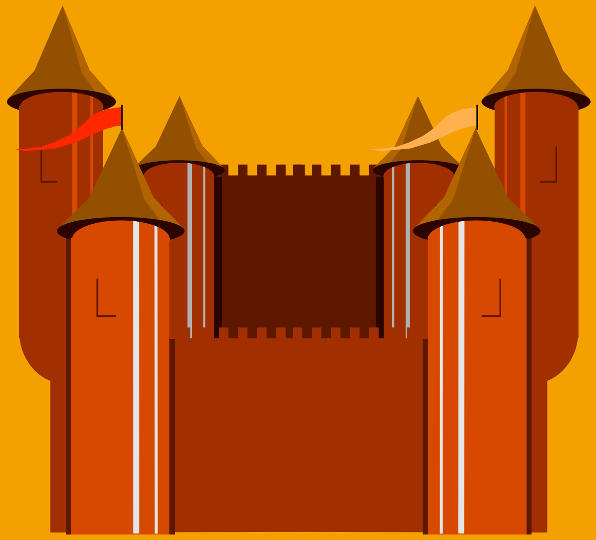
Markov Decision Processes: AI vs. OR

Workshop on Decision-Making in Adversarial Domains
May 23, 2005



Warren Powell
CASTLE Laboratory
Princeton University
http://www.castlelab.princeton.edu/

