# Sum of Squares and Decentralized Stochastic Decision Problems

**Sanjay Lall and Randy Cogill**

Stanford University

MSRI Program in Topological Aspects of Real Algebraic Geometry

Workshop on Algorithmic, Combinatorial and Applicable Real Algebraic Geometry
April 16, 2004

# Acknowledgments

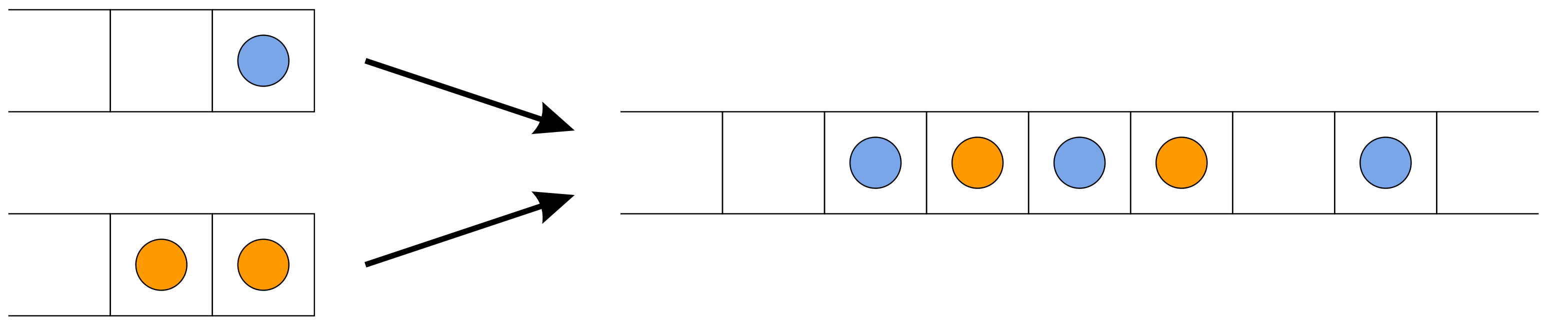

- Work by Randy Cogill, Electrical Engineering, Stanford

- Based on two course projects:

- *An LP Relaxation for Decentralized Decision Problems*
  for *Optimization Projects* by Stephen Boyd

- *A Relaxation for Decentralized Control of Markov Decision Processes*
  for *Advanced Topics in Computation for Control* by S. Lall

# 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

$$\text{number of packets in the queue} \quad \mapsto \quad \text{number of packets sent}$$

- 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

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

## 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

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

# 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

The decentralized problem

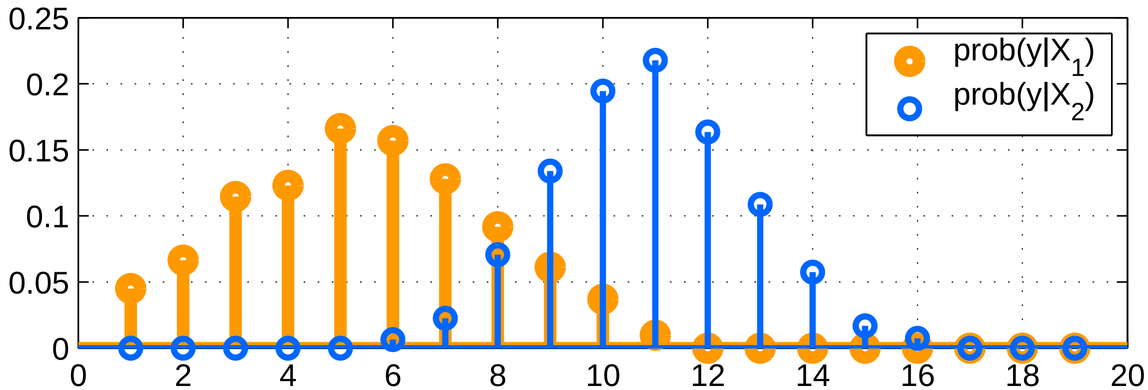
- NP-hard, even with just two policies
- The set of policies achieving a given cost is a *real variety*
- We can use the ideas of optimization of semialgebraic sets to find performance bounds and suboptimal policies

# Classification

Even for *non-dynamic* problems, often decentralized problems are *much harder* than centralized ones.

