MAE 598: Multi-Robot Systems

Fall 2016

Instructor: **Spring Berman**

spring.berman@asu.edu

Assistant Professor, Mechanical and Aerospace Engineering

Autonomous Collective Systems Laboratory

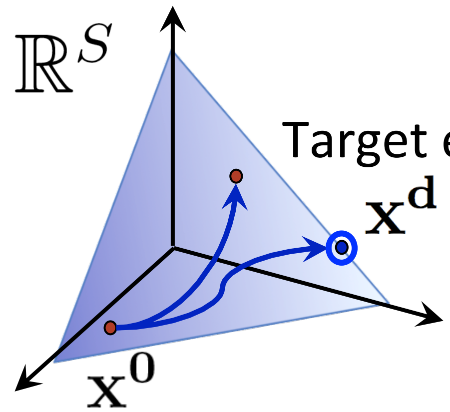
http://faculty.engineering.asu.edu/acs/

Lecture 7

Controller Synthesis using the Macroscopic Model

Optimization

Compute the k_{ij} that minimize a measure of the model's convergence time to $\mathbf{x^d}$



Convex optimization approaches:

➢ Multi-affine model

Relaxation times: time to return to equilibrium after perturbation

[Heinrich and Schuster, *The Regulation of Cellular Systems*, 1996]

- Estimated by linearizing the model around $\mathbf{x^d}$

➢ Linear model

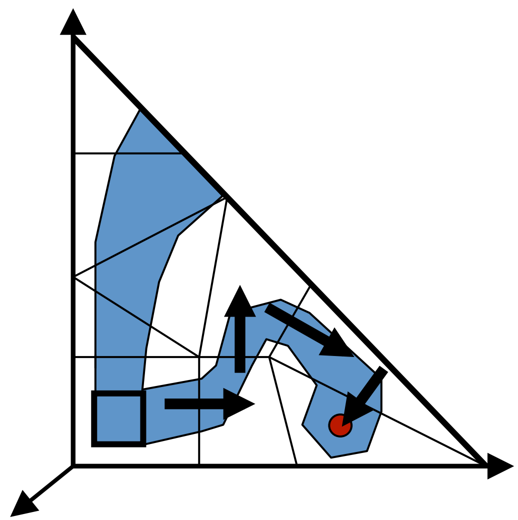
Eigenvalues of $\mathbf{K}$ govern rate of convergence **(ROC)**

- Tradeoff between fast ROC and few task transitions at equilibrium

Hybrid System Macroscopic Models

Controller Synthesis: Vector Fields on Polytopes

$\dot{\mathbf{x}} = f(\mathbf{x}) + \mathbf{B}\mathbf{u}$ P_S = polytope in $\mathbb{R}^S$



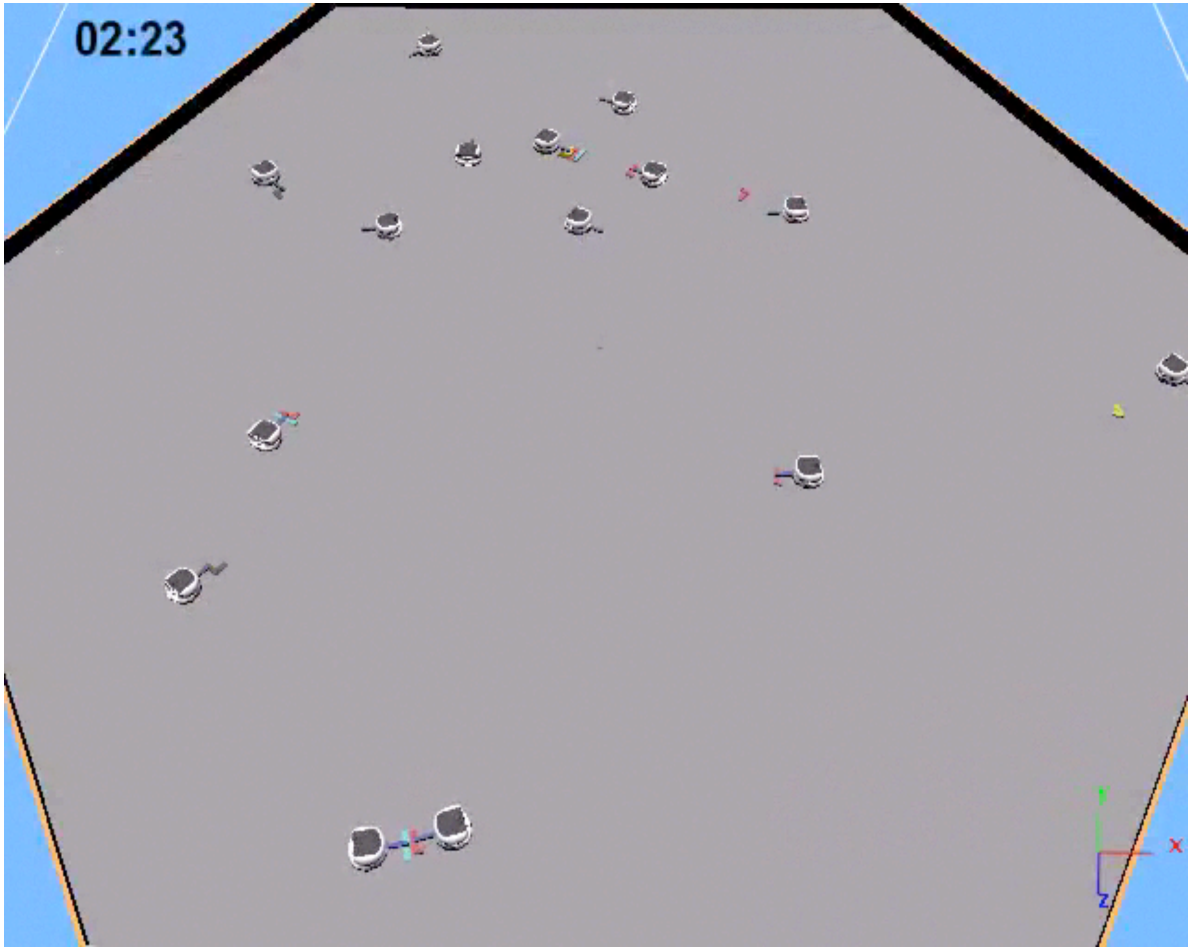
Compute $\mathbf{u}$ that steers $\mathbf{x}$ to a facet of P_S in finite time

[Habets, van Schuppen Automatica '04,
Belta et al., CDC'02]

Swarm Robotic Assembly System

[Loic Matthey, Spring Berman, and Vijay Kumar. "Stochastic Strategies for a Swarm Robotic Assembly System." *ICRA 2009*.]

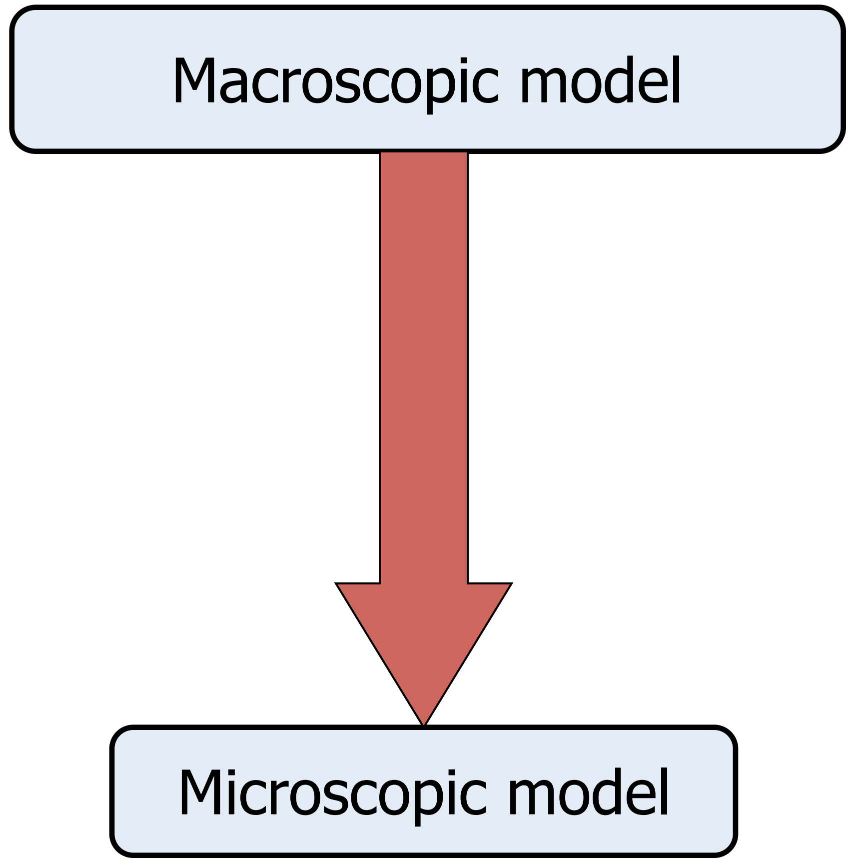
Design a reconfigurable manufacturing system that quickly assembles target amounts of products from a supply of heterogeneous parts



