Markov Decision Processes and their Applications to Supply Chain Management

Jefferson Huang

School of Operations Research & Information Engineering
Cornell University

June 24 & 25, 2018

10th Operations Research & Supply Chain Management (ORSCM) Workshop

National Chiao-Tung University (Taipei Campus)

Taipei, Taiwan

Outline of the (Mini-)Course

1. Examples of SCM[1] Problems Where MDPs[2] Were Useful
2. The MDP Model
3. Performance Measures
4. Performance Evaluation
5. Optimization
6. Additional Topics

[1]SCM = Supply Chain Management
[2]MDPs = Markov Decision Processes

Sourcing from Multiple Suppliers

- Demand for one product over time
- Two procurement options (Dual-Sourcing):
  1. "Regular" Source: cheaper, slower shipping
  2. "Expedited" Source: more expensive, faster shipping
- Revenue from sales
- Costs:
  1. Ordering Cost (cost to place an order)
  2. Holding Cost (cost of ordering too much)
  3. Shortage Cost (cost of ordering too little)

$\boxed{\text{How should the two sourcing options be used?}}$

G. Allon, J. A. Van Mieghem, (2010) Global Dual Sourcing: Tailored Base-Surge Allocation to Near- and Offshore Production. Management Science 56(1):110-124.

S. Veeraraghavan, A. Scheller-Wolf, (2008) Now or Later: A Simple Policy for Effective Dual Sourcing in Capacitated Systems. Operations Research 56(4):850-864.

L. Xin, D. A. Goldberg (2018) Asymptotic Optimality of Tailored Base-Surge Policies in Dual-Sourcing Inventory Systems. Management Science 64(1):437-452.

Routing Inventory to Customers

- Demand for one product over time
- Demands originate from different locations
  - Delivery vehicles have limited carrying capacity
- Costs:
  1. Ordering Cost (cost to place an order)
  2. Holding Cost (cost of ordering too much)
  3. Shortage Cost (cost of ordering too little)

How should the vehicles be stocked and routed?

D. Adelman, (2004) A Price-Directed Approach to Stochastic Inventory/Routing. Operations Research 52(4):499-514.

Deciding What Assortment of Products to Sell

- Demands for a finite number S of products in each period
- Finite number of selling periods
- At the start of each period, select N out of the S products to offer for that period
    1. Revenue from sales
    2. Costs to switch between assortments

What product assortments should be offered, and when?

F. Caro, J. Gallien, (2007) Dynamic Assortment with Demand Learning for Seasonal Consumer Goods. Management Science 53(2):276-292.

Common Features of the Problems

1. Decisions are made over time.
    - Once every period
2. Decisions can be based on observations.
    - Current inventory position (on-hand + on-order - backorders)
    - Past sales of different products
3. Depending on the observations, certain actions make more sense than others.
    - No infinite or negative order quantities
    - Can't load vehicles beyond their capacity
    - Limited shelf space
4. Each action has a reward.
    - Revenue - Costs (ordering, holding, backordering)
    - Switching between assortments
5. Taking an action will affect what you observe next.
    - Next period's inventory position
    - Amounts of each product sold

Part 1

The Markov Decision Process (MDP) Model

Markov Decision Process (MDP) Model

At each time $t = 1, 2, \ldots,$

1. observe the current state $x_t \in \mathbb{X}$
2. select an action $a_t \in A(x_t)$
3. earn a reward $r(x_t, a_t)$
4. the next state is x_{t+1} with probability $p(x_{t+1} | x_t, a_t)$

Example: Single-Sourcing

Periodically procure a single product to sell.

In each period, the following sequence of events occurs:

1. Place an order, which arrives instantaneously.
   - Per-unit ordering cost is c.
2. The demand for that period is revealed, satisfied from on-hand inventory.
   - Each unit sold earns p.
   - Each unsold unit is assessed a holding cost h.
   - Each unsatisfied unit of demand is assessed a shortage cost b.

The demands are independent and identically distributed (iid) between periods.

