Optimal Design of Diffusion-type Distributed Algorithms

Lin Xiao

Joint work with Stephen Boyd, Persi Diaconis, Sanjay Lall, Pablo Parrilo, and Jun Sun

CS286a: Mathematics of Information Seminar
10/29/2004
Setting and general problem

- **distributed process**: communication/computation on graphs

- **examples**: distributed consensus, distributed load balancing, resource allocation, sensing and estimation, coordination of autonomous agents, iterative solution of equations, Markov chains, . . .

- **diffusion-type algorithms**:  
  - no explicit point-to-point message passing or routing  
  - diffuse information by taking weighted average of neighbors’ data  
  - robust to component failures, works in asynchronous environment  
  - **weights** affect convergence behavior and/or stationary performance

how do we choose weights to yield fastest possible convergence?
Example: distributed average consensus

- compute average $\bar{x} = \frac{1}{n} \sum_i x_i$ (using local communication, iteration)
- each node takes a weighted average of its own and neighbors’ values:

$$x_i(t + 1) = W_{ii}x_i(t) + \sum_{j \in N_i} W_{ij}x_j(t)$$

- how do we choose $W$ to make convergence as fast as possible?
Example: Markov chain on a graph

- random walk on graph with symmetric transition probabilities $P_{ij}$

\[ \pi_i(t + 1) = P_{ii}\pi_i(t) + \sum_{j \in \mathcal{N}_i} P_{ij}\pi_i(t) \]

- (under simple conditions) distribution converges to uniform

- what edge transition probabilities give fastest mixing?
Example: distributed resource allocation

- resource allocation on a network

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{n} f_i(x_i) \\
\text{subject to} & \quad \sum_{i=1}^{n} x_i = c
\end{align*}
\]

- distributed weighted gradient method:

\[
x_i(t + 1) = x_i(t) - \sum_{j \in \mathcal{N}_i} W_{ij} \left( f'_i(x_i(t)) - f'_j(x_j(t)) \right)
\]

(exchange resources proportional to differences of marginal costs)

- how do we choose $W$ to make convergence as fast as possible?
Typical results

- establishing convergence conditions
- using SDP, we can optimize convergence rate (or a bound on it)
- by exploiting structure, associated SDPs can be efficiently solved
- many interesting applications
Example: distributed resource allocation

minimize \( \sum_{i=1}^{n} f_i(x_i) \)
subject to \( \sum_{i=1}^{n} x_i = c \)

\[ x_i(t + 1) = x_i(t) - \sum_{j \in \mathcal{N}_i} W_{ij} \left( f'_i(x_i(t)) - f'_j(x_j(t)) \right) \]
Why is this diffusion?

• diffusion equation on a line: \( u_t = k u_{xx} \)

\[
\frac{u_i(t + 1) - u_i(t)}{\Delta t} = k \frac{u_{i-1}(t) - 2u_i(t) + u_{i+1}(t)}{(\Delta x)^2}
\]

let \( s = \frac{k \Delta t}{(\Delta x)^2} \), can write difference equation as

\[
u_i(t + 1) = (1 - 2s)u_i(t) + su_{i-1}(t) + su_{i+1}(t) \]

• fast diffusion problem: given an arbitrary domain (discretized as a graph), design "conductivity" to make diffusion fast

\( u_t = \nabla \cdot (k \nabla u) \)
Fast distributed average consensus
Distributed average consensus

- compute average \( \bar{x} = \frac{1}{n} \sum_i x_i \) (using local communication, iteration)
- each node takes a weighted average of its own and neighbors’ values:

\[
x_i(t + 1) = W_{ii}x_i(t) + \sum_{j \in \mathcal{N}_i} W_{ij}x_j(t)
\]

usually \( W_{ij} > 0 \); but (surprisingly) this is not necessary for all edges
Convergence to average

- distributed iterative algorithm in vector form

\[ x(t + 1) = Wx(t) \quad \implies \quad x(t) = W^tx(0) \]

