IBM Thomas J. Watson Research Center

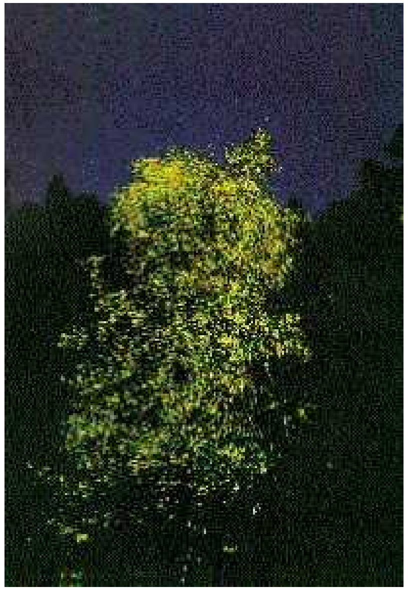
Control and synchronization in systems coupled via a complex network

Chai Wah Wu

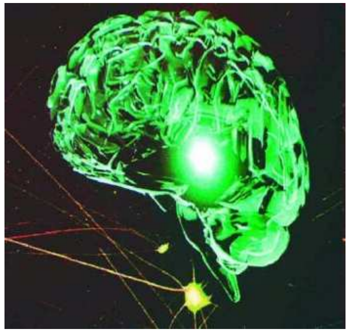
May 29, 2009

City University of Hong Kong

Synchronization in nonlinear dynamical systems



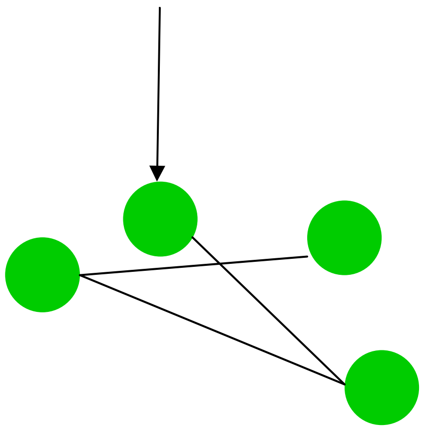
- Synchronization in groups of nonlinear dynamical systems is an active research topic in engineering, biology and systems science.
- Applications:
  - Synchronous firing in groups of fireflies
  - Mobile autonomous robots with limited range of communications
  - Flocking behavior in animals
  - Epileptic behavior



- Coupled system is autonomous, no outside influence to facilitate synchronization

Synchronization via external control

- Synchronization behavior has also been studied where the synchronization is driven by external forcing (Wang and Chen, 2002).
- Want all systems to synchronize to a desired trajectory x(t). Apply x(t) to a subset of systems. What are the conditions under which the entire systems synchronize to x(t)?



State equations

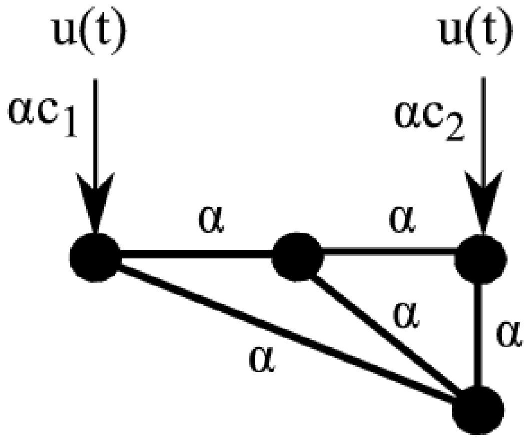
- Unforced network with linear coupling:

$$\frac{dx_i}{dt} = f(x_i, t) - \alpha \sum_j G_{ij} D(t) x_j$$

- Linear control applied to network:

$$\frac{dx_i}{dt} = f(x_i, t) - \alpha \left(\sum_j G_{ij} D(t) x_j + c_i D(t)(x_i - u(t)) \right)$$

- u(t) is the desired trajectory.
- Two sets of parameters describing the coupling: α describes the coupling between *all* systems, c_i describes the control coupling.



Virtual system

- If u(t) is the trajectory of an uncoupled system, i.e. u'(t) = f(u(t),t), then by setting $x_{n+1}(t)$ = u(t) we get a network of n+1 systems:

$$\frac{dx_i}{dt} = f(x_i, t) - \alpha \sum_j G'_{ij} D(t) x_j$$

- With the new coupling matrix related to G as:

$$G' = \begin{pmatrix} G + C & -c \\ 0 & 0 \end{pmatrix}$$

- where C = diag $(c_1,\ldots,c_n)$ and c = $(c_1,\ldots,c_n)^T$.
- Synchronizing control is then reduced to a problem in synchronization.

Synchronization

- Various synchronization theorems relate criteria for synchronization with properties of the matrix G'.
- For a Lyapunov function based approach, we will used the quantity β_{min}: Under suitable conditions, control is achieved if

$$\beta_{\min} \geq \frac{1}{\alpha}$$

- Precise definition of β_{min} can be found in the literature. β_{min} is related to

$$\min_{\lambda} \Re(\lambda(G+C))$$

- In fact, for vertex-balanced networks:

$$\beta_{\min} = \min_{\lambda} \Re(\lambda(G+C)) = \lambda_{\min}\left(\frac{1}{2}(G+G^T)+C\right)$$

Synchronization condition

- Recall the synchronization condition:

