A Space-Filling Visualization Technique for Multivariate Small-World Graphs

Pak Chung Wong, *Member, IEEE*, Harlan Foote, Patrick Mackey, George Chin, *Member, IEEE*, Zhenyu Huang, *Senior Member, IEEE*, and Jim Thomas, *Member, IEEE*

Abstract—We introduce an information visualization technique, known as GreenCurve, for large multivariate sparse graphs that exhibit small-world properties. Our fractal-based design approach uses spatial cues to approximate the node connections and thus eliminates the links between the nodes in the visualization. The paper describes a robust algorithm to order the neighboring nodes of a large sparse graph by solving the Fiedler vector of its graph Laplacian, and then fold the graph nodes into a space-filling fractal curve based on the Fiedler vector. The result is a highly compact visualization that gives a succinct overview of the graph with guaranteed visibility of every graph node. GreenCurve is designed with the power grid infrastructure in mind. It is intended for use in conjunction with other visualization techniques to support electric power grid operations. The research and development of GreenCurve was conducted in collaboration with domain experts who understand the challenges and possibilities intrinsic to the power grid infrastructure. The paper reports a case study on applying GreenCurve to a power grid problem and presents a usability study to evaluate the design claims that we set forth.

Index Terms—Data and knowledge visualization, information visualization, visualization techniques and methodologies.

1 Introduction

We present a novel visualization technique, known as GreenCurve, to interactively analyze large sparse graphs with hundreds of thousands of nodes on a modest desktop computer. The design of GreenCurve is primarily targeted at analyzing multivariate small-world graphs [21] that generally have high degrees of clustering and small average path lengths relative to their number of nodes. Drawing a graph of this size has long been recognized as a computational challenge [6]. Visualizing a large number of data variates associated to the graph nodes turns out to be just as difficult due to limited screen space and pixel resolution. GreenCurve has been developed to address these two major visualization hurdles in graph analytics.

GreenCurve's customized graph layout has proven valuable in a number of visual analytics applications (as described in this paper). Given a sparse graph in Fig. 1a, we first traverse its connections and tag each node with a unique sequential number in the order of traversal in Fig. 1b. This sequential list of nodes is then folded into a fractal [26] pattern to form a very compact layout in Fig. 1c. The result is a highly space-effective graph visualization that gives an overview of the graph with guaranteed visibility of every graph node.

The visualization concept of a GreenCurve graph shares a common design philosophy with that of a hierarchical Treemap [27]; they both rely on spatial cues to approximate the node connections and thus eliminate the links between the nodes in the visualization. The biggest challenge of generating a GreenCurve layout (as compared to a Treemap) is to enforce a reasonable order among the graph nodes based on their connectivity and do so in interactive time for a very large sparse graph.

Instead of using traditional algorithms such as depth-first search to traverse the graph, GreenCurve achieves an approximate ordering by effectively solving the Fiedler vector (the eigenvector corresponding to the second smallest eigenvalue) of the Laplacian matrix [9] of the graph and then use the Fiedler vector to order the neighboring nodes. Because of the potentially enormous size of a Laplacian matrix (square of the number of graph nodes), solving the Fiedler vector on a desktop computer can be a daunting task that requires multiple heuristics to overcome the computation hurdles.

GreenCurve is a part of a larger graph analytics suite, known as Have Green [36], currently under development at the Pacific Northwest National Laboratory (PNNL). Although GreenCurve is developed to support the study of the North American electric power grids [11], we demonstrate in this paper that it can be applied to other types of graphs, such as a social network, that possess small-world [21] properties.

In our power grid application, GreenCurve is used in conjunction with GreenGrid [38] to provide a planning and monitoring platform for power grid operations. The former is mainly used to display a large number, which can go beyond a hundred, of multivariate sensor information of individual network nodes. The latter is applied to analyzing the system dynamics of the entire power grid. (Interested readers are referred to [38] for more real and practical examples on power grid analytics.) Both the case study and

- *P.C. Wong, P. Mackey, G. Chin, and Z. Huang are with the Pacific Northwest National Laboratory, 902 Battelle Blvd., PO Box 999, MSIN J4-32, Richland, WA 99352.*
E-mail: {pak.wong, patrick.mackey, george.chin, zhenyu.huang}@pnl.gov.
- *H. Foote and J. Thomas were with the Pacific Northwest National Laboratory, 902 Battelle Blvd., PO Box 999, MSIN J4-32, Richland, WA 99352 at the time of this work. They are now deceased.*

Manuscript received 23 June 2010; revised 11 May 2011; accepted 13 May 2011; published online 13 June 2011.
Recommended for acceptance by C. North.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-2010-06-0126.
Digital Object Identifier no. 10.1109/TVCG.2011.99.

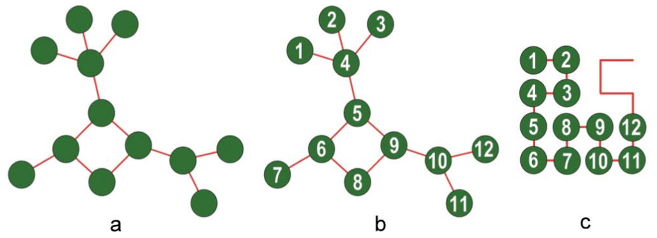


Fig. 1. (a) A force-directed graph layout. (b) Neighboring nodes tagged with unique sequential numbers. (c) Tagged graph nodes are reordered using a Hilbert curve pattern.

the usability study of this paper are based on the use of power grid data.

The paper introduces the rationale that underlies our visualization design, explains the theory that supports the technical approach, demonstrates examples using a diverse assortment of practical graphs, presents a realistic case study based on an electrical power grid's graph, discusses a usability study that investigates the strengths and weaknesses of the technique, and finally describes ongoing extension work on graph analytics.

2 Related Work

GreenCurve's design involves a number of visualization and analytics topics. These topics are not necessarily related to each other in theory and/or applications. This section highlights some of the relevant work previously presented in the corresponding communities.

2.1 Graph Drawing

The graph drawing community has led the studies in many graph drawing and layout issues for decades. Di Battista et al. [6] summarize most of the major graph drawing algorithms and their applications. The proceedings of the annual Graph Drawing Symposia [12], now in its 19th year, and the *Journal of Graph Algorithms and Applications* [18] provide a wealth of information on the cutting-edge technology.

2.2 Graph Visualization

Visualizing graphs and hierarchies has been a major study topic within the data visualization community since its conception in the early 1990s. The two textbooks by Card et al. [2] and Chen [3] cover much of the major research and applications surrounding graph, network, and hierarchical visualization. The survey paper by Herman et al. [15] represents the most complete literature review up to 2000. The annual IEEE Symposium on Information Visualization [17] continues to produce new results on various topics of graph visualization.

A major difference between the graph visualization and graph drawing communities is that the former almost always involves some sort of interaction, whereas the latter focuses heavily on algorithmic developments. The latest challenge is to integrate the best of the two communities and form a new environment of graph analytics.

2.3 Social Network and Small-World Analysis

GreenCurve's design is primarily targeted at large small-world [21] graphs that have high degrees of clustering and small average path lengths relative to their number of nodes. A classic example of a small-world graph is a social network [31] where people generally organize and link to one another through short chains of associations or acquaintances. Beyond social networks, small-world graphs also occur in many other real-world models such as gene regulatory networks and internet network traffic. They are considered a class of random graphs that have been extensively studied in network theory.

2.4 Very Large Graph Analytics

Drawing a graph nicely on screen is computationally expensive [6]. There is an ongoing community effort to speed up the drawing process by developing adaptive algorithms with complexity of $O(n^2)$ and beyond. Notable work in this area has recently been presented by Harel and Koren [14] and Walshaw [30]. In addition to the cutting-edge layout algorithms, the two papers provide resourceful clues and ideas for further improvements on their designs. Their references sections also present a wealth of information covering topics from graph partitioning to Laplacian Eigenvector computation. GreenCurve's design is based partially on Walshaw's algorithm with multiple enhancements [37] to speed up the eigenvector computation.

2.5 Fractal-Based Visualization

Information visualization researchers have previously applied the concept of a space-filling fractal curve to explore different types of data sets. Wong et al. [39] combine a Hilbert-curve design with novel signal processing techniques to visually align whole genome sequences with over a million entries in interactive time on a desktop computer.

