Spectral Graph Theory for Maze Solving

Mathematical Foundations and Applications

Atishaya Maharjan

GADA lab

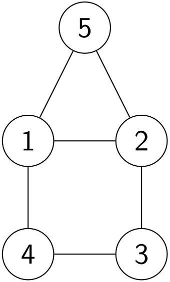
July 23, 2025

Adjacency Matrix

Definition

For a graph G with n vertices, the **adjacency matrix** A is an $n \times n$ matrix where:

$$A_{ij} = \begin{cases} 1 & \text{if edge } (v_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases}$$



$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Adjacency Matrix

Some interesting properties of the adjacency matrix:

- The number of walks of length l from v_i to v_j is encoded at A^l_{ij}.

Adjacency Matrix

Some interesting properties of the adjacency matrix:

- The number of walks of length l from v_i to v_j is encoded at A^l_{ij}.
- The diagonal entries of A^2 gives you the degrees of each vertex.

Adjacency Matrix

Some interesting properties of the adjacency matrix:

- The number of walks of length l from v_i to v_j is encoded at A^l_{ij}.
- The diagonal entries of A^2 gives you the degrees of each vertex.
- $tr(A^2) = \sum_{v \in V} deg(v) = 2|E|$

Adjacency Matrix

Some interesting properties of the adjacency matrix:

- The number of walks of length l from v_i to v_j is encoded at A^l_{ij}.
- The diagonal entries of A^2 gives you the degrees of each vertex.
- $tr(A^2) = \sum_{v \in V} deg(v) = 2|E|$
- $tr(A^3) = 6\times$ (Number of triangles in the graph)

Characteristic Polynomial

Definition (Characteristic Polynomial)

The **characteristic polynomial**, $\chi_A(G)$, of a graph G with n vertices is the determinant $\det(\mathbf{A} - \lambda\mathbf{I})$, for $\lambda \in \mathbb{F}$.

The general form of any characteristic polynomial is:

$$\lambda^n + c_1\lambda^{n-1} + c_2\lambda^{n-2} + \cdots + c_n$$

Where $c_i \in \mathbb{F}$, for $i \in \{1, \ldots, n\}$.

Characteristic Polynomial

Definition (Characteristic Polynomial)

The **characteristic polynomial**, $\chi_A(G)$, of a graph G with n vertices is the determinant $\det(\mathbf{A} - \lambda \mathbf{I})$, for $\lambda \in \mathbb{F}$.

The general form of any characteristic polynomial is:

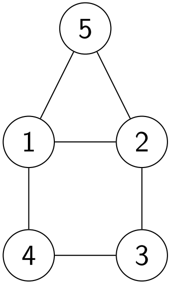
$$\lambda^n + c_1\lambda^{n-1} + c_2\lambda^{n-2} + \cdots + c_n$$

Where $c_i \in \mathbb{F}$, for $i \in \{1, \ldots, n\}$.

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

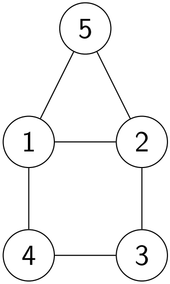
$$\det(\mathbf{A} - \lambda \mathbf{I}) = -\lambda^5 + 6\lambda^3 + 2\lambda^2 - 4\lambda$$

Characteristic Polynomial



$$\det(\mathbf{A} - \lambda\mathbf{I}) = -\lambda^5 + 6\lambda^3 + 2\lambda^2 - 4\lambda$$

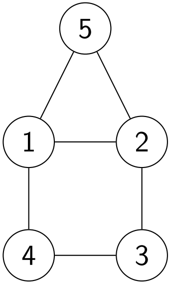
Characteristic Polynomial



$$\det(\mathbf{A} - \lambda \mathbf{I}) = -\lambda^5 + 6\lambda^3 + 2\lambda^2 - 4\lambda$$

- $c_1 = 0$. So $c_1 \lambda^{n-1} = 0$ always.

