Quantitative Verification and Synthesis using MDPs with Ratio Objectives

Christian von Essen
Co-supervised by Saddek Bensalem
and Barbara Jobstmann

UJF Grenoble/Verimag

03.04.2011

Outline

Quantitative Verification and Synthesis

- Verification: Does the system fulfill the specification?

Quantitative Verification and Synthesis

- Verification: Does the system fulfill the specification?
- Synthesis (1): Give me a system fulfilling the specification

Quantitative Verification and Synthesis

- Verification: Does the system fulfill the specification?
- Synthesis (1): Give me a system fulfilling the specification
- Synthesis (2): Give me the *best* system fulfilling the specification

Controlling Pumps

Controlling Pumps

What to Control? (Environment)

- Modes: Off/Slow/Fast

What to Control? (Environment)

- Modes: Off/Slow/Fast
- May break down

What to Control? (Environment)

- Modes: Off/Slow/Fast
- May break down
- Repair pump (or not)

Chance (Environment)

- Pumps behave randomly

Chance (Environment)

- Pumps behave randomly
- We model chance with probabilities

Chance (Environment)

- Pumps behave randomly
- We model chance with probabilities
- Example: Fast pumps wear out sooner

How to Evaluate? (Specification)

- Waterflow

How to Evaluate? (Specification)

- Waterflow
- Repair costs

How to Evaluate? (Specification)

- Waterflow
- Repair costs
- Repair bulk discount

How to Evaluate? (Specification)

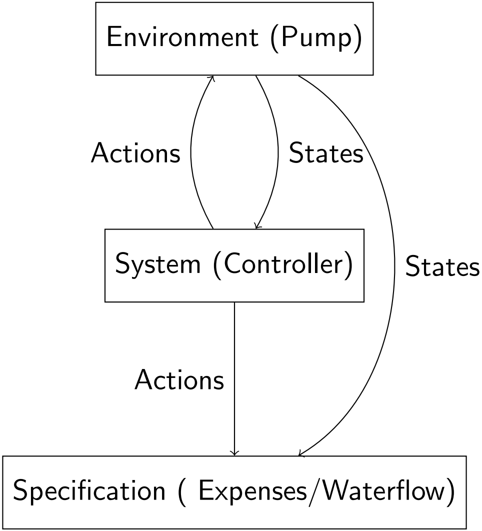
- Waterflow
- Repair costs
- Repair bulk discount
- No pump running $\Longrightarrow$ penalty

How to Evaluate? (Specification)

- Waterflow
- Repair costs
- Repair bulk discount
- No pump running $\Longrightarrow$ penalty

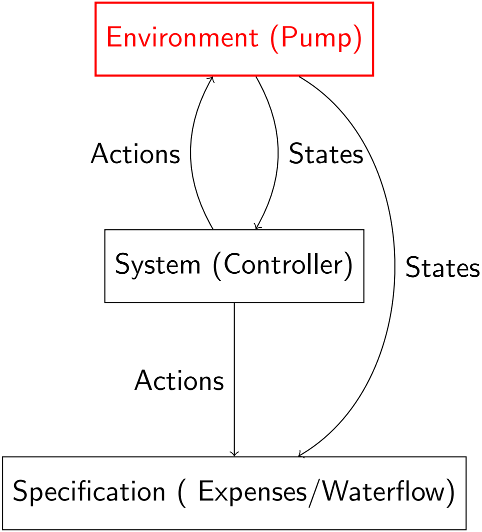
Aim of the controller: Minimize $\frac{Expenses}{Waterflow} \frac{\$}{m^3}$

Overview


Environment (Pump)
Actions
States
System (Controller)
States
Actions
Specification (Expenses/Waterflow)

