EUROGRAPHICS 2010 / G. Melançon, T. Munzner, and D. Weiskopf
(Guest Editors)

Volume 29 (2010), Number 3

GraphDice: A System for Exploring Multivariate Social Networks

A. Bezerianos[1], F. Chevalier[2], P. Dragicevic[2], N. Elmqvist[3], and J.D. Fekete[2]

[1]École Centrale Paris, France [2]INRIA Saclay - Île-de-France, France [3]Purdue University, USA

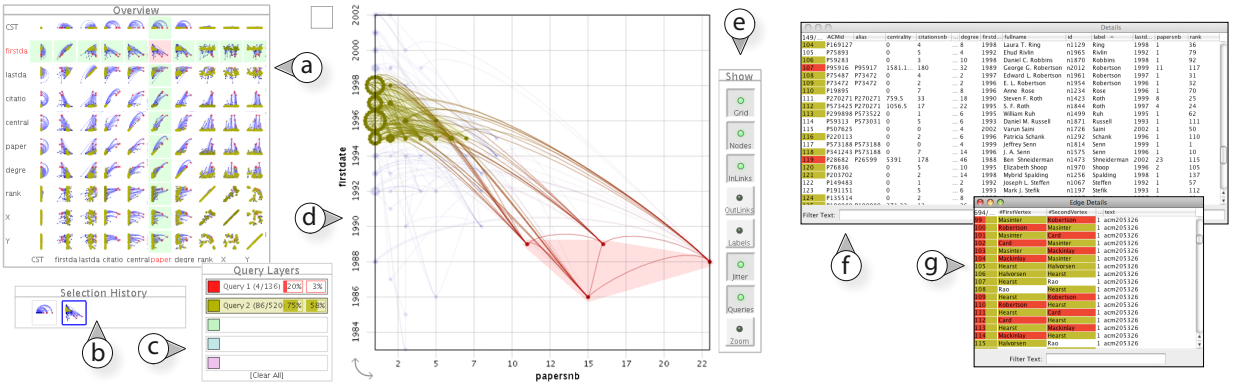

Figure 1: *Exploration of the InfoVis 2004 Contest co-authorship dataset using GraphDice. On the left is the main visualization window of GraphDice including (a) an overview plot matrix, (b) a selection history tool, (c) a selection query window, (d) a main plot, and (e) a toolbar. Overlapping nodes in the main plot are drawn using jitter (visible in the yellow selection query). On the right are actor (f) and link (g) tables with query data entries highlighted in the corresponding color.*

Abstract

Social networks collected by historians or sociologists typically have a large number of actors and edge attributes. Applying social network analysis (SNA) algorithms to these networks produces additional attributes such as degree, centrality, and clustering coefficients. Understanding the effects of this plethora of attributes is one of the main challenges of multivariate SNA. We present the design of GraphDice, a multivariate network visualization system for exploring the attribute space of edges and actors. GraphDice builds upon the ScatterDice system for its main multidimensional navigation paradigm, and extends it with novel mechanisms to support network exploration in general and SNA tasks in particular. Novel mechanisms include visualization of attributes of interval type and projection of numerical edge attributes to node attributes. We show how these extensions to the original ScatterDice system allow to support complex visual analysis tasks on networks with hundreds of actors and up to 30 attributes, while providing a simple and consistent interface for interacting with network data.

Categories and Subject Descriptors (according to ACM CCS): H.5.1 [Information Systems]: Information Interfaces and Presentation—User Interfaces; E.1 [Data]: Data Structures—Graphs and Networks

1. Introduction

A quite recent development in social network analysis (SNA) [WF94] has been the adoption of visualization to explore networks and support social scientists in detecting, understanding, and characterizing unexpected patterns and trends in complex social networks [Ada06, HF06, HF07]. However, with a few exceptions (notably [AS07, PvW08, Wat06]), current state-of-the-art social network visualization tools focus on displaying the topology of the networks, and fail to provide a convenient way of explicitly visualizing more than a few (two to three) attributes associated with the network entities, usually using color and shapes. In contrast,

© 2010 The Author(s)
Journal compilation © 2010 The Eurographics Association and Blackwell Publishing Ltd.
Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

real social networks collected and built by sociologists and historians are rich in attributes for both actors (graph nodes) and relations between actors (graph edges). To compound the problem, the standard network analysis algorithms for computing degree, centrality, and graph layout that these domain experts routinely use create additional node and edge attributes. The process of analyzing such networks is called *multivariate social network analysis*.

To address the dearth of multivariate network visualization tools, we present GraphDice, a tool based on the ScatterDice [EDF08] dimensional navigation method, suitably adapted for network visualization. GraphDice (Fig. 1) is the first tool to use a plot matrix to navigate multivariate graphs. It is designed to be simple to learn and use by social science researchers, while providing very rich navigation and exploration capabilities. Further contributions of this article include the unified visualizations of edge and node attributes, by supporting interval values on the nodes and projecting link attributes on the nodes as intervals (Fig. 2), as well as interesting visual configurations provided through computed attributes to unify the navigation.

In this paper we first discuss previous work on multivariate social network analysis and visualization. We then describe a set of design goals followed during the design of GraphDice, followed by an overview of the system. Finally, we present a number of real datasets visualized by our tool and discuss results from a user feedback session.

2. Background

In this section we review work on social network and general multidimensional visualization, including for graphs.

2.1. Social Network Analysis (SNA)

A *social network* is a graph structure whose vertices are social entities such as persons or organizations, and edges are relations between these entities. *Social network analysis* (SNA) [WF94], correspondingly, consists of finding structural properties in the network, mostly through statistical and graph-theoretical methods. Social networks typically associate data to vertices and edges. For example, if vertices are persons, they can have a name, a birth date, a position in a company, and many more attributes. Similarly, relations can also have attributes, such as date of friendship connection.