Approach

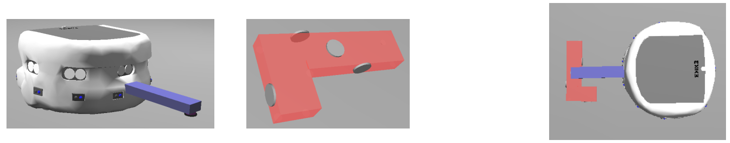
ODEs are functions of probabilities of assembly and disassembly:
Optimize for fast assembly of target amounts of products

$$\dot{\mathbf{x}} = \mathbf{M}\mathbf{K}\mathbf{y}(\mathbf{x})$$

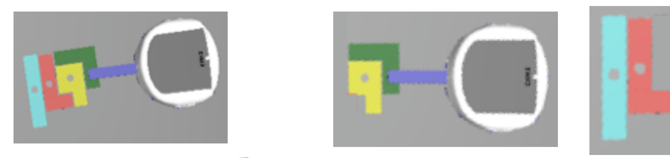


Robots start assemblies and perform disassemblies according to **optimized probabilities**

Decisions Modeled as Chemical Reactions



$$X_R + X_i^u \xrightarrow{e_i} X_i^c \qquad i = 1, \ldots, 8$$

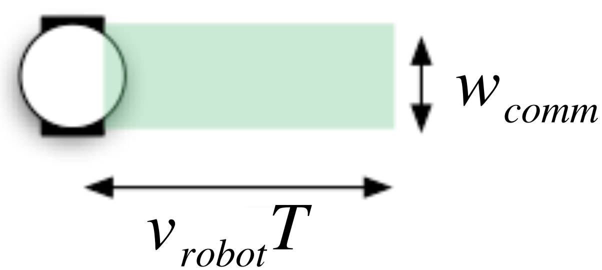


$$X_l^c + X_m^c \xrightarrow{k_j^+} X_n^c + X_R \qquad\qquad X_n^c \xrightarrow{k_j^-} X_l^c + X_m^u$$

$$e_i = A\,p^e, \qquad k_j^+ = A\,p^e\,p_j^a\,p_j^+\,, \qquad k_j^- = p_j^-$$

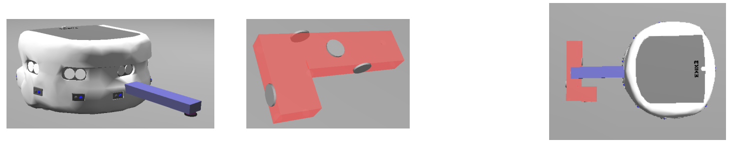
p^e = prob. that a robot encounters a part or another robot $\approx \dfrac{v_{robot} T w_{comm}}{A}$

A = arena area

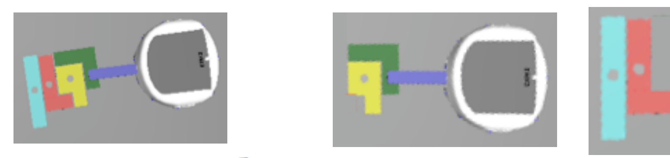


[Correll and Martinoli, Coll. Beh. Workshop, ICRA 2007]

Decisions Modeled as Chemical Reactions



$$X_R + X_i^u \xrightarrow{e_i} X_i^c \qquad i = 1, \ldots, 8$$



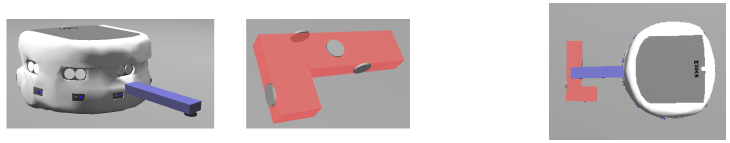
$$X_l^c + X_m^c \xrightarrow{k_j^+} X_n^c + X_R \qquad\qquad X_n^c \xrightarrow{k_j^-} X_l^c + X_m^u$$

$$e_i = A\, p^e \;, \qquad k_j^+ = A\, p^e\, p_j^a\, p_j^+ \;, \qquad k_j^- = p_j^-$$

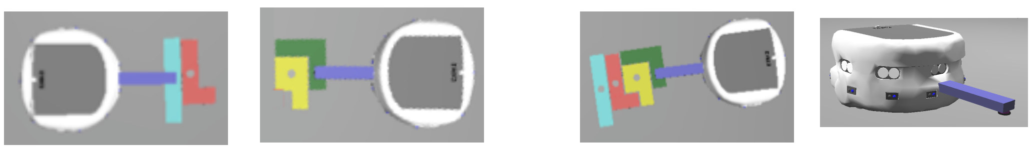
p_j^a = prob. of two robots successfully completing assembly process *j*

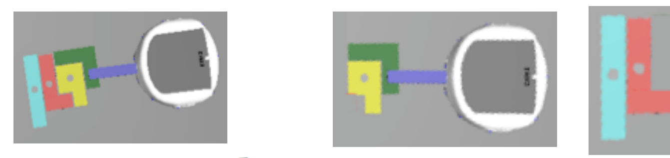
(measured from simulations)

Decisions Modeled as Chemical Reactions



$$X_R + X_i^u \xrightarrow{e_i} X_i^c \qquad i = 1, \ldots, 8$$





$$X_l^c + X_m^c \xrightarrow{k_j^+} X_n^c + X_R \qquad\qquad X_n^c \xrightarrow{k_j^-} X_l^c + X_m^u$$

$$e_i = A\, p^e \;, \qquad k_j^+ = A\, p^e\, p_j^a\, p_j^+ \;, \qquad k_j^- = p_j^-$$

Tunable:

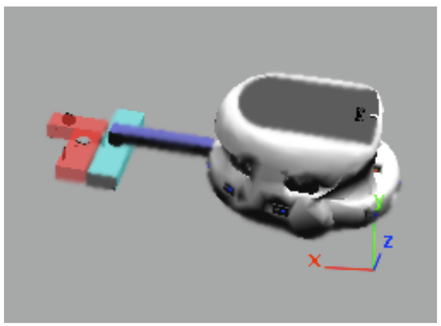
p_j^+ = prob. of two robots starting assembly process j

p_j^- = prob. per unit time of a robot performing disassembly process j

Mapping p_i^+, p_i^- onto the Robot Controllers

Δt = simulation timestep (32 ms)

u = random number uniformly distributed over [0,1]



Robot computes u at each Δt, disassembles the part if

$$u < p_i^- \Delta t$$

Robot computes u, executes assembly if

$$u < p_i^+$$

Reduced Macroscopic Model

Lower-dimensional model (abstract away robots):

$$X_1 + X_2 \underset{k_1^-}{\overset{k_1^+}{\rightleftharpoons}} X_5 \quad X_3 + X_4 \underset{k_2^-}{\overset{k_2^+}{\rightleftharpoons}} X_6 \quad X_5 + X_6 \underset{k_3^-}{\overset{k_3^+}{\rightleftharpoons}} X_7$$

$$X_2 + X_7 \underset{k_4^-}{\overset{k_4^+}{\rightleftharpoons}} X_{F1} \quad X_2 + X_5 \underset{k_5^-}{\overset{k_5^+}{\rightleftharpoons}} X_8 \quad X_6 + X_8 \underset{k_6^-}{\overset{k_6^+}{\rightleftharpoons}} X_{F2}$$

Vector of complexes: $\mathbf{y}(\mathbf{x}) = [x_1x_2 \;\; x_5 \;\; x_3x_4 \;\; x_6 \;\; x_2x_7 \;\; x_{F1} \;\; x_5x_6 \;\; x_7 \;\; x_2x_5 \;\; x_8 \;\; x_6x_8 \;\; x_{F2}]^T$

$$\dot{\mathbf{x}} = \mathbf{M}\mathbf{K}\mathbf{y}(\mathbf{x})$$

We also define a matrix $\mathbf{M} \in \mathbb{R}^{10\times 12}$ in which each entry $\mathbf{M}_{ji}$, $j = 1, ..., 10$, of column $\mathbf{m}_i$ is the coefficient of part type j in complex i (0 if absent). We relabel the rate associated with reaction $(i, j) \in \mathcal{E}$ as k_{ij} and define a matrix $\mathbf{K} \in \mathbb{R}^{12\times 12}$ with entries

$$\mathbf{K}_{ij} = \begin{cases} k_{ji} & \text{if} \quad i \neq j \ , \quad (j,i) \in \mathcal{E} \ , \\ 0 & \text{if} \quad i \neq j \ , \quad (j,i) \notin \mathcal{E} \ , \\ -\sum_{(i,l)\in\mathcal{E}} k_{il} & \text{if} \quad i = j \ . \end{cases} \quad (5)$$