- convergence: for all initial values \( x(0) \in \mathbb{R}^n \)

\[ \lim_{t \to \infty} x(t) = \lim_{t \to \infty} W^tx(0) = \left( \frac{1}{n}1^Tx(0) \right)1 \]

- equivalent to the matrix equation

\[ \lim_{t \to \infty} W^t = \frac{11^T}{n} \]
Convergence conditions and rate

- **Theorem:** \( \lim_{t \to \infty} W^t = \frac{1}{n} \mathbf{1} \mathbf{1}^T \) if and only if
  \[
  1^T W = 1^T, \quad W \mathbf{1} = \mathbf{1}, \quad \rho(W - \frac{1}{n} \mathbf{1} \mathbf{1}^T) < 1
  \]

\( \rho(\cdot) \) denotes spectral radius of a matrix

- **Interpretations**
  - Sum (and therefore average) preserved: \( 1^T x(t+1) = 1^T x(t) \)
  - \( \mathbf{1} \) is fixed point of iteration \( x(t+1) = W x(t) \)
  - Iteration dynamics are stable on \( \mathbf{1}^\perp \)

- Asymptotic convergence rate given by \( \rho(W - \frac{1}{n} \mathbf{1} \mathbf{1}^T) \)
  (second largest eigenvalue modulus of \( W \))
Constant weights

constant weight on all edges:

\[ x_i(t + 1) = (1 - d_i \alpha)x_i(t) + \sum_{j \in \mathcal{N}_i} \alpha x_j(t) \]

\[ = x_i(t) + \sum_{j \in \mathcal{N}_i} \alpha(x_j(t) - x_i(t)) \]

• maximum-degree weight: \( \alpha = \frac{1}{\max_i d_i} \)
  \( d_i = |\mathcal{N}_i| \) is degree (number of neighbors) of node \( i \)

• best constant weight: \( \alpha^* = \frac{2}{\lambda_1(L) + \lambda_{n-1}(L)} \)
  \( L \) is Laplacian of graph
Metropolis weights

Metropolis-Hastings weights:

\[
W_{ij} = \frac{1}{\max\{d_i, d_j\}}, \quad \{i, j\} \in \mathcal{E}
\]

(self-weights given by \(W_{ii} = 1 - \sum_{j \in \mathcal{N}_i} W_{ij}\))

- adapted from Metropolis algorithms in Markov chain Monte Carlo
- based on local information, good for distributed implementation
- often gives reasonable convergence
Fastest distributed linear averaging

\[
\begin{align*}
\text{minimize} & \quad \rho(W - 11^T/n) \\
\text{subject to} & \quad 1^T W = 1^T, \quad W 1 = 1 \\
& \quad W_{ij} = 0 \text{ if } i \neq j \text{ and } \{i, j\} \notin \mathcal{E}
\end{align*}
\]

optimization variable is \(W\); problem data is graph (sparsity pattern of \(W\))

- hard problem when \(W\) is not symmetric

- can minimize convex upper bound \(\|W - 11^T/n\|\)

- for symmetric \(W\), \(\rho(W - 11^T/n) = \|W - 11^T/n\|\)
Semidefinite programming formulation

(for symmetric weights)

introduce scalar variable $s$ to bound spectral norm

\[
\begin{align*}
\text{minimize} & \quad s \\
\text{subject to} & \quad -sI \preceq W - \mathbf{1}\mathbf{1}^T/n \preceq sI \\
& \quad W = W^T, \quad W\mathbf{1} = \mathbf{1} \\
& \quad W_{ij} = 0 \text{ if } i \neq j \text{ and } \{i, j\} \notin \mathcal{E}
\end{align*}
\]

an SDP, hence, efficiently solved, duality theory, . . .
convergence factors and convergence times:

<table>
<thead>
<tr>
<th></th>
<th>max degree</th>
<th>Metropolis</th>
<th>optimal symm.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho(W - 11^T/n)$</td>
<td>0.746</td>
<td>0.743</td>
<td>0.600</td>
</tr>
<tr>
<td>$\tau = 1/\log(1/\rho)$</td>
<td>3.413</td>
<td>3.366</td>
<td>1.958</td>
</tr>
</tbody>
</table>
Optimal symmetric weights

