde{e}_x$ using (3).

Note that, in a sense, the adjustment term is similar to the "height vector" introduced in Vivaldi [5]. However, there are several key differences. First, the computation of the local adjustment term is very simple and does not depend on the adjustment term of other nodes. Hence it does not require any iterative process to stabilize the adjustment term. In contrast, in Vivaldi—partly due to its distributed nature—a small change in the height vector of a node would affect the height vectors of the other nodes and requires an iterative process to stabilize the height vectors of all nodes. Secondly, the local adjustment terms *provably* improve the performance of network distance embedding, as shown in the above theorem. Another good feature of the local adjustment term is that it can be used with any other schemes, not just the coordinate-based schemes. As long as d^E_{xy} is the estimated distance based on the original scheme, the adjustment term can be computed as described above. In this sense, LAT is an *option* that can be used in conjunction with other schemes rather than a totally new scheme. Note that LAT can be used even with Vivaldi.

[11]It is possible that the estimated distance is negative due to negative LAT. In this case, we use the estimation of the Euclidean part as the estimated distance.

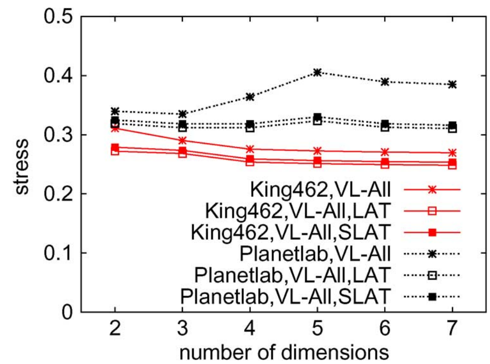


Fig. 15. Stress of Virtual Landmarks method over the number of dimensions. Both LAT and SLAT options are shown together.

B. Evaluation

We evaluate the performance gain obtained by using the LAT option in network distance embedding. For this purpose, we compare the stress of the VL-ALL method without LAT and the VL-ALL method with LAT, where the local adjustment term is computed using all the nodes. We vary the number of dimensions from two to seven. As can be seen in Fig. 15, the use of adjustment term (keys with LAT) reduces the stress significantly compared to the VL-All without LAT. In particular, when the original Euclidean embedding has high stress (large error), the reduction of stress is significant, which is expected from Theorem 4. Next, we evaluate the performance of LAT using only a small number of randomly selected nodes as in (3); we call this option sampled LAT (SLAT). Fig. 15 shows the stress of embedding using SLAT (keys with SLAT) over a different number of dimensions, where the adjustment term is computed using the measurement to ten randomly selected nodes. We see that the performance between LAT and SLAT is very close. This is quite expected because the average of a randomly sampled set is an unbiased estimation of the average of the entire set. This result indicates that the adjustment term can actually be computed quickly with a small number of additional measurements. The results also show that increasing the dimension of the Euclidean embedding does not help very much; in fact, a lower dimension Euclidean embedding plus the local adjustment terms is sufficient to improve the accuracy of the embedding significantly.

In addition to the improved overall stress, the local adjustment terms also improve the relative errors. As an example, Fig. 16(a) compares the cumulative distribution of the relative errors of

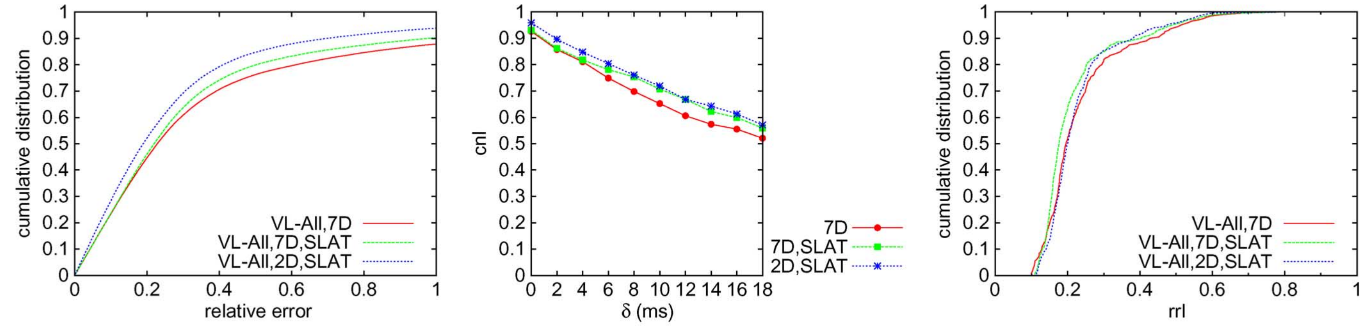


Fig. 16. Performance of VL-All method with SLAT on King462 data set. (a) Relative error, (b) CNL, and (c) RRL.