For example, the *classification problem*; A radar system sends out $n$ pulses, and receives $y$ reflections, where $0 \le y \le n$.

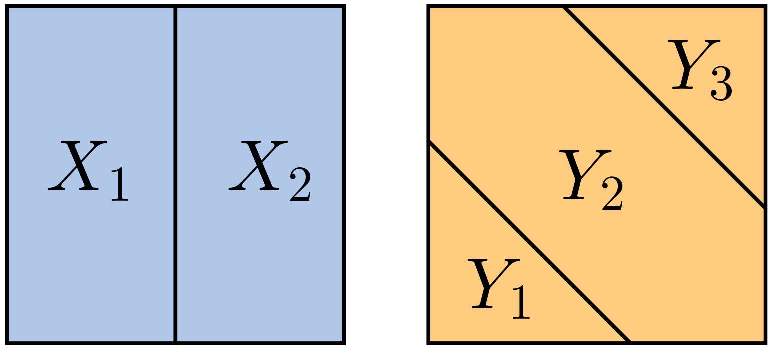
$$p(y|X_1) = \text{prob. of receiving } y \text{ reflections given no aircraft present}$$
$$p(y|X_2) = \text{prob. of receiving } y \text{ reflections given an aircraft present}$$




We measure $y$ reflections, and decide if an aircraft is present. The *cost* depends on the number of false positives/negatives.

# Centralized Classification

- $X = \{X_1, \ldots, X_n\}$ are events that partition $\Omega$, called *hypotheses*
- $Y = \{Y_1, \ldots, Y_m\}$ are events that partition $\Omega$, called *observations*




We know which $Y_i$ occurred, and would like to pick which $X_j$ occurred

i.e., we would like a *policy* $\gamma : Y \to X$, which we specify via a matrix

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

# Error Probabilities

We have for all $x \in X, y \in Y$

- *transition probabilities* $A_{yx} = \mathbf{Prob}(y \mid x)$
- *prior probabilities* $p_x = \mathbf{Prob}(x)$

The *error probability* $E_{zx}$ of $z \in X$ being estimated and $x$ occurring is

$$\begin{aligned} E_{zx} &= \sum_{y \in Y} K_{yx} \, \mathbf{Prob}(y \mid x) \, \mathbf{Prob}(x) \\ &= \sum_{y \in Y} K_{yx} A_{yx} p_x \end{aligned}$$

# Minimum Expected Cost

Assign cost $C_{zx}$ for estimating $z$ when $x$ occurs.

Then we minimize the expected cost

$$\begin{aligned} \text{minimize} \quad & \sum_{x,y,z} C_{zx} A_{yx} p_x K_{yz} \\ \text{subject to} \quad & K\mathbf{1} = \mathbf{1} \\ & K_{yz} \in \{0, 1\} \quad \text{for all } y, x \end{aligned}$$

- An optimization in $nm$ variables $K_{ij}$, with both *linear* and *Boolean* constraints

# Minimum Expected Cost

Let $W_{yz} = \sum_x C_{zx} A_{yx} p_x$ = the cost of estimating $z$ when $y$ occurs.

Then the above problem is

$$
\begin{aligned}
\text{minimize} \quad & \sum_{y,z} W_{yz} K_{yz} \\
\text{subject to} \quad & K \geq 0 \\
& K\mathbf{1} = \mathbf{1} \\
& K_{yz} \in \{0, 1\} \quad \text{for all } y, z
\end{aligned}
$$

- Just $n$ easy problems; pick $\gamma(y) = \arg\max_x W_{yx}$
- Relaxing the Boolean constraints gives a *linear program* whose optimal value is the minimum expected cost

# Decentralized Classification

We have for each player $i = 1, 2$

- observations $Y^i = \{ Y^i_1, \ldots, Y^i_m \}$
- hypotheses $X^i = \{ X^i_1, \ldots, X^i_m \}$

All four of these sets partition $\Omega$.