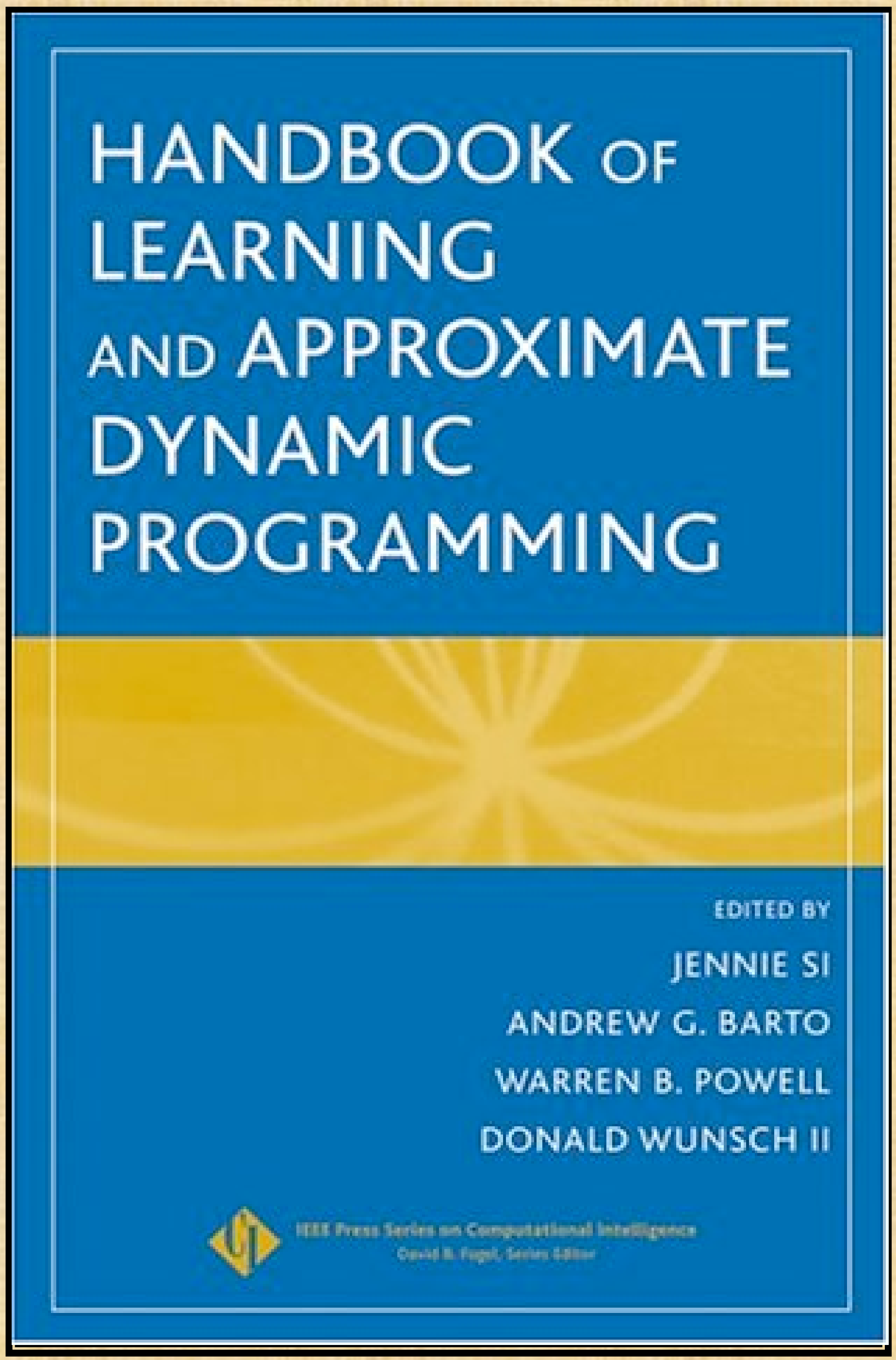
Bellman's equations

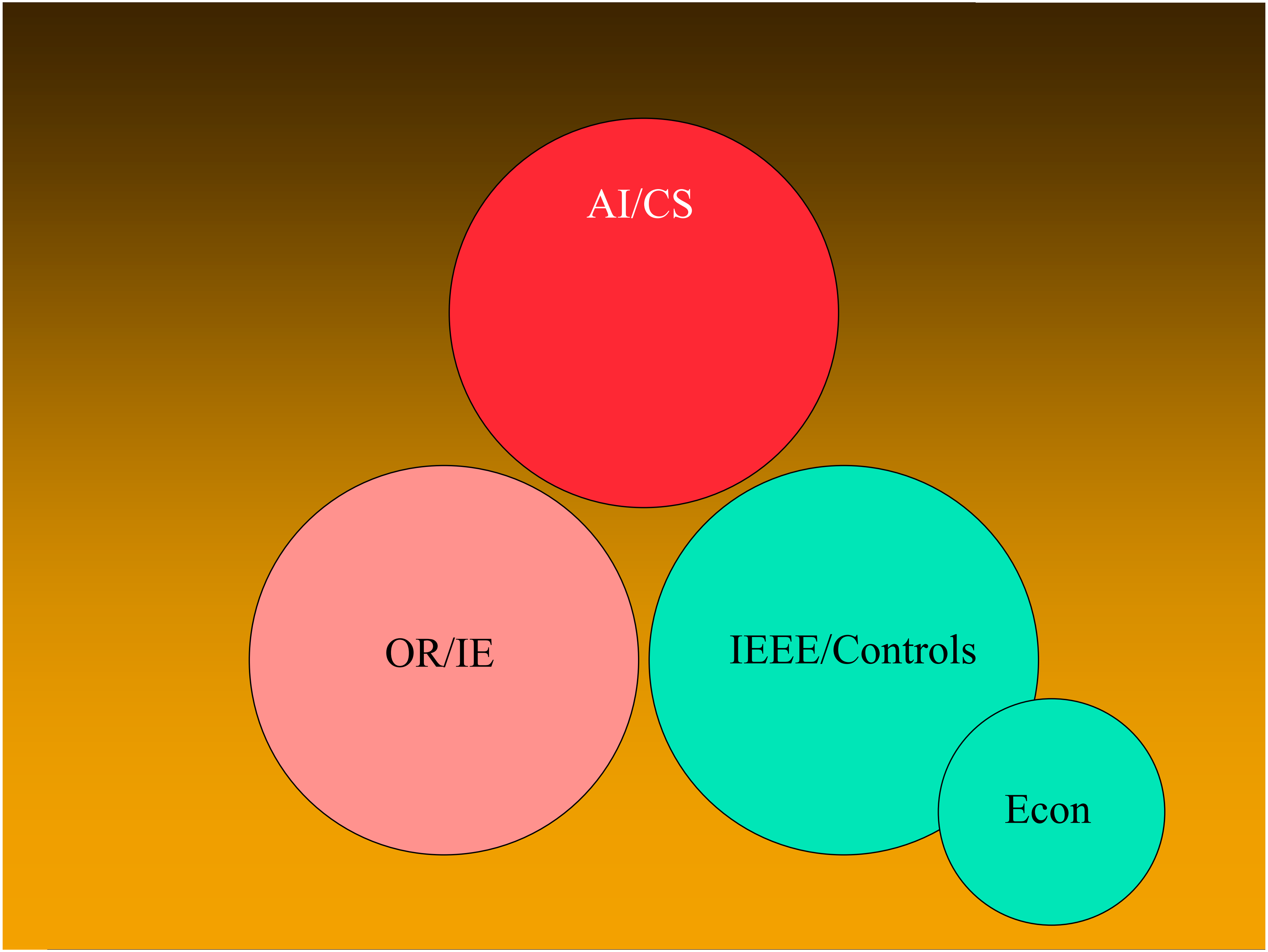
- **The classic view of Bellman's optimality equation:**
  - **Deterministic problems**

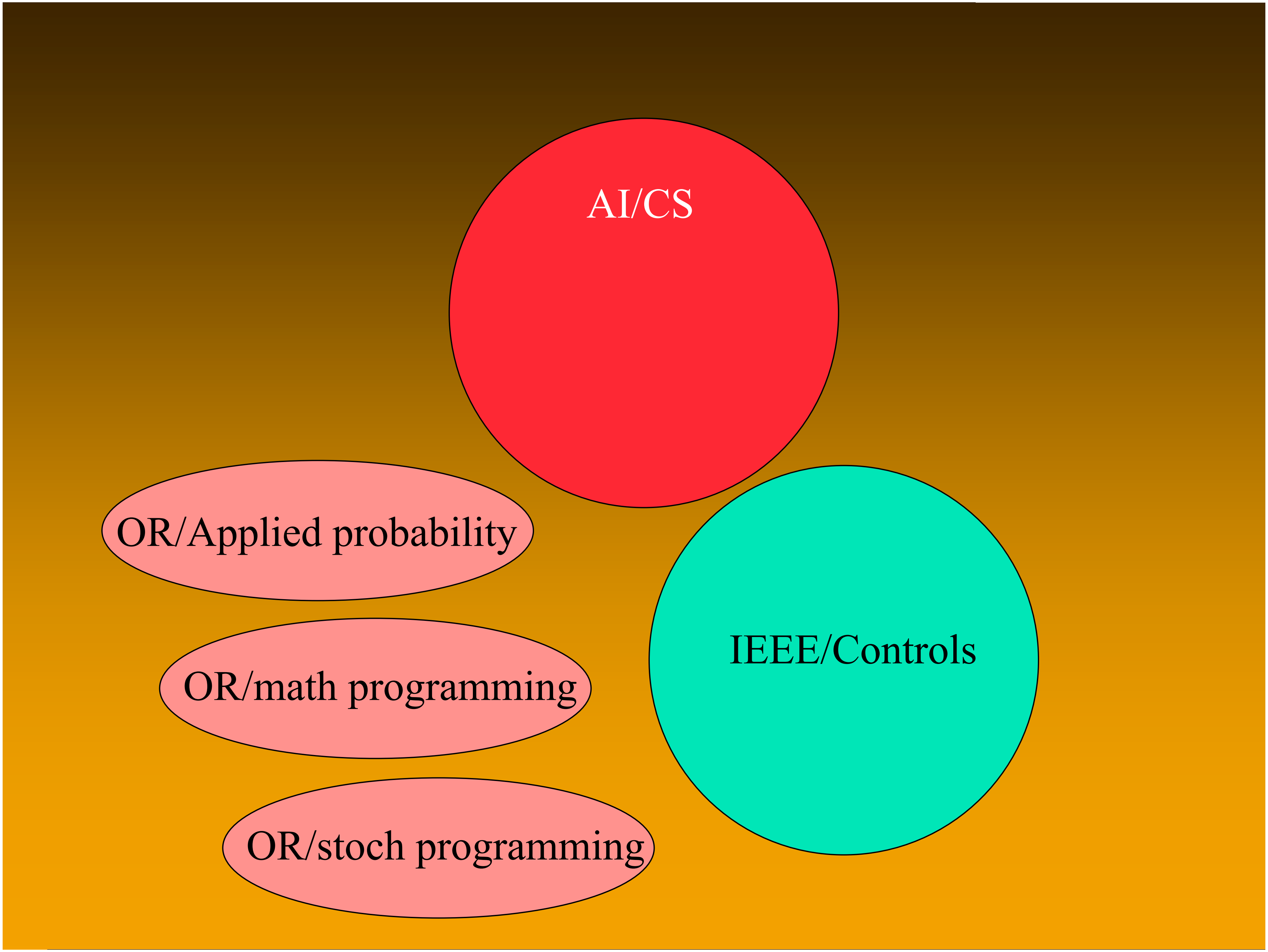
$$V_t(S_t) = \max_{d_t}\left(C(S_t, d_t) + \gamma V_{t+1}(S_{t+1}) \right)$$