Muelder and Ma [22] report a graph visualization approach that is somewhat similar to the one in GreenCurve. Notable differences between the two are: 1) the node-ordering approach used in GreenCurve preserves much of the geographic information of the original graph, and 2) the use of graph links in the visualization. GreenCurve uses distance proximity to approximate the node connections whereas [22] uses graph links explicitly.

2.6 Node Ordering and Clustering Visualization

The use of Fiedler vector to order the graph nodes in GreenCurve is related to the topic of node index ordering frequently found in graph clustering visualization. Muller et al. [23] provide a comparison study of eight popular node-ordering algorithms applied to matrix-based graph visualization. Additionally, Muelder and Ma [22] use a clustering approach that is prioritized by the size of the cluster. In other words, it does not preserve the spatial ordering of the original graph as the Fiedler-based approach used in GreenCurve does.

3 Brief Power Grid Data Background

Because we use power grid data extensively to support our discussion, we describe some of the characteristics of the underlying data and the requirements to analyze it.

Depending on different modes and aspects of the analysis, the size of a power grid network ranges from tens of thousands (distribution feeder level) to tens of millions (street meter level) nodes. Each of the network nodes can

Fig. 2. A snapshot of the Electricity Infrastructure Operations Center at the Pacific Northwest National Laboratory.

have over a hundred data parameters feeding information to the control center continuously around the clock. (The number of parameters will grow significantly when the next-generation Smart-Grid technology is fully utilized and integrated.) Ideally, we want to maximize the available screen pixels and show as many parameters as possible on a wall display, such as the one shown in Fig. 2 in the Electricity Infrastructure Operations Center[1] [8] (EIOC) at PNNL. This display information can be shared among operators in the control room.

The fundamental requirement is to find a way to effectively and efficiently display both the electronics (sensor readings) and spatial (relative senor locations and their connectivity) aspects of the power grid information in real time. This is indeed both a presentation and analytical visualization challenge.

4 Design and Implementation

The design and implementation of GreenCurve can be divided into two equally important parts: 1) ordering the graph nodes, and 2) generating the graph visualization. Technically, GreenCurve projects a high-dimensional graph layout onto a 1D list of graph nodes and then folds the list into a space-filling fractal curve for visualization. This section describes the technical hurdles and presents our solutions.

4.1 Ordering the Graph Nodes

During the development of another Have Green (graph partitioning) tool, we learned that the Fiedler vector of the graph Laplacian [9] provides a robust option to order the neighboring graph nodes. The already eclectic approach is particularly appealing because GreenCurve can share the computation and reuse the same Fiedler vector generated for the graph partitioning tool. Here, we introduce some basic terminology followed by the design and implementation details of the node ordering step.

4.1.1 Laplacian Matrix

The *Laplacian matrix* L of a graph G, also known as the *graph Laplacian*, is defined as $L = D - A$ where D is the *degree matrix* of G and A is the *adjacency matrix* of G. Given a graph G with n vertices, the matrix $L = (l_{i,j})_{n \times n}$ satisfies

$$l_{i,j} = \begin{cases} \deg(v_i), & \text{if } i = j, \\ -1, & \text{if } i \neq j \text{ and } v_i \text{ adjecent } v_j, \\ 0, & \text{otherwise.} \end{cases}$$

TABLE 1
The Fiedler Vector and Their Corresponding Node Numbers of the Graph in Fig. 1

Node Numbers	Fiedler Vector (Sorted)		Node Numbers	Fiedler Vector (Sorted)
1	-0.3672120		7	0.0253342
2	-0.3672120		8	0.0853044
3	-0.3672120		9	0.1345950
4	-0.3018160		10	0.3463890
5	-0.0518793		11	0.4214430
6	0.0208225		12	0.4214430

4.1.2 Algebraic Connectivity and Fiedler Vector

The *algebraic connectivity* [9] of the above graph G is the second smallest eigenvalue of the Laplacian L of G. The eigenvector associated with the algebraic connectivity is known as the *Fiedler vector* [9] of L. The magnitude of the algebraic connectivity has implications for connectivity properties such as clustering and segmentation. For instance, GreenCurve uses the Fiedler vector to assign node numbers of a sparse graph.

The example in Fig. 1 is constructed following this approach. Given the graph shown in Fig. 1a, we first find its graph Laplacian, then compute the eigenvalues, and eventually solve the Fiedler vector. The node numbers shown in Fig. 1b are assigned based on the sorted Fiedler vector of the graph. Table 1 shows the Fiedler vector and the corresponding node numbers.

4.1.3 Very Large Laplacian Matrix

The design of the node-ordering step can be straightforward if the underlying sparse graph has no more than a few hundred nodes. In that case, we can simply use one of the LAPACK [19] routines such as *dsyev* to solve the eigenvectors. For larger graphs with tens to hundreds of thousands of nodes, GreenCurve needs additional heuristics to tackle the large matrix problem.

Instead of solving the entire graph Laplacian all at once, GreenCurve 1) generates a multiscale hierarchy with increasingly coarse graphs $G_0, G_1, \ldots, G_i, \ldots, G_m$ (with G_m being the coarsest), 2) computes the Fiedler vector V_m of the Laplacian matrix L_m of G_m, and then 3) refines and propagates the Fiedler vector V_m to all the graphs in the hierarchy until the finest G_0 is reached.

4.1.4 Multiscale Graph Hierarchy

Our work on generating a multiscale graph hierarchy for a large graph is based partly on Walshaw [30] with a number of significant modifications [37] customized for data that exhibit small-world [21] properties. We have applied a similar multiscale approach to develop another Have Green [36] tool, known as GreenMax [37], which interactively scans very large sparse graphs with up to a million nodes. GreenCurve is designed to share the same multiscale hierarchy with GreenMax. In other words, once the multiscale hierarchy of a graph is generated, it can be reused over and over again for multiple Have Green tasks.

Computationally, the complexity of creating the multiscale hierarchy [37] is $O(|N_i| + |L_i|)$, where N_i and L_i are the number of nodes and links in each level i. Because the coarsening algorithm maintains an approximately 50 percent reduction rate at each level (and thus the number of $\text{levels} = log_2 m$), the complexity of the algorithm remains in the order of $O(n)$.

4.1.5 Multilevel Fiedler Vector Algorithm

For years, the popular Lanczos algorithm [4] has been used extensively to find eigenvalues and eigenvectors of a large square matrix or the singular value decomposition (SVD) of a large rectangular matrix. More advanced algorithms [1], [16] focusing on computing only the Fiedler vector have since been developed for applications such as graph partitioning. GreenCurve modifies and enhances a fast multilevel algorithm suggested by Barnard and Simon [1], which is proven to outperform the Lanczos algorithm [4], to find the Fiedler vector. A performance comparison between the two algorithms is presented later in Section 4.1.7. More recently, Manguoglu [20] presents a parallel implementation of the Fiedler vector.

After a multiscale hierarchy is generated (as described in Section 4.1.4), GreenCurve uses the LAPACK routine (*dsyev*) to find the initial Fiedler vector V_m of the coarsest graph G_m. Because the size of G_m is small (~200 nodes in GreenCurve), the potential delay caused by the primitive LAPACK routine is insignificant.

Using the coarsening information stored in the graph hierarchy, GreenCurve 1) injects (i.e., interpolates an approximate) Fiedler vector V_{m-1} from V_m for the next finer graph G_{m-1}, and 2) constructs the graph Laplacian L_{m-1} for G_{m-1}. Both L_{m-1} andV_{m-1} are then sent to a vector refinement process known as *Rayleigh Quotient Iteration* (RQI) [25] to obtain an increasingly accurate Fiedler vector. The algorithm to find the Fiedler Vector is listed below.

Algorithm: Multilevel Fiedler Vector
Input: A sparse graph G
Output: Fiedler vector of G
1. Generate $G_0, G_1, \ldots G_m$ from G
2. $L_m \leftarrow$ Laplacian Matrix (G_m)
3. $V \leftarrow$ Fiedler Vector (L_m)
4. **for** each G_i from G_m to G_0
5. $L_i \leftarrow$ Laplacian Matrix (G_i)
6. $V_{i,initial} \leftarrow$ Injection (V)
7. $V_{i,final} \leftarrow \mathrm{RQI}(V_{i,initial}, G_i, L_i)$
8. $V \leftarrow V_{i,final}$
9. Fiedler Vector $\leftarrow V$

Given a matrix and vector pair of L_i and V_i (or just L and V for any G_i), RQI iteratively solves the linear sparse system

$$(L - \lambda_k I)V_{k+1} = B_k$$

where B_k is the normalized V_k, $\lambda_k = B_k^T L B_k$, which is the eigenvalue of the last iteration, and V_{k+1} is the refined Fiedler vector at iterative level k.