Conservation constraints:

$$\begin{aligned} x_3 - x_4 &= N_1 \\ x_1 + x_5 + x_7 + x_8 + x_{F1} + x_{F2} &= N_2 \\ x_2 + x_5 + x_7 + 2(x_8 + x_{F1} + x_{F2}) &= N_3 \\ x_3 + x_6 + x_7 + x_{F1} + x_{F2} &= N_4 \end{aligned}$$

Reduced Macroscopic Model

$$\dot{\mathbf{x}} = \mathbf{M}\mathbf{K}\mathbf{y}(\mathbf{x})$$

$$\begin{aligned}
x_3 - x_4 &= N_1 \\
x_1 + x_5 + x_7 + x_8 + x_{F1} + x_{F2} &= N_2 \\
x_2 + x_5 + x_7 + 2(x_8 + x_{F1} + x_{F2}) &= N_3 \\
x_3 + x_6 + x_7 + x_{F1} + x_{F2} &= N_4
\end{aligned}$$

The system has a unique, positive, globally asymptotically stable equilibrium.

Proof: Reaction network is *deficiency zero* and *weakly reversible*, does not admit equilibria with some $x_i = 0$

→ We can design $\mathbf{K}$ such that the system always converges to a target equilibrium, $\mathbf{x^d} > \mathbf{0}$

Design of Optimal p_i^+, p_i^-

$$\dot{\mathbf{x}} = \mathbf{M}\mathbf{K}\mathbf{y}(\mathbf{x})$$

$$\begin{aligned}
x_3 - x_4 &= N_1 \\
x_1 + x_5 + x_7 + x_8 + x_{F1} + x_{F2} &= N_2 \\
x_2 + x_5 + x_7 + 2(x_8 + x_{F1} + x_{F2}) &= N_3 \\
x_3 + x_6 + x_7 + x_{F1} + x_{F2} &= N_4
\end{aligned}$$

Recall that $\mathbf{K}$ is a function of $\mathbf{p}$, the vector of p_j^+, p_j^-

$$k_j^+ = A\, p^e\, p_j^a\, p_j^+ \;, \qquad k_j^- = p_j^-$$

Select $\mathbf{x^d}$ that satisfies conservation constraints

Compute $\mathbf{p}$ that **minimizes the system convergence time** to $\mathbf{x^d}$ subject to constraints:

$$\mathbf{M}\mathbf{K}(\mathbf{p})\mathbf{y}(\mathbf{x^d}) = \mathbf{0}, \quad \mathbf{0} \leq \mathbf{p} \leq \mathbf{1}$$

Optimization Problems

I. Linear Program

Objective: Maximize the average inverse *relaxation time* τ_j

- τ_j = time for system mode to return to equilibrium after perturbation
- Estimated by linearizing the ODE model around $\mathbf{x}^{\mathbf{d}}$

[Heinrich and Schuster, The Regulation of Cellular Systems, 1996]

For reaction $X_k + X_l \underset{k_j^-}{\overset{k_j^+}{\rightleftharpoons}} X_m$: $\tau_j^{-1} = k_j^+(x_k^d + x_l^d) + k_j^-$

II. Monte Carlo Method

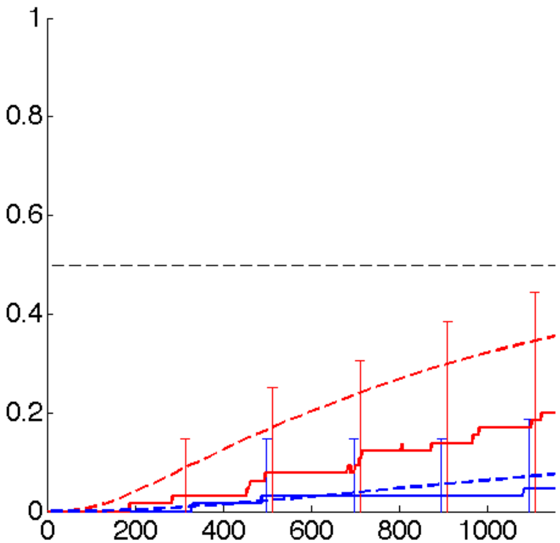
Objective: Minimize time for system to reach $0.1\left\|\mathbf{x}^0 - \mathbf{x}^{\mathbf{d}}\right\|_2$

Optimization Improves Convergence Rate

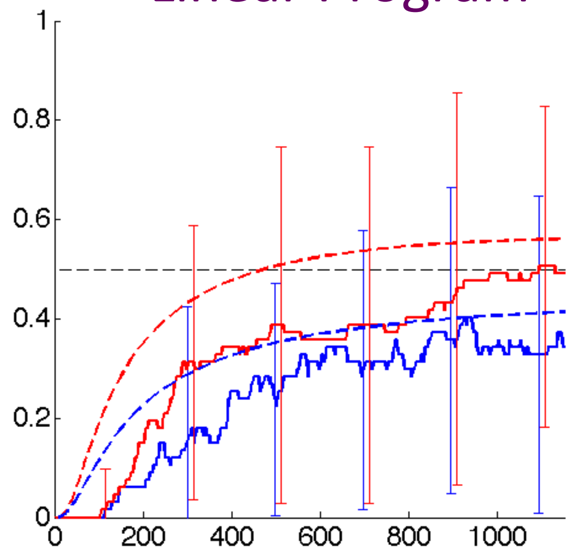
- 15 robots, 15 basic parts
- Simulations averaged over 30 runs

	Simulation	Model	Target
$x_{F1} / (x_{F1}^d + x_{F2}^d)$	solid blue line	dashed blue line	$\boldsymbol{\alpha} = 0.5$
$x_{F2} / (x_{F1}^d + x_{F2}^d)$	solid red line	dashed red line	$(\mathbf{1} - \boldsymbol{\alpha}) = 0.5$

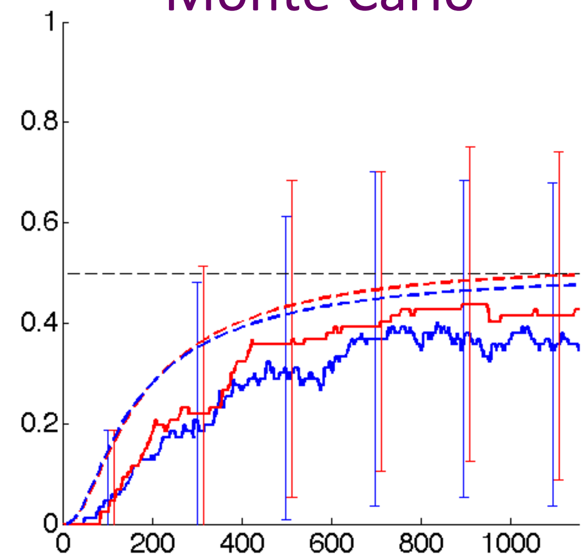
Randomly selected p_i^+, p_i^-



p_i^+, p_i^- from Linear Program



p_i^+, p_i^- from Monte Carlo



Time (sec)

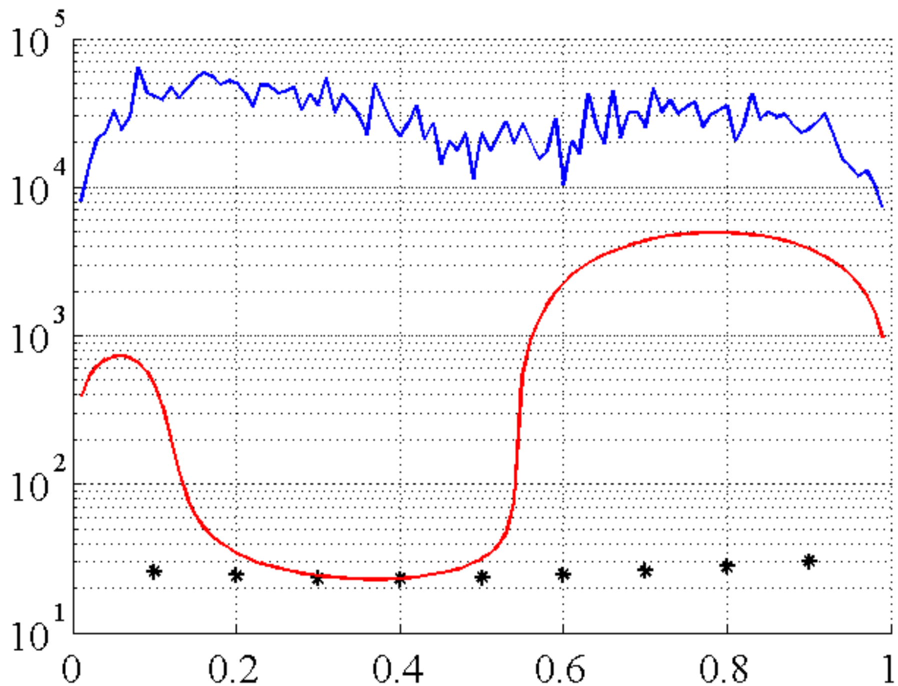
Linearization is most effective for $\boldsymbol{\alpha} \approx 0.2 - 0.5$