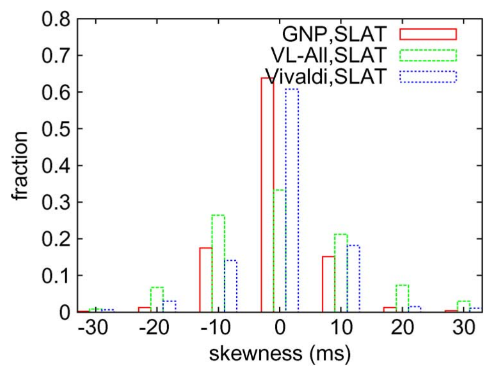


Fig. 17. Cumulative distributions of s_x under various schemes with the SLAT option. The data set is King462. The number of dimensions is seven.

the VL-ALL with seven dimensions (denoted "VL-ALL,7D") with that using the same method with seven dimensions plus SLAT (denoted as "VL-ALL,7D,SLAT") and with *only two dimensions* plus SLAT (denoted as "VL-ALL,2D,SLAT") for the King462 data set.[12] The VL-ALL with two dimensions plus SLAT attains better performance than that of the pure VL-ALL with seven dimensions. For example, ninetieth percentile relative error of "VL-ALL,2D,SLAT" is less than 0.6, but that of "VL-ALL,7D" is larger than 1.0. The performance of "VL-ALL,2D,SLAT" is even better than that of "VL-ALL,7D,SLAT," where seven dimensions is used. We conclude that adding a (non-Euclidean) local adjustment term is far more effective in improving the accuracy of embedding than adding additional dimensions. More in-depth analysis demonstrates that the performance gain comes largely from improved distance estimation for nodes within the same cluster. However, for the metric CNL, as can be seen in Fig. 16(b), the performance degrades with SLAT. It means that the SLAT option is not good for choosing the closest node. For the metric RRL, the performance with SLAT is a little better than that without SLAT, as can be seen in Fig. 16(c).

As implied in Theorem 4, a key reason that the hybrid model improves the performance (in relative error) of *any* Euclidean embedding method lies in the fact that it mitigates its "imperfect estimation"—namely, overestimates or underestimates—by introducing a (non-Euclidean) local adjustment term that smooths out (via averaging) the over- and underestimates. This effect can be seen using the skewness metric. Fig. 17 shows the resulting skewness measures of the various embedding methods with the SLAT option on, when applied to the King462 data set. Compared with the results in Fig. 9, we see that the fraction near zero increases considerably. The reduction in skewness is common to all the schemes.

However, it should be noted that if an embedding scheme has large errors but the sum of errors is zero (not skewed) for all the nodes, LAT cannot improve the performance of the original embedding. For example, in (3), the $\tilde{e}_x$ can be zero even if there are errors. Then, there cannot be any improvement from the original embedding scheme.

Our results suggest that due to the existence of prevalent TIVs in the Internet host distance measurement data sets, instead of attempting to develop more sophisticated Euclidean (or metric-based) embedding method that minimizes a *global* error function (such methods are likely to be more expensive and time-consuming), it is far better to incorporate simpler and less expensive mechanisms to reduce the inevitable (local) estimation errors. Clearly, our proposed hybrid model with LATs is an example of such a simple yet effective mechanism.

VII. Conclusion

This paper investigated the suitability of embedding Internet hosts into a Euclidean space given their pairwise distances (as measured by round-trip time). Using the classical scaling and matrix perturbation theories, we established the (sum of the) magnitude of *negative* eigenvalues of the (doubly centered, squared) distance matrix as a measure of suitability of Euclidean embedding. Using data sets from real Internet host distance measurements, we illustrated that the distance matrix among Internet hosts contains negative eigenvalues of *large magnitude,* implying that embedding the Internet hosts in a Euclidean space would incur considerable errors. We attributed the existence of these large-magnitude negative eigenvalues to the prevalence of triangle inequality violations in the data sets. Furthermore, we demonstrated that the TIVs are likely to occur *locally;* hence the distances among these close-by hosts cannot be estimated very accurately using a *global* Euclidean embedding. In addition, increasing the dimension of embedding does not reduce the embedding errors.

