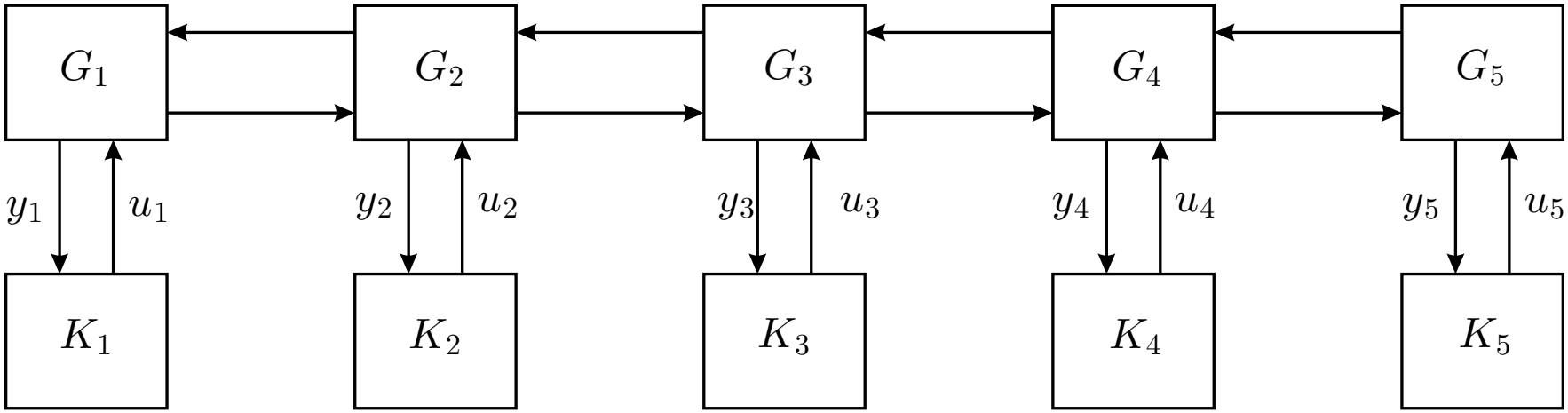
Decentralized Control of Stochastic Systems

Sanjay Lall
Stanford University

CDC-ECC Workshop, December 11, 2005

Decentralized Control



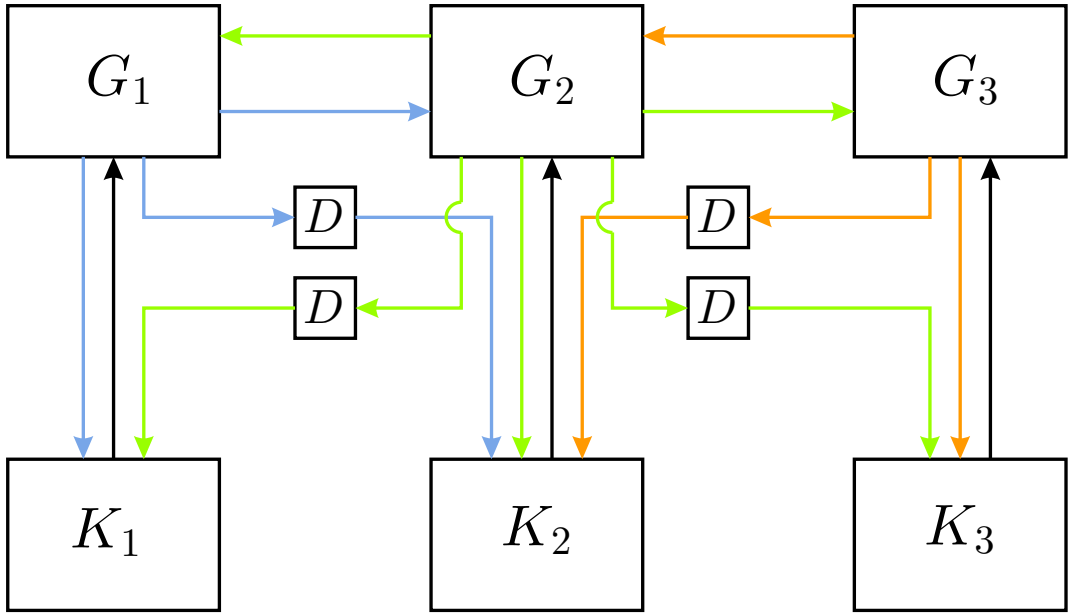
Control design problem is to find K which is block diagonal, whose diagonal blocks are 5 separate controllers.

