The Node Vector Distance Problem in Complex Networks

MICHELE COSCIA, IT University of Copenhagen, Denmark
ANDRES GOMEZ-LIEVANO, JAMES MCNERNEY, and FRANK NEFFKE, GrowthLab – Harvard University, United States

We describe a problem in complex networks we call the Node Vector Distance (NVD) problem, and survey algorithms currently able to address it. Complex networks are a useful tool to map a non-trivial set of relationships among connected entities, or nodes. An agent – e.g. a disease – can occupy multiple nodes at the same time, and can spread through the edges. The node vector distance problem is to estimate the distance traveled by the agent between two moments in time. This is closely related to the Optimal Transportation Problem (OTP) which has received attention in fields such as computer vision. OTP solutions can be used to solve the node vector distance problem, but they are not the only valid approaches. Here we examine four classes of solutions, showing their differences and similarities both on synthetic networks and real world network data. The NVD problem has a much wider applicability than computer vision, being related to problems in economics, epidemiology, viral marketing, and sociology, to cite a few. We show how solutions to the NVD problem have a wide range of applications, and provide a roadmap to general and computationally tractable solutions. We have implemented all methods presented in this paper in a publicly available open source library, which can be used for result replication.

CCS Concepts: • **Information systems** → **Social networks**.

Additional Key Words and Phrases: spreading events, social networks, network epidemics, structural change

ACM Reference Format:
Michele Coscia, Andres Gomez-Lievano, James McNerney, and Frank Neffke. 2020. The Node Vector Distance Problem in Complex Networks. *ACM Comput. Surv.* 1, 1, Article 1 (January 2020), 27 pages. https://doi.org/10.1145/3416509

1 INTRODUCTION

Complex networks are a mathematical tool that has been used in fields as different as physics [4], economics [44], epidemiology [75], marketing [50], and computer science [17]. They can be deployed to tackle a wide array of problems, from finding shortest paths [25] to community discovery [20], and from link prediction [58] to cascade failures [14]. A complex network is composed of a set of nodes connected by edges, frequently representing a constrained space on which a process unfolds, e.g. the movement of a traveler from one place to another. Often, the process involves the spread of an effect across the network. For example, nodes may represent people, who are either healthy or infected (Figure 1). In the toy example of Figure 1, if contact with an infected person is sufficient to transmit the disease, then at the next time step we will have four additional infected individuals.

Authors' addresses: Michele Coscia, mcos@itu.dk, IT University of Copenhagen, Rued Langaards Vej 7, Copenhagen, Denmark; Andres Gomez-Lievano, andres_gomez@hks.harvard.edu; James McNerney, james.mcnerney@gmail.com; Frank Neffke, frank_neffke@hks.harvard.edu, GrowthLab – Harvard University, 79 JFK St, Cambridge, MA, United States.

0360-0300/2020/1-ART1 $15.00
https://doi.org/10.1145/3416509

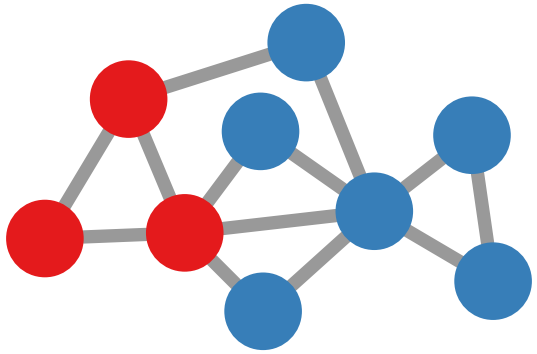


Fig. 1. A toy example of a network describing the spread of an infection. Blue nodes are healthy people, while red nodes are ones infected with the disease.

Instead of focusing on individual nodes, and how or when they get infected, one could also focus on the disease itself. This "agent" – the disease – morphs and moves as time goes by. An interesting question is how far the disease, as a whole, has shifted over time. That is, given observations of the network at two times, how far did the disease travel? When the agent is a single traveler, who can only occupy a single node at any time, many approaches are available. For example, one simple way to define a distance between two nodes is to take the shortest path between them, a well-defined and understood problem [23].

When both the origin and the destination are *sets* of nodes, the situation is different. Drawing a parallel with physics, one could view the movement of the agent on a network as analogous to the movement of a cloud of particles. Each particle moves in a different direction, and one could measure how far individual particles travel in a given period. However, one could also estimate how far the cloud as a whole has moved, for example, by measuring how much its center of mass has shifted.

For clouds of particles the solution is straightforward; in networks, it is much less clear how one should define such a distance. The problem has received little attention, and even the comprehensive encyclopedia of distances [24] fails to list any distance measure between node vectors in a graph, even though this problem arises often. The example above illustrates the case of epidemics [18, 35, 75], while other cases include structural change in economics, where countries/firms diversify their export or production baskets across a network of technologically related products [11, 44, 45, 70, 71, 92]; sensor networks, where a signal is recorded by multiple nodes and then propagates [43, 85]; and viral marketing [28, 50, 54].

Existing efforts often solve the issue with ad-hoc strategies, and without explicitly stating the distance problem involved. For instance, Ref. [45] examines how a country's export basket shifts across a network of related products. In the absence of formal measures, the authors rely on visual inspection of the network to observe changes in the basket.

To the best of our knowledge, the most explicit consideration of the problem has taken place in work on the Optimal Transportation Problem (OTP), also called the earth mover's distance in the computer vision literature [87]. This problem is to find the most efficient way to transport a set of node weights from an initial distribution to a final distribution. The optimal cost – in terms of number of edges to be crossed – is the distance between the two node weight distributions [82, 93]. The optimal transportation problem is a special assignment problem [16], and has been mostly studied in the context of computer vision [78, 80]. Points of interest in a picture are nodes in a planar graph (image plane or pixel grid), and the intensity of light on each node is compared to

decide if the two graphs describe the same image or not. Usually a variation of the Wasserstein metric [38] is used that assumes not a metric space but one described by the graph [93]. This seems to be the standard approach in configural perception models [83], for instance in Elastic Bunch Graph Matching [100].

The Optimal Transportation Problem carries several underlying assumptions; it uses global information, with a solution characterized by an omniscient agent who knows the full network, and defines the distance to be the outcome of an optimization. Given these constraints, it is clear that the OTP is but one of several classes of solutions to the general distance problem we formulate here. To go beyond such limitations, we describe the more general Node Vector Distance (NVD) problem. As we show, the NVD problem can be solved using different approaches, which we group into four classes: generalized Euclidean, shortest paths, spectral approaches, and adaptations of NVD-related algorithms. Metrics in the generalized Euclidean class take the form of a generalized Euclidean distance, accounting in various ways for the constraint of moving through the edges of the network. Metrics in the shortest paths class build off shortest path solutions (including but not limited to OTP approaches), where the distance between two node sets is an aggregation of shortest paths between the nodes that compose them. The spectral class contains approaches that exploit the relationship between the spectral representation of a graph and the diffusion processes happening on it. Finally, we describe a collection of solutions that cannot be classified under any of the other approaches.

Different classes of solutions make different assumptions about how the agent relates to the network. As noted earlier, in OTP the agent has global and perfect information about the network, and tries to minimize its effort. In other applications, it may be more appropriate for the agent to move myopically or randomly across edges. We will see that methods using such assumptions can assess distances across the network differently.

Here our main goal is to define the NVD problem and describe the broad approaches to solving it, though we also begin the work of probing the various solutions with three kinds of tests. In the first, we use synthetic networks to show strengths and weaknesses in controlled environments, where we have prior expectations about how a well-behaved measure should respond. In the second we compare distance measures directly, asking which ones return similar results. Finally, we apply our distance measures to real world data, asking whether they can uncover useful information about the propagation properties of agents.

We have implemented all methods presented in this paper in a publicly available open source library, which can be used for result replication[1].

2 PROBLEM STATEMENT

We start by defining NVD formally. We then discuss properties we expect of solutions, and some possible applications.

2.1 Definition

Let $G = (V, E)$ be a network with a set of nodes V and a set of edges E. For simplicity, we consider only undirected, unweighted graphs here. We assume that the network G is unchanging over time. In the node vector distance problem, we are interested in the change of some collective object that occupies the graph, such as a rumor in a network of gossiping individuals or a firm in a network of related products. At any given time this object, or *agent*, occupies a portion of a network. Formally, we define an agent as a vector A of length $|V|$ that assigns a weight $A(v) \in \mathbb{R}^+$ to each node v in the network. These weights reflect the fact that an agent may occupy nodes with different intensities,

[1] http://www.michelecoscia.com/?page_id=1733

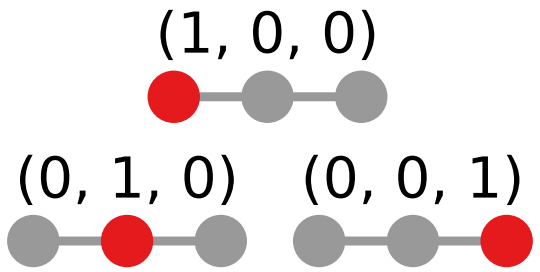


Fig. 2. Three agent vectors A on a simple graph. Red nodes are occupied by the agent.

e.g. a firm may engage more intensively in some products than in others. When node v has a non-zero weight $A(v) > 0$ we say that A occupies node v. For simplicity, we assume A is normalized so that $\sum_{v \in V} A(v) = 1$.

An agent may move across a network or change how intensively it occupies different nodes. Let A_t denote the agent at time t. The node vector distance problem is to measure the distance $\delta_{A_{t_i},A_{t_j},G}$ between the agent A at two times t_i and t_j as it evolves across the unchanging network G. Intuitively, the details of G will significantly affect the distance that is measured. For instance, in Euclidean space it is clear that the vectors $A_{t_1} = (1, 0, 0)$, $A_{t_2} = (0, 1, 0)$, and $A_{t_3} = (0, 0, 1)$ are equidistant – they are all a distance $\sqrt{2}$ apart – but in the simple network of Figure 2, A_{t_1} is expected to be closer to A_{t_2} than to A_{t_3}.

While we consider unweighted networks here, many of the distance measures we consider can be adapted to weighted networks with few modifications. We note that care should be taken to interpret weights appropriately, since in some cases a large edge weight between two nodes indicates a strong connection – the nodes could be considered close together – while in others it indicates a high cost of edge traversal – the nodes are far apart.

2.2 Properties of Solutions

Strictly, a function δ must have four properties to be a distance metric [24]:

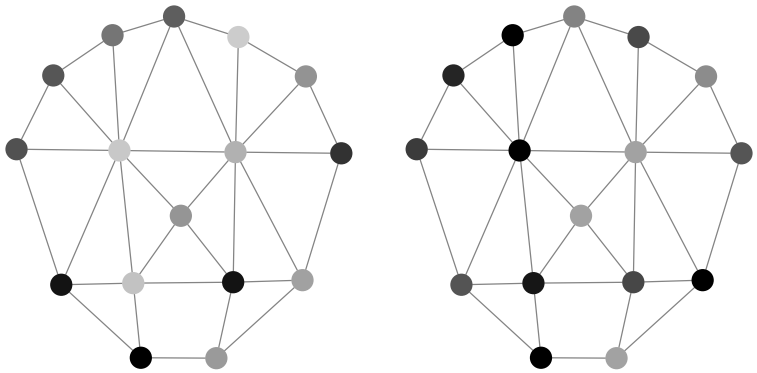
(1) Non-negativity: $\delta_{A_{t_i},A_{t_j},G} \geq 0$.
(2) Identity of indiscernibles: $\delta_{A_{t_i},A_{t_j},G} = 0$ if and only if $A_{t_i} = A_{t_j}$
(3) Symmetry: $\delta_{A_{t_i},A_{t_j},G} = \delta_{A_{t_j},A_{t_i},G}$.
(4) Triangle inequality: $\delta_{A_{t_i},A_{t_j},G} + \delta_{A_{t_j},A_{t_k},G} \geq \delta_{A_{t_i},A_{t_k},G}$.

