Graph-Theoretic Analysis of Complex Stochastic Networks

David H. Collins* Aparna V. Huzurbazar

Abstract

We describe the use of graphs to represent Markovian and semi-Markovian stochastic networks, and how the interplay of visual and algebraic properties of the graphs aids in the determination of quantities such as first passage times. We also show how Markovian stochastic processes are special cases of generalized flowgraphs, with other special cases ranging from transportation networks to representations of finite automata. This leads to the formulation of general algorithms for graph reduction that can be expressed in terms of either visual or matrix-algebraic properties of the graph.

Key Words: Graph, Flowgraph, Markovian, Laplacian, Resistance distance

1. Introduction

In this paper we survey the use of graphs to represent networks of entities which are, in some sense, stochastic. Emphasis is on the use of several complementary representations of graphs to aid in conceptualizing problems and determining quantities such as first passage times. We use the terms "graph" and "network" interchangeably.[1]

We consider finite graphs $\mathcal{G}$ with vertex sets $V(\mathcal{G}) = \{1, 2, \ldots, n\}$ and edge sets $E(\mathcal{G})$ with elements (i, j) (ordered pair) if $\mathcal{G}$ is a directed graph, and $\{i, j\}$ (unordered pair) if it is undirected. Unless otherwise indicated, we assume that $\mathcal{G}$ has no self-loops, and there is at most one edge joining any two vertices i, j.

Stochastic networks imply the existence of mappings that associate graph vertices and edges with random variables (RVs). Specifics of the association may vary—e.g., in the graph of a stochastic process, vertices represent states, with the associated random variables being probabilities of occupying a given state; edges represent transitions between states, with the associated RVs being waiting times prior to the transitions. For simplicity, where no confusion will result, we will talk about relations (e.g., independence) between edges and vertices as a shorthand for the corresponding relations between RVs.

The interplay of topological, algebraic, and visual properties of graphs provides a powerful set of tools for guiding intuition and solving problems that can be represented using graphs. This is illustrated by a huge variety of stochastic graph applications: multistate process analysis (Markov and semi-Markov models) (Huzurbazar (2005)); communications and computer networks (Satyanarayana and Hagstrom (1981)); reliability analysis, using representations such as reliability block diagrams and cut/path sets (Rausand and Høyland (2004)); electrical and electronic circuits represented as signal flow graphs (Mason (1953), Doyle and Snell (1984)); social network analysis (Wasserman and Faust (1994)); epidemiology (Marathe and Vullikanti (2013)); project planning graphs (Pritsker (1977)); causal graphical models (Pearl (2009); Markov random fields (Koller and Friedman (2009)); and many others. In addition to inherently discrete graph models, discrete graph models may be useful analogues of models in continuous spaces; e.g., a diffusion process is the continuum limit of a discrete random walk (Ghez (2010)).

*dcollins@lanl.gov; both authors are with the Statistical Sciences Group, Los Alamos National Laboratory, Los Alamos, New Mexico, USA.

[1]Some authors distinguish the two, but we have not found any definition of "network" that is used consistently.

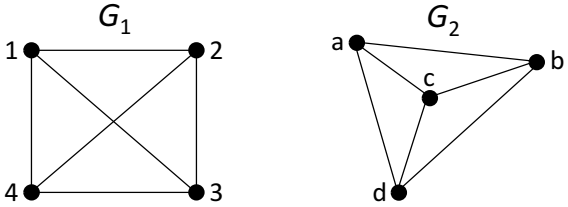


Figure 1: The graphs $\mathcal{G}_1$ and $\mathcal{G}_2$.

1.1 Power of graph representations

As a mathematical discipline, graph theory is quite abstract, and many important results have no obvious physical interpretation. Yet two leading mathematical graph theorists, Hartsfield and Ringel (2003), say "[Graph] vertices are elements of an abstract set, and the edges are pairs of elements of the set. However, every graph theorist in the world visualizes the graphs as structures where the vertices are points and the edges are lines." As an example of this point, consider the following two graphs: $\mathcal{G}_1$, with vertex set $V(\mathcal{G}_1) = \{1, 2, 3, 4\}$ and edge set $E(\mathcal{G}_1) = \{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\}$; and $\mathcal{G}_2$, with $V(\mathcal{G}_2) = \{a, b, c, d\}$ and $E(\mathcal{G}_2) = \{(a, b), (a, c), (a, d), (b, c), (b, d), (c, d)\}$. Now, is $\mathcal{G}_1$ a planar graph? Are $\mathcal{G}_1$ and $\mathcal{G}_2$ isomorphic? These are not difficult questions to answer from the abstract set representations, but verifying the answers is certainly not trivial. Yet the answers are immediately apparent from a visual representation (see Figure 1). One can see that both graphs are projections of the tetrahedron onto the plane, thus planar (of course planarity, the ability to be drawn on the plane with no line crossings, is an inherently visual property). It is also obvious that both $\mathcal{G}_1$ and $\mathcal{G}_2$ are complete graphs on four vertices, thus isomorphic.

In Section 3 we discuss other representations of graphs, using matrices. The ability to select the representation(s) that best allow reasoning about properties of interest is a key reason why graph theory has such a wide range of applications.