  - **Stochastic problems**

$$V_t(S_t) = \max_{d_t}\left(C(S_t, d_t) + \gamma \sum_{s' \in \mathcal{S}} p(S_{t+1} = s' \mid S_t, d_t) V_{t+1}(s') \right)$$


HANDBOOK OF
LEARNING
AND APPROXIMATE
DYNAMIC
PROGRAMMING
EDITED BY
JENNIE SI
ANDREW G. BARTO
WARREN B. POWELL
DONALD WUNSCH II
IEEE Press Series on Computational Intelligence
David B. Fogel, Series Editor


AI/CS
OR/IE
IEEE/Controls
Econ


AI/CS
OR/Applied probability
OR/math programming
OR/stoch programming
IEEE/Controls

Approximate Dynamic Programming for Asset Management:

Solving the curses of dimensionality

Warren B. Powell

March 31, 2004

Department of Operations Research and Financial Engineering
Princeton University, Princeton, NJ 08544

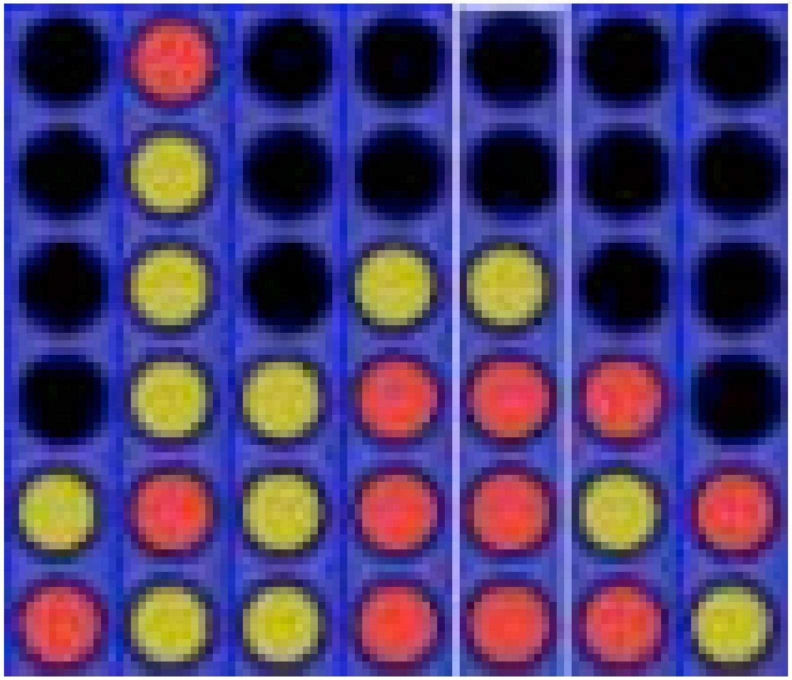
AI vs OR

- **If we are all solving the same problem, why do OR and AI people have such different attitudes?**
  - **Hypothesis: We are solving very different problems**
- **AI problems**
  - **Typically managing a single entity (aka resource, asset)**
    - Playing a game
    - Controlling a robot
- **OR Problems**
  - **Typically managing multiple resources/assets:**
    - Routing a fleet of vehicles
    - Balancing a financial portfolio

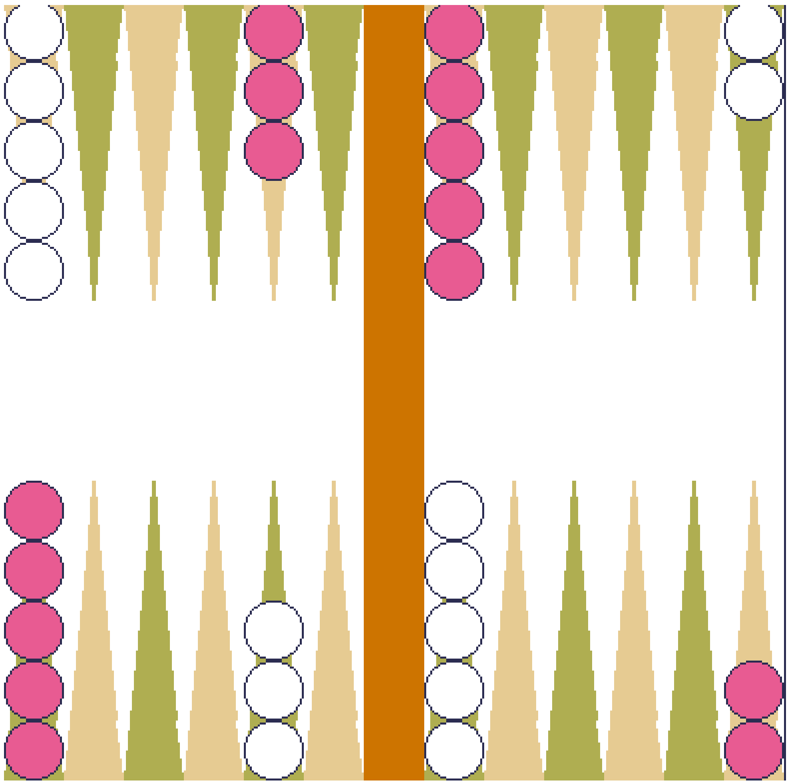
Problem characteristics

- **Single entity/resource/asset**
  - **State space**
    - » "Small" state space (state = attributes of the resource)
      - – Small = small enough to enumerate
    - » Large state space
  - **Reward structure**
    - » Shallow reward structure
      - – Receive rewards at every step, providing guidance
    - » Deeply-nested reward structure
      - – Have to play many steps before receiving a reward