It is reasonable to think that δ need not have all four properties to usefully measure change in the agent A. For instance, in statistics and machine learning, the cosine distance, Pearson correlation distance, and Kullback-Liebler divergence all fail to satisfy one or more of these properties. Here we use 'distance' loosely to mean a function δ that satisfies one or more of the above properties.

Our focus is on undirected networks, though we note that in some directed networks it may be appropriate to modify the symmetry condition to read $\delta_{A_{t_i},A_{t_j},G} = \delta_{A_{t_j},A_{t_i},G'}$, where G' is the graph obtained from transposing the adjacency matrix of G. In directed networks edge directions may make a change from A_{t_i} to A_{t_j} harder or easier than the reverse change.

2.3 Application Scenarios

In this section we briefly discuss applications where the NVD problem arises, and why we may want to look beyond OTP for solutions.



(a) Face #1 (b) Face #2

Fig. 3. An example of face data as represented in elastic bunch graph matching [100]. Each node is a point of interest (right eye, nose, chin, etc.), which is shaded according to the amount of light at each point.

2.3.1 Computer Vision. We start with computer vision [78, 80], a classic application for the OTP. An image is represented as a graph of points of interest, which have different loadings proportional to how much light or color is in them (Figure 3). Two images are compared by computing how much light must be 'transported' from interest points of one to interest points of the other using a measure known as the earth mover's distance. Small amounts indicate the images are similar.

In this application one can see why solutions depend on solving the OTP. A good distance between images should consider the whole image, or equivalently the whole graph of interest points. At the same time one should try to edit the images as little as possible, i.e. we should not need to move light far across the graph to unrelated points of interest if the images are similar.

2.3.2 Epidemics. Another potential application is measuring the speed of epidemics, where nodes are people and the agent is a disease that occupies the nodes of infected individuals. The spread of a disease depends less on the movement or infection rate of individuals than on the movement of the disease as a whole. Figure 4 for example shows a simulation of the spread of a hypothetical disease via the international traveler network using an SI model (see e.g. [18, 35, 75].)

2.3.3 Viral Marketing. A related problem is viral marketing, which is often modeled similarly to a biological epidemic [28, 50, 54]. To learn how effective a marketing campaign was one would like to know if news of a new product has travelled far. Success may mean not just reaching more people per se, but also reaching people in disparate parts of the network.

A related application is the use of distances between groups of individuals to infer similarity in the products they use. Suppose two products each have established user bases among different groups of individuals in the network. One could use the users of one product as the origin A_{t_1} and users of the other product as the destination A_{t_2}. The closer the users are the more similar we expect the products to be.

2.3.4 Economics. Finally, we give an example of an NVD problem that arises in economics, namely in the study of country movements on the Product Space [44, 45]. In the Product Space nodes are products, and two products are connected by an edge if a significant number of countries co-export both products. Often, such connections are interpreted in terms of capabilities needed to produce

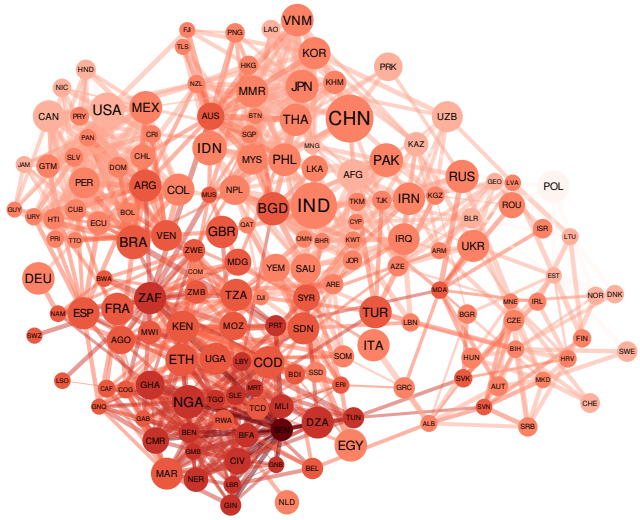


Fig. 4. Simulated spread of a disease originating in Senegal (dark node near the bottom of the graph). Nodes are countries, which are connected by an edge wherever a significant number of travelers move between them. Darker nodes are countries infected early on, while lighter nodes are countries infected later.

different goods. For example, if cars and motorcycles are frequently co-exported, it suggests that similar knowhow is used in the production of both goods.

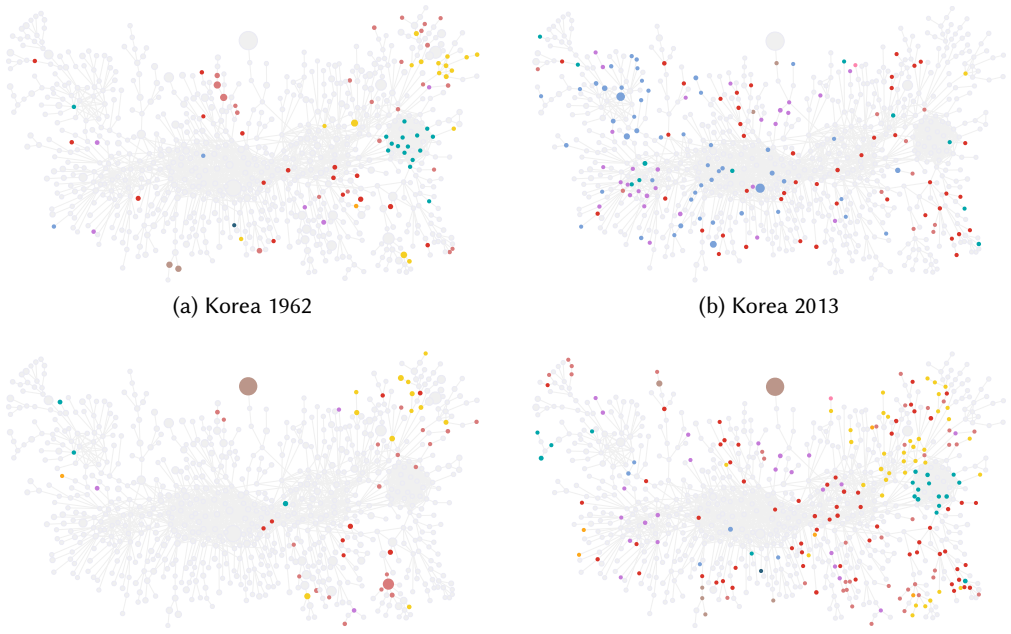
In this case an agent is a country, which occupies a set of products on the network corresponding to its export basket. Over time a country may change its export basket, shifting its economy into new industries that make different products. An interesting (and so far unaddressed) question is how much or how quickly a given country has transformed its economy over time.

Simply counting the products that changed in the export basket may not adequately quantify this transformation because transitions into some products are more difficult than others [44, 45]. Instead, a distance measure must assign smaller distances to countries that move mainly among products in the same connected cluster. For example, Figure 5 shows the change in the positions of Korea and Egypt on the Product Space network between 1962 and 2013. In this period Korea and Egypt added roughly the same number of products, though Korea would generally be regarded as having undergone a larger transformation. In the network, this is reflected in the fact that Korea made a large shift, from garment manufacturing and agriculture (right side in Figure 5) to electronics, machinery, and chemicals (left side). While Egypt also spread to electronics, machinery, and chemical products, more than half of its exports remain in garment manufacturing and agriculture.

3 CLASSES OF SOLUTIONS

Here we develop a taxonomy of solution approaches for the NVD problem. We group the solutions into four classes:

(1) Generalized Euclidean (Section 3.1);
(2) Shortest path-based (Section 3.2);
(3) Spectral (Section 3.3);
(4) Adaptations of NVD-related algorithms (Section 3.4).



(a) Korea 1962 (b) Korea 2013

(c) Egypt 1962 (d) Egypt 2013

Fig. 5. Diversification of exports by Korea and Egypt over the Product Space network between 1962 and 2013. Products not exported in significant quantities are grayed out. Visualizations from http://atlas.cid.harvard.edu/.

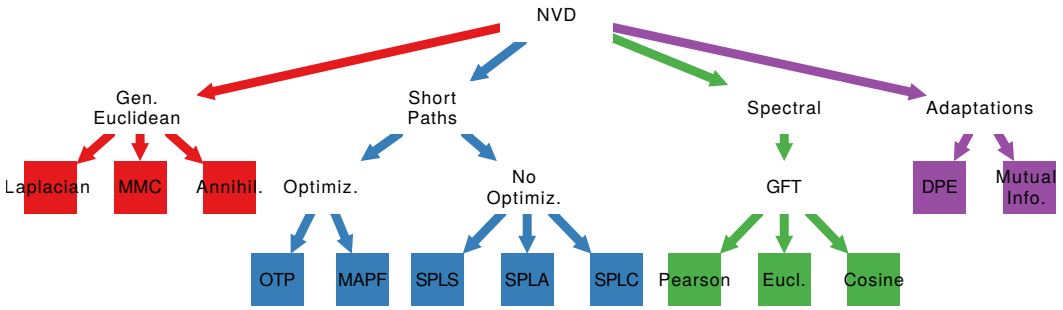


Fig. 6. Our organization of the NVD literature. Each color represents a major branch. Acronyms defined in Table 1.

In some cases, the approaches we suggest do not appear to have been analyzed in the literature yet. Figure 6 elaborates on our organization of the solution categories, and Table 1 gives acronyms for each method.

3.1 Generalized Euclidean

Solutions in the generalized Euclidean class involve metrics of the form

$$\delta_{A_{t_1},A_{t_2}} = \sqrt{(A_{t_1} - A_{t_2})^T Q (A_{t_1} - A_{t_2})},$$

Acronym	Method	Class
Lapl	Laplacian	Generalized Euclidean
MMC	Mean Markov Chain	Generalized Euclidean
Annihil	Annihilation	Generalized Euclidean
OTP	Optimal Transportation Problem	Shortest Paths
MAPF	Multi Agent Path Finding	Shortest Paths
SPLS	Shortest Path Length (Single linkage)	Shortest Paths
SPLA	Shortest Path Length (Average linkage)	Shortest Paths
SPLC	Shortest Path Length (Complete linkage)	Shortest Paths
GFT	Graph Fourier Transform	Spectral
DPE	Discrete Pursue Evasion	Adaptations

Table 1. The acronyms we use throughout the paper to shorten distance names.

where Q is a positive semi-definite matrix that depends on the structure of the network. For metrics in this class, distances on the network have the same form as distances in a high-dimensional Euclidean space. The property of positive semi-definiteness guarantees that $x^T Qx$ (and thus distance δ) will be a non-negative number. We discuss three metrics in this class, each using a different way to construct Q from the graph's adjacency matrix.

