Extraction and analysis of traffic and topologies of transportation networks

Maciej Kurant and Patrick Thiran
Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland
(Received 10 April 2006; revised manuscript received 15 July 2006; published 25 September 2006)

The knowledge of real-life traffic patterns is crucial for a good understanding and analysis of transportation systems. These data are quite rare. In this paper we propose an algorithm for extracting both the real physical topology and the network of traffic flows from timetables of public mass transportation systems. We apply this algorithm to timetables of three large transportation networks. This enables us to make a systematic comparison between three different approaches to construct a graph representation of a transportation network; the resulting graphs are fundamentally different. We also find that the real-life traffic pattern is very heterogenous, in both space and traffic flow intensities, which makes it very difficult to approximate the node load with a number of topological estimators.

DOI: 10.1103/PhysRevE.74.036114 PACS number(s): 89.75.Hc, 89.75.Fb, 89.40.Bb

I. INTRODUCTION

In recent years, studies of transportation networks have drawn a substantial amount of attention in the physics community. The graphs derived from the physical infrastructure of such networks were analyzed in the examples of a power grid [1,2], a railway network [3,4], road networks [5–9], pipeline network [4], and urban mass transportation systems [10–14]. These studies have one important feature in common—they focus exclusively on the topology of the network, and they do not take into account real-life traffic patterns. This makes the view very incomplete, because carrying traffic is the ultimate goal of every transportation system. Facing a lack of real-life traffic data, some authors have tried to estimate traffic patterns based exclusively on topology. Probably the most common load estimator is *betweenness* (used, e.g., in Refs, [15–20]), which assumes that each pair of nodes exchanges the same amount of traffic. But real-life traffic patterns are in fact very heterogenous, in both space and traffic flow intensities. Therefore the most important nodes and edges from a topological point of view might not necessarily carry the most traffic. In Ref. [21] we show that in typical transportation networks the correlation between the real load and betweenness is very low. Therefore it is essential for some applications to know the real traffic pattern.

Interestingly, networks of traffic flows were studied separately; see the example of flows of people within a city [22] and commuting traffic flows between different cities [23]. These studies, in turn, neglect the underlying physical topology, making the analysis incomplete. For instance, it is impossible to detect the most loaded physical edges, which might have crucial meaning for the resilience of the system. A comprehensive view of the system often requires one to analyze both layers (physical and traffic) together.

Unfortunately, data sets including both physical topology and traffic flows are rather sparse and difficult to obtain. In this paper we propose an approach to extract the physical structure and network of traffic flows from *timetables*. Timetables of trains, buses, trams, metros, and other means of mass transportation (henceforth called *vehicles*) are publicly available. They provide us with the available connections and their times. Timetables also contain information about the physical structure of the network and the traffic flows in it, but as we show later, they often require a nontrivial preprocessing to be revealed.

II. SPACES AND THE DIFFICULTY OF THE PROBLEM

In order to position our contribution in the range of works in the field, we begin with a systematic definition of the topology of transportation systems. The set of nodes is defined by the set of all stations (train stations, bus stops, etc.). It is not obvious, however, what should be interpreted as an edge. Its choice depends on what we want to be reflected by the topology of the physical graph. In the literature there are essentially three approaches that define three different "spaces": here we call them "space of changes," "space of stops," and "space of stations."

In the *space of changes*, two stations are considered to be connected by a link when there is at least one vehicle that stops at both stations. In other words, all stations used by a single vehicle are fully interconnected and form a clique. This approach neglects the physical distance between the stations. Instead, in the resulting topology, the length of the shortest path between two arbitrary stations A and B is the *number of changes* of the mean of transportation one needs to get from A to B.[1] This approach was used in Refs. [3,12,13]; in the latter, the authors used the term *space P*.

In the *space of stops*, two stations are connected if they are two consecutive stops on a route of at least one vehicle [13]. Here the length of the shortest path between two stations is the minimal *number of stops* one needs to make. Note that the number of stations traversed on the way might be larger, because the vehicles do not necessary stop on all of them.

[1]In this sense, a graph in the space of changes is closely related to the *dual* interpretation of urban road networks [5,7,42], where streets (of a given name) map to nodes and intersections between streets map to links between the nodes. In a transportation network in the space of changes, the length of the shortest path is the number of changes of the mean of transportation, whereas the length of the shortest path in a dual graph of a city is the number of changes of streets on the way from the starting point to destination.

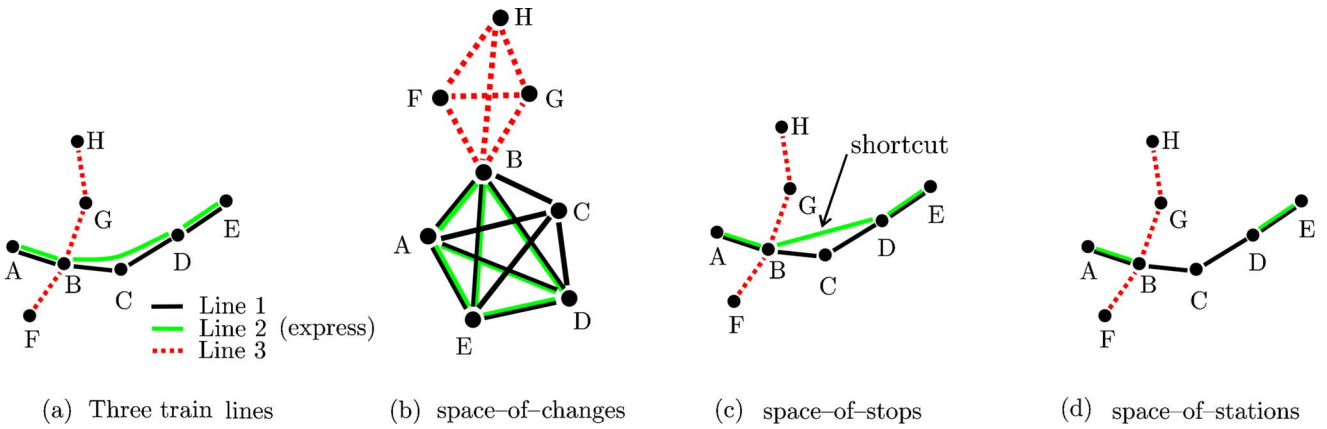


FIG. 1. (Color online) An illustration of the transportation network topology in three spaces. (a) The routes of three vehicles. The route of line 2 passes through node *C* on the way from *B* to *D*, but the vehicle does not stop there. (b) The topology in the space of changes. Each route results in a clique. An edge is indicated by two colors, when it originates from two routes, but is merged into a single link. (c) The topology in the space of stops. The "shortcut" *B*-*D* is a legitimate edge in this space. (d) The topology in the space of stations. This graph reflects the topology of the real-life infrastructure.

