Approximate Dynamic Programming: Solving the curses of dimensionality

Multidisciplinary Symposium on Reinforcement Learning

June 19, 2009

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

© 2009 Warren B. Powell, Princeton University
Truckload trucking
Truckload trucking
The load matching problem
The load matching problem
The load matching problem

Optimizing at a point in time

Optimizing over time
Truckload trucking

- Deterministic time-space networks

How do we make this *stochastic*?
State variables

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

$$|States| = \left( \begin{array}{c} \text{No. trucks} + |Locations| - 1 \\ |Locations| - 1 \end{array} \right)$$
Yellow Freight System

- Deterministic, (integer) multicommodity flow
Yellow Freight System
Yellow Freight System
Yellow Freight System
NetJets created the concept of fractional jet ownership giving individuals and businesses all the benefits of whole aircraft ownership and more at a fraction of the cost.

390,000 flights annually.

EXPERIENCE

as defined by NetJets.

EXPLORE THE NETJETS DIFFERENCES

<table>
<thead>
<tr>
<th>MOST EXPERIENCE</th>
<th>HIGHEST SAFETY STANDARDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MARKET LEADER</td>
<td>SUPERIOR OWNERSHIP EXPERIENCE</td>
</tr>
<tr>
<td>WORLD'S LARGEST FLEET</td>
<td>GLOBAL OPERATIONS</td>
</tr>
<tr>
<td>MOST AIRCRAFT TYPES</td>
<td>A BERKSHIRE HATHAWAY COMPANY</td>
</tr>
</tbody>
</table>

THE NETJETS CLIMATE INITIATIVE

LEARN MORE
High value spare parts

Electric Power Grid

- PJM oversees an aging investment in high-voltage transformers.
- Replacement strategy needs to anticipate a bulge in retirements and failures.
- 1-2 year lag times on orders. Typical cost of a replacement ~ $5 million.
- Failures vary widely in terms of economic impact on the network.

Spare parts for business jets

- ADP is used to determine purchasing and allocation strategies for over 400 high value spare parts.
- Inventory strategy has to determine what to buy, when and where to store it. Many parts are very low volume (e.g. 7 spares spread across 15 service centers).
- Inventories have to meet global targets on level of service and inventory costs.
Energy resource allocation

• What is the right mix of energy technologies?
• How should the use of different energy resources be coordinated over space and time?
• What should my energy R&D portfolio look like?
• Should I invest in nuclear energy?
• What is the impact of a carbon tax?

Energy markets

• How should I hedge energy commodities?
• How do I price energy assets?
• What is the right price for energy futures?
Outline

- Modeling resource allocation problems
- ADP and the post-decision state variable
- A blood management example
- A brief overview of CASTLE Lab research
- Applications
  - Trucking
  - Rail
  - Energy
Outline

- Modeling resource allocation problems
- ADP and the post-decision state variable
- A blood management example
- A brief overview of CASTLE Lab research
- Applications
  - Trucking
  - Rail
  - Energy
Modeling dynamic programs

- Attribute vectors:

\[ a = \begin{bmatrix} 
\text{Blood type} \\
\text{Age} \\
\text{Frozen?} 
\end{bmatrix} \begin{bmatrix} 
\text{Technology} \\
\text{Location} \\
\text{Status} \\
\text{Age} 
\end{bmatrix} \begin{bmatrix} 
\text{Location} \\
\text{ETA} \\
\text{Bus. segment} \\
\text{Single/team} \\
\text{Domicile} \\
\text{Drive hours} \\
\text{Days from home} 
\end{bmatrix} \begin{bmatrix} 
\text{Location} \\
\text{ETA} \\
\text{A/C type} \\
\text{Fuel level} \\
\text{Home shop} \\
\text{Crew} \\
\text{Eqpt1} \\
\vdots \\
\text{Eqpt100} 
\end{bmatrix} \]
Modeling dynamic programs

- Modeling resources:
  - The state of a single resource:
    \[ a = \text{The attributes of a single resource} \]
    \[ a \in \mathcal{A} \quad \text{The attribute space} \]
  - The state of multiple resources:
    \[ R_{ta} = \text{The number of resources with attribute } a \]
    \[ R_t = \left( R_{ta} \right)_{a \in \mathcal{A}} \quad \text{The resource state vector} \]
  - The information process:
    \[ \hat{R}_{ta} = \text{The change in the number of resources with attribute } a. \]
Modeling dynamic programs

■ Modeling demands:

» The attributes of a single demand:
  \[ b = \text{The attributes of a demand to be served.} \]
  \[ b \in \mathcal{B} \quad \text{The attribute space} \]