2. Stochastic networks

Stochastic networks are directed or undirected, possibly cyclic, graphs in which vertices represent states or entities, and edges represent transitions between states or relations between entities. Edges may be temporal or spatial, representing the time/distance for a transition between states, or a property of the relation between entities. Entity graphs may be temporally static, e.g., a graph representing road distance between cities. States may be vector- or graph-valued; e.g., in a graph representing the evolution of a social networks, each vertex may be a graph representing the state of the network at a given point in time.

Edges in a stochastic network are, in some sense, stochastic; for example, an edge may represent the random waiting time to make a transition between states, or the random distance (or cost) to traverse space between destinations represented as graph vertices. As examples, consider communications networks, social networks, epidemic maps, or reliability block diagrams, in which the vertices represent entities; and disease progression, hardware degradation, language parsing, and wildfire progression, where edges may be thought of as transitions between states of a progress, or (for disease and wildfire progression) movement between spatial points to which the process has advanced.

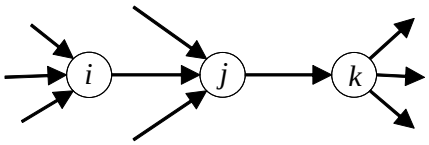


Figure 2: The Markov property.

2.1 Markovian networks

The most desirable stochastic network representations are those that are reasonably faithful to the problem domain, while remaining mathematically tractable. In most cases, the property characterizing such networks is that they are *Markovian*.

The general property characterizing a Markovian network is that any two disjoint sets of vertices are conditionally independent given a separating subset (a separating subset $\mathcal{S} \subset V(\mathcal{G})$ partitions $V(\mathcal{G})$ into $\mathcal{A}, \mathcal{S}, \mathcal{B}$ such that $\mathcal{A} \cap \mathcal{S} \cap \mathcal{B} = \emptyset$, and every path between a vertex in $\mathcal{A}$ and a vertex in $\mathcal{B}$ passes through $\mathcal{S}$). In a *first-order* Markovian network, the smallest separating subset between any two disjoint sets of vertices contains a single vertex. Unless stated otherwise, we assume the first-order property.

Figure 2 illustrates the first-order Markov property in a directed graph—given that the system occupies, say, state j, its further evolution does not depend on the fact that the previous state is i, or on any of the previous states; in this case the separating subset is just $\{j\}$. Markov random fields (Koller and Friedman (2009)) are examples of undirected Markovian networks.

A temporal statement of the Markov property is that the future is independent of the past, given the present (a temporal network implies a directed graph, but many important cases are reversible, i.e., the network has the same statistical characteristics if all the directed edges (arrows) in the graph are reversed (Kelly (1979)). Though this property may seem limiting, it has great power, as this well-known quote from Laplace illustrates: "We may regard the present state of the universe as the effect of its past and the cause of its future" Laplace (1951); all the deterministic dynamical systems of classical physics are Markovian.

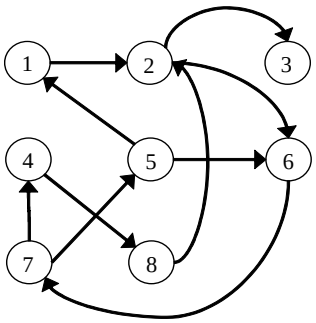
Important examples of Markovian multistate models include

- Discrete time Markov chain (DTMC): transition times identically constant $\equiv 1$, completely determined by the transition probabilities p_{ij}.
- Continuous time Markov chain (CTMC): transition times $\sim \exp(\lambda_i)$, completely determined by transition rates λ_{ij}, $\sum_j \lambda_{ij} = \lambda_i$; $p_{ij} = \lambda_{ij}/\lambda_i$.
- Semi-Markov process (SMP): transition times arbitrarily distributed $\sim F_{ij}$, completely determined by p_{ij} and F_{ij}.

Note that the SMP class includes CTMCs, where $F_{ij} \sim \exp(\lambda_i)$, which in turn includes DTMCs, where $F_{ij} \sim \exp(\delta(0))$. These may also be used to model spatial networks, where instead of transition time distributions, we have spatial distance distributions.

3. Matrix representations of graphs

Kepner and Gilbert (2011) assert that "The duality between the canonical representation of graphs as abstract collections of vertices and edges and a sparse adjacency matrix representation has been a part of graph theory since its inception" (p. 3). This may not be



$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Figure 3: A directed graph and its adjacency matrix.

strictly true (graph theory is usually held to date from Euler's Königsberg bridge problem, a century before the formal development of matrix algebra), but certainly in modern times, matrices are indispensable for representing and solving large graph problems.

We summarize some of the important matrix representations of graphs. The *adjacency matrix* $\mathbf{A}$, where $a_{ij} = 1$ if there is an edge $i \to j$ expresses the (unweighted) connectivity of the graph. For a graph with edge weights w_{ij}, it is often convenient to define $a_{ij} = w_{ij}$. Note that $\mathbf{A}$ is symmetric for an undirected graph. Figure 3 shows a directed graph and its adjacency matrix. In a graph with n vertices, $\left[\sum_{k=1}^{n} \mathbf{A}^k\right]_{ij} = 1$ if i is reachable from j, where sums and products are taken in the Boolean sense. This can be used, for example, to evaluate connectivity in a message-passing network. If graph edges are weighted to represent, say, costs, then $\min_{m \le n} \left[\sum_{k=1}^{m} \mathbf{A}^k\right]_{ij} > 0$, where the product is taken in the conventional matrix sense, is the minimum cost to get from i to j.

