THE HEAT EQUATION ON DISCRETE DOMAINS

JUSTIN TITTELFITZ

ABSTRACT. In this paper, we will explore the properties of the Heat Equation on Discrete Networks, laying out groundwork and giving general forms for solutions, and then exploring the inverse problem. We will also focus on comparisons of Heat Networks with Electrical and Random-Walk Networks.

CONTENTS

1. INTRODUCTION

Author's note: throughout this paper, we will use u_t as well as $\partial_t u$ to indicate the derivative of u with respect to t. The $\partial_t u$ notation is mainly used when additional subscripts become cumbersome.

In this paper, we will explore the Heat Equation on graphs. First, however, we must give a definition of a Heat Network.

Definition 1.1. Let $H = (V, \sigma, \mu)$ be a set of vertices $V = \{v_i\}$, connected by a set of directed edges σ, with weight μ. We will designate a set of boundary vertices $\partial V \subset V$ as well as a set of interior vertices $int(V) = V - \partial V$. An edge from v_i to v_j will be indicated by σ_{ij}; we will also allow edges to begin and end at the same node, which we will designate σ_{ii}, and refer to as loops. Further, we will require $\mu_{ij} = \mu_{ji}$ if $v_i, v_j \in int(V)$ (this requirement will not be enforced if one of the vertices is in the boundary). By convention, we will say $\mu_{ij} = 0$ if there is no edge connecting v_i and v_j. If there is an edge from v_i to v_j, we will say $v_i \sim v_j$, or 'v_j is a neighbor of v_i'. Note that under this definition, if v_i has a loop, then $v_i \sim v_i$. The boundary vertices will then be further divided into two types

(1) Absorbing vertices: The only edges leaving these vertices are loops; ie. for $v_i \in \partial V_A$ $i \neq j \Rightarrow \mu_{ij} = 0$

Date: July 11, 2006.

(2) Non-Absorbing vertices: These vertices can have edges to other vertices, as well as loops.

These two types of nodes correspond to Dirichlet and Neumann conditions, respectively (more discussion will come later). Additionally, boundary nodes of the non-absorbing type behave more or less identically to interior nodes.

At this point, it will also be convenient to define the quantities $\sigma_i = \sum_{v_i \sim v_j} \mu_{ij}$ and $\gamma_{ij} = \mu_{ij} / \sigma_i$. In this sense, γ is just a normalization of μ.

In the continuous case, the Heat Equation is given as

$$u_t = div(\gamma \nabla u) = \Delta_\gamma u$$

In the discrete case, we will interpret Δ_γ in the same manner as we would for the Electrical Conductivity Equation on a graph; that is, as a matrix which acts on a vector $u(t)$ such that:

$$\Delta_\gamma u_i = \sum_{v_i \sim v_j} \gamma_{ij}(u_i - u_j)$$

where γ_{ij} can be thought of as a weight (or in the case of a random walk, the transition probability) from v_i to v_j. This is similar to conductances of edges in the Electrical Conductivity Equation, with two exceptions

(1) These weights are normalized; in other words, $\sum_{v_i \sim v_j} \gamma_{ij} = 1$, and

(2) Weights are not necessarily symmetric; in other words, $\gamma_{ij} \neq \gamma_{ji}$.

From this point on, we will let the matrix K encode the behavior of Δ_γ, so that our equation becomes

$$u_t = -Ku$$

At this point, it should be noted that we are choosing to use the symbol K, however this is not a Kirchoff matrix (though it bears some similarities).

Definition 1.2. Construct K as follows: Given a Heat Network H, with n vertices, number the vertices, starting with boundary nodes, from $1 \leq i \leq n$.
Form the 'Weight Matrix' A of H by setting

$$a_{ij} = \begin{cases} -\mu_{ij}, & \text{if } i \neq j \text{ and there is an edge connecting } v_i \text{ and } v_j \\ \sum_{i \neq j} \mu_{ij}, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases}$$

Form the 'Normalization Matrix' T of H by setting

$$t_{ij} = \begin{cases} 1/\sigma_i, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases}$$

Then $K = TA$. Under this construction, the entries k_{ij} are equal to $-\gamma_{ij}$ if $i \neq j$ and $k_{ii} = -\sum_{i \neq j} k_{ij}$. It is relatively straightforward to show that the rows of K will each sum to zero under this construction.
Another important property of this construction is revealed by considering the following partitioning of K

$$K = \begin{bmatrix} M & P \\ Q & R \end{bmatrix} = \begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix} \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix} = TA$$

Then the submatrix R is the product of a diagonal matrix T_2 and a symmetric matrix A_4.
Additionally, if $\sigma_1 = \ldots = \sigma_i = \ldots = \sigma_n = \sigma$ then $R = (1/\sigma)\ I\ A_4$, and is itself symmetric. It is always possible, given a set of weights on 'non-loop' edges, to choose weights of loops so that the above condition holds, though this is a more restrictive construction. We will refer to this special case as the 'Symmetric Construction'.