- For all α, linear program only changes rates of disassembling F1, F2

- Monte Carlo p_i^+, p_i^- yield fastest convergence but takes ~10 hrs to compute (in 2009), vs. <1 s using the linear program (2 GHz laptop)

Time for reduced model to reach

$$0.1\left\|\mathbf{x}^0 - \mathbf{x}^{\mathbf{d}}\right\|_2$$



$$\alpha = x_{F1}^d / (x_{F1}^d + x_{F2}^d)$$

- 300 basic parts

p_i^+, p_i^- from:

Random selection
(Average of 100 values)

Linear Program

Monte Carlo Method

Linearization is most effective for $\boldsymbol{\alpha} \approx 0.2 - 0.5$

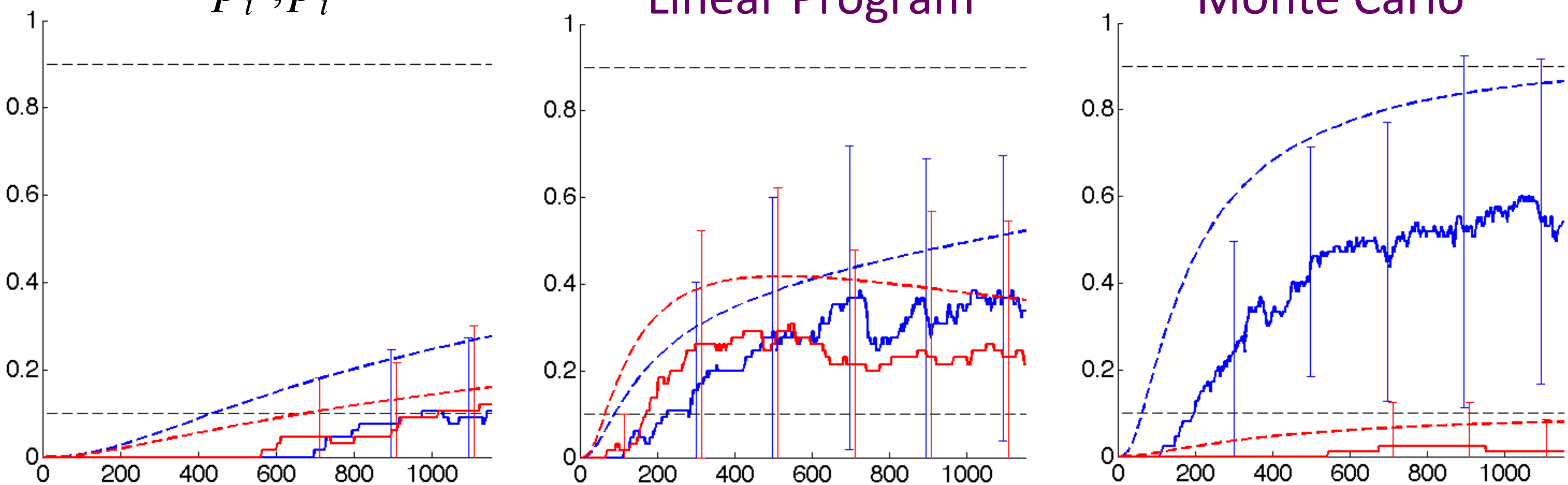
- 15 robots, 15 basic parts
- Simulations averaged over 30 runs

	Simulation	Model	Target
$x_{F1}/(x_{F1}^d + x_{F2}^d)$	solid blue line	dashed blue line	$\boldsymbol{\alpha} = 0.9$
$x_{F2}/(x_{F1}^d + x_{F2}^d)$	solid red line	dashed red line	$(\mathbf{1}-\boldsymbol{\alpha}) = 0.1$

Randomly selected p_i^+, p_i^-

p_i^+, p_i^- from Linear Program

p_i^+, p_i^- from Monte Carlo



Time (sec)

Linearization is most effective for $\alpha \approx 0.2 - 0.5$

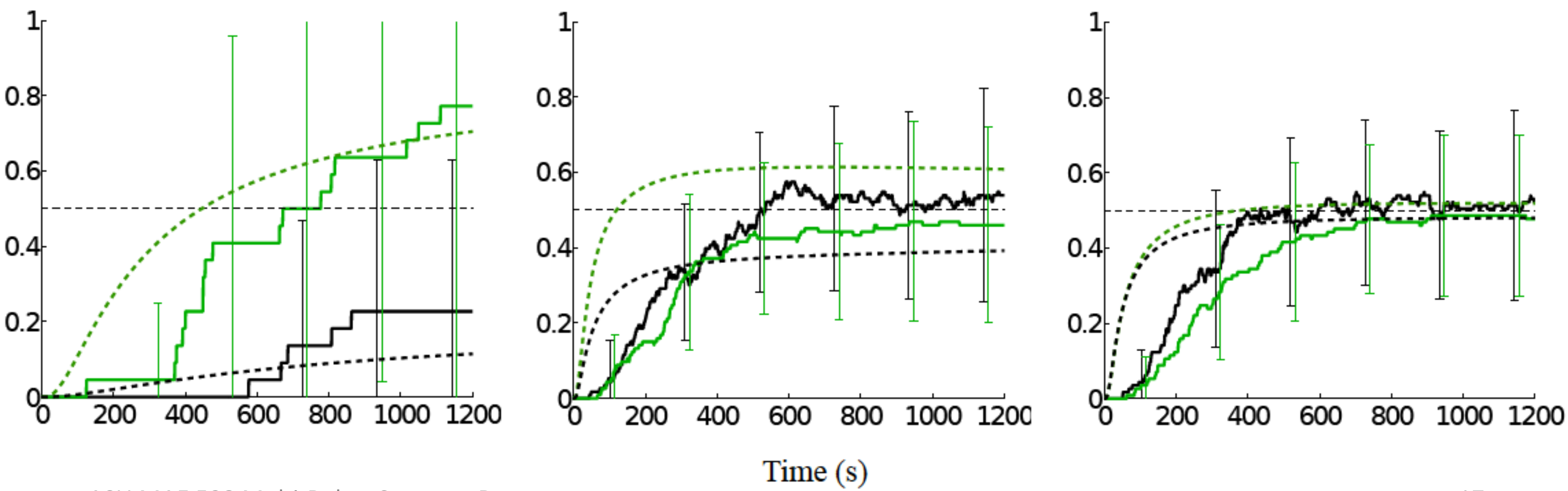
- 50 robots, 50 basic parts
- Simulations averaged over 20 runs

	Simulation	Model	Target	
$x_{F1}/(x_{F1}^d + x_{F2}^d)$	——— (black)	- - - - (black)	α =	0.5
$x_{F2}/(x_{F1}^d + x_{F2}^d)$	——— (green)	- - - - (green)	$(1-\alpha)$ =	0.5

Randomly selected p_i^+, p_i^-

p_i^+, p_i^- from Linear Program

p_i^+, p_i^- from Monte Carlo



Swarm Multi-Site Deployment

[Spring Berman, Adam Halasz, M. Ani Hsieh, and Vijay Kumar. "Optimized Stochastic Policies for Task Allocation in Swarms of Robots." *IEEE Transactions on Robotics*, 2009]