The *degree matrix* $\mathbf{D}$ is a diagonal matrix, where d_{ii} is the degree of i (number of incident edges). For a directed graph, d_{ii} may be the in-degree, out-degree, or total incident degree, depending on the application. For a weighted graph, d_{ii} may be the total of weights on incident edges, i.e., $d_{ii} = \sum_i w_{ij}$ (or $d_{ii} = \sum_j w_{ji}$ for in-weights).

The *Laplacian matrix* $\mathbf{L} = \mathbf{D} - \mathbf{A}$ is the discrete analogue of the continuous Laplacian ∇^2 (Doyle and Snell (1984)). It is fundamental for solving discrete random walk and resistance distance problems, among others (cf. Section 5.1). There are other fruitful analogies to vector-calculus concepts as well: one can define discrete versions of the gradient, divergence, curl, potential function, and harmonic function on graphs, with the standard interrelationships; for example, the summed gradient of a potential is zero over any closed path in the graph. Jiang et al. (2011) provide a detailed introduction to these vector-calculus analogues.

Other useful matrix representations include the incidence matrix, where $\iota_{ij} = 1$ if edge e_j is incident on vertex v_j; the circuit matrix, relating circuits (closed paths with no duplicated vertices or edges) to edges; the cut-set matrix, relating cut sets to edges, where a cut set is a minimal set of edges that partitions the graph; and the path matrix, relating paths to edges. See, e.g., Busacker and Saaty (1965), for more information on these matrices and their applications to transportation networks, reliability analysis, etc.

3.1 Matrix representations of Markovian graphs

For a DTMC, the analogue of the adjacency matrix is the transition probability matrix $\mathbf{P} = [p_{ij}]$, where p_{ij} is the one-step $i \to j$ transition probability. $[\sum_{k=1}^{n} \mathbf{P}^n]_{i,j} = p_{ij}^{(n)}$ is the n-step $i \to j$ transition probability. This matrix is fundamental to all Markovian stochastic processes.

For a CTMC the transition rate matrix is $\mathbf{\Lambda} = [\lambda_{ij}]$, where λ_{ij} is the $i \to j$ transition rate. $\mathbf{P}$ is derivable from this using $p_{ij} = \lambda_{ij} / \sum_j \lambda_{ij}$. It follows from the constant exponential hazard (intensity) that we can solve for $\mathbf{P}(t)$, the matrix of time-dependent state occupancy probabilities, with a simple system of constant-coefficient linear differential equations (Ross (1996)):

$$\begin{aligned} \frac{d\mathbf{P}(t)}{dt} &= \mathbf{\Lambda}\mathbf{P}(t) \\ \mathbf{P}(t) &= \exp(t\mathbf{\Lambda}). \end{aligned}$$

The analogous formulation for an SMP is a system of integral renewal equations (Ross (1992)); these can be solved using Laplace transforms (see Section 4.3).

For an SMP, the Semi-Markov kernel is $\mathbf{Q} = q_{ij}$, where $q_{ij} = p_{ij}F_{ij}$ and F_{ij} is the $i \to j$ waiting time distribution. $[\mathbf{Q}^{*(n)}]_{ik}$ is the n-step $i \to k$ passage time distribution; $^{*(n)}$ represents the n-fold convolution product whose Laplace transform is $\mathfrak{L}^n$, where $\mathfrak{L}$ is the elementwise Laplace transform of $\mathbf{Q}$; see Warr and Collins (2012) and references therein for matrix methods applied to $\mathbf{Q}$ and $\mathfrak{L}[\mathbf{Q}]$.

4. Graph reduction in directed Markovian networks (flowgraphs)

The term "flowgraph" comes from electrical engineering, where these methods were originally developed (Mason (1953)). This is not the only productive electrical analogy for stochastic networks; see Section 5.1, and also Kelly (1979), Doyle and Snell (1984), and Grimmet (2010). Using the semi-Markov process as an example (Huzurbazar (2005), Collins and Huzurbazar (2013)), in the electrical network analogy the *transmittance* of an edge (i, j) in the directed graph is $p_{ij}\mathfrak{L}_{ij}(s)$, the Laplace transform of the corresponding element in the semi-Markov kernel (see Section 3.1).

Given a transmittance for each transition (edge), the flowgraph can be reduced to a single transmittance, the transform of the distribution of first passage times between any two states, using the following three graph reduction rules.

For transitions $i \to j \to k$ in series, where the waiting time in i before transition to j is the random variable X_{ij} and the $j \to k$ waiting time is X_{jk}, the $i \to k$ waiting time is $X_{ik} = X_{ij} + X_{jk}$, with probability density function $f_{ij} \star f_{jk}(x)$, the convolution of the $i \to j$ and $j \to k$ waiting time pdfs. Since the Laplace transform maps convolution to multiplication, the first reduction rule is: *The transmittance of transitions in series is the product of the series transmittances*. For example, in the graph in Figure 4, $\mathfrak{L}_{ik}(s) = \mathfrak{L}_{ij}(s)\mathfrak{L}_{jk}(s)$.

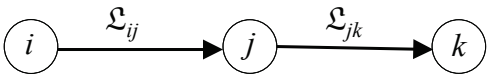


Figure 4: Flowgraph with serial transitions.

Figure 5 adds a second transition in parallel to the $i \to j$ transition. From the linearity of the Laplace transform, the second reduction rule immediately follows: *The transmittance of transitions in parallel is the sum of the parallel transmittances*. Applying Rule 2 to Figure 5, the $i \to j$ transmittance is $p\mathfrak{L}^*_{ij}(s) + (1-p)\mathfrak{L}_{ij}(s)$. Figure 6 shows the reduction of a graph with both series and parallel transitions.