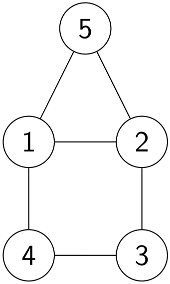
Characteristic Polynomial



$$\det(\mathbf{A} - \lambda \mathbf{I}) = -\lambda^5 + 6\lambda^3 + 2\lambda^2 - 4\lambda$$

- $c_1 = 0$. So $c_1 \lambda^{n-1} = 0$ always.
- $c_2 = |E|$

Characteristic Polynomial



$$\det(\mathbf{A} - \lambda\mathbf{I}) = -\lambda^5 + 6\lambda^3 + 2\lambda^2 - 4\lambda$$

- $c_1 = 0$. So $c_1\lambda^{n-1} = 0$ always.
- $c_2 = |E|$
- $c_3 = 2\times$ (Number of triangles of G).

Eigenvalues

Definition (Eigenvalues)

The roots of the characteristic polynomial are called the **eigenvalues**. That is, the values of λ for which

$$\det(\mathbf{A} - \lambda \mathbf{I}) = 0$$

Eigenvalues

Definition (Eigenvalues)

The roots of the characteristic polynomial are called the **eigenvalues**. That is, the values of λ for which

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0$$

Recall that from our previous adjacency matrix:

$$\det(\mathbf{A} - \lambda\mathbf{I}) = -\lambda^5 + 6\lambda^3 + 2\lambda^2 - 4\lambda$$

The roots of the characteristic equation are:

$$\lambda_1 = -2, \lambda_2 \approx -1.17008, \lambda_3 = 0, \lambda_4 \approx 0.68889, \lambda_5 \approx 2.48119$$

Eigenvalues

Here are some properties related to the eigenvalue(s) of an adjacency matrix of a graph:

- The sum of eigenvalues of a matrix equals its trace.

Eigenvalues

Here are some properties related to the eigenvalue(s) of an adjacency matrix of a graph:

- The sum of eigenvalues of a matrix equals its trace.
- If a matrix is real symmetric, then each eigenvalue of the graph relating to the matrix is real.

Eigenvalues

Here are some properties related to the eigenvalue(s) of an adjacency matrix of a graph:

- The sum of eigenvalues of a matrix equals its trace.
- If a matrix is real symmetric, then each eigenvalue of the graph relating to the matrix is real.

Definition (Multiplicity of an eigenvalue)

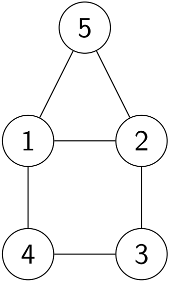
The **(algebraic) multiplicity** of an eigenvalue is the number of times that the value occurs as a root of the characteristic polynomial.

Degree Matrix

Definition

For a graph G with n vertices, the **degree matrix** D is an $n \times n$ matrix where:

$$D_{ij} = \begin{cases} deg(v_i) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$



$$D = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

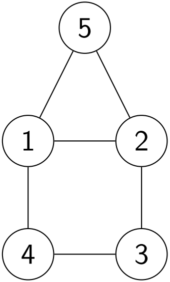
Graph Laplacian

Definition

For a graph G with n vertices, the **Laplacian matrix** is defined as:

$$L = D - A$$

Where D and A are the degree matrix and the adjacency matrix respectively.



$$L = \begin{bmatrix} 3 & -1 & 0 & -1 & -1 \\ -1 & 3 & -1 & 0 & -1 \\ 0 & -1 & 2 & -1 & 0 \\ -1 & 0 & -1 & 2 & 0 \\ -1 & -1 & 0 & 0 & 2 \end{bmatrix}$$

Graph Laplacian

The Laplacian matrix $\mathbf{L}$ of a graph G with n vertices have the following properties:

- $\mathbf{L}$ is a positive semidefinite matrix. That is, it is equal to it's conjugate transpose and it's eigenvalues are all non-negative.