3.1.1 Graph Laplacian. One candidate for Q [19] is the Moore-Penrose pseudoinverse L^+ of the graph Laplacian $L = D - E$, where D is a diagonal matrix with node degrees on the diagonal and E is the adjacency matrix. To see that $Q = L^+$ is positive semi-definite, note that L's singular value decomposition is $L = Q_1 \Sigma Q_2^T$, where Σ is a rectangular diagonal matrix containing L's singular values, while its pseudoinverse is $L^+ = Q_2 \Sigma^+ Q_1^T$, where Σ^+ is a rectangular diagonal matrix whose elements are the reciprocals of those of Σ. Since the eigenvalues (and therefore singular values) of the graph Laplacian L are non-negative, those of L^+ are as well, and thus L^+ is positive semi-definite. Further motivation for this choice of Q can be found in [19], based on the connection of the graph Laplacian to network diffusion processes.

3.1.2 Mean Markov Chain. We can show another way of obtaining Q by thinking of the network as a Markov chain to motivate a z-score-like metric. We view node v's intensity $A_{t_1}(v)$ as the number of random walkers at node v at time t_1, and construct a stochastic matrix P whose elements give transition probabilities between nodes. A simple choice is $P = D^{-1}E$, the matrix obtained from normalizing columns of the adjacency matrix E by each node's degree. Given a starting node vector A_{t_1}, the expected number of random walkers arriving at node v in the next time step is

$$E[A_{t_2}(v)] = \sum_{d \in V} P_{v,d} A_{t_1}(d),$$

and the variance is

$$\sigma_v(t_1) = \sum_{d \in V} A_{t_1}(d) P_{v,d}(1 - P_{v,d}).$$

A z-score-like distance from A_{t_1} to A_{t_2} can be motivated by considering the deviation of A_{t_2} from the expected value, $E[A_{t_2}] = PA_{t_1}$, normalized by the standard deviation of arrivals at each node,

$$\delta = \sqrt{\sum_{v \in V} \left(\frac{A_{t_2}(v) - E[A_{t_2}(v)]}{\sigma_v(t_1)} \right)^2},$$

or in matrix form $\delta = \sqrt{(A_{t_2} - PA_{t_1})Z^{-1}(A_{t_2} - PA_{t_1})}$ where $[Z]_{v,v} = \sigma_v(t_1)^2$. However, in this formulation $Q = Z^{-1}$ is not symmetric with respect to times t_1 and t_2, and the presence of P above causes the metric to fall outside the generalized Euclidean class. To address these issues we break the motivational form above by eliminating P, and make Q symmetric by averaging the standard deviations from taking A_{t_1} and A_{t_2} each as the starting vectors: $[Z]_{d,d} = \left[\sigma_v(t_1)^2 + \sigma_v(t_2)^2\right]/2$.

3.1.3 Annihilation. Like the MMC, the Annihilation measure draws on the intuition of diffusion. In this case, we treat the random walkers as being so numerous that the system is always characterized by its mean behavior, so that after one step $A_{t+1} = E[A_{t+1}] = PA_t$ and after k steps $A_{t+k} = P^k A_t$. In addition, we view both the initial vector A_{t_1} and the final vector A_{t_2} as diffusing, the former with positive weights and the latter with negative weights. After k steps then, the vector of net node vector occupation weights is $P^k(A_{t_2} - A_{t_1})$. One can think of this setup in analogy with the annihilation of positive and negative charges diffusing across the network. If A_{t_1} represents the positive charges at each node and A_{t_2} represents the negative charges, then $P^k(A_{t_2} - A_{t_1})$ is the vector of net charges at each node after k steps.

If the Markov chain is regular, then $P^k A_{t_2}$ and $P^k A_{t_1}$ will each converge to a stationary distribution $\tilde{A}$ as $k \to \infty$. As a result $P^k(A_{t_2} - A_{t_1})$ will converge to a vector of zeros, corresponding in the physical analogy to the eventual annihilation of the positive and negative charges. To motivate a metric, we characterize how long this convergence takes by considering the summation

$$S = \sum_{k=0}^{\infty} P^k(A_{t_1} - A_{t_2}).$$

To the extent that A_{t_2} and A_{t_1} differ, or occupy weakly connected parts of the network, convergence of the summation terms towards the zeros vector will take longer, resulting in a vector S with elements (positive and negative) that are larger in magnitude. We can straightforwardly convert the vector S into a distance by computing $\delta = \sqrt{S^T S}$.

While the summation over $P^k(A_{t_1} - A_{t_2})$ converges (since these terms shrink to the zeros vector), a similar summation over P^k by itself will not because $\lim_{k\to\infty} P^k$ is the positive matrix $P^\infty = [\tilde{A}, \tilde{A}, \ldots]$, i.e. the matrix whose columns each equal the stationary distribution $\tilde{A}$. However, one can show that S can be written in the convenient form $S = [I - (P - P^\infty)]^{-1}(A_{t_1} - A_{t_2})$. The advantage of writing S this way is that $(A_{t_1} - A_{t_2})$ is now multiplied by a well-defined matrix, $F = [I - (P - P^\infty)]^{-1}$, which is known as the fundamental matrix in the theory of regular Markov chains. The metric can then be written $\delta = \sqrt{S^T S} = \sqrt{(A_{t_1} - A_{t_2})^T F^T F (A_{t_1} - A_{t_2})}$, with $Q = F^T F$, which falls again in the general Euclidean class.

3.2 Shortest Paths

Solutions based on shortest paths start by finding shortest distances between pairs of nodes, a well understood problem. Using a shortest path algorithm – for instance Dijkstra's [25] – one can first count edges separating node u to v. Then by aggregating over all shortest paths between origin nodes in A_{t_1} and destination nodes in A_{t_2} one can define several distance functions δ.

In this section we use $L_{A_{t_1}, A_{t_2}}$ to refer to the set of all possible path lengths between origins and destinations. A shortest path length $l_{u,v} \in L$ is a path length with the least edge crossing to move from node u to node v. We discuss two subcategories of this class: non-optimized methods that simply aggregate over shortest paths, and optimized methods that seek the best combination of paths from A_{t_1} to A_{t_2}.

3.2.1 Non-Optimized. Here we take hierarchical clustering as an inspiration for the aggregating function δ. In hierarchical clustering there are three common ways to compute distances between

clusters [91]: single, complete, and average linkage. In single linkage, the distance between clusters is the distance between their closest points, while in complete linkage the distance between clusters is the distance between their farthest points. In average linkage, the cluster distance is an average over the distances between all pairs of points in the two clusters. In the NVD problem we can use the same logic. For example, under the single linkage strategy, we compute the distance between A_{t_1} and A_{t_2} as the distance between each node in A_{t_1} and the closest node in A_{t_2}.

Algorithm 1: Compute $\delta_{A_{t_1},A_{t_2},G}$ for single-linkage, non-optimized shortest path distance.
Input: a graph G, two vectors of reals of length $|V|$ A_{t_1} and A_{t_2}.

1 $\delta_{A_{t_1},A_{t_2},G}$:
2 $\quad \delta \leftarrow 0$;
3 $\quad A_{t_1} \leftarrow A_{t_1} / \sum A_{t_1}$;
4 $\quad A_{t_2} \leftarrow A_{t_2} / \sum A_{t_2}$;
5 $\quad L \leftarrow SPL(G, A_{t_1}, A_{t_2})$;
6 $\quad$ **while** $\sum A_{t_1} > 0$ **do**
7 $\quad\quad \mathcal{P} \leftarrow \{(u,v)$ s.t. $l_{u,v} = \min L$ and $A_{t_1}(u) > 0, A_{t_2}(v) > 0\}$;
8 $\quad\quad u, v \leftarrow \underset{\min(A_{t_1}(u), A_{t_2}(v))}{\arg\max} A_{t_1}, A_{t_2}, \ \forall u, v \in \mathcal{P}$;
9 $\quad\quad w \leftarrow \min(A_{t_1}(u), A_{t_2}(v))$;
10 $\quad\quad \delta \leftarrow \delta + (w \times l_{u,v})$;
11 $\quad\quad A_{t_1}(u) \leftarrow A_{t_1}(u) - w$;
12 $\quad\quad A_{t_2}(v) \leftarrow A_{t_2}(v) - w$;
13 $\quad$ **end**
14 $\quad$ **return** δ
15 **end**

Algorithm 1 shows the calculation of δ under the single linkage strategy. First we normalize the node vectors (lines 3-4). Next we calculate the set of shortest path lengths between nodes in A_{t_1} and A_{t_2} (line 5). We then systematically shift weights one-at-a-time from one node vector to the other via shortest paths (lines 6-13). To prepare to make a shift, we identify all pairs of nodes (u, v) in the two node vectors that are separated by the least shortest path (line 7). There may be ties such that multiple pairs have the least shortest path, so among such pairs $\mathcal{P}$ we find the pair that can exchange the most weight (line 8). The weight w that can be shifted from one node to the other is the lesser of $A_{t_1}(u)$ and $A_{t_2}(v)$ (line 9). This weight, multiplied by the number of edges $l_{u,v}$ to be crossed, is the contribution of this shift to a running total distance δ (line 10). Finally, we deduct the shifted weight w from both $A_{t_1}(u)$ and $A_{t_2}(v)$ (lines 11-12) to ensure this weight is not shifted again and the loop eventually ends.

Potentially, one could modify the algorithm to construct a distance that accounts for changes in the scale of node vectors as well as shifts in position. We could modify lines 3-4 to rescale the node vector with the smaller sum to have the larger of the two sums. Then δ will accumulate not only shifts in the weights from A_{t_1} to A_{t_2}, but also the difference in their total weight.

In single linkage if a node v is part of both A_{t_1} and A_{t_2} then its contribution to δ is zero because $l_{u,v} = 0$. If $A_{t_2} = A_{t_1}$, then $\delta_{A_{t_1},A_{t_2},G} = 0$.

In complete linkage we use the same algorithm but replace arg min with arg max in line 6. In this case δ is zero only when both A_{t_1} and A_{t_2} occupy the same single node. Note that if a node v in A_{t_1} is also part of A_{t_2}, it will only pick itself as a destination if it has some residual weight after

first distributing its weight to every other node in A_{t_2}. This shows that complete linkage does not respect the identity of indiscernibles as single linkage does.

Finally, in average linkage δ is defined as the weighted average of the shortest paths between all $u \in A_{t_1}$ and all $v \in A_{t_2}$:

$$\delta_{A_{t_1},A_{t_2},G} = \frac{\sum_{\forall v \in A_{t_2}} \sum_{\forall u \in A_{t_1}} A_{t_1}(u) A_{t_2}(v) l_{u,v}}{\sum A_{t_1}}.$$

As in the single linkage algorithm we rescale A_{t_1} and A_{t_2} to have the same sum. If we also normalize the vectors, $\sum A_{t_1} = \sum A_{t_2} = 1$, only the numerator matters.

As we will see this measure is similar to OTP: A_{t_1} and A_{t_2} provide the weights, and $l_{u,v}$ the distance. The difference is that, in OTP, the flow is linearly optimized beforehand. In average linkage we move equal fractions of A_{t_1}'s total weight to all destinations in A_{t_2}, while OTP tries to find the "best" way to go from A_{t_1} to A_{t_2}.

3.2.2 Optimized. Methods to find the optimized combination of shortest paths to minimize the distance crossed from A_{t_1} to A_{t_2} are well studied. There are two approaches: OTP, and Multi-Agent Path Finding (MAPF).