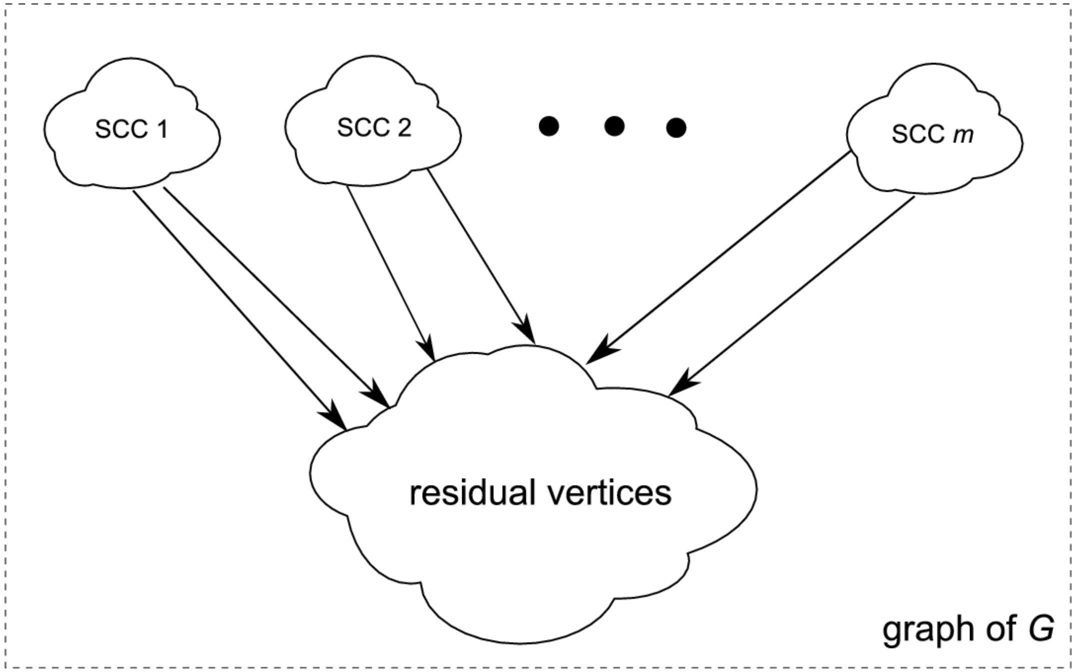
$$\beta_{\min} \geq \frac{1}{\alpha}$$

- If $\beta_{\min} > 0$, then control is achievable for large enough α.
- Theorem: $\beta_{\min} > 0$ if and only if there exists a spanning directed forest in the graph of G such that $c_i > 0$ whenever the i^{th} system is a root of a tree in the forest.

- Thus control can be achieved if and only if forcing is applied to roots of trees in a spanning directed forest of the network.

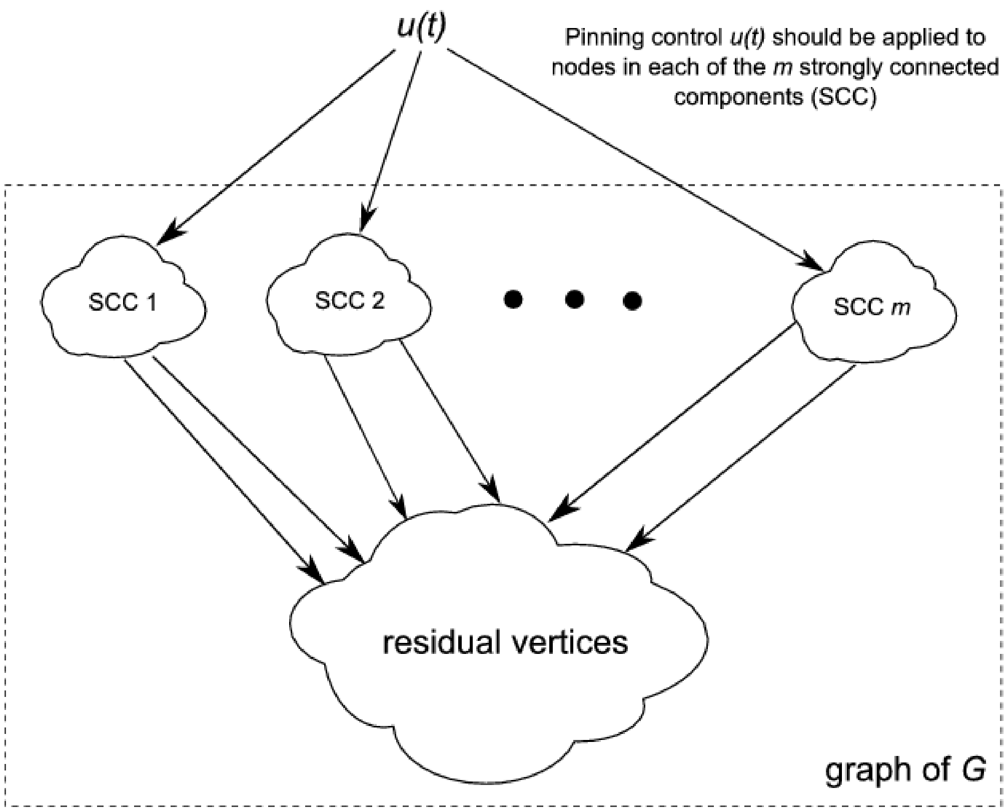
Spanning directed forest

- By writing the matrix of the graph in Frobenius normal form, we find *m* strongly connected components (SCC) which are independent of each other.
- *m* is the degree of irreducibility of the graph matrix and is the smallest number of trees in a spanning directed forest.
- All spanning forest must have roots in each of these SCCs.



Applying control at roots of spanning directed forest.

- Synchronizing control can be achieved if and only if forcing is applied to roots of trees in a spanning directed forest of network.
- Thus at least *m* systems need to be controlled to achieve synchronizing control.



How much and where to apply control

- It's clear that how the control is applied influences its effectiveness.
- Let us now consider 2 questions:
  1. How much control to apply?
  2. Where should control be applied to maximize effectiveness?

How much control to apply?

- Two things to consider here:
  - How many systems to apply control?
  - How much control to apply to each such system?
- Also look at asymptotically behavior as the number of systems *n* grows to infinity.
- Consider the case of undirected connected graph, i.e. there is only 1 strongly connected component.

Upper and lower bounds on β_{min}

- For an undirected graph

$$\beta_{\min} = \lambda_{\min}(G+C)$$

- Upper bounds:

$$\lambda_{\min}(G+C) \le \frac{1}{n}\sum_i c_i$$

$\lambda_{\min}(G+C) \le \lambda_{p+1}(G), \forall p < n$, p is number of nonzero elements in C

- Lower bounds:

$$\beta_{\min} \ge \frac{\lambda_2}{\left(1+\sqrt{1+\frac{\lambda_2}{\sum_i c_i}}\right)^2 n+1} > 0$$

$$\beta_{\min} \ge 2\min(c,1)\left(1-\cos\left(\frac{\pi}{2n+1}\right)\right) > 0$$

How many systems to apply control?