[12]The Euclidean coordinates of the SLAT (2D+1) are the first two coordinates of the Virtual Landmark seven-dimensional embedding.

Based on these insights, we proposed and developed a simple hybrid model that incorporates a localized (non-Euclidean) adjustment term for each node on top of a low-dimensional Euclidean coordinate system. Our hybrid model preserves the advantages of the Euclidean coordinate systems while improving their efficacy and reducing their overheads (by using a small number of dimensions). Through both mathematical analysis and experimental evaluation, our hybrid model improves the performance of existing embedding methods while using only a low-dimension embedding. Lastly, our model can be incorporated into any embedding system (not necessarily Euclidean embedding).

APPENDIX

Proof of Theorem 2: First, verifying whether there exists a maximal TIV-free set of size k among a set of nodes with a distance matrix can be done in polynomial time by enumerating all set of size k and checking the TIVs in a nondeterministic machine. Hence this problem is NP.

Now we prove that the problem is NP-hard by reducing the MAX-CLIQUE problem (namely, finding the maximal clique in a graph G, a well-known NP-complete problem [15]) to the maximal TIV-free set problem. Let G be a connected undirected graph with $n > 2$ nodes. We assume that the size of maximal clique of G is $k > 2$. (The case $k = 2$ is trivial, as any pair of vertices with an edge is a maximal clique.) We construct a distance matrix $D = (d_{ij})$ among the set of vertices of G as follows, where d_{ij} will be the defined distance between vertices i and j. For each vertex i, we set $d_{ii} = 0$. For each edge e_{ij} between vertices i and j, we set $d_{ij} = 1$ and $d_{ji} = 1$. Note that for any triangle in G, the corresponding distances in D do not violate triangle inequality. For the pair of vertices i and j that do not have an edge between them in G, we first set d_{ij} :=undefined. Now, we define all the undefined d_{ij} as follows. For an undefined d_{ij}, we compute $c = \max_k(d_{ik} + d_{kj})$ for all k such that d_{ik} and d_{kj} are already defined. If no such c can be computed because d_{ik} and d_{kj} are undefined for all k, we set $c = 0$. Then, we set $d_{ij} := d_{ji} := c + 1$. This transformation takes polynomial time $O(n^3)$ since there are n^2 entries in D and, for each entry, $O(n)$ computation is required.

It can be easily shown that a triple of nodes (i, j, k) in G forms a triangle if and only if i, j, and k do not violate triangle inequality with d_{ij}, d_{ik}, and d_{jk} in D (we omit the detailed proof for the sake of space). This means that the maximal TIV-free set with the distances defined as D is the maximal clique in G. We conclude that finding a maximal TIV-free set problem is NP-hard. Since the maximal TIV-free set problem is NP and NP-hard, it is NP-complete. ∎

Proof of Lemma 1: Note that $t_c > t_a + t_b$. Let $(\hat{t_a}, \hat{t_b}, \hat{t_c})$ be a metric embedding of the three nodes. Then, they should satisfy the triangle inequality constraint as follows:

$$\hat{t_a} + \hat{t_b} \geq \hat{t_c}, \quad \hat{t_a} + \hat{t_c} \geq \hat{t_b}, \quad \hat{t_b} + \hat{t_c} \geq \hat{t_a}. \tag{4}$$

The squared estimation error e in this embedding is

$$e = (t_a - \hat{t_a})^2 + (t_b - \hat{t_b})^2 + (t_c - \hat{t_c})^2. \tag{5}$$

Let $k := |t_a - \hat{t_a}| + |t_b - \hat{t_b}| + |t_c - \hat{t_c}|$. We now show that $k \geq (t_c - t_a - t_b)$. Suppose $k < (t_c - t_a - t_b)$. There are eight cases based on the signs of $(t_a - \hat{t_a})$, $(t_b - \hat{t_b})$, and $(t_c - \hat{t_c})$. First, consider the case where $(t_a > \hat{t_a})$, $(t_b > \hat{t_b})$, and $(t_c > \hat{t_c})$. Then

$$\begin{aligned} k &- (t_c - t_a - t_b) \\ &= (t_a - \hat{t_a}) + (t_b - \hat{t_b}) + (t_c - \hat{t_c}) - (t_c - t_a - t_b) \\ &= 2(t_a + t_b) - \hat{t_a} - \hat{t_b} - \hat{t_c} \\ &\quad > 2(\hat{t_a} + \hat{t_b}) - \hat{t_a} - \hat{t_b} - \hat{t_c} \\ &\quad > \hat{t_a} + \hat{t_b} - \hat{t_c} \geq 0. \end{aligned}$$

So $k > (t_c - t_a - t_b)$, a contradiction. It can be easily shown that all the other seven cases contradict too. So we conclude that $k \geq (t_c - t_a - t_b)$.