Just like statistics, SNA can be confirmatory or exploratory. Most books on SNA are devoted to confirmatory methods, where there is an initial hypothesis about the structure of the network, and several methods are described for how to confirm this hypothesis. However, exploratory SNA is becoming an increasing opportunity due to the availability of vast amounts of data, either from online social networks, like Facebook, or constructed from social sites, such as Wikipedia or Flickr. Moreover, organizations increasingly maintain information about their members in databases that are easy to explore as social networks by focusing on any relation such as the organizational chart, project history, or email exchanges. These social networks are very rich, and besides confirming a-priori hypotheses, exploring these networks often reveals interesting and unexpected phenomena and structures. This ready availability of data also means that more people with no formal training in social network analysis have access to networks which are meaningful to them, such as relationships between cinema actors, relationships between their family and friends.

However, most established tools for analyzing social networks, such as UCINET [BEF09], are intended for confirmatory analysis, and are therefore weak at exploration and require substantial training. The International Network for Social Network Analysis [INS] lists about 60 software tools that fall into three categories: confirmatory analysis, exploratory analysis and network visualization. The popular Pajek graph visualization system can be used for exploration [dNMB05], but not for interactive information visualization because each exploration step is very indirect, involving long menus, several sub-menus, or large dialog boxes, once again requiring special training.

Computed attributes play an important role in SNA, like centrality or roles (e.g. hub, gate-keeper). Graph-theoretical properties, such as degree of a vertex (in-degree and out-degree for directed graphs) or clustering coefficients, also provide important information about the status of actors in the network. However, most measures have sometimes numerous variants, (e.g. centrality has tens of variants that, depending on the network structure, represent more or less faithfully the intuitive notion of centrality). Therefore, even when analysts want a measure of centrality, they often compute many of them and check — statistically or visually — which best fits their intuition about the reality of the world.

In practice, standard SNA consists of computing useful measures on the network and finding structures based on the topology, the intrinsic attributes and these measures. This is why most analysts — whether professional or casual — often need to explore multivariate networks with typically 10 to 30 measures per actor or relation.

2.2. Tasks

There has recently been considerable work on enumerating important tasks for social network analysis and exploration. Lee et al. [LPP*06] present a general task taxonomy of: low-level tasks, topology-based tasks, attribute-based tasks, browsing tasks, and overview tasks.

For social networks, Aris and Shneiderman [SA06, AS07] propose a set of 6 challenges and 10 tasks. The challenges are about visualizing: *C1*) Basic networks with nodes and links; *C2*) node labels; *C3*) link labels; *C4*) directed networks; *C5*) node attributes; and *C6*) link attributes. The tasks listed start from low-level and add more SNA related tasks:

© 2010 The Author(s)
Journal compilation © 2010 The Eurographics Association and Blackwell Publishing Ltd.

T3) for every node, find the nodes that are distance 1, 2, 3 ... away; *T4*) for every node, find betweenness centrality; *T5*) for every node, find structural prestige; *T6*) find diameter of the network; *T7*) identify strongly connected or compact clusters; *T8*) for a given pair of nodes, find shortest path between them; *T9*) for every node/link, read the label; and *T10*) find all nodes/links with a given label/attribute.

Finally, Henry et al. [HFM07] list three high-level tasks specific to SNA: *U1*) identify communities; *U2*) identify central actors; and *U3*) analyze roles and positions.

2.3. Social Network Visualization

Social networks are usually visualized using node-link diagrams or adjacency matrices. Several social network systems have been designed, including VisOne [BW04], Vizster [HB05] and MatrixExplorer [HF06].

Most general graph visualization tools listed can be used to visually explore social networks but are not specialized for this purpose. They provide graph-layout algorithms and the possibility of assigning data attributes to visual attributes such as color and size, up to three or four attributes, in addition to positions. However, they may lack specialized layout methods for social networks, such as [Noa05].

Systems targeted towards SNA compute a layout based on the topology — sometimes taking some attribute into account in the layout computation — and allow the use of other attributes to decorate the nodes and links. This effectively adds 3 to 5 extra visualized attributes before the visualization becomes too complex (width, height, background color, outline color, transparency). Since social networks typically have 10 to 30 attributes associated with the nodes, exploring their values becomes tedious.

Furthermore, these visualization tools focus on the topology of the networks, not on their attributes (*C5,C6*). Topology is clearly very important (*C1,C2*), but for social networks, other perspectives may be as important to discover correlations between attribute values. Therefore, some recent systems have tried to offer better support for attribute-based navigation in multivariate networks.

2.4. Multivariate Network Visualization

Since 2006, there has been an increasing interest in visualizing multivariate networks in more clever ways than changing visual attributes of nodes and links. Wattenberg's PivotGraph system [Wat06] relies on the OLAP (Online Analytical Processing) database model to aggregate network vertices and drill up or down according to categorical attribute values. Initially, the PivotGraph system shows one node that represents the whole dataset aggregated. The user can drill down using any categorical attribute, causing the initial point to be split into several points — one for each value of the attribute — that are spread evenly on the horizontal axis. The point sizes are proportional to the number of individuals (database rows) holding the specified value; links are drawn between the points with a width proportional to the number of edges connecting the points. A second attribute can be used for splitting the vertical dimension; the points are then laid out in a regular 2D grid. PivotGraph can visualize very large databases along any categorical dimension as long as the number of categories remains small enough (*C5,C6,T9,T10*). However, it is not meant to address the tasks and challenges of full-scale SNA.

Following-up on the idea of exploring multiple attributes of a network, semantic substrates [SA06, AS07] use a different approach with no aggregation but several panes on the same window. The technique displays a social network showing several attributes unfolded. Each pane visualizes two attributes of the vertices using two dimensions. Various dynamic query widgets are placed on the window to allow filtering. Furthermore, edge filtering can be done on the panes by drawing a box around the vertices of interest, hiding links outside the box. While substrates support most of the challenges *C1-C6*, building the panes requires an initial setup phase. This phase needs computer science skills beyond that of social science researchers and, even more difficult, a clear understanding of the important attributes to show and how to lay them out in space to reveal important relationships.

