On a combinatorial optimization problem involving the graph Laplacian matrix

Fu Lin[1] Makan Fardad[2] Mihailo Jovanović[3]

[1]Argonne National Laboratory

[2]Syracuse University

[3]University of Minnesota

October 25, 2013

Research overview

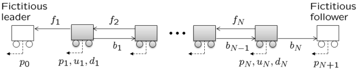
Optimization and control of dynamical systems on networks

SPECIFIC TOPICS:

- Localized control of vehicular formations
- Sparsity-promoting optimal control
- Sparse consensus networks
- Leader selection in consensus networks

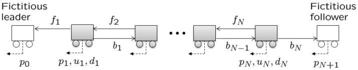
Research overview

Localized control of vehicular formations

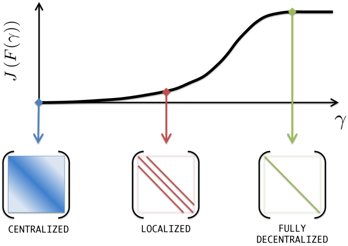


Research overview

Localized control of vehicular formations

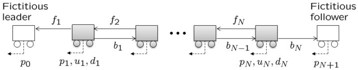


Sparsity-promoting optimal control

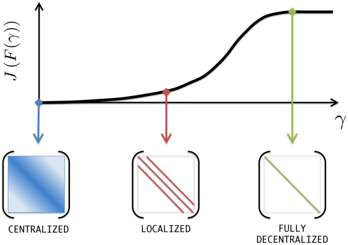


Research overview

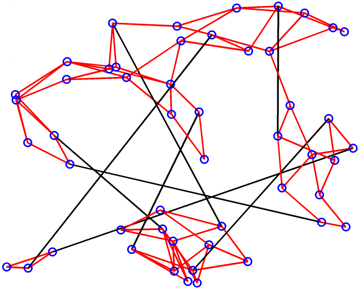
Localized control of vehicular formations



Sparsity-promoting optimal control

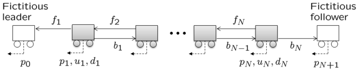


Sparse consensus networks

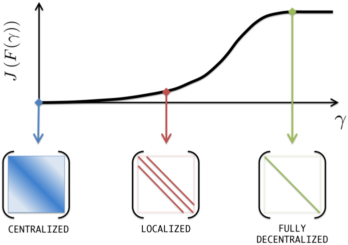


Research overview

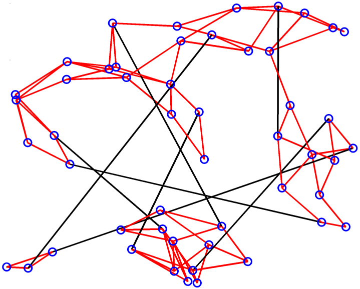
Localized control of vehicular formations



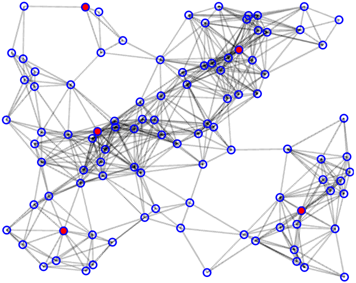
Sparsity-promoting optimal control



Sparse consensus networks



Leader selection



Research overview

CHALLENGES:

- Networks – combinatorial objects
- Optimization – constrained nonconvex problems

Research overview

CHALLENGES:

- Networks – combinatorial objects
- Optimization – constrained nonconvex problems

APPROACH:

- Identify classes of convex problems (or relaxations)
- Exploit problem structure to develop efficient algorithms

In this talk: Leader selection

- A combinatorial problem involving graph Laplacian
- Applications in vehicular formations and sensor localization
- Lower and upper bounds on global optimal solutions
- Examples from regular lattices and random networks

The problem

- Given the Laplacian matrix of a connected graph $L \in \mathbb{R}^{n \times n}$

  delete k columns and rows such that

The problem

- Given the Laplacian matrix of a connected graph $L \in \mathbb{R}^{n \times n}$

  delete k columns and rows such that

$$\text{minimize} \quad \operatorname{trace}\left(L_f^{-1}\right)$$

$$L_f \text{ is the principal submatrix in } L = \begin{bmatrix} L_l & L_0^T \\ L_0 & L_f \end{bmatrix}$$

An example

Deleting one column and one row

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

An example

Deleting one column and one row

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

delete 1st row and col.

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

An example

Deleting one column and one row

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

delete 1st row and col.

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

2nd row and col.

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

An example

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

Deleting one column and one row

delete 1st row and col. 2nd row and col. 3rd row and col.

$$L_f = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & 0 & 0 \\ 0 & 0 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

An example

Deleting one column and one row

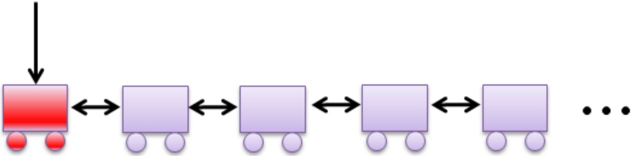
$$L = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

delete 1st row and col. 2nd row and col. 3rd row and col.

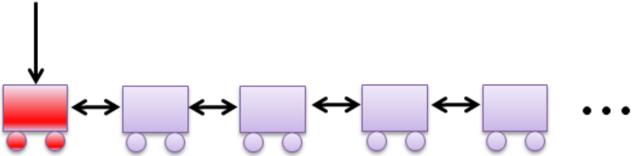
$$L_f = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & 0 & 0 \\ 0 & 0 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

$$\operatorname{trace}\left(L_f^{-1}\right) = 10, \quad 7, \quad 6$$

Applications: Formation of vehicles

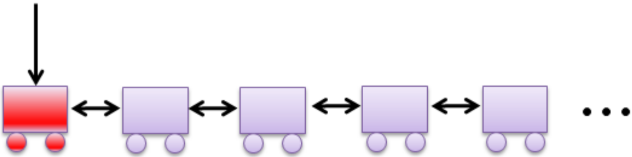


Applications: Formation of vehicles



$$\text{FOLLOWER:} \quad \dot{\psi}_i(t) = -\sum_{j \in \mathcal{N}_i} (\psi_i(t) - \psi_j(t)) + w_i(t) \quad \leftarrow \text{noise}$$

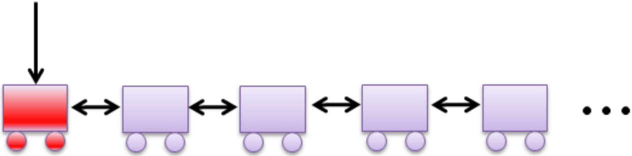
Applications: Formation of vehicles



$$\text{FOLLOWER:} \quad \dot{\psi}_i(t) \;=\; -\sum_{j \in \mathcal{N}_i} (\psi_i(t) \;-\; \psi_j(t)) \;+\; w_i(t) \qquad \leftarrow \text{ noise}$$

$$\text{LEADER:} \quad \dot{\psi}_i(t) \;=\; 0$$ no deviation from desired trajectory