Graph Laplacian

The Laplacian matrix **L** of a graph G with n vertices have the following properties:

- **L** is a positive semidefinite matrix. That is, it is equal to it's conjugate transpose and it's eigenvalues are all non-negative.
- The smallest eigenvalue of **L** is 0.

Graph Laplacian

The Laplacian matrix **L** of a graph G with n vertices have the following properties:

- **L** is a positive semidefinite matrix. That is, it is equal to it's conjugate transpose and it's eigenvalues are all non-negative.
- The smallest eigenvalue of **L** is 0.
- Let $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ be the eigenvalues of **L**. Then,

Graph Laplacian

The Laplacian matrix $\mathbf{L}$ of a graph G with n vertices have the following properties:

- $\mathbf{L}$ is a positive semidefinite matrix. That is, it is equal to it's conjugate transpose and it's eigenvalues are all non-negative.
- The smallest eigenvalue of $\mathbf{L}$ is 0.
- Let $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ be the eigenvalues of $\mathbf{L}$. Then,
  - The multiplicity of the eigenvalue 0 equals the number of connected components in the graph. (Fiedler's Theorem).

Graph Laplacian

The Laplacian matrix $\mathbf{L}$ of a graph G with n vertices have the following properties:

- $\mathbf{L}$ is a positive semidefinite matrix. That is, it is equal to it's conjugate transpose and it's eigenvalues are all non-negative.
- The smallest eigenvalue of $\mathbf{L}$ is 0.
- Let $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ be the eigenvalues of $\mathbf{L}$. Then,
  - The multiplicity of the eigenvalue 0 equals the number of connected components in the graph. (Fiedler's Theorem).
  - $\lambda_2 > 0$ implies that G is connected.

Graph Laplacian

The Laplacian matrix $\mathbf{L}$ of a graph G with n vertices have the following properties:

- $\mathbf{L}$ is a positive semidefinite matrix. That is, it is equal to it's conjugate transpose and it's eigenvalues are all non-negative.
- The smallest eigenvalue of $\mathbf{L}$ is 0.
- Let $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ be the eigenvalues of $\mathbf{L}$. Then,
  - The multiplicity of the eigenvalue 0 equals the number of connected components in the graph. (Fiedler's Theorem).
  - $\lambda_2 > 0$ implies that G is connected.
  - If G has k components: $\lambda_1 = \lambda_2 = \cdots = \lambda_k = 0$.

Graph Laplacian

Consider the Laplacian matrix of the graph we defined a few slides ago:

$$L = \begin{bmatrix} 3 & -1 & 0 & -1 & -1 \\ -1 & 3 & -1 & 0 & -1 \\ 0 & -1 & 2 & -1 & 0 \\ -1 & 0 & -1 & 2 & 0 \\ -1 & -1 & 0 & 0 & 2 \end{bmatrix}$$

Graph Laplacian

Consider the Laplacian matrix of the graph we defined a few slides ago:

$$L = \begin{bmatrix} 3 & -1 & 0 & -1 & -1 \\ -1 & 3 & -1 & 0 & -1 \\ 0 & -1 & 2 & -1 & 0 \\ -1 & 0 & -1 & 2 & 0 \\ -1 & -1 & 0 & 0 & 2 \end{bmatrix}$$

It's characteristic polynomial is:

$$\det(\mathbf{L} - \lambda \mathbf{I}) = -\lambda^5 + 12\lambda_4 - 51\lambda^3 + 90\lambda^2 - 55\lambda$$

Graph Laplacian

Consider the Laplacian matrix of the graph we defined a few slides ago:

$$L = \begin{bmatrix} 3 & -1 & 0 & -1 & -1 \\ -1 & 3 & -1 & 0 & -1 \\ 0 & -1 & 2 & -1 & 0 \\ -1 & 0 & -1 & 2 & 0 \\ -1 & -1 & 0 & 0 & 2 \end{bmatrix}$$