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} = \begin{bmatrix} K_1 & 0 & 0 & 0 & 0 \\ 0 & K_2 & 0 & 0 & 0 \\ 0 & 0 & K_3 & 0 & 0 \\ 0 & 0 & 0 & K_3 & 0 \\ 0 & 0 & 0 & 0 & K_3 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix}$$

Example: Communicating Controllers

Often we do not want *perfectly decentralized* control; we may achieve better performance by allowing controllers to communicate, both

- to receive measurements from other aircraft,
- and to exchange data with other controllers.



Here, we need to design 3 separate controllers.

Research directions

- Spacetime models
  - Infinite and finite spatial extent
  - Synthesis procedures via SDP
  - Decentralization structure via information passing
- Structured synthesis
  - Decentralized structure specified in advance
  - Quadratic invariance conditions on solvability
  - See talk by Mike Rotkowitz
- Dynamic programming
  - Stochastic finite state systems
  - Relaxation approach

Finite-State Stochastic Systems

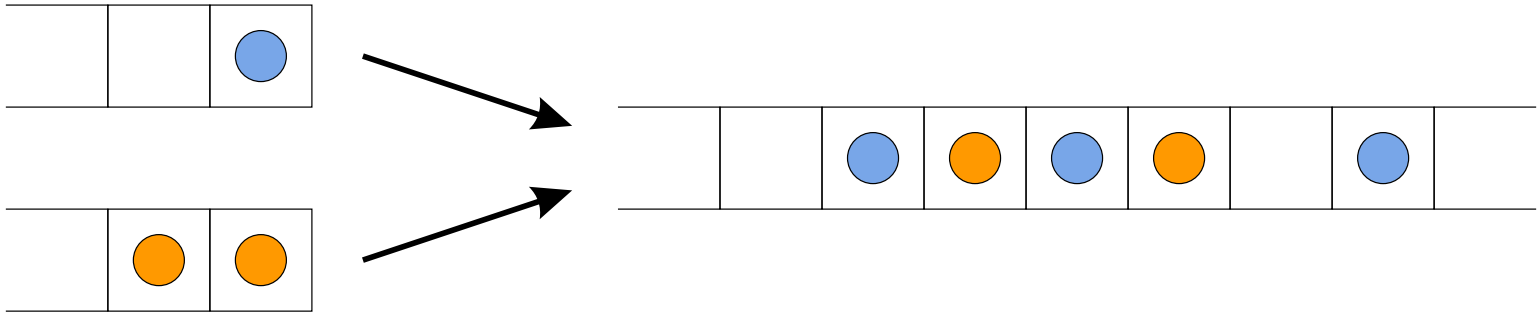
In this talk

- Look at dynamics specified by *Markov Decision Processes*.
- Not linear, but *finite state*
- New *semidefinite programming relaxations*
- Randy Cogill, Mike Rotkowitz, Ben Van Roy

Many applications:

- *Decision problems:* e.g., task allocation problems for vehicles
- *Detection problems:* e.g., physical sensing, communication

Example: Medium-Access Control



- Two *transmitters*, each with a queue that can hold up to 3 packets
- $p_k^a =$ probability that $k-1$ packets arrive at queue a

$$p^1 = \begin{bmatrix} 0.7 & 0.2 & 0.05 & 0.05 \end{bmatrix} \qquad p^2 = \begin{bmatrix} 0.6 & 0.3 & 0.075 & 0.025 \end{bmatrix}$$

- At each time step, each transmitter sees how many packets are in its queue, and sends some of them; then new packets arrive
- Packets are *lost* when queues *overflow*, or when there is a *collision*, i.e., both transmit at the same time

Example: Medium-Access Control

We would like a *control policy* for each queue, i.e., a function mapping

number of packets in the queue $\mapsto$ number of packets sent

- One possible policy; transmit all packets in the queue.

  Causes large packet loss due to collisions.

- The other extreme; wait until the queue is full

  Causes large packet loss due to overflow.

- We'd like to find the policy that minimizes the expected number of packets lost per period.

Centralized Control

- Each transmitter can see how many packets are in the other queue
- In this case, we look for a single policy, mapping

$$\text{pair of queue occupancies} \quad \longmapsto \quad \text{pair of transmission lengths}$$

Decentralized Control

- Each transmitter can only see the number of packets in its own queue
- In this case, we look for *two policies*, each mapping

$$\text{queue occupancy} \quad \longmapsto \quad \text{transmission length}$$

Medium Access: Centralized Solution

The optimal policy is

Transmitter 1 state \ Transmitter 2 state	0	1	2	3
0	(0,0)	(0,1)	(0,2)	(0,3)
1	(1,0)	(1,0)	(0,2)	(0,3)
2	(2,0)	(2,0)	(2,0)	(0,3)
3	(3,0)	(3,0)	(3,0)	(3,0)

- Easy to compute via dynamic programming
- Each transmitter needs to know the state of the other
- We would like a *decentralized* control law

Markov Decision Processes

The above medium-access control problem is an example of a *Markov Decision Process* (MDP)