Connect-4

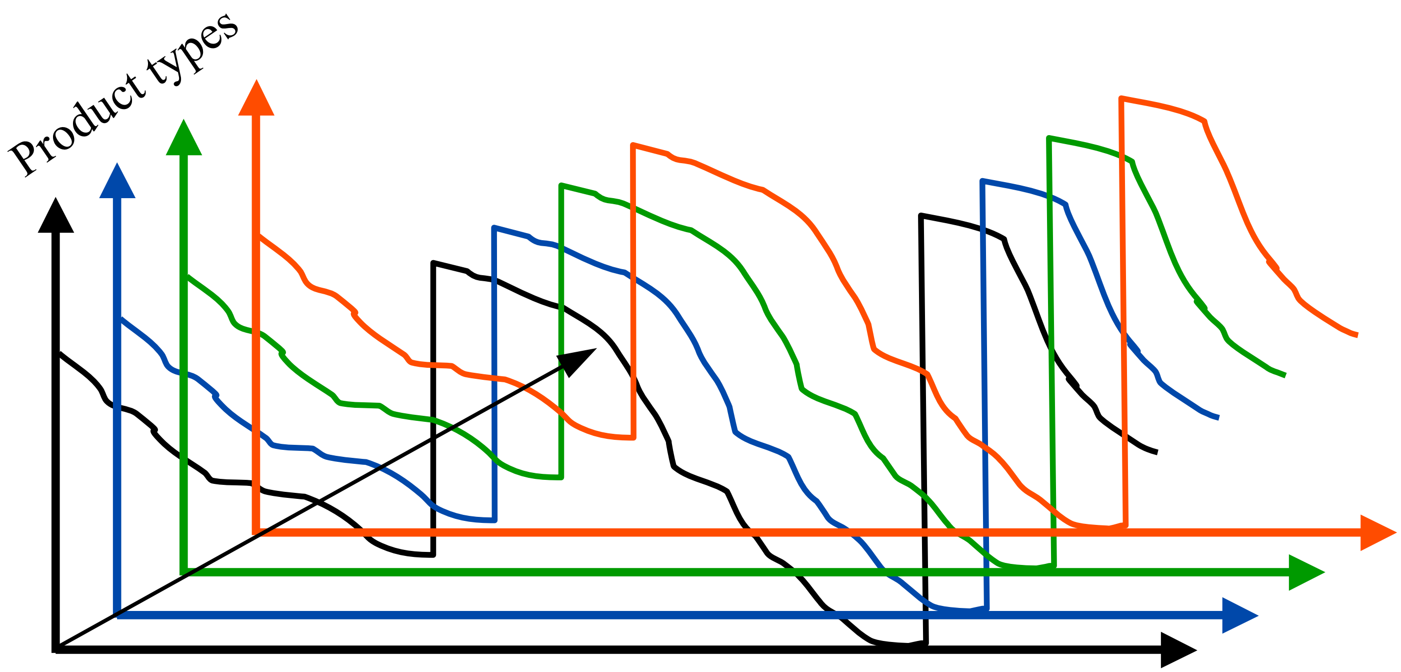


Backgammon

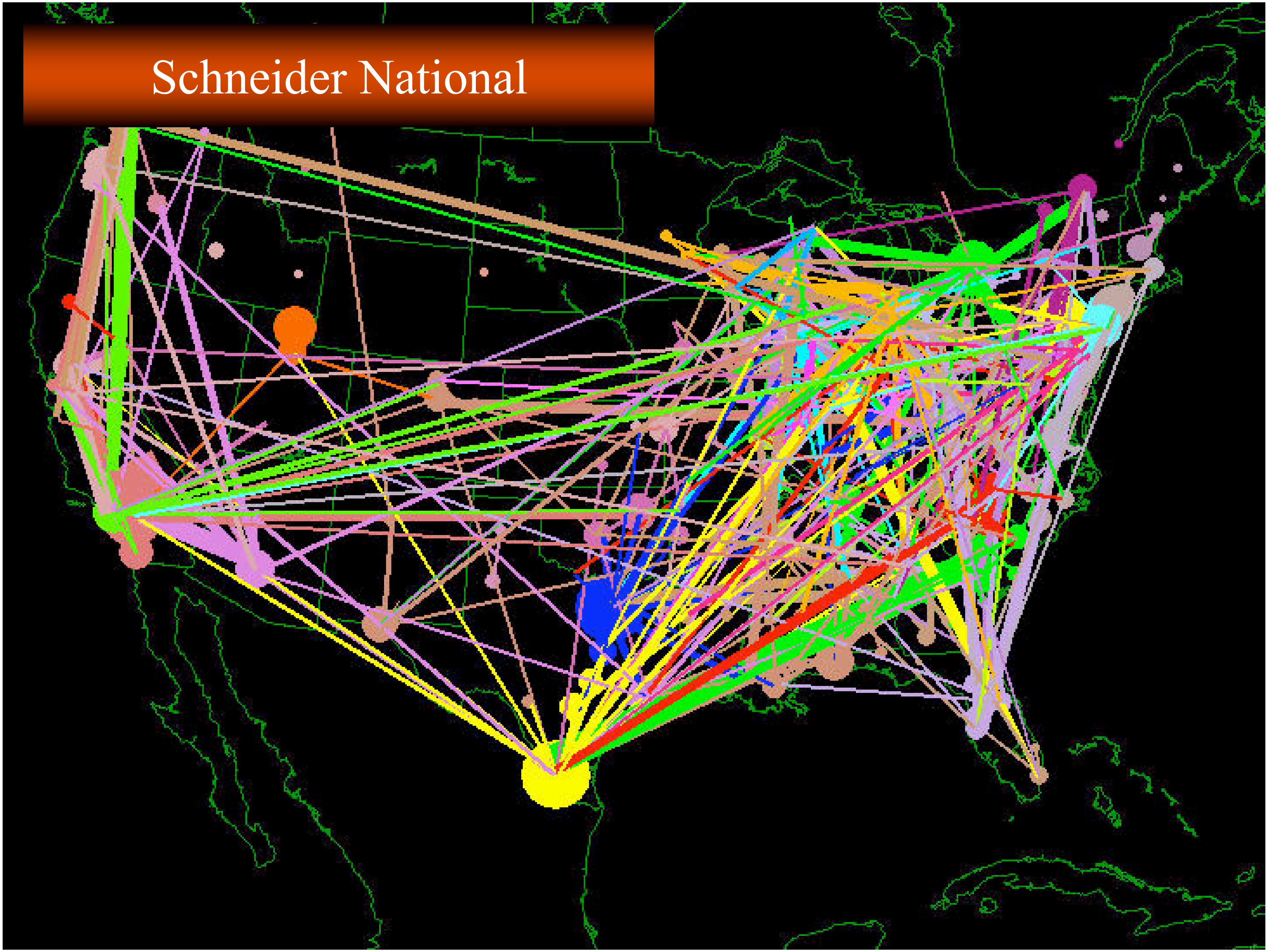


OR problems

- Multiproduct inventory problems



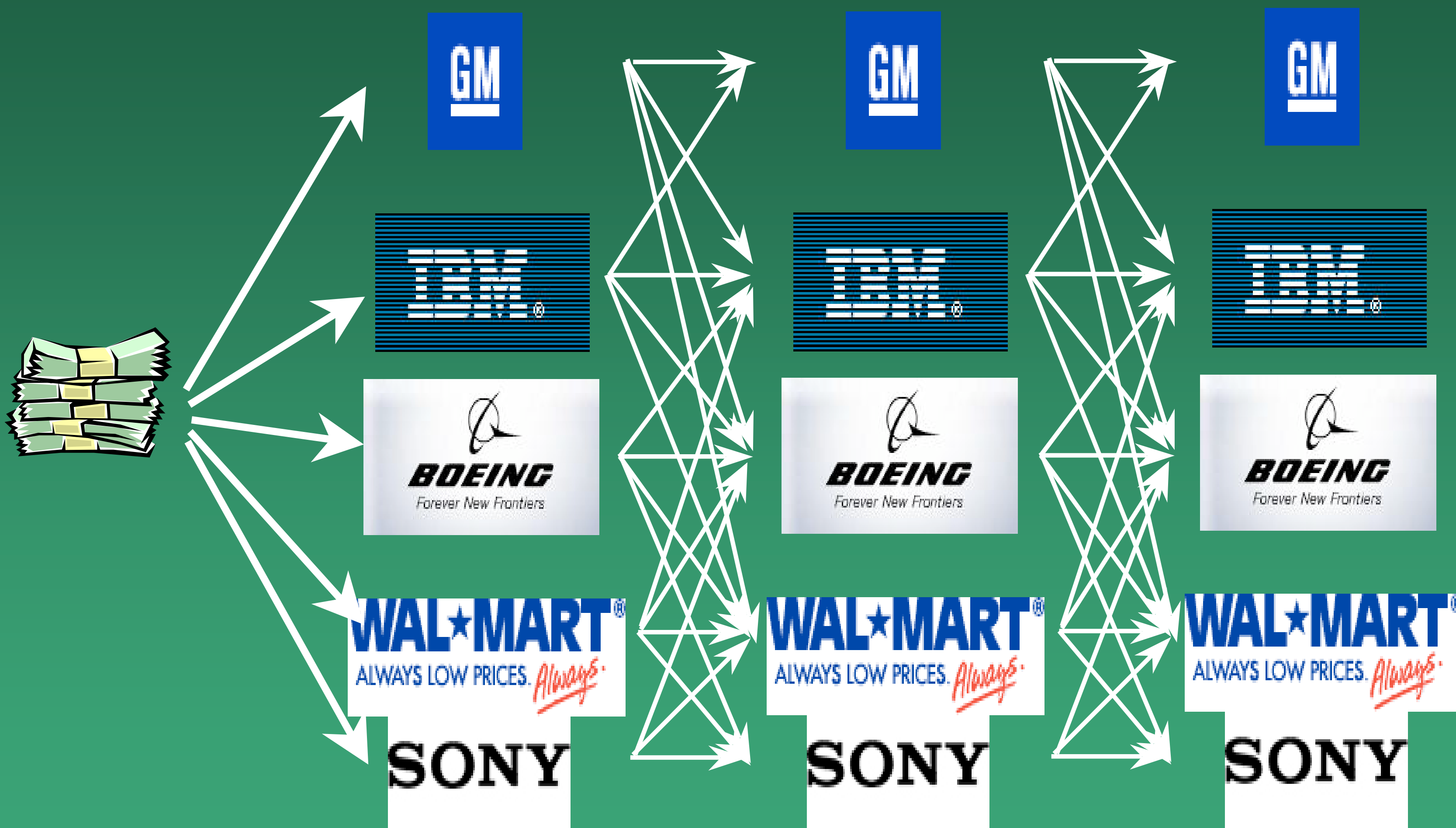
Schneider National

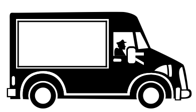


The problem classes

- Class 4: Multicommodity flow problems

- Financial asset management:





State variables

- **How big is the state space?**

If the attribute vector has one dimension:

$$a^{Truck} = \left[\text{Location}\right] \quad |\mathcal{A}| = 100 - 1000 \text{ locations}$$

The attribute space grows with the number of dimensions:

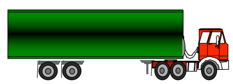
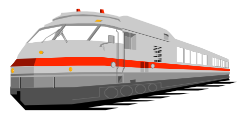
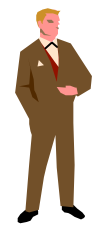
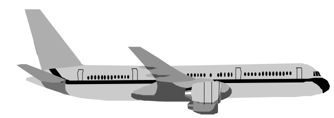
$$a^{Truck} = \begin{bmatrix} \text{Location} \\ \text{Fleet type} \end{bmatrix} \quad \text{500 locations, 5 fleet types} \quad |\mathcal{A}| = 2500$$

$$a^{Truck} = \begin{bmatrix} \text{Location} \\ \text{Fleet type} \\ \text{Domicile} \end{bmatrix} \quad \text{500 locations, 5 fleet types, 100 domiciles} \quad |\mathcal{A}| = 250{,}000$$

... classic curse of dimensionality.