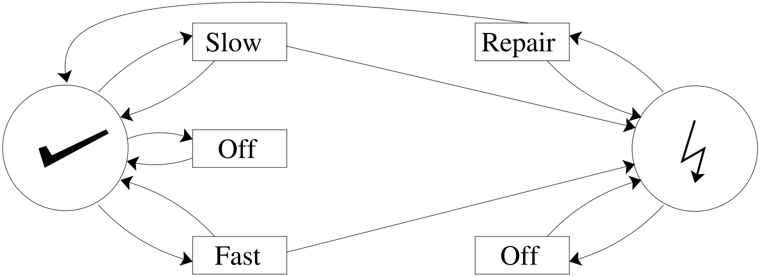
Outline

Overview

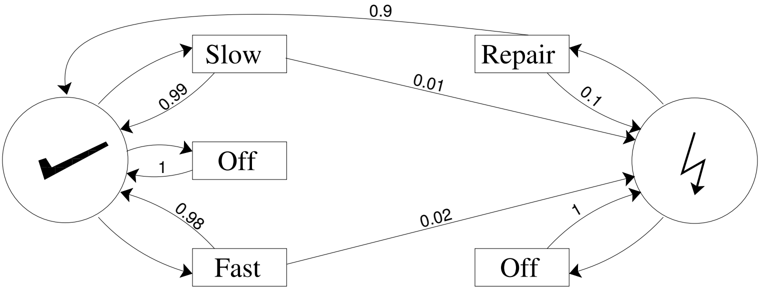


Environment (Pump)
Actions
States
System (Controller)
States
Actions
Specification (Expenses/Waterflow)

Modeling a Pump (Environment)



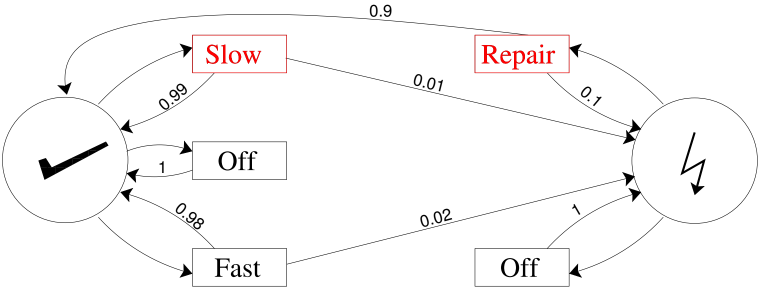
Modeling a Pump (Environment)



Markov Decision Process (MDP)

Modeling a Pump (Environment)

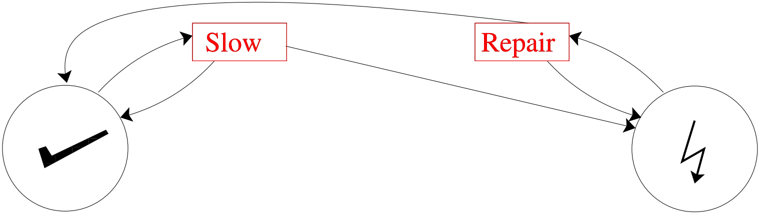
Strategy: Decide what action to takes when



Markov Decision Process (MDP)

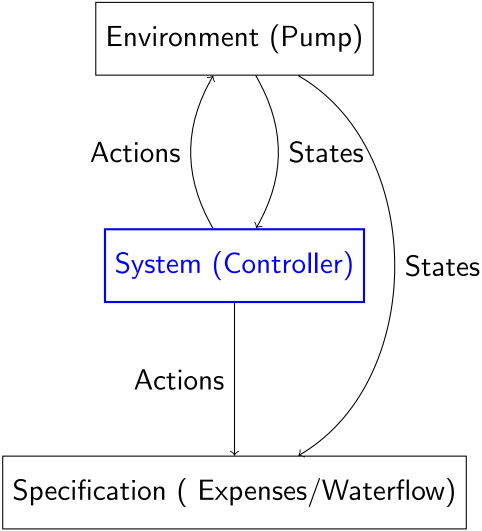
Modeling a Pump (Environment)