Optimal Transportation Problem. In its original formulation [67], OTP focuses on the distance between two probability distributions without an underlying network. However, it has been observed that this problem can also be applied to transportation over an infrastructure, known as a multi-commodity network flow [46]. To adapt to the network case, one must simply specify how distant two dimensions in the node vector are using a formal distance metric. The number of edges in the shortest path between two nodes satisfies this requirement.

Such formulation is known in computer vision as the Earth Mover Distance. In mathematics, it is known as the Wasserstein distance: a distance function defined between probability distributions – our node vectors – on a given metric space, which in this case is specified by the graph G.

In OTP we want to estimate the minimal edge crossings needed to transform the origin distribution into the destination one. This is a high-complexity problem, which has lead to an extensive search for efficient approximations [8, 29, 31, 49, 57, 62, 64, 76, 77, 87]. For our purposes these methods are equivalent, solving the same underlying problem using different approaches to perform the expensive optimization step.

More formally, in OTP we want to find a set of movements M such that:

$$M = \arg\min_{m_{u,v}} \sum_u \sum_v m_{u,v} d_{u,v},$$

where $m_{u,v}$ is the weight to be transferred from node u to node v and $d_{u,v}$ is the distance between them. Then:

$$\delta_{A_{t_1},A_{t_2},G} = \frac{\sum_u \sum_v m^*_{u,v} d_{u,v}}{\sum_u \sum_v m^*_{u,v}},$$

where $m^*_{u,v} \in M$ are the optimal movements. In this paper, we take the distance $d_{u,v}$ to be the shortest path length between u and v, $d_{u,v} = l_{u,v}$.

Multi-Agent Path Finding. Another problem related to NVD is multi-agent path finding (MAPF) on a graph [40, 103]. In this problem, multiple "robots" occupy one node at a time and each robot has an intended destination [32]. The goal is to find the set of moves that allow robots to reach their destinations in the most efficient way possible, or to discover that no solution exists [104].

There are usually constraints, such as edge capacities that limit each edge to be used by one robot at a time [84].

There are a few considerations one must address to use MAPF as a solution to the NVD problem. First, in MAPF one must specify an origin and a destination for each robot in the graph, deciding in advance which nodes $u \in A_{t_1}$ should go to which $v \in A_{t_2}$. This is fundamentally different from NVD, where each weight in A_{t_1} can potentially reach any other destination in A_{t_2}. Here we use the same strategy as in the non-optimized shortest path approach with single linkage: we look for the shortest path length $l_{u,v}$ carrying the largest possible weight $\min(A_{t_1}(u), A_{t_2}(v))$. This strategy is naive and likely results in a suboptimal allocation.

Second, in MAPF robots cannot be on the same node at the same time. Say that we assign a robot to go from u to v in our preprocessing. If $A_{t_1}(u) > A_{t_2}(v)$, then u will have some unallocated weight. Thus we would need to add at least a second robot that can start at u and terminate at some other v'. But this violates MAPF. We solve the issue by running a sequence of MAPF sessions. In the second session, we attempt to move weights leftover after the first session. We keep running smaller and smaller sessions until all weights have been allocated, which we guarantee by normalizing vectors to have the same sum.

Algorithm 2: $\delta_{A_{t_1},A_{t_2},G}$ for multi-agent path finding distance. Input: a graph G, two vectors of reals of length $|V|$ A_{t_1} and A_{t_2}.

1 $\delta_{A_{t_1},A_{t_2},G}$:
2 $\quad \delta \leftarrow 0$;
3 $\quad A_{t_1} \leftarrow A_{t_1} / \sum A_{t_1}$;
4 $\quad A_{t_2} \leftarrow A_{t_2} / \sum A_{t_2}$;
5 $\quad L \leftarrow SPL(G, A_{t_1}, A_{t_2})$;
6 $\quad$ **while** $\sum A_{t_1} > 0$ **do**
7 $\quad\quad R \leftarrow \emptyset$;
8 $\quad\quad L' \leftarrow L$;
9 $\quad\quad$ **while** $L' \neq \emptyset$ **do**
10 $\quad\quad\quad L'' \leftarrow \{(u,v)$ s.t. $l_{u,v} = \min L'\}$;
11 $\quad\quad\quad u, v \leftarrow \underset{\min(A_{t_1}(u), A_{t_2}(v))}{\arg\max} A_{t_1}, A_{t_2}, \ \forall u, v \in L''$;
12 $\quad\quad\quad w \leftarrow \min(A_{t_1}(u), A_{t_2}(v))$;
13 $\quad\quad\quad R \leftarrow R \cup robot(u, v, w)$;
14 $\quad\quad\quad A_{t_1}(u) \leftarrow A_{t_1}(u) - w$;
15 $\quad\quad\quad A_{t_2}(v) \leftarrow A_{t_2}(v) - w$;
16 $\quad\quad\quad L' \leftarrow L' - \{u', v'$ s.t. $u' = u$ or $v' = v\}$
17 $\quad\quad$ **end**
18 $\quad\quad \delta \leftarrow \delta + MAPF(G, R)$;
19 $\quad$ **end**
20 $\quad$ **return** δ
21 **end**

Algorithm 2 shows the details of this strategy. Unlike Algorithm 1, we now remove *all* paths in L' that either start in u or terminate in v (line 15), regardless of the $A_{t_1}(u)$ and $A_{t_2}(v)$ values, rather than only those attached to the node remaining with zero weight (Algorithm 1, line 6). That is because, when we assign a robot to an origin-destination pair, no other robot can have either

endpoint. The function $robot(u, v, w)$ (line 12) creates a robot at node u carrying to node v a weight w. The set R stores all robots created in a session (line 12), and *MAPF* (line 17) is any algorithm solving the MAPF problem given a graph G and a set of robots R. When a robot of weight w reaches v from u over a path of length $l_{u,v}$ it adds a contribution $l_{u,v} \times w$ to distance δ. Note that in this approach $l_{u,v}$ need not be the shortest distance between u and v due to the capacity constraints of edges.

There are many algorithms to solve MAPF [6, 26, 39, 56, 59, 61, 86, 95–97, 101], each providing a different solution to NVD with our preprocessing strategy. In this paper we focus on only one of them [90]. It may also be possible to adapt other MAPF algorithms to solve NVD, without our suboptimal preprocessing strategy[2].

A MAPF-based strategy to solve the NVD problem might be most appropriate when the network represents a system such a road network, where there is a limit on the number of vehicles that can use a road at a given time. We note that, in principle, MAPF should be equivalent to OTP when there exists a solution with no collisions and allocates all weights in a single session. MAPF can be seen as an approach that takes into account congestion, assuming a fixed capacity of one robot per edge. A variation on this idea is to assume each edge has a capacity based on weight, so that multiple robots with small weights could simultaneously pass over an edge.

3.3 Spectral

Another class of solutions to the node vector distance problem exploits the spectrum of the graph. One motivation for these approaches comes from signal processing, where a common problem is to extract the true signal $\hat{s}$ from the noisy and correlated signal data s of a battery of sensors. The relationships between sensor outputs are taken into account and modeled with a network G that connects related sensors. The outputs from these sensors are smoothed using the Graph Fourier Transform [43, 85].

To compute the Graph Fourier Transform we first compute the graph Laplacian, $L = D - E$, and compute its eigenvectors $l_0, l_1, l_2, \ldots$, whose eigenvalues $\lambda \in \mathbb{R}$ satisfy $0 = \lambda_0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_{|V|-1}$. (As usual we assume G is connected.) We then define Φ as the matrix whose columns are these eigenvectors, arranged in increasing order of their eigenvalues: $\Phi = (l_0, l_1, \ldots, l_n)$. The Graph Fourier Transform Φ of a signal A_{t_1} is then $\widehat{A_{t_1}} = \Phi^T A_{t_1}$. Transforming A_{t_1} by Φ converts it from a spatial representation (where the elements of A_{t_1} correspond to nodes) to an eigenmode representation.

To arrive at a distance metric, we can now weight the modes taking into account the topology of the graph. This is usually achieved by filtering the signal in the spectral domain, multiplying it with the diagonal matrix Λ containing the Laplacian's eigenvalues on the diagonal. Applying this transformation to A_{t_1} and A_{t_2} encodes G's topology in the vectors, and then the Euclidean distance between them is the node vector distance that we are looking for:

$$\delta_{A_{t_1}, A_{t_2}, G} = Euclidean(A_{t_1}\Lambda\Phi^T, A_{t_2}\Lambda\Phi^T).$$

Note that this is not one, but a family of measures. Having filtered the node vectors A_{t_1} and A_{t_2}, one could replace the Euclidean distance with another off-the-shelf measure to estimate the distance between $\widehat{A_{t_1}}$ and $\widehat{A_{t_2}}$ because they already contain G's topology in their values. These approaches can establish the distance between two different signals on a graph, though signal processing is just one possible application. Others include signal cleaning [41], frequency analysis [81], sampling

[2]Any of these algorithms allowing robots to share origin/destinations, or to reach any valid destination from their origin, may not need the preprocessing or may need only part of it.

[7], interpolation [69], and trend filtering [98]. We note the optimal transformation could differ from the one here depending on the application.

3.4 Adaptations

In this section we discuss two approaches that were not originally developed to solve the NVD problem, but can solve it using minor adaptations. These approaches could be considered related to other categories already discussed, but we group them together because of similarities in how we adapt them to NVD.

3.4.1 Discrete Pursuit-Evasion. In pursuit-evasion, we populate a space with a set of pursuer and evader robots, where the pursuers' aim is to capture the evaders. In discrete pursuit-evasion (DPE) robots occupy a graph rather than a Euclidean space [74]. Many algorithms have been proposed to model different strategies and constraints on both the pursuer and on the evader side; see [3, 5, 33, 60, 65, 89] for some recent examples.

To adapt DPE to solve NVD we set the pursuers as A_{t_1} and the evaders as A_{t_2}, and then run any DPE solving algorithm. Alternatively, both A_{t_1} and A_{t_2} are pursuers and try to capture each other, with no evasion. Each time pursuers starting from u and v capture each other, the one carrying the lesser weight $w = \min(A_{t_1}(u), A_{t_2}(v))$ disappears, and the other then carries its own weight minus w. The amount of time it takes for all weights to disappear is the distance between A_{t_1} and A_{t_2}. If A_{t_1} and A_{t_2} sum to the same value then the system will eventually converge.

3.4.2 Information Theory. Information theory is another possible source of solutions to the NVD problem. At an abstract level NVD involves the comparison of two vectors, while in information theory it is also common to compare two vectors, e.g. when computing the Kullback-Leibler divergence [52], or mutual information [22]. These approaches have been successfully employed in data clustering [94]. As with the classical Euclidean distance, the challenge with adapting information theory to the NVD problem is to account for the underlying network structure. A large element-wise difference between portions of these vectors might be regarded as a small change if the nodes they represent are clustered in the network. Conversely, small differences should be amplified if they involve nodes that are far from each other.

There is some work to apply KL-derived divergences to networks [34, 63], but these papers focus on estimating topological differences between two graphs, rather than changes in an agent operating in an unchanging topology. For this reason, they are more related to the graph isomorphism problem [9, 51, 102] than the problem discussed here.