(note: some weights are negative!)
A larger example

randomly generated network with 50 nodes, 200 edges

<table>
<thead>
<tr>
<th>$\rho(W - 11^T/n)$</th>
<th>max degree</th>
<th>Metropolis</th>
<th>best constant</th>
<th>optimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.971</td>
<td>0.949</td>
<td>0.947</td>
<td>0.902</td>
<td></td>
</tr>
<tr>
<td>$\tau = 1/\log(1/\rho)$</td>
<td>33.980</td>
<td>19.104</td>
<td>18.363</td>
<td>9.696</td>
</tr>
</tbody>
</table>
Distribution of optimal weights

(note: many weights are negative)
Eigenvalue distributions

- Maximum-degree
- Metropolis
- Best constant
- Optimal
Sparse network design

find a sparse subgraph and associated weight design, with guaranteed convergence rate: 
\[ \rho(W - \mathbf{1}\mathbf{1}^T/n) \leq \rho_{\text{max}} \]

- a difficult combinatorial optimization problem

- effective \( \ell_1 \) heuristic

\[
\begin{align*}
\text{minimize} & \quad \sum_{\{i,j\} \in \mathcal{E}} |W_{ij}| \\
\text{subject to} & \quad -\rho_{\text{max}} \mathbf{I} \preceq W - \mathbf{1}\mathbf{1}^T/n \preceq \rho_{\text{max}} \mathbf{I} \\
& \quad W \in \mathcal{S}, \quad W = W^T, \quad W\mathbf{1} = \mathbf{1}
\end{align*}
\]

\((\mathcal{S} \text{ specifies sparsity pattern of graph})\)
Sparse network design example

- sparse network design with $\rho_{\text{max}} = 0.910$ (best possible $\rho^* = 0.902$)
- number of edges used (with nonzero weights) reduced from 200 to 96
Comparison of weight distribution

distribution of weights by \( \ell_1 \) heuristic with \( \rho_{\text{max}} = 0.910 \)

![Graph showing weight distribution for \( \ell_1 \) heuristic.]

distribution of weights for best possible \( \rho^* = 0.902 \)

![Graph showing weight distribution for best possible \( \rho^* \).]
Distributed data fusion in sensor networks
Maximum likelihood estimation

- estimate a vector of unknown parameters \( \theta \in \mathbb{R}^m \) with \( n \) sensors
  \[ y_i = A_i \theta + v_i, \quad i = 1, \ldots, n \]

  measurements \( y_i \in \mathbb{R}^{m_i} \), noises \( v_i \sim \mathcal{N}(0, \Sigma_i) \) independent

- aggregate measurement \((\sum m_i \geq m)\)
  \[
y = A \theta + v = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{bmatrix} \theta + \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}
  \]

- maximum likelihood estimate given by weighted least-squares solution
  \[
  \hat{\theta}_{ML} = \left( A^T \Sigma^{-1} A \right)^{-1} A^T \Sigma^{-1} y
  \]
Sensor fusion schemes

• centralized sensor fusion
  – sensors send data to fusion center, via multi-hop relay, routing
  – fusion center computes WLS solution
  – vulnerable to component failures

• distributed sensor fusion
  – no fusion center; sensors have no global knowledge (e.g., topology)
  – each node compute $\hat{\theta}_{\text{ML}}$ (good for multiple tasks, active sensing)

• distributed sensor fusion based on average consensus
  – universal data structure, isotropic protocol
  – doesn’t involve routing; instead, diffuses information gradually
  – robust to component failures and network topology changes
Distributed sensor fusion: scalar case

- estimate a scalar parameter $\theta \in \mathbb{R}$, with i.i.d. noises $v_i \sim \mathcal{N}(0, \sigma^2)$

$$y_i = \theta + v_i, \quad i = 1, \ldots, n$$

- maximum likelihood estimate

$$\hat{\theta}_{ML} = \frac{1}{n} 1^T y, \quad \mathbb{E} (\hat{\theta}_{ML} - \theta)^2 = \frac{1}{n} \sigma^2$$