Because we start with an accurate Fiedler vector and RQI converges rapidly, only a few ($k = 1$ to 2 in our experiments) iterations are needed toward a stable answer for graph G_i.

4.1.6 Fast Matrix Solver—UMFPACK

While the design of the multiscale Fiedler vector algorithm is extremely effective, the runtime performance of the algorithm can further benefit from a fast matrix solver that solves the sparse linear system within RQI. Among all proven matrix libraries available to date, we found that UMFPACK [28] consistently outperforms SYMMLQ [24], which is used by [1] to compute the Fiedler vector. And both UMFPACK and SYMMLQ significantly surpass the performance of LAPACK [19].

TABLE 2
All Four Performance Comparison Results (Columns 3 to 6) are Reported in Wall Clock Seconds

#Nodes	#Links	Multiscale Hierarchy	Fiedler GC	Fiedler Lanczos	Fiedler LAPACK
500	1517	0.021	0.037	0.227	0.952
1000	2644	0.049	0.064	0.727	9.440
2000	6757	0.164	0.079	2.205	78.120
4648	5938	0.202	0.129	8.277	732.303
9296	15014	0.439	0.175	32.952	5761.308
18592	30031	1.002	0.400	104.174	N/A
46480	75080	2.725	0.934	571.143	N/A
102256	165178	6.989	2.032	2472.58	N/A
204512	213847	17.033	4.370	9707.57	N/A
409024	660721	35.060	9.636	39526.1	N/A
1022560	1535295	130.456	32.048	N/A	N/A

4.1.7 Computation Performance (Ordering Nodes)

We create synthetic graphs with embedded small-world features to demonstrate the performance of our multiscale Fiedler vector implementation in GreenCurve. The sizes of the graphs range from 500 to slightly over one million nodes. More examples using real-life data sets are demonstrated in the case study later in Section 6. The compiled C++ code is running on a modest Dell Precision 650 with an Intel Xeon 2.8 GHz CPU and 4 GB of memory.

Table 2 shows a performance comparison (in wall clock seconds) between the multiscale algorithm used in GreenCurve (GC in Table 2), the Lanczos algorithm, and the *dsyev* routine from LAPACK. LAPACK is selected because of its simplicity and popularity. It merely gives the readers a benchmark reference to quantify the computation requirement of GreenCurve. Because GreenCurve reuses the multiscale hierarchy generated by the other Have Green tools, we report its creation time in a separate (middle) column in Table 2. Bear in mind that we only need to construct the multiscale graph hierarchy once for every graph ingested into Have Green.

4.2 Generating the Graph Visualization

While the first half of the GreenCurve discussion is computationally heavy, it is the second half that actually draws the visualization. After all the nodes within a sparse graph receive unique sequential numbers based on their connectivity (as discussed in Section 3.1), GreenCurve folds the graph nodes into a space-filling fractal curve in the order of the sequential numbers.

4.2.1 Hilbert Curve

There are a number of proven fractal curves [26] that can be applied to our design. Fig. 3 shows two prevailing examples—a Peano curve and a Hilbert curve—in the literature. Because the design of GreenCurve relies on distance proximity to approximate the graph node connections, the ideal fractal curve candidate should possess a

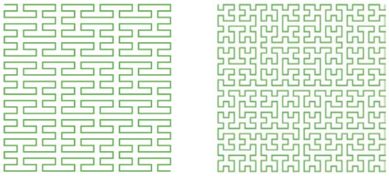


Fig. 3. A Peano (left) and a Hilbert curve (right).

TABLE 3
Time (Measured in Wall Clock Seconds) to Construct Hilbert Curves with up to 10 Million Nodes

Number of Graph Nodes	Recursive Hilbert Generator	Non-recursive Hilbert Generator
1,000	0.000045s	0.000123s
10,000	0.000302s	0.000998s
100,000	0.003000s	0.013900s
1,000,000	0.054300s	0.079200s
10,000,000	0.503000s	0.997000s

high degree of spatial coherence [29]. (The coherence level of a fractal curve is defined as the amount in which neighboring pixels are at sequential positions on the curve.)

4.2.2 Implementation and Performance

We have implemented both a recursive (based on Wirth [35]) and a nonrecursive (based on Peitgen et al. [26]) Hilbert curve generator for GreenCurve. The former one uses four elegant recursion calls in traversing four 2D directions (i.e., left, right, up, and down) repeatedly to form the folding patterns. The implementation of the latter one is slightly more complicated. But it does not have the same stack memory requirement as the recursion one.

Performancewise, both implementations can generate a GreenCurve graph layout with over ten million nodes in less than one wall clock second on the same Dell Precision 650. Table 3 shows the computation results of our implementation.

4.2.3 A Simple Demonstration

We use a benchmark data set previously used in the graph drawing contest at the International Symposium on Graph Drawing 96 (GD96B) [13] to demonstrate the capability of GreenCurve. The relatively small graph, which contains 111 nodes and 193 links, is extracted from a telephone-call database of "800" numbers. We intentionally select a public domain graph with obvious structures that are well known to the community to assess the validity of our design. More complicated graph examples will be presented later in the case study and discussions in Sections 6 and 7 correspondingly. Fig. 4 shows a force-directed layout of GD96B.

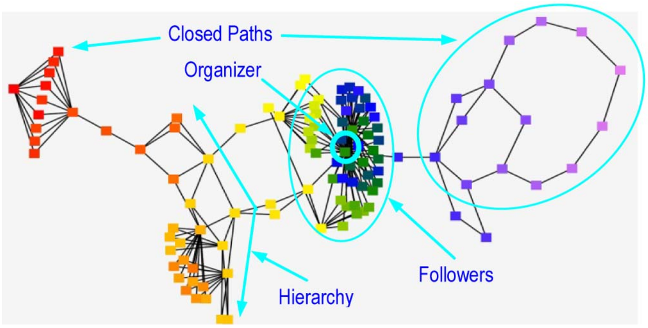


Fig. 4. A force-directed layout of GD96B.

Although the demo graph is small, it is rich in structural features often found in a small-world graph. For example, there are "closed-paths" in the upper right and upper left corners, and an "organizer" surrounded by a large number of "followers" in the middle of the graph as shown in Fig. 4. There is also a "hierarchy" in the lower left side as highlighted in the figure.

We first use GreenCurve to draw a force-directed layout of the graph in Fig. 5a. The graph nodes in Fig. 5a are color-coded to reflect the sequential order suggested by the Fiedler vector of the graph Laplacian. A quick comparison shows that the two closed paths in Fig. 4 are now shown in red and purple in two locations in Fig. 5a. Similarly, the hierarchy in Fig. 4 is shown in a shade of orange in Fig. 5a whereas the organizer/followers structure in Fig. 4 is either in yellow, green, or blue in Fig. 5a.

We then use GreenCurve to morph the changes between a force-directed layout to a GreenCurve layout from Figs. 5a to 5f. When comparing the two visualizations between Figs. 5a and 5f, one can easily see that all the major (color-coded) structural features stay intact during and after the morphing.