Applications: Formation of vehicles



$$\text{FOLLOWER: } \quad \dot{\psi}_i(t) \;=\; -\sum_{j \in \mathcal{N}_i} (\psi_i(t) \;-\; \psi_j(t)) \;+\; w_i(t) \qquad \leftarrow \text{ noise}$$

$$\text{LEADER: } \quad \dot{\psi}_i(t) \;=\; 0 \qquad \text{no deviation from desired trajectory}$$

- Select k leaders to minimize the variance of followers

Applications: Sensor localization

- GOAL: Estimate n sensor positions in 1D $\quad \psi \in \mathbb{R}^n$

Applications: Sensor localization

- GOAL: Estimate n sensor positions in 1D $\quad \psi \in \mathbb{R}^n$

Relative measurements corrupted by white noise

$$y_r = \psi_i - \psi_j + w_r$$

Applications: Sensor localization

- GOAL: Estimate n sensor positions in 1D $\quad \psi \in \mathbb{R}^n$

Relative measurements corrupted by white noise

$$y_r \;=\; \psi_i \;-\; \psi_j \;+\; w_r$$

$$y \;=\; E^T \psi \;+\; w$$

Applications: Sensor localization

- GOAL: Estimate n sensor positions in 1D $\qquad \psi \in \mathbb{R}^n$

Relative measurements corrupted by white noise

$$y_r \;=\; \psi_i \;-\; \psi_j \;+\; w_r$$

$$y \;=\; E^T \psi \;+\; w$$

Anchor nodes with known positions $\quad \psi_l$

$$y \;=\; \begin{bmatrix} E_l \\ E_f \end{bmatrix}^T \begin{bmatrix} \psi_l \\ \psi_f \end{bmatrix} \;+\; w$$

Applications: Sensor localization

Laplacian of measurement graph

$$L = EE^T = \begin{bmatrix} E_l E_l^T & E_l E_f^T \\ E_f E_l^T & E_f E_f^T \end{bmatrix} = \begin{bmatrix} L_l & L_0^T \\ L_0 & L_f \end{bmatrix}$$

Applications: Sensor localization

Laplacian of measurement graph

$$L = EE^T = \begin{bmatrix} E_l E_l^T & E_l E_f^T \\ E_f E_l^T & E_f E_f^T \end{bmatrix} = \begin{bmatrix} L_l & L_0^T \\ L_0 & L_f \end{bmatrix}$$

Total variance of estimation error

$$\operatorname{trace}\left(L_f^{-1}\right)$$

Applications: Sensor localization

Laplacian of measurement graph

$$L = EE^T = \begin{bmatrix} E_l E_l^T & E_l E_f^T \\ E_f E_l^T & E_f E_f^T \end{bmatrix} = \begin{bmatrix} L_l & L_0^T \\ L_0 & L_f \end{bmatrix}$$

Total variance of estimation error

$$\mathrm{trace}\left(L_f^{-1}\right)$$

- Select k anchors to minimize variance of estimation error

Applications: Sensor localization

Laplacian of measurement graph

$$L \;=\; EE^T \;=\; \begin{bmatrix} E_l E_l^T & E_l E_f^T \\ E_f E_l^T & E_f E_f^T \end{bmatrix} \;=\; \begin{bmatrix} L_l & L_0^T \\ L_0 & L_f \end{bmatrix}$$

Total variance of estimation error

$$\mathrm{trace}\left(L_f^{-1}\right)$$

- Select k anchors to minimize variance of estimation error

many other applications in networks...

Related work

- Greedy algorithms with approximations

  Patterson and Bamieh '10

- Submodular optimization with performance guarantees

  Clark and Poovendran '11,   Clark et al. '12, '13

- Semidefinite programming for related sensor selection problem

  Joshi and Boyd '09

- Information-centrality-based approach

  Fitch and Leonard '13

In this talk

- Related problem on diagonally strengthened graph Laplacian
- Efficient algorithms for bounds on global optimal value
  - Convex relaxations – lower bounds
  - Greedy algorithms – upper bounds (exploiting low-rank structure)
- Examples from regular lattices and random networks

Diagonally strengthened graph Laplacian

- Arise in several applications
- Give insights to submatrix selection problem
- Easier to solve ;-)

Diagonally strengthened graph Laplacian

- Given L and $\alpha > 0$, select k diagonal elements of L to strengthen

$$
\begin{array}{ll}
\underset{x}{\text{minimize}} & J(x) \;=\; \operatorname{trace}\left((L + \alpha\,\operatorname{diag}(x))^{-1}\right) \\[1em]
\text{subject to} & x_i \in \{0,1\}, \qquad i \;=\; 1,\ldots,n \\[1em]
& \mathbb{1}^T x \;=\; k
\end{array}
$$

Diagonally strengthened graph Laplacian

- Given L and $\alpha > 0$, select k diagonal elements of L to strengthen

$$\begin{array}{lrcl} \underset{x}{\text{minimize}} & J(x) & = & \operatorname{trace}\left((L + \alpha \operatorname{diag}(x))^{-1}\right) \\ \text{subject to} & x_i \in \{0,1\}, & & i = 1, \ldots, n \\ & \mathbb{1}^T x & = & k \end{array}$$

Recover submatrix selection problem $\alpha \to \infty$

$$\begin{bmatrix} L_l + \alpha I & L_0^T \\ L_0 & L_f \end{bmatrix}^{-1} \to \begin{bmatrix} 0 & 0 \\ 0 & L_f^{-1} \end{bmatrix}$$

Interpretation: Noise-corrupted leaders

FOLLOWERS: $$\dot{\psi}_i(t) \;=\; -\sum_{j\,\in\,\mathcal{N}_i} (\psi_i(t) \;-\; \psi_j(t)) \;+\; w_i(t)$$

LEADERS: $$\dot{\psi}_i(t) \;=\; -\sum_{j\,\in\,\mathcal{N}_i} (\psi_i(t) \;-\; \psi_j(t)) \;-\; \alpha\,\psi_i(t) \;+\; w_i(t)$$

$$\alpha \;>\; 0$$

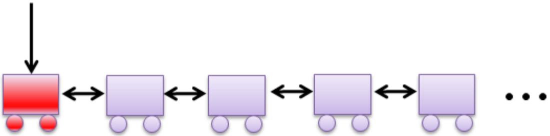
Interpretation: Noise-corrupted leaders

FOLLOWERS: $$\dot{\psi}_i(t) \;=\; - \sum_{j \in \mathcal{N}_i} (\psi_i(t) \;-\; \psi_j(t)) \;+\; w_i(t)$$

LEADERS: $$\dot{\psi}_i(t) \;=\; - \sum_{j \in \mathcal{N}_i} (\psi_i(t) \;-\; \psi_j(t)) \;-\; \alpha\, \psi_i(t) \;+\; w_i(t)$$

$$\alpha \;>\; 0$$

Leaders have GPS devices and know their own positions



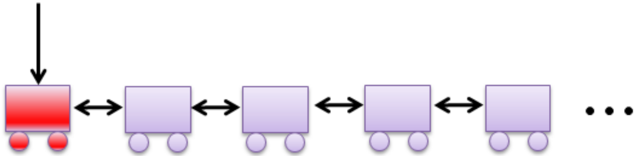
$$\dot{\psi}(t) \; = \; -\left(L \; + \; \alpha \; \mathrm{diag}\,(x)\right)\psi(t) \; + \; w(t)$$

