Improving Network Robustness through Edge Augmentation While Preserving Strong Structural Controllability

W. Abbas[1], M. Shabbir[2], H. Jaleel[3], and X. Koutsoukos[1]

[1]*Vanderbilt University, Nashville, TN, US*

[2]*Information Technology University, Lahore, Pakistan*

[3]*Lahore University of Management Sciences, Pakistan*

ACC2020 · DENVER

American Control Conference, 2020

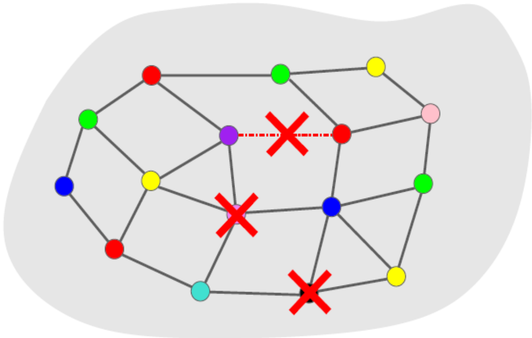
Controllability and Robustness in Networks

Controllability and robustness are crucial attributes of a networked dynamical system.

Network Controllability

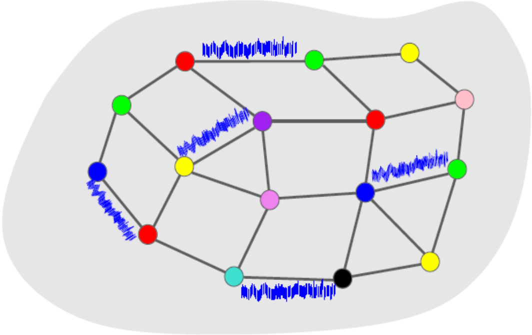
How can we drive a network of agents from some initial state to a final state by controlling only a small subset of agents, referred to as leaders?

Network Robustness



How can we minimize the effect of node/edge removals on the overall network structure?

Structural aspect

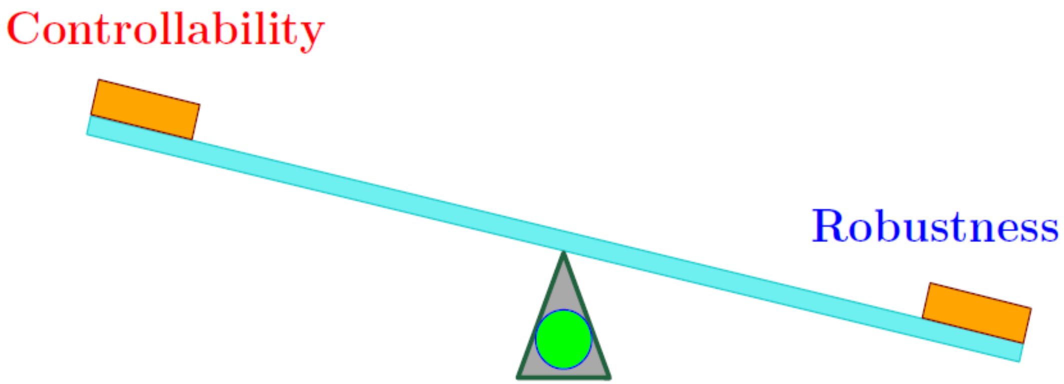


How can we minimize the effect of noisy information on the network's overall performance

Functional aspect

Controllability and Robustness in Networks

Controllability and robustness properties in networks are *conflicting* at times[1,2].



How can we improve one property (for instance, by modifying the network graph) without deteriorating the other property?

[1]F. Pasqualetti, C. Favaretto, S. Zhao, and S. Zampieri, "Fragility and controllability tradeoff in complex networks," ACC 2018.

[2]W. Abbas, M. Shabbir, M. Yazicioğlu and A. Akber, "On the trade-off between controllability and robustness in networks of diffusively coupled agents," ACC 2019.

Network Controllability

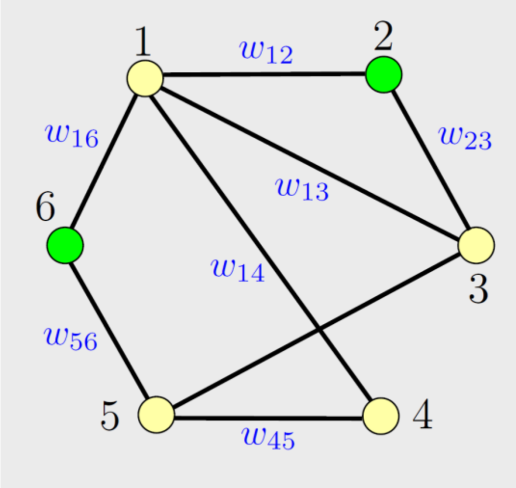
We consider a network of agents with Laplacian dynamics.