In the *space of stations*, two stations are connected only if they are physically directly connected (with no station in between). This reflects the topology of the real-life infrastructure. Here, the length of the shortest path between two stations is the minimal *number of stations* one has to traverse (stopping or not). This approach was used in Ref. [4,10,11,14].

In Fig. 1 we give an illustration of the three spaces. It is easy to see that the graph in the space of stations is a subgraph of the graph in the space of stops, which in turn is a subgraph of the graph in space of changes.

The topologies in the space of changes and space of stops can be directly obtained from timetables. In the space of changes, for each vehicle, we fully connect all stations it stops at. Then we simplify the resulting graph by deleting multiedges. In the space of stops, we connect every two consecutive stops in the routes of vehicles. As shown in Fig. 1(c), the topology in the space of stops can have shortcut links that do not exist in the real-life infrastructure. These shortcuts should be eliminated in the space-of-stations topology, which makes it more challenging to obtain. To the best of our knowledge, the only work on extracting the real physical structure (the topology in the space of stations) from timetables was done in the context of railway networks in the Ph.D. dissertation of Lebers [24]. The proposed solution first obtains the physical graph in the space of stops. Next, specific structures in the initial physical graph, called *edge bundles*, are detected. The Hamiltonian paths[2] within these bundles should indicate the real (nonshortcut) edges. Unfortunately, the bundle recognition problem turned out to be NP complete. The heuristics proposed in Ref. [24] result in a correct real and shortcut classification of 80% of the edges in the studied graphs. The approach we propose in this paper is based on simple observations that were omitted in Ref. [24]. This results in a much simpler and more effective algorithm.

[2]The *Hamiltonian path* is a path that passes through every vertex of a graph exactly once.

III. RELATED WORK

Timetables have been used as a data source for a network construction in Refs. [3,13]. However, the topologies obtained in these works were in either the space of changes or space of stops; neither of them reflected the real-life infrastructure. Moreover, real traffic patterns were not considered in these studies. This is understandable, because it is difficult to interpret a traffic flow in the spaces of changes and stops. In the space of changes every train transforms into a clique. Counting for a given edge the number of cliques it participates in would result in a weighted graph where we could analyze not only the average number of changes, but also the average waiting time on stations (the more trains on a given edge, the shorter, on average, we have to wait). While this approach might be interesting and useful, this is quite far from the concept of traffic flows. In the space of stops, the notion of a traffic flow is also unclear. Does the "traffic" on a shortcut link *B*-*D* in Fig. 1(c) have any physical meaning? We know that this traffic actually traverses the links *B*-*C* and *C*-*D*, increasing their load and interfering with them. Ignoring this effect would give us a biased picture. In contrast, in the space of stations, the traffic flows have an unambiguous and natural interpretation. It is the exact route of a train in the graph representing the real physical infrastructure.

Another class of networks that can be constructed with the help of timetables are airport networks [6,25–27]. There, the nodes are the airports and edges are the flight connections. The weight of an edge reflects the traffic on this connection, which can be approximated by the number of flights that use it during 1 week. In this case, both the topology and traffic information are *explicitly* given by timetables. This is because the routes of planes are not constrained to any physical infrastructure, as opposed to roads for cars or rail tracks for trains. So there are no "real" links and "shortcut" links. In a sense all links are real and the topologies in the space of stops and space of stations actually coincide.

Inferring the space-of-stations topology from timetables becomes simple also in another special case, where the ve-

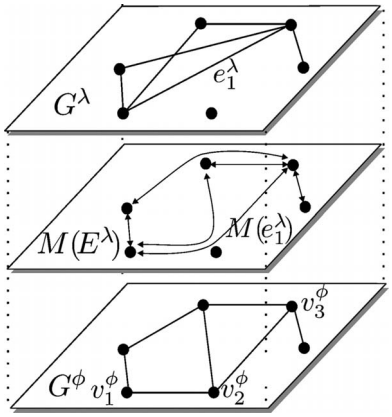


FIG. 2. An illustration of the two-layer model (see [21] for more details) with the actual mapping $M(E^\lambda)$ of the logical graph G^λ on the physical graph G^ϕ. The logical edge e_1^λ is mapped on G^ϕ as the path $M(e_1^\lambda)=(v_1^\phi, v_2^\phi, v_3^\phi)$.

hicles stop at each station they traverse (e.g., in many subway networks). This naturally eliminates the shortcuts, making the topologies in the space of stops and space of stations identical. This is not true in the general case, with both local and express vehicles.

In the remainder of this paper, we introduce necessary notation in Sec. IV. Next, in Sec. V we give an algorithm that extracts the real physical structure (a topology in the space of stations) and the network of traffic flows from timetables. In Sec. VI we test our algorithm on timetables of three large transportation networks at three different scales: city, country and continent. We also analyze the resulting physical topologies and compare them with those obtained by alternative approaches. Finally, in Sec. VII we conclude the paper.

IV. NOTATION

A. Two layers

We follow the *multilayer framework* introduced in Ref. [21]. An example with two layers is shown in Fig. 2. The lower-layer topology is called a *physical graph* $G^\phi=(V^\phi, E^\phi)$, and the upper-layer topology is called a *logical graph* $G^\lambda=(V^\lambda, E^\lambda)$. We assume that the sets of nodes at both layers are identical—i.e., $V^\phi \equiv V^\lambda$—but as a general rule, we keep the indices ϕ and λ to make the description unambiguous. Let $N=|V^\phi|=|V^\lambda|$ be the number of nodes. Every logical edge $e^\lambda=\{u^\lambda, v^\lambda\}$ is mapped on the physical graph as a path $M(e^\lambda)\subset G^\phi$ connecting the nodes u^ϕ and v^ϕ, corresponding to u^λ and v^λ. (A path is defined by the sequence of nodes it traverses.) The set of paths corresponding to all logical edges is called a *mapping* $M(E^\lambda)$ of the logical topology on the physical topology.

This simple multilayer framework was inspired by the well-established and highly specialized ISO/OSI network model in computer networking [28]. A similar layered architecture is also used to model economic systems [29]. It consists of a set of nodes (agents) connected by a number of various topologies, each defining a different layer. These layers are coupled by some interactions, but in contrast to our model, there is no clear hierarchy and mapping of the upper-layer edge as a path in the layer underneath.