Example: Single-Sourcing

State Set:

$$\boxed{\mathbb{X} = \mathbb{R}}$$

- State is the current net inventory level (on-hand minus backorders)

Action Sets:

$$\boxed{A(x) = [0, \infty) =: \mathbb{R}_+}$$

- Action is the quantity ordered

Example: Single-Sourcing

One-Step Rewards:

Since we'll focus on *expected costs*, we let $r(x, a)$ be the expected reward earned when action a is taken in state x.

$$\boxed{\begin{aligned} r(x,a) = -ca + p\mathbb{E}[\min\{D, x+a\}] \\ - h\mathbb{E}[(x+a-D)^+] - b\mathbb{E}[(D-x-a)^+] \end{aligned}}$$

Transition Probabilities:

Let F_D denote the distribution of D.

If the current state is x, and action a is taken, the next state is on the interval $[a, b] \in \mathbb{R}$ with probability

$$\boxed{p([a,b]|x,a) = \int_{\mathbb{R}} \mathbf{1}\{x + a - u \in [a,b]\}\, dF_D(u)}$$

Example: Dual-Sourcing

L_R = lead time for the regular source $\geqslant 1$

L_E = lead time for the expedited source $\geqslant 1$

In each period, the following sequence of events occurs:

1. Place desired orders with the regular supplier and the expedited supplier.
2. Receive (1) the order placed L_R periods ago from the regular supplier, and (2) the order placed L_E periods ago from the expedited supplier.
3. The demand for that period is revealed, and satisfied from on-hand inventory.

Suppose that the demands D are iid between periods, and that we want to minimize expected costs.

Example: Dual-Sourcing

State Set:

$$\boxed{\mathbb{X} = \mathbb{R} \times \mathbb{R}_+^{L_R} \times \mathbb{R}_+^{L_E}}$$

- State $[i, (y_1, \ldots, y_{L_R}), (z_1, \ldots, z_{L_E})]$ means::
  - current inventory level is $i \in \mathbb{R}$
  - for $j = 1, \ldots, L_R$, an order of y_j units from the regular source was placed j periods ago
  - for $j = 1, \ldots, L_E$ an order of z_j units from the expedited source was placed j periods ago

Action Sets:

$$\boxed{A(x) = \mathbb{R}_+ \times \mathbb{R}_+ \qquad \text{for all } x \in \mathbb{X}}$$

- Amounts to order from each source

Example: Dual-Sourcing

p = per-unit selling price

c_R = per-unit ordering cost for the regular source

c_E = per-unit ordering cost for the expedited source

h = cost to hold 1 unit in inventory for 1 period

b = per-unit backordering cost

One-Step Rewards:

For $x = [i, (y_1, \ldots, y_{L_R}), (z_1, \ldots, z_{L_E})] \in \mathbb{X}$ and $a = [a_R, a_E] \in A(x)$,

$$\boxed{\begin{aligned} r(x,a) = & -c_R a_R - c_E a_E \\ & + p\mathbb{E}[\min\{D, i + y_{L_R} + z_{L_E}\}] \\ & - h\mathbb{E}[(i + y_{L_R} + z_{L_E} - D)^+] \\ & - b\mathbb{E}[(D - i - y_{L_R} - z_{L_E})^+] \end{aligned}}$$

Example: Dual-Sourcing

Recall that demand D is iid between periods.

- Suppose the demand is discrete, and that $p_d := \mathbb{P}\{D = d\}$.

Transition Probabilities:

For $x = [i, (y_1, \ldots, y_{L_R}), (z_1, \ldots, z_{L_E})] \in \mathbb{X}$ and $a = [a_R, a_E] \in A(x)$, the next state is

$$y = [i + y_{L_R} + z_{L_E} - d, (a_R, y_1, \ldots, y_{L_R - 1}), (a_E, z_1, \ldots, z_{L_E - 1})]$$

with probability

$$\boxed{p(y|x, a) = p_d.}$$

Selecting Actions

We assume the decision-maker follows a policy for selecting actions.

A policy is a rule for selecting actions.

- Let π denote a policy.

If π is used, actions are selected as follows:

if the history up to time t is $x_0 \ldots x_{t-1} a_{t-1} x_t$

then select action $a_t \in A(x_t)$ with probability $\pi_t(a_t | x_0 \ldots x_{t-1} a_{t-1} x_t)$.

- In general, policies may be randomized and history-dependent.