- distributed fusion based on average consensus
  - initialize $x_i(0) = y_i$, then iterate

$$x_i(t + 1) = W_{ii}x_i(t) + \sum_{j \in \mathcal{N}_i} W_{ij}x_j(t)$$

$$\lim_{t \to \infty} x_i(t) = \hat{\theta}_{ML} \text{ for all } i$$
Distributed sensor fusion: general case

\[ \hat{\theta}_{\text{ML}} = \left( A^T \Sigma^{-1} A \right)^{-1} A^T \Sigma^{-1} y = \left( \sum_{i=1}^{n} A_i^T \Sigma_i^{-1} A_i \right)^{-1} \sum_{i=1}^{n} A_i^T \Sigma_i^{-1} y_i \]

- each sensor initializes

\[ P_i(0) = A_i^T \Sigma_i^{-1} A_i, \quad q_i(0) = A_i^T \Sigma_i^{-1} y_i \]

- use distributed average consensus to compute (entrywise)

\[ \lim_{t \to \infty} P_i(t) = \frac{1}{n} \sum_{i=1}^{n} A_i^T \Sigma_i^{-1} A_i = P(\infty) \]
\[ \lim_{t \to \infty} q_i(t) = \frac{1}{n} \sum_{i=1}^{n} A_i^T \Sigma_i^{-1} y_i = q(\infty) \]

- each node can eventually compute \( \hat{\theta}_{\text{ML}} = P(\infty)^{-1} q(\infty) \)
Intermediate estimates

• instead of waiting for convergence, use

\[ \hat{\theta}_i(t) = P_i(t)^{-1} q_i(t) \]

available at node \( i \) as soon as \( P_i(t) \) invertible

• some nice properties
  – each \( \hat{\theta}_i(t) \) is a WLS solution; \( \lim_{t \to \infty} \hat{\theta}_i(t) = \hat{\theta}_{ML} \) for all \( i \)
  – unbiased estimate: \( \mathbb{E} \hat{\theta}_i(t) = \theta \) for all \( i \) and \( t \)
  – error covariance converges to optimal

\[ Q_i(t) = \mathbb{E} \left[ (\hat{\theta}_i(t) - \theta)(\hat{\theta}_i(t) - \theta)^T \right] \quad \longrightarrow \quad Q_{ML} = \left( A^T \Sigma^{-1} A \right)^{-1} \]
Distributed sensor fusion: example

• estimate position \((x_1, x_2)\) of target located in \([0, 1] \times [0.5, 1]\)

• 20 range sensors uniformly located in \([0, 1] \times [0, 0.5]\), with positions \((s_{i1}, s_{i2})\)

• each sensor measures distance to target \(r_i\), with additive noise \(v_i \sim \mathcal{N}(0, 0.1)\)

• sensor output

\[
y_i = r_i - a_i^T \begin{bmatrix} s_{i1} \\ s_{i2} \end{bmatrix} \approx a_i^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + v_i
\]

\(a_i\): unit vector from sensor to target
Summary of distributed sensor fusion scheme

- universal data structure for storage and communication

\[ P_i(t) \in \mathbb{R}^{m \times m}, \quad q_i(t) \in \mathbb{R}^m \]

independent of local dimension \( m_i \)

- isotropic protocol: taking weighted average of neighbors’ data

- every node can have medium accuracy within a few iterations; all converge to the global ML estimate

- using Metropolis weights, this scheme is robust to link failures and network topology changes, works in asynchronous environment
Average consensus in asynchronous environment

- communication links may work or fail at random (due to mobility, fading, power constraints); network topology changes with time

- some notations
  - $G(t) = (\mathcal{E}(t), \mathcal{V})$ time-varying communication graph
  - $\mathcal{N}_i(t) = \{j \in \mathcal{V} \mid \{i, j\} \in \mathcal{E}(t)\}$ instantaneous neighborhood
  - $\{G(t)\}_{t=0}^{\infty}$ can be either deterministic or stochastic