Strategy: Decide what action to takes when

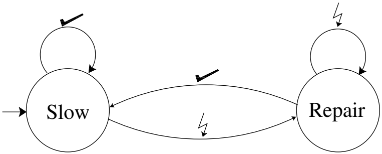


Markov Chain

Overview


Environment (Pump)
Actions
States
System (Controller)
States
Actions
Specification (Expenses/Waterflow)

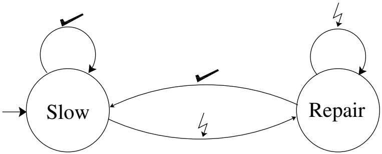
Modeling Systems (Controllers)



Example: Controlling a single pump

- Finite-state machine

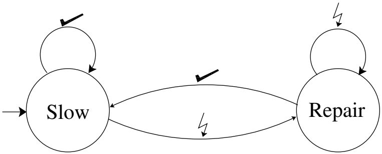
Modeling Systems (Controllers)



Example: Controlling a single pump

- Finite-state machine
- Input: MDP states

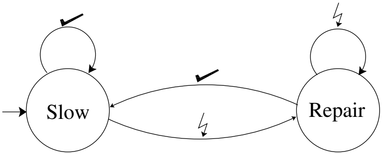
Modeling Systems (Controllers)



Example: Controlling a single pump

- Finite-state machine
- Input: MDP states
- Output: Actions

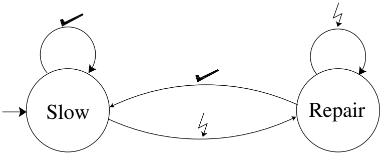
Modeling Systems (Controllers)



Example: Controlling a single pump

- Finite-state machine
- Input: MDP states
- Output: Actions
- Output defined by state (Moore Machine)

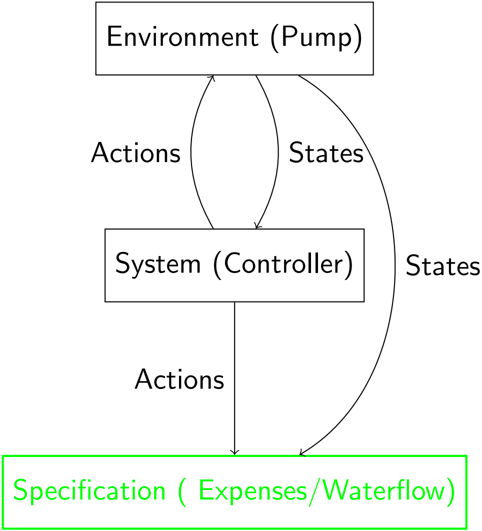
Modeling Systems (Controllers)



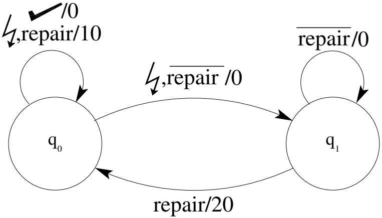
Example: Controlling a single pump

- Finite-state machine
- Input: MDP states
- Output: Actions
- Output defined by state (Moore Machine)
- System is a strategy

Overview

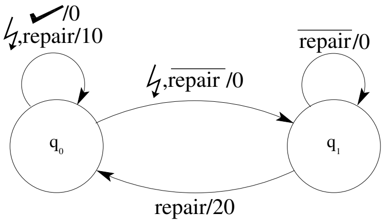

Environment (Pump)
Actions
States
System (Controller)
States
Actions
Specification (Expenses/Waterflow)

Evaluating Systems (Specification)



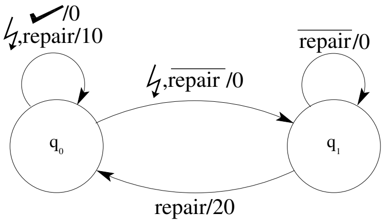
- Automaton with numbers on transitions

Evaluating Systems (Specification)