Now, for any such $k \geq (t_c - t_a - t_b)$, consider another embedding $(\tilde{t_a}, \tilde{t_b}, \tilde{t_c})$ such that $\tilde{t_a} = t_a + k/3$, $\tilde{t_b} = t_b + k/3$, and $\tilde{t_c} = t_c - k/3$. Since it can be easily shown that $(\tilde{t_a}, \tilde{t_b}, \tilde{t_c})$ satisfies the triangle inequality constraint, it is a metric embedding.

Furthermore, for any such $k \geq (t_c - t_a - t_b)$, (5) is minimized with this embedding because

$$|t_a - \tilde{t_a}| = |t_b - \tilde{t_b}| = |t_c - \tilde{t_c}| = \frac{k}{3}. \tag{6}$$

Therefore, $e \geq (k/3)^2 + (k/3)^2 + (k/3)^2 = (k^2/3) \geq (t_c - t_a - t_b)^2/3$. ∎

Proof of Theorem 3: Let E be the sum of squared error of n nodes. $E = \sum_i (\hat{d_i} - d_i)^2$, where d_i is a distance between a pair of nodes (called i) and $\hat{d_i}$ is the embedded distance of the pair i. There are $n(n-1)/2$ pairs. Since there are $n(n-1)(n-2)/6$ triples among n nodes, E can be rewritten by the triples of nodes as follows: $E = (1/n-2) \sum_{t \in T}((\hat{t_a} - t_a)^2 + (\hat{t_b} - t_b)^2 + (\hat{t_c} - t_c)^2)$, where T is the set of triples; t_a, t_b, and t_c are the three distances of a triple t; and $\hat{t_a}$, $\hat{t_b}$, and $\hat{t_c}$ are the corresponding embedded distances. Clearly, $E \geq (1/n-2) \sum_{t \in V}((\hat{t_a} - t_a)^2 + (\hat{t_b} - t_b)^2 + (\hat{t_c} - t_c)^2)$, where $V \subset T$ is the set of TIV triples. From Lemma 1, we have $E \geq (1/3(n-2)) \sum_{t \in V}(t_c - t_a - t_b)^2$. ∎

Proof of Theorem 4: We just described a sketch of the proof. Let s_1 be the stress of using the pure Euclidean-based scheme. Let s_2 be the stress of using the pure Euclidean-based scheme with the adjustment term

$$s_1^2 = \frac{\sum_{x,y}\left(d_{xy} - d_{xy}^E\right)^2}{\sum_{xy} d_{xy}^2} \tag{7}$$

$$s_2^2 = \frac{\sum_{x,y}\left(d_{xy} - d_{xy}^E - e_x - e_y\right)^2}{\sum_{x,y} d_{xy}^2}. \tag{8}$$

Since the denominators are the same, we compute $(\sum_{x,y} d_{xy}^2)(s_1^2 - s_2^2)$ to compute $s_1^2 - s_2^2$

$$\begin{aligned} &\left(\sum_{x,y} d_{xy}^2\right)\left(s_1^2 - s_2^2\right) \\ &\qquad = \sum_{x,y}\left(d_{xy} - d_{xy}^E\right)^2 - \sum_{x,y}\left(d_{xy} - d_{xy}^E - e_x - e_y\right)^2. \end{aligned}$$

Using (2) and reformatting the formula, the final result can be easily obtained. ∎

References

[1] S. Lee, Z.-L. Zhang, S. Sahu, and D. Saha, "On suitability of Euclidean embedding of Internet hosts," in *Proc. ACM SIGMETRICS*, Saint Malo, France, Jun. 2006, pp. 157–168.

[2] T. E. Ng and H. Zhang, "Predicting Internet network distance with coordinates-based approaches," in *Proc. IEEE INFOCOM*, New York, Jun. 2002, pp. 170–179.

[3] L. Tang and M. Crovella, "Virtual landmarks for the Internet," in *Proc. IMC*, Miami, FL, Oct. 2003, pp. 143–152.