$x_i \;\in\; \{0,1\}$, $\quad 1$ – LEADER, $\quad 0$ – FOLLOWER

$$\dot{\psi}(t) \;=\; -\,(L + \alpha\,\mathrm{diag}\,(x))\,\psi(t) \;+\; w(t)$$

$x_i \in \{0, 1\}$, $\qquad$ 1 – LEADER, $\qquad$ 0 – FOLLOWER

$$L = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}, \qquad \mathrm{diag}\,(x) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$



Algorithms for noise-corrupted formulation

$$
\begin{aligned}
\underset{x}{\text{minimize}} \quad & J(x) = \operatorname{trace}\left((L + \alpha \operatorname{diag}(x))^{-1}\right) \\
\text{subject to} \quad & x_i \in \{0,1\}, \qquad i = 1,\ldots,n \\
& \mathbb{1}^T x = k
\end{aligned}
$$

Algorithms for noise-corrupted formulation

$$
\begin{aligned}
\underset{x}{\text{minimize}} \quad J(x) &= \operatorname{trace}\left((L + \alpha \operatorname{diag}(x))^{-1}\right) \\
\text{subject to} \quad x_i &\in \{0, 1\}, \qquad i = 1, \ldots, n \\
\mathbb{1}^T x &= k
\end{aligned}
$$

FEATURES:

- Convex objective function
- Boolean constraints

Algorithms for noise-corrupted formulation

$$\begin{array}{rrcl} \underset{x}{\text{minimize}} & J(x) & = & \operatorname{trace}\left((L + \alpha \operatorname{diag}(x))^{-1}\right) \\ \text{subject to} & x_i & \in & \{0,1\}, \qquad i = 1, \ldots, n \\ & \mathbb{1}^T x & = & k \end{array}$$

FEATURES:

- Convex objective function
- Boolean constraints

APPROACH:

- Convex relaxation $\Rightarrow$ lower bound
- Greedy algorithm $\Rightarrow$ upper bound

Convex relaxation

$$
\begin{array}{ll}
\underset{x}{\text{minimize}} & J(x) \;=\; \operatorname{trace}\left((L \;+\; \alpha \operatorname{diag}(x))^{-1}\right) \\[2ex]
\text{subject to} & x_i \;\in\; [0,1], \qquad i \;=\; 1,\ldots,n \\[2ex]
& \mathbb{1}^T x \;=\; k
\end{array}
$$

Enlarge feasible set $\Rightarrow$ lower bound

Convex relaxation

$$\begin{array}{lrcl} \underset{x}{\text{minimize}} & J(x) & = & \operatorname{trace}\left((L \; + \; \alpha \operatorname{diag}(x))^{-1}\right) \\ \text{subject to} & x_i & \in & [0,1], \qquad i \; = \; 1, \ldots, n \\ & \mathbb{1}^T x & = & k \end{array}$$

Enlarge feasible set $\Rightarrow$ lower bound

- SDP formulation with complexity $O(n^4)$ – number of nodes
- Customized interior point method $O(n^3)$

Greedy algorithm

- One-leader-at-a-time

$$L + \alpha\, e_i e_i^T$$

RANK-1 UPDATE: $O(n^2)$ per leader

number of leaders $k \ll n \quad \Rightarrow \quad O(n^3)$ with one matrix inverse

Greedy algorithm

- One-leader-at-a-time

$$L \; + \; \alpha \, e_i e_i^T$$

RANK-1 UPDATE: $O(n^2)$ per leader

number of leaders $k \ll n \quad \Rightarrow \quad O(n^3)$ with one matrix inverse

After selecting k leaders

- Swap a leader and a follower

$$L \; - \; \alpha \, e_i e_i^T \; + \; \alpha \, e_j e_j^T$$

RANK-2 UPDATE: $O(n^2)$ per swap

Recap

$$\begin{array}{lrcl} \underset{x}{\text{minimize}} & J(x) & = & \operatorname{trace}\left((L + \alpha \operatorname{diag}(x))^{-1}\right) \\ \text{subject to} & x_i & \in & \{0,1\}, \qquad i = 1,\ldots,n \\ & \mathbb{1}^T x & = & k \end{array}$$

- Convex relaxation $\Rightarrow$ lower bound

  Standard SDP formulation $O(n^4)$

  Customized interior point method $O(n^3)$

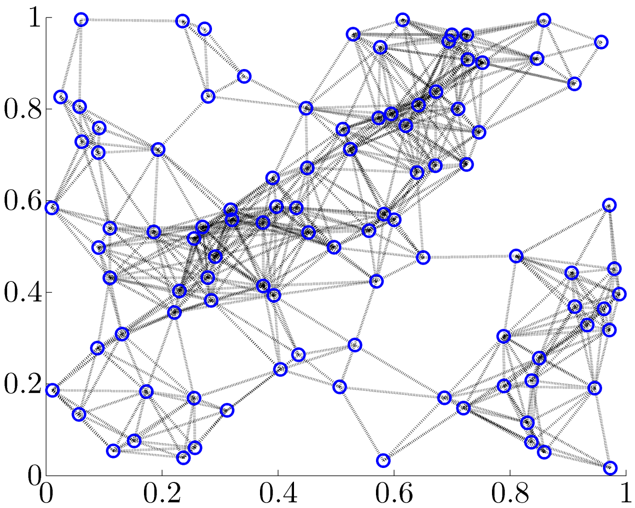
- Greedy algorithm $\Rightarrow$ upper bound

  Without exploiting structure $O(n^4 k)$

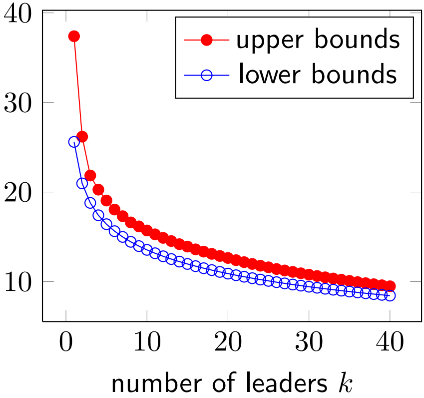
  Low rank updates $O(\max\{n^3, n^2 k\})$