Theorem 1.3. *If there exists a solution to the Boundary Value - Initial Value Problem (BV-IVP)*

$$u_t = -Ku;\ u(0) = f;\ u|_{\partial V} = g$$

then it is unique.

Proof. Assume another solution, w exists. Let $z = u - w$; we want to show z is identically zero. Note that z satisfies our PDE by linearity, and $z(0) = 0$; $z|_{\partial V} = 0$. Then $z_t(0) = -Kz(0) = 0$ implies z is identically zero. □

Theorem 1.4. *If a steady-state solution to the Heat Equation exists, it is also a solution to the Dirichlet problem for the Electrical Conductivity Equation, with temperature interpreted as potential.*

Proof. This fact follows almost trivially. If we let v be our steady-state solution; that is $u \to v$ as $t \to \infty$, then clearly $v_t = 0$. Then

$$-\Delta_\gamma v = v_t \ \Rightarrow\ \Delta_\gamma v = 0$$

And v satisfies the Electrical Conductivity Equation. □

2. Time as a Discrete Variable

First, we will consider time as a discrete variable. We will always take $\Delta t = t_{i+1} - t_i = 1$. In this case, our equation becomes

$$u_t = u(t_{i+1}) - u(t_i) = -K\ u(t_i)$$

A simple rearrangement yields

$$u(t_{i+1}) = (I - K)\ u(t_i)$$

As long as the values of K are independent of time, we can attain the closed form

$$u(t_n) = (I - K)^n\ u(t_0)$$

Lemma 2.1. *If we write*

$$(I - K) = \begin{bmatrix} I & 0 \\ X & Y \end{bmatrix}$$

Then $(I - K)^n$ converges as $n \to \infty$, and has a closed form

$$\lim_{n\to\infty} (I - K)^n = \begin{bmatrix} I & 0 \\ B & 0 \end{bmatrix}$$

Where $B = (I - Y)^{-1}X$

Proof. By construction, $(I - K)$ has all of the properties of the transition matrix for an absorbing Markov chain. A proof can be found in most elementary texts on Markov chains, as well as in Timothy DeVries paper [3]. □

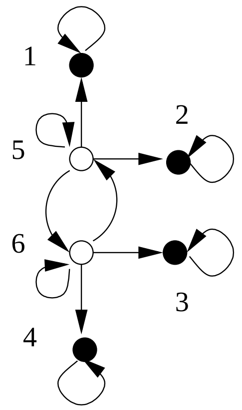

FIGURE 1. A heat network on the "Top-Hat" graph

If we then allow $n \to \infty$, the result will describe the steady-state solution of the Heat Equation, which is a direct analog of the solution to the Dirichlet problem for the Electrical Conductivity Equation, as well as the absorbtion probability for a Random Walk.

2.1. **Time-Independent Boundary Conditions.** At this point, it may be productive to consider a pair of examples. Consider figure 1 on page 4. We will, for our first example, set the weight of interior loops to be zero, and also set all paths leading out of a vertex to be equally weighted. In this case, the relevant matrices are

$$K = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1/3 & -1/3 & 0 & 0 & 1 & -1/3 \\ 0 & 0 & -1/3 & -1/3 & -1/3 & 1 \end{bmatrix}$$

$$(I - K) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 0 & 1/3 \\ 0 & 0 & 1/3 & 1/3 & 1/3 & 0 \end{bmatrix}$$

$$\lim_{n \to \infty} (I - K)^n = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 3/8 & 3/8 & 1/8 & 1/8 & 0 & 0 \\ 1/8 & 1/8 & 3/8 & 3/8 & 0 & 0 \end{bmatrix}$$

The lower left submatrix of the final result can be used to find equilibrium temperature distributions given a fixed boundary temperature. In other words, if u_B is

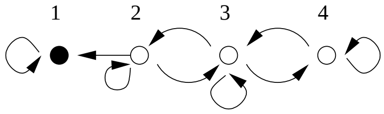

FIGURE 2. A heat network on the simple line graph

given as $(1,0,0,0)^T$, then the equilibrium temperature will be 3/8 at v_5, and 1/8 at v_6. At this point, we encourage the reader to verify that, for an Electrical Conductivity Network, this is the same result one would obtain by setting the potential to 1 at v_1, 0 at all other boundary vertices, and then finding the resulting potential at the interior vertices.