- distributed average consensus (same form)

\[
x_i(t + 1) = W_{ii}(t)x_i(t) + \sum_{j \in \mathcal{N}_i(t)} W_{ij}(t)x_j(t)
\]

what conditions on $\{G(t), W(t)\}_{t=0}^{\infty}$ guarantee convergence?
Choice of weights

- Maximum-degree weights

\[
W_{ij}(t) = \begin{cases} 
\frac{1}{n} & \text{if } \{i, j\} \in \mathcal{E}(t) \\
\frac{1}{n} - \frac{d_i(t)}{n} & \text{if } i = j \\
0 & \text{otherwise}
\end{cases}
\]

- Metropolis weights

\[
W_{ij}(t) = \begin{cases} 
\frac{1}{1 + \max\{d_i(t), d_j(t)\}} & \text{if } \{i, j\} \in \mathcal{E}(t) \\
1 - \sum_{\{i, k\} \in \mathcal{E}(t)} W_{ik}(t) & \text{if } i = j \\
0 & \text{otherwise}
\end{cases}
\]
Asynchronous convergence

- **Theorem**: if the communication graphs that occur infinitely often in \( \{G(t)\}_{t=0}^{\infty} \) are *jointly connected*, then the iteration

\[
x(t + 1) = W(t)x(t)
\]

converges to the average for any \( x(0) \in \mathbb{R}^n \), with either Metropolis weights or maximum-degree weights.

- A finite set of graphs with common vertex set \( G_i = (\mathcal{E}_i, \mathcal{V}) \), \( i = 1, \ldots, r \), are *jointly connected* if their union graph \( G = (\bigcup_{i=1}^r \mathcal{E}_i, \mathcal{V}) \) is connected.

- Roughly speaking: converges provided graphs "connected in a long run"
Asynchronous distributed sensor fusion

- extend directly to the general case: conduct average consensus on
  \[ P_i(0) = A_i^T \Sigma_i^{-1} A_i, \quad q_i(0) = A_i^T \Sigma_i^{-1} y_i \]
  and obtain \( \hat{\theta}_i(t) = P_i(t)^{-1} q_i(t) \)

- all nice properties still hold
  - each \( \hat{\theta}_i(t) \) is a WLS solution; \( \lim_{t \to \infty} \hat{\theta}_i(t) = \hat{\theta}_{ML} \) for all \( i \)
  - unbiased estimate: \( \mathbb{E} \hat{\theta}_i(t) = \theta \) for all \( i \) and \( t \)
  - error covariance converges to optimal
    \[ Q_i(t) = \mathbb{E} \left[ (\hat{\theta}_i(t) - \theta)(\hat{\theta}_i(t) - \theta)^T \right] \quad \rightarrow \quad Q_{ML} = (A^T \Sigma^{-1} A)^{-1} \]

- simulation example
Computational methods
Computational methods

- **interior-point methods:**
  - exploit sparsity and graph structure
  - can solve problems with a few thousand edges

- **subgradient-type methods:**
  - compute subgradient efficiently with Lanczos method
  - can solve problems with up to $10^6$ edges

- **exploiting symmetry:**
  - reduce number of variables
  - reduce size of matrices
Random walk on a graph

- transition probabilities $P_{ij} = \text{Prob} \left( X(t+1) = j \mid X(t) = i \right)$
- state distribution $\pi_i(t) = \text{Prob} (X(t) = i); \quad \pi(t+1)^T = \pi(t)^T P$
- convergence to stationary distribution: $\lim_{t \to \infty} \| \pi(t) - 1/n \| = 0$

what edge transition probabilities do we use to get fastest mixing?
Fastest mixing Markov chain (FMMC) problem

minimize \[ \mu(P) = \| P - (1/n)11^T \|_2 \]
subject to \[ P \geq 0, \quad P1 = 1, \quad P = P^T \]
\[ P_{ij} = 0, \quad i \neq j \text{ and } \{i, j\} \notin \mathcal{E} \]