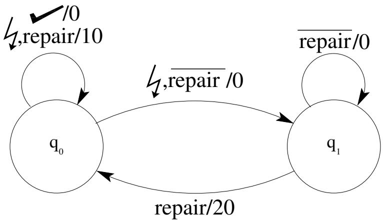
- Automaton with numbers on transitions
- Input: MDP states and actions

Evaluating Systems (Specification)



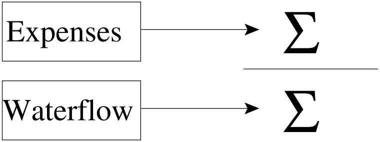
- Automaton with numbers on transitions
- Input: MDP states and actions
- Numbers represent how good or bad a decision is

Evaluating Systems (Specification)

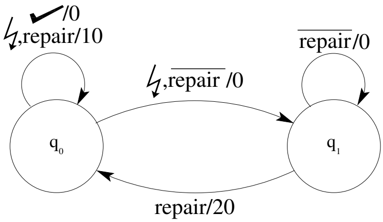


- Automaton with numbers on transitions
- Input: MDP states and actions
- Numbers represent how good or bad a decision is

Minimize

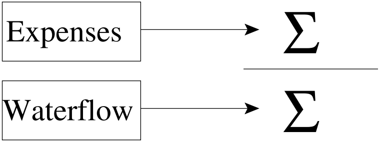


Evaluating Systems (Specification)



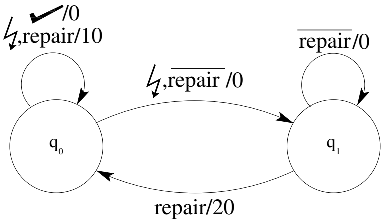
- Automaton with numbers on transitions
- Input: MDP states and actions
- Numbers represent how good or bad a decision is

Minimize



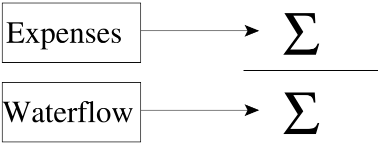
$$= \frac{\sum_{i=l}^{m} c_1(s_i,a_i)}{1+\sum_{i=l}^{m} c_2(s_i,a_i)}$$

Evaluating Systems (Specification)



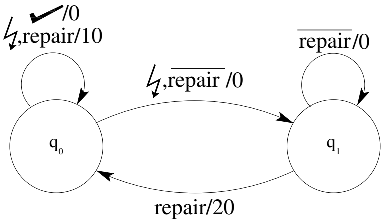
- Automaton with numbers on transitions
- Input: MDP states and actions
- Numbers represent how good or bad a decision is

Minimize



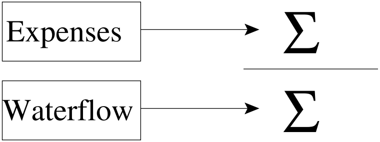
$$= \quad \liminf_{m\to\infty} \frac{\sum_{i=l}^{m} c_1(s_i,a_i)}{1+\sum_{i=l}^{m} c_2(s_i,a_i)}$$

Evaluating Systems (Specification)



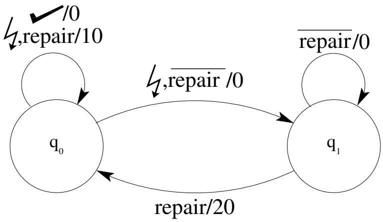
- Automaton with numbers on transitions
- Input: MDP states and actions
- Numbers represent how good or bad a decision is

Minimize



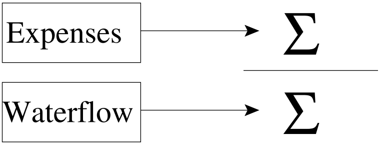
$$= \quad \lim_{l\to\infty} \liminf_{m\to\infty} \frac{\sum_{i=l}^{m} c_1(s_i,a_i)}{1+\sum_{i=l}^{m} c_2(s_i,a_i)}$$