- Upper bound:

  $\lambda_{\min}(G+C) \le \lambda_{p+1}(G), \forall p < n$, *p* is number of systems where control is applied

- This implies that if *k* is the largest integer such that

  $$\lambda_{k+1}(G) < \frac{1}{\alpha}$$

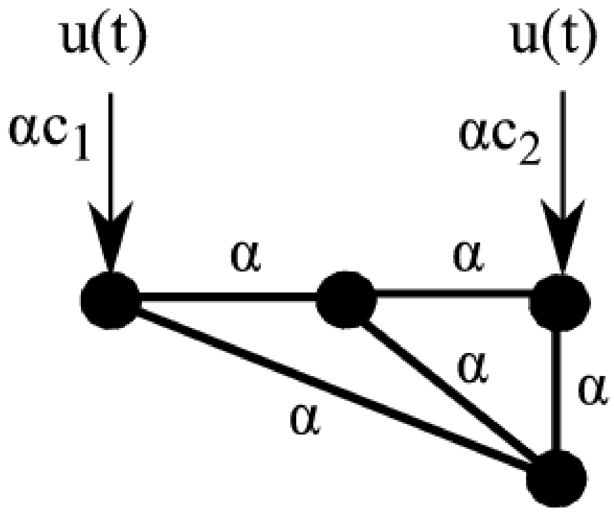
  then you need to apply control to at least *k+1* systems

How much control to apply?

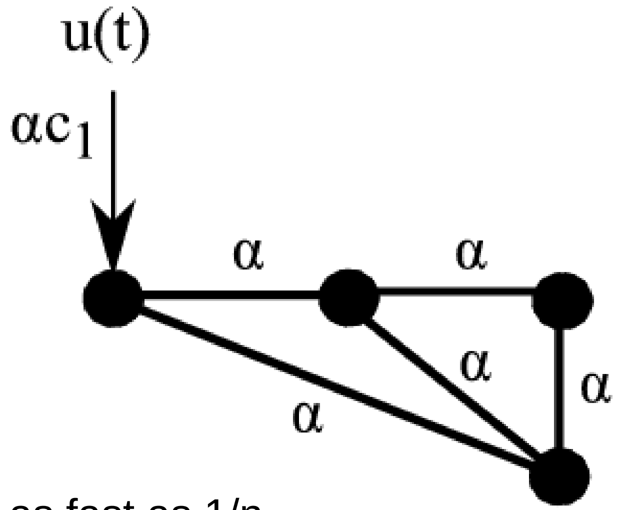
- It's clear that making α and c_i large enough, we obtained synchronized control.
- The lower bound

$$\beta_{\min} \geq 2\min(c,1)\left(1-\cos\left(\frac{\pi}{2n+1}\right)\right) > 0$$

provides a measure of how much is needed.



Asymptotic behavior of β_{min} as function of n



- Consider the upper bound

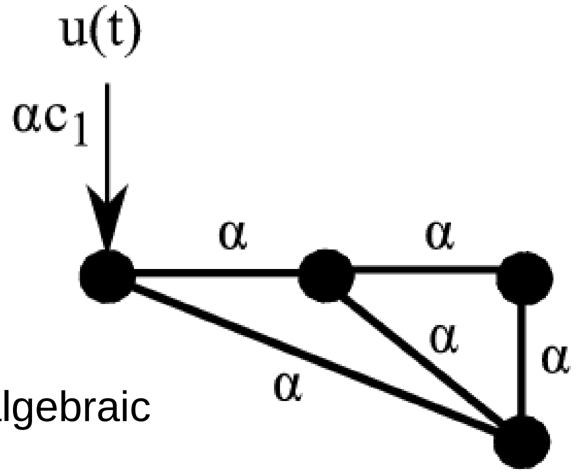
$$\lambda_{\min}(G+C) \le \frac{1}{n}\sum_i c_i$$

- **Assume a single control site ($c_1 > 0$, $c_j = 0$ for j>1).**
- For fixed control parameter c_1, β_{min} decreases at least as fast as 1/n.
- This means that for bounded control parameter c_1 and α, control **cannot** be achieved for large *n*
- In other words, for fixed α, it is *necessary* for c_1 to grow on the order of *n* in order to achieve control.
- Same conclusion apply to vertex-balanced networks, i.e. graphs where the indegree of each vertex is equal to its outdegree.

Asymptotic behavior of β_{min} as function of n

- For fixed α, and m=1, it is *necessary* for c_1 to grow on the order of *n* in order to achieve synchronizing control.
- When is a growth rate of *o(n) sufficient*?
- Consider the lower bound

$$\beta_{\min} \geq \frac{\lambda_2}{\left(1+\sqrt{1+\frac{\lambda_2}{c_1}}\right)^2 n+1} > 0$$



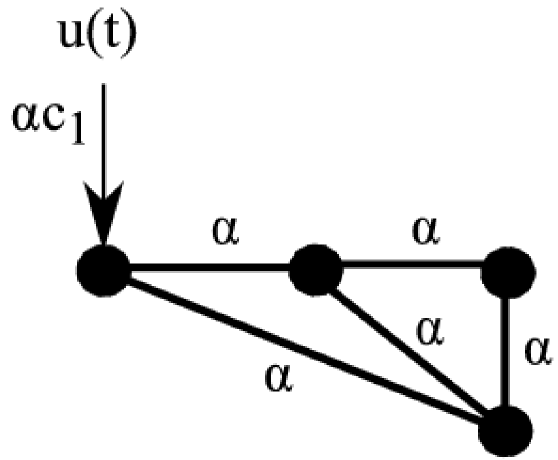
- Where λ_2 is the second smallest eigenvalue of G, i.e. the algebraic connectivity of the network.
- For **fully connected** graphs, or **random** graphs, λ_2 grows on the order of *n* and thus β_{min} will not vanish for large *n* and thus control is achievable for large enough c_1