$$\dot{x}_i = \sum_{j \in \mathcal{N}_i} w_{ij}(x_j - x_i)$$

Follower
(no external input)

$$\dot{x}_\ell = \sum_{j \in \mathcal{N}_\ell} w_{\ell j}(x_j - x_\ell) + u_\ell$$

Leaders
(external input)



$$\dot{x} = -\mathcal{L}_w x + \mathcal{B} u$$

$\mathcal{L}_w$: Weighted Laplacian

$\mathcal{B}$: Input matrix

$$\mathcal{L}_w = \begin{bmatrix} \sum w_{1,j} & -w_{12} & -w_{13} & -w_{14} & 0 & -w_{16} \\ -w_{12} & \sum w_{2,j} & -w_{23} & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

Network Controllability

Controllability matrix:

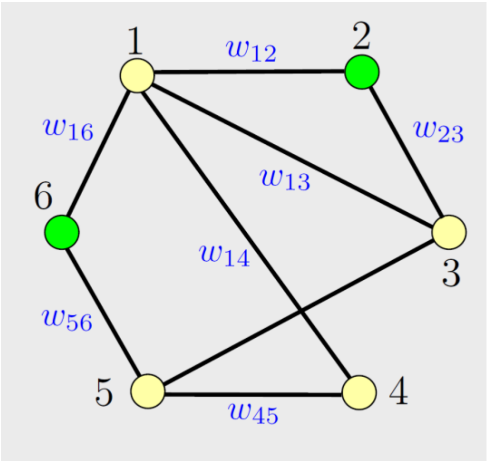
$$\Gamma = \left[\ \mathcal{B} \quad -\mathcal{L}_w\mathcal{B} \quad (-\mathcal{L}_w)^2\mathcal{B} \quad \cdots \quad (-\mathcal{L}_w)^{n-1}\mathcal{B}\ \right]$$

Controllable subspace: **Range (Γ)**

Controllability measure: **Rank (Γ)**

input matrix $\mathcal{B}$

structure of graph
weights of edges $\mathcal{L}_w$



Sometimes weights are unknown due to system uncertainties. So we want a controllability notion that is independent of edge weights.

Strong **S**tructural **C**ontrollability

$$\Gamma = \left[\ \mathcal{B} \quad -\mathcal{L}_w\mathcal{B} \quad \cdots \quad (-\mathcal{L}_w)^{n-1}\mathcal{B}\ \right]$$

$$\min_{\mathbf{w}} \ \mathrm{Rank}(\boldsymbol{\Gamma})$$

Dimension of SSC

(measure of SSC)

$$\mathcal{L} = \begin{bmatrix} \times & \times & \times & \times & 0 & \times \\ \times & \times & \times & 0 & 0 & 0 \\ \times & \times & \times & 0 & \times & 0 \\ \times & 0 & 0 & \times & \times & 0 \\ 0 & 0 & \times & \times & \times & \times \\ \times & 0 & 0 & 0 & \times & \times \end{bmatrix}$$

Network Controllability and Graph Distances

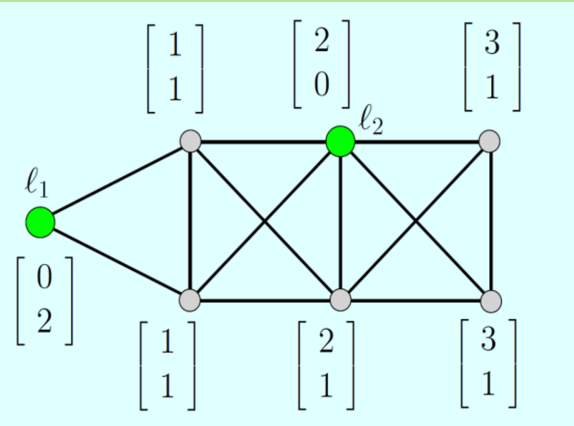
Computing $\min_w \text{Rank}(\Gamma)$ is very challenging (NP-hard in general).

Graph distances between nodes are useful in obtaining a tight lower bound on $\min_w \text{Rank}(\Gamma)$.

Distance-to-leader (DL) vector for a node v:

$$\begin{bmatrix} d(\ell_1, v) & d(\ell_2, v) & \cdots & d(\ell_k, v) \end{bmatrix}^T$$

$\min_w \text{Rank}(\Gamma) \geq$ **Length of a certain *sequence* of DL vectors**



Preserving distances between certain node pairs guarantees a lower bound on the SSC dimension.

A. Y. Yazıcıoğlu, W. Abbas, and M. Egerstedt, "Graph distances and controllability of networks," IEEE TAC, 2016

Network Controllability and Graph Distances

A subset of pair-wise distances between nodes in a graph provides a tight lower bound on the dimension of SSC.

