Decision Making
in Robots and Autonomous Agents

Dynamic Programming Principle:
How should a robot go from "A to B"?

Subramanian Ramamoorthy
School of Informatics

26 January, 2018

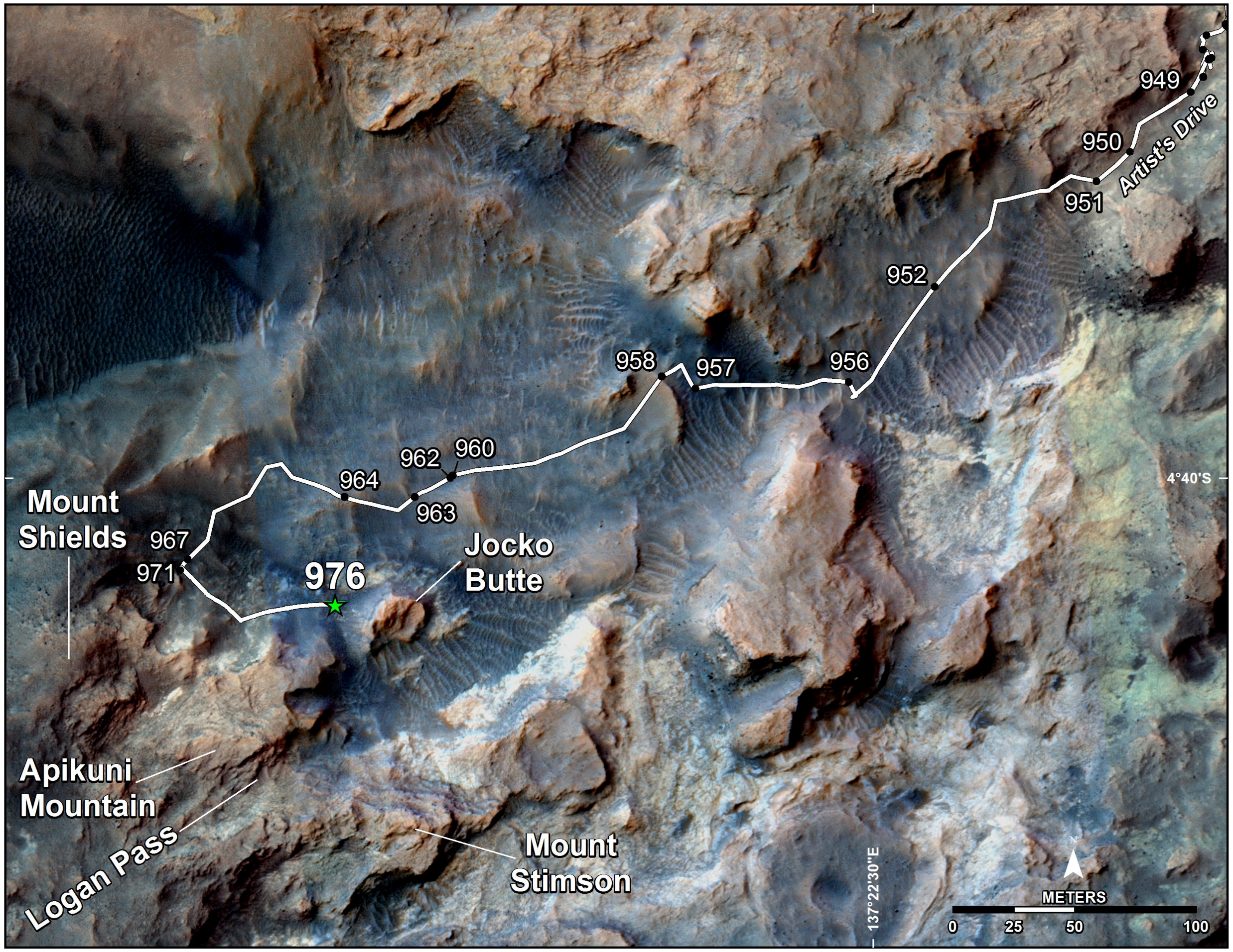
Objectives of this Lecture

- Introduce the dynamic programming principle, a way to solve sequential decision problems (such as path planning)