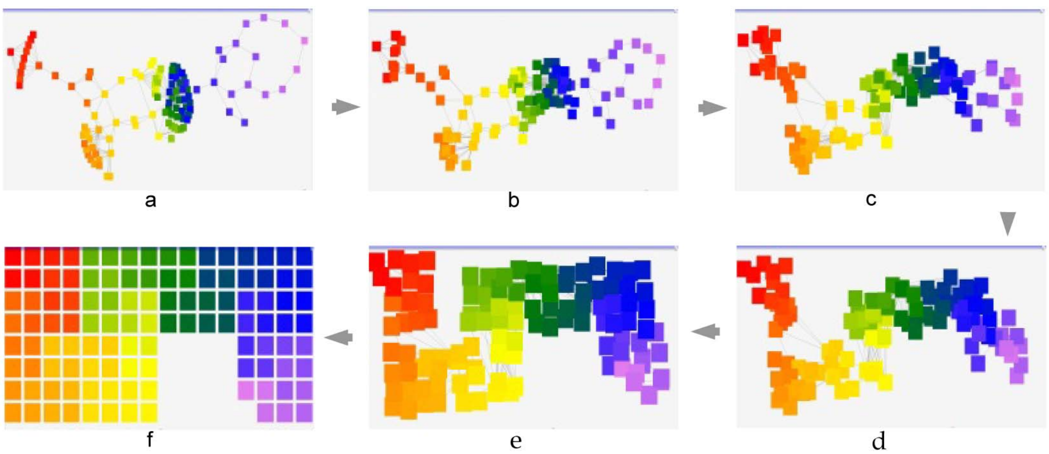
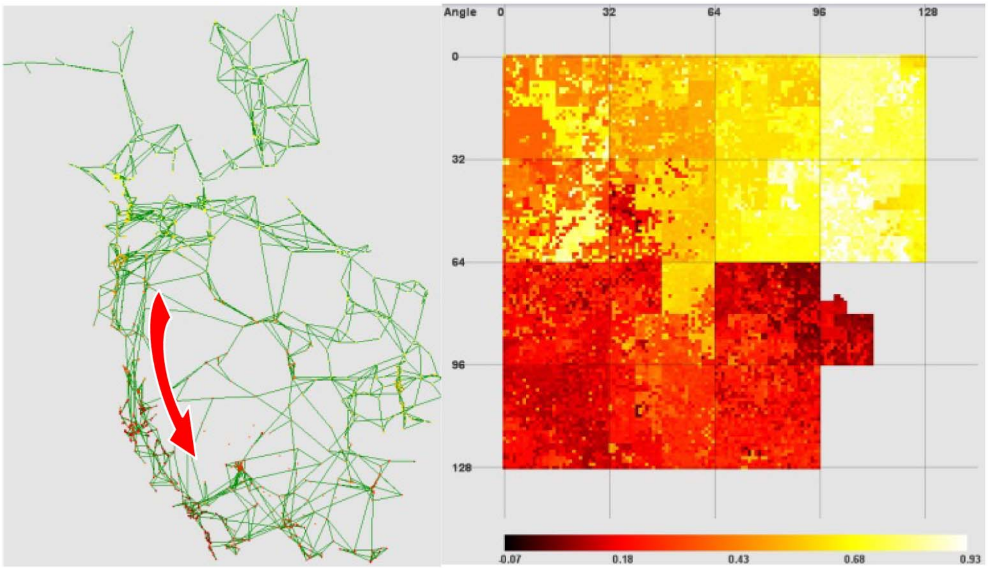


Fig. 5. A morphing of the GD96 graph from a force-directed layout (a) to a Hilbert curve layout (f) using GreenCurve. The rainbow colors (from red to purple) indicate the order of the graph nodes suggested by the Fiedler vector of the graph Laplacian.



a b

Fig. 6. WECC power grid in (a) graph layout and (b) GreenCurve layout.

The once congested green and blue areas in the middle of Fig. 5a, which represent the organizer/followers structure, are now spread out widely and evenly with guaranteed visibility in Fig. 5f. We can quickly determine that the green and blue nodes represent approximately one half of the graph node population in Fig. 5f. On the other hand, the same structural feature is poorly represented by the misleading visualization shown in Fig. 5a, which does not reflect the size of the corresponding node population.

Perhaps more importantly, the visualization footprint of Fig. 5f is potentially a lot smaller than the one in Fig. 5a. This design advantage gives us a big and certain head start toward visualizing a large number of multivariate information that ties to the graph. We will revisit this topic again in our case study discussion in Section 6.

5 Application-Oriented User Interfaces

Our discussion so far has focused mainly on the generation of the visualization. Depending on the applications, additional user interfaces are implemented to support the analysis of the underlying data using GreenCurve.

In our power grid application, for example, there is a slider bar that allows the user to morph the data set back and forth among a GreenCurve, a geographic map, and a force-directed graph layout. Our usability study, Section 7.2.4, has a brief discussion on this feature.

Additionally, interactive and animated tapered arrows are implemented to link all the visualization features (down to basic entity level such as nodes, links, their geographic locations, and their sensor readings). Readers are referred to [38] for more details on tapered arrow examples.

Finally, as GreenCurve is tightly integrated with GreenGrid [38] in our power grid application, all the visualization can be done in multiple spatial and geographical levels for both global and local analyses.

6 Power Grid Analytics Case Study

This case study was conducted independently by a power grid engineer, who is not a member of the GreenCurve development team, in the EIOC [8]. In the study, GreenCurve is applied to a power grid model to illustrate the validity and advantages of the method. The power grid model is a large network of over 14,000 nodes derived from the WECC [32] power grid, whose footprint encompasses a geographical area equivalent to over half of the United States as shown in Fig. 6a. One of the major tasks in operating power grids is to continuously monitor the electric quantities like node voltages and line flows throughout the network. Node voltages must be within a prespecified range, typically [0.9, 1.1] per unit. Power grid sensors measure the quantities and send them to power grid control centers where power grid operation tools process the data and visualize them on a geographical-map-based display for operator monitoring. The display is updated in every few seconds.

From the network perspective, the WECC power grid is highly heterogeneous as some parts of the grid are much denser than others. The network density is very well correlated with population and electricity sources. For example, major metropolitan areas in California are major load centers because of their high electricity consumption, and the Pacific Northwest has abundant hydropower resources from the river system. These two parts of the power grid are much more developed and therefore denser. These heterogeneous characteristics pose a great challenge in visualizing the power grid status on a network graph. The nodes in dense areas have significant overlap and are difficult to view on the display. This problem results in a false indication of power grid stability.

An example is illustrated in Fig. 6a, where voltage phase angles are visualized. The underlying power grid data simulate heavy summertime use. The power grid is heavily stressed with a large amount of electricity generated in the

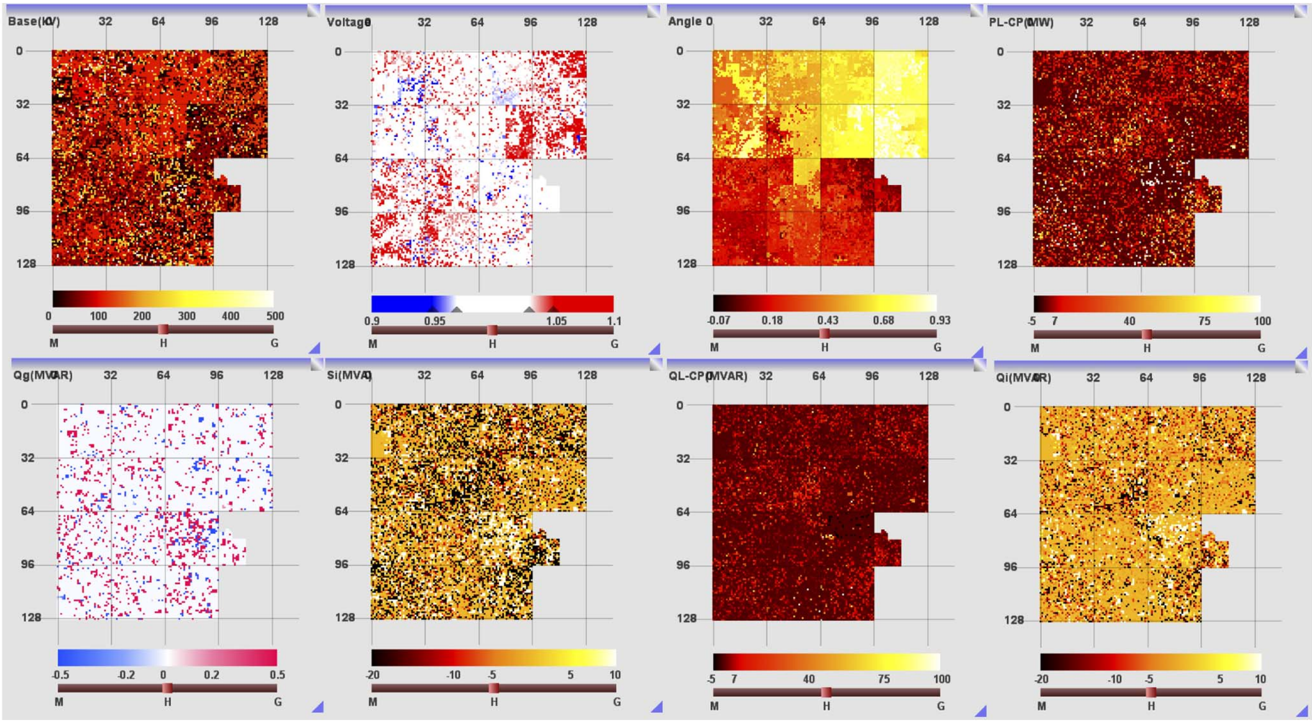


Fig. 7. Multiparameter displays for maximum screen utilization.