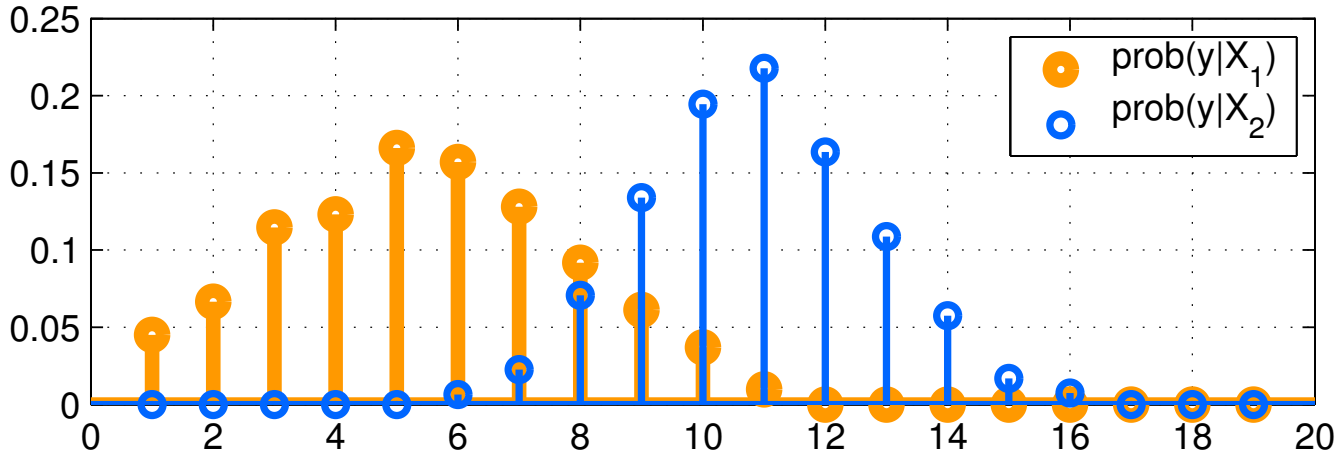
- n states, and m actions, hence m^n possible centralized policies
- However, the centralized problem is solvable by linear programming

Decentralized control

- NP-hard, even with just two policies, even non-dynamic case
- Would like efficient algorithms
- Hence focus on *suboptimal solutions*

Static Problems

- Called classification or hypothesis testing
- e.g., a radar system sends out n pulses, and measures y reflections
- The *cost* depends on the number of false positives/negatives.



$p(y|X_1) =$ prob. of receiving y reflections given no aircraft present

$p(y|X_2) =$ prob. of receiving y reflections given an aircraft present

Static Centralized Formulation

- x is the *state*, with prior pdf $\mathbf{Prob}(x) = p(x)$
- $y = h(x, w)$ is the *measurement*
- w is *sensor noise*

Choosing control action u results in cost $C(u, x)$

Represent sensing by transition matrix $A(y, x) = \mathbf{Prob}(y \,|\, x)$

Problem: find $\gamma : \mathcal{Y} \to \mathcal{U}$ to minimize the expected cost:

$$\sum_{x,y} p(x) A(y, x) C\big(\gamma(y), x\big)$$

Static Decentralized Formulation

- x is the *state*, with prior pdf $\mathbf{Prob}(x) = p(x)$
- $y_i = h_i(x, w_i)$ is the i'th *measurement*
- w_i is *noise* at i'th sensor

Choosing control actions $u_1, \ldots, u_m$ results in cost $C(u_1, \ldots, u_m, x)$

$y = (y_1, \ldots, y_m)$ given by transitions $A(y, x) = \mathbf{Prob}(y \,|\, x)$

Problem: find $\gamma_1, \ldots, \gamma_m$ to minimize the expected cost:

$$\sum_{x,y} p(x) A(y, x) C\big(\gamma_1(y_1), \ldots, \gamma_m(y_m), x\big)$$

Change of variables

Represent the policy by a matrix

$$K_{yu} = \begin{cases} 1 & \text{if } \gamma(y) = u \\ 0 & \text{otherwise} \end{cases}$$

This matrix K satisfies

- $K\mathbf{1} = \mathbf{1}$, that is K is a *stochastic matrix*.
- K is Boolean

Then the expected cost is

$$\sum_{x,y} p_x A_{yx} C\big(\gamma(y), x\big) = \sum_{x,y,u} p_x A_{yx} C_{ux} K_{yu}$$

Optimization

Let $W_{yu} = \sum_x C_{ux} A_{yx} p_x$, the cost of decision u when y is measured.