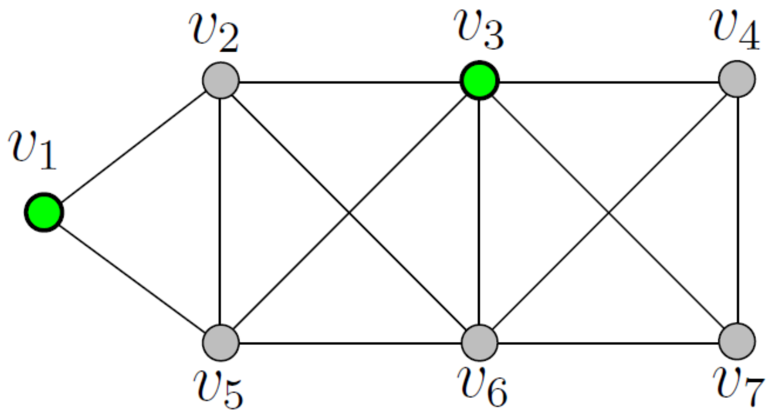
Preserving these distances guarantees a lower bound on the SSC dimension.

Example:

Preserving distances between nodes in the following node pairs ensures that the dimension of SSC is at least 5.

(v_1, v_2) (v_1, v_3) (v_1, v_6) (v_1, v_7)

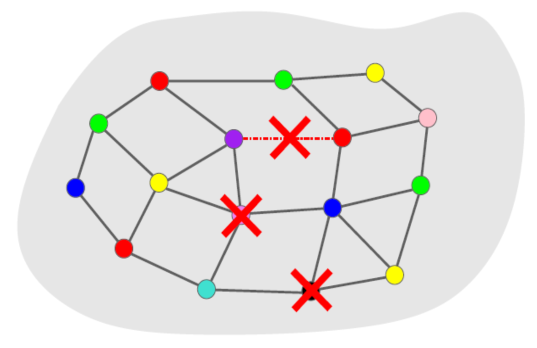
(v_3, v_2) (v_3, v_6) (v_3, v_7)



Network Robustness

Kirchhoff index **(K_f)** of a graph is widely used[1,2] to measure network's robustness to node/link failures and to noise.

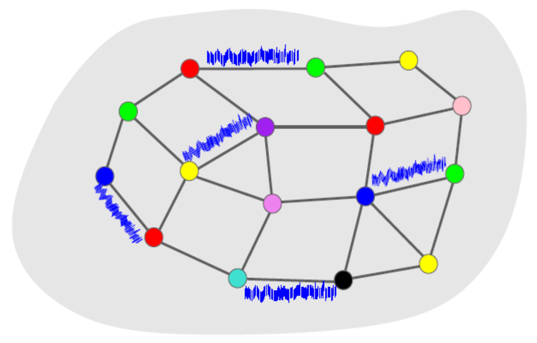
In fact, network robustness, as measured by K_f, increases monotonically with edge additions.



Structural robustness

However, adding edges could also deteriorate network's controllability.

How can we maximally add edges in a network to improve robustness while preserving its SSC?



Functional robustness

[1]W. Ellens, et al. "Effective graph resistance," *Linear Algebra and its Applications* (2011)

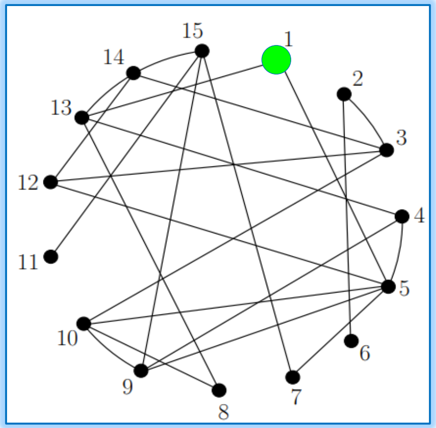
[2]G. F. Young, L. Scardovi, and N. E. Leonard, "Robustness of noisy consensus dynamics with directed communication," ACC 2010.

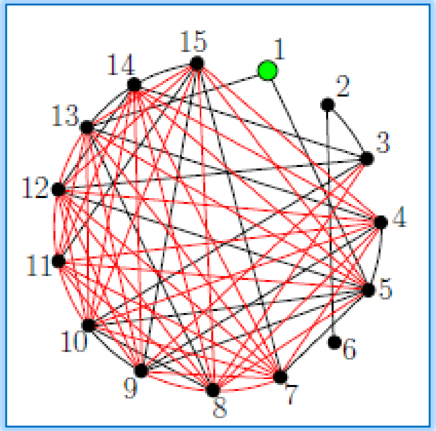
Improving Network Robustness while Preserving Controllability

Approach:

Add edges while preserving a SSC controllability bound.

Add edges while preserving distances between leaders and 'some' other nodes.