Pacific Northwest and transferred to California (illustrated by the red arrow in Fig. 6a) to meet the high demand due to hot weather. High stress translates to large angle separation, and thus we should see more of the colors of the two extreme ends on the display: yellow in the Pacific Northwest and red in California. Even though a large part of the power grid is stressed, the display only exhibits a small concentration of the yellow and red on the map and is not visually effective for operators to gauge the stability status of the grid. Fig. 6b visualizes the same data using the GreenCurve technique. GreenCurve spreads out the color on the display as each node gets an equal share of the screen. With this display, power grid operators can consistently determine the degree of the grid's stability using rules like "the more yellow and red, the less stable the system is."

Besides phase angles, many other quantities are of interest to operators in determining the stability status and devising remedial actions when necessary. The other quantities include voltage magnitudes, generator real power output ("Pg"), generator reactive power output ("Qg"), real power flow ("PL"), reactive line flow ("QL"), real load power ("Pi"), and reactive power load ("Qi"), as shown in Fig. 7. Normalized voltage magnitudes should have a value around 1.0 per unit, shown as white in Fig. 7. Voltage magnitudes out of the normal range are red or blue. Low voltage areas are usually heavy load centers, and high voltage areas are generation centers. Voltage angles and magnitudes are correlated, but they have different indications in power grid operations. Voltage angles mainly indicate the levels of real power of generation and load, while voltage magnitudes mainly indicate the levels of reactive power of generation and loads. Unlike the other quantities, which are related to nodes, line flows are related to links. Heavy line flows are indications that operators need to adjust power flow to unload the line—otherwise, the line would be overheated and tripped off by protection devices; electricity customers would lose power; and the power system would be exposed to the possibility of cascading blackouts. Lines with heavy flows can be related back to nodes to identify locations for power flow pattern adjustment. Therefore, these quantities need to be examined at the same time, and grid operations need to consider the collective observations. As shown in Fig. 7, all the quantities can be shown simultaneously on a single screen; with geographic maps, it is impossible to show the same number of quantities simultaneously with acceptable legibility.

7 Usability Study

We conducted a usability evaluation of GreenCurve in the EIOC [8] using a controlled experiment. For the evaluation, we assembled five network analysis problems that participants were to solve by analyzing power grid data using a force-directed graph layout and a GreenCurve layout. The analysis problems were provided by EIOC power grid engineers as authentic and common analysis problems that they would solve using network analysis tools. We selected the force-directed graph layout for comparison because it was developed and available to power engineers in an existing network analysis tool suite. A motivation for GreenCurve is to replace or supplement the existing force-directed graph layout capability.

The objective of the controlled experiment was to assess and compare the usefulness and usability of the GreenCurve and force-directed graph layout approaches. We expected that both layouts would have particular advantages and drawbacks and wished to explore those conditions and situations that the individual or combined layout approaches best served. In this paper, we have touted specific advantages and benefits of GreenCurve, including

TABLE 4
Controlled Experiment Problem Descriptions

Problem Number	Problem Description	Salient Feature(s)	Hypothesis
1	Given visualization of power grid, estimate number of electrical generators (nodes) in power grid.	Node visibility, intuitive visual paradigm	GreenCurve will achieve greater accuracy than force-directed layout for estimating node counts because it visually exposes every node, but will require more time to use because visual paradigm is less familiar.
2	Given visualization of power grid where nodes are weighted by phase angle values, identify the two regions of generators with low phase angle values (blue regions) and one region of generators with high phase angle values (red region).	Node placement accuracy, intuitive visual paradigm	GreenCurve will be similar in accuracy to force-directed layout for identifying node regions because node placement accuracy is comparable, but will require more time to use because visual paradigm is less intuitive.
3	Given visualization of power grid where nodes are weighted by phase angle values, estimate overall percentages of electrical generators with low (red), medium (white), and high (blue) phase angle values.	Node visibility, intuitive visual paradigm	GreenCurve will achieve greater accuracy than force-directed layout for estimating relative population sizes because it visually exposes every node, but will require more time to use because visual paradigm is less intuitive.
4	Given visualizations of power grid showing conditions before and after a particular event occurs and where nodes are weighted by phase angle values, estimate percentage increase or decrease of electrical generators with low (red), medium (white), and high (blue) phase angle values.	Node visibility, footprint uniformity, intuitive visual paradigm	GreenCurve will achieve greater accuracy than force-directed layout for distinguishing variations across populations because it visually exposes every node and provides a uniform footprint, but will require more time to use because visual paradigm is less intuitive.
5	Given visualization of power grid where nodes are weighted by phase angle values, estimate number of generators (nodes) in each of the three islands of the power grid.	Node visibility, graph connectivity, intuitive visual paradigm	Combined GreenCurve and force-directed layout will achieve higher performance than either layout alone for estimating the node counts of graph components because GreenCurve visually exposes every node while force-directed layout shows graph structure and connectivity. The combined method should be faster to use since the analysis task becomes much more difficult to solve if using just one of the individual methods.

guaranteed visibility of every node, compact footprint, and high fidelity to the traditional node-link graph layout. A potential weakness of GreenCurve is its inability to show graph structure and connectivity, because edge information and details are lacking from its visualization. Another potential weakness of GreenCurve is the lack of intuitiveness of its visual paradigm. We expect participants will be unfamiliar with the relationships, patterns, and organization of the GreenCurve visualization much more than those of the force-directed layout, which most, if not all, power engineers have seen and used.

Through the controlled experiment, we seek to evaluate how well GreenCurve supports its intended benefits and use as well as evaluate its perceived weaknesses in the context of authentic network analysis problems and in comparison to a force-directed layout. The set of network analysis problems is described in Table 4 along with the salient features and hypotheses that will be evaluated. The specific features under evaluation are node visibility, node placement accuracy, uniform footprint, and visual paradigm of GreenCurve and the graph structure and connectivity and visual paradigm of the force-directed layout. We hypothesize that GreenCurve should result in improved accuracy and user satisfaction over the force-directed layout due to its enhanced visual features, while requiring more time for task completion due to the lack of intuitiveness of its visual paradigm. We expect participants will require more time and effort to orient, map, and comprehend the GreenCurve visualization in the context of the network analysis problems.

7.1 Participants, Setup, and Procedures

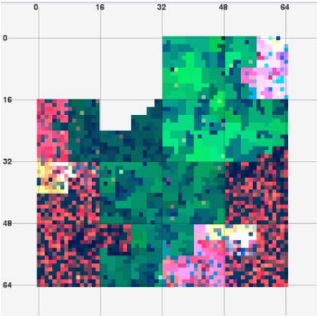
Twelve participants participated in the usability evaluation. Due to strict security and compliance requirements of a power grid center, we could only invite those who have access to the facility. Careful measures were made later on to ensure a fair and unbiased statistical analysis of the small population data (see Section 7.2 for details). None of the participants had prior knowledge of or experience with GreenCurve. However, they were generally familiar with traditional node-link graph layouts.

Participants took part in individual sessions that lasted 20 to 30 minutes. Evaluations were conducted in the EIOC using a Windows desktop computer on which the GreenCurve software had been installed. Participants interacted with the software mainly through the use of a mouse. The GreenCurve software is capable of displaying network data in a GreenCurve or force-directed layout.

Each usability session involved a single participant and a single evaluator. During a session, the participant was directed to work through the five different network analysis problems, one problem at a time, using both the GreenCurve and force-directed layouts.

The data presented in the problems were variations of the WECC power grid data we previously described. We generated new variations of the data for each problem so that participants could not draw from the analyses of previous problems. The nodes in both layouts were color-coded to represent voltage level or phase angle values. A colormap legend was displayed to the participant for each problem.