Swarm Multi-Site Deployment

- Model interconnection topology of sites as a directed graph

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$ $\quad \mathcal{V}$ = set of sites $\quad \mathcal{E} = \{(i,j) \in \mathcal{V} \times \mathcal{V} \mid \underline{i \sim j}\}$

can travel from *i* to *j*

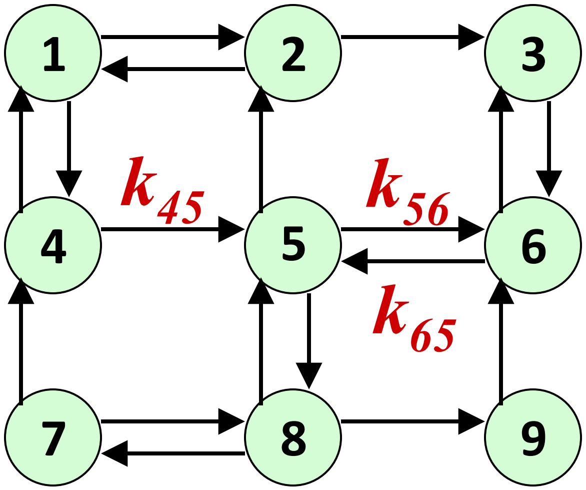
- Assume that $\mathcal{G}$ is strongly connected (directed path btwn. each pair of sites)

k_{ij} = Transition probability per unit time for one robot at site *i* to travel to site *j*

- Choose for rapid, efficient redistribution

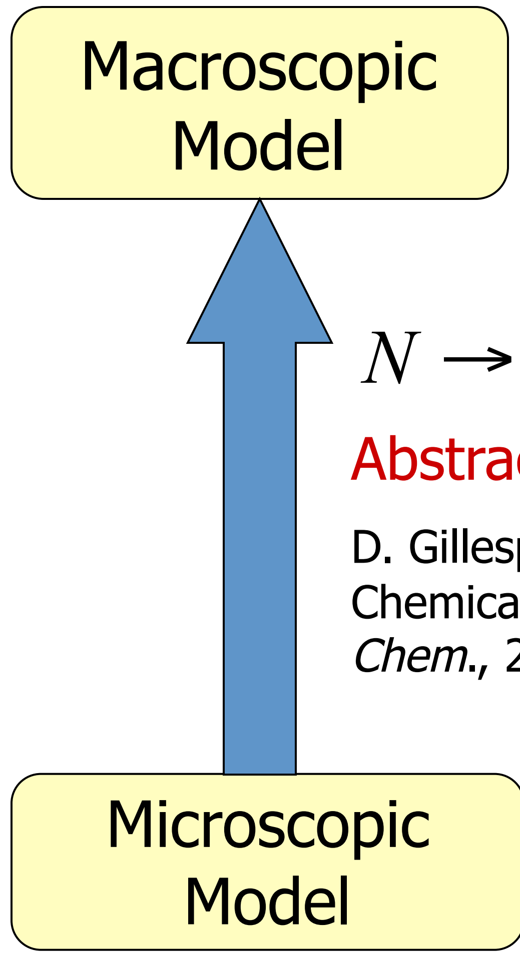
- Assume that each robot:
  - knows $\mathcal{G}$, all k_{ij}, task at each site
  - can navigate between sites
  - can sense neighboring robots

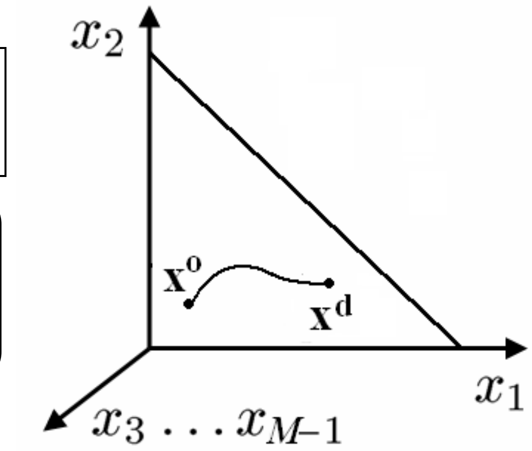
M Sites



Approach

$$\dot{\mathbf{x}} = -\mathbf{K}\mathbf{x}$$





- Ordinary differential equations in terms of k_{ij} and the fraction of robots x_i at each site *i*

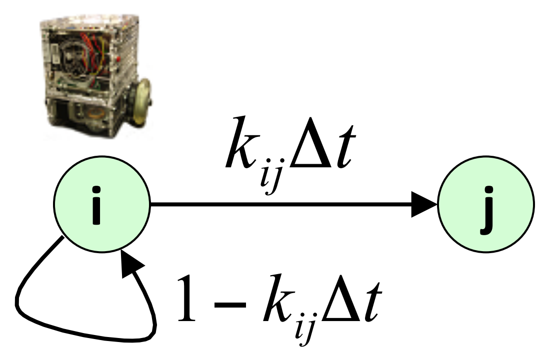
$N \rightarrow \infty$

Abstraction

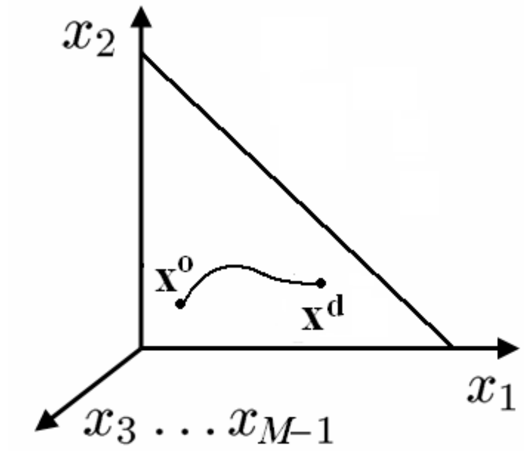
D. Gillespie, "Stochastic Simulation of Chemical Kinetics," *Annu. Rev. Phys. Chem.*, 2007

- *N* robots, *M* behavior states: **{Doing task at site 1, Doing task at site 2, …, Doing task at site *M*}**

- Could also include states that represent travel between pairs of sites



Approach



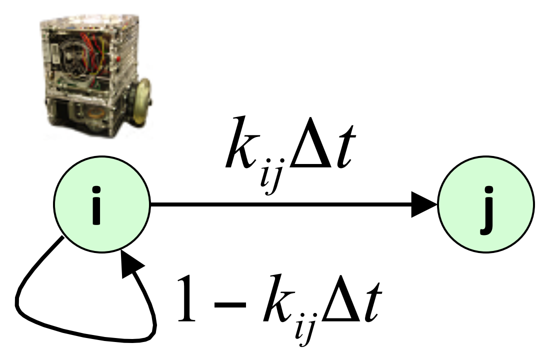
- Analysis and optimization tools to choose k_{ij}

Macroscopic Model

"Top-down" controller synthesis approach is computationally inexpensive and gives guarantees on performance

Microscopic Model

- Switch according to k_{ij}; motion control for tasks at sites, navigation



Macroscopic Model

$x_i(t)$ = Fraction of robots at site *i* at time *t* $\quad \mathbf{x} = [x_1 \ \dots \ x_M]^T$

$$\dot{x}_i(t) = \sum_{j \sim i} k_{ji} x_j(t) - \sum_{i \sim j} k_{ij} x_i(t)$$

$$\boxed{\dot{\mathbf{x}} = -\mathbf{K}\mathbf{x}}$$

(a) $\mathbf{K}^T \mathbf{1} = \mathbf{0}$,

(b) $\mathbf{K}_{ij} \leq 0 \quad \forall (i,j) \in \mathcal{E}$

Conservation constraint: $\mathbf{1}^T \mathbf{x} = 1$

Macroscopic Model

$$\boxed{\dot{\mathbf{x}} = -\mathbf{K}\mathbf{x}} \qquad \mathbf{1}^T\mathbf{x} = 1$$

(a) $\mathbf{K}^T\mathbf{1} = \mathbf{0}$, (b) $\mathbf{K}_{ij} \leq 0 \quad \forall (i,j) \in \mathcal{E}$

- There is a unique, stable equilibrium [Proof uses Perron-Frobenius Theorem]

x_i^d = Target fraction of robots at site *i* $\qquad \mathbf{x^d} = [x_1^d \; ... \; x_M^d]^T$

→ If k_{ij} are chosen so that (c) $\mathbf{K}\mathbf{x^d} = 0$,
the system always converges to the target distribution

Macroscopic Model

$$\dot{\mathbf{x}} = -\mathbf{K}\mathbf{x} \qquad \mathbf{1}^T\mathbf{x} = 1$$

(a) $\mathbf{K}^T\mathbf{1} = \mathbf{0}$, (b) $\mathbf{K}_{ij} \leq 0 \quad \forall (i,j) \in \mathcal{E}$, (c) $\mathbf{K}\mathbf{x^d} = 0$

- Real parts of eigenvalues of $\mathbf{K}$ govern rate of convergence to $\mathbf{x^d}$

  → High k_{ij} for fast redistribution

- Probability that a robot at *i* starts moving to *j* in a time step is proportional to k_{ij}