It's characteristic polynomial is:

$$\det(\mathbf{L} - \lambda\mathbf{I}) = -\lambda^5 + 12\lambda_4 - 51\lambda^3 + 90\lambda^2 - 55\lambda$$

The eigenvalues of $\mathbf{L}$ are:

$$\lambda_1 = 0,\ \lambda_2 \approx 1.38196,\ \lambda_3 \approx 2.38196,\ \lambda_4 \approx 3.61803,\ \lambda_5 \approx 4.61803$$

Algebraic Connectivity

Definition (Algebraic Connectivity)

The **algebraic connectivity** $\alpha(G) = \lambda_2$ measures how well-connected the graph is.

Algebraic Connectivity

Definition (Algebraic Connectivity)

The **algebraic connectivity** $\alpha(G) = \lambda_2$ measures how well-connected the graph is.

Some properties of it are:

- $\alpha(G) = 0 \iff G$ is disconnected.
- Larger $\alpha(G)$ means better connectivity.
- Related to expansion properties.
- Used on bounds on diameter and vertex/edge connectivity

Algebraic Connectivity

Definition (Algebraic Connectivity)

The **algebraic connectivity** $\alpha(G) = \lambda_2$ measures how well-connected the graph is.

Some properties of it are:

- $\alpha(G) = 0 \iff G$ is disconnected.
- Larger $\alpha(G)$ means better connectivity.
- Related to expansion properties.
- Used on bounds on diameter and vertex/edge connectivity

Intuitively, $\alpha(G)$ measures the "cost" of disconnecting the graph.

Algebraic Connectivity

Definition (Algebraic Connectivity)

The **algebraic connectivity** $\alpha(G) = \lambda_2$ measures how well-connected the graph is.

Some properties of it are:

- $\alpha(G) = 0 \iff G$ is disconnected.
- Larger $\alpha(G)$ means better connectivity.
- Related to expansion properties.
- Used on bounds on diameter and vertex/edge connectivity

Intuitively, $\alpha(G)$ measures the "cost" of disconnecting the graph.
Examples:

- Path on 5 vertices: $\alpha(G) \approx 0.38$
- Complete graph on 5 vertices: $\alpha(G) = 5$