Pretorius et al. [Pre08, PvW08] describe several methods to visualize multivariate networks, mainly applied to transition networks: brushing and linking of a node-link diagram with parallel histograms for each dimension, linear graph visualization of a hierarchical clustering of vertices with various attributes superimposed, and hybrid parallel coordinates. All these methods seem promising but, except the first, they are also quite complex, both visually and cognitively. The list of tasks supported by these visualizations do not focus on structural groups (*T7,T8*) but on attributes; thus, it remains to be seen whether they can be used effectively for SNA tasks.

In summary, visualizing multivariate networks is recognized as an important but difficult challenge. PivotGraph remains simple but only shows categorical attributes and does not support most SNA tasks. Semantic substrates support the identified tasks, but are complex to set up and only support visualization of a small set of attributes at a time. Pretorius' visualizations are complex, difficult to learn, and not meant to support the group tasks required by SNA. Our challenge is to provide a simple visualization tool supporting a more complete set of the SNA tasks.

3. Design Rationale

Henry and Fekete [HF06] list 13 requirements for SNA, as collected from participatory design sessions. We take into account overall system design requirements: R3) show an overview; R6) use analytical information (SNA measures); R7) preference for interaction (direct manipulation) vs. parameter tuning; R8) layout automatically and interactively;

© 2010 The Author(s)
Journal compilation © 2010 The Eurographics Association and Blackwell Publishing Ltd.

R9) filtering is necessary. Particularly important for this article are requirements related to multiple attributes:

R5, R12 - Attributes & Consensus: Taking attributes into account makes the difference between graph drawing and information visualization. Participants were not interested in displaying a unique graph; they wanted to build several representations according to the different attributes of the edges and vertices (R5). They also wanted to compare and identify a consensus among actor clusters across representations (R12). The structure of the graph may differ depending on the chosen attributes. Comparing these structures, understanding why they are similar or how they differ was a major concern. Users needed a visualization system which helps them to choose visual variables for each attribute and create multiple views of their dataset. Consulting details for each vertex or edge was also a primary interest and therefore, details should always be visible or quickly accessible.

Multivariate social networks are complex structures and the tools available for their exploration are themselves complex and require substantial training to learn and understand their features, making multivariate SNA a very challenging task. In GraphDice we thus attempt to efficiently support R5 & R12. Our goal is to simplify exploration, feature understanding, and training, by providing a tool that is simple to learn and use, yet powerful enough to analyze networks with hundreds of nodes of up to 30 dimensions. To achieve this goal, we rely on a set of design principles derived from the above requirements, as well as ones that can be seen as a stronger interpretation of the well known rules of direct manipulation [Shn97] and dynamic querying [Shn94]:

P1 Use direct manipulation (R7). Allow users to interact directly on the visualizations instead of using complex widgets that take time to understand, and act indirectly on items of interest; limit the number of hidden functions only reachable through menu navigation that disrupts the user's main focus.
P2 Keep the widgets simple and small in number. Devote most of the screen to the data, and not on widgets that consume screen real-estate.
P3 Keep the number of visual representations small to shorten learning time and limit cognitive fatigue;
P4 Keep the interactions and visualizations consistent, to facilitate learning and invite exploration;
P5 Visualize the navigation and interaction history (undo/redo states) to facilitate learning and invite traceable, reversable and thus inexpensive exploration.
P6 Support (R6) the computation of basic SNA metrics (e.g. centrality, degree, etc).
P7 Provide overview, details and attribute filtering (R3,R5).
P8 Maintain visual groupings across attributes (R12).

4. The GraphDice System

GraphDice is a visualization system for exploring multivariate social networks, following the design guidelines of Sec. 3. It extends the ScatterDice [EDF08] tool, that was originally designed for navigating and exploring multidimensional tables visualized as multiple scatterplots. Node-link diagrams can be seen as extensions of scatterplots where points are connected with links. However, the application of scatterplots to network data implies an increase of complexity of its data model that requires different features and design choices than for 2D scatterplots.

The GraphDice system (Fig. 1) consists of a visualization window and data table views. The visualization window in turn includes several components discussed next. Images come from real datasets described in detail in Sec. 6.

4.1. Overview Plot Matrix

Like ScatterDice, the overview plot matrix displays a cross-dimensional node-link plot for every combination of attributes and arranges them in a large plot matrix [FFT88]. All attributes are assigned a row and a column and a plot thumbnail is drawn at their intersection (Fig. 1a). Scatterplot matrices have never before been used to visualize graphs.

The main plot can be changed by navigating in the plot matrix, similarly to ScatterDice. This navigation is restricted to orthogonal movement along the same row or axis in the matrix: one axis is preserved while the other changes using an animated transition. Instead of simply interpolating the position of each data point, a 3D rotation is performed, as if rotating a cube or rolling dice. These transitions allow the user to extract structure from motion [Ull79] and compare the two dimensions. It also reinforces the metaphor of navigating in the attribute space defined by the plot matrix (**P4**).

As an extension to ScatterDice, user selections have been made visible on the thumbnails: each selected node is displayed with its selection color. This feature is useful for tracking selections across dimension pairs, and finding attributes where selected nodes remain grouped (**P8**).

4.2. Main Plot

Node-link diagrams can be seen as extensions of scatterplots where data points are connected with links. Therefore, the use of node-link diagrams is consistent with the ScatterDice "dice rolling" paradigm. Because GraphDice allows to focus on a single visual representation at a time (Fig. 1d), the system minimizes learning time and cognitive fatigue (**P3**).

4.2.1. Attribute Drawing

In multivariate data, attributes are often categorical (e.g. Gender or Country), resulting in multiple nodes overlapping on one or more dimension axes. GraphDice uses jitter [AS07] to provide a clearer visual indication that multiple nodes occupy the same space in the graph. Overlapping nodes are placed around a circle (Fig. 1d), centered on their original location, with a radius proportional to the number of overlapping nodes. Using this jitter mechanism, users can

© 2010 The Author(s)
Journal compilation © 2010 The Eurographics Association and Blackwell Publishing Ltd.