A possible sketch of a solution is the following. First, let us recall the discrete KL-divergence formulation:

$$\delta_{A_{t_1},A_{t_2},G} = -\sum_{v \in V} A_{t_1}(v) \log\left(\frac{A_{t_2}(v)}{A_{t_1}(v)}\right),$$

where both A_{t_1} and A_{t_2} are normalized to sum to one. This is the expectation of the logarithmic difference between the probabilities A_{t_1} and A_{t_2}, where the expectation is taken using the probabilities A_{t_1}. This formulation has three obvious problems. First, it only considers the divergence of node v with itself. Second, it ignores G's topology. Finally, it is undefined for all nodes v for which $A_{t_1}(v) = 0$ or $A_{t_2}(v) = 0$. One possible solution for the first two problems is to transform it into a weighted entropy measure, using G's topology to weight the contributions of node pair u, v to the distance – for instance, the length of the shortest path between them. However, how to deal with the third problem is not trivial. Moreover, the measure, just like KL-divergence, is not symmetric,

since $(A_{t_2}(v)/A_{t_1}(v)) \neq (A_{t_1}(v)/A_{t_2}(v))$ unless $A_{t_2}(v) = A_{t_1}(v)$. How to solve these issues is left as future work.

3.4.3 Edit Distances. Another possible source of distance measures from information theory is the concept of edit distance. The most simple possible edit distance is the Hamming distance [42]. This is defined for strings of equal length: it is the number of positions at which the corresponding symbols are different. If we were to translate the Hamming distance to continuous numerical vectors, this could be simply the sum of their absolute elementwise differences.

More sophisticated edit distances, for instance Levhenstein [55], Jaro [48], and Jaro-Winkler [99] distances, are variations of an optimized Hamming distance: they attempt to find the smallest possible number of edits to transform string A_{t_1} into string A_{t_2}. All these measures work on monodimensional strings and they thus need to be adapted to our setting, in which the vectors live on a graph.

This does not mean to transform these into a graph equivalent of the string edit distance for two reasons. First, there already exist graph edit distances [10, 15, 68, 79], but these measure the differences between two graphs with different topologies, not differences in node weight on the same graph with the same topology. In other words, they ask: how many node/edge additions/deletions does one need to perform to transform graph G_1 into graph G_2 [36]?

Second, the mentioned string edit distances could actually be enhanced by the solution of the NVD problem. Finding distances between strings are, in a sense, looking for an NVD distance. For example, $\delta(\text{analyze}, \text{analyse})$ should be smaller than the distance $\delta(\text{analyze}, \text{analyqe})$, because the similarity between letters "z" and "s" is greater than the similarity between "z" and "q", given their phonetic and cultural uses (American vs British English). Here, letters are nodes in a graph and edges connect similar letters. However, most measures of string distance do not account for such letter similarities.

The sketch of the adaptation of edit distances to solve the NVD problem is similar to the one we used to adapt the KL-divergence. Namely, each difference between elements from A_{t_1} and A_{t_2} needs to be weighted with a scheme that takes G's topology into account, for instance the length of the shortest path between the two nodes, or any measure of node similarity [13]. This is similar to the spirit of the Jaro-Winkler distance, since in that case elements in different parts of the string contribute differently to the distance – specifically, mismatches at the beginning of the string weight more than mismatches at the end of it. We also leave the development of such measure as future work.

3.4.4 Fingerprint Encodings. This class of solution comes from the computer-aided drug discovery literature [37, 66]. The approach here is to compare two molecules that have different nodes *and* edges by using either the Jaccard coefficient [27, 72, 88], or one of its generalizations, e.g. the Tversky index [73].

Note that here we are putting a few constraints on the NVD problem, while relaxing others. Specifically, we force node differences to be binary, meaning that either two nodes are matching or they are not. So A_{t_1} and A_{t_2} cannot contain continuous values. Second, these solutions also consider changes in the graph's structure. One can avoid considering this part of the measure, or they could be used to define NVD on a changing topology, namely when the graph G also changes over time. We leave such considerations as future work.

4 EVALUATION

In this section we perform experiments to answer the following questions:

Fig. 7. Instances of the Chain Test. We set as origin and destination the two endpoints of the chain, with progressively longer chains.

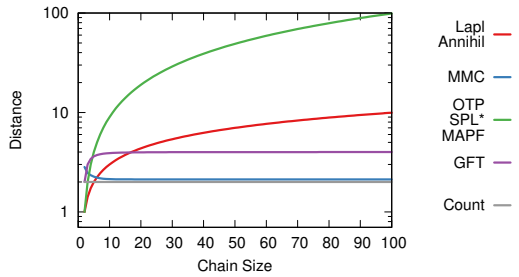


Fig. 8. The distance between A_{t_1} and A_{t_2} in the chain test as a function of chain length.

- Section 4.1: Do the distance measures discussed here make sense? Do they agree with human intuition?
- Section 4.2: How do the distances given by different measures compare with one another when applied to the same node vectors?
- Section 4.3: How do the measures behave in real world scenarios? What are their strengths and weaknesses? What questions can we answer with them?

Code and data to replicate the results of this section are included in a publicly-available repository[3]. This includes implementations of all node vector distance metrics described in Section 3.

4.1 Validation

Do the metrics discussed here behave in intuitive ways? In this section we study the behavior of the metrics in three tests using simple network topologies and occupation strategies by the agents. In the first test we examine how metric distance increases in a chain network as its length increases. In the second and third tests, we consider whether longer distances traveled by an infectious agent correlate with higher infectiousness in an epidemic model.

4.1.1 Chain Test. We consider a chain graph of length n, i.e. n nodes connected by $n-1$ edges (Figure 7). At time t_1, we set the agent entirely at one end of the chain ($A_{t_1}(v) = 1$ for node v at one end of the chain and zero at all other nodes) and at time t_2 we set the agent entirely at the other end. Figure 8 shows how the distance varies as a function of chain length n for each metric.

As we expect most metric distances increase with chain length. The OTP, all variants of shortest path metrics, and MAPF all give the same distance, equal to the number of edges $n-1$. Similarly the Laplacian and Annihilation metrics give the same distance, which is equal to $\sqrt{n-1}$. GFT rises but quickly plateaus, assigning nearly the same value to chains of length 10 and length 100. Alone among the metrics the MMC actually shrinks with chain length. While this test is simple it provides a straightforward criterion to distinguish (and select among) different behaviors.

[3]http://www.michelecoscia.com/?page_id=1733. The library depends on reLOC, an algorithm to solve the MAPF problem, which one should retrieve and compile from http://surynek.com/research/files/reLOC-0.20-kruh_043.tgz.

Topology	Lapl	MMC	Annihil	OTP	SPLS	SPLA	SPLC	GFT	MAPF	Count
ER	0.4229***	0.4644***	0.5916***	0.6630***	0.6747***	0.3132***	0.0125	0.3022***	-0.2131***	0.9736***
BA	0.4187***	0.2717***	0.5995***	0.6212***	0.7076***	0.2535***	0.1006**	0.1735***	-0.2674***	0.9667***
PC	0.4640***	0.3017***	0.6294***	0.6301***	0.7222***	0.2746***	0.1510***	0.1642***	-0.1437*	0.9608***
LFR	0.4403***	0.4082***	0.4420***	0.5962***	0.5921***	0.1714***	0.0881**	0.1671***	-0.1570**	0.9721***

Table 2. Correlation coefficients of each measure with the infection parameter β in the SI model. Bold indicates the measures with a correlation coefficient significantly higher than the Count baseline (t-test based on bootstrapping). Red indicates measures negatively correlated with β. * $p < 0.1$, ** $p < 0.01$, *** $p < 0.001$.

4.1.2 Epidemics Test. In the second test we consider a contagion model, the Susceptible-Infected (SI) model of an epidemic outbreak. Nodes can be in either of two states: Susceptible (S) or Infected (I). When a node in state S has at least one neighbor in state I, it will transition to state I with probability β. Intuitively, the higher that β is (i.e. the more infectious the disease), the farther the disease will spread across in a given period. We therefore expect the distance traveled by the disease to be positively correlated with the infectiousness parameter β. To test this we simulate an epidemic on a network, measuring the correlation between the distance traveled according to each metric and β.

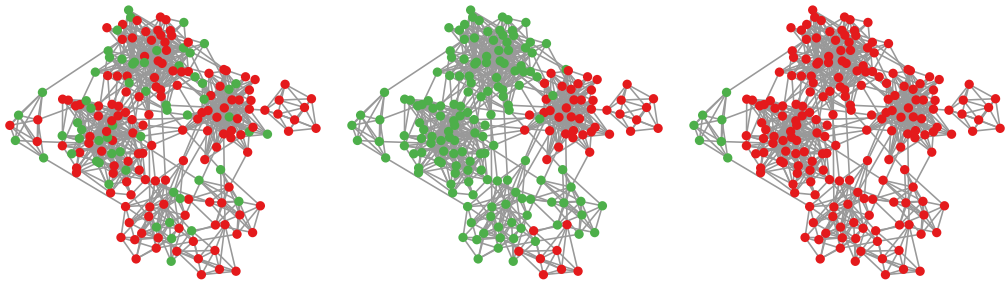
Let A_{t_1} denote the vector of infected nodes at the beginning of the outbreak and A_{t_2} the infected nodes after several infection steps. We run an SI model on a network of $|V| = 100$ nodes using 4 different topologies:

- Erdos-Renyi random networks [30]: We connect pairs of nodes uniformly at random with probability $p = .095$. We force the graph to have a single connected component.
- Barabasi-Albert preferential attachment [12]: We grow a network one node at a time until it has 100 nodes. Each node connects to 5 existing nodes picked randomly with probability proportional to their current degree.
- Clustered power-law networks [47]: We follow the same procedure as above, except that in addition every time we add an edge there is a 1% probability we also close a triangle in the network.
- LFR benchmarks [53]: This model imposes a power law degree distribution, high clustering, and a community partition on the network, with nodes more likely to connect to other nodes if they are part of the same community. We set an average degree of 5.

Thus, we start from a uniformly random network and make it progressively more complex by adding, in order, a power law degree distribution, clustering, and community structure. For each topology we run 300 SI models, in each run drawing β from a uniform distribution between 0 and 1.

Table 2 reports the correlations between β and $\delta_{A_{t_1},A_{t_2}}$. As we expect most methods return distances that are positively correlated with β. The strongest correlations are seen in the shortest path metric with single linkage. In contrast, the shortest path metric with complete linkage shows a much weaker correlation with β, and MAPF shows a strongly significant negative correlation. For comparison, we also look at a simple count of the nodes that became infected during the epidemic. As expected, a larger epidemic is associated with a higher infectiousness with a nearly perfect correlation.

4.1.3 Viral Marketing Test. In simple contagion, nodes need no reinforcement to transition between states. One infected neighbor is sufficient to transmit the disease. In the third test we examine a complex contagion model, specifically a cascade model, in which a node becomes infected whenever a fraction γ or greater of its neighbors are infected. In this model nodes can also transition back into the S state, which happens whenever the fraction of a node's infected neighbors falls below γ.



(a) (b) (c)

Fig. 9. An example run of the cascade model. (a) Starting condition, interested nodes in green, non interested nodes in red. (b) End of the process for a low γ threshold. (c) End of the process for a high γ threshold.