» The demand state vector:
  \[ D_{tb} = \text{The number of demands with attribute } b \]
  \[ D_t = \left( D_{tb} \right)_{b \in \mathcal{B}} \quad \text{The demand state vector} \]

» The information process:
  \[ \hat{D}_{tb} = \text{The change in the number of demands with attribute } b. \]
Modeling dynamic programs

System state: \( S_t = (R_t, D_t, \rho_t) \)

- Resource state
- Market demands

“System parameters”:
- State of technology (costs, performance)
- Climate, weather (temperature, rainfall, wind)
- Government policies (tax rebates on solar panels)
- Market prices (oil, coal)
Modeling dynamic programs

- The three states of our system
  » The state of a single resource/entity
    \[
    a_t = \begin{bmatrix}
    a_{t1} \\
    a_{t2} \\
    a_{t3}
    \end{bmatrix}
    \]
  » The resource state vector
    \[
    R_t = \begin{bmatrix}
    R_{ta_1} \\
    R_{ta_2} \\
    R_{ta_3}
    \end{bmatrix}
    \]
  » The system state vector
    \[
    S_t = (R_t, D_t, \rho_t)
    \]
Modeling dynamic programs

The decision variables:

\[ x_t = \begin{cases} 
\text{Purchase new assets} \\
\text{Move equipment} \\
\text{Clean/repair equipment} \\
\text{Put in storage} \\
\text{Assign resource to task} 
\end{cases} \]
Modeling dynamic programs

- Exogenous information:

\[ W_t = \text{New information} = (\hat{R}_t, \hat{D}_t, \hat{\rho}_t) \]

- \( \hat{R}_t \): Exogenous changes in capacity, reserves
- \( \hat{D}_t \): New demands for energy by type
- \( \hat{\rho}_t \): Exogenous changes in parameters.

Information can be:
- Fine-grained:
  - Wind, solar, demand, prices, ...
- Coarse-grained:
  - Changes in technologies, policies, climate
Modeling dynamic programs

- The transition function

\[ S_{t+1} = S^M(S_t, x_t, W_{t+1}) \]

*Also known as the “system model,” “plant model” or just “model.”*

*All the physics of the problem.*
Modeling dynamic programs

- A general optimization formulation:
  » Let $\Pi$ be a class of policies (this requires specifying the structure of the function).
  » An element $\pi \in \Pi$ refers to the specific parameters that define the function.
  » Define

$$X_t^\pi(S_t) = \text{Decision function (or policy) that returns a feasible decision vector } x \text{ given state } S_t.$$ 

» The optimization problem is:

$$\max/\min_{\pi \in \Pi} \mathbb{E} \left\{ \sum_{t=0}^{T} \gamma^t C(S_t, X_t^\pi(S_t)) \right\}$$
Outline

- Modeling resource allocation problems
- ADP and the post-decision state variable
- A blood management example
- A brief overview of CASTLE Lab research
- Applications
  - Trucking
  - Rail
  - Energy
Introduction to ADP

- We can optimize decisions using Bellman’s equation

\[ V_t(S_t) = \max_{x \in \mathcal{X}} \left( C_t(S_t, x_t) + E \{ V_{t+1}(S_{t+1}) \mid S_t \} \right) \]
Introduction to ADP

The computational challenge:

\[ V_t(S_t) = \max_{x \in \mathcal{X}} \left( C_t(S_t, x_t) + E \{ V_{t+1}(S_{t+1}) | S_t \} \right) \]

How do we find \( V_{t+1}(S_{t+1}) \)?

How do we compute the expectation?

How do we find the optimal solution?
Introduction to ADP

- Classical ADP
  - Most applications of ADP focus on the challenge of handling multidimensional state variables
  - Start with
    \[
    V_t(S_t) = \max_{x \in \mathcal{X}} \left( C_t(S_t, x_t) + E \{ V_{t+1}(S_{t+1}) | S_t \} \right)
    \]
  - Now replace the value function with some sort of approximation
    \[
    V_{t+1}(S_{t+1}) \approx \tilde{V}_{t+1}(S_{t+1}) = \sum_{f \in \mathcal{F}} \theta_f \phi_f (S_{t+1})
    \]
  - May draw from the entire field of statistics/machine learning.
Introduction to ADP

- But this does not solve our problem
  » Assume we have an approximate value function.
  » We still have to solve a problem that looks like

\[
V_t(S_t) = \max_{x \in \mathcal{X}} \left( C_t(S_t, x_t) + E \sum_{f \in \mathcal{F}} \theta_f \phi_f(S_{t+1}) \right)
\]

» This means we still have to deal with a maximization problem (might be a linear, nonlinear or integer program) with an expectation.
Do not use weather report