Figure 7, adding a loop to the series flowgraph of Figure 4, illustrates the third reduction rule. The possible paths for a first passage from state i to state k are: $i \to j \to k$,

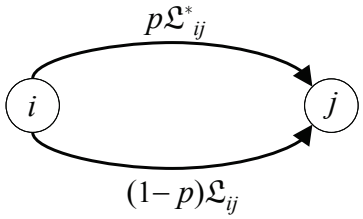


Figure 5: Flowgraph with parallel transitions.

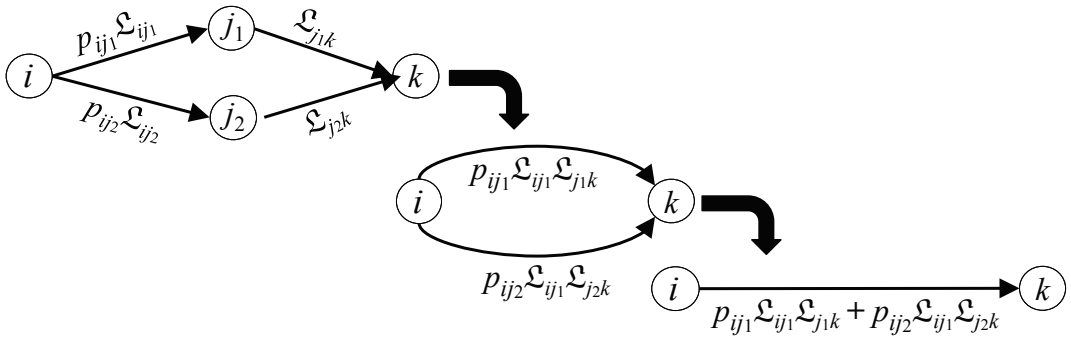


Figure 6: Graph reduction with series and parallel transitions.

$i \to j \to i \to j \to k$, $i \to j \to i \to j \to i \to j \to k, \ldots$, (zero or more traversals of the loop $i \to j \to i$ followed by a direct passage to k). The third rule reduces this to a single transmittance: *The transmittance of a loop is the sum of infinitely many parallel transmittances, one for each possible sequence of loop iterations*.

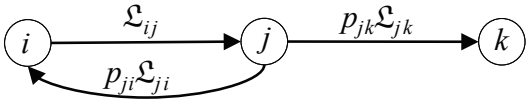


Figure 7: Flowgraph with feedback loop.

Applying this to Figure 7, the event of $i \to k$ first passage occurs given one of the events "$i \to j \to k$ occurs after n traversals of the $i \to j \to i$ loop," $n = 0, 1, \ldots$ Since these are parallel paths, we have

$$\mathfrak{L}_{ik}(s) = \sum_{n=0}^{\infty} \mathfrak{L}_{ij}(s) p_{jk} \mathfrak{L}_{jk}(s) [\mathfrak{L}_{ij}(s) p_{ji} \mathfrak{L}_{ji}(s)]^n = \frac{\mathfrak{L}_{ij}(s) p_{jk} \mathfrak{L}_{jk}(s)}{1 - \mathfrak{L}_{ij}(s) p_{ji} \mathfrak{L}_{jiu}(s)}. \tag{1}$$

Since $p_{ji} < 1$ (otherwise the loop would never be left), $|\mathfrak{L}_{ij}(s) p_{ji} \mathfrak{L}_{ji}(s)| < 1$ for s in some open interval about 0, which justifies the summation of (1) as a geometric series.

For directed graphs of any complexity, these rules can be applied in the form of a general algorithm called Mason's rule (Mason (1953, 1956), Huzurbazar (2005)) to compute the transform of the distribution of first passage times in Markovian graphs, or analogous quantities in other types of graphs.

This reduction algorithm generalizes to any graph whose edge weights combine with an appropriate algebra. Other examples include finite automata for parsing regular expressions (Kozen (1997), Csenki (2008), Csenki (2009)); software complexity analysis (McCabe and Butler (1989)); and multiplicative flowgraphs (see Sections 4.2 and 4.3).

4.1 Matrix example - movement of corporate bond ratings

Figure 8 shows a flowgraph for the year-to-year movement of Standard & Poor's ratings of U.S. corporate bonds; state labels are S & P bond ratings, with "D" being default of the obligation by its issuer. Markov transition rates for this graph were estimated from S & P credit reviews by Jarrow et al. (1997); interest lies in the cumulative probability of default for a bond with a given current rating. Given the number of states and transitions, solution by matrix methods is an alternative to the topological analysis of Section 4. See Collins and Huzurbazar (2013) for details of the solution, both by differential equations and by flowgraph reduction. On the right in Figure 8 is a typical output from the analysis.

