Confronting the Challenges of Graphs and Networks

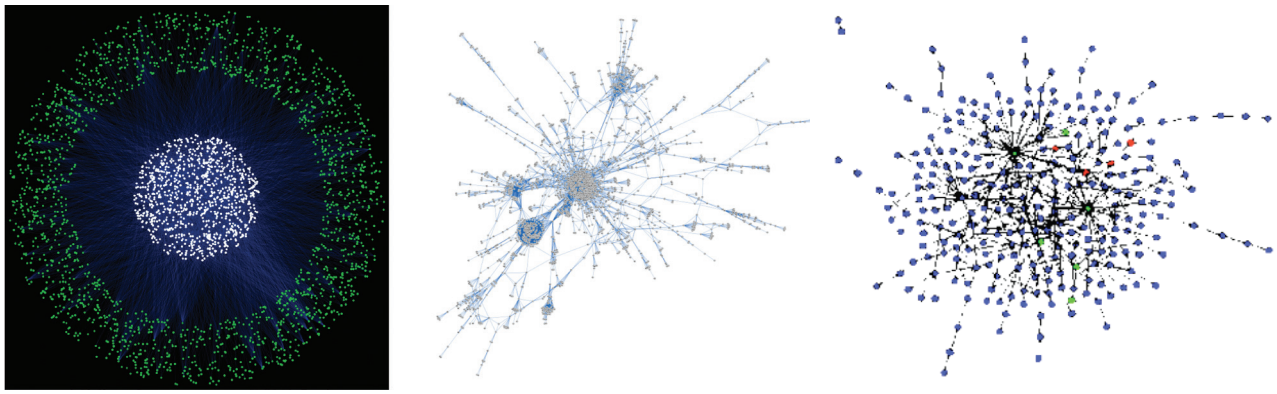
Nadya T. Bliss and Matthew C. Schmidt

New application needs, combined with the rapidly increasing sizes of datasets, are driving the development of a new field at the intersection of computer science, mathematics, signal processing, and the social sciences. Lincoln Laboratory has been pioneering research to address the challenges of using graphs and networks to exploit these vast databases. This issue of the *Lincoln Laboratory Journal* focuses on some of this innovative work.

Many current and emerging Department of Defense (DoD) challenges require the analysis of very large datasets containing millions, billions, or even trillions of entities. While the sheer number of entities presents a significant challenge by itself, the analysis task is further complicated by the fact that the activities, objects, or individuals of interest are commonly tightly embedded within large, noisy background data, have few or no individual distinguishing characteristics, and are changing rapidly.

One way to address the challenge of identifying subtle events in large datasets is to consider not only individual entities and their attributes but also relationships between them. Considering relationships naturally leads to the analysis of graphs and networks. Simply, a graph is a mathematical representation of a set of entities (vertices) and their relationships (edges between those vertices). In this issue and other literature, the terms *graph* and *network* are often used interchangeably. One way to distinguish between those terms is to use the word *graph* when referring to the formal mathematical structure and the word *network* when referring to the specific instantiation. For example, a set of computers and the communications between them form a network. This network can be represented by a graph whose vertices represent the computers and whose edges represent the communication between computers.

Graph analysis techniques can be used in a broad range of applications in which graphs and networks arise naturally (Figure 1). In the cyber security domain, graphs representing computer networks, such as the ones described in the previous paragraph and Figure 1a, can be analyzed to identify cyber threats. For intelligence, sur-



(a) (b) (c)