For Problems 1-4, participants were tasked to solve each of the problems twice—once using the GreenCurve layout



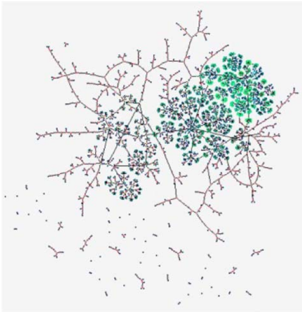


Fig. 8. Sample GreenCurve (left) and force-directed (right) graph visualizations of electric power grid data as applied in controlled experiment. The node colors are assigned based on a rainbow colormap.

and once using the force-directed layout. Six of the participants were asked to solve the set of problems using the force-directed layout first, while the other six were asked to use GreenCurve first. Each participant would alternate the layout that was applied first across the different problems. For Problem 5, participants were tasked to solve the problem three times using the GreenCurve layout, force-directed layout, and the two layouts in combination. For this problem, the order of application was randomly selected by the evaluator.

Given that each participant applied multiple tools to the same problem, the controlled experiment was of a repeated measures design with the layout tool serving as the independent variable. Fig. 8 presents sample GreenCurve and force-directed graph visualizations that are typical of those analyzed during the controlled experiment.

The performance measures collected were solution accuracy, task completion time, and user satisfaction. For every problem with the exception of Problem 2, participants were to estimate quantities. In these cases, the solution accuracy A of the estimate was computed by

$$A = 1 - \frac{|X - Y|}{X},$$

where X is the actual value and Y is the estimated value. For Problem 2, solution accuracy was measured as the percentage of targets or regions (of three possible) correctly identified. Task completion time was measured as the time interval from when the participant signaled being ready to solve the problem to when he or she signaled the completion of the task. Time required reading and understanding the problem was not factored into the task completion time. For user satisfaction, participants rated their satisfaction with each layout toward the solving of each problem using a 5-point scale (1—strongly disagree, 2—mildly disagree, 3—neither agree nor disagree, 4—mildly agree, 5—strongly agree).

7.2 Experimental Results

We applied standard statistical tests based on the properties of the various experiments [5]. To analyze the effect of the layout tool on solution accuracy in Problems 1, 3, and 4, and task completion time in Problems 1-4, we applied the standard t-test for correlated samples because only two conditions (GreenCurve and force-directed layout) existed for each problem. For Problem 5, a one-way Analysis of Variance (ANOVA) for correlated samples was applied because the experiment consisted of more than two conditions: 1) GreenCurve, 2) force-directed layout, and 3) GreenCurve and force-directed layout combined. For Problem 2, accuracy was limited to three possible results based on how many of three regions were correctly identified. The nonparametric Wilcoxon Signed-Rank Test [34] was appropriate in this case, because two conditions (GreenCurve and force-directed layout) existed and the results did not follow a normal distribution.

As for user satisfaction measures, participant scores were limited to a 1-5 satisfaction rating. For Problems 1-4, the nonparametric Wilcoxon Signed-Rank Test was again appropriate because two conditions (GreenCurve and force-directed layout) existed and the results did not follow a normal distribution. For Problem 5, the nonparametric Friedman Test [10] was applied because more than two conditions existed (GreenCurve, force-directed layout, and GreenCurve and force-directed layout combined) and the results did not follow a normal distribution.

With all statistical tests, we sought the universally accepted 0.05 level of significance [33]. Experimental results for each problem and measure are presented in Table 5 showing mean scores, standard deviations, p-values, and associated significance tests.

7.2.1 Node Visibility

Characteristic of small-world graphs, the graphs of Problems 1 and 3 had very dense regions separated by large sparse regions. The GreenCurve layout visually exposed every node, while nodes were obscured in the dense regions of the force-directed layout. GreenCurve's guaranteed visibility of every node allowed participants to better estimate the number of total nodes for Problem 1 and the relative population sizes of Problem 3. As they were better able to solve the problems using GreenCurve, participants were more satisfied with the GreenCurve layout over the force-directed layout for both problems. The differences in accuracy and user satisfaction for both problems were found to be significant. More recent work on graph drawing, such as [7], may help improve overall node visibility as compared to the force-directed layout used in this study.

7.2.2 Node Placement Accuracy

For Problem 2, we hypothesized that GreenCurve and the force-directed layout would perform comparably because they both had similar node placement accuracy. Participants were able to fully solve Problem 2 using either layout, because GreenCurve is able to maintain the locality of nodes comparable to the force-directed layout. The red and blue regions were well delineated in both visualizations. The difference in accuracy between the two layouts was not found to be significant.

As for user satisfaction, GreenCurve was rated lower on average than the force-directed layout. Participants expressed that the GreenCurve representation is not as familiar or intuitive as a graph. Participants grew more accustomed to the GreenCurve layout as they worked through the evaluation problems, but they never achieved the level of comfort they have with graphs that they see and use in their day-to-day activities. The difference in user satisfaction ratings was relatively small but still significant.

TABLE 5
The Mean Scores, Standard Deviations, p-Values, and Associated Significance Test Results for the Measures of Solution Accuracy, Satisfaction Rating, and Task Completion Time

Problem Number	Condition	Solution Accuracy (%)				Task Completion Time (s)				Satisfaction Rating (1-5)			
		μ	σ	p-value	Test	μ	σ	p-value	Test	μ	σ	p-value	Test
1	GreenCurve	79%	29%	<0.001	1	27.70	16.65	<0.05	1	3.90	0.32	<0.005	2
	Force-Directed	37%	21%			15.70	8.39			1.60	0.70		
2	GreenCurve	100%	0%	>0.05	1	29.08	12.08	<0.05	1	3.33	0.97	<0.05	2
	Force-Directed	100%	0%			20.17	6.22			4.00	0.47		
3	GreenCurve	79%	13%	<0.05	1	23.80	11.14	<0.05	1	3.70	0.48	<0.01	2
	Force-Directed	63%	13%			15.90	6.76			2.40	0.52		
4	GreenCurve	78%	17%	<0.05	1	31.30	15.14	<0.0001	1	3.70	0.82	<0.01	2
	Force-Directed	62%	12%			24.60	7.92			2.20	0.92		
5	GreenCurve	25%	38%	<0.05	3	7.30	3.68	<0.0005	3	2.00	1.25	<0.005	4
	Force-Directed	51%	19%			15.90	13.62			2.30	0.82		
	Combined	83%	17%			18.70	8.43			4.10	0.57		

Test 1: t-test, correlated samples. Test 2: Wilcoxon Signed-Rank. Test 3: 1-way ANOVA, correlated samples. Test 4: Friedman.

7.2.3 Uniform Footprint

Problem 4 required comparing two visualizations of different data sets using the same layout and distinguishing variations across them. Participants found that variations in the colored areas across GreenCurve visualizations were easily seen. Participants had greater difficulty in detecting the variations across the force-directed graph visualizations, because the variations were occurring in dense regions of the graph where they were shifting in density but not in area or size. In contrast, the GreenCurve visualization shifted in size and shape but not in density. Participants noted difficulty in analyzing white nodes in both layouts, because they blended into the background and were hard to detect. A general guideline in the use of either layout would be to minimize the amount of light colors in the colormap.

Participants found that the visual cues and overall visual effect were more prominent in the GreenCurve layout than in the force-directed layout. Problem 4 benefited from GreenCurve's guaranteed visibility of every node and its footprint, which provided a fixed and uniform spatial context for comparison. Achieving more accurate results using GreenCurve, participants noted higher satisfaction with the GreenCurve layout over the force-directed layout. The differences in accuracy and user satisfaction for Problem 4 were found to be significant.

7.2.4 Graph Structure and Connectivity

Effectively solving Problem 5 requires the ability to view graph connectivity as well as the entire data set in order to make a reasonable estimate of population. Neither of the individual layouts alone seems sufficient to solve the problem. The accuracy of the GreenCurve layout was low because identifying connected components within the Hilbert Curve is not generally supported in the current implementation. A possible refinement to the GreenCurve layout would be to visually demarcate connected components within the Hilbert curve. In the case of Problem 5, the islands of the graph could be shown as partitioned segments. Using the force-directed layout alone suffers from the same issues as described in Problem 1, where estimating node counts is difficult because nodes may be obscured in dense regions of the graph.

For Problem 5, participants repeated the task three times using GreenCurve, force-directed layout, and the two layouts in combination. The GreenCurve software has a slider bar that allows the user to morph the data set back and forth between a GreenCurve and force-directed graph layout. In combining the use of both layouts, participants were able to view both graph connectivity and all the nodes in full resolution. From this, participants were able to derive more accurate estimates and gain greater satisfaction using the combined layouts over using the individual layouts alone in solving Problem 5. The differences in accuracy and user satisfaction were found to be significant.