It can be shown that the weight of interior loops will not change the values in our limiting matrix (see [3] or a text discussing Markov chains for details). As a demonstration, let us consider instead the weights of our interior loops being 7 times more than the other edges leaving the vertex. The resulting matrix is

$$(I-K) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1/10 & 1/10 & 0 & 0 & 7/10 & 1/10 \\ 0 & 0 & 1/10 & 1/10 & 1/10 & 7/10 \end{bmatrix}$$

While this present example will converge to its limit slower than the previous example, calculating the limit produces the same result. This certainly introduces some concerns if we ultimately become interested in an inverse problem; the steady-state solution seems to "hide" information about interior loops.

2.2. **Time-Dependent Boundary Conditions.** Another interesting situation to look at is one where boundary conditions change as a function of time. Consider the simple-line graph in figure 2 on page 5. In this case, we will use 1/20 as the value of edges leading from an interior vertex to any other vertex. We will also require that the temperature at v_1 be a function of time, in this case $f(t) = \cos(t/10) + 1$. The relevant matrix will be a function of time, and will be indicated by $A(t)$ rather than the usual $(I-K)$.

$$A(t_i) = \begin{bmatrix} f(t_{i+1})/f(t_i) & 0 & 0 & 0 \\ 1/20 & 9/10 & 1/20 & 0 \\ 0 & 1/20 & 9/10 & 1/20 \\ 0 & 0 & 1/20 & 19/20 \end{bmatrix}$$

The construction of this matrix, in particular the top-left entry, may benefit from some explanation. At each time step, we will obtain our updated temperatures by multiplying this matrix by a vector containing our present temperatures. The entry in the first element of u will contain $f(t_i)$, and thus, multiplication under this construction will properly update the temperature at that vertex. In this scenario,

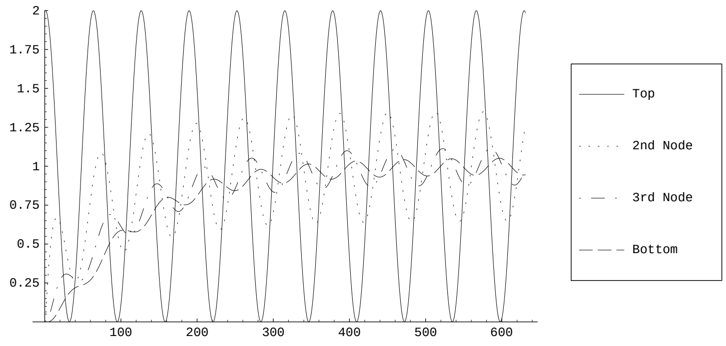

FIGURE 3. Fourier's Wine Cellar

we must express $u(t)$ recursively

$$u(0) = \begin{bmatrix} f(0) \\ 0 \\ 0 \\ 0 \end{bmatrix};\ u(t_i) = A(t_{i-1})\ u(t_{i-1})$$

A plot of the results of this process, can be found on page 6. The most interesting aspect is that each vertex in the chain is successively more damped, as well as out of phase with the source vertex. In fact, the value of $1/20$ was chosen because it results in the 4th vertex being more or less π radians out of phase with the source. One could certainly appreciate the usefulness of this physical property if one were, for instance, trying to keep wine warm in the winter and cool in the summer. Finally, it should be noted that, while this technique produces some interesting results, it also has some flaws. For instance, if the value of f is ever zero, then A is poorly defined.