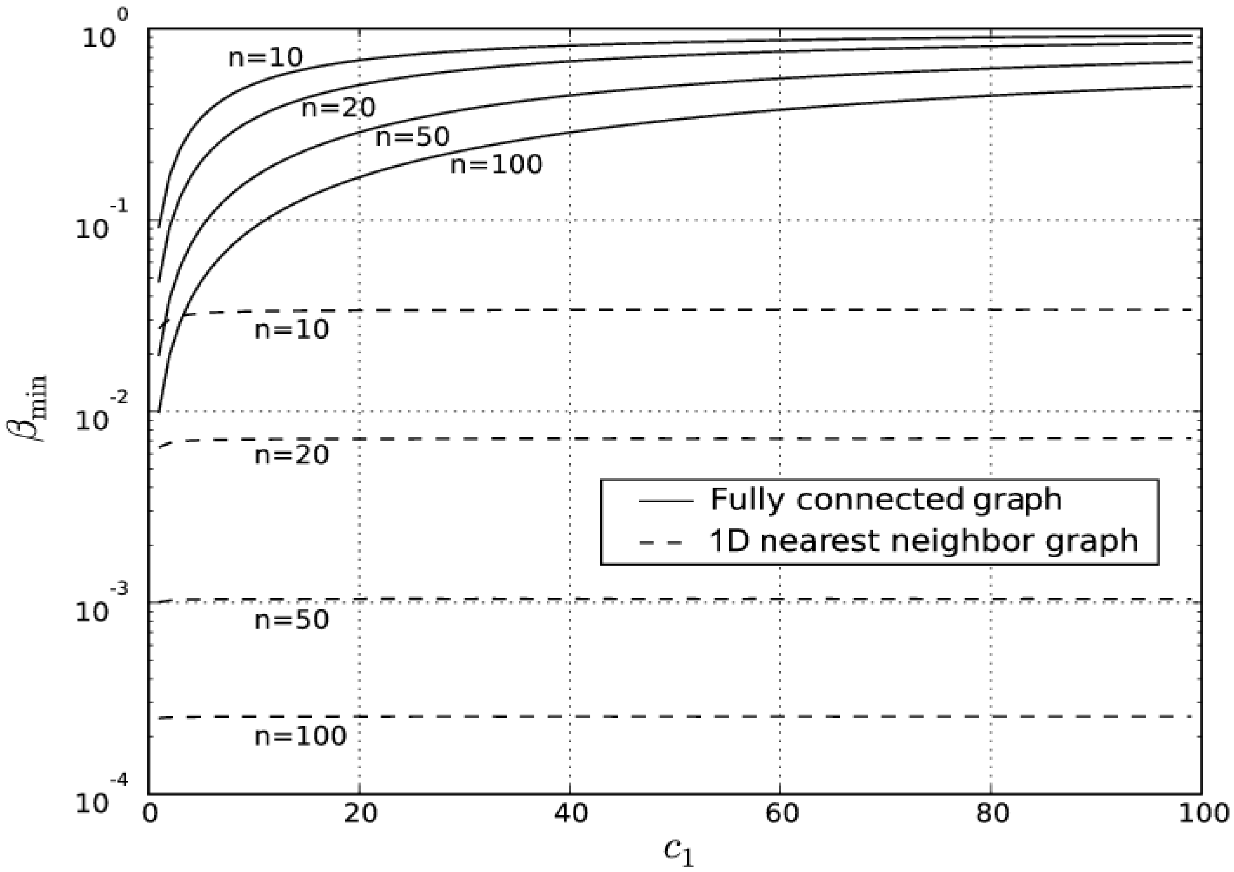
Locally connected graphs

- Theorem: $\beta_{\min} \leq \lambda_2(G)$
- Theorem: for locally connected graphs, $\lambda_2(G) \to 0$ as $n \to \infty$
- Recall the synchronization condition:

$$\beta_{\min} \geq \frac{1}{\alpha}$$

- This means that for locally connected networks, control is not achievable for fixed α as $n \to \infty$

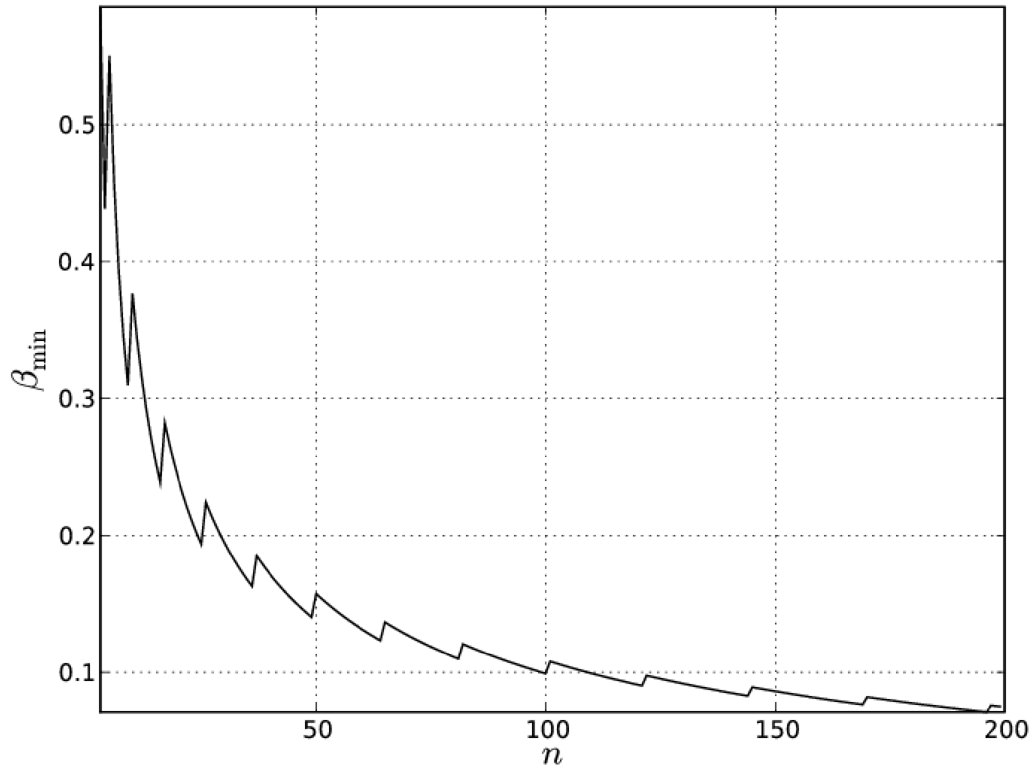




- We see that for the locally connected graph, β_{min} decreases as n increases, regardless of c_1.
- For the fully connected graph, for fixed c_1, β_{min} will decrease as n decreases, but for large c_1, β_{min} will remain bounded from below.