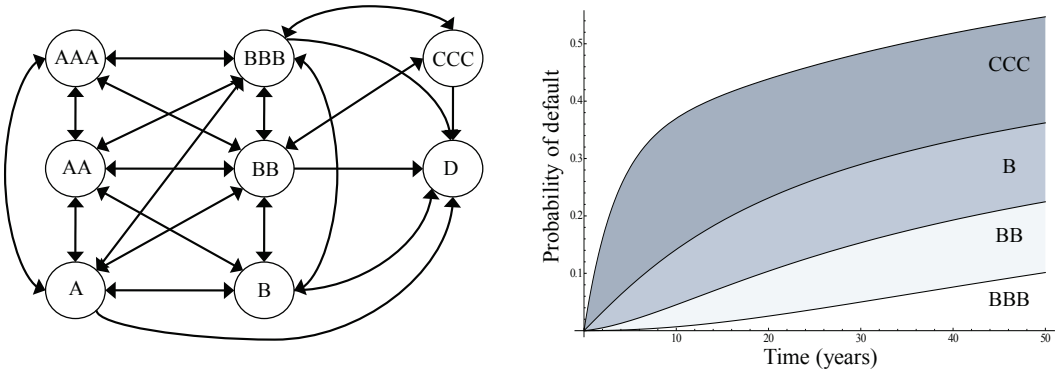


Figure 8: Left: flowgraph for movement of corporate bond ratings; right: cumulative probability of default for various initial ratings.

4.2 Multiplicative flowgraph example - circular arbitrage

Arbitrage, in finance, is the exploitation of a price imbalance between two or more markets in order to turn a profit. Circular arbitrage is buying and selling the same commodity to take advantage of a price imbalance (Ellerman (1984)). Figure 9 shows an example of circular arbitrage in a currency exchange market: a trader might exchange U.S. dollars (USD) for either British pounds (GBP) or Euros (EUR), then exchange that currency for Japanese yen (JPY), then exchange yen for dollars, in hopes of exploiting some exchange-rate imbalance to make a profit, which will occur if $(p_{12}r_{12}r_{24} + p_{13}r_{13}r_{34})r_{41} > 1$, where the p_{ij} are probabilities of exchanging currency i for currency j and the r_{ij} are stochastically varying exchange rates. Solving this network requires a convolution algebra that maps multiplication of random variables in the time domain to multiplication in the transform domain, which is provided by Mellin transforms. See Collins (2011) for more details on this and similar multiplicative networks.

4.3 Transform analysis of stochastic networks

In the additive flowgraph examples at the beginning of Section 4, the pdf of $T_1 + T_2$ is $f_{T_1} * f_{T_2}(z) = \int_0^\infty f_{T_1}(x) f_{T_2}(z - x)dx$; the convolution is mapped to multiplica-

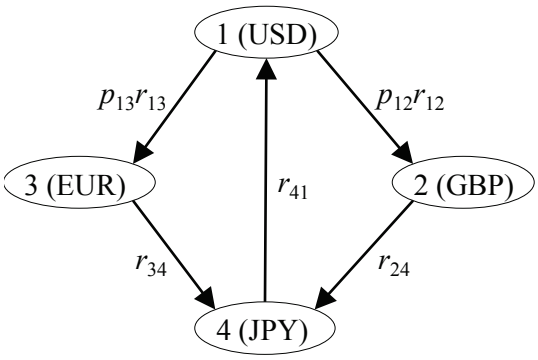


Figure 9: Multiplicative flowgraph for currency exchange arbitrage.

tion by the Fourier transform, discrete Fourier transform, Laplace transform, or moment generating function (MGF), and accurate numerical transform inversion algorithms exist to recover the pdf or CDF from the transform; see Huzurbazar (2005), Warr and Collins (2012). In the multiplicative flowgraph of Section 4.2, the pdf of $T_1 \cdot T_2$ is $f_{T_1} \circ f_{T_2}(z) = \int_0^\infty f_{T_1}(x) f_{T_2}\left(\frac{z}{x}\right) \frac{1}{x} dx$, which is mapped to multiplication by the Mellin transform (Springer (1979)). Analytic inversion of Mellin transforms is typically more difficult than inversion of the additive transforms, but numerical inversion algorithms exist (e.g., Theocaris and Chrysakis (1977)). As mentioned earlier, there are other possibilities as well, as long as there is an edge weighting with the right algebraic properties–e.g., graphs for parsing regular expressions (Csenki (2008), Csenki (2009)). Formulation of these graphs in terms of transforms is a future research topic.

5. Graph representations of complex systems

We present a few examples of modeling systems with potentially large numbers of states (or entities), and complex behavior. Though representing such systems in their entirety with visual graphs is often impossible, it is usually fruitful to begin with a graphic representation of a portion of the system, or a simplified abstraction, and use that to guide construction of a more complete model using, e.g., matrices.

5.1 Resistance distance in social networks

Social networks, recently prominent due to "social software" applications such as Facebook, have been studied analytically by sociologists and anthropologists for decades; e.g., see Wasserman and Faust (1994). An important question for such networks is to determine the strength of connections, or the "closeness" of network participants. If, for example, the left-hand side of Figure 10 shows relations (e.g., "likes") between actors in a network, Is Sarah closer to Jennifer than Andy? The classical distance measure in a network is based on the number of links in the shortest path between vertices. In this sense, the shortest path (fewest links) is from Sarah is to Jennifer. On the other hand, there are more paths from Sarah to Andy.

A measure that takes both factors into account is the *resistance distance*: replace each edge in the graph with a fixed resistor, then solve by Ohm's and Kirchoff's laws for the net resistance between any two edges (see the right side of the figure). By resistance distance, Sarah is closer to Andy—it is easy to show that if all resistors are 1Ω, the resistance distance from Sarah to Jennifer is 2Ω, and from Sarah to Andy is $\leq 6/5\Omega$. Note that this graph is

drawn as undirected, but in general a social link may be directed because it is stronger in one direction than the other. More generally, resistances can vary stochastically depending on the closeness/strength of individual links.