Important Classes of Policies

Markov Policies

- Actions are selected based on the current time and current state only:

$$\pi_t(a_t|x_0 \ldots x_{t-1}a_{t-1}x_t) = \pi_t(a_t|x_t)$$

Stationary Policies

- Actions are selected based on the current state only:

$$\pi_t(a_t|x_0 \ldots x_{t-1}a_{t-1}x_t) = \pi(a_t|x_t)$$

Deterministic Stationary Policies

- Indentified with functions $\varphi : \mathbb{X} \to \cup_{x\in\mathbb{X}} A(x)$ where $\varphi(x) \in A(x)$ for all $x \in \mathbb{X}$.
- Whenever the current state is $x \in \mathbb{X}$, select action $\varphi(x)$.

Example: Single-Sourcing

Some possible policies:

- Base-Stock Policy: Whenever the current net inventory level is less than s, order up to s.
- (s, S)-Policy: Whenever the current net inventory level is less than s, order up to S.

Both of these are deterministic stationary policies.

Example: Dual-Sourcing

$$\begin{aligned} IP &= \text{inventory position} \\ &= (\text{inventory level}) + (\text{all orders to arrive in the next } L_R \text{ periods}) \end{aligned}$$

Some possible policies:

- Fixed Ratio Base-Stock Policy: Whenever $IP < \underline{x}$, order $\alpha_R(\underline{x} - IP)$ from the regular source and $\alpha_E(\underline{x} - IP)$ from the expedited source, where $\alpha_R + \alpha_E = 1$.
- Tailored Base-Surge Policy: In each period, order the same amount from the regular source. Whenever the $IP < \underline{x}$, order up to $\underline{x}$ from the expedited source.
- Dual-Index Policy: Use one base-stock level with IP for ordering from the regular source, and use another base-stock level with

  $$IP_E = (\text{inventory lv.}) + (\text{all orders to arrive in next } L_E \text{ periods})$$

  for ordering from the expedited source.

Part 2

Performance Measures

Performance Measure: Finite-Horizon Total Reward

$\mathbb{E}_x^\pi$ = expectation given that the initial state is x and the policy π is used

T = time horizon $\geqslant 1$

X_t = state at time t

A_t = action taken at time t

Finite-Horizon Expected Total Reward:

$$\boxed{V_T^\pi(x) := \mathbb{E}_x^\pi \sum_{t=0}^{T-1} r(X_t, A_t), \qquad x \in \mathbb{X}}$$

Performance Measure: Discounted Total Reward

$\mathbb{E}_x^\pi$ = expectation given that the initial state is x and the policy π is used

β = discount factor (between 0 and 1)

X_t = state at time t

A_t = action taken at time t

Infinite-Horizon Expected Discounted Total Reward:

$$\boxed{v_\beta^\pi(x) := \mathbb{E}_x^\pi \sum_{t=0}^{\infty} \beta^t r(X_t, A_t), \qquad x \in \mathbb{X}}$$

Performance Measure: Average Reward

$\mathbb{E}_x^\pi =$ expectation given that the initial state is x and the policy π is used

$X_t =$ state at time t

$A_t =$ action taken at time t

Long-Run Expected Average Reward:

$$\boxed{w^\pi(x) := \limsup_{T\to\infty} \frac{1}{T} \mathbb{E}_x^\pi \sum_{t=0}^{T-1} r(X_t, A_t) \qquad x \in \mathbb{X}}$$

Part 3

Performance Evaluation

Performance Evaluation: Finite-Horizon Total Reward

Suppose a Markov policy π is used.

- The state process $X_0, X_1, \ldots$ is a (possibly nonhomogeneous) Markov chain.

V_T^π can be computed using backward induction:

$$V_0^\pi(x) = 0, \qquad x \in \mathbb{X}$$

$$V_t^\pi(x) = \mathbb{E}_x^\pi[r(X_0, A_0)] + \mathbb{E}_x^\pi[V_{t-1}^\pi(X_1)], \quad x \in \mathbb{X},\ t = 1, \ldots, T$$

Performance Evaluation: Discounted Reward

Suppose a stationary policy π is used.

- The state process $X_0, X_1, \dots$ is a homogeneous Markov chain.