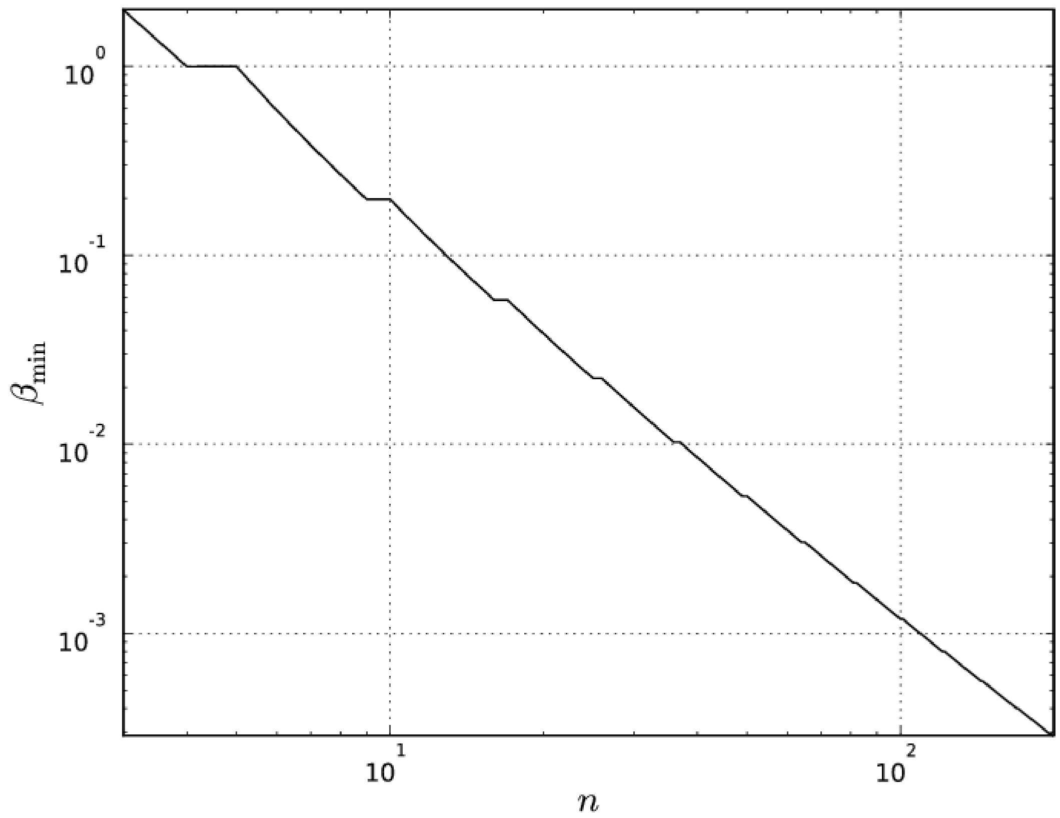
Small number of control sites

- Similar conclusions as for m=1.
- If the number of control sites *m* grows slower than *n*, then α or c_i need to grow to maintain synchronization.
- Example: fully connected graph, with $c_i = \alpha = 1, m = \left\lfloor \sqrt{n} \right\rfloor$



Small number of control sites

- For a bounded parameter α and a locally connected network, control is not possible if *m* grows slower than *n*.
- Example: cycle graph with $c_i = 100n, \alpha = 1, m = \left| \sqrt{n} \right|$



Where to apply control?

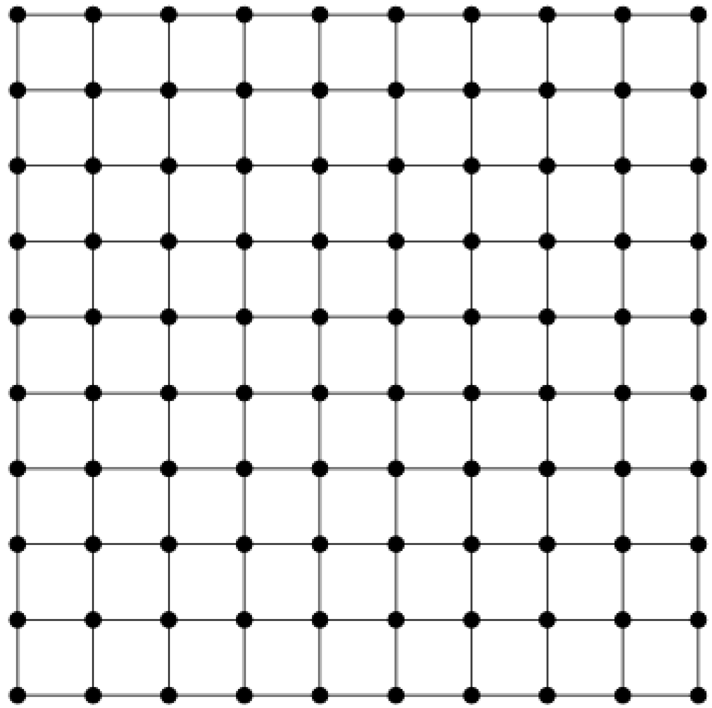
- So far we look at the number of forced systems and the amount of control to apply to each such system.
- The choice of the set of forced system is also important and depends on the underlying network.
- Where should control be applied to maximize effectiveness?

Localization to maximize control effectiveness

- Give a control budget $\sum_i c_i \le \gamma$, how and where should the control signal be applied in order to maximize control effectiveness, i.e. maximize β_{min}?
- Recall that $\beta_{\min} \le \sum c_i / n$
- Answer is simple if we are allowed to apply control to every system (m = n).
- By applying control $c_i = \gamma/n$ to every system, we get $C = \gamma/n\ I$ and $\beta_{min} = \gamma/n$. Thus β_{min} is maximized subject to the control budget.
- What happens if $m < n$?