State variables

- **Attributes can get complicated:**

$$a = \begin{bmatrix} \text{Location} \\ \text{Equipment type} \end{bmatrix} \quad \begin{bmatrix} \text{Location} \\ \text{ETA} \\ \text{Equipment type} \\ \text{Train priority} \\ \text{Pool} \\ \text{Due for maint} \\ \text{Home shop} \end{bmatrix} \quad \begin{bmatrix} \text{Location} \\ \text{ETA} \\ \text{Bus. segment} \\ \text{Single/team} \\ \text{Domicile} \\ \text{Drive hours} \\ \text{Duty hours} \\ \text{8 day history} \\ \text{Days from home} \end{bmatrix} \quad \begin{bmatrix} \text{Location} \\ \text{ETA} \\ \text{A/C type} \\ \text{Fuel level} \\ \text{Home shop} \\ \text{Crew} \\ \text{Eqpt1} \\ \vdots \\ \text{Eqpt100} \end{bmatrix}$$

State variables

- **We need to distinguish between the state of a single resource, and the state of our system**

For a single resource:

a = Attribute (state) of a single resource.

$\mathcal{A}$ = Attribute space (state space of a single resource).

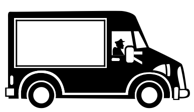
For multiple resources:

R_{ta} = Number of resources with attribute (at time)

R_t = Resource state vector

$$= \left(R_{ta}\right)_{a \in \mathcal{A}}$$

$\mathcal{R}$ = Resource state space (set of all possible outcomes of R_t)



State variables

- **What if we have N > 1 trucks?**

$$|\mathcal{R}| = \binom{N+|\mathcal{A}|-1}{|\mathcal{A}|-1}$$

Why are they hard?

- **AI problems are hard because**
  - **“attribute space” is often very large.**
  - **Reward structure is deeply nested.**

- **OR problems are hard because**
  - **we are managing many resources.**

The curses of dimensionality

- **Actually the situation is somewhat worse…**

$$V_t(R_t) = \max_{x \in \mathcal{X}} \left\{ C_t(R_t, x_t) + E\, V_{t+1}(R_{t+1}) \right\}$$

The curses of dimensionality

- **The computational challenge:**

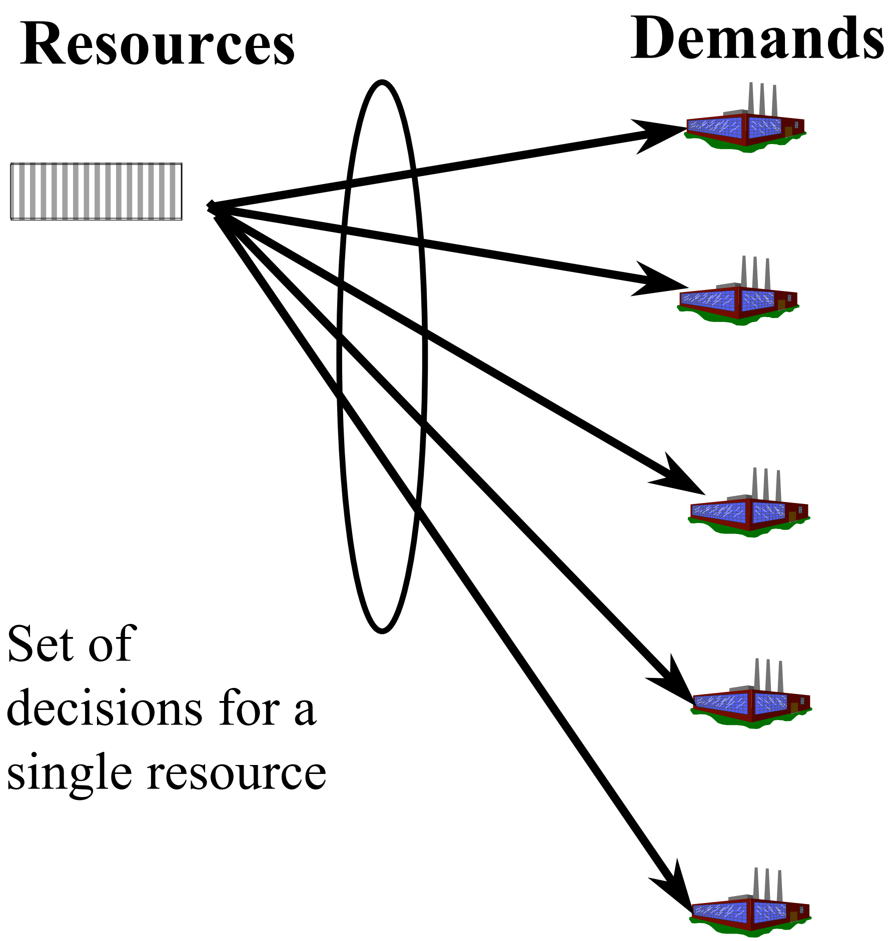
$$V_t(R_t) = \max_{x \in \mathcal{X}} \left\{ C_t(R_t, x_t) + E\{V_{t+1}(R_{t+1})\} \right\}$$

How do we find $V_{t+1}(R_{t+1})$?

How do we compute the expectation?

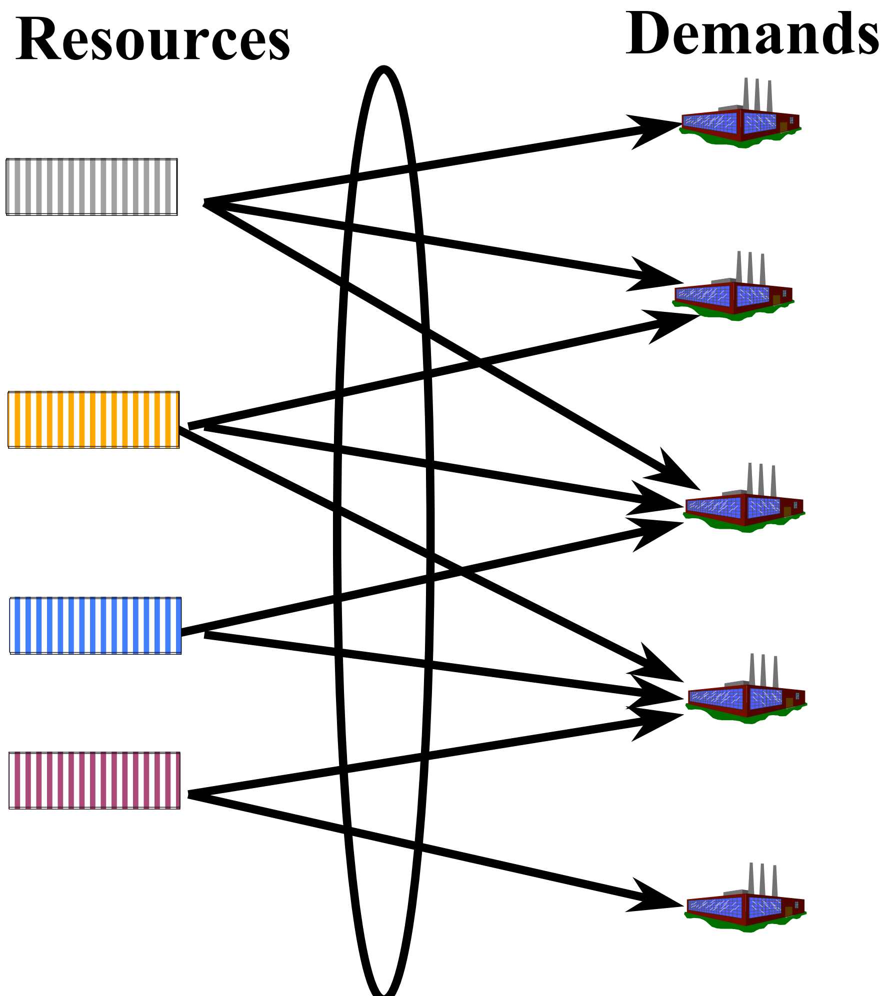
How do we find the optimal solution?