The set of possible observations is therefore $Y = Y^1 \times Y^2$

# Notation

- $y = (y_1, y_2)$ occurs means $y_1 \cap y_2$ occurs
- We will use $y_1$ to mean both the event $y_1 \in Y^1$ as well as the integer $y_1 \in \{1, \ldots, m\}$ in the natural way

# Joint Cost Function

The cost is $C_{zx}$ for estimating $z \in X$ and $x \in X$ occurs.

i.e, the cost is $C_{z_1 z_2 x_1 x_2}$ when

- player 1 estimates $z_1 \in X^1$ and $x_1 \in X^1$ occurs
- player 2 estimates $z_2 \in X^2$ and $x_2 \in X^2$ occurs

# Decentralization Constraints

We need estimator $\gamma : (y_1, y_2) \mapsto (x_1, x_2)$ to be *decentralized*, i.e.,

$$\gamma : (y_1, y_2) \mapsto \big(\gamma^1(x_1),\ \gamma^2(x_2)\big)$$

So we have

$$\begin{aligned}
K_{yx} &= \begin{cases} 1 & \text{if } \gamma(y) = x \\ 0 & \text{otherwise} \end{cases} \\
&= \begin{cases} 1 & \text{if } \gamma^1(y_1) = x_1 \text{ and } \gamma^2(y_2) = x_2 \\ 0 & \text{otherwise} \end{cases} \\
&= \begin{cases} 1 & \text{if } K^1_{y_1 x_1} = 1 \text{ and } K^2_{y_2 x_2} = 1 \\ 0 & \text{otherwise} \end{cases}
\end{aligned}$$

i.e., $K$ is decentralized iff $K_{yx}$ factorizes as $K_{yx} = K^1_{y_1 x_1} K^2_{y_2 x_2}$

# Minimum Expected Cost

To find the decentralized estimator with minimum expected cost

$$
\begin{array}{ll}
\text{minimize} & \displaystyle\sum_{y,z} W_{yz} K_{yz} \\
\text{subject to} & K_{yx} = K^1_{y_1 x_1} K^2_{y_2 x_2} \\
& K^i \geq 0 \\
& K^i \mathbf{1} = \mathbf{1} \\
& K^i_{yz} \in \{0, 1\} \quad \text{for all } y, z
\end{array}
$$

- This is a *polynomial program*
- In addition to the Boolean and linear constraints, we have *bilinear* constraints

# Boolean Constraints

Consider the above problem, but dropping the Boolean constraints.

$$\begin{aligned} \text{minimize} \quad & \sum_{y,z} W_{yz} K_{yz} \\ \text{subject to} \quad & K_{yx} = K^1_{y_1 x_1} K^2_{y_2 x_2} \\ & K^i \geq 0 \\ & K^i \mathbf{1} = \mathbf{1} \end{aligned}$$

- If there exists a non-Boolean solution, then there exists a Boolean solution with the *same objective value*
- Because if we fix $K^1$ and optimize $K^2$, we can find a solution with $K^2$ Boolean which does not increase the cost. Similarly for $K^1$.

# 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 inequalities* 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

# Lifting for Decentralized Estimation

$$\begin{aligned} \text{minimize} \quad & \sum_{y,z} W_{yz} K_{yz} \\ \text{subject to} \quad & K_{yx} = K^1_{y_1 x_1} K^2_{y_2 x_2} && \textit{lifted variables} \\ & \left.\begin{aligned} \sum_{x_1} K_{yx} = K^2_{y_2 x_2} \\ \sum_{x_2} K_{yx} = K^1_{y_1 x_1} \end{aligned}\right\} && \textit{new valid inequalities} \\ & K^i \geq 0, \ \ K^i \mathbf{1} = \mathbf{1} \end{aligned}$$

- Relax the constraint $K_{yx} = K^1_{y_1 x_1} K^2_{y_2 x_2}$.
- The resulting linear program gives a lower bound on the optimal cost

# Lifting for Decentralized Estimation

We solve

$$\begin{aligned} \text{minimize} \quad & \sum_{y,z} W_{yz} K_{yz} \\ \text{subject to} \quad & \sum_{x_1} K_{yx} = K^2_{y_2 x_2} \\ & \sum_{x_2} K_{yx} = K^1_{y_1 x_1} \\ & K^i \geq 0, \ K^i \mathbf{1} = \mathbf{1} \end{aligned}$$