Lin, Fardad, and Jovanović, IEEE CDC '11

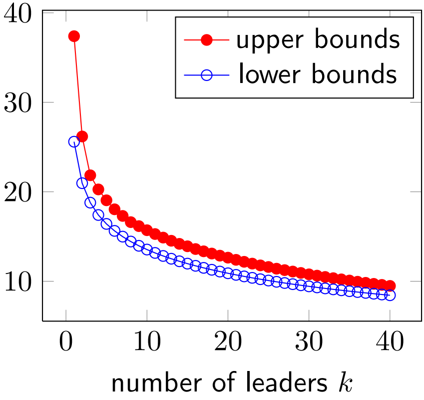
A random network with 100 nodes

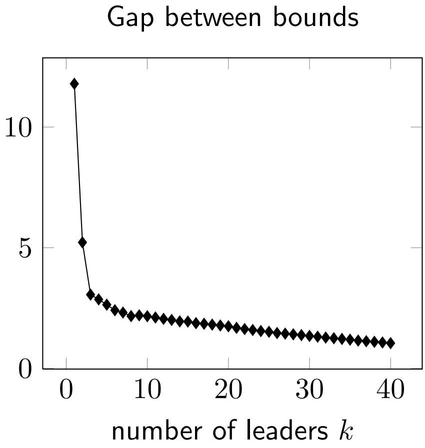


Lower and upper bounds

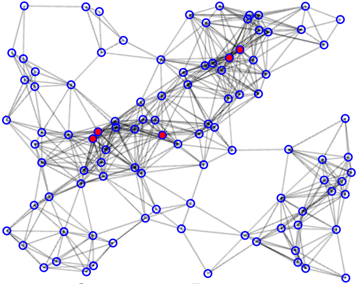


Lower and upper bounds



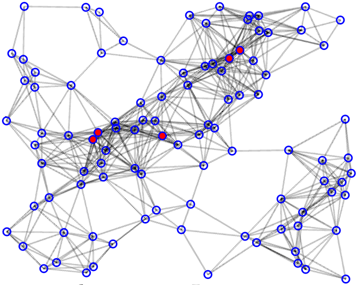


Degree heuristics vs. greedy algorithm

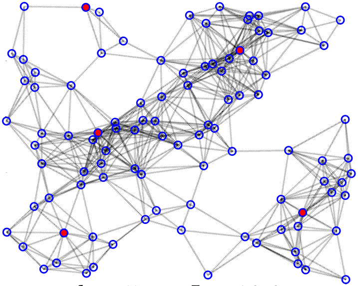


$k = 5 \qquad J = 27.8$

Degree heuristics vs. greedy algorithm

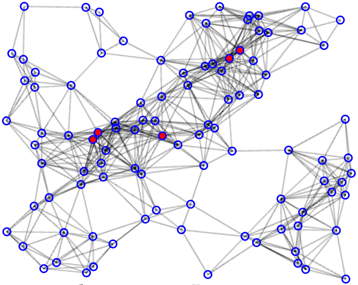


$k = 5 \qquad J = 27.8$

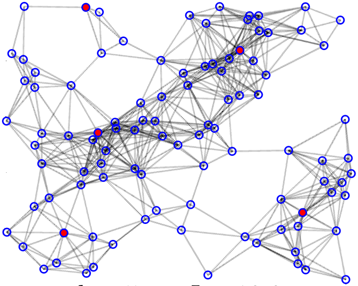


$k = 5 \qquad J = 19.0$

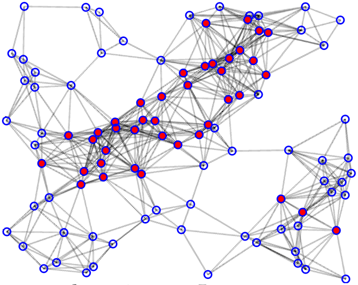
Degree heuristics vs. greedy algorithm



$k = 5 \qquad J = 27.8$

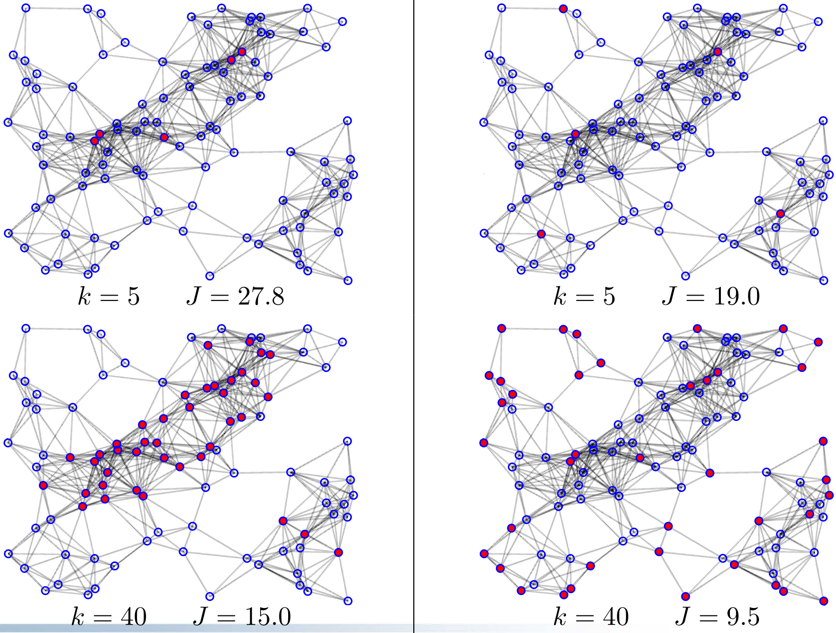


$k = 5 \qquad J = 19.0$

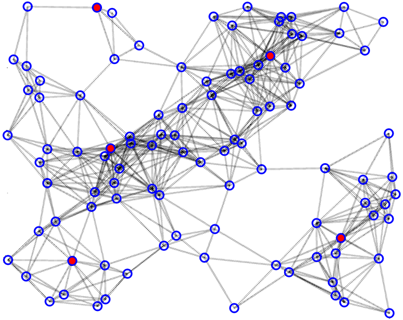


$k = 40 \qquad J = 15.0$

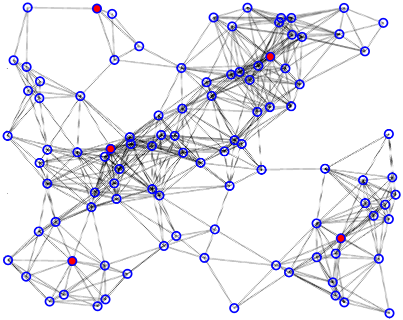
Degree heuristics vs. greedy algorithm



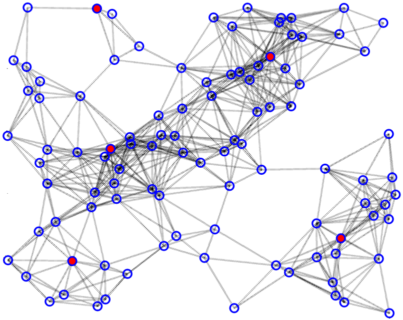
Few leaders vs. many leaders



Few leaders vs. many leaders

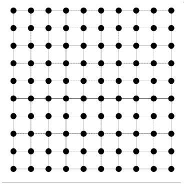


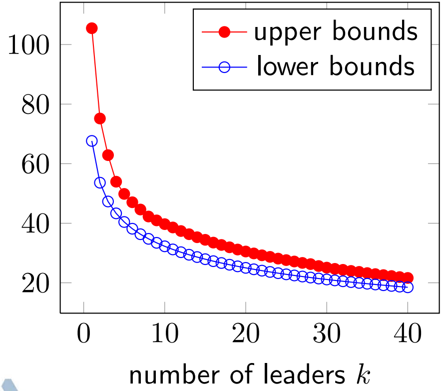
Few leaders vs. many leaders



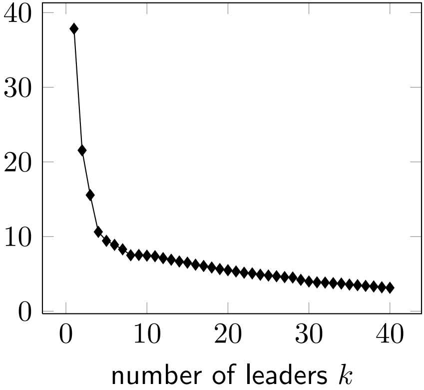
- Few leaders: Partition graphs and spread leaders
- Many leaders: Boundary with low-degree nodes