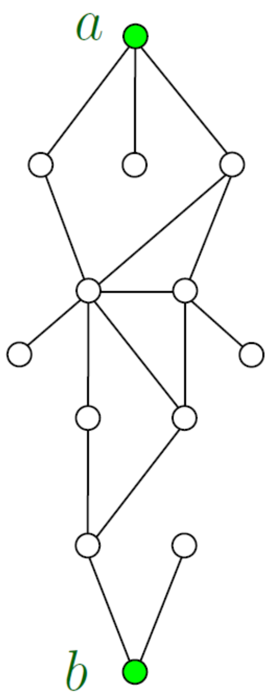
Basic problem:

Given a node pair (a, b), add maximum edges while preserving a distance between those two nodes

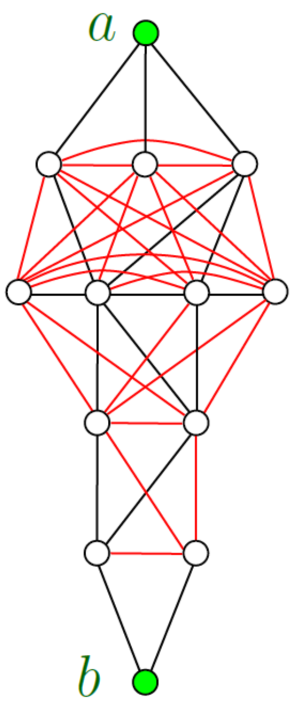
Distance Preserving Edge Augmentation (DPEA)

Given G = (V, E), and two nodes a, $b \in$ V such that $d_G(a, b) = k$.

Add maximum no. of edges in G while preserving the distance between a and b.