In the field of transportation networks the undirected, unweighted physical graph G^ϕ captures the topology of the physical infrastructure (i.e., in the space of stations). In contrast, the weighted logical graph G^λ reflects the undirected traffic flows and is closely related to the concept of "traffic matrix" known in transportation science [30].[3] The logical topology is therefore (usually) very different from the physical one. Every logical edge e^λ is created by connecting the first and last nodes of the corresponding traffic flow (omitting the intermediate stations) and by assigning a weight $w(e^\lambda)$ that represents the intensity of this flow. The mapping $M(e^\lambda)$ of the edge e^λ is the path taken by this flow.

B. Timetable data

We take a list of all vehicles departing in the system within some period (e.g., one weekday). Denote by $R=\{r_i\}_{i=1,\ldots,|R|}$ the list of routes followed by these vehicles, where $|R|$ is the total number of vehicles. A route r_i of the ith vehicle is defined by the list of nodes it traverses. Note that since there are usually more vehicles (than one) following the same path on one day, some of the routes may be identical.

V. ALGORITHM

In this section we sketch our algorithm that extracts the real physical structure and the network of traffic flows from timetables. The details are described in the Appendix.

The algorithm consists of three phases. In the first one, initialization, based on the set of routes R, we create the set of nodes $V^\phi=V^\lambda$ and the physical topology $G^\phi_{stop}=(V^\phi, E^\phi_{stop})$ in the space of stops. In the second, the main phase, the sets R and E^ϕ_{stop} are iteratively refined by detecting and erasing the shortcut links in the physical graph G^ϕ_{stop}, resulting in the physical topology $G^\phi_{stat}=(V^\phi, E^\phi_{stat})$ in the space of stations. Finally, in the third phase, we group the vehicles with identical routes and obtain the weighted logical graph G^λ and the mapping $M(E^\lambda)$ of the logical edges on the physical graph G^ϕ_{stat}.

VI. STUDY OF THREE REAL-LIFE NETWORKS

In this section we apply our algorithm to extract the data from the timetables of three examples of transportation networks, with sizes ranging from city to continent. As an example of a city, we take the mass transportation system (buses, trams, and metros) of Warsaw (WA), Poland; its timetables are available at Ref. [31]. At a country level, we study the railway network of Switzerland (CH). Finally, we investigate the railway network formed by major trains and

[3]The logical graph and the traffic matrix are not the same objects. The traffic matrix reflects the people's demands, whereas the logical topology is the result of an an optimization process taking into account many factors, such as continuity of the path, traveling times, availability of stock, and, of course, the traffic matrix.

TABLE I. The studied data sets. "Area" is the surface occupied by the region covered by the network. N is the number of nodes (stations and stops). $|R|$ is the total number of vehicles departing in the network during one weekday. $|E^{\lambda}|$ is the number of edges in the logical graph (number traffic flows); it is much smaller than $|R|$, because the vehicles following the same route are grouped together in phase 3 of our algorithm. All the remaining parameters are computed for the physical graphs G^{ϕ}: $|E^{\phi}|$ is the number of edges, $\langle k^{\phi}\rangle$ is the average node degree, d^{ϕ} stands for the diameter, $\langle l^{\phi}\rangle$ is the average shortest path length, and c^{ϕ} is the clustering coefficient.

General			Traffic		Physical graph					
Dataset	Area [km^2]	N	$\|R\|$	$\|E^{\lambda}\|$	Space	$\|E^{\phi}\|$	$\langle k^{\phi}\rangle$	d^{ϕ}	$\langle l^{\phi}\rangle$	c^{ϕ}
WA (Warsaw)					changes	78437	102.3	4	2.3	0.6829
	480	1533	25'995	221	stops	2249	2.9	76	19.0	0.1681
					stations	1832	2.4	90	28.1	0.0092
CH (Switzerland)					changes	19827	24.6	8	3.6	0.9095
	41'300	1613	6'957	505	stops	1922	2.4	61	16.3	0.0949
					stations	1680	2.1	136	46.6	0.0004
EU (Europe)					changes	88329	36.4	8	3.7	0.7347
	2'081'000	4853	60'775	6703	stops	8600	3.5	48	12.6	0.3401
					stations	5765	2.4	184	50.9	0.0129

stations in most countries of central Europe (EU).[4] The timetables of both CH and EU networks are available at Ref. [32]. The basic parameters of the data sets and of the resulting graphs can be found in Table I.

This section is organized as follows. First, we focus on a particular data set in order to study the performance of our algorithm. Next, we analyze and compare the physical graphs originating from all three data sets in each of the considered spaces. Finally, we focus our attention on the logical graphs and traffic flows extracted by our algorithm.

A. Example: The railway network of Switzerland

As an illustration, let us consider more closely the railway network of Switzerland. According to our timetable, on a typical weekday there are $|R|=6957$ different trains that follow $|E^{\lambda}|=505$ different routes (usually there is more than one train following the same route during one day). Our data contains $N=1613$ stations in Switzerland, together with their physical coordinates. In Fig. 3 we present the graphs obtained from this data set. The physical graphs in the three spaces are shown in Figs. 3(a)–3(c). The graph in the space of stations was obtained with the help of the algorithm introduced in the previous section. The number of vertices is the same in all three spaces. The number of edges in the space of changes, $|E^{\phi}_{change}|=19\,827$, is much larger than in the other two spaces. Although at first sight the physical graphs in the space of stations and space of stops look comparable, the latter has a number of (nonexisting in reality) shortcut links. For a visual verification of correctness of our algorithm, we show in Fig. 3(d) the real map of the Swiss railway system; we observe only minor differences between (c) and (d). Finally, in Fig. 3(e), we present the logical graph that reflects the traffic flows in the network. This graph is very heterogenous both in the weights of edges and in the layout of traffic.

[4]In the EU data set, Paris has originally several stations that are not directly connected between each other. Following the approach in Ref. [4], we merged them into one common node.

B. Physical graph in three spaces

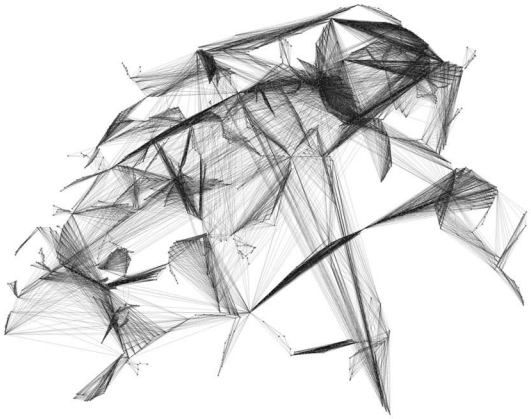
How does the choice of space affect the topology? We study in this section the physical graphs in the three spaces with respect to the basic metrics often used in the analysis of complex networks.