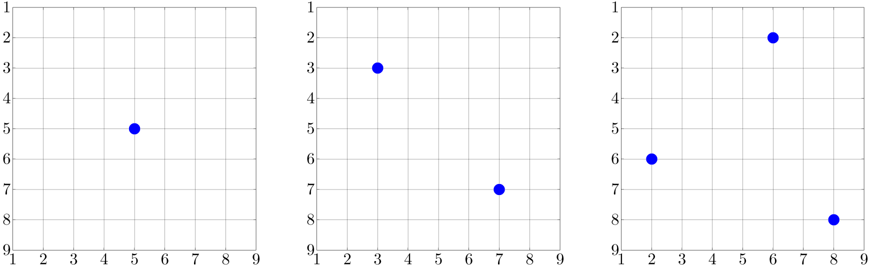
A 2D lattice

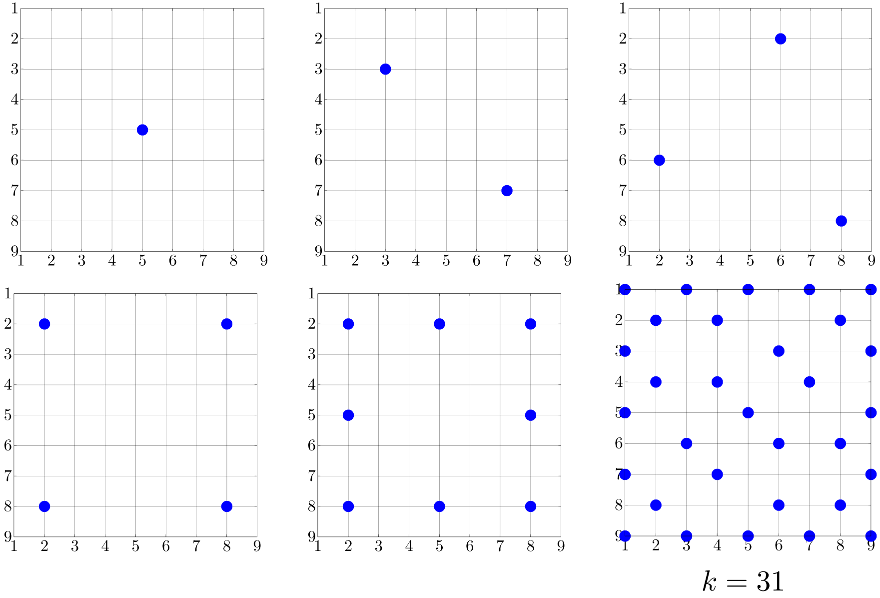




Gap between bounds




1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9



Leaders spread out from center

...So far

- Principal submatrix of graph Laplacian (noise-free leaders)
- Applications in vehicular formations and sensor localization
- Diagonally strengthened Laplacian (noise-corrupted leaders)
- Algorithms for lower and upper bounds on global solutions

Next...

- Alternative formulation for noise-free leader selection
- Algorithms for lower and upper bounds on global solutions
- A flexible framework – amenable to other applications

Alternative formulation

$$J_f(x) \;=\; \operatorname{trace}\left(L_f^{-1}\right) \qquad \textsc{not explicit in} \;\; x$$

$$x_i \;\in\; \{0,1\}, \qquad 1 \;-\; \textsc{leader}, \qquad 0 \;-\; \textsc{follower}$$

Alternative formulation

$$J_f(x) \;=\; \operatorname{trace}\left(L_f^{-1}\right) \qquad \text{NOT EXPLICIT IN } \; x$$

$$x_i \;\in\; \{0,1\}, \qquad 1 \;-\; \text{LEADER}, \qquad 0 \;-\; \text{FOLLOWER}$$

With permutation : $\quad L \;=\; \begin{bmatrix} L_l & L_0^T \\ L_0 & L_f \end{bmatrix}, \qquad x \;=\; \begin{bmatrix} \mathbb{1}_k \\ 0_{n-k} \end{bmatrix}$

Alternative formulation

$$J_f(x) \;=\; \mathrm{trace}\,(L_f^{-1}) \qquad \text{NOT EXPLICIT IN} \;\; x$$

$$x_i \;\in\; \{0,1\}, \qquad 1 \;-\; \text{LEADER}, \qquad 0 \;-\; \text{FOLLOWER}$$

With permutation : $\quad L \;=\; \begin{bmatrix} L_l & L_0^T \\ L_0 & L_f \end{bmatrix}, \qquad x \;=\; \begin{bmatrix} \mathbb{1}_k \\ 0_{n-k} \end{bmatrix}$

$$L \circ ((\mathbb{1} - x)(\mathbb{1} - x)^T) \;=\; \begin{bmatrix} L_l & L_0^T \\ L_0 & L_f \end{bmatrix} \circ \begin{bmatrix} 0 & 0 \\ 0 & \mathbb{1} \end{bmatrix} \;=\; \begin{bmatrix} 0 & 0 \\ 0 & L_f \end{bmatrix}$$

Alternative formulation

$$J_f(x) = \operatorname{trace}(L_f^{-1}) \qquad \text{NOT EXPLICIT IN } x$$

$$x_i \in \{0,1\}, \qquad 1 - \text{LEADER}, \qquad 0 - \text{FOLLOWER}$$

With permutation : $L = \begin{bmatrix} L_l & L_0^T \\ L_0 & L_f \end{bmatrix}, \qquad x = \begin{bmatrix} \mathbb{1}_k \\ 0_{n-k} \end{bmatrix}$

$$L \circ ((\mathbb{1} - x)(\mathbb{1} - x)^T) = \begin{bmatrix} L_l & L_0^T \\ L_0 & L_f \end{bmatrix} \circ \begin{bmatrix} 0 & 0 \\ 0 & \mathbb{1} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & L_f \end{bmatrix}$$

$$[L \circ ((\mathbb{1} - x)(\mathbb{1} - x)^T) + \operatorname{diag}(x)]^{-1} = \begin{bmatrix} I_k & 0 \\ 0 & L_f^{-1} \end{bmatrix}$$

Alternative formulation

$$J_f(x) \;=\; \mathrm{trace}\,(L_f^{-1}) \qquad \text{NOT EXPLICIT IN} \;\; x$$

$$x_i \;\in\; \{0,1\}, \qquad 1 \;-\; \text{LEADER}, \qquad 0 \;-\; \text{FOLLOWER}$$