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{y,u} W_{yu} K_{yu} \\ \text{subject to} & K\mathbf{1} = \mathbf{1} \\ & K_{yu} \in \{0,1\} \quad \text{for all } y, u \end{array}$$

- nm variables K_{ij}, with both *linear* and *Boolean* constraints
- Just m easy problems; pick $\gamma(y) = \arg\min_u W_{yu}$

Linear Programming Relaxation

Another solution: *relax* the Boolean constraints:

- For estimation: interpret K_{yu} as belief in the estimate u
- For control: we have a *mixed policy* $K_{yu} = \mathbf{Prob}(u \,|\, y)$

Relaxation gives the *linear program:*

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{y,u} W_{yu} K_{yu} \\ \text{subject to} & K\mathbf{1} = \mathbf{1} \\ & K \geq 0 \end{array}$$

It is easy to show that this gives the optimal cost and rounding gives the optimal controller.

Decentralization Constraints

Decentralization imposes constraints on $K(y_1, \ldots, y_m, u_1, \ldots, u_m)$.

Measurement constraint: controller i can only measure measurement i

$$\mathbf{Prob}(u_i \,|\, y) = \mathbf{Prob}(u_i \,|\, y_i)$$

Independence constraint: for mixed policies, controllers i and j use independent sources of randomness to generate u_i and u_j

$$\mathbf{Prob}(u \,|\, y) = \prod_{i=1}^{m} \mathbf{Prob}(u_i \,|\, y_i)$$

Decentralization Constraints

We call a controller K *decentralized* if

$$K(y_1, \ldots, y_m, u_1, \ldots, u_m) = K^1(y_1, u_1) \ldots K^m(y_m, u_m)$$

for some $K^1, \ldots, K^m$

If K is a deterministic controller $y = \gamma(u)$, this is equivalent to

$$\gamma(u_1, u_2) = \big(\gamma_1(u_1), \gamma_2(u_2)\big)$$

The Decentralized Static Control Problem

$$
\begin{array}{ll}
\text{minimize} & \displaystyle\sum_{y,u} W_{y_1 y_2 u_1 u_2} K_{y_1 y_2 u_1 u_2} \\
\text{subject to} & K_{y_1 y_2 u_1 u_2} = K^1_{y_1 u_1} K^2_{y_2 u_2} \\
& K^i \geq 0 \\
& K^i \mathbf{1} = \mathbf{1} \\
& K^i_{ab} \in \{0, 1\} \text{ for all } a, b
\end{array}
$$

- This problem is NP-hard
- In addition to linear constraints, we have *bilinear* constraints

Boolean Constraints

One can show the following facts

- Removing the Boolean constraints does not change the optimal value

  i.e., we can *simplify* the constraint set

- Then there exists an optimal solution which is Boolean

  i.e., using a *mixed policy* does not achieve better performance than a deterministic policy

Relaxing the Constraints

Now we would like to solve

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{y,u} W_{y_1 y_2 u_1 u_2} K_{y_1 y_2 u_1 u_2} \\ \text{subject to} & K_{y_1 y_2 u_1 u_2} = K^1_{y_1 u_1} K^2_{y_2 u_2} \\ & K^i \geq 0 \\ & K^i \mathbf{1} = \mathbf{1} \end{array}$$

- Still NP-hard, because it would solve the original problem.
- Removing the bilinear constraint gives the centralized problem. This is a trivial lower bound on the achievable cost

Valid Constraints

For any decentralized controller, we know

$$\sum_{u_1} K_{y_1 y_2 u_1 u_2} = K^2_{y_2 u_2}$$

- The interpretation is that, given measurements y_1 and y_2, averaging over the decisions of controller 1 does not affect the distribution of the decisions of controller 2

- Easy to see, since K^1 is a stochastic matrix, and

$$K_{y_1 y_2 u_1 u_2} = K^1_{y_1 u_1} K^2_{y_2 u_2}$$

Valid Constraints

- Adding these new constraints leaves the optimization unchanged.
- But it *changes the relaxation*. Dropping the bilinear constraint gives a tighter lower bound than the centralized problem.

$$\begin{aligned}
\text{minimize} \quad & \sum_{y,u} W_{y_1 y_2 u_1 u_2} K_{y_1 y_2 u_1 u_2} \\
\text{subject to} \quad & K_{y_1 y_2 u_1 u_2} = K^1_{y_1 u_1} K^2_{y_2 u_2} \\
& \left.\begin{aligned} \sum_{u_1} K_{y_1 y_2 u_1 u_2} &= K^2_{y_2 u_2} \\ \sum_{u_2} K_{y_1 y_2 u_1 u_2} &= K^1_{y_1 u_1} \end{aligned}\right\} \text{new valid constraints} \\
& K^i \geq 0, \;\; K^i \mathbf{1} = \mathbf{1}
\end{aligned}$$