1. Diameter d^{ϕ} and average shortest path length $\langle l^{\phi}\rangle$

The average shortest path length $\langle l\rangle$ is computed over the lengths of shortest paths between all pairs of vertices. The diameter d is the longest of all shortest path lengths. These parameters are usually closely related.

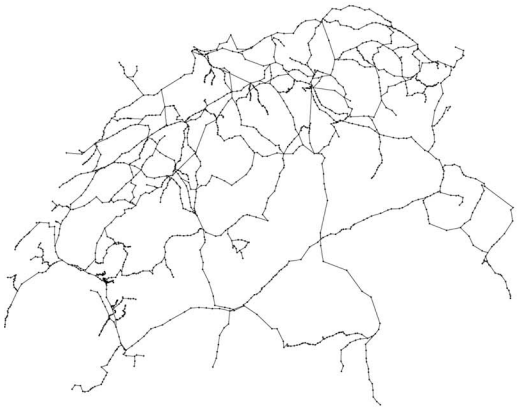
The diameter d^{ϕ} and the average shortest path length $\langle l^{\phi}\rangle$ of the graphs in the space of stations are large (see Table I). Moreover, $\langle l^{\phi}\rangle$ scales roughly as $\sqrt{N}$ with the number of nodes N [e.g., $\langle l^{\phi}\rangle \sim x^{0.45}$ for the EU data set—see Fig. 4(a)]. This behavior is typical of many planar, latticelike infrastructure networks embedded in a two-dimensional space.

The graphs in the space of stops have about 10%–15% more edges than their counterparts in the space of stations. The difference is not large, and one could possibly expect similar values and scaling of the diameter and the average shortest path length. However, these 10%–15% edges are fundamentally different from typical edges in the space of stations; they are shortcut links. It was shown in Ref. [33] that the diameter of a graph is very sensitive to the existence of shortcuts. Even a relatively small number of shortcuts can dramatically bring down the diameter and the average shortest path length. We observe this phenomenon in our graphs. For instance, in the EU data set, the diameter drops about 4 times, from $d^{\phi}=184$ in the space of stations to 48 in the space of stops. Similarly, the average shortest path length drops by roughly the same factor. Moreover, the scaling of $\langle l^{\phi}\rangle$ in N is no longer $\sqrt{N}$, but logarithmic. For instance, for

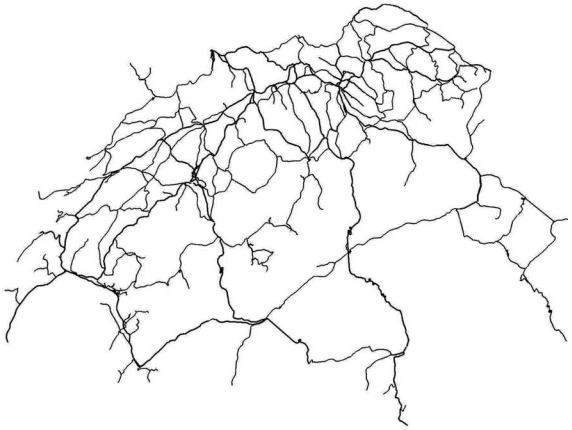


(a) Physical graph G^{ϕ}_{change} in space–of–changes

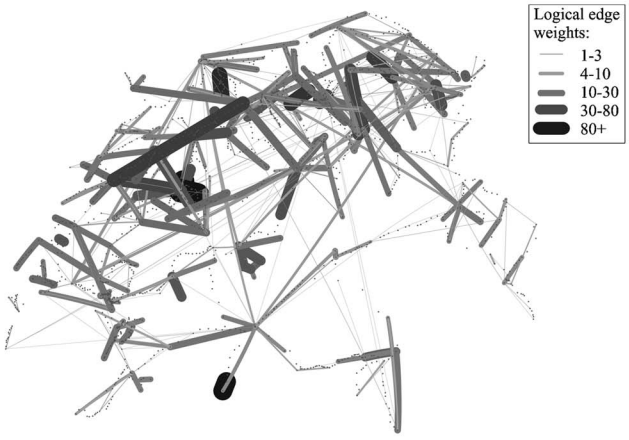
(b) Physical graph G^{ϕ}_{stop} in space–of–stops



(c) Physical graph G^{ϕ}_{stat} in space–of–stations

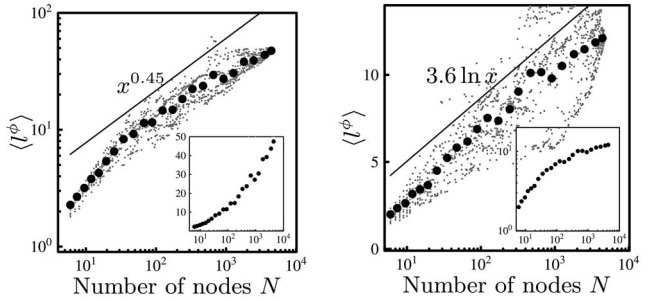


(d) Real physical map



(e) Logical graph

FIG. 3. The railway network in Switzerland (CH). (a), (b), (c) Physical graphs in the space of changes, stops, and stations, respectively. (d) The real map of the rail tracks in Switzerland. (e) The logical graph. Every edge connects the first and last stations of a particular train route; its weight reflects the number of trains following this route in any direction.



(a) space–of–stations (b) space–of–stops

FIG. 4. The scaling of the average shortest path length $\langle l^\phi \rangle$ with the number of nodes N for the EU data set. The physical graph was randomly partitioned with multiple horizontal and vertical cuts into a number of smaller components. For these components we plot scatter plots of $\langle l^\phi \rangle$ vs N (little gray dots in the background), as well as their binned values (large black disks). The space-of-stations plot (a) is made in a log-log scale with a log-linear inset of the same distribution for comparison; the space-of-stops plot (b) is made in a log-linear scale with a log-log inset. In both plots we draw the linear best fit in the respective log-log and log-lin plots.

the EU data set we have $\langle l^\phi \rangle \sim 3.6 \ln x$ [see Fig. 4(b)]. This type of scaling is typical of random graphs and small worlds [33]. Therefore, the shortcut edges, although not very numerous, play a very important role and make the graphs in the space of stops very different from those in the space of stations. [This effect is not so strongly pronounced in the WA data set. The underlying reason is the relatively short length of shortcuts (usually two hops), which was shown to affect the diameter only to a small extent [34].]

Finally, the graphs in the space of changes have very small diameters and average shortest path lengths. This is mainly because of their high density (number of edges).