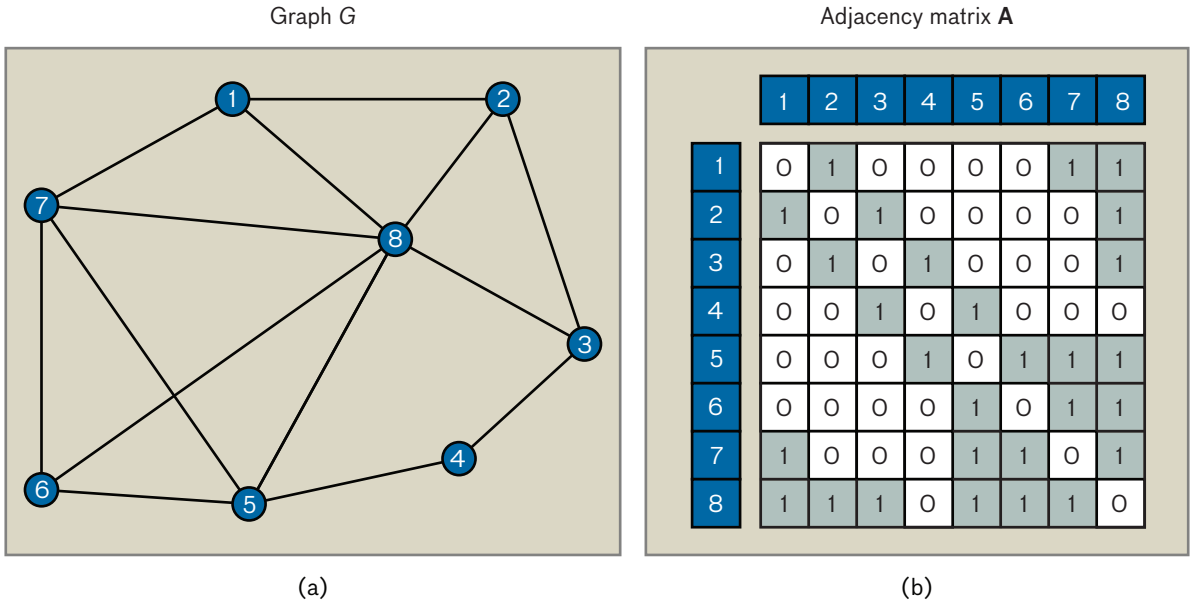
FIGURE 1. Examples of graphs from various applications. (a) is a graph constructed from computer network traffic data. Analysis of computer networks enables detection of anomalous cyber events and suspicious activity. (b) is a co-author graph constructed from a publications database. Analysis of such networks enables detection of emerging research trends or collaboration groups. Finally, (c) is a vehicle track network constructed from data collected by an imaging sensor. Vehicle track networks enable patterns-of-life analysis.

veillance, and reconnaissance (ISR) objectives, vehicle networks derived from sensor data can be analyzed as graphs, as illustrated in Figure 1c. Analysis of social networks derived from multiple data sources can facilitate the identification of groups, organizational structure, or incipient trends and activities. In biology, the analysis of graphs representing networks of protein interactions can lead to the design of medications that target biochemical pathways of interest. Graphs can also be exploited for the analysis of datasets in which the network of interest is not as well defined. For example, a set of documents may not seem as naturally suited to graph analysis as a computer or social network; however, patterns and trends in the documents can be identified by analyzing a graph of relationships between documents that mention the same topics, locations, or individuals.

While leveraging relationships enables significant capabilities and allows for detection of activities that would not be detectable through the analysis of entities alone, the analysis of graphs presents a number of technical challenges:

- Detection theory for graphs is a new area and, for most scenarios, well-defined performance bounds do not exist. Specifically, for most network datasets, there is no rigorous way to specify what type of patterns will stand out (i.e., be detectable) and what type of patterns will be subsumed in the noise.
- Few truthed datasets of relevant scales exist that allow for rigorous evaluation of detection techniques on graph data. The definitions of backgrounds (noise and normal patterns) and foregrounds (target and anomalous patterns) are still being formalized.
- Relationships of interest to many applications are highly dynamic. Dealing with large-scale dynamic graphs is an emerging research area.
- Often, many graphs can be constructed from a given dataset. Determining what graph representation will be the most effective for a particular task is highly non-trivial.
- It is desirable to be able to quickly construct multiple different graphs from the same dataset or from multiple datasets. However, storage, representation, and data-access techniques are often hard-coded, making this a difficult, error-prone, and timely task.
- While many graph algorithms have computational complexity on the order of the number of entities squared, the efficiency of those algorithms often yields performance on the order of 1/100 to 1/1000% of peak performance of a processing platform. Many factors contribute to this inefficiency, for example, sparsity of data and poor data locality of operations.