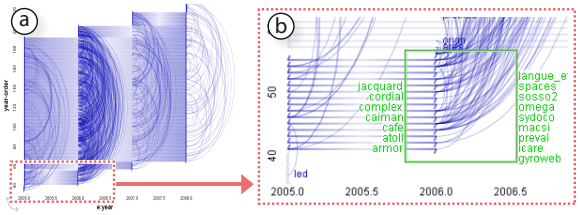

Figure 2: *An edge e:year attribute (x-axis) projected as an interval node attribute (straight lines along x-axis, 1 per author). The author nodes (y-axis) are ordered so that similar intervals are close to each other. In this co-authorship network (a), we use an excentric label to focus on part of the network (b). Notice hooks at ends of intervals.*

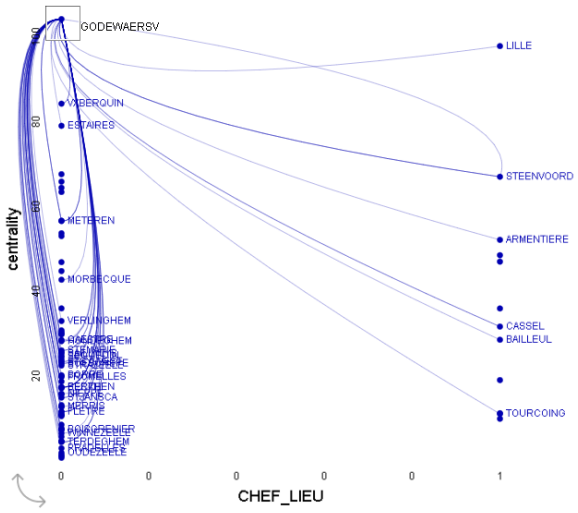

Figure 3: *Link directionality in a migration network of cities in 2 discrete administrative departments. The excentric label is used to focus on a single node and its links.*

identify at a glance dense data regions. Moreover, they can see the connections between these overlapping nodes as the circular layout provides space to draw the interconnection links. When showing categorical axes, the resulting plot is very similar to a PivotGraph [Wat06] but the aggregation is only visual, so all details remain visible.

Axes can also encode intervals, i.e. value ranges. Intervals allow aggregatation of numerical attributes of actors, e.g., the years of a researcher's publications can be replaced with the year of her first and last publications. GraphDice visualizes intervals as straight lines with short hooks in the ends, to distinguish them from links between actors (Fig. 2).

For edges with a numerical or an ordered categorical attribute, GraphDice automatically creates an associated interval node attribute and aggregates the edge values at each node. In the co-authoring network of Fig. 2, edges are articles with a publication year. GraphDice automatically creates an interval node attribute for the "year" attribute named "e:year". When this attribute is set as an axis in the plot, the links are drawn starting and ending at a position computed from their attribute value. So if "e:year" is assigned to the horizontal x-axis, authors are drawn as straight horizontal lines showing the intervals of their publication years. Links are drawn as curves with their endpoints positioned according to the value of their "year" attribute. Thus GraphDice allows the exploration of node and edge attributes in a unified way (**P3, P4**). To benefit from this feature, GraphDice supports multiple (parallel) edges between the same nodes.

4.2.2. Link Drawing

Links are drawn using splines rather than straight lines, as it has been shown to improve the legibility of dense node-link diagrams [FWDP03, AS07]. Directed links are drawn using biased splines as in [FWDP03], where bias and orientation express the link direction (Fig. 3). For undirected networks, GraphDice attempts to find the best link drawing method using a set of simple heuristics for each configuration in the plot matrix. In diagonal scatterplots and 1-D scatterplots links are drawn as semicircles, like in Arc Diagrams [Wat02]. When one of the two axes is categorical, ordered (e.g., rank, x, and y), or is an interval, links are splines bent towards the orthogonal direction (Fig. 2). When both axes are categorical, a mix of vertically and horizontally-oriented splines is used to minimize link overlapping.

The user can further tweak the rendering of individual networks in the matrix by changing the type, direction and amount of link curvature. Finally, links are colored by drawing a gradient between the two endpoint node colors.

4.2.3. Link Animation

To aid navigation between different attributes and help users maintain a mental model of the dataset, every plot maintains the links between nodes, even during plot transitions, irrespective of the visualized attributes. To our knowledge, most multivariate visualization systems for social networks do not provide persistent link presence between attribute views.

As in ScatterDice, GraphDice uses 3D rotations to animate between visualizations: a 2D network is extruded in 3D, rotated, and flattened back to a new 2D network. In Fig. 5c the plot is in a partial transition stage between two attributes. Nodes are animated the same way as datapoints in ScatterDice. In the case of interval attributes, the animation involves a progressive expansion/shrink of single points (nodes) to straight lines (intervals).

Animating links is technically more challenging than nodes, as it is not possible to reconstruct arbitrary links in 3D. E.g., there is no 3D curve that appears as an upward semicircle in a given projection and as a downward semicircle after a 90° rotation about the vertical axis. To address this

© 2010 The Author(s)
Journal compilation © 2010 The Eurographics Association and Blackwell Publishing Ltd.

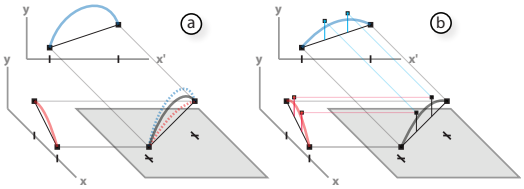

Figure 4: *(a) A consensus 3D curve is built by extruding the initial and final links and averaging them. (b) When links are splines whose intermediary control points are kept unchanged in terms of height and relative x-position, this method reduces to a 3D curve reconstruction.*

issue, we devised the following technique: before the animation starts, we compute *consensus 3D curves* from the initial and final 2D links (Fig. 4). During the initial plot extrusion phase, 2D links are morphed to consensus 3D curves by linearly interpolating the curves. At the rotation phase, the consensus 3D curves remain unchanged. Finally, at the flattening stage, 3D curves are morphed into the new 2D links.