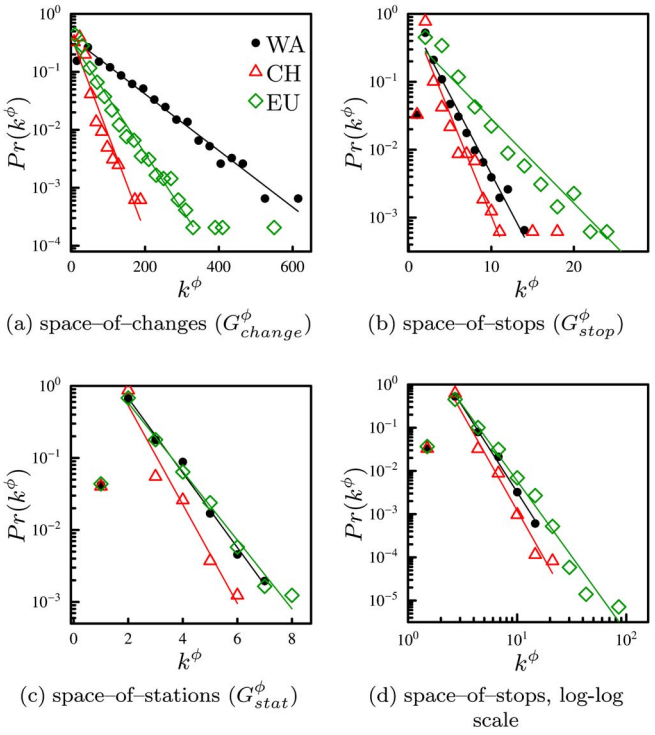
2. Node degree k

The node degree distributions in all three spaces are plotted in a semilogarithmic scale in Figs. 5(a)–5(c). Additionally, for the space of stops, we plot the degree distributions in a log-log scale [Fig. 5(d)], because it is not obvious which fit is better, exponential or power law (it was also pointed out in [13]). For the other two spaces we observe a clear linear trend indicating the exponential behavior. This was expected in the space of stations, because the degree distribution of many infrastructure networks was shown to be narrow (here one decade) and to decay exponentially (see, e.g., power lines in [35]). In the space of stations the vast majority of nodes have degree equal to 2, indicating long segments of stations without junctions.

3. Clustering coefficients c

We have studied the clustering coefficients c defined as a probability that two randomly chosen neighbors of a node are also direct neighbors of each other [33].

The clustering coefficients of topologies in the space of changes are very high, which is a direct consequence of a very high density and the existence of many cliques. What is



(a) space–of–changes (G^ϕ_{change}) (b) space–of–stops (G^ϕ_{stop})

(c) space–of–stations (G^ϕ_{stat}) (d) space–of–stops, log-log scale

FIG. 5. (Color online) Node degree distributions in physical graphs in the three spaces, for the data sets WA, CH, and EU. Plots (a)–(c) use a semilogarithmic scale, plot (d) uses a log-log scale. If necessary, the data are lin-binned or log-binned, accordingly.

more interesting is that in all three data sets, the clustering coefficient in the space of stops is one to two orders of magnitude larger than in the space of stations. As in the case of the graph diameter, here again the shortcut links turn out play a very important role in the topology.

C. Traffic flows and the logical graph

Now we turn our attention to the traffic that flows in our networks. We extracted this scarce data with the help of the algorithm introduced in this paper. As we argued in Sec. III, the interpretation of traffic flowing through networks in the space of changes and space of stops is rather cumbersome. Therefore we restrict our analysis to the traffic flows traversing the physical graph in the space of stations.

In Fig. 6 we compare the lengths of traffic flows before and after application of our algorithm. A new traffic flow can be either equal in length to the original one (if no shortcut was detected on its path) or longer. We observe that for all three data sets, there are a significant number of flows that become longer. In some cases this increase in length is by as much as 10 times. Generally, the longer the original flow is, the less extended it gets during a run of our algorithm. This is expected, because a long flow in a timetable usually corresponds to a local train that stops at all stations (i.e., uses no shortcuts).

In Fig. 7 we present basic distributions measured for logical graphs in the three data sets. Recall that the edges in a logical graph reflect the traffic flows. Therefore, the node

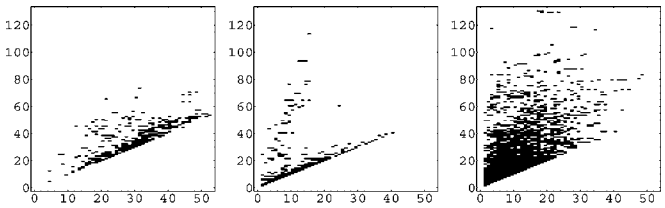


FIG. 6. The lengths of original timetable routes (x axis) versus these lengths after the application of our algorithm (y axis). All three data sets are drawn in the same scale.

degree k^λ is the number of *different* connections starting or ending at the corresponding station [Fig. 7(a)]. The strength s^λ of a node is the sum of the weights of neighboring edges [25]; here, it is the number of *all* connections starting or ending at this station [Fig. 7(b)]. Finally, the weight $w(e^\lambda)$ of a logical edge is the traffic flow intensity [Fig. 7(c)].

All three distributions are heavily right skewed, meaning that there is a small number of nodes and edges with very high values of the observed parameter. We conclude that the real-life traffic patterns are very heterogenous, in both space (node degree and strength) and traffic flow intensities. This was shown in Ref. [21] to be the reason for the high unpredictability of the load distribution in transportation networks.

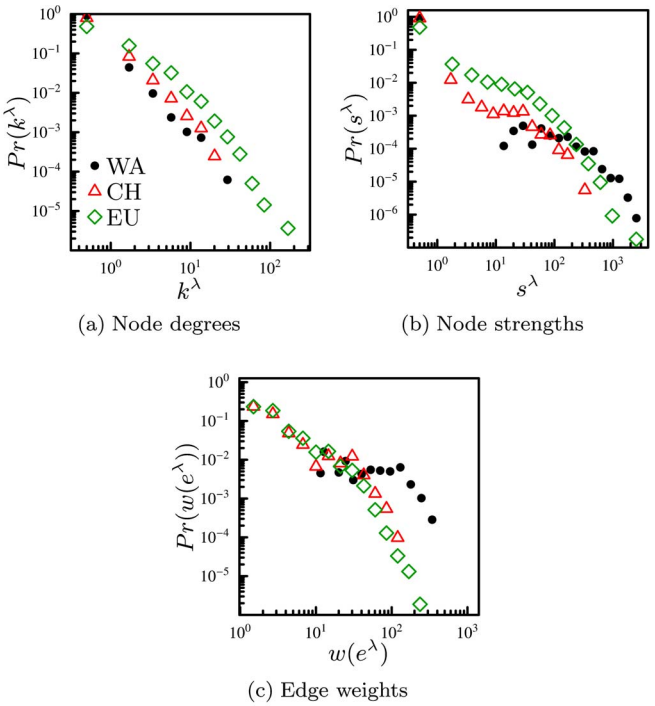


FIG. 7. (Color online) Properties of logical graphs for the data sets WA, CH, and EU. (a) Node degree distribution. Many nodes are isolated—they represent intermediate stations on which no train starts or terminates its journey. The isolated nodes we represent here as having "degree" equal to 0.1. (b) Node strength distribution. (c) Edge weight (traffic flow intensities) distribution. All data are log-binned and plotted in a log-log scale.