3. TIME AS A CONTINUOUS VARIABLE

While valuable insights can be gained by thinking of time as a discrete variable, it is certainly more natural to consider it as a continuous one. Returning to our original equation, $u_t = -Ku$, we will partition our vector u and our matrix K as follows

$$\partial_t \begin{bmatrix} u_B \\ u_I \end{bmatrix} = - \begin{bmatrix} M & P \\ Q & R \end{bmatrix} \begin{bmatrix} u_B \\ u_I \end{bmatrix}$$

If we have n boundary nodes, and m interior nodes, then u_B and u_I are vectors of length n and m, respectively, M is $n \times n$, R is $m \times m$, P is $n \times m$, and Q is $m \times n$. Additionally, if we have time-independent Dirichlet boundary conditions, then M and P are zero matrices.

Theorem 3.1. *Using the above partitioning, the solution to the BV-IVP*

$$u_t = -Ku;\ u(0) = f;\ u|_{\partial V} = g$$

is given by

$$u_I(t) = \exp(-Rt)\ f - \int_0^t Exp[-R(t-s)]\ Q\ g\ ds$$

Proof. Carrying out the matrix multiplication defined above yields the following differential equations

$$\partial_t u_B = -Mu_B - Pu_I$$
$$\partial_t u_I = -Qu_B - Ru_I$$

Since we assume we already have our boundary information, we will only concern ourselves with the second equation. We can solve this differential equation using the method of an integrating factor. In this case, the factor we need is the exponentiation of the matrix R times the scalar t, $\exp(Rt)$. Then

$$\begin{aligned}
&\partial_t u_I = -Qu_B - Ru_I \\
&\partial_t u_I + Ru_I = -Qu_B \\
&\exp(Rt)\ \partial_t u_I + \exp(Rt)\ R\ u_I = -\exp(Rt)\ Q\ u_B \\
&\partial_t(\exp(Rt)\ u_I) = -\exp(Rt)\ Q\ u_B \\
&\exp(Rt)\ u_I(t) = u_I(0) - \int_0^t \exp(Rs)\ Q\ u_B\ ds \\
&u_I(t) = \exp(-Rt)\ u_I(0) - \int_0^t \exp(-R(t-s))\ Q\ u_B\ ds \\
&u_I(t) = \exp(-Rt)\ f - \int_0^t \exp(-R(t-s))\ Q\ g\ ds
\end{aligned}$$

□

Claim 3.2. The fundamental solution for our BV-IVP is

$$G = \exp(-Rt)$$

We will not seek to extensively defend this claim. We will, however, motivate it by comparing the form of our solution to the integral from of the solution to the heat equation in the continuous case

$$u(x,t) = \int_\Omega G\ f\ dx + \int_0^t \int_{\partial\Omega} \nabla_{e\perp} G\ g\ ds\ dx$$

The matrix Q encodes the relationship between interior and boundary vertices, thus multiplying by it is the discrete analog of the normal derivative. The integration with respect to time remains, and spatial integration is effectively replaced by matrix multiplication. This argument is similar to that presented by Karen Perry for the Green's Function for the Electrical Conductivity Equation [4].

We can also express the solution in the form of an eigenvector expansion.

Theorem 3.3. *Consider the submatrix R of K. In the introduction, we discussed the fact that R is the product of a diagonal matrix T_2 and a symmetric matrix A_4. Now let*

$$\lambda_1 \leq ... \leq \lambda_i \leq ... \leq \lambda_n$$

be the eigenvalues of A_4, and $\{\phi_i\}$ be the corresponding eigenvectors, chosen orthonormally; ie

$$i \neq j \Rightarrow \phi_i \cdot \phi_j = 0 \text{ and } |\phi_i| = 1$$

Then, if there exists an equilibrium temperature distribution (steady-state solution) u_E, the solution to the BV-IVP defined by

$$u_t = -Ku;\ u(0) = f;\ u|_{\partial V} = g$$

can also be stated as

$$u_I(t) = u_E - \sum_{i=1}^{n} (u_E - g) \cdot \phi_i \ \exp(-T_2 \lambda_i t)\ \phi_i$$

Proof. We expect u to eventually reach a steady state; we will then assume we can solve for u as the difference between this equilibrium state and another function with homogeneous boundary conditions; that is

$$u(t) = u_E - w(t)$$

Thus, we must first solve the BV-IVP defined by

$$w(t) = u_E - u(t);\ w_t = -Kw;\ w(0) = u_E - f;\ w|_{\partial V} = 0$$

Now we assume that $w_I(t)$ can be written in the form

$$w_I(t) = \sum_{i=1}^{n} a_i(t)\phi_i$$

Then

$$\begin{aligned} \partial_t w_I + Rw_I &= \sum_{i=1}^{n} \partial_t a_i(t)\phi_i + a_i(t)T_2 A_4 \phi_i \\ -Qw_B &= \sum_{i=1}^{n} \partial_t a_i(t)\phi_i + a_i(t)T_2 A_4 \phi_i \\ 0 &= \sum_{i=1}^{n} \partial_t a_i(t)\phi_i + a_i(t)T_2 \lambda_i \phi_i \end{aligned}$$