GraphDice employs two different methods for computing the consensus 3D curves. When the initial and final 2D links have the same direction of curvature (e.g., upwards), they are both extruded in 3D; the two parametric curves obtained are then averaged to form the consensus 3D curve (Fig. 4a). Since the initial and final links have a similar shape, the visual transition to and from the 3D curve appears natural. When the two extruded curves are confounded, this technique reduces to a regular 3D curve reconstruction (Fig. 4b).

In case the initial and final 2D links have different directions of curvature, the above technique produces complex-looking 3D curves with inflection points. We found that using straight lines as consensus 3D curves produces animations that are easier to follow. Therefore, when initial and final curvatures do not match, the link is straightened during the extrusion phase, after which a 3D line rotation is performed, and finally the line bends again as the cube flattens.

4.2.4. Computed Attributes

Before loading a network, we compute additional attributes for each vertex that are useful in network exploration (**P6**). These attributes are either structural, layouts or orders.

Structural attributes include the node degree (or in and out for directed networks) and betweenness centrality [Fre77] and are computed using the JUNG network library [OFK03].

For the *layout*, we assign the X and Y attributes with positions computed using the edge-repulsion LinLog graph layout [Noa05] (default view). When the network already contains an x and y attribute (lower case), we use it as the initial layout before applying the force-directed layout algorithm. Therefore, networks containing geographical coordinates have a layout as consistent as possible with the expected geographical positions.

Orders are attributes of consecutive values, from 1 to the number of nodes in the network. When the attribute is assigned to an axis, all the nodes take an evenly distributed position along that axis following that particular order. Orders are useful to visually analyze the network. We compute an order that minimizes the bandwidth of the network using the reverse Cuthill-McKee algorithm [CM69]. This ordering groups connected nodes since the link lengths are kept short, and can augment any attribute with a meaningful topological order. Since GraphDice creates an interval node attribute for each edge attribute, we create an order that shows a meaningful progression of the intervals, sorted by increasing average values first, then by increasing range length (Fig. 2).

Contrary to ScatterDice, the order of columns and rows in the overview plot matrix is meaningful and computed automatically as a function of the similarity between dimensions [ABK98] (except for the X, Y layout dimensions that are placed at the end of the plot matrix). This feature was a request of our social scientist user group who are always interested in the correlations of their attributes. The attribute order can be manually changed through interactive drag-and-drop of rows and columns. Details on how to navigate within the plot matrix by selecting or dragging on plot thumbnails can be found in the ScatterDice article [EDF08].

As this paper focuses on visualizing rather than computing network attributes, our prototype automatically computes only a small subset of the structural and layout algorithms of JUNG [OFK03]. But as it is important for the user to be able to choose among multiple layout and structural algorithms, we plan to fully integrate JUNG in GraphDice.

4.3. Visual Queries on Edges and Nodes

Beyond basic navigation functionality, the original ScatterDice tool also provides advanced visual queries/selections on nodes (**P1**), called *query sculpting*, using bounding volumes that allow the user to iteratively filter the dataset. The process consists of simple lasso-selecting data items in the main scatterplot using 2D bounding volumes (boxes or convex hulls) and then iteratively refining the selection from other viewpoints while navigating the plot matrix (similarly to high-dimensional brushing [MW95]). The active query hull is blinking to highlight nodes affected by the refinements. Several color-coded queries are available to the user, visible in a *query widget* (Fig. 1c), much like a Photoshop layer pane. Queries are sculpted in the data space of the current dataset, two dimensions at a time (the dimensions currently viewed in the main plot) and the selection may either be a union or an intersection with the existing items in the current query. The visual representation of queries conforms to the overall navigation metaphor and is thus also "rolled" as a 3D convex hull when the user navigates in the scatterplot matrix. This helps the user to perceive correlations between the extents of a query in two adjacent plots (**P12**).

© 2010 The Author(s)
Journal compilation © 2010 The Eurographics Association and Blackwell Publishing Ltd.

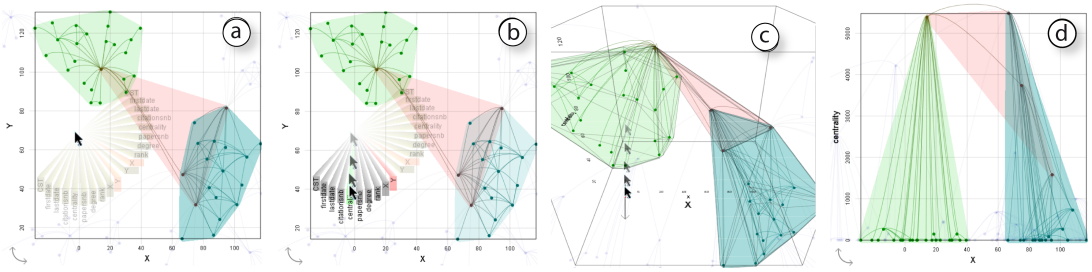

Figure 5: *(a) X,Y layout for a graph and queries. Menu navigation: (a) Using the right mouse button the user invokes the dimension menu, with the currently selected dimensions marked in red. (b) As the user drags over a menu the new vertical dimension is highlighted in green. If the user releases the menu, a default speed animation transitions to the new centrality dimension (d). (c) If the user continues dragging, a fast slider appears allowing the user to control the animation speed.*

Queries behave in the same way on both edge and actor axes, providing a consistent direct manipulation approach (**P1, P4**). Nodes are drawn as small points with a color related to their selection in a visual query (described below) and an intensity mapped to a user selected attribute (initially constant). When a node is selected, it and its edges belong to one or several query layers — each with its unique color. When performing a visual querying in an edge interval attribute view, only edges that belong to this selection are part of the query, making it easy to focus on patterns of a specific edge value, such as a time period. Each query layer shows the proportion of the nodes and links over the number of all actors and links in the network. Contrary to ScatterDice, we color the nodes according to the layers they belong to; accordingly blended if a node belongs to several layers. A selection layer is visualized as the convex hull of all the nodes it contains, colored with the layer's color 90% translucent.