Managing multiple resources



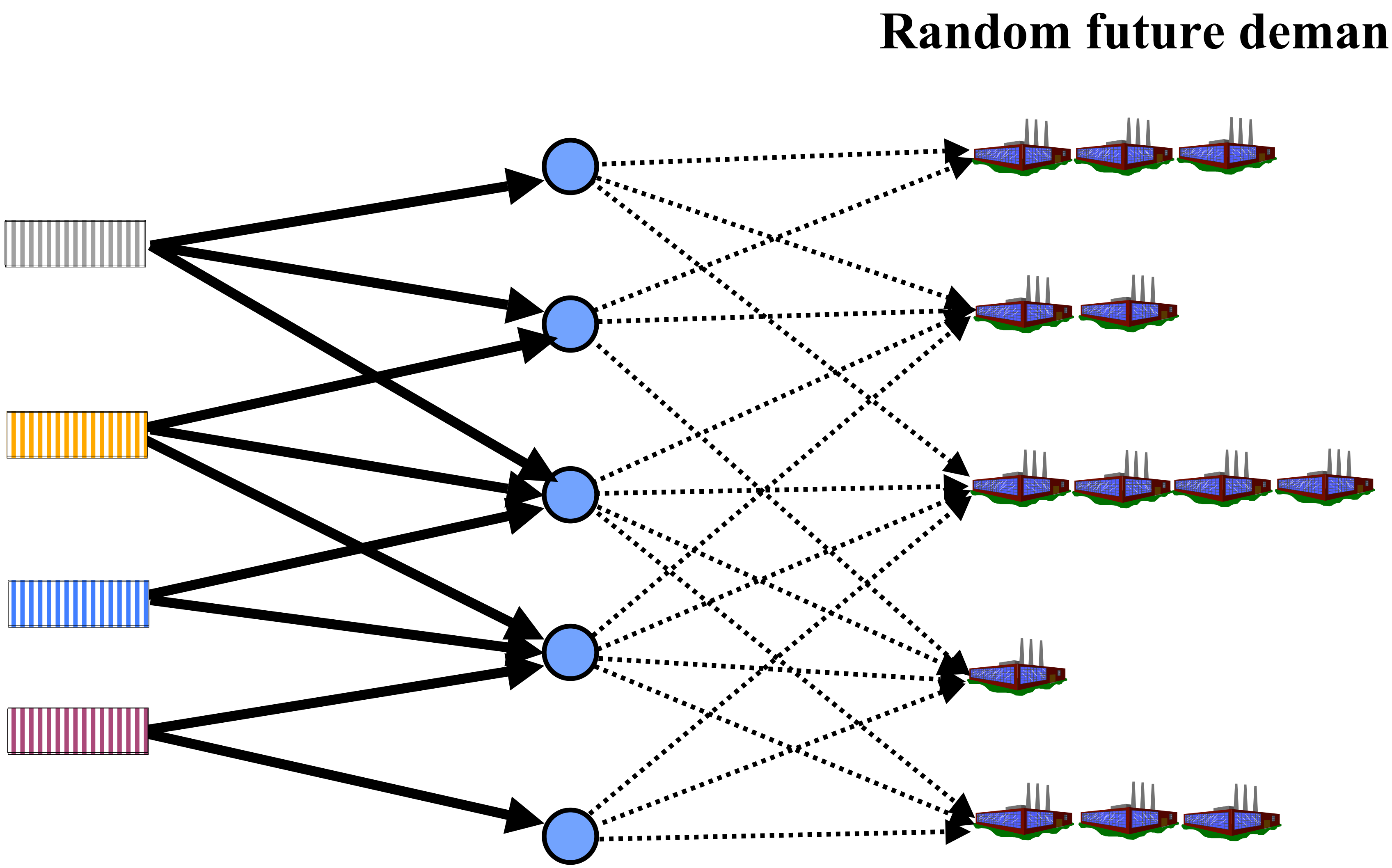
$x_d = 1$ if we choose this decision.