Evaluating Systems (Specification)



- Automaton with numbers on transitions
- Input: MDP states and actions
- Numbers represent how good or bad a decision is

Minimize



$$= \mathbb{E}\Big(\lim_{l\to\infty} \liminf_{m\to\infty} \frac{\sum_{i=l}^{m} c_1(s_i,a_i)}{1+\sum_{i=l}^{m} c_2(s_i,a_i)} \Big)$$

Specification for Multiple Pumps

- Parallel product

Specification for Multiple Pumps

- Parallel product
- Water costs: sum output

Specification for Multiple Pumps

- Parallel product
- Water costs: sum output
- Repair costs: maximum

Specification for Multiple Pumps

- Parallel product
- Water costs: sum output
- Repair costs: maximum
- Additional costs if no pump is running

Outline

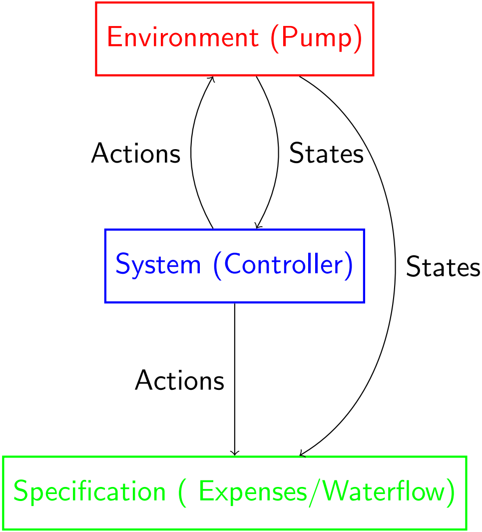
Main Goals

1. Evaluate the quality of a controller (Verification)
2. Automatically find the best controller (Synthesis)

Main Goals

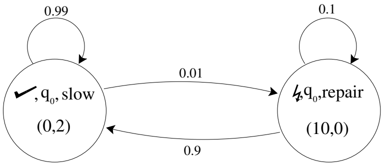
1. Evaluate the quality of a controller (Verification)
2. Automatically find the best controller (Synthesis)

Overview (for Synthesis)



Verification

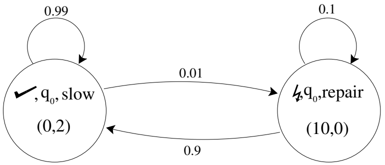
MDP with cost $\times$ controller $=$ Markov Chain



$$\mathbb{E}(\lim \lim\inf \frac{\sum_{i=l}^{m} c_1(s_i)}{1+\sum_{i=l}^{m} c_2(s_i)})$$

Verification

MDP with cost $\times$ controller $=$ Markov Chain

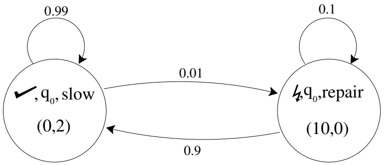


$$\mathbb{E}(\lim \lim\inf \frac{\sum_{i=l}^{m} c_1(s_i)}{1+\sum_{i=l}^{m} c_2(s_i)})$$

$$= \frac{\sum_s c_1(s)\nu(s)}{1+\sum_s c_2(s)\nu(s)}$$

Verification

MDP with cost $\times$ controller $=$ Markov Chain



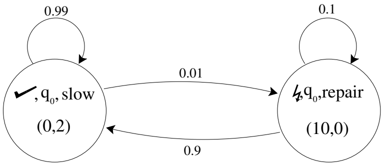
$$\mathbb{E}(\lim \lim\inf \frac{\sum_{i=l}^{m} c_1(s_i)}{1+\sum_{i=l}^{m} c_2(s_i)})$$

$$= \frac{\sum_s c_1(s)\nu(s)}{1+\sum_s c_2(s)\nu(s)}$$

- Calculate ν by linear equations (stationary distribution)