Use weather report

Forecast sunny .6

Forecast cloudy .3

Forecast rain .1

Schedule game

Cancel game

$2400

$3500

$2300

$200

$3500

-$200

-$200
$V_t(S_t) = \max_{x \in \mathcal{X}} \left( C_t(S_t, x_t) + E \left\{ V_{t+1}(S_{t+1}) \mid S_t \right\} \right)$
- **Decision nodes**
- **Outcome nodes**

**Forecast sunny .6**
- **Outcome nodes**
  - **Weather Forecast**
    - **Rain .8** - $2000
      - **Outcome nodes**
        - **Weather Forecast**
          - **Clouds .2** - $1000
          - **Weather Forecast**
            - **Sun .0** - $5000
          - **Weather Forecast**
            - **Rain .8** - $200
          - **Weather Forecast**
            - **Clouds .2** - $200
          - **Weather Forecast**
            - **Sun .0** - $200
          - **Weather Forecast**
            - **Rain .1** - $2000
          - **Weather Forecast**
            - **Clouds .2** - $1000
          - **Weather Forecast**
            - **Sun .4** - $5000
          - **Weather Forecast**
            - **Rain .1** - $200
          - **Weather Forecast**
            - **Clouds .5** - $1000
          - **Weather Forecast**
            - **Sun .4** - $200
    - **Weather Forecast**
      - **Weather Forecast**
        - **Rain .2** - $2000
        - **Weather Forecast**
          - **Clouds .3** - $1000
          - **Weather Forecast**
            - **Sun .5** - $5000
          - **Weather Forecast**
            - **Rain .2** - $2000
          - **Weather Forecast**
            - **Clouds .3** - $200
          - **Weather Forecast**
            - **Sun .5** - $200
    - **Weather Forecast**
      - **Weather Forecast**
        - **Rain .2** - $2000
        - **Weather Forecast**
          - **Clouds .3** - $1000
          - **Weather Forecast**
            - **Sun .5** - $5000
          - **Weather Forecast**
            - **Rain .2** - $2000
          - **Weather Forecast**
            - **Clouds .3** - $200
          - **Weather Forecast**
            - **Sun .5** - $200

**Forecast cloudy .3**
- **Outcome nodes**
  - **Weather Forecast**
    - **Rain .1** - $2000
      - **Outcome nodes**
        - **Weather Forecast**
          - **Clouds .5** - $1000
          - **Weather Forecast**
            - **Sun .4** - $5000
          - **Weather Forecast**
            - **Rain .1** - $200
          - **Weather Forecast**
            - **Clouds .5** - $1000
          - **Weather Forecast**
            - **Sun .4** - $200
      - **Weather Forecast**
        - **Rain .2** - $2000
        - **Weather Forecast**
          - **Clouds .3** - $1000
          - **Weather Forecast**
            - **Sun .5** - $5000
          - **Weather Forecast**
            - **Rain .2** - $2000
          - **Weather Forecast**
            - **Clouds .3** - $200
          - **Weather Forecast**
            - **Sun .5** - $200

**Forecast rain .1**
- **Outcome nodes**
  - **Weather Forecast**
    - **Rain .8** - $2000
      - **Outcome nodes**
        - **Weather Forecast**
          - **Clouds .2** - $1000
          - **Weather Forecast**
            - **Sun .0** - $5000
          - **Weather Forecast**
            - **Rain .8** - $200
          - **Weather Forecast**
            - **Clouds .2** - $200
          - **Weather Forecast**
            - **Sun .0** - $200
          - **Weather Forecast**
            - **Rain .1** - $2000
          - **Weather Forecast**
            - **Clouds .2** - $1000
          - **Weather Forecast**
            - **Sun .4** - $5000
          - **Weather Forecast**
            - **Rain .1** - $200
          - **Weather Forecast**
            - **Clouds .5** - $1000
          - **Weather Forecast**
            - **Sun .4** - $200
    - **Weather Forecast**
      - **Weather Forecast**
        - **Rain .2** - $2000
        - **Weather Forecast**
          - **Clouds .3** - $1000
          - **Weather Forecast**
            - **Sun .5** - $5000
          - **Weather Forecast**
            - **Rain .2** - $2000
          - **Weather Forecast**
            - **Clouds .3** - $200
          - **Weather Forecast**
            - **Sun .5** - $200
      - **Weather Forecast**
        - **Rain .2** - $2000
        - **Weather Forecast**
          - **Clouds .3** - $1000
          - **Weather Forecast**
            - **Sun .5** - $5000
          - **Weather Forecast**
            - **Rain .2** - $2000
          - **Weather Forecast**
            - **Clouds .3** - $200
          - **Weather Forecast**
            - **Sun .5** - $200
      - **Weather Forecast**
        - **Rain .2** - $2000
        - **Weather Forecast**
          - **Clouds .3** - $1000
          - **Weather Forecast**
            - **Sun .5** - $5000
          - **Weather Forecast**
            - **Rain .2** - $2000
          - **Weather Forecast**
            - **Clouds .3** - $200
          - **Weather Forecast**
            - **Sun .5** - $200

**Do not use weather report**

**Use weather report**

**Schedule game**

**Cancel game**
The post-decision state

■ “New” concept:

» The “pre-decision” state variable:
  • \( S_t \) = The information required to make a decision \( x_t \)
  • Same as a “decision node” in a decision tree.