When the one-step rewards $r(x,a)$ are bounded, we can approximate v_β^π using value iteration: Letting

$$f_0(x) := 0, \qquad x \in \mathbb{X}$$

and

$$f_N(x) := \mathbb{E}_x^\pi[r(X_0, A_0)] + \beta \mathbb{E}_x^\pi[f_{N-1}(X_1)], \qquad x \in \mathbb{X},$$

we get

$$\boxed{v_\beta^\pi(x) = \lim_{N\to\infty} f_N(x) \text{ for all } x \in \mathbb{X}.}$$

Performance Evaluation: Discounted Reward

If the state and action sets are finite, can π can be evaluated by inverting a matrix:

P_π = transition matrix of the homogeneous Markov chain induced by π

r_π = vector with components $r_\pi(x) := \sum_{a \in A(x)} r(x,a)\pi(a|x)$ for $x \in \mathbb{X}$

$I = |\mathbb{X}| \times |\mathbb{X}|$ identity matrix

v_β^π = vector with components $v_\beta^\pi(x)$ for $x \in \mathbb{X}$

Then

$$\boxed{v_\beta^\pi = (I - \beta P_\pi)^{-1} r_\pi}$$

Performance Evaluation: Average Reward

Suppose a stationary policy π is used.

- The state process $X_0, X_1, \ldots$ is a homogeneous Markov chain P_π.

To evaluate π, we need to understand the structure of P_π.

Suppose the state space $\mathbb{X}$ is finite.

- P_π is unichain if it has a single recurrent class (may have transient states)
- P_π is multichain if it has at least two recurrent classes

Performance Evaluation: Average Reward

If P_π is unichain, then:

- it has a unique stationary distribution ν^π
- the average reward under π is independent of the initial state, and is equal to

$$w^\pi := \sum_{x \in \mathbb{X}} \nu^\pi(x) \sum_{a \in A(x)} \pi(a|x) r(x,a)$$

- If the constant w and the function h satisfy the evaluation equations

$$w + h(x) = \sum_{a \in A(x)} \pi(a|x) r(x,a) + \sum_{y \in \mathbb{X}} \sum_{a \in A(x)} \pi(a|x) p(y|x,a) h(y)$$

  for all $x \in \mathbb{X}$, then $w = w^\pi$.

If P_π is multichain, the average cost depends on the initial state.

- We won't dwell on this case.

Example: Single-Sourcing

For the computational examples, we'll assume the following:

- No fractional orders, e.g., each order is for a certain number of units (e.g., pallets) of the product.
- The warehouse has a finite capacity.
- There is a limit to how much can be ordered at once.
- The demand is discrete.
- All unmet demand is lost.

Under these assumptions, the single-sourcing problem is an MDP with finite state and action sets.

Example: Single-Sourcing

Consider the following instance of the aforementioned **capacitated single-sourcing** problem with **discrete demand** and **lost sales**:

- The warehouse can only store 2 pallets of the product.
- At most 2 pallets can be ordered at once.
- The demand is uniformly distributed on $\{0, 1, 2\}$.
- Rewards/Costs:
  - Selling price: $p = \$4$ per unit
  - Holding cost rate: $h = \$1$ per unit per period
  - Ordering cost: $c = \$2$ per unit

Let π denote the base-stock policy with base-stock level $s = 2$.

Example: Single-Sourcing

To make evaluating π easier, note that under π the process is a Markov reward process.

- i.e., the state evolves according to a Markov chain P_π, and a reward $r_\pi(x)$ is earned whenever state x is visited.

For the base-stock policy with base-stock level $s = 2$,

$$P_\pi = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

$$r_\pi = \begin{bmatrix} -1 \\ 1 \\ 3 \end{bmatrix}$$

Example: Single-Sourcing

Finite-Horizon Total Reward:

$$V_T^\pi = r_\pi + P_\pi r_\pi + P_\pi^2 r_\pi + \cdots + P_\pi^{T-1} r_\pi$$

To avoid having to compute powers of P_π, use backward induction:

$$V_0^\pi = \mathbf{0} \qquad \text{(the zero vector)}$$

$$V_t^\pi = r_\pi + P_\pi V_{t-1}^\pi, \qquad t = 1, \ldots, T$$

When the base-stock policy π is followed for $T = 3$ periods, the vector of expected total rewards is

$$\boxed{V_T^\pi = \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix}}$$

Example: Single-Sourcing

Discounted Total Reward:

$$v_\beta^\pi = r_\pi + \beta P_\pi r_\pi + \beta^2 P_\pi^2 r_\pi + \cdots = (I - \beta P_\pi)^{-1} r_\pi$$