Verification

MDP with cost $\times$ controller $=$ Markov Chain



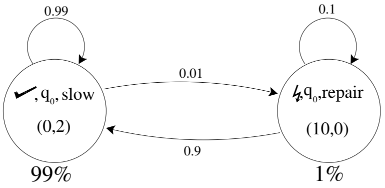
$$\mathbb{E}(\lim \lim \inf \frac{\sum_{i=l}^{m} c_1(s_i)}{1+\sum_{i=l}^{m} c_2(s_i)})$$

$$= \frac{\sum_s c_1(s)\nu(s)}{1+\sum_s c_2(s)\nu(s)}$$

- Calculate ν by linear equations (stationary distribution)
- $\frac{0.01\times 10}{0.99\times 2}$

Verification

MDP with cost $\times$ controller $=$ Markov Chain



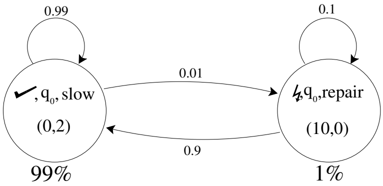
$$\mathbb{E}(\lim \lim\inf \frac{\sum_{i=l}^{m} c_1(s_i)}{1+\sum_{i=l}^{m} c_2(s_i)})$$

$$= \frac{\sum_s c_1(s)\nu(s)}{1+\sum_s c_2(s)\nu(s)}$$

- Calculate ν by linear equations (stationary distribution)
- $\frac{0.01\times 10}{0.99\times 2}$
- Corner case: Value infinity (all denominators zero)

Verification

MDP with cost × controller = Markov Chain



$$\mathbb{E}(\lim \lim \inf \frac{\sum_{i=l}^{m} c_1(s_i)}{1+\sum_{i=l}^{m} c_2(s_i)})$$

$$= \frac{\sum_s c_1(s)\nu(s)}{1+\sum_s c_2(s)\nu(s)}$$

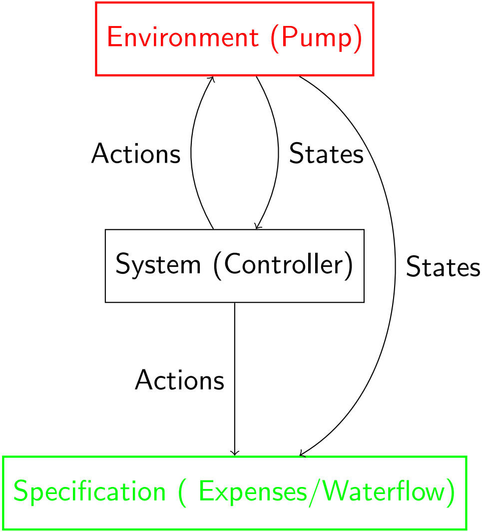
- Calculate ν by linear equations (stationary distribution)
- $\frac{0.01\times 10}{0.99\times 2}$
- Corner case: Value infinity (all denominators zero)

For non-SCC MC: SCCs, then probability of reaching these components

Main Goals

1. Evaluate the quality of a controller (Verification)
2. Automatically find the best controller (Synthesis)

Overview (for Verif. and Synthesis)



(One) Algorithm

How to find an optimal strategy?

- $x(s,a)$ Prob. of being in s and choosing a

$\forall s\ \forall a\ x(s,a)$

Minimize $\frac{\sum x(s,a) c_1(s,a)}{\sum x(s,a) c_2(s,a)}$

subject to

1. $\sum_{s,a} x(s,a) = 1$
2. $\forall s\ \sum_a x(s,a) = \sum_{s',a} x(s',a) \cdot p(s',a,s)$

(One) Algorithm

How to find an optimal strategy?