Topology	Lapl	MMC	Annihil	OTP	SPLS	SPLA	SPLC	GFT	MAPF	Count
ER	**0.2452*****	**0.2929*****	**0.2844*****	**-0.0354**	**0.0718**	**-0.1785*****	**-0.2829*****	**0.2421*****	**-0.0025**	-0.6419***
BA	**0.3277*****	**0.2906*****	**0.4118*****	**0.0398**	**0.0993***	**-0.3046*****	**-0.3966*****	**0.3075*****	**0.0933**	-0.6734***
PC	**0.2378*****	**0.2678*****	**0.3121*****	**-0.0120**	**0.0175**	**-0.2888*****	**-0.2957*****	**0.2893*****	**0.0084**	-0.6128***
LFR	**0.4827*****	**0.4368*****	**0.4838*****	**0.3406*****	**0.3198*****	**-0.1858*****	**-0.3399*****	**0.3565*****	**0.3877*****	-0.8287***

Table 3. Correlation coefficients of each measure with the infection parameter γ in the cascade model. Bold indicates the measures with correlation significantly higher than the Count baseline (t-test based on bootstrapping). Red indicates measures negatively correlated with γ. * $p < 0.1$, ** $p < 0.01$, *** $p < 0.001$.

Higher γ thus makes both infection harder and recovery easier. This process could represent a viral marketing campaign, where peer pressure drives sustained interest from a target audience.

The changes to the simple contagion model have a significant impact. When people have a low interest threshold, they are easily swayed. The final state may have individuals throughout the population adopting the interest (Figure 9(b)). When people have a high interest threshold, they need to be surrounded by many interested individuals to become (and remain) interested themselves. In the final state, only small communities of self-reinforcing interest remain (Figure 9(c)). As a result, the absolute change in interested individuals (Count) in this case is smaller. This effect can be seen in the negative correlation between Count and γ in Table 3.

Nevertheless, as Figure 9 shows visually, the change in the agent as a whole is larger in the high γ case. With low γ, the agent occupies all parts of the network in both the initial and final states. With high γ, the agent has a dramatically different pattern of occupation in the initial and final states. This change is not reflected in the Count variable, but it is reflected in several of the distance metrics, in particular GFT and the generalized Euclidean metrics, which are positively correlated with γ, as Table 3 shows. This further illustrates the key point that the movement of an agent on the network may differ significantly from the sum of changes across individual nodes.

4.2 Similarity

We now explore how the various measures compare with one another when applied to the same node vectors on a given network. This is a practical question; for example, it may be that some measures give similar results, regardless of how they are theoretically motivated. We generate 150 networks with each of the 4 topologies above (Erdos-Renyi, Barabasi-Albert, Power-Cluster, LFR Benchmark), together with random A_{t_1} and A_{t_2} vectors, giving 600 distances for each measure. To generate node vectors we choose between two and ten nodes at random, assigning each a random

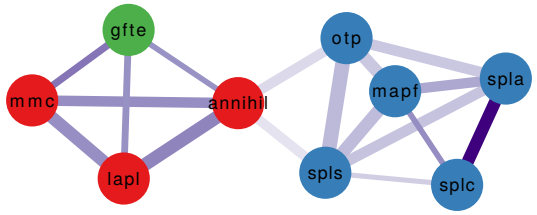


Fig. 10. Measures connected to the other measures to which they are most similar. We color nodes according to the classification proposed in Figure 6. The edge's width is proportional to the Spearman correlation value, the edge's color is proportional to how much it exceeds the connected measures' expected correlation. All links shown are significant, with darker color indicating higher significance.

weight between zero and one. For each pair of distance measures we then calculate the Spearman rank correlation across the 600 observations.

We summarize the results in Figure 10. For each distance measure we first compute its average correlation with all other measures, and then draw links to other distances for which the pairwise correlation exceeds this average. We only show correlations that significantly exceed the average, after applying a backboning of the network to correct for noise [21].

The results show that distance measures are more similar to measures that fall in the same class, as categorized in Section 3. Shortest path-based measures (blue nodes) form one cluster, while approaches based on diffusion (red and green) form another. This suggests that our classification may not just be conceptually or theoretically useful, but also indicates how numerical results may work out in practice. Notably, the two clusters align with two broad paradigms to construct metrics: a global approach, where full knowledge of the network is used to minimize edge crossings, and a myopic approach, where distances are based on random walks.

4.3 Applications

In this section we elaborate on the applications in Section 2.3. For each we explore how NVD measures can be used to characterize diffusion-like processes on networks.

4.3.1 Product Space. Recent research shows how the concept of economic complexity is useful to understand and predict the economic development of a country [44]. Informally, a country's complexity is a count of its underlying production capabilities, where having a wider range of capabilities enables more diverse products to be made. The complexity of an economy is often inferred from observing the products it makes. Estimating complexity begins by building the 'Product Space', a network of tradable products. Two products are strongly connected if they often co-appear in the export baskets of countries. In terms of the NVD problem, a country is an agent that occupies a network of goods, and seeks to diversify, spreading to parts of the network that it has not yet reached.

Typically, the amount of diversification a country has realized over some period is estimated by the number of new exports it has added. However, some products are more closely related than others (as the Product Space itself shows), so a simple count of new products could provide poor estimates of diversification when these products are too closely related to one another. In principle, NVD measures could capture diversification in a more sophisticated way, accounting for the distance the country traverses across the network.

To examine this, we take the export vectors of 76 countries from 1962 to 2013. We average their exports in each decade to smooth out short-term fluctuations. Then we calculate the speed with which a country diversifies from one decade to the next using our NVD measures. We compute the

Measure	1 Digit (10)	2 Digit (68)	3 Digit (237)	4 Digit (774)
Lapl	0.2086^{***}	0.1104^{*}	0.1870^{***}	0.2154^{***}
MMC	0.2215^{***}	0.1785^{***}	0.1670^{***}	0.1314^{**}
Annihil	$\mathbf{0.2314}^{***}$	0.1844^{***}	0.2302^{***}	0.2607^{***}
OTP	0.2125^{***}	0.2317^{***}	0.2865^{***}	0.3056^{***}
SPLS	0.2148^{***}	$\mathbf{0.2401}^{***}$	$\mathbf{0.2917}^{***}$	$\mathbf{0.3073}^{***}$
SPLA	0.1000	0.0780	0.2250^{***}	0.2464^{***}
SPLC	0.0991	0.0563	0.2323^{***}	0.2288^{***}
GFT	0.1924^{***}	0.1012	0.1022^{*}	0.1729^{***}
MAPF	0.2113^{***}	NA	NA	NA
Count	0.0994	0.0418	0.1847^{***}	0.2612^{***}
ECI	0.0123	0.0471	0.1845^{***}	0.1735^{***}

Table 4. The Spearman correlation of each NVD measure with GDP per capita change, for different aggregation levels in the Product Space. The numbers in parentheses in the column headers are the number of products ($|V|$) in the network. $^{***} = p < 0.01$, $^{**} = p < 0.05$, $^{*} = p < 0.1$.

Spearman rank correlation of this speed with the absolute value of GDP per capita change over the following decade, with a small time overlap (Table 4). For example, we observe the distance covered by a country from the 1962-72 period to the 1973-83 period, and compute its correlation to the country's absolute GDP per capita change between 1980 and 1990.

Consistent with our expectation, countries that realize faster economic change are also traversing the network more quickly. The number of products in existence is inherently ambiguous, so we compute these correlations at four aggregation levels of SITC product categories, with 10, 68, 237, or 774 products. Notably, the change in diversity (Count) in the previous decade is not consistently associated with GDP change in the next decade. Its performance improves when there are more goods, but this lack of consistency makes a direct count of products risky in practical settings; it is difficult to know if the number of goods used in a given data set is large enough. On the other hand, a number of NVD measures consistently have a significant association with GDP change. We also note that more sophisticated measures such as the Economic Complexity Index [44] (ECI) behave similarly to the count of goods.

4.3.2 Epidemics. With adaptation, NVD metrics could potentially shed light on how dangerous a disease is, by quantifying spreading outcomes on a network as a whole, rather than local infectiousness among individuals. To demonstrate this idea notionally, we look at data from the World Health Organization (WHO)[4] on four outbreaks between 1996 and 2019: Ebola, Dengue, Avian Flu, and Zika. For each outbreak, we consider the movement of individuals across countries as characterized by the network of worldwide international flight traffic using data from OAG [5]. Using each distance measure, we compute how far the disease traveled in each month after the disease had first appeared, and then average these monthly distances to obtain a typical speed (Table 5).

With some exceptions, most methods agree on the relative order of the outbreaks, ranking Zika the fastest and Ebola the slowest. While Ebola can cause immediate infection when in contact with a symptomatic individual, an asymptomatic Ebola infected person cannot spread the virus[6], while a symptomatic patient is so debilitated as to be extremely noticeable. This lowers the potential of

[4]https://www.who.int/csr/don/archive/disease/en/

[5]https://www.oag.com/

[6]https://www.who.int/news-room/fact-sheets/detail/ebola-virus-disease

Measure	Ebola	Dengue	Avian Flu	Zika
Lapl	0.0048	0.0152	0.0172	0.0476
MMC	0.0283	0.0545	0.0580	0.1931
Annihil	0.0262	0.0843	0.0945	0.1963
OTP	0.0435	0.1032	0.1502	0.3977
SPLS	0.0435	0.1032	0.1522	0.4272
SPLA	0.2412	0.4843	0.7385	1.3350
SPLC	0.2985	0.6162	0.9524	1.6623
GFT	0.4408	1.9936	1.8378	5.8700
MAPF	0.3040	0.6828	1.1037	1.6773

Table 5. The average monthly distance covered on the flight network by the four diseases.

Measure	Title	Author
Lapl	The Girl Who Kicked the Hornets' Nest	Stieg Larsson
MMC	The Girl Who Kicked the Hornets' Nest	Stieg Larsson
Annihil	The Girl Who Kicked the Hornets' Nest	Stieg Larsson
OTP	The Girl Who Kicked the Hornets' Nest	Stieg Larsson
SPLS	The Girl Who Kicked the Hornets' Nest	Stieg Larsson
SPLA	Harry Potter and the Half-Blood Prince	J.K. Rowling
SPLC	Harry Potter and the Half-Blood Prince	J.K. Rowling
GFT	Fight Club	Chuck Palahniuk

Table 6. The fastest moving books in the Anobii dataset, per measure.

the disease to be a pandemic. In contrast for Zika the majority of infected individuals can carry the virus even if not symptomatic[7], and most carriers are not easily detected.

4.3.3 Viral Marketing. A similar analysis can be done to gauge the spread of products on a social network. To demonstrate this idea we use data from the online social network Anobii [1, 2]. In Anobii, users have "bookshelves" where they keep the books they have read. Users can connect to friends, and discover the books they have read. In terms of the NVD problem a book is an agent, occupying more nodes as more users add it to their bookshelves.

We consider the spread of books through the network using data collected in six two-week snapshots over twelve weeks from September to December 2009. For each book we calculate its average speed across the observation period. For each measure we report the fastest- (Table 6) and slowest- (Table 7) moving books. In both cases a number of measures agree on the same books.

The results here show how NVD measures could be useful in understanding whether an agent is propagating by a given network's edges or not. Anecdotally, the fastest-moving books are well-explained by external shocks. For instance, the Harry Potter movie adaptation of the observed book was released in July 2009, just before the observation window, and likely fostered book sales. Movies adapting prequels to the Stieg Larsson book were released in May and September 2009, also not long before our observation window. The slowest-moving books were also adapted into movies, but in contrast these were released years before our observation period (the latest in 2001) and likely played little role in their diffusion in 2009. (Anobii is especially popular in Italy, and so we link the explanation of the results to the cultural landscape in that country.)