» The “post-decision” state variable:
  • \( S_t^x \) = The state of what we know immediately after we make a decision.
  • Same as an “outcome node” in a decision tree.
  • Also known as:
    – “Afterstate” variable (Sutton and Barto)
    – “End-of-period state” (Judd)
The post-decision state

- Representations of the post-decision state:
  - Decision trees:
    \[ S_t^x = S_t^{M,x}(S_t, x_t) \]  
    Decision node $\rightarrow$ Outcome node
  
    \[ S_{t+1} = S_{t+1}^{M,W}(S_t^x, W_{t+1}) \]  
    Outcome node $\rightarrow$ Decision node
  
  - Q-learning:
    \[ S_t^x = (S_t, x_t) \]  
    State-action pair
  
  - Transition function with expectation:
    \[ S_t^x = S_t^M(S_t, x_t, \bar{W}_{t,t+1}) \]  
    $\bar{W}_{t,t+1} =$ Forecast of $W_{t+1}$ at time $t$. 
The post-decision state

- An inventory problem:
  - Our basic inventory equation:
    \[ R_{t+1} = \max\{0, R_t + x_t - \hat{D}_{t+1}\} \]
  - Where
    \[ R_t = \text{Inventory at time } t. \]
    \[ x_t = \text{Amount we ordered.} \]
    \[ \hat{D}_{t+1} = \text{Demand in next time period.} \]
  - Using pre- and post-decision states:
    \[ R_t^x = R_t + x_t \quad \text{Post-decision state} \]
    \[ R_{t+1} = \max\{0, R_t^x - \hat{D}_{t+1}\} \quad \text{Pre-decision state} \]
The post-decision state

- Pre-decision, state-action, and post-decision

<table>
<thead>
<tr>
<th>Pre-decision state</th>
<th>State</th>
<th>Action</th>
<th>Post-decision state</th>
</tr>
</thead>
<tbody>
<tr>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X</td>
<td>O</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td>O</td>
<td>X</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>3⁹ states</td>
<td>3⁹ × 9 state-action pairs</td>
<td>3⁹ states</td>
<td></td>
</tr>
</tbody>
</table>
The post-decision state

- Pre- and post-decision attributes for a trucking problem:

<table>
<thead>
<tr>
<th>Time</th>
<th>City</th>
<th>ETA</th>
<th>Equip</th>
<th>Decision</th>
<th>Post-decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>Dallas</td>
<td>41.2</td>
<td>Good</td>
<td>-</td>
<td>(Chicago)</td>
</tr>
<tr>
<td>40</td>
<td>Chicago</td>
<td>-</td>
<td>-</td>
<td>Good</td>
<td>(Repair)</td>
</tr>
<tr>
<td>50</td>
<td>Chicago</td>
<td></td>
<td></td>
<td></td>
<td>(Chicago)</td>
</tr>
</tbody>
</table>
The post-decision state

- Pre-decision: resources and demands

\[ S_t = (R_t, D_t) \]
The post-decision state

\[ S_t^x = S_t^{M,x} (S_t, x_t) \]
The post-decision state

\[ S_t^x = S_{t+1}^{M,W}(S_t^x, W_{t+1}) \]

\[ W_{t+1} = (\hat{R}_{t+1}, \hat{D}_{t+1}) \]
The post-decision state

$S_{t+1}$
The post-decision state

- Classical form of Bellman’s equation:
  \[ V_t(S_t) = \max_{x \in \mathcal{X}} \left( C_t(S_t, x_t) + E \{ V_{t+1}(S_{t+1}) \mid S_t \} \right) \]

- Bellman’s equations around pre- and post-decision states:
  - Optimization problem (making the decision):
    \[ V_t(S_t) = \max_x \left( C_t(S_t, x_t) + V^x_t \left( S_t^{M,x}(S_t, x_t) \right) \right) \]
    - Note: this problem is deterministic!
  - Simulation problem (the effect of exogenous information):
    \[ V^x_t(S^x_t) = E \{ V_{t+1}(S_t^{M,W}(S^x_t, W_{t+1})) \mid S^x_t \} \]
The post-decision state

**Challenges**

» For most practical problems, we are not going to be able to compute $V_t^x(S_t^x)$.