With permutation : $\quad L \;=\; \begin{bmatrix} L_l & L_0^T \\ L_0 & L_f \end{bmatrix}, \qquad x \;=\; \begin{bmatrix} \mathbb{1}_k \\ 0_{n-k} \end{bmatrix}$

$$L \circ ((\mathbb{1} - x)(\mathbb{1} - x)^T) \;=\; \begin{bmatrix} L_l & L_0^T \\ L_0 & L_f \end{bmatrix} \circ \begin{bmatrix} 0 & 0 \\ 0 & \mathbb{1} \end{bmatrix} \;=\; \begin{bmatrix} 0 & 0 \\ 0 & L_f \end{bmatrix}$$

$$[L \circ ((\mathbb{1} - x)(\mathbb{1} - x)^T) \;+\; \mathrm{diag}\,(x)]^{-1} \;=\; \begin{bmatrix} I_k & 0 \\ 0 & L_f^{-1} \end{bmatrix}$$

$$J_f(x) \;=\; \mathrm{trace}\left([L \circ ((\mathbb{1} - x)(\mathbb{1} - x)^T) \;+\; \mathrm{diag}\,(x)]^{-1}\right) \;-\; k$$

$$J_f(x) \;=\; \mathrm{trace}\left([L \circ ((\mathbb{1} - x)(\mathbb{1} - x)^T) \;+\; \mathrm{diag}\,(x)]^{-1}\right) \;-\; k$$

$$y \;=\; \mathbb{1} \;-\; x$$

$$J_f(x) \;=\; \operatorname{trace}\left([L \circ ((\mathbb{1} - x)(\mathbb{1} - x)^T) \;+\; \operatorname{diag}(x)]^{-1}\right) \;-\; k$$

$$y \;=\; \mathbb{1} \;-\; x$$

$$\begin{array}{lrcl} \underset{y}{\text{minimize}} & J_f(y) & = & \operatorname{trace}\left([L \circ yy^T \;+\; \operatorname{diag}(\mathbb{1} \;-\; y)]^{-1}\right) \;-\; k \\ \text{subject to} & y_i & \in & \{0, 1\}, \qquad i \;=\; 1, \ldots, n \\ & \mathbb{1}^T y & = & n - k \end{array}$$

$$J_f(x) \;=\; \operatorname{trace}\left(\left[L \circ ((\mathbb{1} - x)(\mathbb{1} - x)^T) \;+\; \operatorname{diag}(x)\right]^{-1}\right) \;-\; k$$

$$y \;=\; \mathbb{1} \;-\; x$$

$$\begin{array}{ll} \underset{y}{\text{minimize}} & J_f(y) \;=\; \operatorname{trace}\left(\left[L \circ yy^T \;+\; \operatorname{diag}(\mathbb{1} \;-\; y)\right]^{-1}\right) \;-\; k \\ \text{subject to} & y_i \;\in\; \{0,1\}, \qquad i \;=\; 1,\ldots,n \\ & \mathbb{1}^T y \;=\; n - k \end{array}$$

$$\begin{array}{ll} \underset{Y,\,y}{\text{minimize}} & J_f(Y,y) \;=\; \operatorname{trace}\left(\left[L \circ Y \;+\; \operatorname{diag}(\mathbb{1} \;-\; y)\right]^{-1}\right) \;-\; k \\ \text{subject to} & Y \;=\; yy^T \\ & y_i \;\in\; \{0,1\}, \qquad i \;=\; 1,\ldots,n \\ & \mathbb{1}^T y \;=\; n - k \end{array}$$

$$
\begin{array}{llcll}
\underset{Y,\, y}{\text{minimize}} & J_f(Y, y) & = & \operatorname{trace}\left((L \circ Y \ + \ \operatorname{diag}(\mathbb{1} - y))^{-1} \right) \ - \ k & \\
\text{subject to} & Y & = & yy^T & \\
 & y_i & \in & \{0, 1\}, & i \ = \ 1, \ldots, n \\
 & Y_{ij} & \in & \{0, 1\}, & i, j \ = \ 1, \ldots, n \\
 & \mathbb{1}^T y & = & n - k & \\
 & \mathbb{1}^T Y \mathbb{1} & = & (n - k)^2 &
\end{array}
$$

$$
\begin{array}{lrcl}
\underset{Y,\, y}{\text{minimize}} & J_f(Y, y) & = & \operatorname{trace}\left((L \circ Y + \operatorname{diag}(\mathbb{1} - y))^{-1} \right) - k \\
\text{subject to} & Y & = & y y^T \\
 & y_i & \in & \{0, 1\}, \qquad i = 1, \ldots, n \\
 & Y_{ij} & \in & \{0, 1\}, \qquad i, j = 1, \ldots, n \\
 & \mathbb{1}^T y & = & n - k \\
 & \mathbb{1}^T Y \mathbb{1} & = & (n - k)^2
\end{array}
$$

$$
Y = y y^T \quad \Longleftrightarrow \quad \{\, Y \succeq 0, \ \mathbf{rank}\,(Y) = 1 \,\}
$$

$$\begin{array}{lrcl}
\underset{Y,\,y}{\text{minimize}} & J_f(Y,y) & = & \operatorname{trace}\left((L \circ Y + \operatorname{diag}(\mathbb{1} - y))^{-1}\right) - k \\
\text{subject to} & Y & = & yy^T \\
 & y_i & \in & \{0,1\}, \qquad i = 1,\ldots,n \\
 & Y_{ij} & \in & \{0,1\}, \qquad i,j = 1,\ldots,n \\
 & \mathbb{1}^T y & = & n - k \\
 & \mathbb{1}^T Y \mathbb{1} & = & (n-k)^2
\end{array}$$

$$Y = yy^T \quad \Longleftrightarrow \quad \{\, Y \succeq 0,\ \mathbf{rank}\,(Y) = 1 \,\}$$

Drop rank constraint $+$ relax Boolean constraints

$$\Downarrow$$

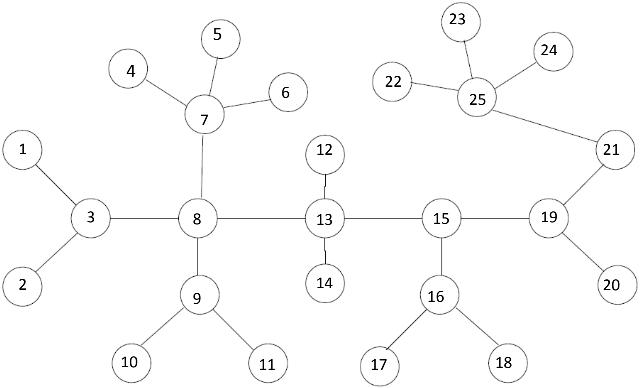
convex relaxation

Convex relaxation

$$
\begin{array}{lrcl}
\underset{Y,\,y}{\text{minimize}} & J_f(Y,y) & = & \operatorname{trace}\left(\left(L \circ Y + \operatorname{diag}(\mathbb{1} - y)\right)^{-1}\right) - k \\
\text{subject to} & Y & \succeq & 0 \\
 & y_i & \in & [0,1], \qquad i = 1,\ldots,n \\
 & Y_{ij} & \in & [0,1], \qquad i,j = 1,\ldots,n \\
 & \mathbb{1}^T y & = & n-k \\
 & \mathbb{1}^T Y \mathbb{1} & = & (n-k)^2
\end{array}
$$

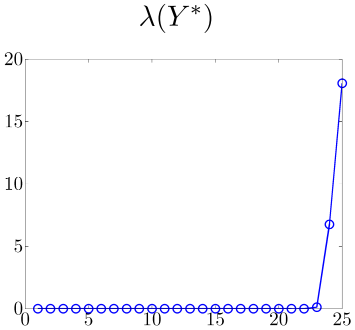
How important is rank-1 constraint?