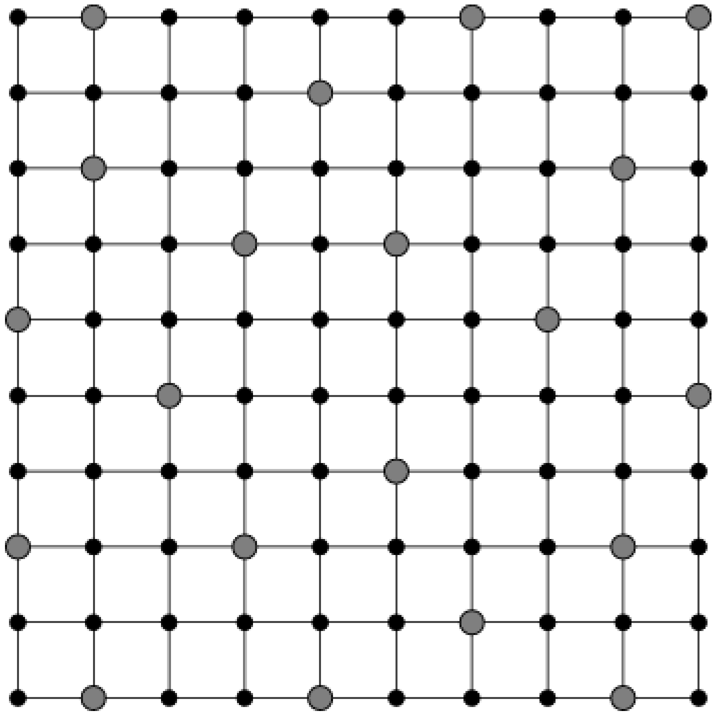
Example: 2D grid graph

- Consider a 2D grid graph where every node is connected to its 4 nearest neighbours on a rectangular grid.
- Employ the following heuristic to maximize β_{min}
  1. Randomly assign m locations to be the control locations
  2. Randomly move one of these locations if such a move increases β_{min}
  3. Repeat step 2



Maximize β_{min}

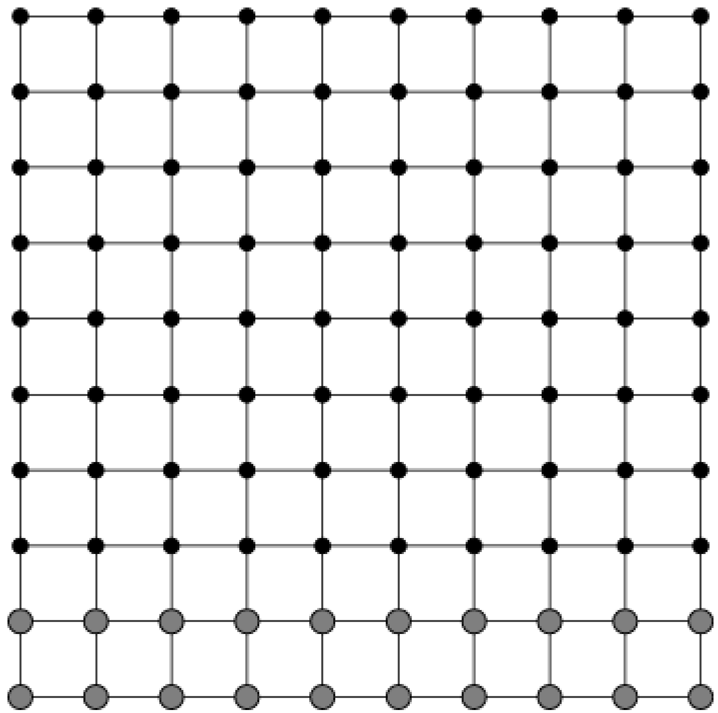
- Apply heuristic for the following parameters: n = 100, m = 20, c = 100, α = 1.
- The resulting control locations are:



- We note that the control locations are spread out.

Minimize β_{min}

- Repeat same heuristic, but move control locations only if it decreases β_{min}
- Result for n = 100, m = 20, c = 100, α = 1.



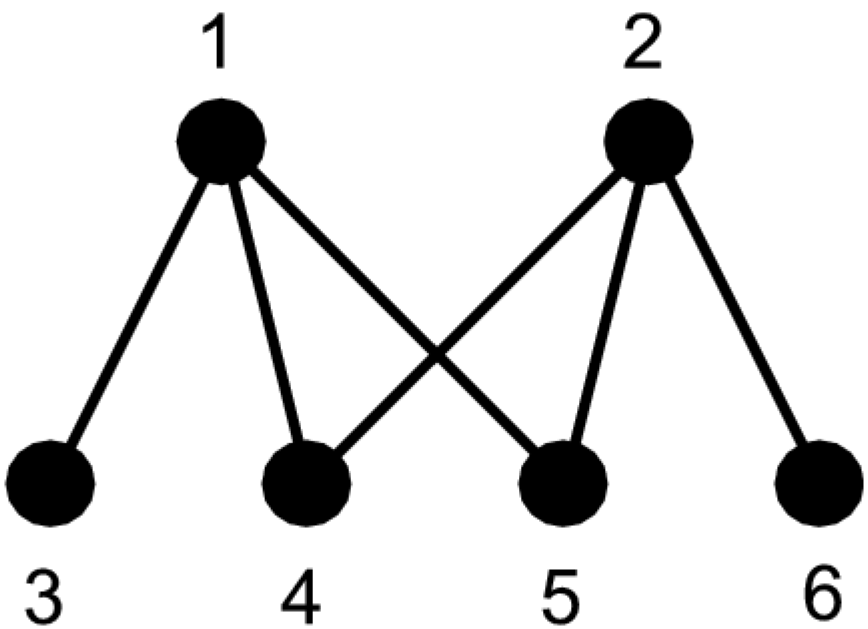
- We note that the control locations are clustered.