Other applications of resistance distance include chemical structure analysis (Klein and Randič (1993)), message routing in networks (Tizghadam and Leon-Garcia (2010)), similarity measures for document search (Wang and Hauskrecht (2010)), and ecological modeling (McRae et al. (2008))

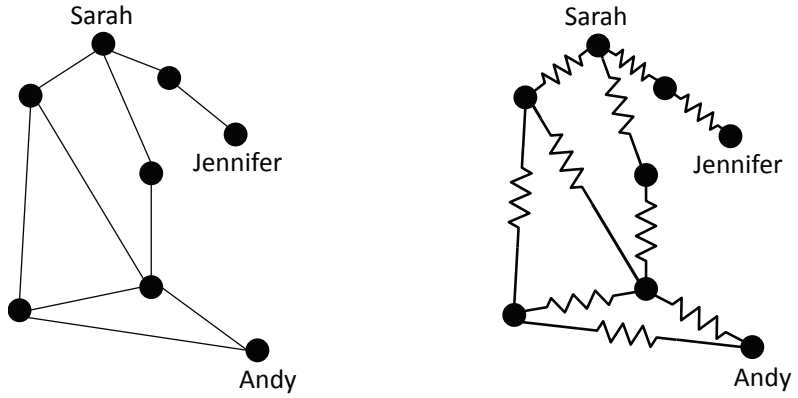


Figure 10: Left: a simple social network; right: resistance distance model.

5.2 Stochastic Petri nets

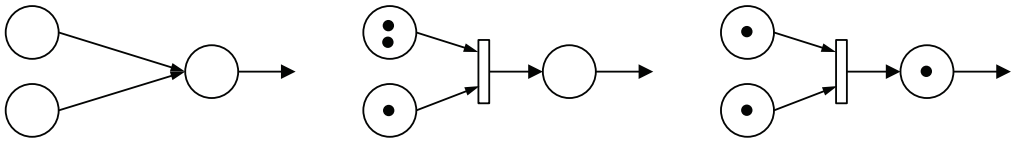


Figure 11: Left: state graph; center: Petri net, before firing; right: Petri net, after firing.

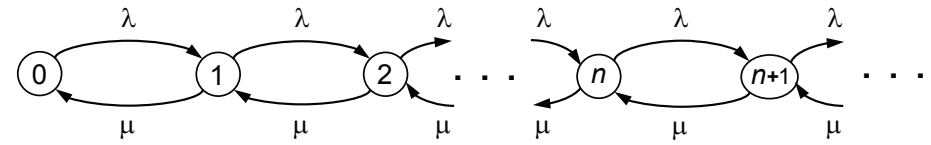
There are significant features of many complex systems that cannot be modeled easily by any of the graph formalisms discussed so far. For example, every sample path of a Markov or semi-Markov process is a linear sequence of states, whereas in many systems concurrent activities take place, which would require multiple simultaneous, perhaps interacting, linear paths. Petri nets consists of places, explicit transitions, arcs and a marking (a distribution of "tokens" over places). Places and transitions are vertices in a bipartite graph in which arcs are the edges, and tokens represent entities whose presence in places enable transitions. Conceptually, Petri nets can model multiple sample paths in a multistate process simultaneously—for example, they can model synchronization, deadlock, etc. in a parallel computer system, or the concurrent actions of attackers and defenders in a military engagement. Reisig (1992) gives a detailed but non-technical introduction to modeling systems and processes with Petri nets; Reisig (1985) and Murata (1989) provide details on the mathematical properties of Petri nets.

Both temporal duration and probability are absent from the basic Petri net formalism. Many extensions are possible: where multiple arcs leaving a place or entering a transition are enabled (as in the middle of Figure 11), they can be assigned transition probabilities;

waiting time distributions can be assigned to transitions or arcs, most often via a Markov transition rate matrix $\mathbf{\Lambda}$ (Ajmone Marsan (1989)); semi-Markov transitions are also used (e.g., see Ciardo and Lindemann (1994)). Balbo (2001) provides a detailed introduction to all the "stochasticizing" possibilities in Petri nets.

Figure 12 shows three representations of a Markov process model for an M/M/1 queue (Poisson arrivals with mean λ, exponentially distributed service times with mean μ). The top shows a state transition diagram, the bottom left the Markov transition rate matrix, and the bottom left a stochastic Petri net (from Ajmone Marsan (1989)). Note the simplicity of the Petri net representation, though the underlying computations are the same in all cases.

Figure 12 illustrates a common tradeoff between representational power and mathematical tractability. The full machinery of stochastic Petri nets takes models well beyond what can be analytically solved, so properties such as first passage times must be approximated using simulation. In many cases, the main interest lies in determining qualitative characteristics of systems, so the greater representational power of Petri nets is a key advantage; see Reisig (1992) .



$$\begin{pmatrix} -\lambda & \lambda & 0 & 0 & \\ \mu & -(\lambda+\mu) & \lambda & 0 & \\ 0 & \mu & -(\lambda+\mu) & \lambda & \dots \\ 0 & 0 & \mu & -(\lambda+\mu) & \\ & & \vdots & & \ddots \end{pmatrix}$$

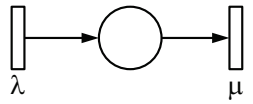


Figure 12: Top: state graph for M/M/1 queue; bottom left: Markov transition matrix $\mathbf{\Lambda}$; bottom right: stochastic Petri net representation.