  → Low k_{ij} for few idle trips between sites at equilibrium

⇒ Optimal $\mathbf{K}$ maximizes convergence rate of system subject to a constraint on inter-site traffic at equilibrium

Macroscopic Model

$$\boxed{\dot{\mathbf{x}} = -\mathbf{K}\mathbf{x}} \qquad \mathbf{1}^T\mathbf{x} = 1$$

(a) $\mathbf{K}^T\mathbf{1} = \mathbf{0}$, (b) $\mathbf{K}_{ij} \leq 0 \quad \forall (i,j) \in \mathcal{E}$, (c) $\mathbf{K}\mathbf{x}^{\mathbf{d}} = 0$

Traffic along edge ***(i,j)*** = $k_{ij}x_i$ (fraction of robots per unit time exiting *i* to go to *j*)

Possible constraints on inter-site traffic at equilibrium:

(1) Total limit: $\sum_{(i,j)\in\mathcal{E}} k_{ij}x_i^d \leq c_{tot}$ or

(2) Edge limits: $k_{ij}x_i^d \leq c_{ij}, \quad (i,j) \in \mathcal{E}$

Design of Optimal $\mathbf{K}$ Matrix

- Maximize a measure of the convergence rate of model

$$\dot{\mathbf{x}} = -\mathbf{K}\mathbf{x}$$

subject to one of the 2 constraints on equilibrium traffic

- Measure the degree of convergence to $\mathbf{x^d}$ in terms of the *fraction of misplaced robots,*

$$\mu_n(\mathbf{x}) = ||\mathbf{x} - \mathbf{x^d}||_n \qquad n = 1 \text{ or } 2$$

- One problem minimizes convergence time directly using a Monte Carlo method; the others maximize functions of the eigenvalues of $\mathbf{K}$ using convex optimization

$\mathbb{K}$ Matrix Optimization Problems

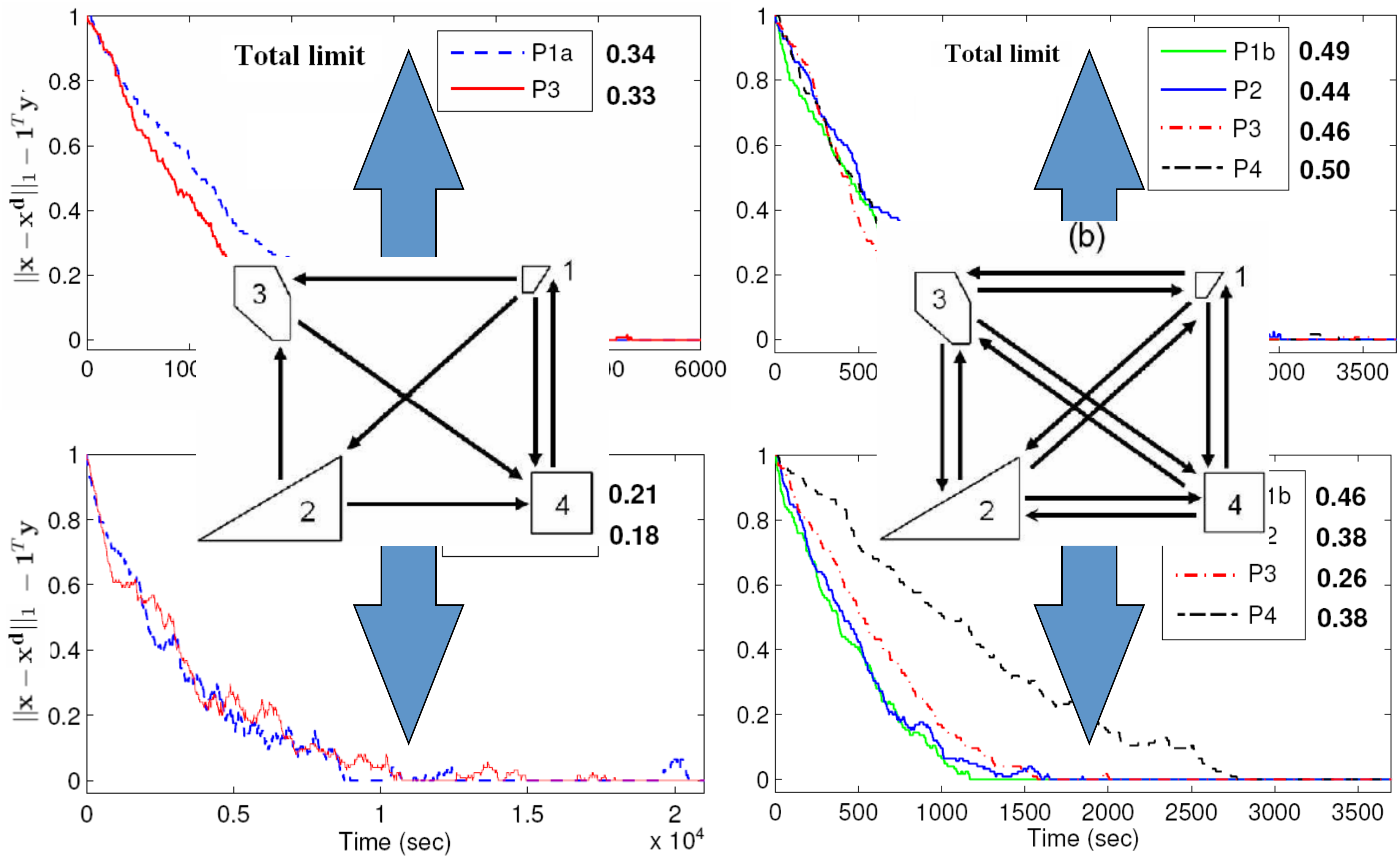
Prob.	*Objective*	*FC*	*DB*	$\mathbf{x}^0$
P1a	Maximize asymptotic ROC			
P1b	Maximize asymptotic ROC		√	
P2	Maximize overall ROC	√		
P3	Minimize time to reach $0.1\mu_2(\mathbf{x}^0)$			√
P4	Maximize ROC along $\mathbf{x}^\mathrm{d} - \mathbf{x}^0$	√		√

FC = fully connected (each site accessible from all other sites)