As demonstrated by the articles in this special issue of the *Lincoln Laboratory Journal*, Lincoln Laboratory has been pioneering research to address many of the aforemen-



	1	2	3	4	5	6	7	8
1	0	1	0	0	0	0	1	1
2	1	0	1	0	0	0	0	1
3	0	1	0	1	0	0	0	1
4	0	0	1	0	1	0	0	0
5	0	0	0	1	0	1	1	1
6	0	0	0	0	1	0	1	1
7	1	0	0	0	1	1	0	1
8	1	1	1	0	1	1	1	0

FIGURE 2. A simple graph (a) and its matrix equivalent (b). The eight vertices (circles) in the graph each have a corresponding row and column in the matrix. Similarly, each edge (line) connecting vertex *i* and vertex *j* in the graph corresponds to a non-zero entry at index (*i*, *j*) and index (*j*, *i*) in the matrix.

tioned challenges. Together, the technical challenges and the application relevance of graph analysis are precisely the reasons that make this an area both well suited to and fascinating for a research agenda of a national laboratory.

Definitions

All articles in this issue describe algorithms, representations, and processing techniques for graphs and networks. In this section, basic definitions of a few common graph terms are highlighted. These definitions appear in many computer science textbooks, for example, the text by Cormen, Leiserson, and Rivest [1].

A graph (G) is simply a set of vertices (V) and edges (E). An example graph is presented in Figure 2a. A graph can also be represented as an adjacency matrix $\mathbf{A}$ (Figure 2b), in which entry $\mathbf{A}(i,j)$ is nonzero if an edge exists between vertex i and vertex j. The graph in Figure 2 is a simple graph. A simple graph is undirected (that is, its edges have no orientation) and unweighted (its edges have no attributes associated with them). A few basic graphs and their matrix equivalents are presented in Table 1.

Overview of Graph Research

Researchers, particularly mathematicians, have been interested in graphs for hundreds of years. In 1735, Leonhard Euler published what is considered to be the first graph theory paper on the seven bridges of Königsberg problem. This problem asked if the seven bridges spanning the Preger River in the city of Königsberg could be traversed in a single round-trip without retracing any of the route [2].

Table 1. Basic Graph Types and Related Matrices

GRAPH *G*	EQUIVALENT MATRIX **A**
is simple and unweighted	$\mathbf{A}(i,j)$ = 0 or 1
is weighted	$\mathbf{A}(i,j)$ = weight
is simple and undirected	**A** is symmetric, $\mathbf{A}(i,j) = \mathbf{A}(j,i)$
is directed	**A** could be symmetric $\mathbf{A}(i,j)$ does not imply $\mathbf{A}(j,i)$
is multipath	**A** is multidimensional