[4] H. Lim, J. C. Hou, and C.-H. Choi, "Constructing Internet coordinate system based on delay measurement," in *Proc. IMC*, Miami, FL, Oct. 2003, pp. 129–142.

[5] F. Dabek, R. Cox, F. Kaashoek, and R. Morris, "Vivaldi: A decentralized network coordinate system," in *Proc. ACM SIGCOMM*, Portland, OR, Aug. 2004, pp. 15–26.

[6] M. Costa, M. Castro, A. Rowstron, and P. Key, "Pic: Practical Internet coordinates for distance estimation," in *Proc. ICDCS*, Tokyo, Japan, Mar. 2004, pp. 178–187.

[7] H. Zheng, E. K. Lua, M. Pias, and T. G. Griffin, "Internet routing policies and round-trip-times," in *Proc. 6th Annu. Passive Active Meas. Workshop*, Boston, MA, Mar. 2005, pp. 236–250.

[8] E. K. Lua, T. Griffin, M. Pias, H. Zheng, and J. Crowcroft, "On the accuracy of embeddings for Internet coordinate systems," in *Proc. IMC*, Boston, MA, Apr. 2005.

[9] B. Wong, A. Slivkins, and E. G. Sirer, "Meridian: A lightweight network location service without virtual coordinates," in *Proc. ACM SIGCOMM*, Philadelphia, PA, Aug. 2005, pp. 85–96.

[10] I. Borg and P. Groenen, *Modern Multidimensional Scaling : Theory and Applications*. New York: Springer, 1997.

[11] G. H. Gulub and C. F. van Loan, *Matrix Computation*, 3rd ed. Baltimore, MD: Johns Hopkins Univ. Press, 1996.

[12] King2305 data set, [Online]. Available: http://www.cs.cornell.edu/People/egs/meridian/data.php

[13] A. Gupta, "Embedding tree metrics into low-dimensional Euclidean spaces," *Discrete Comput. Geom.*, vol. 24, no. 1, pp. 105–116, May 2000.

[14] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Adv. Neural Inf. Process. Syst.*, vol. 14, pp. 849–856, 2001.

[15] D. B. West, *Introduction to Graph Theory*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2001.

Sanghwan Lee received the B.S. and M.S. degrees in computer science from Seoul National University, Seoul, Korea, in 1993 and 1995, respectively, and the Ph.D. degree in computer science and engineering from the University of Minnesota, Minneapolis, in 2005.

He is currently an Assistant Professor at Kookmin University, Seoul, Korea. From June 2005 to February 2006, he was with the IBM T. J. Watson Research Center, Hawthorne, NY. His main research interest is theory and services in the Internet. He is currently applying the embedding schemes to a live peer-to-peer multimedia streaming system to see the real impact of the accuracy on the application perspective performance.

Zhi-Li Zhang (M'97) received the Ph.D. degree in computer science from the University of Massachusetts, Boston, in 1997.

In 1997, he joined the Faculty of the Computer Science and Engineering Department, University of Minnesota, Minneapolis, where he is currently a full Professor.

Dr. Zhang received the National Science Foundation CAREER Award in 1997. He has also received the McKnight Land-Grant Professorship and George Taylor Distinguished Research Award from the University of Minnesota and the Miller Visiting Professorship from the Miller Institute for Basic Sciences, University of California, Berkeley. He was a co-recipient of an ACM SIGMETRICS Best Paper Award and an IEEE International Conference on Network Protocols Best Paper Award.

Sambit Sahu (M'06) received the Ph.D. degree in computer science from the University of Massachusetts at Amherst in 2001.

He is a Research Staff Member with the Networking Software and Services Group, IBM T. J. Watson Research Center, Hawthorne, NY. His recent research has focused on various aspects of services management in a virtualized data center and design and analysis of network services. He has published a number of papers in the area of server and network management, differentiated services, multimedia, and overlay-based communication. He has received more than 40 patents in these areas.

Debanjan Saha (F'08) received the Ph.D. degree in computer science from the University of Maryland, College Park, in 1995.

He is an innovator and an entrepreneur with more than 15 years of experience in the IT industry. He is currently Director of Storage Software and Solution Enablement with IBM's Systems and Technology Group. In this role, he leads a global team spanning across multiple sites and is responsible for development execution of various storage products. Prior to his current role, he spent several years with IBM Research and Tellium Inc., an optical networking pioneer, which he helped grow from an early stage startup to a public company.

Dr. Saha is a Distinguished Scientist of the Association for Computing Machinery (ACM) and an IBM Master Inventor.