Synthesis Procedure

1. Solve the linear program

$$\begin{aligned} \text{minimize} \quad & \sum_{y,u} W_{y_1 y_2 u_1 u_2} K_{y_1 y_2 u_1 u_2} \\ \text{subject to} \quad & \sum_{u_1} K_{y_1 y_2 u_1 u_2} = K^2_{y_2 u_2} \\ & \sum_{u_2} K_{y_1 y_2 u_1 u_2} = K^1_{y_1 u_1} \\ & K^i \geq 0, \;\; K^i \mathbf{1} = \mathbf{1} \end{aligned}$$

   The optimal value gives a lower bound on the achievable cost

2. If $K_{y_1 y_2 u_1 u_2} = K^1_{y_1 u_1} K^2_{y_2 u_2}$ then the solution is optimal.
   If not, use K^1 and K^2, as *approximate* solutions.

Lifting

Lifting is a general approach for constructing *primal relaxations*; the idea is

- Introduce new variables Y which are polynomial in x
  This embeds the problem in a *higher dimensional* space
- Write *valid constraints* in the new variables
- The feasible set of the original problem is the *projection* of the lifted feasible set

Example: Minimizing a Polynomial

We'd like to find the minimum of $f = \sum_{k=0}^{6} a_k x^k$

Pick new variables $Y = g(x)$ where

$$g(x) = \begin{bmatrix} 1 & x & x^2 & x^3 \\ x & x^2 & x^3 & x^4 \\ x^2 & x^3 & x^4 & x^5 \\ x^3 & x^4 & x^5 & x^6 \end{bmatrix} \qquad \text{and} \qquad C = \begin{bmatrix} a_0 & \frac{a_1}{2} & \frac{a_2}{2} & \frac{a_3}{2} \\ & 0 & 0 & \frac{a_4}{2} \\ & & 0 & \frac{a_5}{2} \\ & & & a_6 \end{bmatrix}$$

Then an equivalent problem is

$$\begin{array}{ll} \text{minimize} & \mathbf{trace}\, CY \\ \text{subject to} & Y \succeq 0 \\ & Y_{11} = 1 \quad Y_{24} = Y_{33} \quad Y_{22} = Y_{13} \quad Y_{14} = Y_{23} \\ & Y = g(x) \end{array}$$

Dropping the constraint $Y = g(x)$ gives an *SDP relaxation* of the problem

The Dual SDP Relaxation

The SDP relaxation has a dual, which is also an SDP.

Example

Suppose $f = x^6 + 4x^2 + 1$, then the SDP dual relaxation is

$$\begin{array}{ll} \text{maximize} & t \\ \text{subject to} & \begin{bmatrix} 1-t & 0 & 2+\lambda_2 & -\lambda_3 \\ 0 & -2\lambda_2 & \lambda_3 & \lambda_1 \\ 2+\lambda_2 & \lambda_3 & -2\lambda_1 & 0 \\ -\lambda_3 & \lambda_1 & 0 & 1 \end{bmatrix} \succeq 0 \end{array}$$

this is exactly the condition that $f - t$ be *sum of squares*

Lifting for General Polynomial Programs

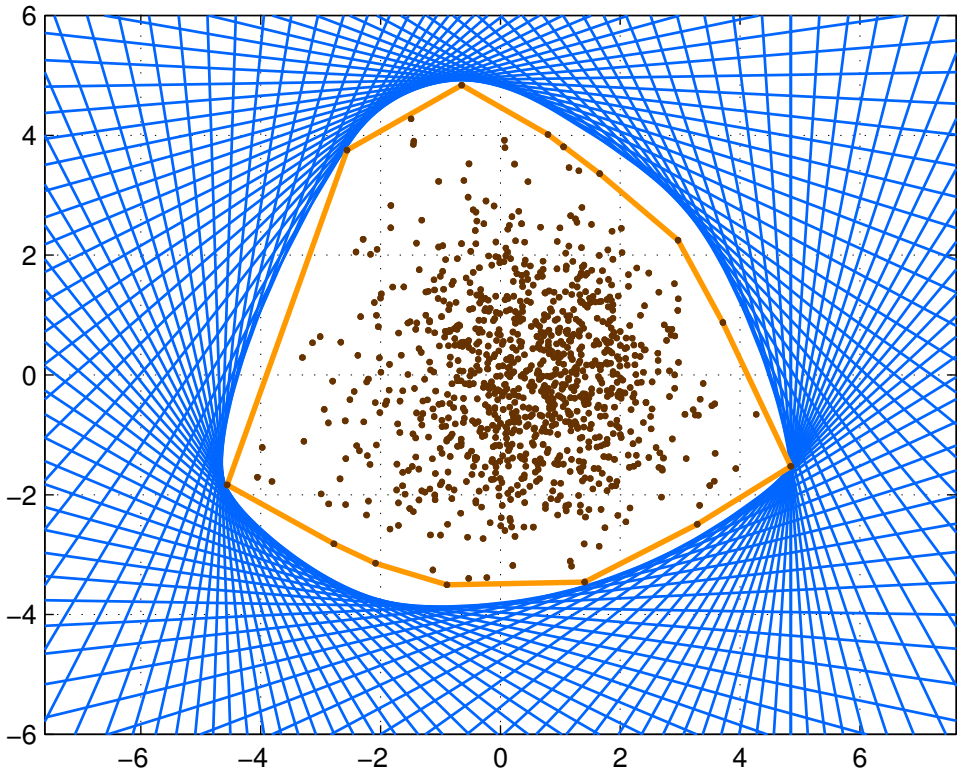
- When minimizing a polynomial, lifting gives an SDP relaxation of whose dual is an SOS condition
- When solving a general polynomial program with multiple constraints, there is a similar lifting
- This gives an SDP, whose feasible set is a relaxation of the feasible set of the original problem
- The corresponding dual SDP is a *Positivstellensatz refutation*
- Solving the dual *certifies* a lower bound on the original problem