We then take an inner product of this sum with ϕ_j. Since we have chosen our eigenvectors to be orthonormal, only one term of this sum survives.

$$\begin{aligned} 0 &= \partial_t a_j(t) + a_i(t)T_2\lambda_j \\ \partial_t a_j(t) &= -T_2\lambda_j a_i(t) \end{aligned}$$

The solution of this differential equation is $a_j(t) = a_j(0)\exp(-T_2\lambda_j t)$. To solve for $a_j(0)$, we apply our initial condition.

$$w_I(0) = \sum_{i=1}^{n} a_i(0)\phi_i$$
$$(u_E - f) = \sum_{i=1}^{n} a_i(0)\phi_i$$

Again, we use an inner product

$$(u_E - f)\cdot\phi_j = (\sum_{i=1}^{n} a_i(0)\phi_i)\cdot\phi_j$$
$$(u_E - f)\cdot\phi_j = a_j(0)$$

And thus

$$w_I(t) = \sum_{i=1}^{n}(u_E - g)\cdot\phi_i \ \exp(-T_2\lambda_i t)\ \phi_i$$

finally yielding

$$u_I(t) = u_E - \sum_{i=1}^{n}(u_E - g)\cdot\phi_i \ \exp(-T_2\lambda_i t)\ \phi_i$$

□

Corollary 3.4. *If we are using a symmetric construction, then $T_2 = (1/\sigma)I$ and*

$$u_I(t) = u_E - \sum_{i=1}^{n}(u_E - g)\cdot\phi_i \ \exp(-\lambda_i t/\sigma)\ \phi_i$$

4. Inverse Problems for the Heat Equation

Previously, papers (see [2] and [3]) have been written on recoverability of random-walk networks, which are extremely similar to heat networks. Through this work, various sufficient and/or necessary conditions for recoverability of the probability matrix have been stated

(1) If loops are allowed on interior nodes, the probabilities are unrecoverable.
(2) If a random walk network is recoverable, then all of the edges leaving any interior vertex can be simultaneously extended to vertex- disjoint paths to the boundary.

Our analysis will focus mostly on ways around the first condition, and, as such, our statement of the second condition is brief, and the discussion is non-existant. We highly encourage the reader to look at [2] for an in-depth analysis of the second condition, as well as a sufficiency condition. In both the case of a random-walk or heat network, the absorption matrix effectively hides information about interior loops, and only produces the ratio between edges leading out of a vertex. It seems, however, to be a reasonable intuition that the eigenvalues of the matrix are in some way linked to the weights of these loops, because "heavier" loops will lead to smaller eigenvalues. As seen in the previous section, the time-dependent part

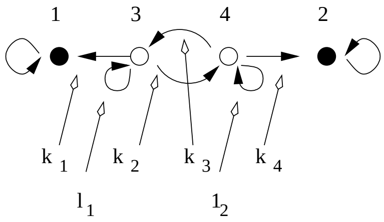

FIGURE 4. Recoverability Example

of our solutions are directly tied to the eigenvalues, or spectrum, of K. Originally, we had conjectured that, given a heat network H, possibly with interior loops, and which satisfies the revised Card Conjecture outlined in [2], then knowledge of the steady-state solution, in addition to the spectrum of K will lead to recoverability. Through exploration, however, this was found to be false. Consider the line graph in figure 4, we can write the Kirchoff matrix as

$$K = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -k_1 & 0 & s_1 & -k_2 \\ 0 & -k_3 & -k_4 & s_2 \end{bmatrix}$$

where $s_1 = k_1 + k_2$ and $s_2 = k_3 + k_4$. It is clear that two of the eigenvalues are zero. We can find the other two by calculating the characteristic polynomial. This yields

$$\lambda = \frac{1}{2}(s_1 + s_2 \pm \sqrt{(s_1 + s_2)^2 - 4(s_1 s_2 - k_2 k_4)})$$