$$V_t(S_t) = \max_x \left( C_t(S_t, x_t) + V_t^x(S_t^x) \right)$$

» Concept: replace it with an approximation $\tilde{V}_t(S_t^x)$ and solve

$$V_t(S_t) = \max_x \left( C_t(S_t, x_t) + \tilde{V}_t(S_t^x) \right)$$

» So now we face:
  - What should the approximation look like?
  - How do we estimate it?
The post-decision state

- Value function approximations:
  - Linear (in the resource state):
    \[ \bar{V}_t (R^x_t) = \sum_{a \in \mathcal{A}} \bar{v}_{ta} \cdot R^x_{ta} \]
  - Piecewise linear, separable:
    \[ \bar{V}_t (R^x_t) = \sum_{a \in \mathcal{A}} \bar{V}_{ta} (R^x_{ta}) \]
  - Indexed PWL separable:
    \[ \bar{V}_t (R^x_t) = \sum_{a \in \mathcal{A}} \bar{V}_{ta} \left( R^x_{ta} \mid \text{features}_t \right) \]
The post-decision state

- Value function approximations:
  - Ridge regression (Klabjan and Adelman)
  - Benders cuts

\[
\bar{V}_t(R_t^x) = \sum_{f \in \mathcal{F}} \bar{V}_{tf} \left( \bar{R}_{tf} \right) \quad \bar{R}_{tf} = \sum_{a \in \mathcal{A}_f} \theta_{fa} R_{ta}
\]

- Benders cuts

![Diagram showing Value function approximations and corresponding equations]
The post-decision state

- Comparison to other methods:
  - Classical MDP (value iteration)
    \[ V^n(S) = \max_x \left( C(S, x) + \gamma E V^{n-1}(S_{t+1}) \right) \]
  - Classical ADP (pre-decision state):
    \[ \hat{v}_t^n = \max_x \left( C_t(S^n_t, x_t) + \sum p(s'| S^n_t, x_t) \overline{V}^{n-1}_{t+1}(s') \right) \]
    \[ \overline{V}^n_t(S^n_t) = (1 - \alpha_{n-1}) \overline{V}^{n-1}_t(S^n_t) + \alpha_{n-1} \hat{v}_t^n \]
  - Our method (update \( \overline{V}^{x,n-1}_t \) around post-decision state):
    \[ \hat{v}_t^n = \max_x \left( C_t(S^n_t, x_t) + \overline{V}^{x,n-1}_t(S^{M,x}_t (S^n_t, x_t)) \right) \]
    \[ \overline{V}^n_{t-1}(S^{x,n}_{t-1}) = (1 - \alpha_{n-1}) \overline{V}^{n-1}_{t-1}(S^{x,n}_{t-1}) + \alpha_{n-1} \hat{v}_t^n \]
    \[ \hat{v}_t \text{ updates } \overline{V}_{t-1}(S^{x}_{t-1}) \]
Approximate dynamic programming

Step 1: Start with a pre-decision state \( S^n_t \)

Step 2: Solve the deterministic optimization using an approximate value function:
\[
\hat{v}^n_t = \max_x \left( C_t(S^n_t, x_t) + \bar{V}^{n-1}_t(S_{t-1}^{x^n}) \right)
\]
to obtain \( x^n_t \).

Step 3: Update the value function approximation
\[
\bar{V}^{n}_{t-1}(S_{t-1}^{x^n}) = (1 - \alpha_n - 1) \bar{V}^{n-1}_{t-1}(S_{t-1}^{x^n}) + \alpha_n \hat{v}^n_t
\]

Step 4: Obtain Monte Carlo sample of \( W_t(\omega^n) \) and compute the next pre-decision state:
\[
S^n_{t+1} = S^M(S^n_t, x^n_t, W_{t+1}(\omega^n))
\]

Step 5: Return to step 1.
Approximate dynamic programming

Step 1: Start with a pre-decision state $S^n_t$

Step 2: Solve the deterministic optimization using an approximate value function:

$$\hat{v}^n_t = \max_{x_t} \left( C_t(S^n_t, x_t) + \hat{V}^{n-1}_t(S^{M,x}(S^n_t, x_t)) \right)$$

to obtain $x^n_t$. 

Deterministic optimization
Approximate dynamic programming

Step 1: Start with a pre-decision state $S^n_t$

Step 2: Solve the deterministic optimization using an approximate value function:
\[
\hat{\nu}^n_t = \max_x \left( C_t(S^n_t, x_t) + \bar{V}^{n-1}_t(S^{M,x}_t(S^n_t, x_t)) \right)
\]
to obtain $x^n_t$.