Example: The Cut Polytope

The feasible set of the MAXCUT problem is

$$C = \left\{ X \in \mathbb{S}^n \mid X = vv^T,\ v \in \{-1, 1\}^n \right\}$$

A simple relaxation gives the outer approximation to its convex hull.
Here $n = 11$; the set has affine dimension 55; a projection is shown below



Example

Suppose the sample space is $\Omega = \{f_1, f_2, f_3, f_4\} \times \{g_1, g_2, g_3, g_4\}$

The unnormalized probabilities of $(f, g) \in \Omega$ are given by

	g_1	g_2	g_3	g_4
f_1	1	6	2	0
f_2	0	1	2	4
f_3	6	2	0	1
f_4	4	0	1	2

- Player 1 measures f, i.e., Y^1 is the set of horizontal strips and would like to estimate g, i.e, X^1 is the set of vertical strips
- Player 2 measures g and would like to estimate f

Example

Objective: maximize the *expected number of correct estimates*

	g_1	g_2	g_3	g_4
f_1	1	6	2	0
f_2	0	1	2	4
f_3	6	2	0	1
f_4	4	0	1	2

Optimal decision rules are

$$K^1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Y^1	f_1	f_2	f_3	f_4
X^1_{est}	g_2	g_4	g_1	g_2

$$K^2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Y^2	g_1	g_2	g_3	g_4
X^2_{est}	f_3	f_1	f_2	f_2

- The optimal is 1.1875
- These are simply the *maximum a-posteriori probability* classifiers

Example

Objective: maximize the probability that *both estimates* are correct

	g_1	g_2	g_3	g_4
f_1	1	6	2	0
f_2	0	1	2	4
f_3	6	2	0	1
f_4	4	0	1	2

Optimal decision rules are

Y^1	f_1	f_2	f_3	f_4
X^1_{est}	g_2	g_4	g_1	g_3

Y^2	g_1	g_2	g_3	g_4
X^2_{est}	f_3	f_1	f_4	f_2

- The relaxation of the lifted problem is tight
- The optimal probability that both estimates are correct is 0.5313
- MAP estimates are not optimal; they achieve 0.5

Example

Objective: maximize the probability that *at least one estimate* is correct

	g_1	g_2	g_3	g_4
f_1	1	6	2	0
f_2	0	1	2	4
f_3	6	2	0	1
f_4	4	0	1	2

- The relaxation of the lifted problem is *not tight*; it gives upper bound of 0.875
- The following decision rules (constructed by projection) achieve 0.8438

Y^1	f_1	f_2	f_3	f_4
X^1_{est}	g_2	g_4	g_1	g_1

Y^2	g_1	g_2	g_3	g_4
X^2_{est}	f_1	f_3	f_1	f_4

- MAP estimates achieve 0.6875

Heuristic 1: MAP

	g_1	g_2	g_3	g_4
f_1	$\frac{1}{28}$	0	0	0
f_2	$\frac{1}{14}$	$\frac{3}{28}$	0	0
f_3	0	$\frac{1}{7}$	$\frac{5}{28}$	0
f_4	0	0	$\frac{3}{14}$	$\frac{1}{4}$

The optimal probability that both estimates are correct is $\frac{4}{7}$, achieved by

Player 1 sees:	f_1	f_2	f_3	f_4
Player 1 guesses:	g_1	g_2	g_3	g_4

Player 2 sees:	g_1	g_2	g_3	g_4
Player 2 guesses:	f_1	f_2	f_3	f_4

If both just guess most likely outcome, achieved probability is $\frac{1}{4}$.