SDP formulation

minimize \[ s \]
subject to \[ -sI \preceq P - (1/n)11^T \preceq sI \]
\[ P \geq 0, \quad P1 = 1, \quad P = P^T \]
\[ P_{ij} = 0, \quad i \neq j \text{ and } \{i, j\} \notin \mathcal{E} \]
Graph automorphisms

graph $G = (\mathcal{V}, \mathcal{E})$

- **automorphism**: a permutation $\pi$ of $\mathcal{V}$ such that

$$\{i, j\} \in \mathcal{E} \iff \{\pi(i), \pi(j)\} \in \mathcal{E}$$

induced action on edges

$$\pi\{i, j\} = \{\pi(i), \pi(j)\}$$

- **automorphism group** $\text{Aut}(G)$: set of all such permutations, with group operation being composition
Example of graph automorphisms

\[
\text{Aut}(G) \text{ isomorphic to } S_3
\]
Classes of graph symmetry

- **vertex-transitive**: for every two vertices \(i, k \in V\), there exists \(\pi \in \text{Aut}(\mathcal{G})\) such that \(\pi(i) = k\)

- **edge-transitive**: for every two edges \(\{i, j\}, \{h, k\} \in E\), there exists \(\pi \in \text{Aut}(\mathcal{G})\) such that \(\pi\{i, j\} = \{h, k\}\)

- **distance-transitive**: for any four vertices \(u, v, x, y\) with \(\delta(u, v) = \delta(x, y)\), there exists \(\pi \in \text{Aut}(\mathcal{G})\) with \(\pi(u) = x\) and \(\pi(v) = y\)
Action on transition probability matrix

- representation of automorphism group $\rho : \text{Aut}(\mathcal{G}) \to \text{GL}(n, \mathbb{C})$

\[ \rho(\pi_1 \pi_2) = \rho(\pi_1) \rho(\pi_2) \]

here we use the usual permutation matrix

\[ \rho(\pi) = Q, \quad \text{where} \quad Q_{ij} = \begin{cases} 1 & \text{if } i = \pi(j) \\ 0 & \text{otherwise} \end{cases} \]

- induced action on transition probability matrix

\[ \pi(P) = QPQ^T \]

sparsity pattern preserved: $h = \pi(i), k = \pi(j) \implies (\pi(P))_{hk} = P_{ij}$
Fix-point subset

• feasible set invariant under action $\pi(P) = QPQ^T$

$$\mathcal{C} = \left\{ P \in \mathbb{R}^{n \times n} \mid \begin{array}{l} P \geq 0, \; P1 = 1, \; P = P^T, \\ P_{ij} = 0, \; i \neq j \; \text{and} \; \{i, j\} \notin \mathcal{E} \end{array} \right\}$$

• define fixed-point subset

$$\mathcal{F} = \{ P \in \mathcal{C} \mid \pi(P) = P, \; \pi \in \text{Aut}(\mathcal{G}) \}$$

• what happens if we restrict FMMC problem on $\mathcal{F}$?
**Theorem:** FMMC problem always has an optimal solution in $\mathcal{F}$

**Proof:**

- There is at least one optimal solution $P^* \in \mathcal{C}$
- Define the average over orbit of $P^*$

\[ \overline{P} = \frac{1}{|\text{Aut}(\mathcal{G})|} \sum_{\pi \in \text{Aut}(\mathcal{G})} \pi(P^*) \]

$\overline{P}$ is feasible; moreover, $\overline{P} \in \mathcal{F}$

- $\mu(\overline{P}) \leq \mu(P^*)$ by convexity of $\mu$; hence $\overline{P}$ is optimal
Reducing number of variables

can solve

\[
\begin{align*}
\text{minimize} & \quad \mu(P) \\ 
\text{subject to} & \quad P \in \mathcal{F}
\end{align*}
\]

instead of

\[
\begin{align*}
\text{minimize} & \quad \mu(P) \\ 
\text{subject to} & \quad P \in \mathcal{C}
\end{align*}
\]

corollaries

- number of distinct edge transition probabilities we need to consider at most equal to number of orbits of \( \mathcal{E} \) under \( \text{Aut}(\mathcal{G}) \)