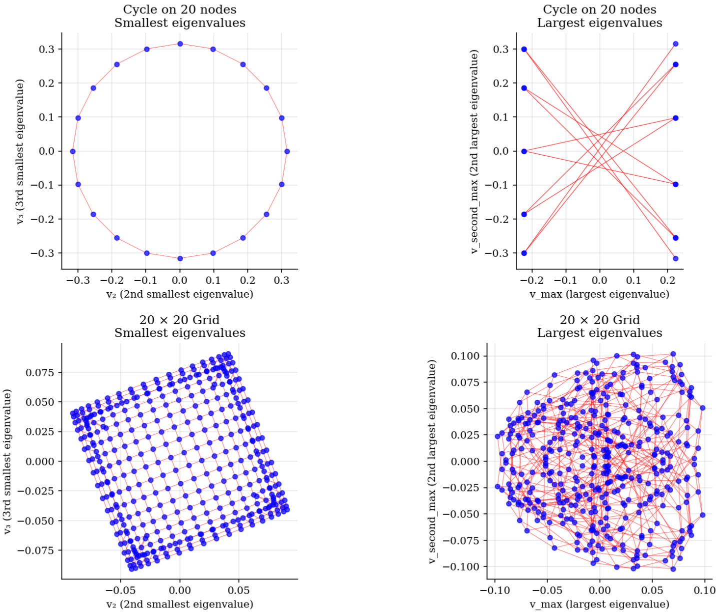
The Fiedler Vector

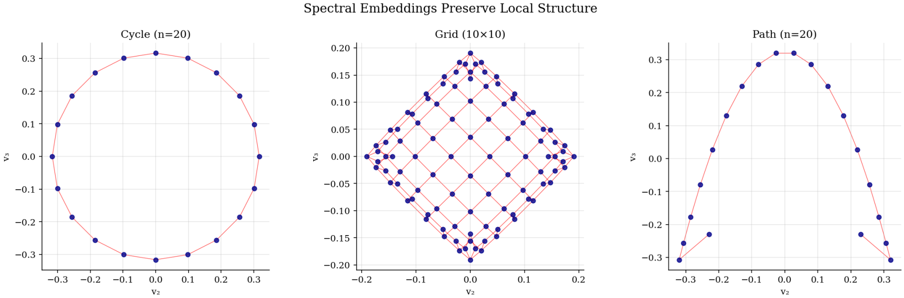
Definition

The **Fiedler vector** $\mathbf{v}_2$ is the eigenvector corresponding to λ_2.

It is pretty useful for stuff like graph connectivity, graph partitioning, embeddings, etc.

Spectral Embeddings: Smallest vs Largest Eigenvalues




Spectral Embeddings Preserve Local Structure
Cycle (n=20)
Grid (10×10)
Path (n=20)
v₂
v₃
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20

Switching gears, we now see an application of spectral graph theory for finding optimal paths in mazes!

Motivation

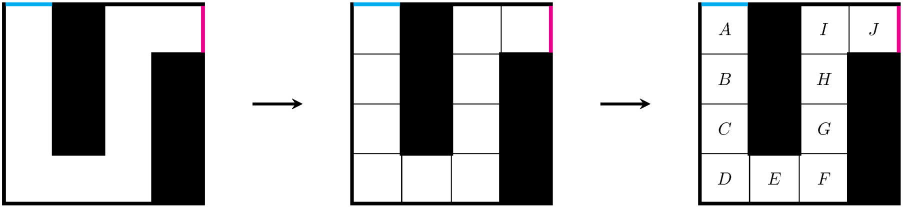
The rest of this talk is based on the paper: Martín-Nieto et al. 2024.

- Standard path-finding algorithms (Dijkstra, A*) prioritize shortest paths.
- But in design problems (e.g., robots, flow channels), **connectivity structure** matters more.
- We want a method that encourages:
  - Single-component (connected) regions
  - Smooth, realizable shapes
  - Optimization over a field rather than discrete steps

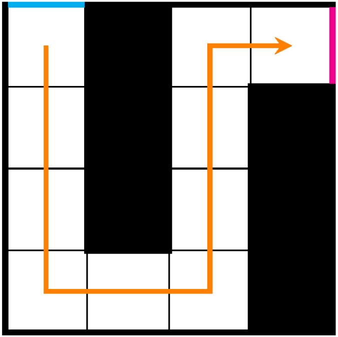
Overview of the Approach

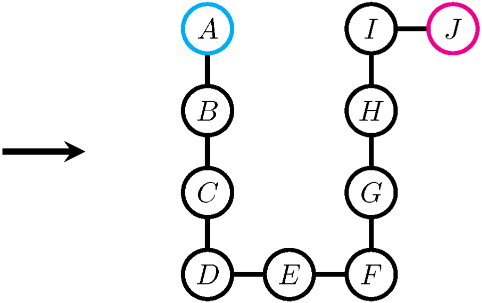
- Embed path-finding into a **continuous optimization** framework.
- Discretize the domain (image, mesh, space) into a graph $G = (V, E)$.
- Define a **mask function** $w : V \to [0, 1]$ over vertices, indicating inclusion in a region.
- Optimize the connectivity of the region $\{v \in V : w(v) > 0\}$ using **spectral properties** of G.