Control locations

- It seems intuitive that control should be applied to locations such that any node is reachable by some control signal via a short path.
- Two characterizations to quantify this idea.
- Let P be the set of control locations
  1. $D_P = \max_{v \in V} d(v, P)$
  2. $D_P^a = \dfrac{\sum_{v \in V \setminus P} d(v, P)}{|V \setminus P|}$
- D_P describes the maximal distance between vertices in P and any other vertex, where D_P^a describes the average distance from P to vertices outside of P.
- D_P provides a lower bound to β_{min}

$$\beta_{\min} \geq \frac{c}{2}\left(2\left(r + \frac{1}{2}(2r)^{-D_P}\right)\right)^{-D_P} > 0$$

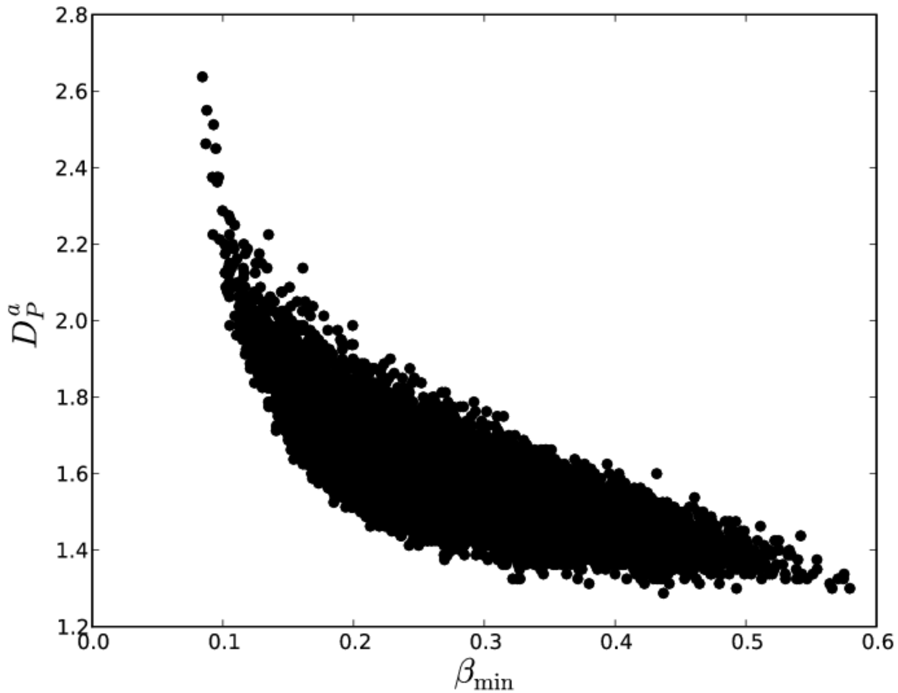
Minimizing D_p does not maximize β_{min}



- Example: single control site (m=1), c = 10.
- Applying control at vertices 4 or 5 minimizes D_P whereas applying control at vertices 1 or 2 maximizes $\beta_{min.}$

How about $\mathbf{D_p^a}$?

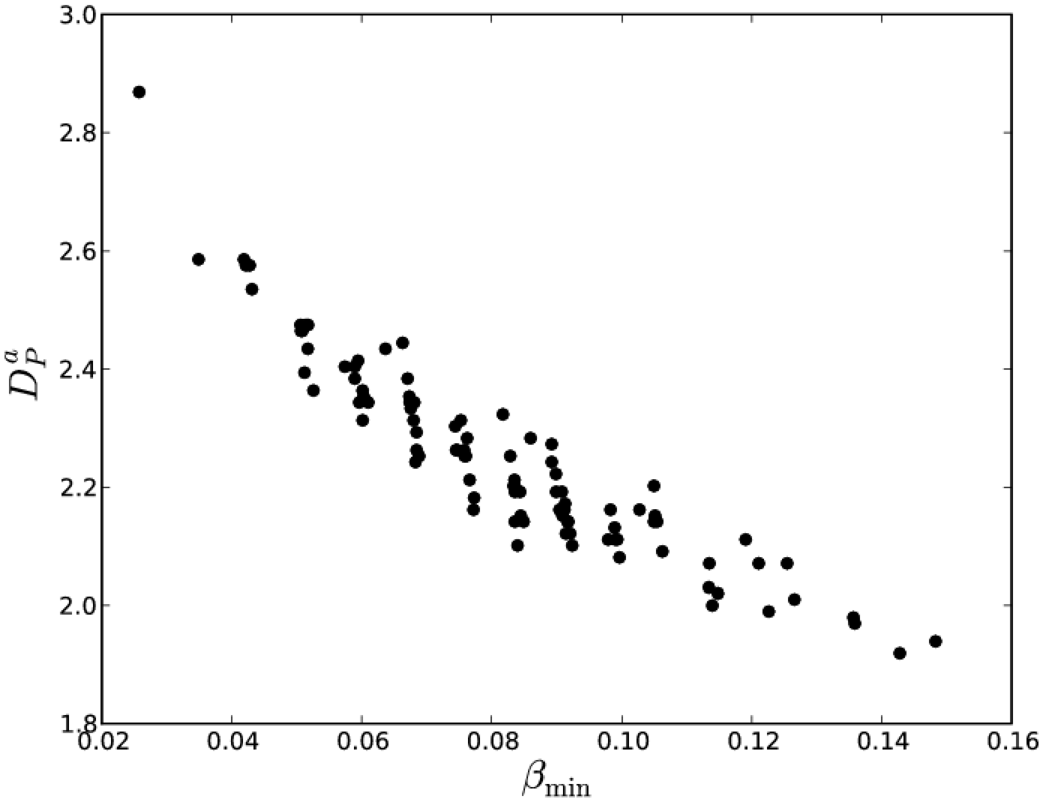
- **For a fixed number of control sites *m*, is β_{min} maximized by a configuration that minimize D_p^a?**
- Experiment 1: 20000 random sets of 20 control locations on the grid graph.



- Upper and lower envelope appears to be convex.