D. Node load and its estimators

Our algorithm extracts from timetables not only the real physical and logical topologies, but also their actual mapping—i.e., the exact routes of the traffic flows. This allows us to compute the real load $l(v)$ of node v, naturally defined as the sum of the weights of all traffic flows traversing v [21]. As this information is often missing in many existing data sets, many attempts were developed to approximate the node load with some topological metrics. In this section we test on our data sets the performance of four different approaches.

Our first load estimator is *node degree* k^ϕ. It seems natural that the nodes with high degree carry more traffic than the less connected nodes.

Our second metric is *betweenness* b^ϕ [36]. The betweenness of a vertex v is the fraction of shortest paths between all pairs of vertices in a network that pass through v. If there is more than one shortest path between a given pair of vertices, then all such paths are taken into account with equal weights summing to 1. As betweenness aims at capturing the amount of information passing through a vertex, it is often taken directly as a measure of the load [15–20].

Our third load estimator is inspired by the approach in Ref. [2]. In this analysis of a U.S. power grid, the authors know not only the network topology, but also the set of all electricity generators. This additional information is used to estimate the load by constructing the shortest paths of equal weight from all sources (generators) to all other nodes in the graph (power consumers). Similarly, in the context of railway networks, we can identify the set of all traffic sources (destinations) as the set of all first (last) stations of all trains. Next, we generate unweighted shortest paths from every source to every destination. The number of these paths traversing a given node v is called the *restricted betweenness* b_r^ϕ of v.

Our last load estimator uses a more detailed knowledge of a real traffic pattern. Instead of generating traffic from each source to all destinations, we identify the actual destination(s) of traffic originating from every source and construct the unweighted shortest paths only between these source-destination pairs. This metric is very similar to the node load l, except that now we assume that all traffic flows have the same intensity (weight). Therefore, we call this load estimator the *simple load* l_s.

In Fig. 8 we present scatter plots of the four load estimators versus the real load l, separately for WA, CH, and EU. The value of the corresponding Pearson's correlation coefficient is shown in the top left corner of every plot. As we pointed out in Ref. [21], in our data set the node degree approximates the real load better than the betweenness (its Pearson's coefficient is higher). Nevertheless, both of them are very far from being satisfactory. So large disparities may strongly affect the results of the network performance analysis based on the topological load estimators. Surprisingly, knowledge of the sets of all traffic sources and destinations (but not source-destination pairs) does not help—the restricted betweenness b_r^ϕ gives roughly the same results as the

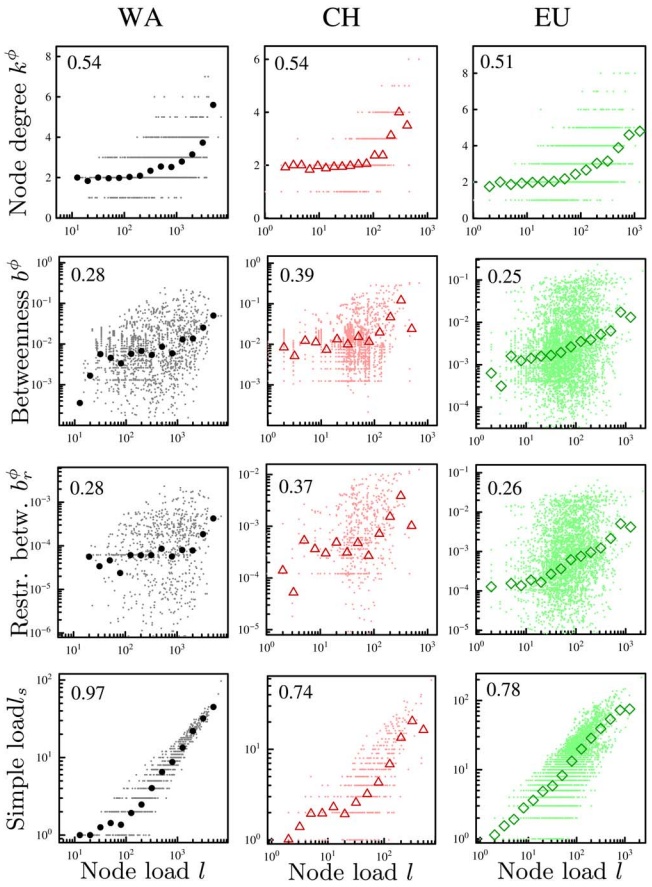


FIG. 8. (Color online) The scatter plots of the node degree d^ϕ (top), betweenness b^ϕ (upper middle), restricted betweenness b_r^ϕ (lower middle), and simple load l_s (bottom) versus the real load l. Each column corresponds to a different data set: WA (left), CH (middle), and EU (right). The average log-binned values are set in bold in every plot. In the top left corner of every plot we give the value of the corresponding Pearson's correlation coefficient.

"pure" betweenness b^ϕ. In contrast, the simple load l_s estimator performed much better, especially for the WA data set.

We conclude that in order to have at least a rough approximation of the real load, we need some additional knowledge on the traffic pattern. Knowing only the sets of traffic sources and destinations is clearly not enough in our data sets, but knowledge of all source-destination *pairs* (equivalent to the logical unweighted topology) might be sufficient in some applications.

Finally, we also tried weighted combinations of the four load estimators. Would the combined knowledge of betweenness and node degree reveal some hidden correlation between the two, resulting in a more accurate real load estimator? Unfortunately, all such attempts only degrade the performance of the best involved estimator.