Then simple calculation shows

$$\begin{aligned} \lambda_1 + \lambda_2 &= k_1 + k_2 + k_3 + k_4 \\ \lambda_1 \times \lambda_2 &= (k_1 + k_2)(k_3 + k_4) - k_2 k_4 \end{aligned}$$

From the absorbtion matrix, we know how to express the ratio between k_1 and k_2, and between k_3 and k_4. Combining these relationships will yield a quadratic equation. In many cases, we can find two solutions which satisfy this equation, meaning our recovery process is non-unique (though finitely so). In fact, the situations where the roots of this polynomial coincide (and thus give a unique solution) happen when the weights of the loops being equal to each other. This assertion can be proved by examining the discriminant. Furthermore, the analysis only gets worse for larger graphs; if we assume that symmetric polynomials will give the needed relationships, a graph with n interior vertices will eventually produce n polynomials, from a linear function to an n-th degree polynomial, and yield $n!$ solutions, at worst (see Bezout's theorem). In the case with three interior vertices, the equations would be

$$\begin{aligned} P_1 &= \lambda_1 + \lambda_2 + \lambda_3 \\ P_2 &= \lambda_1\lambda_2 + \lambda_2\lambda_3 + \lambda_1\lambda_3 \\ P_3 &= \lambda_1\lambda_2\lambda_3 \end{aligned}$$

These equations, along with the ratios, should provide enough relationships to solve for all the interesting quantities. However, the expansions of these polynomials, even in this case, became somewhat daunting. Even with the use of Mathematica, the calculation became nearly impossible in some cases. We can see it only gets worse with more and more interior vertices. There may well be another approach that avoids these gruesome polynomials, but we have been unable to unearth it. We have included a copy of our Mathematica code in the Appendix.

5. Future Work

Results for the inverse problem were less than thrilling; however someone with an affinity for polynomials may find a way to plow through the recovery process. The interpretation and meaning of the boundary of these graphs has been dealt with in a manner which leads to some problems with symmetry. The normalizing of the edge weights also leads to some symmetry issues. While it may be unavoidable, this is a bit troublesome, because the Laplacian used in the continuous version of this problem is a symmetric operator.

References

[1] Curtis, B., and James A. Morrow. "Inverse Problems for Electrical Networks." Series on applied mathematics – Vol. 13. World Scientific, ©2000.
[2] Diamondstone, D. "Walk Network Recoverability and The Card Conjecture", 2004
[3] DeVries, T. "Recoverability of Random Walk Networks", 2003
[4] Perry, K. "Discrete Complex Analysis", 2003

6. Appendix

This is some of the Mathematica code used to explore recoverability for a network with three interior vertices.

Off[General::spell1]

$$\mathbf{k1 = 1/2;}$$
$$\mathbf{k2 = 1/4;}$$
$$\mathbf{k3 = 1/8;}$$
$$\mathbf{k4 = 1/4;}$$
$$\mathbf{k5 = 1/4;}$$
$$\mathbf{k6 = 3/8;}$$
$$\mathbf{k7 = 3/8;}$$
$$\mathbf{k8 = 3/8;}$$
$$\mathbf{k9 = 1/8;}$$
$$\boldsymbol{\sigma}\mathbf{1 = k1 + k2 + k3;}$$
$$\boldsymbol{\sigma}\mathbf{2 = k4 + k5 + k6;}$$
$$\boldsymbol{\sigma}\mathbf{3 = k7 + k8 + k9;}$$
$$\boldsymbol{\tau}\mathbf{1 = x1 + x2 + x3;}$$
$$\boldsymbol{\tau}\mathbf{2 = x4 + x5 + x6;}$$
$$\boldsymbol{\tau}\mathbf{3 = x7 + x8 + x9;}$$

$$K = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -\mathrm{k1} & 0 & 0 & \sigma 1 & -\mathrm{k2} & -\mathrm{k3} \\ 0 & -\mathrm{k4} & 0 & -\mathrm{k5} & \sigma 2 & -\mathrm{k6} \\ 0 & 0 & -\mathrm{k7} & -\mathrm{k8} & -\mathrm{k9} & \sigma 3 \end{pmatrix};$$

$$H = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -\mathrm{x1} & 0 & 0 & \tau 1 & -\mathrm{x2} & -\mathrm{x3} \\ 0 & -\mathrm{x4} & 0 & -\mathrm{x5} & \tau 2 & -\mathrm{x6} \\ 0 & 0 & -\mathrm{x7} & -\mathrm{x8} & -\mathrm{x9} & \tau 3 \end{pmatrix};$$