4.4. Query History

Maintaining the history of the exploration process (**P5**) is important to help users revisit important views and keep track of their exploration [HMSA08]. In the main window, we have added an interactive query history tool (Fig. 1b) to help users in revisiting previous steps of exploration. Each time a new query is formulated by the use of the lasso or directly in the table, the current state of the selection and the current axes assignment are stored as a new query step in the history panel. The thumbnail of the current scatterplot is added to the history panel as being the last (and current) query state, showing the current state of the selection over the thumbnail. The thumbnails with the same selection as the current ones are outlined. If the new selection corresponds to the previous recorded step, we consider that the new query corresponds to an undo action: we remove the last step from the history and revert to the previous step; if the current selection state is not the last in the history when a new query is formulated, the states after the current one are removed.

Each thumbnail is a three-state button. While hovering over a thumbnail, the selections in that history state are shown transiently on the main plot. Clicking on a thumbnail permanently installs the selections on the current attribute plot. Clicking again on that thumbnail "rolls the dice" to its associated stored plot view. The history panel thus allows for going back to a previous state, both in reinstalling the selections in a first step, and then going back to the attribute plot view where the queries were performed.

4.5. Other functionality

As in ScatterDice, users can navigate in the overview plot matrix by clicking on individual cells or by pressing the arrows keys. In GraphDice, users can additionally navigate from within the plot view by invoking a control menu [PLVB00] (Fig. 5). This menu allows them to select both the axis to "roll" (either x or y) and the new dimension to display. Releasing the mouse button on a menu item triggers the animation, and dragging past a menu item pop-ups a control slider that allows precise control of the 3D transition.

A toolbar is also provided for turning on and off different visualization options (Fig. 1e). Options include auto-zoom, labels, and the display of in- or out- links within selections. The toolbar supports fast pre-visualization by using three-state buttons similar to the 3 state history buttons. Finally, an excentric labels lens [FP99] shows details of specific nodes.

4.6. Data Tables

As a supplement to the main visualization window showing node attributes, we provide users with two tabular views (Fig. 1 right) of their dataset (nodes and attributes/dimensions) and a table of their links. Interacting with the dataset table is similar to interacting with a spreadsheet: users can sort data by dimension, select multiple entries, rearrange dimensions, perform text searches, etc. By default, this table includes all data, but as data gets selected by users on the main visualization, the table entries are updated to reflect or adjust the current user selection. More precisely, when the excentric label is active, the table only shows the nodes inside the lens. When zoom is active, the table only shows the selected nodes.

© 2010 The Author(s)
Journal compilation © 2010 The Eurographics Association and Blackwell Publishing Ltd.

Thus, the tables act as a details-on demand mechanism by providing information on entries (**P7**), as well as a verification and correction mechanism for user data selections. Finally, the data table is used for dynamic queries: when rows are selected, a right click triggers a popup menu to choose a query layer, assigning the selected rows to this layer.

5. Following Guidelines and Supporting Tasks

We designed GraphDice following the principles of Sec. 3, as indicated in our system description. Although our system supports all the described principles, we kept the number of widgets small and consistent in functionality, concentrating more on having space for the main network plot (**P1, P2**).

Our system is designed to address a large number of challenges and tasks described in Sec. 3. We support visualization of the network (*C1*) , and node and link label visualization (*C2,C3*) through our excentric label, 3 state buttons and interactive data tables. By using curved links we display directionality (*C4*), while our plot matrix visualization and plot transitions allows us to display a large number of node and edge attributes (*C5,C6*). Moreover, our data table is used to implement the requirements *C5* (show node attributes).

When it comes to tasks, we support betweenness and centrality finding (*T4, U2*) through our computed attributes and relevant plots. Moreover, users can visually identify strongly connected and compact actor clusters based on proximity in our layout algorithms (*T7,U1*), and compare them across dimensions using visual queries and plot transitions. By using either excentric label or visual queries, node/link labels are readable (*T9*). Finally, our data tables act as a details-on demand mechanism (providing information on entries) for task *T10* (find all nodes with a given attribute), as well as a verification and correction mechanism for user data selections.

6. Datasets

We built GraphDice to aid visual exploration of multivariate social networks. To evaluate its utility, we used it to visualize 3 real datasets, that differ in number of nodes, edges, node dimensions and connection patterns. Our goal was to determine if dataset characteristics and patterns were visible using GraphDice as an exploratory visualization tool. The images referenced show examples of dataset views and the mentioned exploratory findings. While our paper does not focus on scalability, we have successfully rendered in real-time graphs of up to 2000 nodes and 6000 edges. Visualizing larger graphs while allowing readability would require aggregation, an issue that is out of the scope of this paper.

6.1. InfoVis Co-authorship Dataset

The first exploration dataset is the largest connected component of the InfoVis coauthorship dataset from 1998-2002, extracted from the InfoVis 2004 Contest [FGP04]. It contains 694 publications (edges), among 149 authors (nodes), categorized in 10 dimensions. Of these dimensions, 5 are the main dataset attributes (papers, citations, first publication date, last publication date), while the rest are computed automatically by the system (centrality, rank, degree, constant, as well as X, Y coordinates for a 2D graph layout). One edge dimension (publication year) was present.

We focus on part of the network (Fig. 5a), by zooming into query selections representing 2 closely connected groups of co-authors (blue, green) and their connections (red). We observe through centrality (Fig. 5d) that the first group has a single central actor, indicating student-advisor relations, whereas the second has multiple such actors pointing to a more connected research group, like a research organization. The 2 different groups are linked to each other only through connections between these major actors.