How important is rank-1 constraint?



- A tree network with 25 nodes
- Select $k = 5$ noise-free leaders

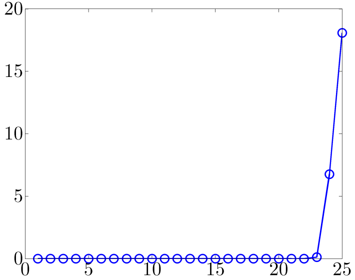
Solution from convex relaxation



Solution Y^* is low-rank

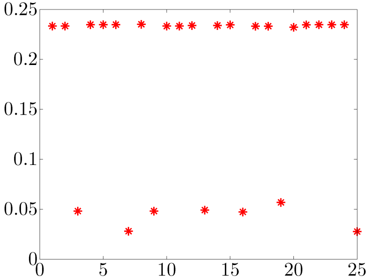
Solution from convex relaxation

$\lambda(Y^*)$



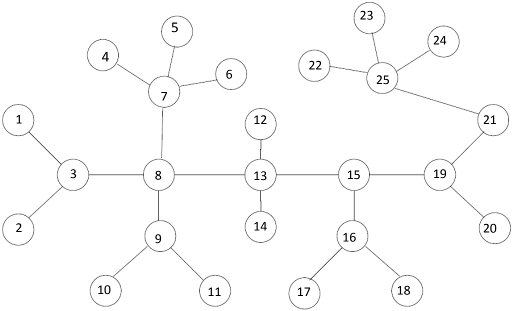
Solution Y^* is low-rank

$v_{\max}$ with $\lambda_{\max}(Y^*)$



clear separation in magnitude

Leader selection based on $v_{\max}$



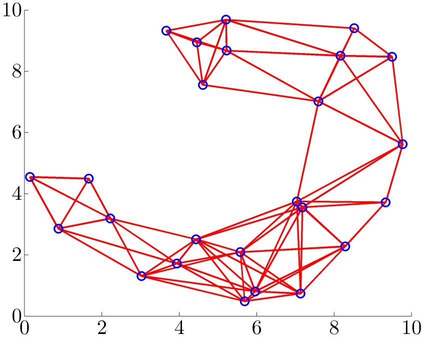
	global solution		convex relaxation	
k	J_f	leaders	J_f	leaders
1	66.0	13	112.0	25
2	38.4	8, 25	43.3	7, 25
3	30.0	8, 16, 25	32.1	7, 16, 25
4	25.3	7, 9, 16, 25	25.3	7, 9, 16, 25
5	20.7	3, 7, 9, 16, 25	20.7	3, 7, 9, 16, 25

Can we say something in general?

Can we say something in general?

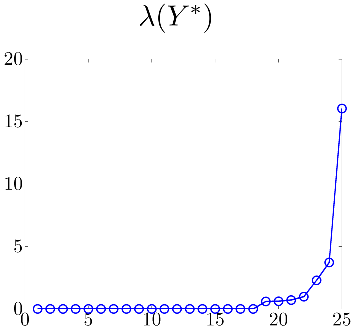
Not really ... ;-(

A "bad" example



- A random network with 25 nodes
- Select $k = 5$ noise-free leaders

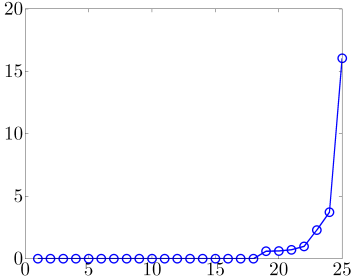
Solution from convex relaxation



Y^* is still approximately low-rank

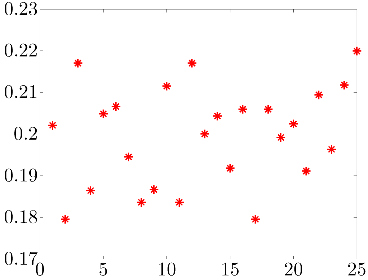
Solution from convex relaxation

$\lambda(Y^*)$



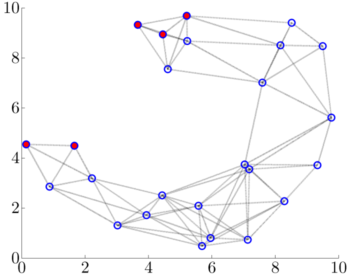
Y^* is still approximately low-rank

$v_{\max}$ with $\lambda_{\max}(Y^*)$

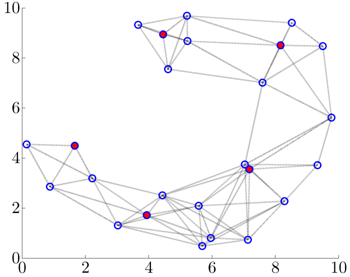


No clear-cut separation

Leader selection

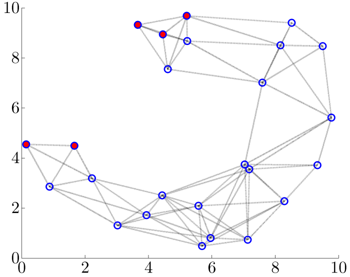


Based on magnitude of v_{max}

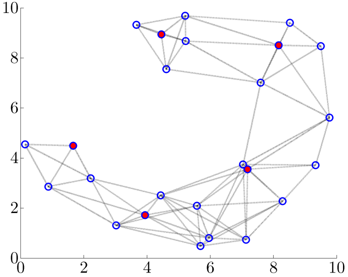


Globally optimal solution (exhaustive search)

Leader selection



Based on magnitude of $v_{\max}$



Globally optimal solution
(exhaustive search)

Back to greedy algorithm

Greedy algorithm

- One-leader-at-a-time
  - RANK-2 UPDATE: $O(n^2)$ per leader

Greedy algorithm

- One-leader-at-a-time
  - RANK-2 UPDATE: $O(n^2)$ per leader

Without exploiting structure $O(n^4 k)$

Low-rank updates $O(n^3 k)$

Greedy algorithm

- One-leader-at-a-time
  - RANK-2 UPDATE: $O(n^2)$ per leader

  Without exploiting structure $O(n^4 k)$

  Low-rank updates $O(n^3 k)$

- Swap a leader and a follower
  - RANK-2 UPDATE: $O(n^2)$ per swap

Convex relaxation

$$
\begin{array}{lrcl}
\underset{Y,\, y}{\text{minimize}} & J_f(Y, y) & = & \mathrm{trace}\left((L \circ Y \,+\, \mathrm{diag}\,(\mathbb{1} - y))^{-1} \right) \,-\, k \\
\text{subject to} & Y & \succeq & 0 \\
& Y_{ij} & \in & [0,1], \qquad i, j \,=\, 1, \ldots, n \\
& \mathbb{1}^T Y \mathbb{1} & = & (n-k)^2 \\
& y_i & \in & [0,1], \qquad i \,=\, 1, \ldots, n \\
& \mathbb{1}^T y & = & n - k
\end{array}
$$