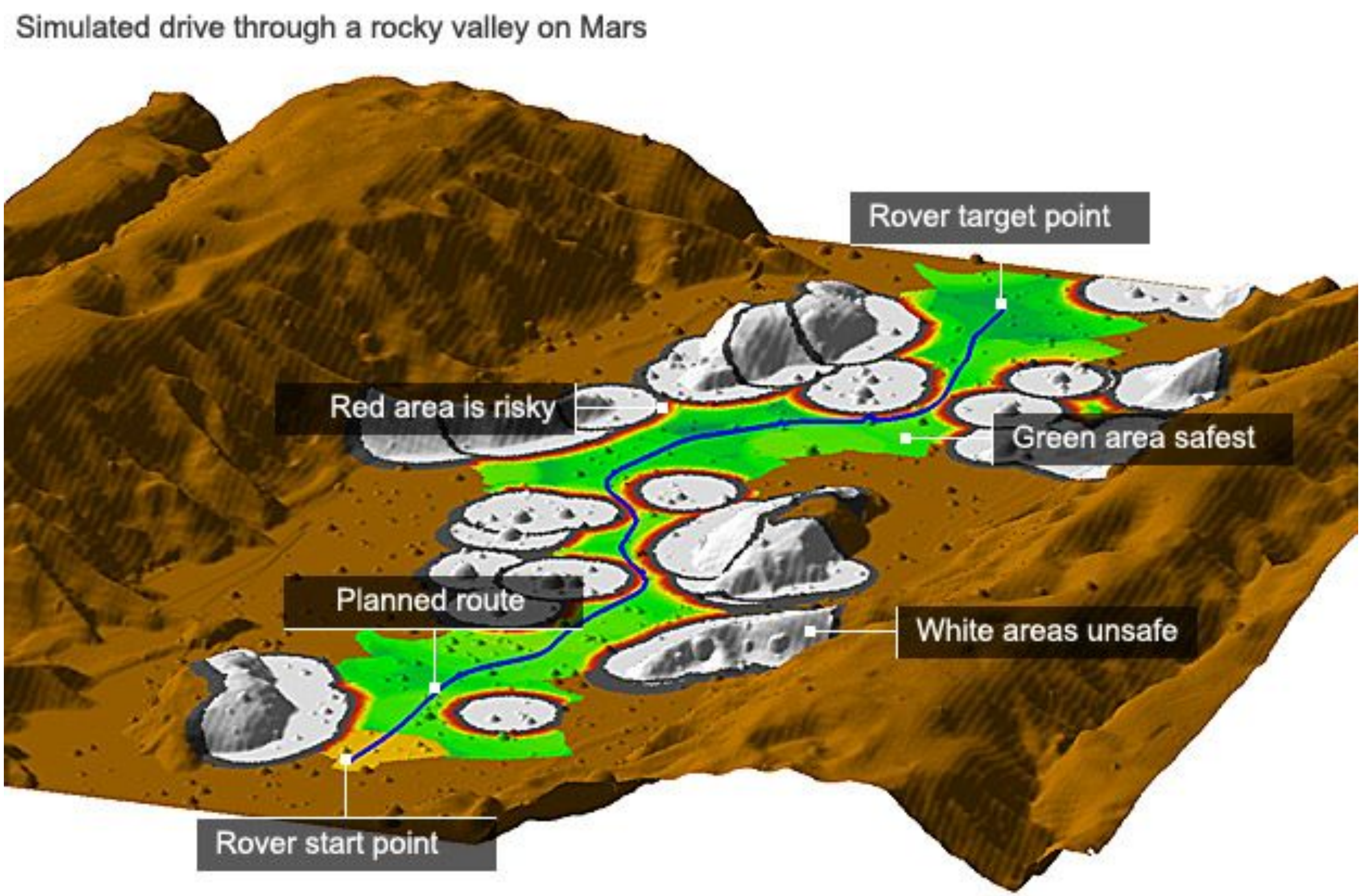
- Introduce the Markov Decision Process model, and discuss the nature of the policy arising in a similar sequential decision problem with probabilistic transitions
  - Includes recap of the notion of Markov Chains

Problem of Determining Paths

Getting from "A to B": Bird's Eye View



Getting from "A to B": Local View



How could we calculate the best path?