VII. CONCLUSIONS

The knowledge of real-life traffic patterns is crucial in the analysis of transportation systems. These data are usually much more difficult to get than the pure topology of a network. In this paper we have proposed an algorithm for extracting both the physical topology and network of traffic flows from timetables of public mass transportation systems. We have applied our algorithm to three large transportation networks. This enabled us to make a systematic comparison between three different approaches (or "spaces") to construct a graph representation of a transportation network. The resulting physical topologies are very different. In particular, the seemingly similar graphs in the space of stops and space of stations turn out to be very different in terms of basic graph-theory metrics such as diameter, average shortest path length, clustering coefficient and node degree distribution. This is due to the existence of shortcut links in the space of stops. Our algorithm detects and eliminates these shortcuts and extracts the topology in the space of stations. Only this graph reflects the real-life physical infrastructure that is used by the traffic flows, becomes congested, or can be prone to failures or susceptible to attacks. In contrast, the edges in the space of changes and space of stops are somewhat "virtual" and the notion of traffic in these graphs is unclear, if it at all makes any sense. What is important is that the results are consistent across three different scales of the studied networks (city, country, and continent).

We have also tested different approaches to estimate the real load in the network. In our data sets, the purely topological metrics, such as node degree and betweenness, give very bad approximates. Surprisingly, knowledge of all traffic sources and destinations does not improve the situation. The minimum information we need to obtain at least a rough approximation of the real load is the (unweighted) set of all source-destination pairs.

This work has several possible directions for the future. For instance, knowledge of real traffic patterns allows us to reexamine the error and attack tolerance [37] of transportation systems, which might look completely different when focusing on traffic instead of topology. Another direction would be to exploit the additional information available in some timetables. For instance, in our data sets CH and EU, we also know the geographical coordinates of the nodes. They fall therefore in the category of *spatial networks* that have been recently intensively studied [4,6,9,38–40]. In particular, we think that incorporating real traffic patterns in the models can help to understand the processes that govern the evolution of spatial networks.

Finally, the data are available at Ref. [41].

ACKNOWLEDGMENT

The work presented in this paper was financially supported by Grant No. DICS 1830 of the Hasler Foundation, Bern, Switzerland.

APPENDIX

In this appendix we give the details of our algorithm to extract the physical and logical network data from timetables. It consists of three phases, as follows.

Phase 1: Initialization

In this phase we interpret every two consecutive nodes in any route $r_i \in R$ as directly connected. Consequently, we connect these nodes with a link, which can be written as

$$E^{\phi}_{stop} = \bigcup_{i=1,\ldots,|R|} E(r_i),$$

where $E(r_i)$ is the set of all pairs of adjacent nodes in r_i (i.e., all edges in r_i). This results in the physical topology $G^{\phi}_{stop} = (V^{\phi}, E^{\phi}_{stop})$ in the space of stops.

Phase 2: Deleting shortcuts

In this phase, at each iteration, we detect a shortcut in the set of physical edges, delete it, and update all routes r_i that use this shortcut. Denote by $e^{\phi}_{(1)}$ and $e^{\phi}_{(2)}$ the two end nodes of e^{ϕ} and by $\mathrm{Rev}(P_{e^{\phi}})$ the reversed version of $P_{e^{\phi}}$ (the sequence from the last node to the first one). The algorithm is as follows:

(1) $E^{\phi}_{stat} := E^{\phi}_{stop}$
(2) Find a tuple (e^{ϕ}, r_i) such that e^{ϕ} is a shortcut for r_i: $e^{\phi}_{(1)} \in r_i$ and $e^{\phi}_{(2)} \in r_i$ and $e^{\phi} \notin E(r_i)$.
(3) IF no (e^{ϕ}, r_i) found THEN RETURN E^{ϕ}_{stat} and R.
(4) $P_{e^{\phi}} :=$ subpath of r_i from $e^{\phi}_{(1)}$ to $e^{\phi}_{(2)}$
(5) FOR all $r_j \in R$ DO:
  (a) If $(e^{\phi}_{(1)}, e^{\phi}_{(2)}) \in r_j$ THEN replace it with $P_{e^{\phi}}$
  (b) If $(e^{\phi}_{(2)}, e^{\phi}_{(1)}) \in r_j$ THEN replace it with $\mathrm{Rev}(P_{e^{\phi}})$
(6) $E^{\phi}_{stat} := E^{\phi}_{stat} \backslash \{e^{\phi}\}$
(7) GOTO 2

In step 2, we look for a physical link that is a shortcut. We declare a physical link e^{ϕ} to be a shortcut if there exists a route $r_i \in R$ such that e^{ϕ} connects two *nonconsecutive* nodes in r_i. For example, in Fig. 1(c), $e^{\phi} = \{B, D\}$ is a shortcut because it connects two not neighboring nodes in the route r_1 of line 1. If no physical edge can be declared a shortcut, the algorithm quits in step 3, returning E^{ϕ}_{stat} and R. Otherwise, in step 4, we find the path $P_{e^{\phi}}$ that this shortcut should take. In Fig. 1(c) this path is $P_{e^{\phi}} = (B, C, D)$. In step 5, we update the set of routes R by replacing every shortcut link e^{ϕ} in every route using it with the corresponding path $P_{e^{\phi}}$. In our example, the updated route of line 2 becomes $r_2 = (A, B, C, D, E)$. It is thus identical to the route of line 1. Finally, in step 6 we delete the shortcut e^{ϕ} from the physical graph. We iterate these steps until no shortcut is found (step 2). The resulting physical graph $G^{\phi}_{stat} = (V^{\phi}, E^{\phi}_{stat}) \subset G^{\phi}_{stop}$, is a graph in the space of stations.

Phase 3: Grouping the same routes together

Finally, based on the list R of routes updated in phase 2, we find groups of vehicles that follow the same path (in any direction). Each such group defines one edge e^{λ} in the logical graph; e^{λ} connects the first and last nodes of the route, omitting all the intermediate stations. The number of vehicles that follow this route becomes the weight $w(e^{\lambda})$ of the logical edge e^{λ}; the route itself becomes the mapping $M(e^{\lambda})$ of e^{λ} on the physical graph.

Denote by $r_{i(first)}, r_{i(last)}$ the first and last nodes in r_i and by $E(M(e^{\lambda}))$ the set of all physical edges in the mapping of e^{λ}. Now, phase 3 can be stated as follows:

(1) $E^{\lambda} = \emptyset$, $M = \emptyset$
(2) FOR $i = 1$ TO $|R|$ DO:
  (a) $e^{\lambda}_i = \{r_{i(first)}, r_{i(last)}\}$
  (b) IF $e^{\lambda}_i \in E^{\lambda}$ THEN $w(e^{\lambda}_i) := w(e^{\lambda}_i) + 1$
  ELSE $E^{\lambda} = E^{\lambda} \cup \{e^{\lambda}_i\}$, $M(e^{\lambda}_i) = r_i$, $w(e^{\lambda}_i) = 1$
(3) $E^{\phi}_{stat} = \cup_{e^{\lambda} \in E^{\lambda}} E(M(e^{\lambda}))$

In the example in Fig. 1, after phase 2 the routes of lines 1 and 2 become identical; therefore, in phase 3 they are grouped together defining a logical edge $e^{\lambda}_1 = \{A, E\}$ with the weight $w(e^{\lambda}_1) = 2$ and the mapping $M(e^{\lambda}_1) = (A, B, C, D, E)$. A second logical edge is $e^{\lambda}_2 = \{F, H\}$ with $w(e^{\lambda}_2) = 1$ and $M(e^{\lambda}_2) = (F, B, G, H)$.