Convex relaxation

$$\begin{array}{lrcl} \underset{Y,\, y}{\text{minimize}} & J_f(Y,y) & = & \operatorname{trace}\left((L \circ Y \;+\; \operatorname{diag}(\mathbb{1} - y))^{-1}\right) \;-\; k \\ \text{subject to} & Y & \succeq & 0 \\ & Y_{ij} & \in & [0,1], \qquad i,j \;=\; 1,\ldots,n \\ & \mathbb{1}^T Y \mathbb{1} & = & (n-k)^2 \\ & y_i & \in & [0,1], \qquad i \;=\; 1,\ldots,n \\ & \mathbb{1}^T y & = & n-k \end{array}$$

- SDP solvers $\quad O(n^6)$
- Exploit problem structure:

positive semidefinite cone $\;+\;$ simplex sets

Alternating direction method of multipliers (ADMM)

- First-order method – not for high accuracy

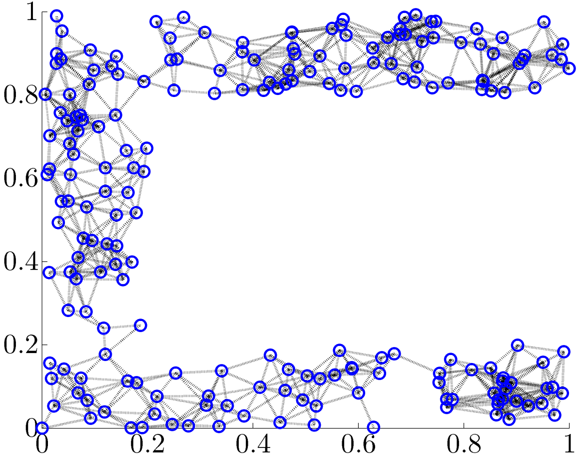
- Solve a sequence of subproblems

- Optimization over positive semidefinite cone and simplex

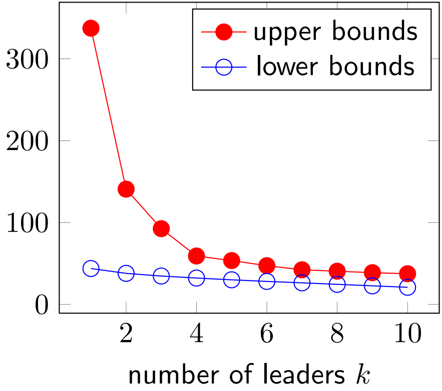
- Each subproblem costs $O(n^3)$

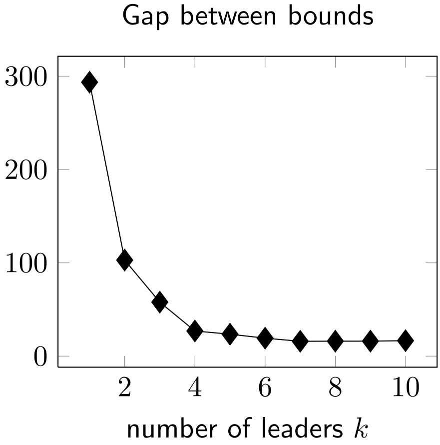
An example

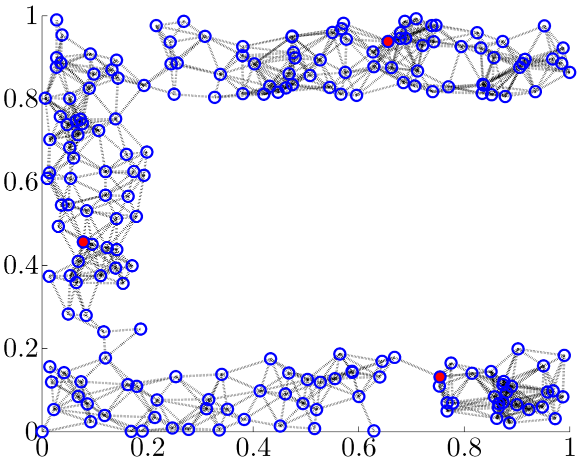
200 randomly distributed nodes in a C-shaped region



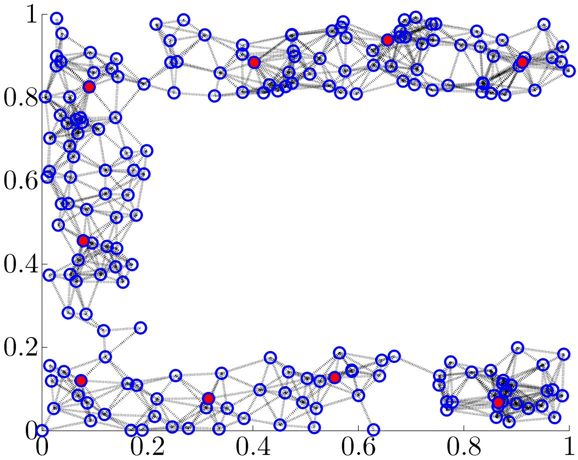
Srirangarajan, Tewfik, and Luo IEEE TSP '08


upper bounds
lower bounds
300
200
100
0
2
4
6
8
10
number of leaders k

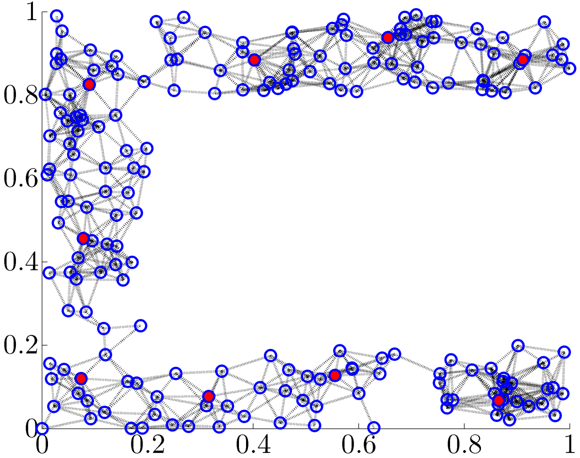

Gap between bounds
300
200
100
0
2
4
6
8
10
number of leaders k



$k = 3$



$k = 9$



$$k = 9$$

Both noise-free and noise-corrupted formulations

yield similar selection of leaders

More details

Papers:

- Lin, Fardad, and Jovanović, IEEE CDC '11
- Lin, Fardad, and Jovanović, IEEE TAC '13 (accepted)

Matlab implementation:

www.mcs.anl.gov/∼fulin/software

Extensions:

- Robustness of leader selection
- Applications to social networks

Questions

- How good are the lower and upper bounds?
- For what graphs, these bounds are reasonably tight?
- Can we use solution from convex relaxation to choose leaders?

Any suggestions are welcome! ;-)