Adding edges



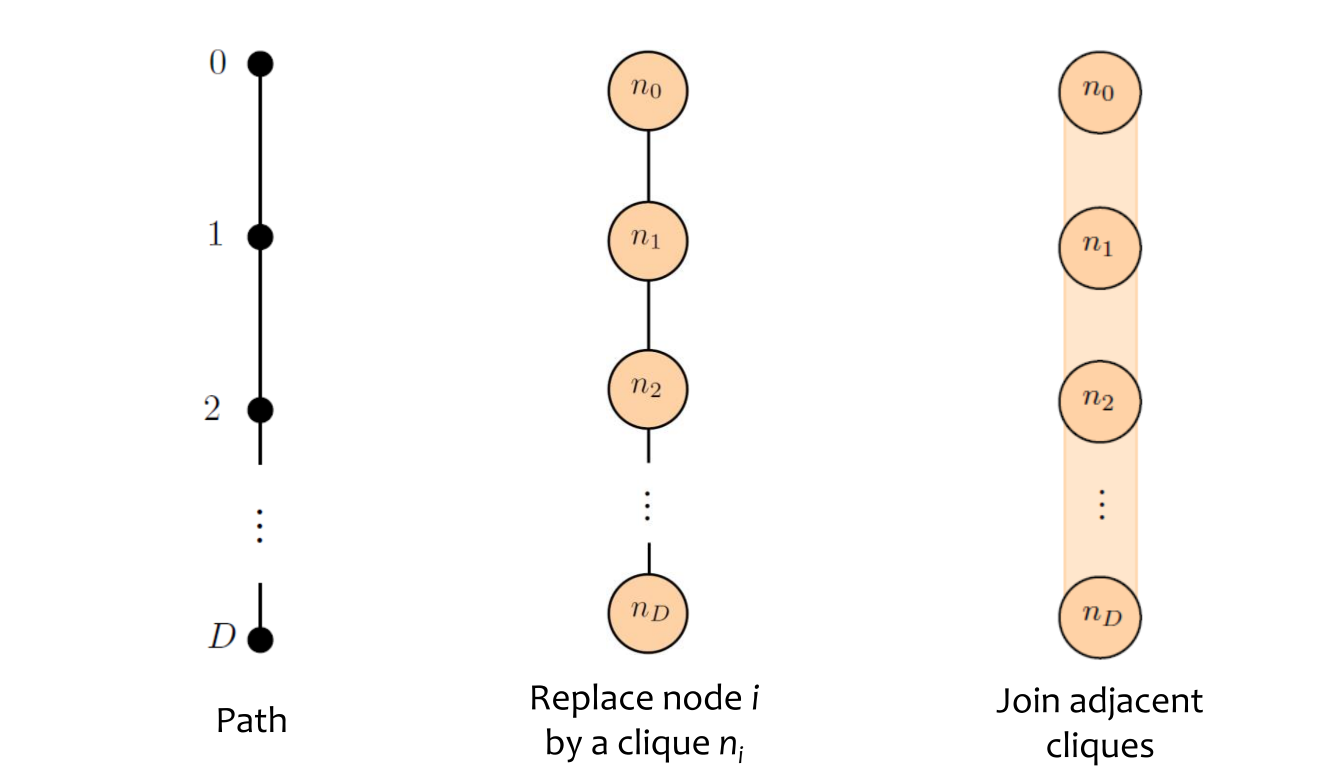
$d_G(a, b) = 5$

$d_{G'}(a, b) = 5$

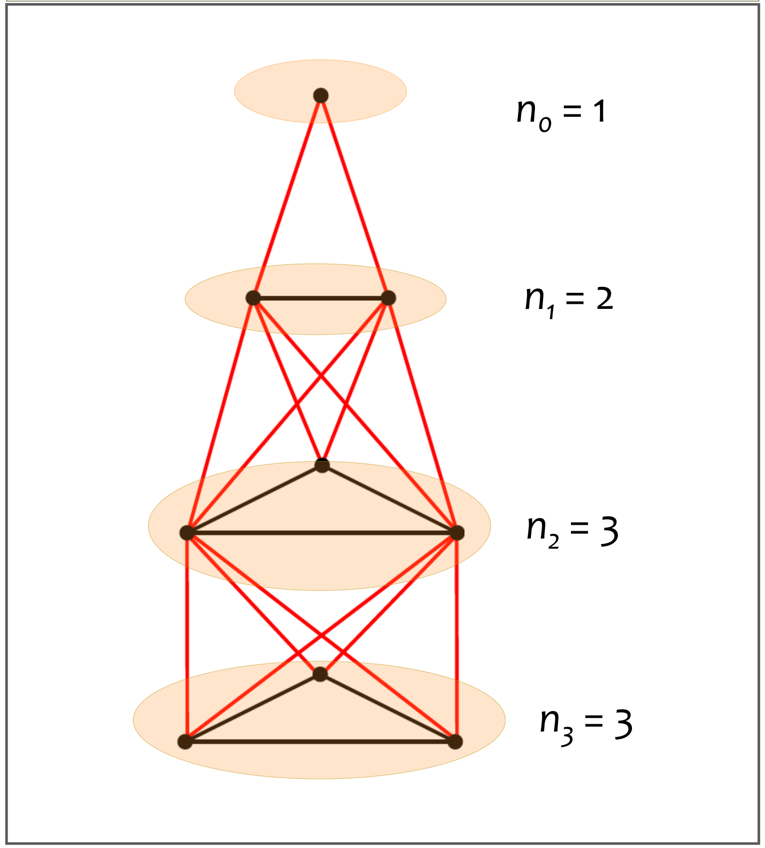
Clique Chains

Optimal solution of the DPEA problem is related to a special class of graphs known as **clique chains**.

Clique chain: $G_D(n_0, n_1, \ldots, n_D)$
Diameter: D
No. of nodes: $N = \sum_{i=0}^{D} n_i$



$G_3(1, 2, 3, 3)$



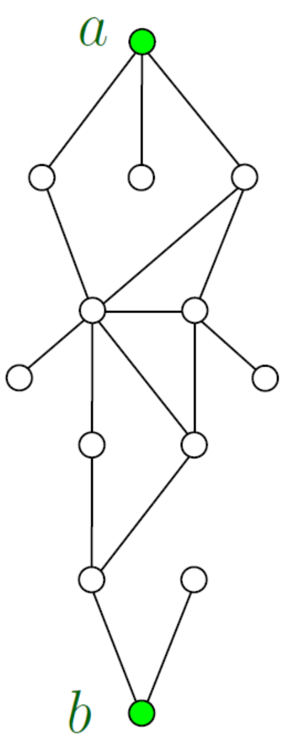
$D = 3$
$N = 9$

DPEA and Clique Chains

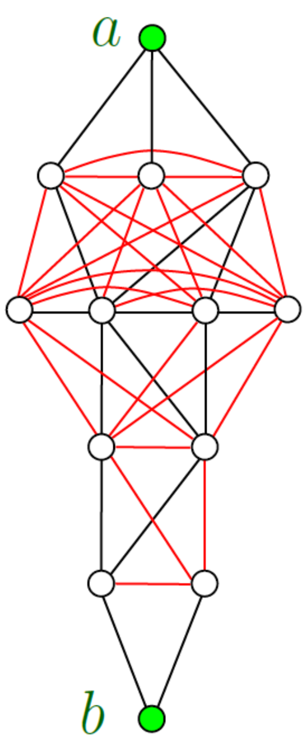
Theorem: For a given G = (V, E), and nodes a, $b \in$ V where $d_G(a, b) = k > 1$, optimal solution to the DPEA problem is a clique chain of the form

$$G_k(n_0 = 1,\ n_1, \dots,\ n_k = 1).$$

d(a, b) = 5



G



G_k(1, 3, 4 , 2, 2, 1)

We provide a method to construct such clique chains.

d(a, b) = 5

Clique Chain Construction for DPEA

Given: $G = (V, E)$, $a, b \in V$, $d_G(a, b) = k$

Construct: Clique chain $G_k(n_0 = 1, n_1, \dots, n_k = 1)$ solving DPEA.