7.2.5 Intuitive Visual Paradigm

For Problems 1-4, participants required more time to complete the tasks using GreenCurve compared to the force-directed layout. As participants conducted analysis, they had to recall the relationships and meanings of the Hilbert curve pattern, adjacent blocks, colored blocks, and different block sizes in order to fully utilize the data and visualization. This required much more cognitive effort than the force-directed layout, which was very familiar and natural to most participants. As previously mentioned, we believe that the nonintuitiveness of GreenCurve also negatively affected user satisfaction with the technique. The differences in task completion times between the two layouts were found to be significant for Problems 1-4.

Our hypothesis for Problem 5 that the combined use of the two techniques would result in a faster task completion time than the individual techniques was not observed as the average task completion time was higher for the combined technique than either of the individual techniques. In this case, participants were choosing to settle quickly on a rough solution using an individual technique versus performing a more intricate, thorough, and accurate analysis using the combined technique. In effect, the participants settled on a less optimal solution with the individual techniques than they were able achieve with the combined technique as illustrated in the Problem 5 accuracy results shown in Fig. 9. The differences in task completion times among the three approaches of Problem 5 were found to be significant.

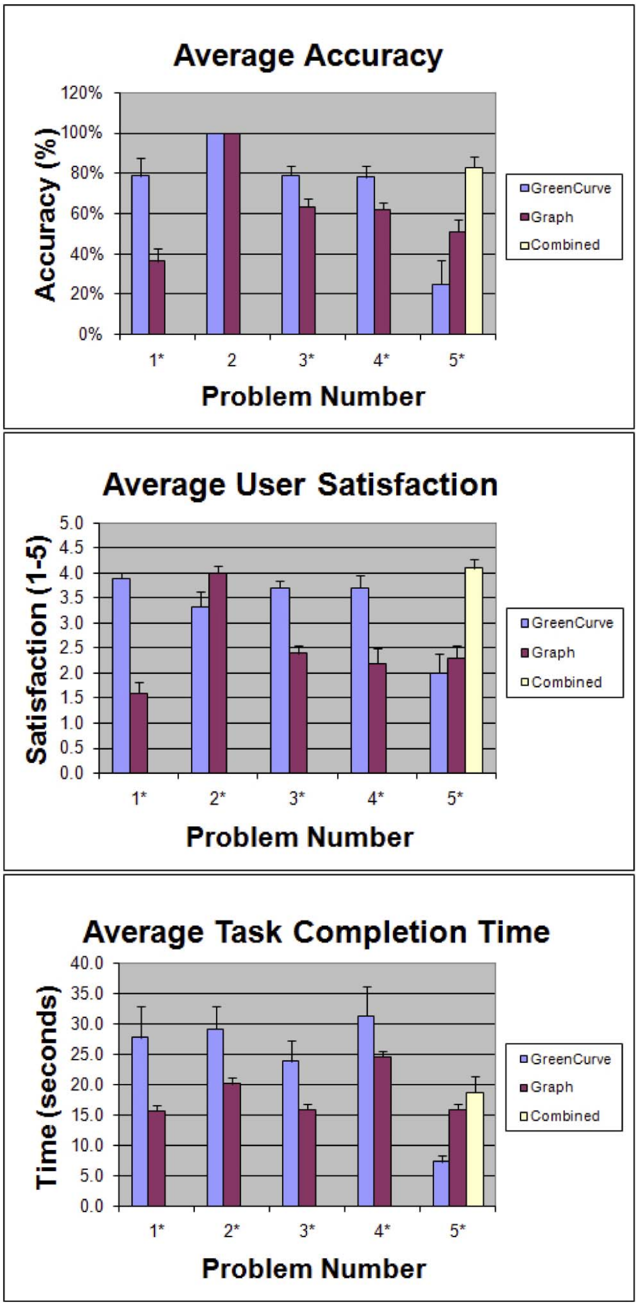


Fig. 9. Controlled experiment results from five electric power grid analysis problems showing average and standard error values for the measures of accuracy (top), average user satisfaction (middle), and average task completion time (bottom). Problem numbers denoted with asterisk indicate p-values <0.05.

7.2.6 Summary of Evaluation Results

Fig. 9 presents a summary of our usability evaluation results. As shown, the GreenCurve layout offers comparable or improved performance over the force-directed layout in terms of accuracy and user satisfaction for most problems in head-to-head comparisons. The exceptions are part of Problem 2 where user satisfaction was found to be higher for the force-directed layout and Problem 5 where the combined use of the GreenCurve and force-directed layouts proved to be extremely effective. Problem 5 illustrates how GreenCurve may complement other techniques and tools to solve a wider range and more challenging set of network analysis problems.

In terms of task completion time, the GreenCurve layout consistently required more time to analyze than the force-directed layout in head-to-head comparisons as hypothesized. For Problem 5, however, the combined technique did require more time to complete the task than the GreenCurve or force-directed layout alone, but this was because participants settled more quickly on less optimal solutions using individual techniques rather than what would have been achieved through the combined technique. To a large degree, the evaluation results confirm that GreenCurve generally meets the intended design goals, benefits, and uses, while also exhibiting specific weaknesses.

8 Conclusion and Future Work

GreenCurve represents an intriguing and innovative compact visualization technique for displaying multivariate small-world graphs. It employs a space-filling fractal-based approach that is distinct from traditional graph representations such as matrices and node-link graph layouts. Compactness is achieved by ordering the neighboring nodes of a graph by solving the Fiedler vector of its graph Laplacian and then folding the nodes into a fractal curve based on the Fielder vector.

Given our application of GreenCurve to the visualization of electric power grids, GreenCurve is required to be highly interactive, computationally efficient, and capable of processing and displaying large-scale, high-volume data. The design and implementation of GreenCurve required intensive matrix and linear algebraic computation to accomplish the interactive time and computational requirements imposed by power grid applications. We have demonstrated that GreenCurve can identify the Fiedler vector of a graph with over 100,000 nodes in less than one second as well as process and visualize very large small-world graphs of up to one million nodes.

The resultant GreenCurve visual representation offers many compelling and practical features, including a compact footprint that supports multivariate analysis, and guaranteed visibility of every node. In this paper, we have shown how these specific features have improved power grid analytics in the real-world analysis of the WECC power grid. GreenCurve's small footprint and general support for multiparameter displays was particularly insightful and useful for comprehending and analyzing the full range of conditions that must be correlated and merged to assess the overall stability of a power grid.

We have also verified the capabilities and usefulness of GreenCurve in a controlled experiment, where participants have found that they are able to solve critical network analysis problems more accurately and with higher satisfaction than using a force-directed graph layout. The controlled experiment also showed that the GreenCurve and the force-directed graph layouts are highly complementary for specific kinds of network problems.

We are applying GreenCurve to other infrastructure applications such as those related to computer networks as well as exploring its use in other scientific and engineering fields. In the process, we continue to evolve and refine the

GreenCurve technique as we apply it to new problem areas and domains.

Acknowledgments

This work has been supported by the National Visualization and Analytics Center$^{\mathrm{TM}}$(NVAC$^{\mathrm{TM}}$) located at the Pacific Northwest National Laboratory in Richland, WA and the US Department of Energy Office of Electricity Delivery and Energy Reliability (DOE-OE). NVAC is sponsored by the US Department of Homeland Security (DHS) Science and Technology (S&T) Division. The Pacific Northwest National Laboratory is managed for the US Department of Energy by Battelle Memorial Institute under Contract DE-AC05-76RL01830. We would like to dedicate this paper to the lives and accomplishments of our colleagues, Jim Thomas and Harlan Foote.

References

[1] S.T. Barnard and H.D. Simon, "A Fast Multilevel Implementation of Recursive Spectral Bisection for Partitioning Unstructured Problems," *Concurrency: Practice and Experience,* vol. 6, no. 2, pp. 101-117, Apr. 1994.

[2] S.K. Card, J.D. Mackinlay, and B. Shneiderman, *Readings in Information Visualization, Using Vision to Think.* Morgan Kaufmann, 1999.

[3] C. Chen, *Information Visualization beyond the Horizon,* second ed. Springer, 2006.

[4] J.K. Cullum and R.A. Willoughby, *Lanczos Algorithms for Large Symmetric Eigenvalue Computations: Theory.* SIAM, 2002.

[5] J. Devore, *Probability and Statistics for Engineering and the Sciences,* sixth ed. Brooks/Cole, 2007.