Heuristic 2: Person by Person Optimality

1. Choose arbitrary initial policies γ_1 and γ_2.
2. Hold γ_2 fixed. Choose best γ_1 given γ_2.
3. Hold new γ_1 fixed. Choose best γ_2 given this γ_1.
4. Continue until no more improvement.

	$u_1 = 0$	$u_1 = 1$
$u_2 = 0$	1000	1001
$u_2 = 1$	1001	1

PBPO value is 1000 times the optimal! Almost as bad as the worst policy.

Lift and Project Performance Guarantee

- We would like to maximize *reward*; all rewards are nonnegative.
- Use the above LP relaxation together with a projection procedure
- The computed policies always achieve a a reward of at least

$$\frac{\text{optimal reward}}{\min\{|U_1|, |U_2|\}}$$

This gives an *upper bound*, a *feasible policy*, and a *performance guarantee*

Markov Decision Processes

We will now consider a *Markov Decision Process* where

- $X_i(t)$ is the event that the system is in state i at time t
- $A_j(t)$ is the event that action j is taken at time t

We assume for simplicity that for every stationary policy the chain is irreducible and aperiodic

- *Transition probabilities:* $A_{ijk} = \mathbf{Prob}(X_i(t+1) \,|\, X_j(t) \cap A_k(t))$
- *Mixed policy:* $K_{jk} = \mathbf{Prob}(X_j(t) \cap A_k(t))$
- *Cost function:* $W_{jk} =$ cost of action k in state j

Markov Decision Processes

The centralized solution

$$\begin{aligned}
\text{minimize} \quad & \sum_{j,k} W_{jk} K_{jk} \\
\text{subject to} \quad & \sum_{r} K_{ir} = \sum_{j,k} A_{ijk} K_{jk} \\
& K \geq 0 \\
& \sum_{j,k} F_{jk} = 1
\end{aligned}$$

Decentralized Markov Decision Processes

- Two sets of states $X^p = \{X_1^p, \ldots, X_n^p\}$
- Two transition matrices $A_{ijk}^p = \mathbf{Prob}(X_i^p(t+1) \,|\, X_j^p(t) \cap A_k^p(t))$
- Two controllers $K_{jk}^p = \mathbf{Prob}(X_j^p(t) \cap A_k^p(t))$
- Cost function $W_{j_1 j_2 k_1 k_2}$ = cost of actions k_1, k_2 in states j_1, j_2

Decentralized Markov Decision Processes

$$\text{minimize} \quad \sum_{j_1,j_1,k_1,k_2} W_{j_1 j_1 k_1 k_2} K_{j_1 j_2 k_1 k_2}$$

$$\text{subject to} \quad K_{j_1 j_2 k_1 k_2} = K^1_{j_1 k_1} K^2_{j_2 k_2}$$

$$\sum_r K^p_{ir} = \sum_{j,k} A^p_{ijk} K^p_{jk} \tag{1}$$

$$K^p \geq 0 \tag{2}$$

$$\sum_{j,k} K^p_{jk} = 1 \tag{3}$$

- Each of constraints (1)–(3) can be multiplied by K^{3-p} to construct a valid constraint in lifted variables K
- The resulting linear program gives a lower bound on the optimal cost

Exact Solution

If the solution K to the lifted linear program has the form

$$K_{j_1 j_2 k_1 k_2} = K^1_{j_1 k_1} K^2_{j_2 k_2}$$

then the controller is an optimal decentralized controller.

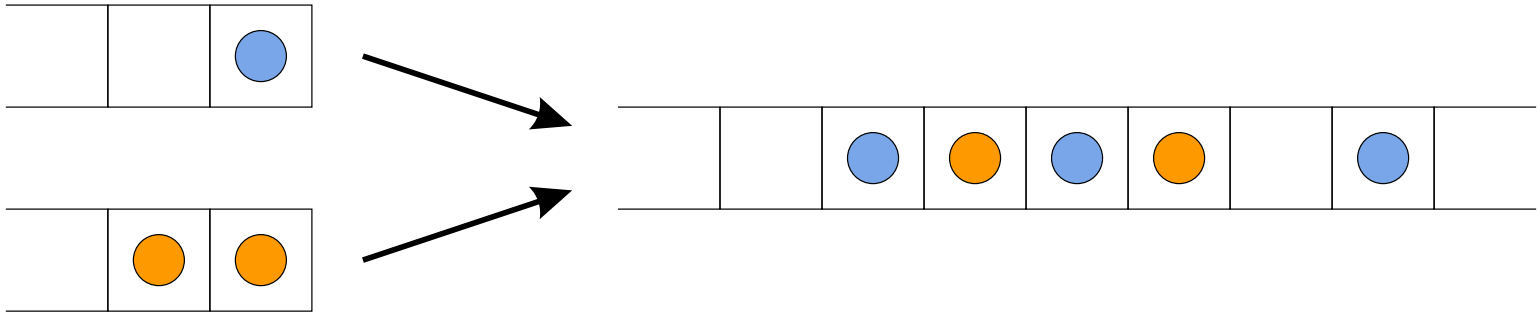
This corresponds to the usual rank conditions in e.g., MAXCUT.