- for edge-transitive graphs, all edge transition probabilities can be assigned same value

(staying probabilities at vertices eliminated using \( P_{ii} = 1 - \sum_{j \neq i} P_{ij} \))
Edge-transitive graphs

let $\alpha$ be the transition probability assigned on all edges

$$P = I - \alpha L, \quad L = \text{diag}(d_1, \ldots, d_n) - A$$

($L$: Laplacian, $A$: adjacency matrix, $d_i$: degree of vertex $i$)

**Theorem:** FMMC on edge-transitive graphs is given by

$$\alpha^* = \min \left\{ \frac{1}{d_{\max}}, \frac{2}{\lambda_1(L) + \lambda_{n-1}(L)} \right\}$$

$$\mu^* = \max \left\{ 1 - \frac{\lambda_{n-1}(L)}{d_{\max}}, \frac{\lambda_1(L) - \lambda_{n-1}(L)}{\lambda_1(L) + \lambda_{n-1}(L)} \right\}$$

where $d_{\max} = \max_{i \in \mathcal{V}} d_i$ is maximum valency
proof

• relationship between eigenvalues

\[ P = I - \alpha L \implies \lambda_i(P) = 1 - \alpha \lambda_{n+1-i}(L), \quad i = 1, \ldots, n \]

• SLEM

\[ \mu(P) = \max\{\lambda_2(P), -\lambda_n(P)\} \]

\[ = \max\{1 - \alpha \lambda_{n-1}(L), \alpha \lambda_1(L) - 1\} \]

• minimizing \( \mu(P) \) subject to \( 0 < \alpha \leq \frac{1}{d_{\text{max}}} \) gives optimal solution
Examples

complete graph $K_n$

Petersen graph

$\alpha^* = \frac{1}{n}, \quad \mu^* = 0$

$\alpha^* = \frac{2}{7}, \quad \mu^* = \frac{3}{7}$
Block diagonalization

• decompose \( \rho : \text{Aut}(G) \to \text{GL}(n, \mathbb{C}) \) into irreducible representations

\[
\rho = m_1 \vartheta_1 \oplus m_2 \vartheta_2 \oplus \cdots \oplus m_h \vartheta_h
\]

• isotypic decomposition of representation space \( \mathbb{C}^n \)

\[
\mathbb{C}^n = V_1 \oplus \cdots \oplus V_h, \quad V_i = V_{i1} \oplus \cdots \oplus V_{imi}, \quad \dim(\vartheta_i) = n_i
\]

• for \( P \in \mathcal{F} \) (\( \rho(\pi)P = P \rho(\pi), \forall \pi \in \text{Aut}(G) \)), there exists \( T \) such that

\[
T^{-1}PT = \begin{bmatrix}
P_1 & & \\
& \ddots & \\
& & P_h
\end{bmatrix}, \quad P_i = \begin{bmatrix}
\tilde{P}_1 & & \\
& \ddots & \\
& & \tilde{P}_i
\end{bmatrix}, \quad \dim(\tilde{P}_i) = m_i
\]

• columns of \( T \): symmetry adapted basis; can be computed explicitly
Symmetry reduced LMI constraints

- original formulation

\[- sI_n \preceq P - (1/n)11^T \preceq sI_n\]

- with block diagonal form

\[- sI_{m_i} \preceq \tilde{P}_i - \tilde{J}_i \preceq sI_{m_i}, \quad i = 1, \ldots, h\]

note: \( J = (1/n)11^T \) also satisfies \( \rho(\pi)J = J\rho(\pi), \forall \pi \in \text{Aut}(\mathcal{G}) \)

symmetry reduction

- number of variables: number of edges \( \longrightarrow \) number of orbits
- size of matrices:

one LMI with dimension \( n = \sum_{i=1}^{h} m_i n_i \quad \longrightarrow \quad h \) LMIs, each with dimension \( m_i \)
Example: FMMC on grid graph