$S_i^a = \{v \in V \mid d_G(a, v) = i\}$

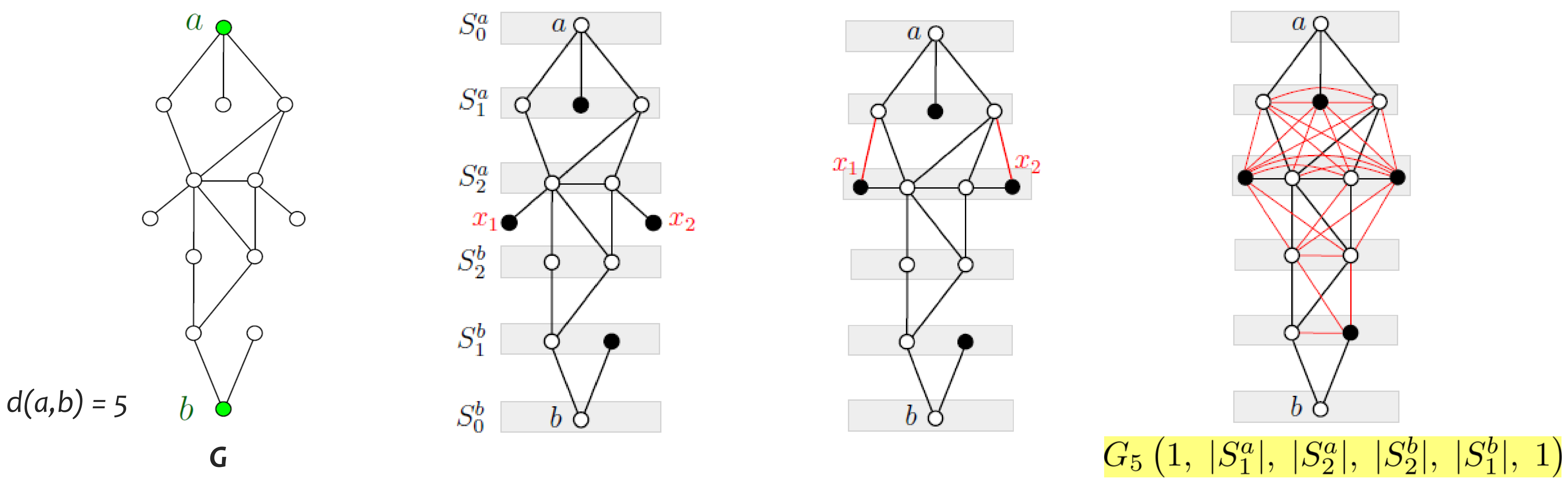
$S_i^b = \{v \in V \mid d_G(b, v) = i\}$

Fixed: nodes included in some ***shortest path*** between *a* and *b*.

Free: remaining nodes.

Every fixed node lies in a unique S_i^a (S_i^b).

Free nodes can be placed in appropriate S_i^a or S_i^b by creating edges.



Edge Augmentation to Preserve SSC Controllability Bound

First Approach:

Add edges while preserving a SSC bound.

↓

Add edges while preserving distances between leaders and 'some' nodes.

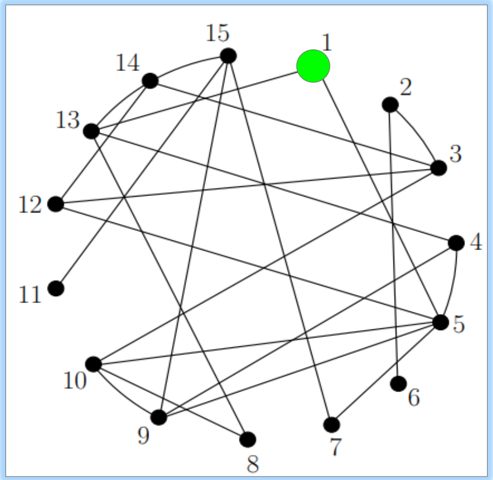
↓

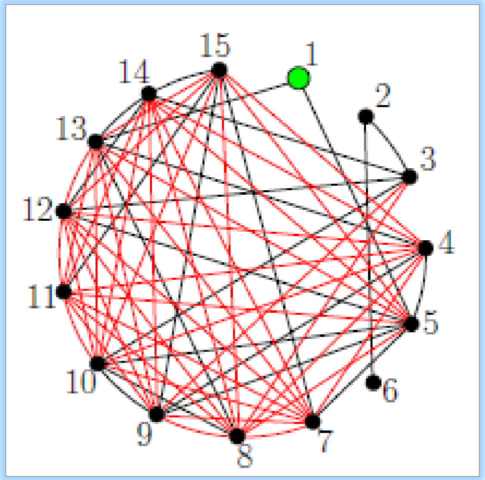
Solve multiple instances of DPEA problems.

↓

Obtain edges that are common in solutions of all DPEA instances.