Discretization of the Maze



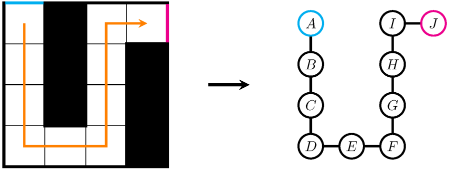
Solving the discretized maze





Computing the Laplacian

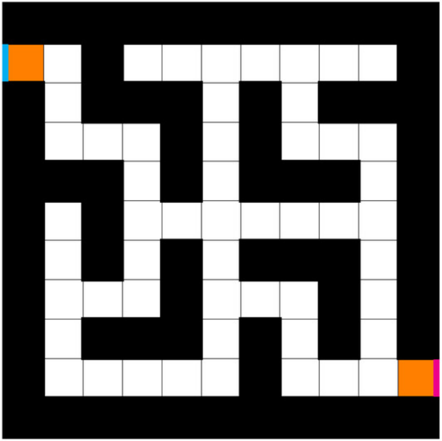
$$\mathcal{L} = \begin{pmatrix} 1 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \cdots & 0 \\ 0 & -1 & 2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$



With $\lambda_1 = 0$ and $\lambda_2 = \alpha(G) = 0.1$ which implies that the graph is made up of a single joined element.

What if the maze is not solved?

Consider this discretized unsolved maze:



The entrance and exit are cyan and magenta respectively.

What is being optimized?

Definition (Path weights)

Assign a **path weight** ρ_i for each cell in the grid. Each $\rho_i \in [0, 1]$ where:

- $\rho_i \approx 1 \rightarrow$ Cell is path of the path.
- $\rho_i \approx 0 \rightarrow$ Cell is a wall.

What is being optimized?

Definition (Path weights)

Assign a **path weight** ρ_i for each cell in the grid. Each $\rho_i \in [0, 1]$ where:

- $\rho_i \approx 1 \rightarrow$ Cell is path of the path.
- $\rho_i \approx 0 \rightarrow$ Cell is a wall.

Objective Function

Try to minimize the objective function:

$$\sum_i \rho_i \cdot w_i$$

Where w_i is some weight (e.g., Euclidean distance or just 1)

Main challenge

When minimizing the objective function:

$$\sum_i \rho_i \cdot w_i$$

The set of ρ_i that could be disconnected blobs. What could we do to ensure connectivity?

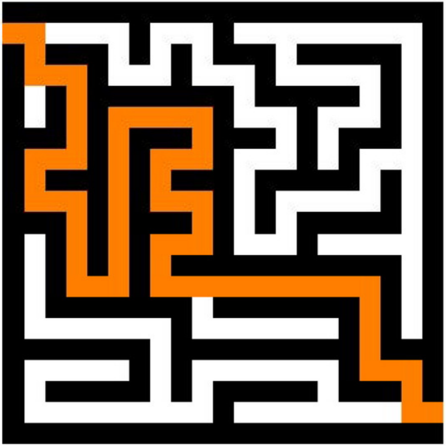
Enter the Graph Laplacian

Build a Graph Laplacian $L(\rho)$ weighted by the $\rho_i's$ and then enforce that:

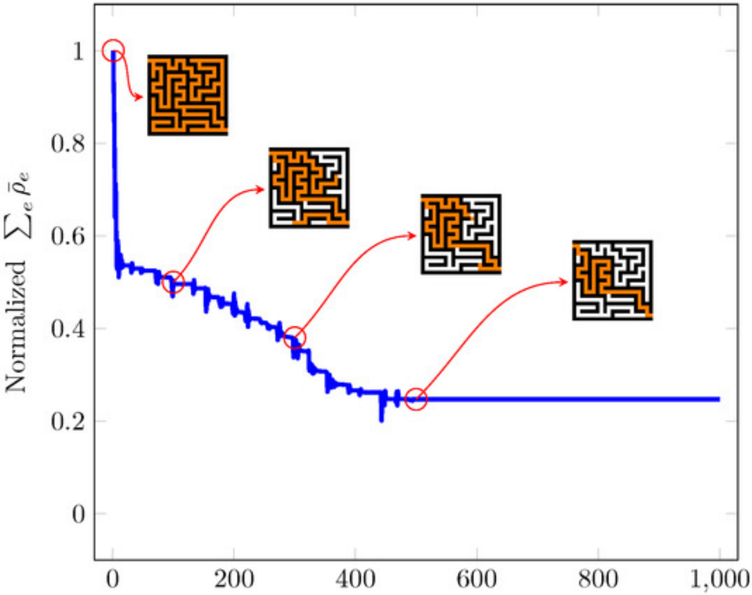
$$\alpha(G) = \lambda_2(L(\rho)) \geq \varepsilon$$

- This ensures that the blobs are connected!
- They then solve it using numerical methods that we will omit here due to insufficient time and my brain cannot comprehend that yet.