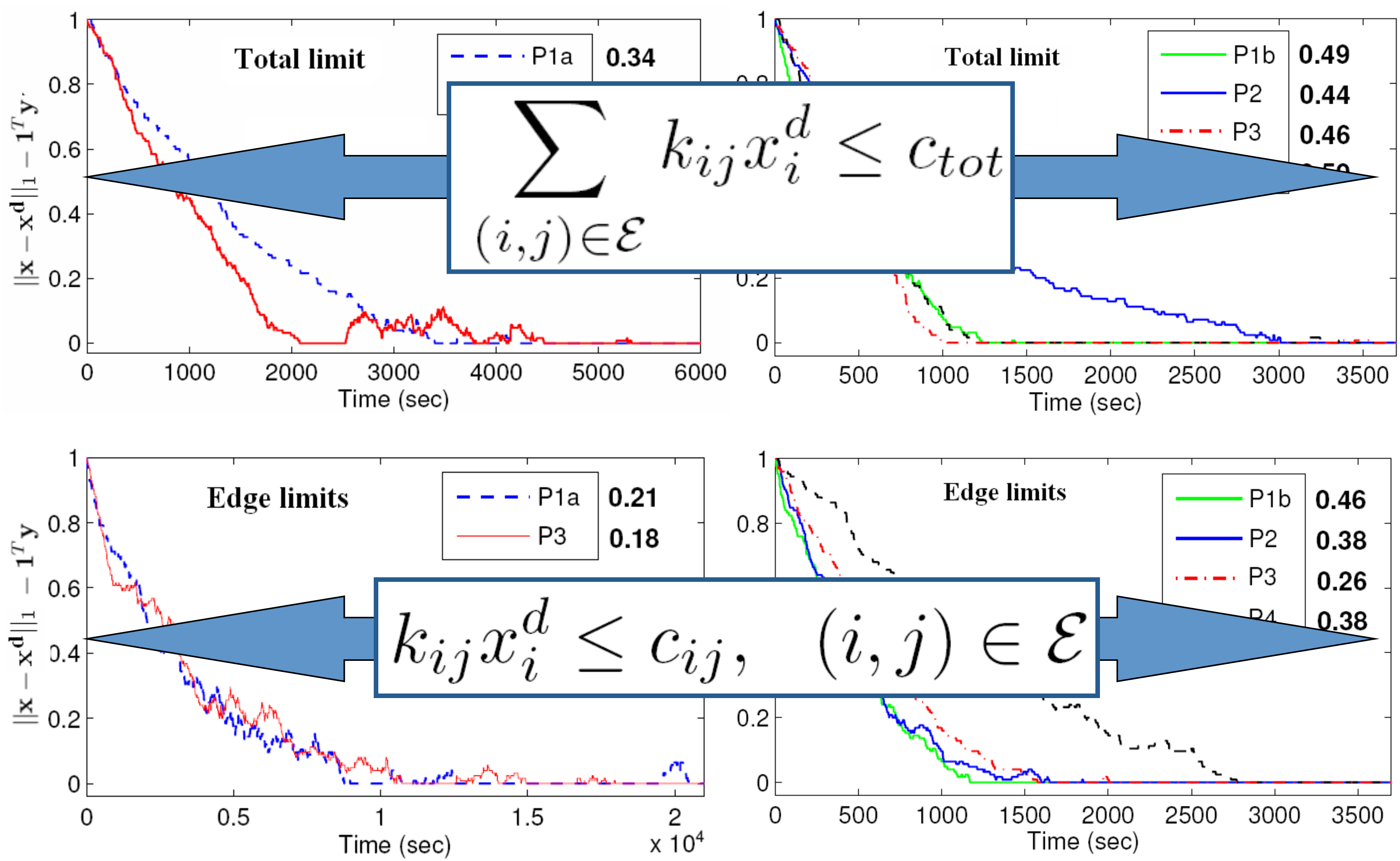
DB = detailed balance condition holds

$\mathbf{x}^0$ = initial distribution known   ROC = rate of convergence

Optimal K Comparison



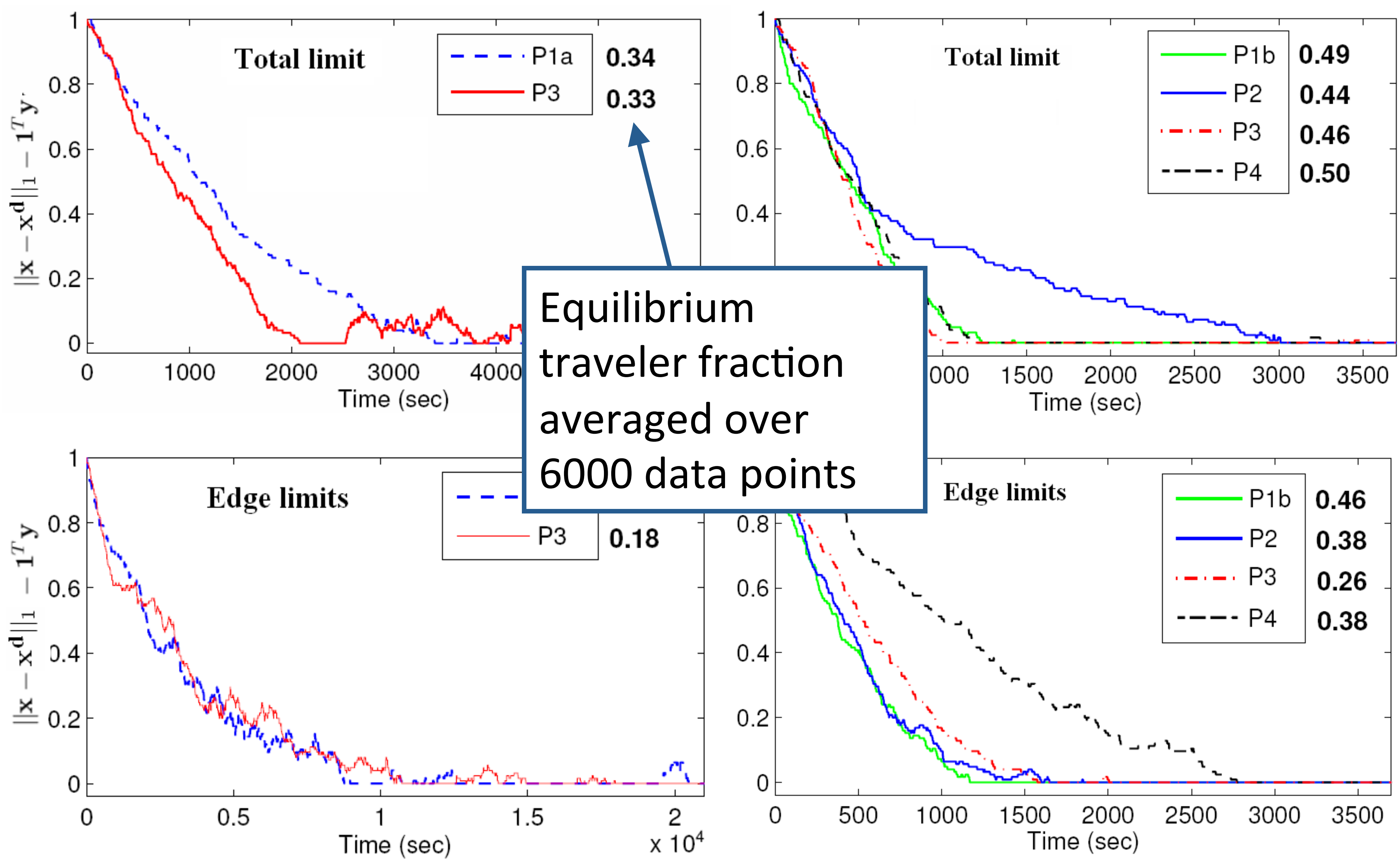
Optimal K Comparison



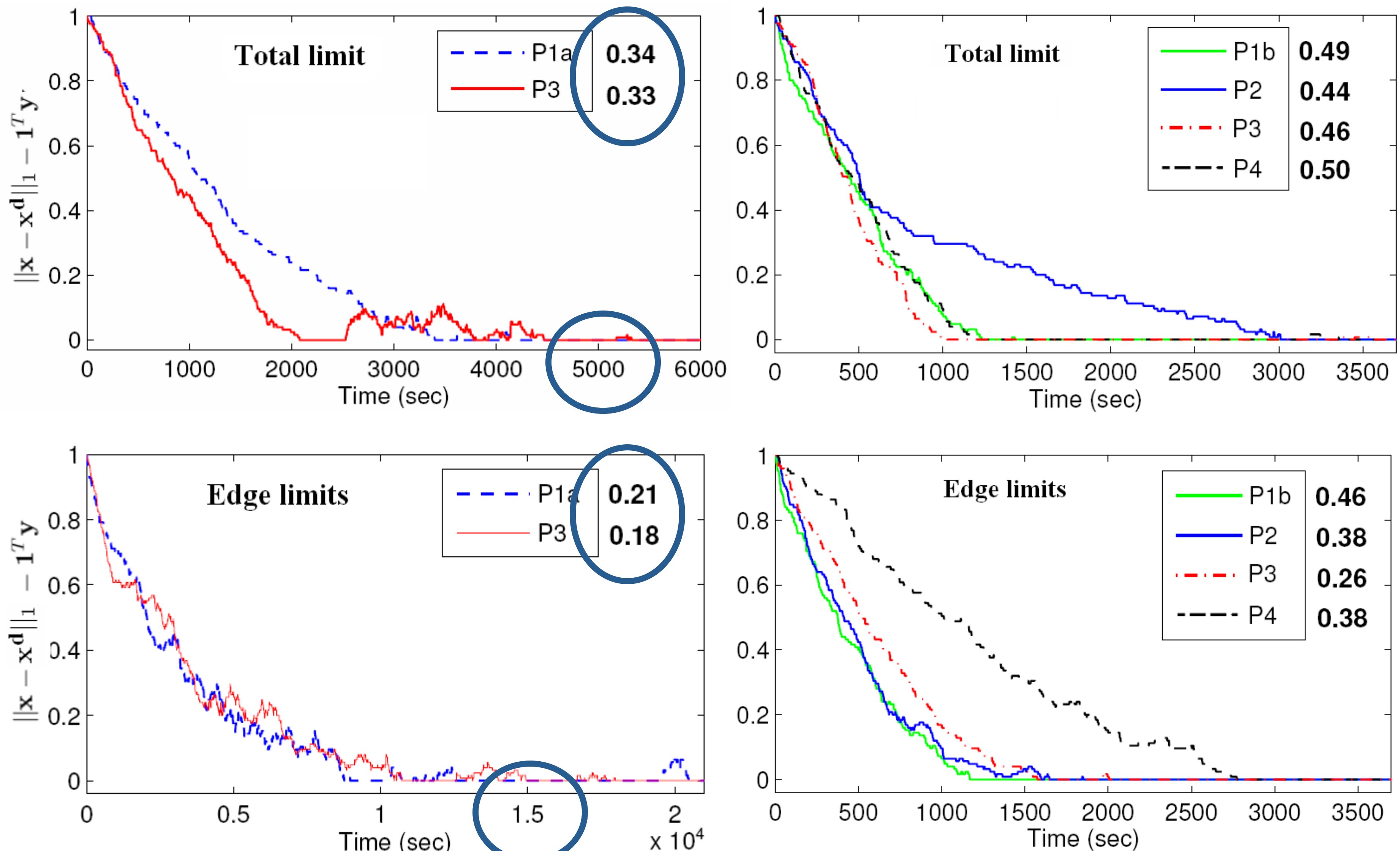
$$\sum_{(i,j)\in\mathcal{E}} k_{ij}x_i^d \le c_{tot}$$