[6] G. Di Battista, P. Eades, R. Tamassia, and I.G. Tollis, *Graph Drawing: Algorithms for the Visualization of Graphs.* Prentice Hall, 1999.

[7] T. Dwyer and L. Nachmanson, "Fast Edge-Routing for Large Graphs," *Proc. 17th Int'l Symp. Graph Drawing,* pp. 147-158, 2009.

[8] Electricity Infrastructure Operation Centers (EIOC), http://wioc.pnl.gov, 2010.

[9] M. Fiedler, "Laplacian of Graphs and Algebraic Connectivity," *Combinatorics and Graph Theory,* vol. 25, pp. 57-70, 1989.

[10] M. Friedman, "The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance," *J. Am. Statistical Assoc.,* vol. 32, no. 200, pp. 675-701, 1937.

[11] J.D. Glover and M.S. Sarma, *Power System Analysis and Design,* third ed. Brooks/Cole, 2002.

[12] *GD2011, 19th In'l Symp. Graph Drawing 2011,* http://www.win.tue.nl/GD2011/, 2011.

[13] GD96 Graph Drawing Contest, http://vlado.fmf.uni-lj.si/pub/networks/data/gd/gd96/B96.gph, 2011.

[14] D. Harel and Y. Koren, "A Fast Multi-Scale Algorithm for Drawing Large Graphs," *J. Graph Algorithms and Applications,* vol. 6, no. 3, pp. 179-202, 2003.

[15] Herman, G. Melancon, and M.S. Marshall, "Graph Visualization and Navigation in Information Visualization: A Survey," *IEEE Trans. Visualization and Computer Graphics,* vol. 6, no. 1, pp. 24-43, Jan.-Mar. 2000.

[16] M. Holzrichter and S. Oliveira, "A Graph Based Method for Generating the Fiedler Vector of Irregular Problems," *Proc. Int'l Parallel and Distributed Processing Symp./Int'l Parallel Processing Symp. (IPDPS/IPPS),* pp. 978-985, 1999.

[17] *IEEE InfoVis 2011, IEEE Conf. Information Visualization,* http://www.visweek.org/visweek/2011/info/infovis-welcome/infovis-welcome, 2011.

[18] *J. Graph Algorithms and Applications,* http://jgaa.info, 2011.

[19] LAPACK, http://www.netlib.org/lapack, 2011.

[20] M. Manguoglu, "A Highly Efficient Parallel Algorithm for Computing the Fiedler Vector," CoRR abs/1008.1700, 2010.

[21] S. Milgram, "The Small-World Problem," *Psychology Today,* vol. 2, pp. 60-67, 1967.

[22] C. Muelder and K.-L. Ma, "Rapid Graph Layout Using Space Filling Curves," *IEEE Trans. Visualization and Computer Graphics,* vol. 14, no. 6, pp. 1301-1308, Nov./Dec. 2008.

[23] C. Mueller, B. Martin, and A. Lumsdaine, "A Comparison of Vertex Ordering Algorithms for Large Graph Visualization," *Proc. Asia Pacific Symp. Information Visualisation (APVIS),* 2007.

[24] C.C. Paige and M.A. Saunders, "Solution of Sparse Indefinite Systems of Linear Equations," *SIAM J. Numerical Analysis,* vol. 12, no. 4, pp. 617-629, 1975.

[25] B.N. Parlett, *The Symmetric Eigenvalue Problem.* SIAM, 1987.

[26] H.-O. Peitgen, H. Jürgens, and D. Saupe, *Chaos and Fractals, New Frontiers of Science.* Springer, 1993.

[27] TreeMap,http://www.cs.umd.edu/hcil/treemap, 2011.

[28] UMFPACK,http://www.cise.ufl.edu/research/sparse/umfpack, 2011.

[29] D. Voorhies, "Space-Filling Curves and A Measure of Coherence," *Graphics Gems,* pp. 26-30, Academic Press, 1991.

[30] C. Walshaw, "A Multilevel Algorithm for Forced-Directed Graph-Drawing," *J. Graph Algorithms and Applications,* vol. 7, no. 3, pp. 253-285, 2003.

[31] S. Wasserman and K. Faust, *Social Network Analysis-Methods and Applications.* Cambridge Univ. Press, 1999.

[32] Western Electricity Coordinating Council (WECC), http://www.wecc.biz, 2011.

[33] Why P = 0.05?,http://www.jerrydalla.com/LHSP/p05.htm. 2011.

[34] F. Wilcoxon, "Individual Comparisons by Ranking Methods," *Biometrics Bull.,* vol. 1, no. 6, pp. 80-83, Dec. 1945.

[35] N. Wirth, Algorithms +Data Structures = Programs, Prentice-Hall, 1976.

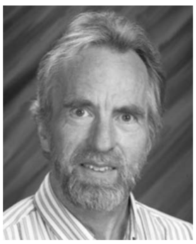
[36] P.C. Wong, G. Chin JR, H. Foote, P. Mackey, and J. Thomas, "Have Green-A Visual Analytics Framework for Large Semantic Graphs," *Proc. IEEE Symp. Visual Analytics Science and Technology,* pp. 67-74, Oct. 2006.

[37] P.C. Wong, H. Foote, P. Mackey, G. Chin JR, H. Sofia, and J. Thomas, "A Dynamic Multiscale Magnifying Tool for Exploring Large Sparse Graphs," *Information Visualization,* vol. 7, no. 2, pp. 105-117, 2008.

[38] P.C. Wong, K. Schneider, P. Mackey, H. Foote, G. Chin JR, R. Guttromson, and J. Thomas, "A Novel Visualization Technique for Electric Power Analytics," *IEEE Trans. Visualization and Computer Graphics,* vol. 15, no. 3, pp. 410-423, May/June 2009.

[39] P.C. Wong, K.K. Wong, H. Foote, and J. Thomas, "Global Visualization and Alignments of Whole Bacteria Genomes," *IEEE Trans. Visualization and Computer Graphics,* vol. 9, no. 3, pp. 361-371, July-Sept. 2003.

Pak Chung Wong received the PhD degree in computer science from the University of New Hampshire in 1997. He is a chief scientist and project manager at the Pacific Northwest National Laboratory. He serves on the editorial board of IEEE Computer Graphics and Applications and will cochair IEEE VisWeek Conference in 2012. His current research interests include visual analytics, extreme scale data analytics, scientific visualization, graph analytics, and multimedia analytics. He is a member of the IEEE and the IEEE Computer Society.

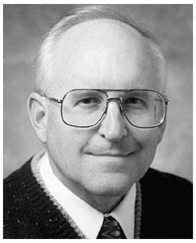


Harlan Foote received the BS degree in physics from Washington State University in 1966. He was a senior research scientist at the Pacific Northwest National Laboratory, where he worked on a wide variety of remote sensing projects. His interests included hyperspectral image classification, multiscale image processing for stereo matching, multisensor data fusion, and feature extraction from 3D millimeter wave holographic images. He passed away in February 2010.

Patrick Mackey received the BS degree in computer science from Washington State University in 2004 and is working toward the MS degree in computer science at the same university. He is a research scientist at the Pacific Northwest National Laboratory, where he has worked on multiple visual analytics projects. His research interests include visualization, scientific computation, and computer graphics.

George Chin received the PhD degree in computer science from Virginia Tech. He is a chief scientist at the Pacific Northwest National Laboratory. His main area of expertise is in human-computer interaction. He has conducted extensive user and work studies with intelligence analysts and scientists from various domains. Other current research interests include visual analytics, computer-supported collaborative work, social networks, scientific workflow, and scientific problem-solving environments. He is a member of the IEEE and the IEEE Computer Society.

Zhenyu Huang received the PhD degree from Tsinghua University, Beijing, China in 1999. From 1998 to 2002, he conducted research at the University of Alberta, McGill University, and the University of Hong Kong. Currently, he is a senior research engineer at the Pacific Northwest National Laboratory, Richland, WA. His research interests include power system stability and control, high-performance computing applications, and power system signal processing. He is a senior member of the IEEE and the IEEE Computer Society.



Jim Thomas received the MS degree in computer science from Washington State University. He was a PNNL lab fellow and chief scientist for Information Technologies at the Pacific Northwest National Laboratory. He was the director of the US Department of Homeland Security National Visualization and Analytics Center. He specialized in the research, design, and implementation of innovative information and scientific visualization, multimedia, analytics, and human computer interaction technology. He passed away in August 2010. He is a member of the IEEE and the IEEE Computer Society.