- $\text{Aut}(G)$ isomorphic to dihedral group $D_4$ (8 elements)
- neither vertex- nor edge-transitive
- two orbits, assigned transition probability $a$ and $b$; matrix $P =$

\[
\begin{bmatrix}
1 - 2a & a & 0 & a & 0 & 0 & 0 & 0 & 0 \\
a & 1 - 2a - b & a & 0 & b & 0 & 0 & 0 & 0 \\
0 & a & 1 - 2a & 0 & 0 & a & 0 & 0 & 0 \\
a & 0 & 0 & 1 - 2a - b & b & 0 & a & 0 & 0 \\
0 & b & 0 & b & 1 - 4b & b & 0 & b & 0 \\
0 & 0 & a & 0 & b & 1 - 2a - b & 0 & 0 & a \\
0 & 0 & 0 & a & 0 & 0 & 1 - 2a & a & 0 \\
0 & 0 & 0 & 0 & b & 0 & a & 1 - 2a - b & a \\
0 & 0 & 0 & 0 & 0 & a & 0 & a & 1 - 2a \\
\end{bmatrix}
\]
- coordinate transformation matrix

\[ T = \frac{1}{2} \begin{bmatrix}
0 & 1 & 0 & 1 & 0 & \sqrt{2} & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & -1 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & -1 & 0 & 0 & 0 & \sqrt{2} & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & -1 \\
2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & -1 & 0 & 1 \\
0 & 1 & 0 & -1 & 0 & 0 & 0 & -\sqrt{2} & 0 \\
0 & 0 & 1 & 0 & -1 & 0 & -1 & 0 & -1 \\
0 & 1 & 0 & 1 & 0 & -\sqrt{2} & 0 & 0 & 0
\end{bmatrix} \]

- block diagonalization: \( T^T P T = \)

\[
\begin{bmatrix}
1 - 4b & 0 & 2b \\
0 & 1 - 2a & 2a \\
2b & 2a & 1 - 2a - b
\end{bmatrix}
\begin{bmatrix}
1 - 2a & 1 - 2a - b \\
1 - 2a & 2a \\
\sqrt{2a} & 1 - 2a - b
\end{bmatrix}
\begin{bmatrix}
1 - 4b & 0 & 2b \\
0 & 1 - 2a & 2a \\
2b & 2a & 1 - 2a - b
\end{bmatrix}
\]

- optimal solution: \( a^* \approx 0.363, \quad b^* \approx 0.211, \quad \mu^* \approx 0.693 \)
Beyond computational methods . . .
FMMC on a path

how would the answer change as $n$ increases?
A good guess

the same answer for all $n$?
FMMC on a path

\[
P^* = \begin{bmatrix}
  \frac{1}{2} & \frac{1}{2} \\
  \frac{1}{2} & 0 & \frac{1}{2} \\
  & \ddots & \ddots & \ddots \\
  \frac{1}{2} & 0 & \frac{1}{2} \\
  \frac{1}{2} & \frac{1}{2} & \frac{1}{2}
\end{bmatrix}, \quad \mu(P^*) = \cos \left( \frac{\pi}{n} \right)
\]

**Theorem:** \( P^* \) attains the smallest value of \( \mu \), among all symmetric stochastic tridiagonal matrices

(proof by SDP duality)
Summary

- **applications**: distributed communication/computation on graphs

- **diffusion-type algorithms**:
  - no explicit point-to-point message passing or routing
  - diffuse information by taking weighted average of neighbors’ data
  - robust to component failures, works in asynchronous environment
  - **weights** affect convergence behavior and/or stationary performance

**typical results:**

- using SDP, we can optimize convergence rate (or a bound on it)
- by exploiting structure, associated SDPs can be efficiently solved
- many variations: sparsity design, asynchronous convergence, minimum mean-square deviation, duality with manifold unfolding, . . .