[7]https://www.who.int/news-room/fact-sheets/detail/zika-virus

Measure	Title	Author
Lapl	Novecento	Alessandro Baricco
MMC	Siddharta	Hermann Hesse
Annihil	Novecento	Alessandro Baricco
OTP	Excursion to Tindari	Andrea Camilleri
SPLS	The Snack Thief	Andrea Camilleri
SPLA	The Snack Thief	Andrea Camilleri
SPLC	The Snack Thief	Andrea Camilleri
GFT	Novecento	Alessandro Baricco

Table 7. The slowest moving books in the Anobii dataset, per measure.

This analysis is corroborated by Google Trends data[8], which give average Trends score during the observation period of 29.8, 14.5, and 2.7 for the fastest books and 1.0, 0.2, 0.06, and 0.04 for the slowest books. Note that the seven books all have comparable popularity in Anobii in the observation window. This suggests that the fast-moving books are fast because social interest, likely driven by external news sources, allowed them to ignore the network structure. In contrast slow-moving books instead used the social network's edges to propagate via word-of-mouth.

5 CONCLUSION

Usually distances on a network are measured between individual nodes. In this paper, we define the Node Vector Distance problem, which is to quantify the distance between two vectors of node weights on a network. We show that distances between groups of nodes can be meaningfully defined, and that in fact such distances have already appeared implicitly in a wide range of problems in network science. We present a broad discussion of the problem that includes its applications, solution approaches, and a characterization of how a variety of metrics differ.

We outline a few broad approaches to solving the NVD problem: generalizations of the Euclidean distance, shortest path-based distances, spectral approaches, and adaptations of common computer science problems or metrics related to NVD. The classification here is meant as a guide to help show different ways in which such metrics can be motivated. We would not be surprised if other approaches can be taken that do not fit any of our four categories. For example, we envision that new information-theoretic network distance measures could appear in the near future.

In our experiments, we see how a number of metrics compare with the intuitive expectations of a distance measure. For instance, most metrics here rise monotonically with chain length when comparing node vectors at opposite ends of a chain network. The metrics also show correlations with the infectiousness parameter in epidemic models.

Node vector distances have important and overlooked applications in network science. In this review we sample just a few of these applications, showing how measures of distances between groups of nodes are useful in diverse problem areas that include computer vision, viral marketing, epidemiology, and economics. Given how diverse these applications are, it is very likely that many other problems in network science could be usefully posed in terms of distance metrics of the kind we describe here.

[8]https://trends.google.com/trends/explore?date=2009-09-11%202009-12-24&geo=IT&q=%22La%20regina%20dei%20castell%20di%20carta%22,%22Harry%20Potter%20e%20il%20principe%20mezzosangue%22,%22Fight%20Club%22,%22La%20gita%20a%20Tindari%22,%22novecento%20baricco%22 and https://trends.google.com/trends/explore?date=2009-09-11%202009-12-24&geo=IT&q=%22Harry%20Potter%20e%20il%20principe%20mezzosangue%22,%22il%20ladro%20di%20merendine%22,%22siddharta%20hesse%22

We conclude by briefly describing possible directions for future research. These fall outside the scope of this paper, but could be fruitful avenues for further work:

- Deeper explorations of the measures we sketched here, such as discrete pursuit-evasion games, the KL-divergence, and string edit distances;
- Development of new measures, such as adaptations of the cosine or correlation distances in analogy to our adaptation of the Euclidean distance;
- Relaxing the condition that the network topology is fixed. Many real world networks evolve over time, and NVD metrics could be generalized to account for this evolution;
- Exploring the statistical properties of NVD metrics. For some NVD metrics, such as those in the spectral class, it may be possible to find closed-form descriptions of their distributional properties. For others, bootstrapping methods may provide this information.

In particular, we look forward to applications of NVD metrics to new use cases, such as to study cascading failures or metabolic networks. As the number of applications grows, meta-analyses could provide intuition into how NVD metrics perform and how best to match the characteristics of the metric with the network it is applied to. For instance, it may be that spectral metrics are natural for networks with diffusion-like dynamics, while shortest path metrics may be more appropriate in cases where the dynamics involve strategic actions by agents. We believe such applications, and the other directions above, would bring insight to myriad processes on networked systems.

REFERENCES

[1] Luca Maria Aiello, Alain Barrat, Ciro Cattuto, Giancarlo Ruffo, and Rossano Schifanella. 2010. Link creation and profile alignment in the aNobii social network. In *2010 IEEE Second International Conference on Social Computing*. IEEE, 249–256.

[2] Luca Maria Aiello, Martina Deplano, Rossano Schifanella, and Giancarlo Ruffo. 2012. People are strange when you're a stranger: Impact and influence of bots on social networks. In *Sixth International AAAI Conference on Weblogs and Social Media*.

[3] Saeed Akhoondian Amiri, Lukasz Kaiser, Stephan Kreutzer, Roman Rabinovich, and Sebastian Siebertz. 2015. Graph searching games and width measures for directed graphs. In *LIPIcs-Leibniz International Proceedings in Informatics*, Vol. 30. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

[4] Réka Albert and Albert-László Barabási. 2002. Statistical mechanics of complex networks. *Reviews of modern physics* 74, 1 (2002), 47.

[5] Brian Alspach. 2006. Searching and sweeping graphs: a brief survey. *Le matematiche* 59, 1, 2 (2006), 5–37.

[6] Anton Andreychuk, Konstantin Yakovlev, Dor Atzmon, and Roni Sternr. 2019. Multi-Agent Pathfinding with Continuous Time. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI*, Vol. 19.

[7] Aamir Anis, Akshay Gadde, and Antonio Ortega. 2014. Towards a sampling theorem for signals on arbitrary graphs. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 3864–3868.

[8] Ira Assent, Andrea Wenning, and Thomas Seidl. 2006. Approximation techniques for indexing the earth mover's distance in multimedia databases. In *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*. IEEE, 11–11.

[9] László Babai. 2016. Graph isomorphism in quasipolynomial time. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 684–697.

[10] Vladimír Baláž, Jaroslav Koča, Vladimír Kvasnička, and Milan Sekanina. 1986. A metric for graphs. *Časopis pro pěstování matematiky* 111, 4 (1986), 431–433.

[11] Alfons Balmann. 1997. Farm-based modelling of regional structural change: A cellular automata approach. *European review of agricultural economics* 24, 1 (1997), 85–108.

[12] Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *science* 286, 5439 (1999), 509–512.

[13] Vincent D Blondel, Anahí Gajardo, Maureen Heymans, Pierre Senellart, and Paul Van Dooren. 2004. A measure of similarity between graph vertices: Applications to synonym extraction and web searching. *SIAM review* 46, 4 (2004), 647–666.

[14] Sergey V Buldyrev, Roni Parshani, Gerald Paul, H Eugene Stanley, and Shlomo Havlin. 2010. Catastrophic cascade of failures in interdependent networks. *Nature* 464, 7291 (2010), 1025–1029.

[15] Horst Bunke. 1997. On a relation between graph edit distance and maximum common subgraph. *Pattern Recognition Letters* 18, 8 (1997), 689–694.

[16] Rainer Burkard, Mauro Dell'Amico, and Silvano Martello. 2012. *Assignment problems: revised reprint.* SIAM.

[17] Eunjoon Cho, Seth A Myers, and Jure Leskovec. 2011. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1082–1090.

[18] Vittoria Colizza, Alain Barrat, Marc Barthélemy, and Alessandro Vespignani. 2006. The role of the airline transportation network in the prediction and predictability of global epidemics. *Proceedings of the National Academy of Sciences of the United States of America* 103, 7 (2006), 2015–2020.

[19] Michele Coscia. 2020. Generalized Euclidean Measure to Estimate Network Distances. *to appear* (2020).

[20] Michele Coscia, Fosca Giannotti, and Dino Pedreschi. 2011. A classification for community discovery methods in complex networks. *Statistical Analysis and Data Mining: The ASA Data Science Journal* 4, 5 (2011), 512–546.

[21] Michele Coscia and Frank MH Neffke. 2017. Network backboning with noisy data. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*. IEEE, 425–436.

[22] Thomas M Cover and Joy A Thomas. 2012. *Elements of information theory.* John Wiley & Sons.

[23] Narsingh Deo and Chi-Yin Pang. 1984. Shortest-path algorithms: Taxonomy and annotation. *Networks* 14, 2 (1984), 275–323.

[24] Michel Marie Deza and Elena Deza. 2009. Encyclopedia of distances. In *Encyclopedia of distances*. Springer, 1–583.

[25] Edsger W Dijkstra. 1959. A note on two problems in connexion with graphs. *Numerische mathematik* 1, 1 (1959), 269–271.

[26] Andrew Dobson, Kiril Solovey, Rahul Shome, Dan Halperin, and Kostas E Bekris. 2017. Scalable asymptotically-optimal multi-robot motion planning. In *2017 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE, 120–127.

[27] Paul D Dobson, Karin Lanthaler, Stephen G Oliver, and Douglas B Kell. 2009. Implications of the dominant role of transporters in drug uptake by cells (supplementary material). *Current topics in medicinal chemistry* 9, 2 (2009), 163–181.

[28] David Easley and Jon Kleinberg. 2010. *Networks, crowds, and markets: Reasoning about a highly connected world.* Cambridge University Press.

[29] Matthias Erbar, Martin Rumpf, Bernhard Schmitzer, and Stefan Simon. 2017. Computation of Optimal Transport on Discrete Metric Measure Spaces. *arXiv preprint arXiv:1707.06859* (2017).

[30] Paul Erdos and Alfréd Rényi. 1960. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci* 5, 1 (1960), 17–60.

[31] Montacer Essid and Justin Solomon. 2017. Quadratically-Regularized Optimal Transport on Graphs. *arXiv preprint arXiv:1704.08200* (2017).

[32] Klaus-Tycho Foerster, Linus Groner, Torsten Hoefler, Michael Koenig, Sascha Schmid, and Roger Wattenhofer. 2017. Multi-agent Pathfinding with n Agents on Graphs with n Vertices: Combinatorial Classification and Tight Algorithmic Bounds. In *International Conference on Algorithms and Complexity*. Springer, 247–259.

[33] Fedor V Fomin and Dimitrios M Thilikos. 2008. An annotated bibliography on guaranteed graph searching. *Theoretical computer science* 399, 3 (2008), 236–245.

[34] David J Galas, Gregory Dewey, James Kunert-Graf, and Nikita A Sakhanenko. 2017. Expansion of the Kullback-Leibler Divergence, and a new class of information metrics. *Axioms* 6, 2 (2017), 8.

[35] Ayalvadi Ganesh, Laurent Massoulié, and Don Towsley. 2005. The effect of network topology on the spread of epidemics. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, Vol. 2. IEEE, 1455–1466.

[36] Xinbo Gao, Bing Xiao, Dacheng Tao, and Xuelong Li. 2010. A survey of graph edit distance. *Pattern Analysis and applications* 13, 1 (2010), 113–129.

[37] Fahimeh Ghasemi, Afshin Fassihi, Horacio Pérez-Sánchez, and Alireza Mehri Dehnavi. 2017. The role of different sampling methods in improving biological activity prediction using deep belief network. *Journal of computational chemistry* 38, 4 (2017), 195–203.

[38] Clark R Givens, Rae Michael Shortt, et al. 1984. A class of Wasserstein metrics for probability distributions. *The Michigan Mathematical Journal* 31, 2 (1984), 231–240.

[39] Julio E Godoy, Ioannis Karamouzas, Stephen J Guy, and Maria Gini. 2015. Adaptive learning for multi-agent navigation. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1577–1585.