Accuracy of the algorithm

There are potential sources of mistakes and inaccuracies in our approach. First, the links that we delete as being shortcuts might actually exist in reality. However, a comparison of the results of our algorithm with the real maps (see Sec. VI) reveals very few differences, which means that this source of failures occurs very rarely in real data sets.

A second problem lies in the estimation of the traffic pattern. Interpreting the routes of trains, buses, trams, metros, etc., as traffic flows gives us a picture at a low level of granularity. We view every vehicle as a traffic unit, regardless of its size or the number of people it carries. Moreover, people usually use these vehicles only on a portion of their total journey, not from the first to the last station. Clearly, the vehicle routes are the result of an optimization process that take into account many factors, such as people's demand, continuity of the path, traveling times, and availability of stock. However, we believe that they reflect well the general direction and intensity of travels, and we take a vehicle as a basic traffic unit. After all, these are the vehicles that appear on the roads and cause traffic, not the people they transport.

[1] D. J. Watts and S. H. Strogatz, Nature (London) **393**, 440 (1998).
[2] Reka Albert, Istvan Albert, and Gary L. Nakarado, Phys. Rev. E **69**, 025103(R) (2004).
[3] Parongama Sen, Subinay Dasgupta, Arnab Chatterjee, P. A. Sreeram, G. Mukherjee, and S. S. Manna, Phys. Rev. E **67**, 036106 (2003).
[4] Michael T. Gastner and M. E. J. Newman, J. Stat. Mech.: Theory Exp. 2006 P01015.
[5] Sergio Porta, Paolo Crucitti, and Vito Latora, Physica A **369**, 853 (2006).
[6] Michael T. Gastner and M. E. J. Newman, Eur. Phys. J. B **49**,

247 (2006).
[7] M. Rosvall, A. Trusina, P. Minnhagen, and K. Sneppen, Phys. Rev. Lett. **94**, 028701 (2005).
[8] Sergio Porta, Paolo Crucitti, and Vito Latora, Environmental Planning B (to be published), e-print physics/0506009.
[9] Alessio Cardillo, Salvatore Scellato, Vito Latora, and Sergio Porta, Phys. Rev. E **73**, 066107 (2006).
[10] V. Latora and M. Marchiori, Phys. Rev. Lett. **87**, 198701 (2001).
[11] V. Latora and M. Marchiori, Physica A **314**, 109 (2002).
[12] Katherine A. Seaton and Lisa M. Hackett, Physica A **339**, 635 (2004).
[13] J. Sienkiewicz and J. A. Hołyst, Phys. Rev. E **72**, 046127 (2005).
[14] I. Vragović, E. Louis, and A. Diaz-Guilera, Phys. Rev. E **71**, 036122 (2005).
[15] K.-I. Goh, B. Kahng, and D. Kim, Phys. Rev. Lett. **87**, 278701 (2001).
[16] G. Szabó, M. Alava, and J. Kertész, Phys. Rev. E **66**, 026101 (2002).
[17] B. Bollobás and O. Riordan, Phys. Rev. E **69**, 036114 (2004).
[18] P. Holme and Beom Jun Kim, Phys. Rev. E **65**, 066109 (2002).
[19] L. Zhao, K. Park, and Y.-C. Lai, Phys. Rev. E **70**, 035101(R) (2004).
[20] Adilson E. Motter, Phys. Rev. Lett. **93**, 098701 (2004).
[21] M. Kurant and P. Thiran, Phys. Rev. Lett. **96**, 138701 (2006).
[22] G. Chowell, J. M. Hyman, S. Eubank, and C. Castillo-Chavez, Phys. Rev. E **68**, 066102 (2003).
[23] Andrea De Montis, Marc Barthélemy, Alessandro Chessa, and Alessandro Vespignani, e-print physics/0507106.
[24] Annegret Liebers, Ph.D. thesis, University of Konstanz, 2001.
[25] A. Barrat, M. Barthélemy, R. Pastor-Satorras, and A. Vespignani, Proc. Natl. Acad. Sci. U.S.A. **101**, 3747 (2004).
[26] L. Hufnagel, D. Brockmann, and T. Geisel, Proc. Natl. Acad. Sci. U.S.A. **101**, 15124 (2004).
[27] R. Guimerà, S. Mossa, A. Turtschi, and L. A. N. Amaral, Proc. Natl. Acad. Sci. U.S.A. **102**, 7794 (2005).
[28] James F. Kurose and Keith W. Ross, *Computer Networking* (Pearson Addison Wesley, Boston, 2003).
[29] Tom Erez, Martin Hohnisch, and Sorin Solomon, in *Economics: Complex Windows*, edited by M. Salzano and A. Kiman (Springer, Berlin, 2005), p. 201, e-print cond-mat/0406369.
[30] William R. Black, *Transportation. A Geographical Analysis* (Guilford Press, New York, 2003).
[31] http://www.ztm.waw.pl
[32] http://www.sbb.ch
[33] D. J. Watts, *Small Worlds* (Princton University Press, Princeton, 1999).
[34] Jon Kleinberg, in Proceedings of 32nd ACM Symposium on Theory of Computing, 2000. Also appears as Cornell Computer Science Technical Report 99-1776 (Oct. 1999), http://www.cs.cornell.edu/home/kleinber/swn.d/swn.html.
[35] Steven H. Strogatz, Nature (London) **410**, 268 (2001).
[36] L. C. Freeman, Sociometry **40**, 35 (1977).
[37] R. Albert, H. Jeong, and A.-L. Barabsi, Nature (London) **406**, 378 (2000).
[38] Naoki Masuda, Hiroyoshi Miwa, and Norio Konno, Phys. Rev. E **71**, 036108 (2005).
[39] Thomas Petermann and Paolo De Los Rios, e-print cond-mat/0501420.
[40] Alain Barrat, Marc Barthelemy, and Alessandro Vespignani, J. Stat. Mech.: Theory Exp., P05003 (2005).
[41] http://icawww.epfl.ch/kurant/
[42] Vamsi Kalapala, Vishal Sanwalani, Aaron Clauset, and Cristopher Moore, e-print physics/0510198, p. 2005.