5.3 Hierarchical composition/decomposition

Complex systems are often decomposable or "nearly decomposable" (Simon (1996)). In terms of a graph model of system components or states, suppose the vertices of a directed graph $\mathcal{G}$ can be decomposed into sets $\mathcal{S}$ and $\mathcal{S}' = \mathcal{G} \backslash \mathcal{S}$ such that there is a small number of directed edges $\mathcal{S} \to \mathcal{S}'$ and a small number of directed edges $\mathcal{S}' \to \mathcal{S}$. Then the subgraph consisting of $\mathcal{S}$ and all edges that do not leave $\mathcal{S}$ can be contracted to a single vertex representing, for example, a subsystem. In the opposite direction, an initial model based on subsystems and their interactions can be expanded by filling in components and their interactions. This hierarchical decomposition of state or entity graphs may simplify analysis by hiding unimportant details, and facilitate a "divide and conquer" approach to system analysis.

Examples may be found in reliability analysis and queueing network analysis, where decomposition of the state space is a well-known technique (Lazowska et al.; 1984, Chapter 8). Decomposition of entity graphs has been widely applied in computer software design; e.g., see Parnas et al. (1985). Figure 13 illustrates decomposition in a reliability example: state 2 is decomposed into a set of states and transitions wherein subcomponents with

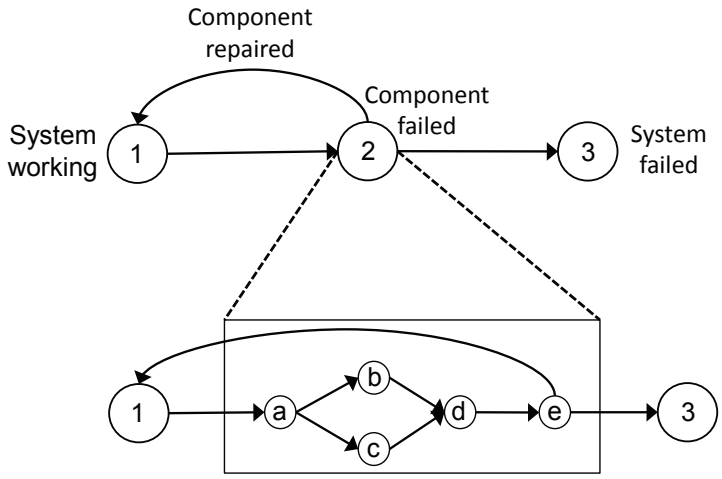


Figure 13: Decomposition of a graph vertex in a system reliability application.

redundancy, and subject to repair, are composed into the higher-level state "component failed" (Collins and Huzurbazar (2012)). For examples of hierarchical composition and decomposition of Petri nets, see Chapter 6 in Reisig (1992).

6. Summary

We have reviewed various applications of graph theory to stochastic problems, emphasizing the power of the interplay of topological, algebraic, and visual properties of graphs, which provides a powerful set of tools for solving problems and guiding intuition. Tools include matrix analysis, graph reduction algorithms, and hierarchical composition and decomposition of graph models. The power of these tools is illustrated by many applications, a few of which were surveyed here: Markov and semi-Markov models in finance, queueing, and reliability analysis; Petri nets for queueing analysis; and resistance distance in social networks. As the developers of the resistance distance concept state, " . . . possibilities for the development of the field of graph theory are ever open for innovative ideas" (Klein and Randič (1993)); many other applications await.

7. Acknowledgments

The authors thank presentation audiences at JSM and at a meeting of the ASA Albuquerque Chapter for helpful comments. This work was funded by the Enhanced Surveillance Campaign (C8) at Los Alamos National Laboratory.

References

Ajmone Marsan, M. (1989). Stochastic Petri nets: An elementary introduction, *in* G. Rozenberg (ed.), *Advances in Petri Nets*, Springer-Verlag, Berlin. (download from `http://www.fd.cvut.cz/department/k611/pedagog/tho_a/a_soubory/spn_introduction.pdf`).

Balbo, G. (2001). Introduction to stochastic Petri nets:, *in* E. Brinksma, H. Hermanns and J.-P. Katoen (eds), *Lectures on Formal Methods and Performance Analysis*, Springer-

Verlag, Berlin. (Download from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.130.1476&rep=rep1&type=pdf).

Busacker, R. G. and Saaty, T. L. (1965). *Finite Graphs and Networks: An Introduction with Applications*, McGraw-Hill, New York.

Ciardo, G. and Lindemann, C. (1994). A characterization of the stochastic process underlying a stochastic Petri net, *IEEE Transactions on Software Engineering* **20**(7): 506–515.

Collins, D. H. (2011). The relationship between Fourier and Mellin transforms, with applications to probability and stochastic processes, *Technical Report LA-UR-11-03715*, Los Alamos National Laboratory, Los Alamos, NM.

Collins, D. H. and Huzurbazar, A. V. (2012). Prognostic models based on statistical flowgraphs, *Applied Stochastic Models in Business and Industry* **28**: 141–151.

Collins, D. H. and Huzurbazar, A. V. (2013). Multi-state stochastic processes: A statistical flowgraph perspective, *International Statistical Revirew* **81**: 78–106.

Csenki, A. (2008). Flowgraph models in reliability and finite automata, *IEEE Transactions on Reliability* **57**(2): 643–657.