Step 3: Update the value function approximation
\[
\bar{V}^{n}_{t-1}(S^{x,n}_{t-1}) = (1 - \alpha_{n-1})\bar{V}^{n-1}_{t-1}(S^{x,n}_{t-1}) + \alpha_{n-1}\hat{\nu}^n_t
\]
Approximate dynamic programming combines simulation and optimization in a rigorous yet flexible framework.

### Optimization
- **Strengths**
  - Produces optimal decisions.
  - Mature technology.
- **Weaknesses**
  - Cannot handle uncertainty.
  - Cannot handle high levels of complexity.

### Simulation
- **Strengths**
  - Extremely flexible.
  - High level of detail.
  - Easily handles uncertainty.
- **Weaknesses**
  - Models decisions using user-specified rules.
  - Low solution quality.

---

Approximate Dynamic Programming
Solving the Curse of Dimensionality

Warren B. Powell
Outline

- Modeling resource allocation problems
- ADP and the post-decision state variable
- A blood management example
- A brief overview of CASTLE Lab research
- Applications
  - Trucking
  - Rail
  - Energy
Blood management

- Managing blood inventories

![Blood Type Compatibility Diagram]

Type of Donated Blood
- AB+
- AB-
- A+
- A-
- B+
- B-
- 0+
- 0-

Type of Recipient Blood
- AB+
- AB-
- A+
- A-
- B+
- B-
- 0+
- 0-
Blood management

- Managing blood inventories over time

Week 0

Week 1

Week 2

Week 3

\( t=0 \)

\( t=1 \)

\( t=2 \)

\( t=3 \)
$S_t = \left( R_t, \hat{D}_t \right)$

$R_{t,(AB+,0)} \rightarrow AB+,0$
$R_{t,(AB+,1)} \rightarrow AB+,1$
$R_{t,(AB+,2)} \rightarrow AB+,2$

$R_{t,(O-,0)} \rightarrow O-,0$
$R_{t,(O-,1)} \rightarrow O-,1$
$R_{t,(O-,2)} \rightarrow O-,2$

$\hat{D}_{t,AB+}$
$\hat{D}_{t,AB-}$
$\hat{D}_{t,A+}$
$\hat{D}_{t,A-}$
$\hat{D}_{t,B+}$
$\hat{D}_{t,B-}$
$\hat{D}_{t,O+}$
$\hat{D}_{t,O-}$

Satisfy a demand
Hold

$R_t^x$
$AB+,0$
$AB+,1$
$AB+,2$
$AB+,3$
$O-,0$
$O-,1$
$O-,2$
$O-,3$
$O-,4$
$O-,5$
$O-,6$

\[ \text{Satisfy a demand} \quad \text{Hold} \]
Solve this as a linear program.
Dual variables give value additional unit of blood..
Updating the value function approximation

- Estimate the gradient at $R^n_t$
Updating the value function approximation

- Update the value function at $R_{t-1}^{x,n}$
Updating the value function approximation

- Update the value function at $R_{t-1}^{x,n}$
Updating the value function approximation

- Update the value function at $R_{t-1}^{x,n}$
Iterative learning
Iterative learning
Iterative learning
Iterative learning
Approximate dynamic programming

- With luck, the objective function will improve steadily
Approximate dynamic programming

... but performance can be jagged.
Outline

- Modeling resource allocation problems
- ADP and the post-decision state variable
- A blood management example
- A brief overview of CASTLE Lab research
- Applications
  - Trucking
  - Rail
  - Energy
The challenges of ADP

Areas of research in my group:

» Convergence proofs for approximation strategies for resource allocation problems without requiring exploration steps

• There is a mature literature on methods using Benders cuts, but convergence slows dramatically when the dimensionality of the resource state variable grows.
• We have a series of papers proving convergence for:
  – SHAPE algorithm – For two-stage problems with continuously differentiable value functions.
  – CUPPS – For multistage nonseparable problems.
  – Piecewise linear separable – For two-stage separable and multistage scalar problems.
  – Convergence proof for approximate policy iteration algorithm for vector-valued continuous states and actions with known basis functions.
The challenges of ADP

Areas of research in my group:

» Stepsizes
  • Can’t live with ‘em, can’t live without ‘em.
  • Too small, you think you have converged but you have really just stalled (“apparent convergence”)
  • Too large, and the system is unstable.
  • Developed “optimal” stepsize rule which balances noise and bias:

\[
\alpha_n = 1 - \frac{\sigma^2}{(1 + \lambda^{n-1}) \sigma^2 + (\beta^n)^2}
\]

As \( \sigma^2 \) increases, stepsize decreases toward \( 1/n \)
As \( \beta^n \) increases, stepsize increases toward 1.