6.2. Organization Collaboration Dataset

The second dataset (Fig. 2b) is the collaboration network in terms of document creation, between the research teams of a large nation-wide research organization. The dataset consists of 197 teams (nodes) and 1581 collaboration articles between teams (edges). There are 8 dimensions for the teams: system computed ones (as above) as well as creation year and publication count. There are 2 dimensions for the publications (edges), year of publication and count (how many such publications between two teams within a year).

By observing the co-authorship trends of different teams across multiple years, we see an increased amount of publications in 2006 (Fig. 2b). What seems to differentiate this year from others is that a large number of teams disbanded because their project finished (end of interval associated with teams in the y-axis). Focusing on these projects (Fig. 2a), we see that the number of publications the in year they disbanded increased from previous years. We hypothesize that as this was the last year of the projects, a large number of research conducted in previous years was finally published.

6.3. Historical Migration Dataset

The third dataset comes from a historian specializing in SNA of migration patterns in 19th century France. It consists of 1776 population migration paths (edges) between 75 cities (nodes) in northern France in 3 time periods. The 18 dimensions of the dataset included, but were not limited to, dominant language of city (French or Flemish), percentage of population occupation (worker, farmer, etc), proximity to riverbanks, administration department, longitude and latitude, etc., as well as automatically computed measures (as above). An edge attribute (migration year) was also present.

By observing the migration network using GraphDice in our user feedback session, the historian was able to form a new hypothesis (Fig. 3). Migrations to and from Godewaersv and cities within the same department are balanced and more frequent, but mainly outgoing migrations take place between this city and cities in the other department.

© 2010 The Author(s)
Journal compilation © 2010 The Eurographics Association and Blackwell Publishing Ltd.

7. Preliminary User Feedback

We conducted a full day workshop with a historian specializing in SNA. Although a more formal user study is required, this session helped us gather informal preliminary user feedback and examine wether GraphDice can be used for realistic SNA tasks. The participant used GraphDice to explore her historical migration dataset (Sec. 6.3) and was able to use the tool effectively with less than 15 min. of training.

Our user immediately commented on the usefulness of the plot matrix with the miniature plots, stating that it is useful to "have all combinations of attributes immediately available to quickly explore hypotheses". This was more pronounced when queries were active as "the query feedback in the overview matrix reveals interesting patterns and then guides the exploration as it stirs up unexpected questions.". She found the default X,Y view very useful as "it allows to quickly get back to the traditional representation of the network to rebuild my referring mental map". She also mentioned that having the matrix always visible made it "easier to remember which combinations she had already viewed". We thus feel we have provided users with a successful representation of the dimension space that is easy to navigate.

Animated transitions between dimensions were deemed useful as a means to "compare similar attributes". By interactively controlling the transition between two attributes (e.g. two centrality measures) she could "quickly dismiss one from further analysis because they are too similar".

Queries were deemed very useful in "quickly exploring correlations between attributes". Using the convex hulls of a selection in one dimension, the historian was able to explain the selection by attributes in other dimensions. She also commented that suppressing links inside and outside a selection helped in the understanding of inter-community patterns. The synchronized data tables were also deemed extremely useful, as they "provide details and security and confidence that you are looking at the correct data".

As for the drawing of biased splines to show direction, she enjoyed "being able to see immediately reciprocity of migrations between pairs or groups of nodes", something she was "not able to see with the classical arrow representation".

Through the edge interval view tied to different time periods, she was able to see the evolution of the different cities across time. She found it very easy to identify stable and changing patterns between the three periods, as they are shown concurrently without interfering with each other.

She added that jitter helped to quickly identify clusters and decide which attribute values to focus on (e.g. medium cities at river-banks), while the animated transition between dimensions gave more information on the jittered nodes.

She also requested additional features for graph annotation. She found the query history useful for re-tracing exploration steps, but would also like to take snapshots of "preferred moments" (including the cursor position) and store them in a history of favorites, so as to quickly revisit views of interest and replay scenarios. Being able to mark preferred views in the matrix was also mentioned: she visually marked one graph by customizing link curvature to make it distinct, but a more light-weight mechanism was requested.

Our historian commented on how easy it is to manipulate GraphDice compared to her other tools, like UCINET or Pajek. She pointed out it is very well suited for discovering patterns and learning new datasets, as it provides visual representations of a network from many points of view. She added that even if she doesn't want to learn statistics (a requirement in most current tools), she can still get results with GraphDice. Lastly she mentioned she would use GraphDice to communicate her findings, as she surmised that a live demo of a well-driven scenario on an easy to use tool, would make it easier to understand and correlate her findings than showing a set of well-chosen snapshots of the datasets.

8. Conclusions and Future Work

We have presented GraphDice, a tool for multivariate visualization of social networks. The GraphDice tool is based on a comprehensive approach to visualizing attributes — derived and intrinsic alike — using an attribute plot for information display and an overview plot matrix for navigation, similar to the ScatterDice system [EDF08]. Attribute plots draw nodes according to their values in the two attributes displayed by the plot, and also show edges between the nodes as visual links. Of course, point occlusion is a major issue when aggregating nodes according to a potentially small set of attribute values, and we address this by introducing node jitter proportional to the amount of overlap in the visualization. We have also showcased usage of the tool through an initial participatory design workshop involving a historian.

Our future work entails extending the GraphDice system to sophisticated history storage and navigation beyond simple undo mechanisms and query storage. We plan to augment our system to allow users to take snapshots of their navigation history as landmarks in their dataset navigation, and to provide notification mechanisms to inform users when they have reached a navigation state that is similar to previously visited states. To further support navigation and long term exploration within the attribute space, we also plan to create annotation mechanisms (such as bookmarking interesting plots in the plot matrix or annotating the main plot). Furthermore, we intend to allow users to compute additional graph metrics, including different clustering and centrality coefficients. We also plan to perform a longitudinal user study of social scientists analyzing their networks using GraphDice.