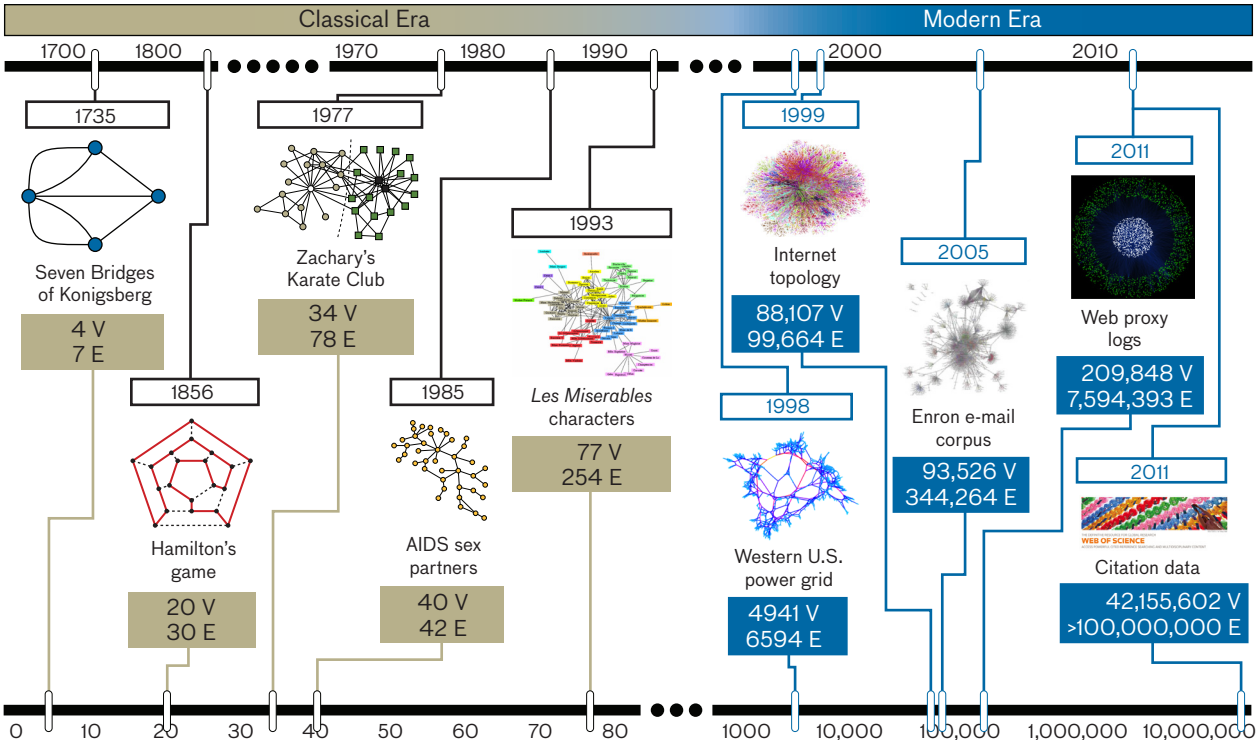


FIGURE 3. A timeline demonstrating the recent explosion in scale of graph datasets that has distinguished the "Modern Era" (2000 to present) of graph analysis from the "Classical Era" (pre-2000). The date each dataset was first analyzed is marked on the timeline at the top of the figure, and the number of vertices in the graph is marked along the line at the bottom. The number of vertices (V) and edges (E) in each dataset are noted within the chart.

Euler represented the network of land masses connected by bridges as a graph and proved that it was not possible to create such a route. Since then, graphs have been used to interpret and exploit data in diverse fields. The timeline in Figure 3 highlights a number of famous dataset analyses and illustrates the growth both in the size of datasets and the complexity of their representative graphs.

The 2000s, particularly the last five to seven years, have ushered in a new chapter in the history of graph theory (Modern Era in Figure 3). The timeline shows that the scales of datasets have evolved significantly and the modern graph era has witnessed graphs with more than a million vertices. Many problems in classical graph theory, such as the seven bridges of Königsberg problem, can often be analyzed by hand and thus are well suited to the study and optimization of combinatorial and traditional graph traversal algorithms. However, as the graphs become larger, the need arises for new techniques capable of processing datasets on the order of millions, billions, and beyond entities.

In the context of such enormous datasets, traditional techniques, such as those based on path computations and search, are no longer tractable. Furthermore, the applications of interest have been changing as well—from defining the shortest paths between two cities and identifying maximum flows on networks to the detection of anomalies, patterns, and trends, often in rapidly changing environments. Finally, while in the past, matrix-based techniques, such as presented in [3], have provided insight into some numerical properties of graphs (such as spectrum characteristics), they have just recently emerged as powerful tools for application-driven tasks such as community detection [4] and signal processing for graph-based data [5].

These new application needs, combined with the current and emerging scales of datasets, are driving the development of a new field that is at the intersection of computer science, mathematics, signal processing, and, often, social, behavioral, and biological sciences.

Table 2. Summary of Articles in Journal 20(1)