$Y = \text{Take}[K, \{4,6\}, \{4,6\}];$
$X = \text{Take}[K, \{4,6\}, \{1,3\}];$
$B = \text{Inverse}[-Y].X;$
$R = \text{Solve}[\{B[[1,1]] == \mathrm{r1} + \mathrm{r2} * B[[2,1]] + \mathrm{r3} * B[[3,1]],$
$B[[1,2]] == \mathrm{r2} * B[[2,2]] + \mathrm{r3} * B[[3,2]],$
$B[[1,3]] == \mathrm{r2} * B[[2,3]] + \mathrm{r3} * B[[3,3]],$
$B[[2,1]] == \mathrm{r5} * B[[1,1]] + \mathrm{r6} * B[[3,1]],$
$B[[2,2]] == \mathrm{r4} + \mathrm{r5} * B[[1,2]] + \mathrm{r6} * B[[3,2]],$
$B[[2,3]] == \mathrm{r5} * B[[1,3]] + \mathrm{r6} * B[[3,3]],$
$B[[3,1]] == \mathrm{r8} * B[[1,1]] + \mathrm{r9} * B[[2,1]],$
$B[[3,2]] == \mathrm{r8} * B[[1,2]] + \mathrm{r9} * B[[2,2]],$
$B[[3,3]] == \mathrm{r7} + \mathrm{r8} * B[[1,3]] + \mathrm{r9} * B[[2,3]]\}];$
$\mathrm{c1} = \mathrm{r1}/.R[[1]];$
$\mathrm{c2} = \mathrm{r2}/.R[[1]];$
$\mathrm{c3} = \mathrm{r3}/.R[[1]];$
$\mathrm{c4} = \mathrm{r4}/.R[[1]];$
$\mathrm{c5} = \mathrm{r5}/.R[[1]];$
$\mathrm{c6} = \mathrm{r6}/.R[[1]];$
$\mathrm{c7} = \mathrm{r7}/.R[[1]];$
$\mathrm{c8} = \mathrm{r8}/.R[[1]];$
$\mathrm{c9} = \mathrm{r9}/.R[[1]];$

Null
Off[General::spell1]
$\text{Eigenvalues}[K];$
$\lambda 1 = \text{Eigenvalues}[K][[1]];$
$\lambda 2 = \text{Eigenvalues}[K][[2]];$
$\lambda 3 = \text{Eigenvalues}[K][[3]];$

$\text{Eigenvalues}[H];$
$\lambda\mathrm{A} = \text{Eigenvalues}[H][[4]];$
$\lambda\mathrm{B} = \text{Eigenvalues}[H][[5]];$
$\lambda\mathrm{C} = \text{Eigenvalues}[H][[6]];$

$\Sigma 1 = \lambda 1 + \lambda 2 + \lambda 3;$
$\Sigma 2 = \lambda 1 * \lambda 2 + \lambda 2 * \lambda 3 + \lambda 1 * \lambda 3;$
$\Pi 1 = \lambda 1 * \lambda 2 * \lambda 3;$

$\mathrm{x2} = \mathrm{x1} * \mathrm{c2}/\mathrm{c1}$
$\mathrm{x3} = \mathrm{x1} * \mathrm{c3}/\mathrm{c1}$
$\mathrm{x5} = \mathrm{x4} * \mathrm{c5}/\mathrm{c4}$
$\mathrm{x6} = \mathrm{x4} * \mathrm{c6}/\mathrm{c4}$

```
x8 = x7 * c8/c7
x9 = x7 * c9/c7

ΣA = λA + λB + λC;
ΣB = λA * λB + λB * λC + λA * λC;
ΠA = λA * λB * λC;

Simplify[ΣA]
Simplify[ΣB]
ToRadicals[Simplify[ΠA]]

soln = Solve[{Σ1 == ΣA,
Σ2 == ΣB,
Π1 == ToRadicals[Simplify[ΠA]]}];
Needs["Miscellaneous`RealOnly`"]
ToRadicals[Simplify[soln]]
```

University of Oregon
E-mail address: jtittelfitz@gmail.com