When the base-stock policy π is followed, and the discount factor is $\beta = 0.9$, the vector of expected discounted total rewards is

$$\boxed{v_\beta^\pi} = \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \beta \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix} \right)^{-1} \begin{bmatrix} -1 \\ 1 \\ 3 \end{bmatrix} = \boxed{\begin{bmatrix} 8 \\ 10 \\ 12 \end{bmatrix}}$$

Example: Single-Sourcing

Average Reward:

$$w^{\pi} = \limsup_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} P_{\pi}^{t} r_{\pi}$$

When the base-stock policy π is followed, the state process is an irreducible finite-state Markov chain P_{π}. This means:

- P_{π} has a unique stationary distribution ν_{π}.
- By the Strong Law of Large Numbers for Markov chains,

$$w^{\pi} = \sum_{y \in \mathbb{X}} \nu_{\pi}(y) r_{\pi}(y) \qquad \forall x \in \mathbb{X}.$$

(Here w^{π} is a constant, not a vector.)

Since the stationary distribution of P_{π} is $\nu_{\pi} = [1/3, 1/3, 1/3]$,

$$\boxed{w^{\pi} = v_{\pi} r_{\pi} = 1}$$

Example: Single-Sourcing

If the MDP is unichain, we can avoid dealing with the stationary distribution by using the evaluation equations to find w^π:

$$w^\pi \mathbf{1} + h = r_\pi + P_\pi h \tag{1}$$

($\mathbf{1}$ = vector of all ones)

In particular:

1. Select any state x_* and set $h(x_*) = 0$.
2. Solve the system (1) with this constraint.

For our problem, setting $x_* = 0$ turns (1) into

$$\begin{bmatrix} 1 & -\frac{1}{3} & -\frac{1}{3} \\ 1 & 1-\frac{1}{3} & -\frac{1}{3} \\ 1 & -\frac{1}{3} & 1-\frac{1}{3} \end{bmatrix} \begin{bmatrix} w^\pi \\ h(1) \\ h(2) \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ 3 \end{bmatrix} \implies h = \begin{bmatrix} 0 \\ 2 \\ 4 \end{bmatrix}, \ \boxed{w^\pi = 1}$$

Part 4

Optimization

Optimal Policies

Let J denote an optimality criterion (e.g., V_T, v_β, or w).

A policy π_* is optimal with respect to the optimality criterion associated with J if

$$J^{\pi_*}(x) = \max_\pi J^\pi(x) \qquad \text{for all } x \in \mathbb{X}.$$

If the state and action sets are finite, then:

- Under finite-horizon, discounted, and average rewards, there exists an optimal Markov policy.
- Under both discounted and average rewards, this optimal Markov policy can be taken to be deterministic and stationary.

There is a long line of mathematical work (from the 1960s to the present) on the existence of optimal policies for MDPs with infinite state or action spaces

E. A. Feinberg, P. O. Kasyanov, N. V. Zadoianchuk, (2012) Average cost Markov decision processes with weakly continuous transition probabilities. Mathematics of Operations Research 37(4):591-607.

Computing Optimal Policies

Main Approaches:

1. Value Iteration
2. Policy Iteration (Discounted, Average)
3. Linear Programming (Discounted, Average)
    - If the simplex method is used, this is equivalent to Policy Iteration.

Practical Considerations:

- When computationally feasible, policy iteration is typically the best option (especially for discounted MDPs).

  M. L. Puterman (2005). Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons.

- For many problems of interest, the MDP is just too big for these exact methods to work, but approximate versions of them can be useful.

  W. B. Powell (2007). Approximate Dynamic Programming: Solving the Curses of Dimensionality. John Wiley & Sons.

- Sometimes, there are optimal policies whose structure makes them particularly easy to implement.

  J. Gittins, K. Glazebrook, & R. Weber (2011). Multi-Armed Bandit Allocation Indices. John Wiley & Sons.

Optimization: Finite-Horizon Total Reward

Assume the state and action sets are finite.

Theorem (Bellman, 1957)

There exists an optimal Markov policy that is deterministic.