$\forall s\ \forall a\ x(s,a)$

Minimize $\frac{\sum x(s,a) c_1(s,a)}{\sum x(s,a) c_2(s,a)}$

subject to

1. $\sum_{s,a} x(s,a) = 1$
2. $\forall s\ \sum_a x(s,a) = \sum_{s',a} x(s',a) \cdot p(s',a,s)$

- $x(s,a)$ Prob. of being in s and choosing a
- Probability of being in s: $\sum x(s,a)$

(One) Algorithm

How to find an optimal strategy?

$\forall s \; \forall a \; x(s,a)$

Minimize $\frac{\sum x(s,a) c_1(s,a)}{\sum x(s,a) c_2(s,a)}$

subject to

1. $\sum_{s,a} x(s,a) = 1$
2. $\forall s \; \sum_a x(s,a) = \sum_{s',a} x(s',a) \cdot p(s',a,s)$

- $x(s,a)$ Prob. of being in s and choosing a
- Probability of being in s: $\sum x(s,a)$
- $x(s,a)$ are a steady state distribution

(One) Algorithm

How to find an optimal strategy?

$\forall s \; \forall a \; x(s,a)$

Minimize $\frac{\sum x(s,a) c_1(s,a)}{\sum x(s,a) c_2(s,a)}$

subject to

1. $\sum_{s,a} x(s,a) = 1$
2. $\forall s \; \sum_a x(s,a) = \sum_{s',a} x(s',a) \cdot p(s',a,s)$

- $x(s,a)$ Prob. of being in s and choosing a
- Probability of being in s: $\sum x(s,a)$
- $x(s,a)$ are a steady state distribution
- We search for the best steady state distribution

(One) Algorithm

How to find an optimal strategy?

$\forall s \; \forall a \; x(s,a)$

Minimize $\frac{\sum x(s,a)c_1(s,a)}{\sum x(s,a)c_2(s,a)}$

subject to

1. $\sum_{s,a} x(s,a) = 1$
2. $\forall s \; \sum_a x(s,a) = \sum_{s',a} x(s',a) \cdot p(s',a,s)$

- $x(s,a)$ Prob. of being in s and choosing a
- Probability of being in s: $\sum x(s,a)$
- $x(s,a)$ are a steady state distribution
- We search for the best steady state distribution
- First equation: $x(s,a)$ has to be a probability distribution

(One) Algorithm

How to find an optimal strategy?

$\forall s\ \forall a\ x(s,a)$

Minimize $\frac{\sum x(s,a) c_1(s,a)}{\sum x(s,a) c_2(s,a)}$

subject to

1. $\sum_{s,a} x(s,a) = 1$
2. $\forall s\ \sum_a x(s,a) = \sum_{s',a} x(s',a) \cdot p(s',a,s)$

- $x(s,a)$ Prob. of being in s and choosing a
- Probability of being in s: $\sum x(s,a)$
- $x(s,a)$ are a steady state distribution
- We search for the best steady state distribution
- First equation: $x(s,a)$ has to be a probability distribution
- Second equation:
  - $\sum_{s',a} x(s',a) \cdot p(s',a,s)$: Prob. of going to s

Outline

Experimental Results (Behavior)

How smart are our controllers?
2 Pumps:

- Play it safe
- Turn one pump on in fast mode.
- When one pump is broken, repair it, turn the other one on in slow mode.

Experimental Results (Behavior)

How smart are our controllers?
3 Pumps:

- Turn all pumps on fast
- When one pump is broken, turn one on fast, the other two off
- When two pumps are broken, turn the last one on fast, repair the rest

Experimental Results (Behavior)

How smart are our controllers?
4 Pumps:

- Turn all pumps on fast
- When one pump is broken, turn two pumps on fast, the other two off
- When two pumps are broken, turn one on fast
- When three pumps are broken, turn the last one on fast, repair the rest

Thank you

Fin.