Observed values
BAKF stepsize rule
BAKF stepsize rule
1/n stepsize rule
The challenges of ADP

- Areas of research in my group:
  - Exploration vs. exploitation
    - One of the unsolved problems of life.
    - Have developed the “knowledge gradient” algorithm which estimates the marginal value of a single measurement on future decisions.

\[ \nu^KG_x = E\left\{\max_y F(y, K(x))\right\} - \max_y F(y, K) \]
The challenges of ADP

- Areas of research in my group:
  - Adaptive, hierarchical estimation of value functions
    - ADP needs to start with coarse approximations that it uses in the early iterations, and transition to more accurate approximations in later iterations.
    - Simple formula using estimate of bias and variance produces state-dependent weights that adapts to the number of observations.

\[
\bar{v}_a = \sum_g w_a^{(g)} v_a^{(g)}
\]

\[
w_a^{(g)} \propto \left( \text{Var}(\bar{v}_a^{(g)}) + (\beta^{(g)})^2 \right)^{-1}
\]

\[
\sum_g w_a^{(g)} = 1
\]
The challenges of ADP

- Areas of research in my group:
  
  » Machine learning
    
    - We are chasing the Holy Grail of a general purpose statistical method for approximating general functions.
    - We are looking into Dirichlet mixture models to produce a very general-purpose approximation technique which combines aggregation with generalized regression.
    - Completing general convergence proof.

\[
G \sim DP\left(G_0, \alpha \right) = (p_c, \theta_c)_{c=1,2,\ldots}
\]

\[
\theta_i = (\mu_i, \Sigma_i, \beta_{0:d,i}, \sigma_{\varepsilon,i}) | G \sim G
\]

\[
x_i | \theta_i \sim N\left(\mu_i, \Sigma_i\right)
\]

\[
y_i | x_i, \theta_i \sim N\left(\beta_{0,i} + \beta_{1:d}^t x_i, \sigma_{\varepsilon,i}^2\right)
\]
Outline

- Modeling resource allocation problems
- ADP and the post-decision state variable
- A blood management example
- A brief overview of CASTLE Lab research
- Applications
  - Trucking
  - Rail
  - Energy
Case study: truckload trucking

![Graph showing revenue per WU and utilization across capacity categories with historical data and simulation results.]

- **Revenue per WU**
  - Historical maximum
  - Simulation
  - Historical minimum

- **Utilization**
  - Historical maximum
  - Simulation
  - Historical minimum

---

**Equation:**

$$\text{Revenue per WU} = \text{Historical maximum} \times \text{Utilization}$$

---

**Calibrated model:**

- Historical data
- Simulation results
- Model validation
Case study: truckload trucking

Average LOH for Solos

Vanilla simulator

Using approximate dynamic programming

Acceptable region
Case study: truckload trucking

Simulation objective function

# of drivers with attribute a

- s1
- s2
- s3
- s4
- s5
- s6
- s7
- s8
- s9
- s10
- avg
- pred
Case study: truckload trucking

simulation objective function

# of drivers with attribute a

\( V_a \)
Case study: truckload trucking
Case study: truckload trucking
Case study: truckload trucking
An Approximate Dynamic Programming Algorithm for Large-Scale Fleet Management: A Case Application

Hugo P. Simão
Department of Operations Research and Financial Engineering, Princeton University,
Princeton, New Jersey 08544, hpsimao@princeton.edu

Jeff Day
Schneider National, Green Bay, Wisconsin 54306, dayj@schneider.com

Abraham P. George
Department of Operations Research and Financial Engineering, Princeton University,
Princeton, New Jersey 08544, ageorge@princeton.edu

Ted Gifford, John Nienow
Schneider National, Green Bay, Wisconsin 54306, giffordt@schneider.com, nienowj@schneider.com

Warren B. Powell
Department of Operations Research and Financial Engineering, Princeton University,
Princeton, New Jersey 08544, powell@princeton.edu
Outline

- Modeling resource allocation problems
- ADP and the post-decision state variable
- A blood management example
- A brief overview of CASTLE Lab research

Applications
  - Trucking
  - Rail
  - Energy
Locomotive optimization at Norfolk Southern

The value of six-axle high-adhesion locomotives in Baltimore

The value of locomotives in Baltimore
Locomotive optimization at Norfolk Southern

- Train delay curve – October 2007

![Train delay curve - October 2007](image)
Locomotive optimization at Norfolk Southern

- Train delay curve – March 2008
Locomotive optimization at Norfolk Southern

- Optimize over entire company at each point in time
Locomotive optimization at Norfolk Southern

- Decompose by “desk” (region)
Locomotive optimization at Norfolk Southern

- System vs. “Desk”

![Graph showing train delay vs. fleet size](image-url)
Locomotive optimization at Norfolk Southern

- System vs. “Desk”