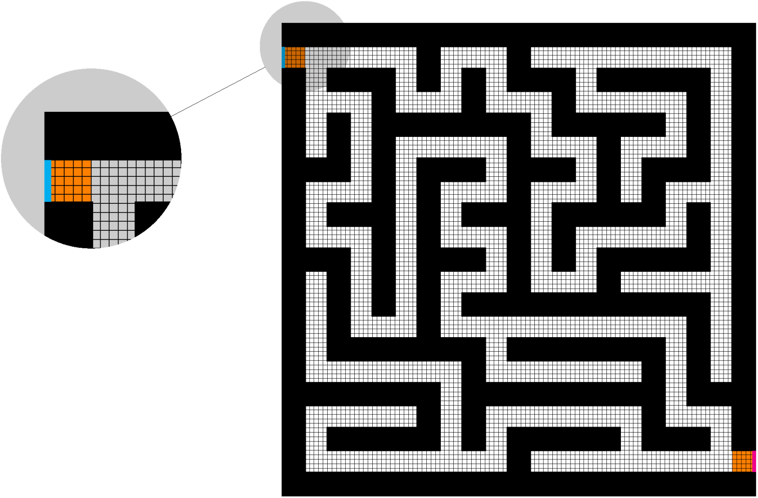
Solved Maze!



Evolution of the objective function

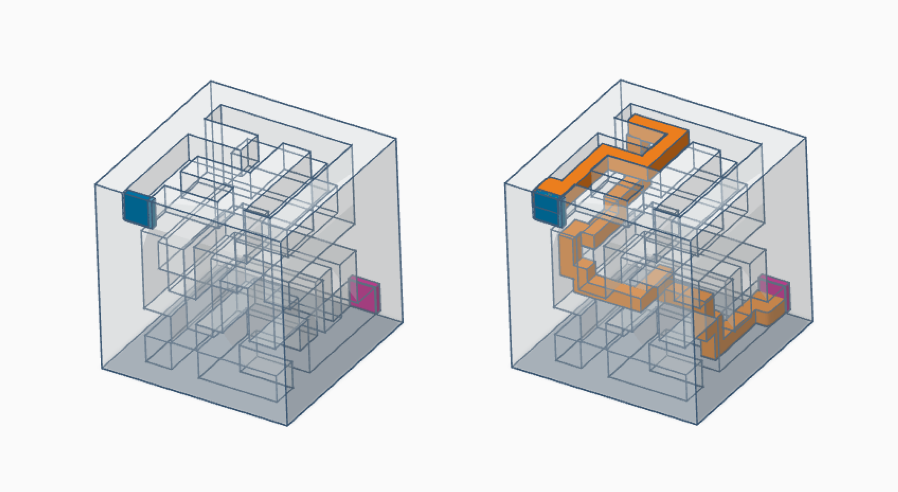


An example with finer discretization of the maze



An example with finer discretization of the maze

An example with a 3D maze



Comparison to Dijkstra/A*

- Dijkstra and A*:
  - Give shortest path.
  - Do not care about global structure.
- Spectral method:
  - No explicit path returned — returns connected regions.
  - Better suited for design problems: e.g., channel shapes, sensor regions, etc.
  - Optimizes over shapes, not paths.

References I

Martín-Nieto, Marta et al. (2024). "Solving Mazes: A New Approach Based on Spectral Graph Theory". In: *Mathematics* 12.15, p. 2305. DOI: 10.3390/math12152305. URL: https://doi.org/10.3390/math12152305.