Proof Sketch. For any function $f : \mathbb{X} \to \mathbb{R}$, let

$$Tf(x) := \max_{a \in A(x)} \left[r(x,a) + \sum_{y \in \mathbb{X}} p(y|x,a) f(y) \right].$$

Let $V_0 = \mathbf{0}$ and

$$V_t = TV_{t-1}, \qquad t = 1, \ldots, T.$$

Let π be any deterministic Markov policy that satisfies

$$\pi_t(x) \in \arg\max_{a \in A(x)} \left[r(x,a) + \sum_{y \in \mathbb{X}} p(y|x,a) V_{t-1}(y) \right], \qquad t = 1, \ldots, T.$$

Then π is optimal. $\square$

Optimization: Finite-Horizon Total Reward

The backward induction algorithm used in the proof (sketch) can be used to compute an optimal deterministic Markov policy.

To simplify the statement of the algorithm, for $f : \mathbb{X} \to \mathbb{R}$ let $\mathcal{G}(f)$ denote the set of all "greedy" decision rules φ with respect to f, where

$$\varphi(x) \in \arg\max_{a \in A(x)} \left[r(x,a) + \sum_{y \in \mathbb{X}} p(y|x,a) f(y) \right] \qquad \forall x \in \mathbb{X}.$$

Backward Induction

```
1: Set $V_0 = \mathbf{0}$.
2: **for** $t = 1, \dots, T$ **do**
3:     Set $V_t = TV_{t-1}$.
4:     Select any $\pi_t^* \in \mathcal{G}(V_{t-1})$.
5: **return** the policy $\pi^* = \{\pi_1^*, \dots, \pi_T^*\}$.
```

Example: Single-Sourcing

Let's revisit the capacitated single-sourcing problem with discrete demand and lost sales, described on Slides 29 and 30.

Compute an optimal policy for $T = 3$: Letting $V_0 = \mathbf{0}$,

$$V_1 = \begin{bmatrix} 0.33 \\ 2.33 \\ 3 \end{bmatrix} \implies \pi_1^* = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},$$

$$V_2 = \begin{bmatrix} 1.33 \\ 3.33 \\ 4.89 \end{bmatrix} \implies \pi_2^* = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},$$

$$V_3 = \begin{bmatrix} 2.33 \\ 4.33 \\ 6.19 \end{bmatrix} \implies \pi_3^* = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

Optimization: Discounted Reward

Assume the state and action sets are finite.

Theorem (Howard, 1960)

For any $\beta \in [0,1)$, there exists an optimal policy that is deterministic and stationary.

Proof (Sketch). Consider any $\beta \in [0,1)$. The value function

$$v_\beta(x) := \inf_\pi v_\beta^\pi(x)$$

uniquely satisfies the optimality equation

$$v_\beta(x) = \max_{a \in A(x)} \left[r(x,a) + \beta \sum_{y \in \mathbb{X}} p(y|x,a) v_\beta(y) \right], \qquad x \in \mathbb{X}.$$

Moreover, a deterministic stationary policy π is optimal if and only if

$$\pi(x) \in \arg\max_{a \in A(x)} \left[r(x,a) + \beta \sum_{y \in \mathbb{X}} p(y|x,a) v_\beta(y) \right], \qquad x \in \mathbb{X}. \quad \square$$

Optimization: Discounted Reward

Value Iteration:

Consider any $f : \mathbb{X} \to \mathbb{R}$. Let $\|f\|_\infty := \max_{x \in \mathbb{X}} |f(x)|$ for $x \in \mathbb{X}$, and

$$T_\beta f(x) := \max_{a \in A(x)} \left[r(x,a) + \beta \sum_{y \in \mathbb{X}} p(y|x,a) f(y) \right], \quad x \in \mathbb{X}.$$

The operator T_β is a contraction mapping, i.e., for all real-valued functions f, g on $\mathbb{X}$,

$$\|T_\beta f - T_\beta g\|_\infty \leqslant \beta \|f - g\|_\infty.$$

Hence, letting $v_0 := \mathbf{0}$ and $v_k := (T_\beta)^k v_0$ for $k = 1, 2, \ldots,$ we have

$$\|v_\beta - v_k\|_\infty \leqslant \beta^k \|v_\beta - v_0\|_\infty, \quad k = 1, 2, \ldots$$

Optimization: Discounted Reward

Based on the preceding the following algorithm can be used to compute an ϵ-optimal policy, i.e., a policy π_ϵ such that

$$v_\beta^{\pi_\epsilon}(x) \geqslant v_\beta(x) - \epsilon \qquad \forall x \in \mathbb{X},$$

for any $\epsilon > 0$.