(Intersection)





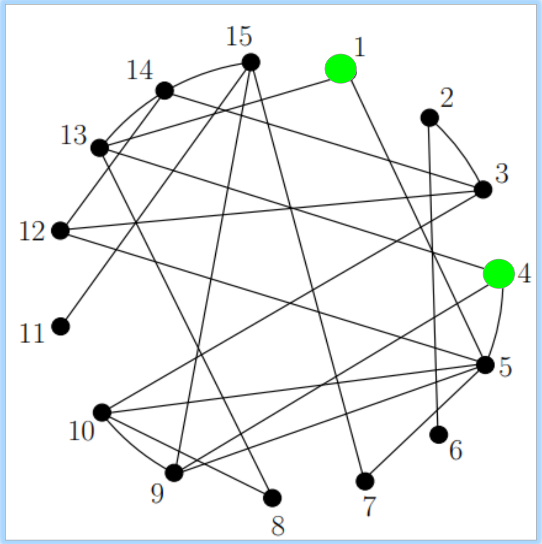
Node pairs:

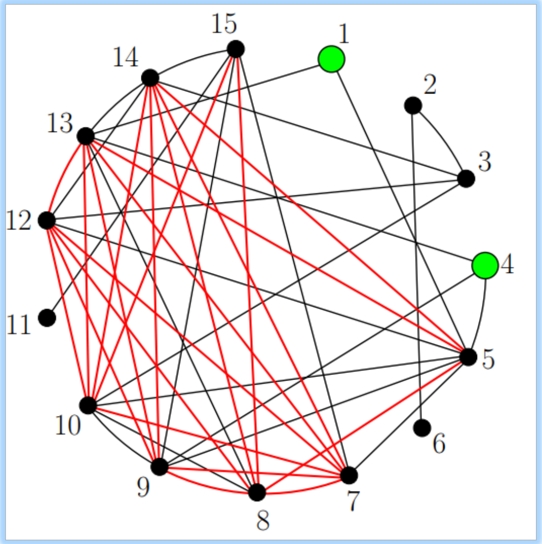
(v_1, v_5) (v_1, v_4) (v_1, v_3) (v_1, v_2) (v_1, v_6)

Solve DPEA for each node pair

Edge Augmentation to Preserve Controllability

First Approach (Intersection)





(v_1, v_4) (v_1, v_5) (v_1, v_9) (v_1, v_7) (v_1, v_{15}) (v_1, v_3) (v_1, v_{11}) (v_1, v_2) (v_1, v_6)

(v_4, v_5) (v_4, v_9) (v_4, v_7) (v_4, v_{15}) (v_4, v_3) (v_4, v_{11}) (v_4, v_2) (v_4, v_6)

Solve DPEA for each node pair and then take common (intersecting) edges

Edge Augmentation to Preserve Controllability

Second Approach (Randomized Algorithm)

Basic idea remains the same:

Add edges while preserving a SSC bound.

↓

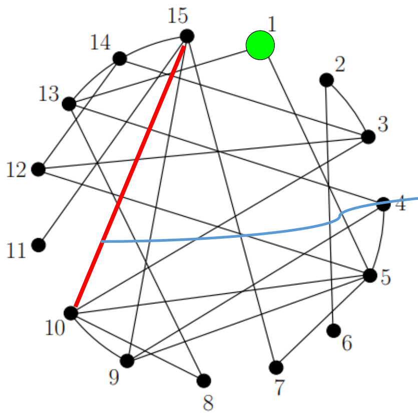
Add edges while preserving distances between leaders and *'some'* nodes.

Obtain all missing edges ***E'*** .

Randomly select a missing edge ***e*** ∈ ***E'***.

If adding ***e*** does not change distances between desired node pairs, then keep it. Otherwise, discard it.

Repeat until no more missing edge is left



Does not change any desired distance, **so keep it.**

Node pairs:

(v_1, v_5) (v_1, v_4) (v_1, v_3) (v_1, v_2) (v_1, v_6)

Edge Augmentation to Preserve Controllability

Second Approach (Randomized Algorithm)

Basic idea remains the same:

Add edges while preserving a SSC bound.

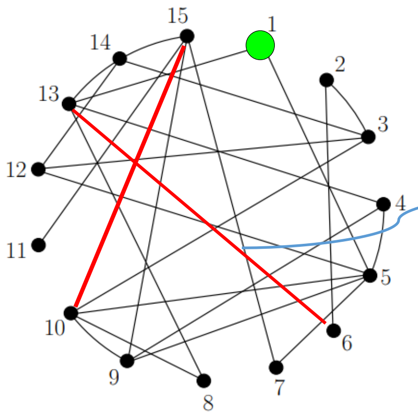
Add edges while preserving distances between leaders and *'some'* nodes.

Obtain all missing edges ***E'*** .

Randomly select a missing edge ***e*** ∈ ***E'***.

If adding ***e*** does not change distances between desired node pairs, then keep it. Otherwise, discard it.