Csenki, A. (2009). Towards a unified view of finite automata and semi-Markov flowgraph models, *International Journal of General Systems* **38**(6): 643–657.

Doyle, P. G. and Snell, L. (1984). *Random Walks and Electric Networks*, Mathematical Association of America, Washington, DC.

Ellerman, D. P. (1984). Arbitrage theory: A mathematical introduction, *SIAM Review* **26**(2): 241–261.

Ghez, R. (2010). *Diffusion Phenomena: Cases and Studies*, second edn, Kluwer Academic/Plenum Publishers, New York.

Grimmet, G. (2010). *Probability on Graphs*, Cambridge University Press, Cambridge, UK.

Hartsfield, N. and Ringel, G. (2003). *Pearls in Graph Theory: A Comprehensive Introduction*, Dover, Mineola, NY.

Huzurbazar, A. V. (2005). *Flowgraph Models for Multistate Time-to-Event Data*, Wiley-Interscience, Hoboken, NJ.

Jarrow, R. A., Lando, D. and Turnbull, S. M. (1997). A Markov model for the term structure of credit risk spreads, *The Review of Financial Studies* **10**(2): 481–523.

Jiang, X., Lim, L.-H., Yao, Y. and Ye, Y. (2011). Statistical ranking and combinatorial Hodge theory, *Mathematical Programming* **127**(1): 203–244.

Kelly, F. P. (1979). *Reversibility and Stochastic Networks*, John Wiley & Sons, Chichester, UK.

Kepner, J. and Gilbert, J. (eds) (2011). *Graph Algorithms in the Language of Linear Algebra*, SIAM, Philadelphia, PA.

Klein, D. and Randič, M. (1993). Resistance distance, *Journal of Mathematical Chemistry* **12**: 81–95.

Koller, D. and Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*, MIT Press, Cambridge, MA.

Kozen, D. C. (1997). *Automata and Computability*, Springer, New York.

Laplace, P. S. (1951). *A Philosophical Essay on Probabilities*, English translation edn, Dover Publications, New York.

Lazowska, E. D., Zahorjan, J., Graham, G. S. and Sevcik, K. C. (1984). *Quantitative System Performance: Computer System Analysis Using Queueing Network Models*, Prentice-Hall, Englewood Cliffs, NJ.

Marathe, M. and Vullikanti, A. K. S. (2013). Computational epidemiology, *Communications of the ACM* **56**(7): 88–96.

Mason, S. J. (1953). Feedback theory—some properties of signal flow graphs, *Proc. I.R.E.* **41**: 1144–1156.

Mason, S. J. (1956). Feedback theory—further properties of signal flow graphs, *Proc. I.R.E.* **44**: 920–926.

McCabe, T. J. and Butler, C. W. (1989). Design complexity measurement and testing, *Communications of the ACM* **32**(12): 1415–1425.

McRae, B. H., Dickson, B. G., Keit, T. H. and Shah, V. B. (2008). Using circuit theory to model connectivity in ecology, evolution, and conservation, *Ecology* **89**(10): 27122724.

Murata, T. (1989). Petri nets: Properties, analysis, and applications, *Proceedings of the IEEE* **77**(4): 541–580.

Parnas, D. L., Clements, P. C. and Weiss, D. M. (1985). The modular structure of complex systems, *IEEE Transactions on Software Engineering* **11**(3): 259–266.

Pearl, J. (2009). *Causality: Models, Inference, and Reasonig*, second edn, Cambridge University Press, Cambridge, UK.

Pritsker, A. B. (1977). *Modelling and Analysis Using Q-GERT Networks*, John Wiley & Sons, New York.

Rausand, M. and Høyland, A. (2004). *System Reliability Theory: Models, Statistical Methods, and Applications*, second edn, Wiley-Interscience, Hoboken, NJ.

Reisig, W. (1985). *Petri Nets: An Introduction*, Springer-Verlag, Berlin.

Reisig, W. (1992). *A Primer in Petri Net Design*, Springer-Verlag, Berlin.

Ross, S. M. (1992). *Applied Probability Models with Optimization Applications*, Dover Publications, New York.

Ross, S. M. (1996). *Stochastic Processes*, 2nd edn, John Wiley & Sons, New York.

Satyanarayana, A. and Hagstrom, J. N. (1981). Combinatorial properties of directed graphs useful in computing network reliability, *Networks* **11**: 357–366.

Simon, H. A. (1996). *The Sciences of the Artificial*, 3rd edn, MIT Press, Cambridge MA.

Springer, M. D. (1979). *The Algebra of Random Variables*, John Wiley & Sons.

Theocaris, P. S. and Chrysakis, A. C. (1977). Numerical inversion of the Mellin transform, *Journal of the The Institute of Mathematics and its Applications* **20**: 73–83.

Tizghadam, A. and Leon-Garcia, A. (2010). Betweenness centrality and resistance distance in communication networks, *IEEE Network* **24**(6): 10–16.

Wang, S. and Hauskrecht, M. (2010). Effective query expansion with the resistance distance based term similarity metric, *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval (SIGIR'10)*, pp. 715–716.

Warr, R. L. and Collins, D. H. (2012). An introduction to solving for quantities of interest in finite-state semi-Markov processes. *arXiv* preprint arXiv:1212.1440.

Wasserman, S. and Faust, K. (1994). *Social Network Analysis: Methods and Applications*, Cambridge University Press, Cambridge, UK.