- If the optimal solution satisfies $K_{yx} = K^1_{y_1 x_1} K^2_{y_2 x_2}$ then it is the optimal decentralized classifier
- If not, then we need a method for *projection*

# 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

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

| $Y^1$ | $f_1$ | $f_2$ | $f_3$ | $f_4$ |
|---|---|---|---|---|
| $X^1_{\text{est}}$ | $g_2$ | $g_4$ | $g_1$ | $g_2$ |

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

| $Y^2$ | $g_1$ | $g_2$ | $g_3$ | $g_4$ |
|---|---|---|---|---|
| $X^2_{\text{est}}$ | $f_3$ | $f_1$ | $f_2$ | $f_2$ |

- 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

| $Y^1$ | $f_1$ | $f_2$ | $f_3$ | $f_4$ |
|---|---|---|---|---|
| $X^1_{\text{est}}$ | $g_2$ | $g_4$ | $g_1$ | $g_3$ |

| $Y^2$ | $g_1$ | $g_2$ | $g_3$ | $g_4$ |
|---|---|---|---|---|
| $X^2_{\text{est}}$ | $f_3$ | $f_1$ | $f_4$ | $f_2$ |

- 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 $0.8438$

| $Y^1$ | $f_1$ | $f_2$ | $f_3$ | $f_4$ |
|---|---|---|---|---|
| $X^1_{\text{est}}$ | $g_2$ | $g_4$ | $g_1$ | $g_1$ |

| $Y^2$ | $g_1$ | $g_2$ | $g_3$ | $g_4$ |
|---|---|---|---|---|
| $X^2_{\text{est}}$ | $f_1$ | $f_3$ | $f_1$ | $f_4$ |

- MAP estimates achieve $0.6875$

# 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:* $A_{ijk} = \mathbf{Prob}(X_i(t+1) \mid X_j(t) \cap A_k(t))$
- *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

We would like to solve

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

# Decentralized Markov Decision Processes

- Two sets of states $X^p = \{X_1^p, \ldots, X_n^p\}$
- Two transition matrices $A_{ijk}^p = \mathbf{Prob}(X_i^p(t+1) \mid X_j^p(t) \cap A_k^p(t))$
- 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

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

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

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

$$K^p \geq 0 \qquad (2)$$

$$\sum_{j,k} K^p_{jk} = 1 \qquad (3)$$

- 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.

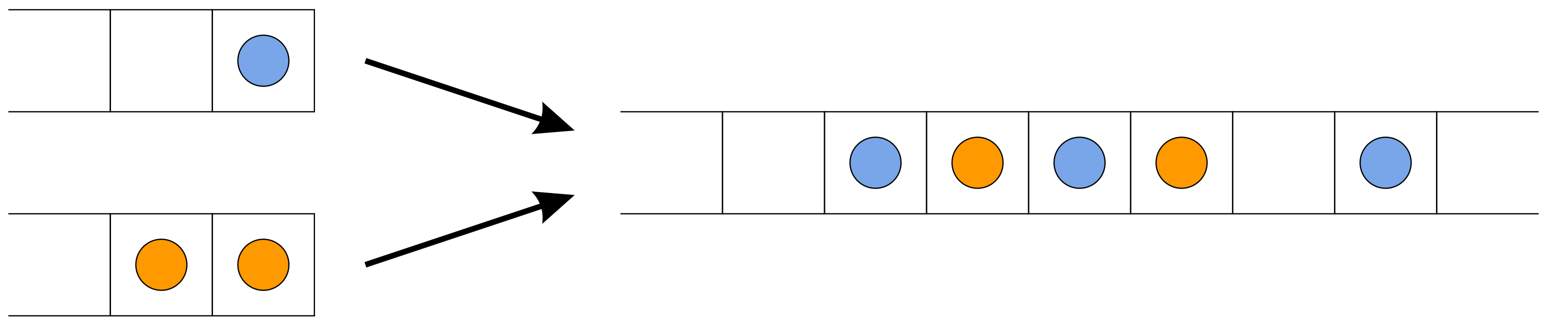
This corresponds to the usual rank conditions in e.g., MAXCUT.

## 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