Projection

If not, we need to project the solution

- K defines a pdf on $X^1 \times X^2 \times U^1 \times U^2$
- We project by constructing the marginal pdf on $X^p \times U^p$

Example: Medium-Access Control



- Two *transmitters*, each with a queue that can hold up to 3 packets
- $p_k^a =$ probability that $k-1$ packets arrive at queue a

$$p^1 = \begin{bmatrix} 0.7 & 0.2 & 0.05 & 0.05 \end{bmatrix} \qquad p^2 = \begin{bmatrix} 0.6 & 0.3 & 0.075 & 0.025 \end{bmatrix}$$

- At each time step, each transmitter sees how many packets are in its queue, and sends some of them; then new packets arrive
- Packets are *lost* when queues *overflow*, or when there is a *collision*, i.e., both transmit at the same time

Example: Medium Access

This is a Decentralized Markov Decision Process, where

- Each MDP has 4 states; the no. of packets in the queue
- Each MDP has 4 actions; transmit $0, 1, 2, 3$ packets
- State transitions are determined by arrival probabilities and actions
- Cost is total number of packets lost;

  Each queue loses all packets sent if there is a collision

  Each queue loses packets due to overflows

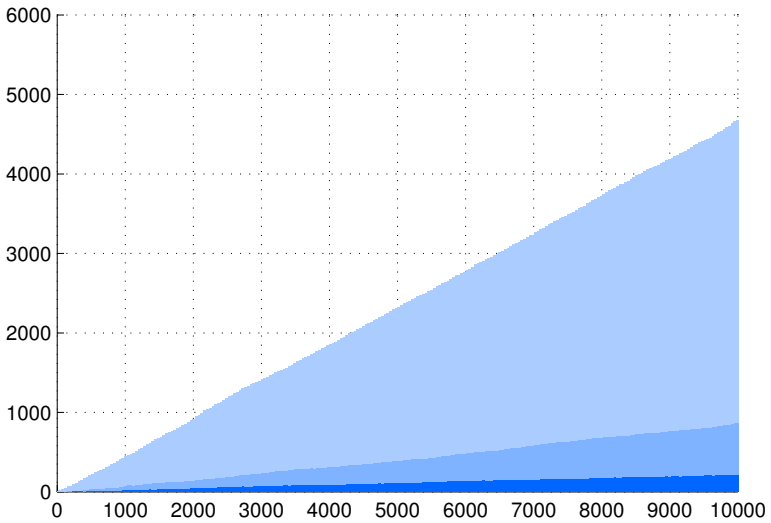
Example: Medium Access

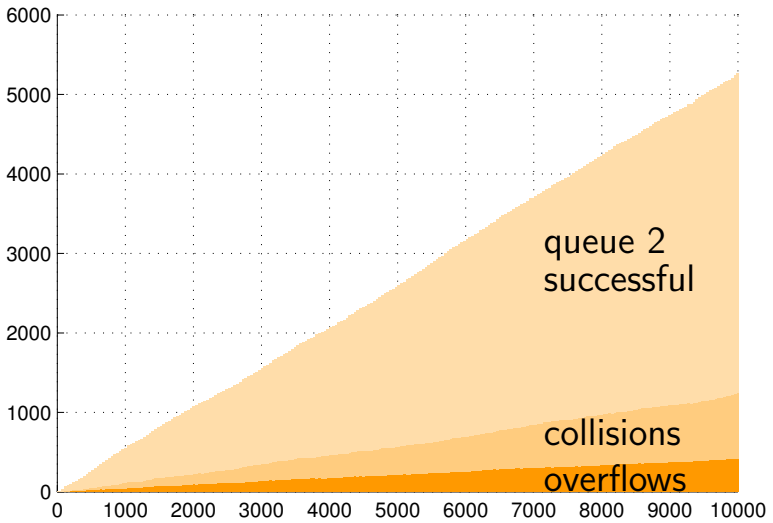
Optimal policies for each player are

queue occupancy	0	1	2	3
number sent	0	0	2	3

queue occupancy	0	1	2	3
number sent	0	0	0	3

- Expected number of packets lost per period is 0.2202
- The policy *always transmit* loses 0.3375 per period





Summary

- Decentralized control problems are inherently computationally hard
- We have developed
  - Efficient algorithms to construct suboptimal solutions
  - Upper bounds on achievable performance
  - Guaranteed performance ratio
- Much more
  - *topology independent controllers*
  - algorithms that *exploit structure*
  - *peer-to-peer* techniques
  - other applications: robotics, sensor networks