[40] Oded Goldreich. 2011. Finding the Shortest Move-Sequence in the Graph-Generalized 15-Puzzle Is NP-Hard.

[41] Patric Hagmann, Leila Cammoun, Xavier Gigandet, Reto Meuli, Christopher J Honey, Van J Wedeen, and Olaf Sporns. 2008. Mapping the structural core of human cerebral cortex. *PLoS biology* 6, 7 (2008), e159.

[42] Richard W Hamming. 1950. Error detecting and error correcting codes. *The Bell system technical journal* 29, 2 (1950), 147–160.

[43] David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. 2011. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis* 30, 2 (2011), 129–150.

[44] Ricardo Hausmann, César A Hidalgo, Sebastián Bustos, Michele Coscia, Alexander Simoes, and Muhammed A Yildirim. 2014. *The atlas of economic complexity: Mapping paths to prosperity.* Mit Press.

[45] César A Hidalgo, Bailey Klinger, A-L Barabási, and Ricardo Hausmann. 2007. The product space conditions the development of nations. *Science* 317, 5837 (2007), 482–487.

[46] Frank L Hitchcock. 1941. The distribution of a product from several sources to numerous localities. *Studies in Applied Mathematics* 20, 1-4 (1941), 224–230.

[47] Petter Holme and Beom Jun Kim. 2002. Growing scale-free networks with tunable clustering. *Physical review E* 65, 2 (2002), 026107.

[48] Matthew A Jaro. 1989. Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. *J. Amer. Statist. Assoc.* 84, 406 (1989), 414–420.

[49] George Karakostas. 2008. Faster approximation schemes for fractional multicommodity flow problems. *ACM Transactions on Algorithms (TALG)* 4, 1 (2008), 13.

[50] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 137–146.

[51] Johannes Kobler, Uwe Schöning, and Jacobo Torán. 2012. *The graph isomorphism problem: its structural complexity.* Springer Science & Business Media.

[52] Solomon Kullback. 1959. *Information theory and statistics.* Technical Report.

[53] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. 2008. Benchmark graphs for testing community detection algorithms. *Physical review E* 78, 4 (2008), 046110.

[54] Jure Leskovec, Lada A Adamic, and Bernardo A Huberman. 2007. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)* 1, 1 (2007), 5.

[55] Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, Vol. 10. 707–710.

[56] Jiaoyang Li, Pavel Surynek, Ariel Felner, Hang Ma, TK Satish Kumar, and Sven Koenig. 2019. Multi-agent path finding for large agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 7627–7634.

[57] WUCHEN Li, ERNEST K Ryu, STANLEY Osher, WOTAO Yin, and WILFRID Gangbo. 2017. A parallel method for earth mover's distance. *UCLA Comput. Appl. Math. Pub.(CAM) Rep* (2017), 17–12.

[58] David Liben-Nowell and Jon Kleinberg. 2007. The link-prediction problem for social networks. *journal of the Association for Information Science and Technology* 58, 7 (2007), 1019–1031.

[59] Minghua Liu, Hang Ma, Jiaoyang Li, and Sven Koenig. 2019. Task and Path Planning for Multi-Agent Pickup and Delivery. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1152–1160.

[60] Flaminia L Luccio. 2007. Intruder capture in Sierpinski graphs. In *FUN*. Springer, 249–261.

[61] Hang Ma, TK Satish Kumar, and Sven Koenig. 2017. Multi-agent path finding with delay probabilities. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[62] Jan Maas. 2011. Gradient flows of the entropy for finite Markov chains. *Journal of Functional Analysis* 261, 8 (2011), 2250–2292.

[63] Christopher L McClendon, Lan Hua, Gabriela Barreiro, and Matthew P Jacobson. 2012. Comparing conformational ensembles using the Kullback–Leibler divergence expansion. *Journal of chemical theory and computation* 8, 6 (2012), 2115–2126.

[64] Andrew McGregor and Daniel Stubbs. 2013. Sketching earth-mover distance on graph metrics. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Springer, 274–286.

[65] Victor Gabriel Lopez Mejia, Frank L Lewis, Yan Wan, Edgar N Sanchez, and Lingling Fan. 2019. Solutions for Multiagent Pursuit-Evasion Games on Communication Graphs: Finite-Time Capture and Asymptotic Behaviors. *IEEE Trans. Automat. Control* (2019).

[66] Kenneth M Merz Jr, Dagmar Ringe, and Charles H Reynolds. 2010. *Drug design: structure-and ligand-based approaches.* Cambridge University Press.

[67] Gaspard Monge. 1781. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris* (1781).

[68] Richard Myers, RC Wison, and Edwin R Hancock. 2000. Bayesian graph edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 6 (2000), 628–635.

[69] Sunil K Narang, Akshay Gadde, Eduard Sanou, and Antonio Ortega. 2013. Localized iterative methods for interpolation in graph structured data. In *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*. IEEE, 491–494.
[70] Frank Neffke, Matté Hartog, Ron Boschma, and Martin Henning. 2018. Agents of structural change: The role of firms and entrepreneurs in regional diversification. *Economic Geography* 94, 1 (2018), 23–48.
[71] Frank Neffke, Martin Henning, and Ron Boschma. 2011. How do regions diversify over time? Industry relatedness and the development of new growth paths in regions. *Economic Geography* 87, 3 (2011), 237–265.
[72] Steve O'Hagan and Douglas B Kell. 2015. Understanding the foundations of the structural similarities between marketed drugs and endogenous human metabolites. *Frontiers in pharmacology* 6 (2015), 105.
[73] Steve O'Hagan and Douglas B Kell. 2016. MetMaxStruct: a Tversky-similarity-based strategy for analysing the (sub) structural similarities of drugs and endogenous metabolites. *Frontiers in pharmacology* 7 (2016), 266.
[74] Torrence D Parsons. 1978. Pursuit-evasion in a graph. In *Theory and applications of graphs*. Springer, 426–441.
[75] Romualdo Pastor-Satorras and Alessandro Vespignani. 2001. Epidemic dynamics and endemic states in complex networks. *Physical Review E* 63, 6 (2001), 066117.
[76] Ofir Pele and Michael Werman. 2008. A linear time histogram metric for improved sift matching. In *European conference on computer vision*. Springer, 495–508.
[77] Ofir Pele and Michael Werman. 2009. Fast and robust earth mover's distances. In *Computer vision, 2009 IEEE 12th international conference on*. IEEE, 460–467.
[78] Shmuel Peleg, Michael Werman, and Hillel Rom. 1989. A unified approach to the change of resolution: Space and gray-level. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11, 7 (1989), 739–742.
[79] Kaspar Riesen and Horst Bunke. 2009. Approximate graph edit distance computation by means of bipartite graph matching. *Image and Vision computing* 27, 7 (2009), 950–959.
[80] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. 2000. The earth mover's distance as a metric for image retrieval. *International journal of computer vision* 40, 2 (2000), 99–121.
[81] Aliaksei Sandryhaila and Jose MF Moura. 2014. Discrete Signal Processing on Graphs: Frequency Analysis. *IEEE Trans. Signal Processing* 62, 12 (2014), 3042–3054.
[82] Filippo Santambrogio. 2015. Optimal transport for applied mathematicians. *Birkäuser, NY* (2015).
[83] Evangelos Sariyanidi, Hatice Gunes, and Andrea Cavallaro. 2015. Automatic analysis of facial affect: A survey of registration, representation, and recognition. *IEEE transactions on pattern analysis and machine intelligence* 37, 6 (2015), 1113–1133.
[84] Guni Sharon, Roni Stern, Ariel Felner, and Nathan R Sturtevant. 2015. Conflict-based search for optimal multi-agent pathfinding. *Artificial Intelligence* 219 (2015), 40–66.
[85] David I Shuman, Benjamin Ricaud, and Pierre Vandergheynst. 2016. Vertex-frequency analysis on graphs. *Applied and Computational Harmonic Analysis* 40, 2 (2016), 260–291.
[86] Jamie Snape, Jur Van Den Berg, Stephen J Guy, and Dinesh Manocha. 2011. The hybrid reciprocal velocity obstacle. *IEEE Transactions on Robotics* 27, 4 (2011), 696–706.
[87] Justin Solomon, Raif Rustamov, Leonidas Guibas, and Adrian Butscher. 2016. Continuous-flow graph transportation distances. *arXiv preprint arXiv:1603.06927* (2016).
[88] O Steve, Neil Swainston, Julia Handl, Douglas B Kell, et al. 2015. A 'rule of 0.5'for the metabolite-likeness of approved pharmaceutical drugs. *Metabolomics* 11, 2 (2015), 323–339.
[89] Nicholas M Stiffler and Jason M O'Kane. 2016. Pursuit-evasion with fixed beams. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 4251–4258.
[90] Pavel Surynek. 2014. Compact representations of cooperative path-finding as SAT based on matchings in bipartite graphs. In *2014 IEEE 26th International Conference on Tools with Artificial Intelligence*. IEEE, 875–882.
[91] Gabor J Szekely and Maria L Rizzo. 2005. Hierarchical clustering via joint between-within distances: Extending Ward's minimum variance method. *Journal of classification* 22, 2 (2005), 151–183.
[92] Andrea Tacchella, Matthieu Cristelli, Guido Caldarelli, Andrea Gabrielli, and Luciano Pietronero. 2012. A new metrics for countries' fitness and products' complexity. *Scientific reports* 2 (2012), 723.
[93] Cédric Villani. 2003. *Topics in optimal transportation*. Number 58. American Mathematical Soc.
[94] Nguyen Xuan Vinh, Julien Epps, and James Bailey. 2010. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research* 11, Oct (2010), 2837–2854.
[95] Glenn Wagner and Howie Choset. 2015. Subdimensional expansion for multirobot path planning. *Artificial Intelligence* 219 (2015), 1–24.
[96] Thayne T Walker, David M Chan, and Nathan R Sturtevant. 2017. Using hierarchical constraints to avoid conflicts in multi-agent pathfinding. In *Twenty-Seventh International Conference on Automated Planning and Scheduling*.

[97] Thayne T Walker, Nathan R Sturtevant, and Ariel Felner. 2018. Extended Increasing Cost Tree Search for Non-Unit Cost Domains.. In *IJCAI*. 534–540.

[98] Yu-Xiang Wang, James Sharpnack, Alex Smola, and Ryan J Tibshirani. 2016. Trend filtering on graphs. *Journal of Machine Learning Research* 17, 105 (2016), 1–41.

[99] William E Winkler. 1990. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. (1990).

[100] Laurenz Wiskott, Norbert Krüger, N Kuiger, and Christoph Von Der Malsburg. 1997. Face recognition by elastic bunch graph matching. *IEEE Transactions on pattern analysis and machine intelligence* 19, 7 (1997), 775–779.

[101] Konstantin Yakovlev and Anton Andreychuk. 2017. Any-angle pathfinding for multiple agents based on SIPP algorithm. In *Twenty-Seventh International Conference on Automated Planning and Scheduling*.

[102] Xifeng Yan and Jiawei Han. 2002. gspan: Graph-based substructure pattern mining. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*. IEEE, 721–724.

[103] Jingjin Yu and Steven M LaValle. 2013. Multi-agent path planning and network flow. In *Algorithmic Foundations of Robotics X*. Springer, 157–173.

[104] Jingjin Yu and Daniela Rus. 2015. Pebble motion on graphs with rotations: Efficient feasibility tests and planning algorithms. In *Algorithmic Foundations of Robotics XI*. Springer, 729–746.