An online WebStart-enabled version of the system can be accessed at: `http://www.aviz.fr/graphdice/`

Acknowledgments

We would like to thank Claire Lemercier for her feedback.

© 2010 The Author(s)
Journal compilation © 2010 The Eurographics Association and Blackwell Publishing Ltd.

References

[ABK98] ANKERST M., BERCHTOLD S., KEIM D. A.: Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In *Proceedings of the IEEE Symposium on Information Visualization* (1998), pp. 52–60.

[Ada06] ADAR E.: GUESS: a language and interface for graph exploration. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems* (2006), pp. 791–800.

[AS07] ARIS A., SHNEIDERMAN B.: Designing semantic substrates for visual network exploration. *Information Visualization 6*, 4 (2007), 281–300.

[BEF09] BORGATTI S. P., EVERETT M. G., FREEMAN L. C.: UCINET 6 for Windows: Software for social network analysis, 2009.

[BW04] BRANDES U., WAGNER D.: *Graph Drawing Software*. Springer-Verlag, 2004, pp. 321–340.

[CM69] CUTHILL E., MCKEE J.: Reducing the bandwidth of sparse symmetric matrices. In *Proceedings of the National ACM Conference* (1969), pp. 157–172.

[dNMB05] DE NOOY W., MRVAR A., BATAGELJ V.: *Exploratory Social Network Analysis with Pajek*. Structural Analysis in the Social Sciences. Cambridge University Press, Mar. 2005.

[EDF08] ELMQVIST N., DRAGICEVIC P., FEKETE J.-D.: Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1141–1148.

[FFT88] FISHERKELLER M. A., FRIEDMAN J. H., TUKEY J. W.: PRIM-9: An interactive multi-dimensional data display and analysis system. In *Dynamic Graphics for Statistics*, Cleveland W. S., McGill M. E., (Eds.). Wadsworth & Brooks/Cole, 1988, pp. 91–110.

[FGP04] FEKETE J.-D., GRINSTEIN G., PLAISANT C.: IEEE InfoVis 2004 Contest, the history of InfoVis. www.cs.umd.edu/hcil/iv04contest, 2004.

[FP99] FEKETE J.-D., PLAISANT C.: Excentric labeling: dynamic neighborhood labeling for data visualization. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems* (1999), pp. 512–519.

[Fre77] FREEMAN L. C.: A set of measures of centrality based on betweenness. *Sociometry 40*, 1 (March 1977), 35–41.

[FWDP03] FEKETE J.-D., WANG D., DANG N., PLAISANT C.: Overlaying graph links on treemaps. IEEE Symposium on Information Visualization Conference Compendium (demonstration), October 2003.

[HB05] HEER J., BOYD D.: Vizster: Visualizing online social networks. In *Proceedings of the IEEE Symposium on Information Visualization* (2005), pp. 32–39.

[HF06] HENRY N., FEKETE J.-D.: MatrixExplorer: a dual-representation system to explore social networks. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 677–684.

[HF07] HENRY N., FEKETE J.-D.: MatLink: Enhanced matrix visualization for analyzing social networks. In *Human-Computer Interaction — INTERACT 2007* (2007), vol. 4663 of *LNCS*, pp. 288–302.

[HFM07] HENRY N., FEKETE J.-D., MCGUFFIN M. J.: NodeTrix: A Hybrid Visualization of Social Networks. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1302–1309.

[HMSA08] HEER J., MACKINLAY J., STOLTE C., AGRAWALA M.: Graphical histories for visualization: Supporting analysis, communication, and evaluation. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1189–1196.

[INS] International network for social network analysis. http://www.insna.org/.

[LPP*06] LEE B., PLAISANT C., PARR C. S., FEKETE J.-D., HENRY N.: Task taxonomy for graph visualization. In *Proceedings of BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV)* (2006), pp. 82–86.

[MW95] MARTIN A. R., WARD M. O.: High dimensional brushing for interactive exploration of multivariate data. In *Proceedings of the IEEE Conference on Visualization* (1995), pp. 271–278.

[Noa05] NOACK A.: Energy-based clustering of graphs with nonuniform degrees. In *Proceedings of the 13th International Symposium on Graph Drawing* (2005), pp. 309–320.

[OFK03] O'MADADHAIN J., FISHER D.AND NELSON T., KREFELDT J.: JUNG: Java universal network/graph framework, 2003.

[PLVB00] POOK S., LECOLINET E., VAYSSEIX G., BARILLOT E.: Control menus: execution and control in a single interactor. In *Extended Abstracts of the ACM CHI Conference on Human Factors in Computing Systems* (2000), pp. 263–264.

[Pre08] PRETORIUS A. J.: *Visualization of State Transition Graphs*. PhD thesis, Eindhoven University of Technology, November 2008.

[PvW08] PRETORIUS A. J., VAN WIJK J. J.: Visual inspection of multivariate graphs. *Computer Graphics Forum 27*, 3 (2008), 967–974.

[SA06] SHNEIDERMAN B., ARIS A.: Network visualization by semantic substrates. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 733–740.

[Shn94] SHNEIDERMAN B.: Dynamic queries for visual information seeking. *IEEE Software 11*, 6 (Nov. 1994), 70–77.

[Shn97] SHNEIDERMAN B.: Direct manipulation for comprehensible, predictable and controllable user interfaces. In *Intelligent User Interfaces* (1997), pp. 33–39.

[Ull79] ULLMAN S.: *The Interpretation of Visual Motion*. MIT Press, 1979.

[Wat02] WATTENBERG M.: Arc diagrams: Visualizing structure in strings. In *Proceedings of the IEEE Symposium on Information Visualization* (2002), pp. 110–116.

[Wat06] WATTENBERG M.: Visual exploration of multivariate graphs. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems* (2006), pp. 811–819.

[WF94] WASSERMAN S., FAUST K.: *Social Network Analysis*. Cambridge University Press, 1994.

© 2010 The Author(s)
Journal compilation © 2010 The Eurographics Association and Blackwell Publishing Ltd.