$$k_{ij}x_i^d \le c_{ij}, \quad (i,j)\in\mathcal{E}$$

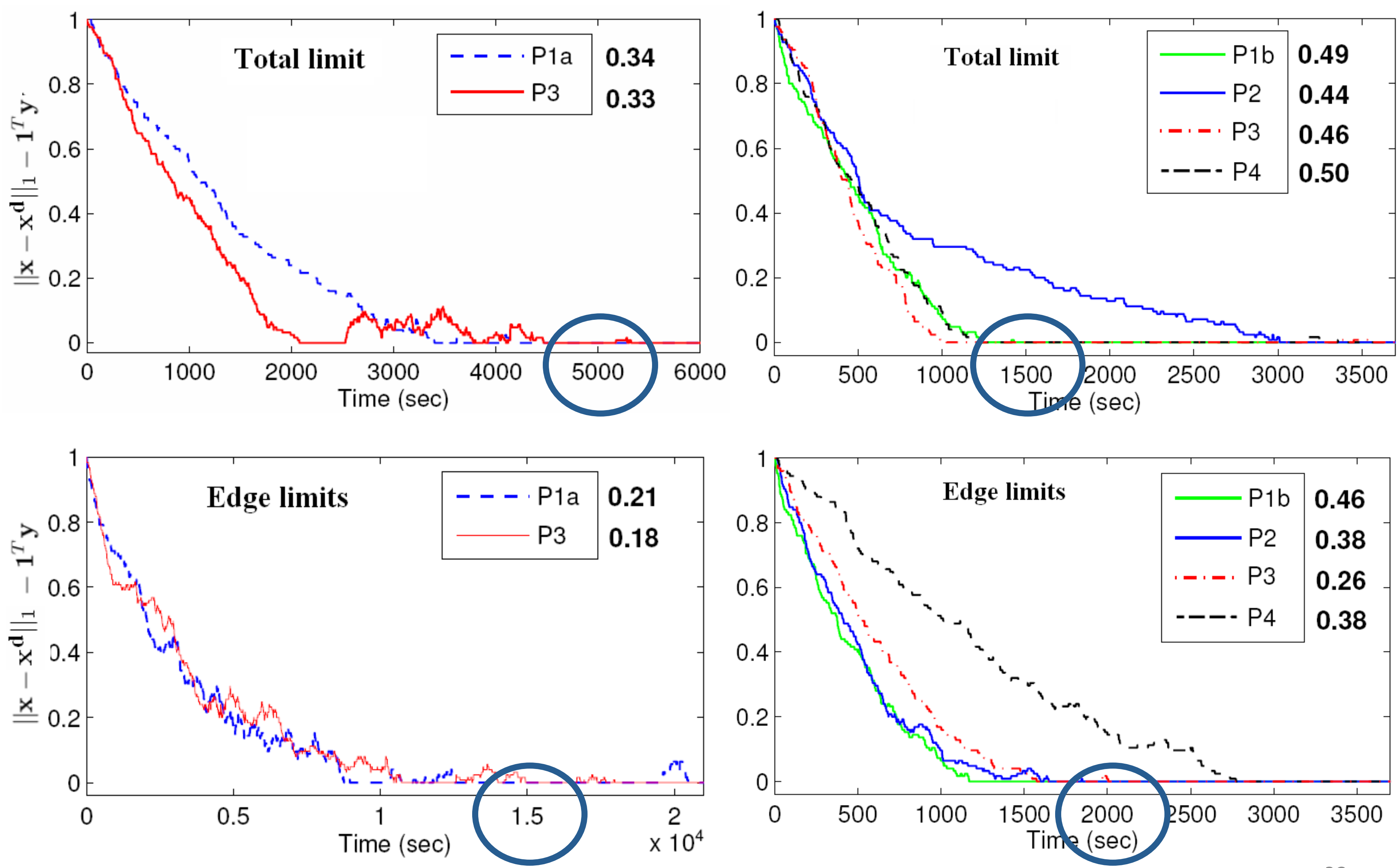
- Tradeoff between convergence rate, equilibrium traffic



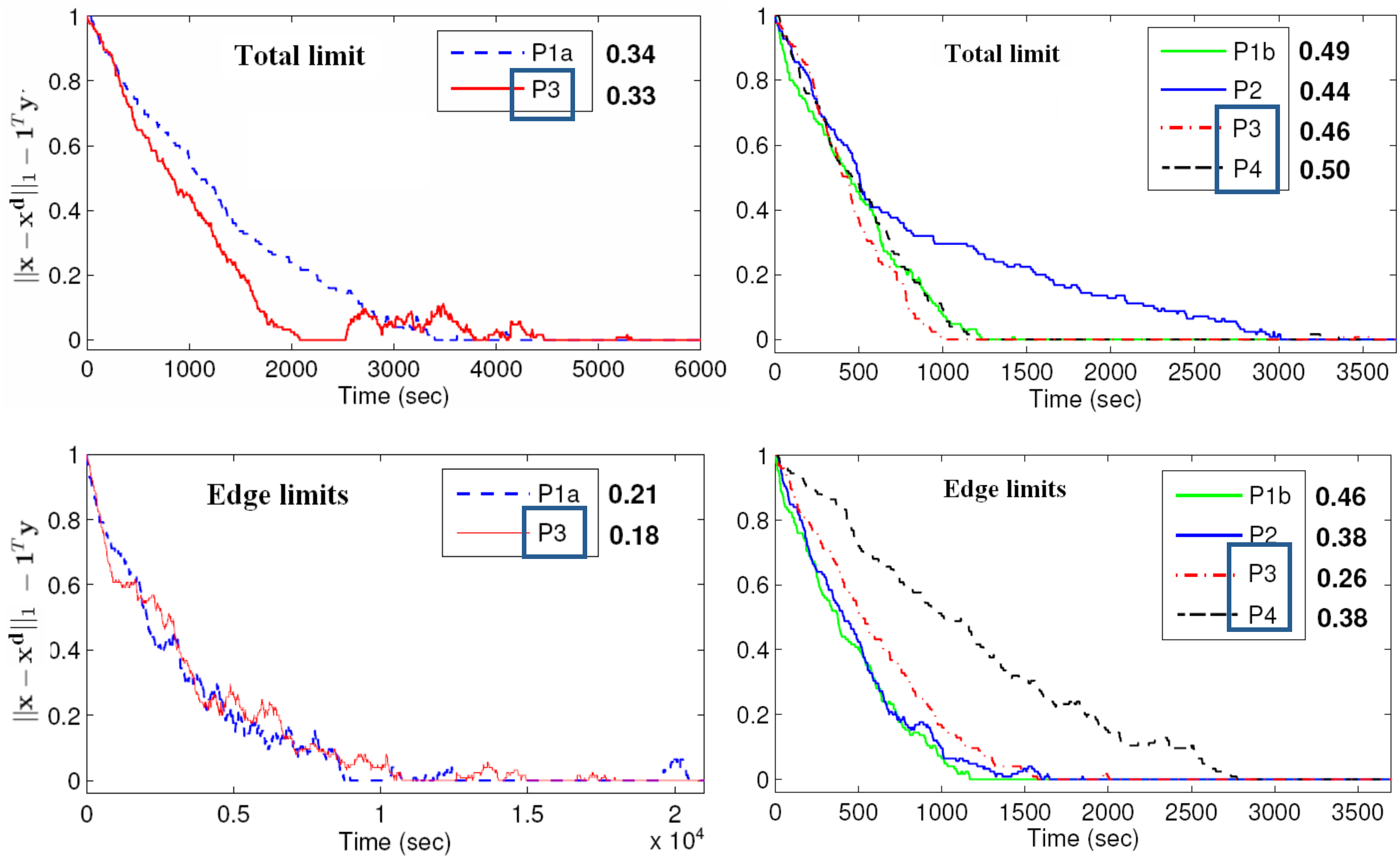
• Tradeoff between convergence rate, equilibrium traffic



• Faster convergence with increased site connectivity



- Limits on edge traffic eliminate advantage of knowing $\mathbf{x}^0$



• Monte Carlo runs are consistently optimal but computationally slow