![Graph showing train delay vs. fleet size](image-url)
Outline

- Modeling resource allocation problems
- ADP and the post-decision state variable
- A blood management example
- A brief overview of CASTLE Lab research
- Applications
  - Trucking
  - Rail
  - Energy
The energy resource planning problem
Wind
Wind

30 days

1 year
Storage

Hydroelectric

Flywheels

Batteries

Ultracapacitors
Energy resource modeling

- Hydro.Rsrvr --> Hydro.Elect
- Wind.Rsrc --> Wind.Elect
- Solar.Rsrc --> Solar.Elect
- Nucl.Rsrc --> Nucl.Elect
- Coal.SW.Rsrc
- Coal.Pulv.Elect.SW
- Coal.IGCC.Elect.SW
- Coal.IGCC.Crb.Cpt.SW
- Elect.Trans.CV
- Resid.Elect.Dmnd
- Export.Elect.Dmnd
- Comm.Elect.Dmnd
- Indust.Elect.Dmnd

Intermediate node:
- Capt.CO2
- Vent.CO2
Energy resource modeling

- **Hourly model**
  - Decisions at time $t$ impact $t+1$ through the amount of water held in the reservoir.
Energy resource modeling

- **Hourly model**
  - Decisions at time $t$ impact $t+1$ through the amount of water held in the reservoir.

**Hour t**

Value of holding water in the reservoir for future time periods.
Energy resource modeling
Energy resource modeling
Energy resource modeling
Energy resource modeling

2008

2009

...
Energy resource modeling

\begin{itemize}
\item \( \text{oil} \)
\item \( \text{wind} \)
\item \( \text{nat gas} \)
\item \( \text{coal} \)
\end{itemize}

\begin{align*}
\text{2008} & \quad 2009 \\
& \quad \nu_{\text{oil}}^{2009} \\
& \quad \nu_{\text{wind}}^{2009} \\
& \quad \nu_{\text{nat gas}}^{2009} \\
& \quad \nu_{\text{coal}}^{2009}
\end{align*}
Energy resource modeling
Energy resource modeling
Energy resource modeling

~5 seconds ~5 seconds ~5 seconds ~5 seconds ~5 seconds • • • 2038

2008 2009 2010 2011

Slide 135
Energy resource modeling

- Use statistical methods to learn the value of resources in the future.
- Resources may be:
  - Stored energy
    - Hydro
    - Flywheel energy
    - …
  - Storage capacity
    - Batteries
    - Flywheels
    - Compressed air
  - Energy transmission capacity
    - Transmission lines
    - Gas lines
    - Shipping capacity
  - Energy production sources
    - Wind mills
    - Solar panels
    - Nuclear power plants

Unlike our transportation applications, these functions are continuous.
Energy resource modeling

- Benchmarking
  - Compare ADP to optimal LP for a deterministic problem
    - Annual model
      - 8,760 hours over a single year
      - Focus on ability to match hydro storage decisions
    - 20 year model
      - 24 hour time increments over 20 years
      - Focus on investment decisions
  - Comparisons on stochastic model
    - Stochastic rainfall analysis
      - How does ADP solution compare to LP?
    - Carbon tax policy analysis
      - Demonstrate nonanticipativity
Energy resource modeling

- ADP objective function relative to optimal LP

Percentage error from optimal

Iterations

0.06% over optimal
Energy resource modeling

- Optimal from linear program
Energy resource modeling

- Approximate dynamic programming

![Graph showing ADP solution, Reservoir level, Rainfall, and Demand](image-url)
Energy resource modeling

- Optimal from linear program
Energy resource modeling

- Approximate dynamic programming

![Graph showing ADP solution, reservoir level, rainfall, and demand.](image)
Annual energy model

- ADP vs optimal reservoir levels for stochastic rainfall

**ADP at last iteration**

**Optimal for individual scenarios**
The challenge of ADP/RL

“I think you give a too rosy a picture of ADP….”
Andy Barto, in comments on a recent paper

\[ \tau = 1, \ldots, \tau^{ph}. \] To run an ADP algorithm, we have to assume a distribution (for example, normal with mean \( F_{tt'} \) and variance \( \bar{\sigma}_{t,t'-t}^2 \)) to generate random observations of demands.

13.7 IF IT WORKS, PATENT IT!

A successful ADP algorithm for an important problem class is, we believe, a patentable invention. While we continue to search for the holy grail of a general class of algorithms that work reliably on all problems, we suspect the future will involve fairly specific problem classes with a well-defined structure. For such problems, we can, with some confidence, perform a set of experiments that can determine whether an ADP algorithm solves the problems that are likely to come up in that class. Of course, we encourage developers to keep these results in the public domain, but our point is that these represent specific technological breakthroughs which could form the basis of a patent application.