Repeat until no more missing edge is left



Changes the distance between v_1 and v_6, **so discard it.**

Node pairs:

(v_1, v_5) (v_1, v_4) (v_1, v_3) (v_1, v_2) (v_1, v_6)

Edge Augmentation to Preserve Controllability

Proposition: The randomized algorithm returns an α-approximate solution with probability at least $1 - e^{-c\left(\frac{t}{T}\right)^{\alpha t}}$, when repeated *c* times.

Here,

- $T \leq E'$ is the number of edges that are (individually) legal to add to the input graph.
- $t \leq T$ is the size of an (unknown) optimal solution.

Example:

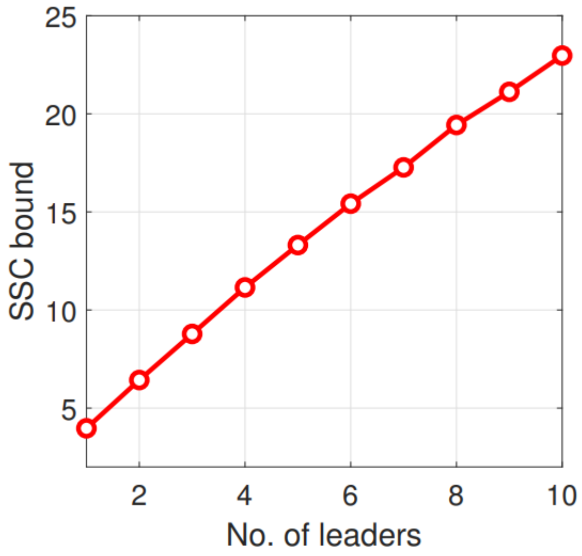
If $T = 100$ and $t = 0.92\ T$, then repeating the randomized algorithm $c = 500$ times gives a (3/4) - approximate solution with probability at least 0.8.

Simulation Results

Erdos – Renyi (**ER**) Random Graphs $G(N,p)$:

$N = 50, \quad p = 0.2$

(Each point is an average of 100 randomly generated instances.)



Lower bound on the dimension of SSC as a function of no. of leaders.

A comparison of Intersection (algo 1) and Randomized (algo 2) algorithms to add edges.

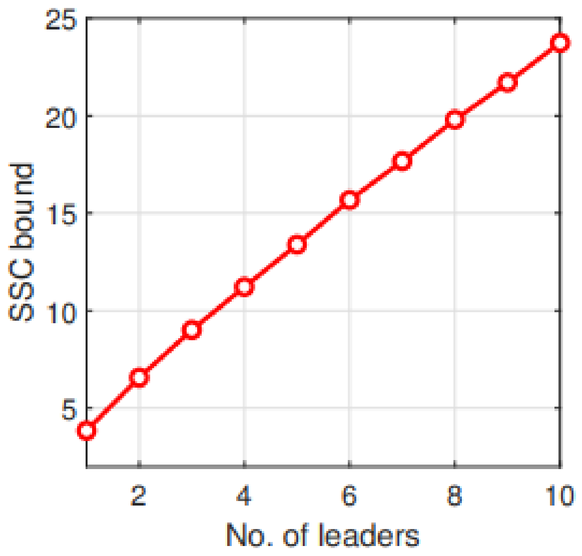
(Randomized algorithm is repeated c = 150 times.)

Simulation Results

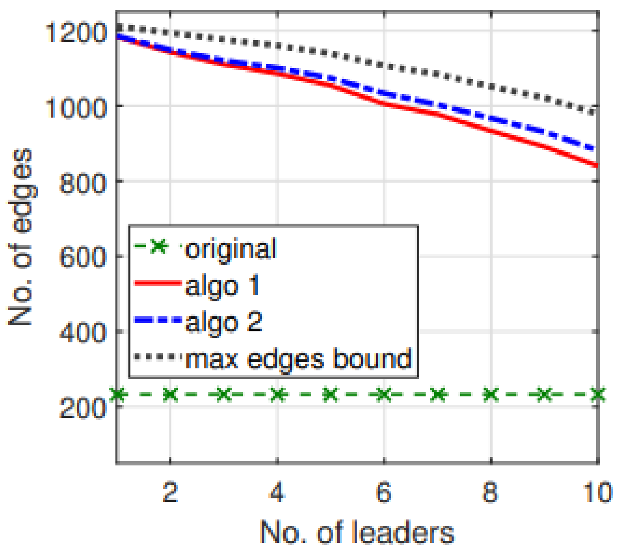
Barabasi – Albert (**BA**) Random Graphs G(N, γ):

N = 50, γ = 5

(Each point is an average of 100 randomly generated instances.)



Lower bound on the dimension of SSC as a function of no. of leaders.



A comparison of Intersection (algo 1) and Randomized (algo 2) algorithms to add edges.

(Randomized algorithm is repeated c = 150 times.)

Summary & Conclusions

Add edges to improve robustness while preserving SSC

Add edges while preserving distances between certain nodes

DPEA problem

Intersection algorithm

Randomized algorithm

Thank You

(waseem.abbas@vanderbilt.edu)