Managing multiple resources



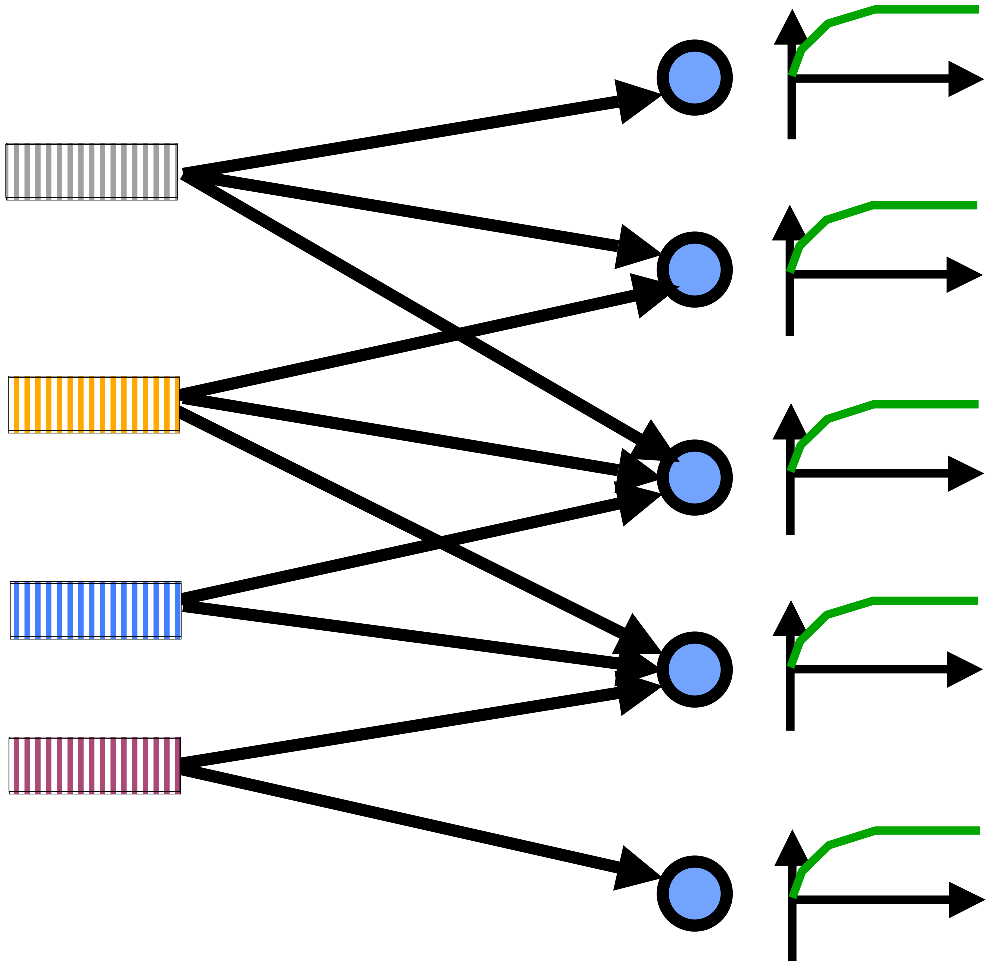
$x x c = \in = \left(A x \geq b\right) x_{d} \quad {}_{d \in D}$ argmax:,0

Managing multiple resources



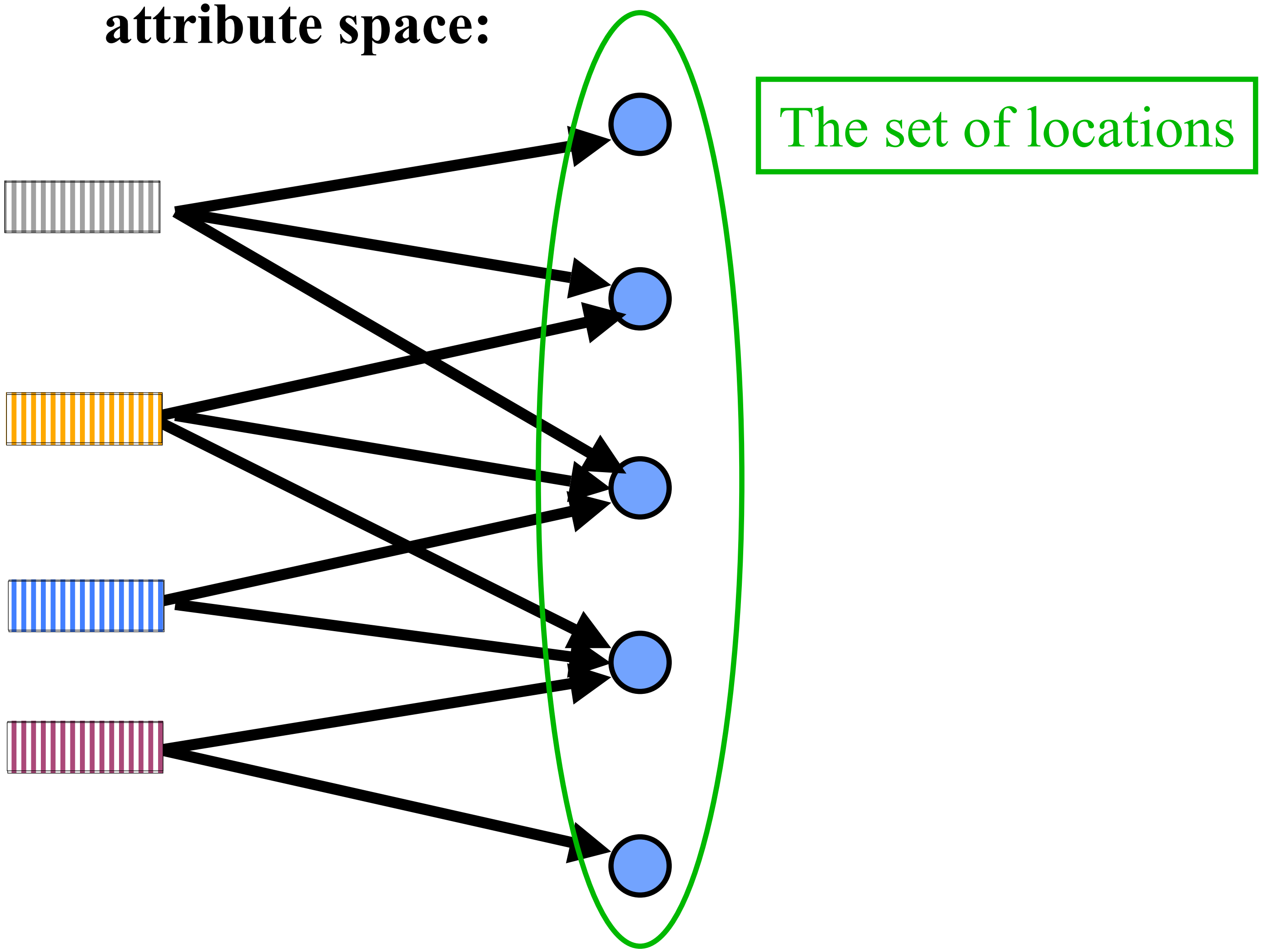
Managing multiple resources

- We can use a separable approximation of the future:



Managing multiple resources

- OR people normally assume we can enumerate the attribute space:



Managing multiple resources

- … in reality, we often cannot.