AUTHORS	ARTICLE FOCUS AREA	KEY CONTRIBUTIONS
Miller, Bliss, Wolfe, & Beard	Signal processing framework for graph-based data	General framework for detection of signals in graph-based data Detection algorithm with linear computational complexity Performance results on both simulated and real data (social network, vehicle tracks)
Yee, Philips, Condon, Jones, Kao, Smith, Anderson, & Waugh	Algorithms for cued network discovery	Novel threat propagation algorithm Demonstration of algorithm on both simulation (graph generators and vehicle track simulator) and real data (e-mail corpus)
Smith, Senne, Philips, Kao, & Bernstein	Performance bounds for cued network detection	Novel Bayesian detection framework Neyman-Pearson-like optimality analysis Generative model for covert networks
Campbell, Dagli, & Weinstein	Social network analysis	End-to-end approach to social network analysis Comparison of community detection techniques and role prediction Analysis of community dynamics in context of reality mining dataset
Kepner, Ricke, & Hutchison	Associative array and graph analytics framework	Parallel algebraic interface to triple stores to enable rapid prototyping of graph analytics Ease of use and performance results on DNA sequence comparison dataset
Song, Kepner, Gleyzer, Nguyen, & Kramer	Efficient, scalable graph processing architecture	Architectural innovations co-designed with graph algorithms Simulated performance results demonstrating four orders of magnitude improvement over current alternatives
Cho & Snavely	Imagery analysis and 3D scene reconstruction	Framework for 3D scene reconstruction Application of graph-based techniques to imagery exploitation Demonstration of techniques on various imagery datasets

In This Issue

The articles in this issue cover a range of technologies that are making significant progress in addressing the technical challenges inherent in large graphs and networks. Individual articles highlight these technologies as employed in specific applications. Table 2 summarizes the key contributions of articles in this issue.

More work remains to be done. Datasets are only getting larger and more diverse. New application domains require increasingly complex processing under highly dynamic conditions. Sensor and information processing platforms are becoming smaller and increasingly more size, weight, and power constrained. System-level solutions coupling together all of the individual technologies, from processing architectures to algorithmic frameworks, will ensure our nation's competitiveness and agility in the face of various emerging threats, whether these are threats to cyber security, to critical infrastructure such as the electrical grid, to satellite communications, or to any military or civilian domain that may be vulnerable to malicious intrusions. ■

References

1. T.H. Cormen, C.E. Leiserson, and R.L. Rivest, *Introduction to Algorithms*. Cambridge, Mass.: MIT Press, 1990.
2. D.B. West, *Introduction to Graph Theory. 2nd Edition.* Englewood Cliffs, N.J.: Prentice Hall, 2001.
3. F.R.K. Chung, *Spectral Graph Theory*. Providence, R.I.: American Mathematical Society, 1997.
4. M.E.J. Newman, "Modularity and Community Structure in Networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, 2006, pp. 8577–8582.
5. B.A. Miller, N.T. Bliss, and P.J. Wolfe, "Toward Signal Processing Theory for Graphs and Non-Euclidean Data," *Proceedings of the 2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 5414–5417.

About the Authors

Nadya T. Bliss is the Director of Strategic Projects Development at Arizona State University (ASU). In that role, she is responsible for developing and maintaining strategic relationships with funding agencies, working with university leaders on developing large-scale, interdisciplinary proposals, and monitoring the university's strategic investments. She also leads and participates in sponsored research. Prior to joining ASU, she spent 10 years at Lincoln Laboratory, most recently as the founding group leader of the Computing and Analytics Group. In that capacity, she developed, led, and managed research initiatives in advanced analytics, high-performance computing systems, and computer architectures to address the challenges facing the Department of Defense and the intelligence community. In 2011, she received the inaugural MIT Lincoln Laboratory Early Career Technical Achievement Award, recognizing her work in parallel computing, computer architectures, and graph processing algorithms and her leadership in anomaly detection in graph-based data. She has authored or co-authored 60 publications and presentations, holds a patent, and has served as a chair and member of multiple technical and organizational committees. She received bachelor's and master's degrees in computer science from Cornell University and is a Senior Member of IEEE and a member of the Association for Computing Machinery.

Matthew C. Schmidt is currently a technical staff member of the Computing and Analytics Group at Lincoln Laboratory. His work includes research into the use of graphs and graph analytics to cue analysts to interesting and relevant information, activities, and patterns in large-scale datasets. This work makes use of a general framework developed at the Laboratory for addressing "Big Data" challenges through a computing architecture consisting of graph analytics, composable high-level languages, distributed data storage and indexing, and high-performance parallel processing. He received a bachelor's degree in computer science and mathematics from Purdue University in 2004 and earned his doctoral degree from North Carolina State University in 2011. His doctoral thesis research was on the use of distributed graph algorithms to analyze biological networks.