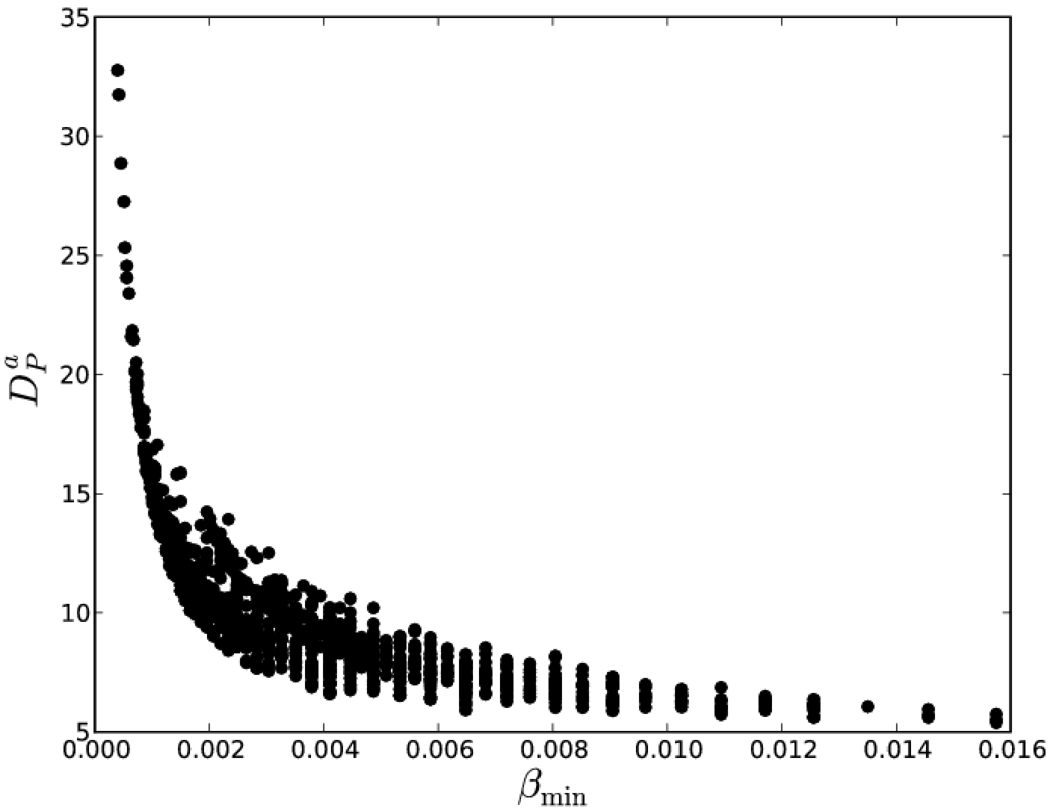
Experiment 2

- A random graph with 100 vertices and 500 edges, c = 100. Single control site



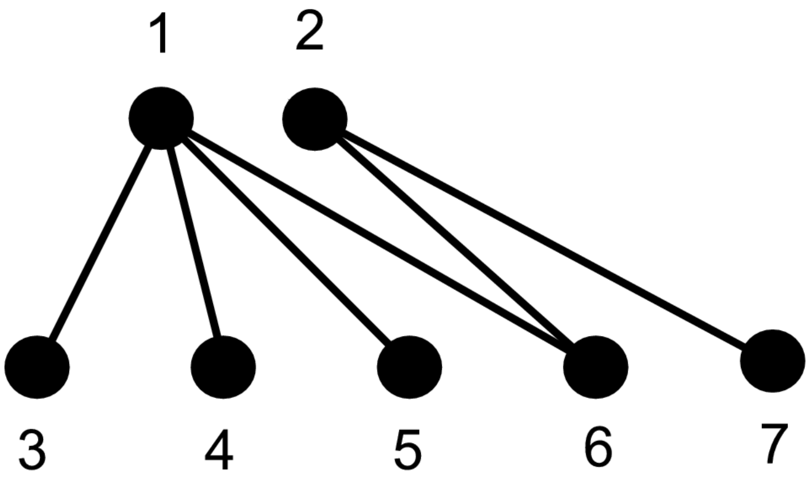
Experiment 3

- Path graph with 100 vertices and 5 control sites, c = 100.



Experiment 4

- Undirected graphs of n vertices are enumerated
- For each graph, a single control is applied with c = 10.
- Check whether the location where β_{min} is maximized corresponds to a minimal D_P^a
- Experiments show that this is true for all graphs of 6 vertices or less, single control site (m=1), c = 10.
- Counterexamples exist for n=7



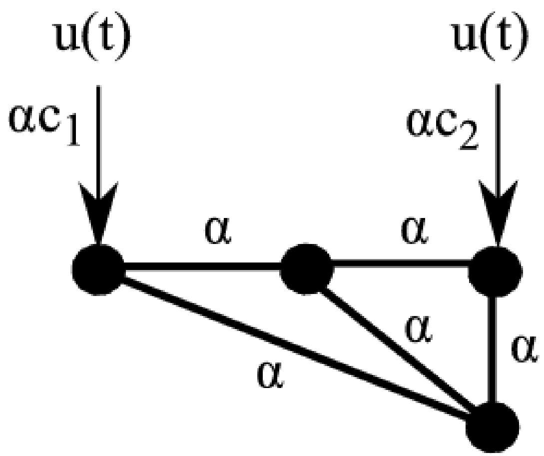
- D_P^a is minimized at vertex 1, whereas β_{min} is maximized by applying control at vertex 6.
- β_{min} is second largest at vertex 1.
- Similar results for the question whether the location where β_{min} is minimized corresponds to a maximal D_P^a (true for $n \leq 6$, false for $n = 7$)

Some provable results

- Theorem: for the cycle graph and large enough forcing, the optimal placement of P minimizes D_P, i.e. the forcing systems are spread out on the cycle.
- A similar result can be shown for path graphs.

Summary

- To achieve synchronizing control, it is necessary to apply control to roots of a spanning directed forest.
- For vertex-balanced networks and a small number of control sites, it is necessary to increase c_i as *n* increases.
- This is sufficent for graphs with a strong connectivity such as fully connected graphs and random graphs.
- However, for locally connected networks, control is not achievable for fixed α as *n* increases, i.e. the intersystem coupling α also need to increase.
- There appears to be an inverse relationship between D_P^a and β_{min} .
- Applying control to location that minimizes the average distance to non-control vertices will (almost?) maximize control effectiveness.



Thank you very much.
Any questions?