Value Iteration

```
1: Select any $\epsilon > 0$.
2: Set $v_0 = \mathbf{0}$.
3: Set $k = 0$.
4: do
5:     Set $v_{k+1} = T_\beta v_k$.
6:     Set $k = k + 1$.
7: while $\|v_k - v_{k-1}\|_\infty \geqslant \epsilon(1-\beta)/(2\beta)$
8: return any policy $\pi_\epsilon \in \mathcal{G}(v_k)$
```

Optimization: Discounted Reward

Policy Iteration:

If it's better to switch to another policy for one step, then it's better to use that policy all the time.

Let φ and ψ be two deterministic stationary policies, and let (φ, ψ) denote the policy that uses φ for one step, and then follows ψ.

Lemma

If $v_\beta^{(\varphi,\psi)} \geqslant v_\beta^\psi$, then $v_\beta^\varphi \geqslant v_\beta^\psi$. ($f \geqslant g$ means $f(x) \geqslant g(x)$ for all x)

Proof (Sketch). We have

$$r_\varphi + \beta P_\varphi v_\beta^\psi = v_\beta^{(\varphi,\psi)} \geqslant v_\beta^\psi \implies r_\varphi \geqslant (I - \beta P_\varphi) v_\beta^\psi.$$

Since $(I - \beta P_\varphi)^{-1} = \sum_{t=0}^\infty \beta^t P_\varphi^t$ is a nonnegative matrix,

$$v_\beta^\varphi = (I - \beta P_\varphi)^{-1} r_\varphi \geqslant v_\beta^\psi.$$

□

Optimization: Discounted Reward

For a deterministic stationary policy ϕ and $f : \mathbb{X} \to \mathbb{R}$,

$$T_\beta^\phi f := r_\phi + \beta P_\phi f$$

Policy Iteration

```
1: Select any deterministic stationary policy $\varphi$.
2: do
3:     Compute $v_\beta^\varphi = (I - \beta P_\varphi)^{-1} r_\varphi$.
4:     Select a policy $\varphi_+$ where $T_\beta^{\varphi+} v_\beta^\varphi = T_\beta v_\beta^\varphi$, setting $\varphi_+ = \varphi$ if possible.
5: while $\varphi^+ \neq \varphi$
6: return $\varphi$
```

Theorem (Howard, 1960)

The policy iteration agorithm terminates after a finite number of steps with an optimal policy.

Optimization: Average Cost

Assume the state and action sets are finite.

Theorem (Derman, 1962)

There exists an optimal policy that is deterministic and stationary.

Proof (Sketch). For $\beta \in [0,1)$, let π_β be an optimal deterministic stationary (DS) policy under discounted costs.

Since the set of DS policies is finite, there exists a sequence $\{\beta_n\} \subseteq [0,1)$ such that $\beta_n \to \infty$ and $\pi_{\beta_n} = \pi_*$, for some DS policy π_*. Moreover, since the state and action sets are finite, it follows from a "Tauberian theorem" that

$$\lim_{n\to\infty} (1-\beta_n) v_{\beta_n}(x) = w^{\pi_*}(x) \qquad \forall x \in \mathbb{X}.$$

So, for every policy π, the aforementioned "Tauberian theorem" implies that

$$w^{\pi}(x) \geqslant \limsup_{n\to\infty} (1-\beta_n) v^{\pi}_{\beta_n}(x) \geqslant \lim_{n\to\infty} (1-\beta_n) v_{\beta_n}(x) = w^{\pi_*}(x) \quad \forall x \in \mathbb{X}.$$

Hence the DS policy π_* is optimal. $\square$

Optimization: Average Cost

Assume the state and action sets are finite, and that the MDP is unichain.

Policy Iteration

```
1: Select any deterministic stationary policy $\varphi$.
2: do
3:     Compute $(w^\pi, h)$ satisfying $w^\varphi \mathbf{1} + h = r_\varphi + P_\varphi h$.
4:     Select a policy $\varphi_+$ where $r_{\varphi_+} + P_{\varphi_+} h = Th$, setting $\varphi_+ = \varphi$ if possible.
5: while $\varphi^+ \neq \varphi$
6: return $\varphi$
```

Theorem (Howard, 1960)

The policy iteration algorithm terminates after a finite number of steps with an optimal policy.

Example: Single-Sourcing

Code available at
https://people.orie.cornell.edu/jh2543/.

Part 5

Additional Topics

Some Active Research Areas on MDPs

1. Approximate Dynamic Programming (ADP)
    - aka. Reinforcement Learning
2. Robust MDPs
3. Learning when Demand Distributions are Unknown
4. Partially Observable MDPs

Approximate Dynamic Programming (ADP)

(called Reinforcement Learning (RL) in Computer Science (CS))

Motivated by problems that can be modeled as MDPs, but where one of the following issues exists:

- The MDP is too big (too many states, too many actions, ...).
- The transition probabilities are unknown.

One classic method from CS is **Q-Learning:** Approximate the "Q-factors"

$$Q(x,a) := r(x,a) + \beta \sum_{y \in \mathbb{X}} p(y|x,a) v_\beta(y), \quad x \in \mathbb{X},\ a \in A(x),$$

via stochastic approximation.

- Useful when the transition probabilities are unknown, but where one can easily interact with the system.
- Applications to SCM?

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Petersen, S. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529.

Wang, Y., & Jiang, D. R. (2018). Structured actor-critic for anaging and dispensing public health inventory. arXiv preprint arXiv:1806.02490.

Approximate Dynamic Programming

Value Function Approximation: Useful for large state spaces.

1. Find a good approximation of the value function: $v_\beta \approx \hat{v}_\beta$
2. Use a policy $\hat{\varphi}_*$ where

$$\hat{\varphi}_*(x) \in \arg\max_{a \in A(x)} \left[r(x,a) + \sum_{y \in \mathbb{X}} p(y|x,a)\hat{v}_\beta(y) \right], \quad x \in \mathbb{X}.$$

$\hat{v}_\beta$ is often taken to be the best approximation to v_β within some structured class of functions, e.g.,

- the output of a neural network
- linear combinations of basis functions:

$$v_\beta(x) \approx \sum_{k=1}^{s} \alpha_k \phi_k(x), \quad x \in \mathbb{X}.$$

Dai, J. and Shi, P., Inpatient overflow: An approximate dynamic programming approach (2018). Manufacturing and Service Operations Management, Forthcoming. Available at SSRN: https://ssrn.com/abstract=2924208.

Powell, W. B. (2016). Perspectives of approximate dynamic programming. Annals of Operations Research, 241(1-2), 319-356.

Other Topics Relevant to SCM

- Robust MDPs

  Lim H. S., Xu, H., Mannor S. (2016) Reinforcement learning in robust Markov decision processes. Mathematics of Operations Research 41(4):1325-1353.

  Sun, J. and Van Mieghem, J. A., Robust dual sourcing inventory management: Optimality of capped dual index policies and smoothing (April 18, 2018). Manufacturing & Service Operations Management, Forthcoming. Available at SSRN: https://ssrn.com/abstract=2991250.

- Learning when Demand Distributions are Unknown

  Cheung, W. C., Simchi-Levi, D., & Wang, H. (2017). Dynamic pricing and demand learning with limited price experimentation. Operations Research, 65(6), 1722-1731.

  Zhang, H. and Chao, X. and Shi, C., Closing the gap: A learning algorithm for the lost-sales inventory system with lead times (March 2018). Available at SSRN: https://ssrn.com/abstract=2922820.

- Partially Observable MDPs

  Kim, M. J. (2016) Robust control of partially observable failing systems. Operations Research 64(4):999-1014.

  Saghafian, S., Hopp, W. J., Iravani, S. M., Cheng, Y., & Diermeier, D. (2018). Workload management in telemedical physician triage and other knowledge-based service systems. Management Science *Articles in Advance*, https://doi.org/10.1287/mnsc.2017.2905

Summary

1. MDPs can model many decision-making problems relevant to SCM.
2. MDPs have attracted the attention of many OR researchers since the 1950s, up to the present day.
3. There are still many research opportunities!

Thank You!

jh2543@cornell.edu