Dynamic Programming (DP) Principle

- Mathematical technique often useful for making a sequence of inter-related decisions
- Systematic procedure for determining the combination of decisions that maximize overall effectiveness
- There may not be a "standard form" of DP problems, instead it is an approach to problem solving and algorithm design

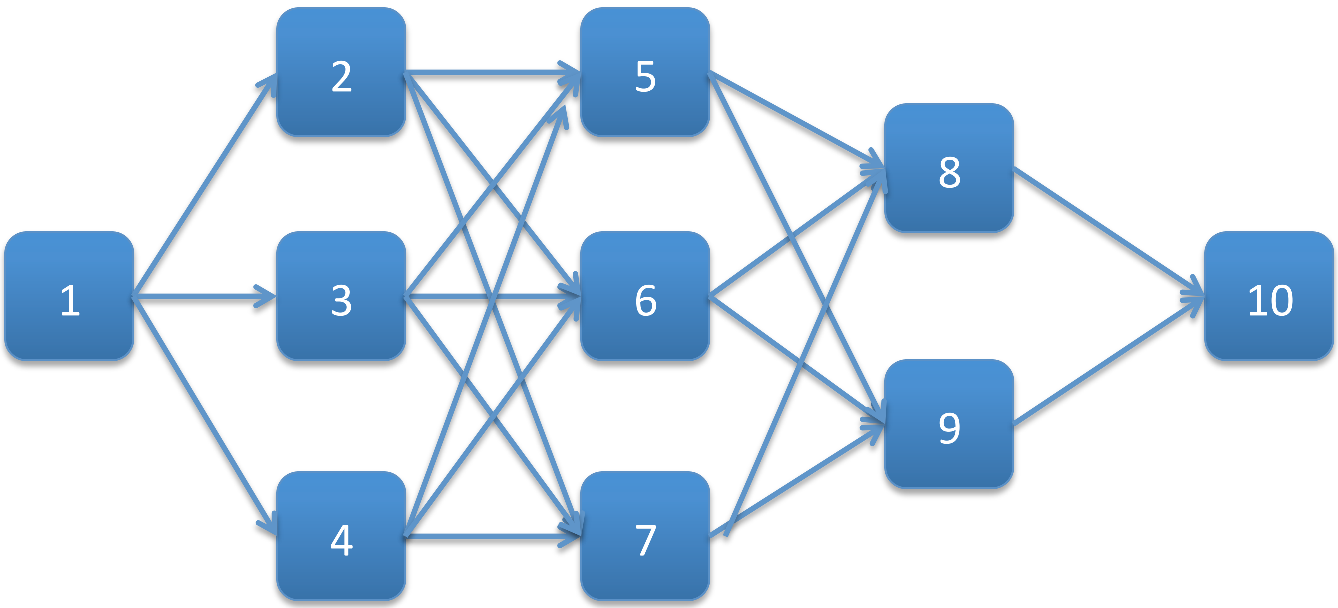
- We will try to understand this through a few example models, solving for the "optimal policy" (the notion of which will become clearer as we go along)

Stagecoach Problem

- Simple thought experiment due to H.M. Wagner at Stanford
- Consider a mythical American salesman from over a hundred years ago. He needs to travel west from the east coast, through unfriendly country with bandits.
- He has a well defined start point and destination, but the states he visits en route are up to his own choice

- Let us visualize this, using numbered blocks for states

Stagecoach Problem: Possible Routes



Each box is a state (generically indexed by an integer, i)
Transitions, i.e., edges, can be annotated with a "cost"

Stagecoach Problem: Setup

- The salesman needs to go through four stages to travel from his point of departure in state 1 to destination in state 10
- This salesman is concerned about his safety – does not want to be attacked by bandits
- One approach he could take (as envisioned by Wagner):
  - Life insurance policies are offered to travellers
  - Cost of each policy is based on evaluation of safety of path
  - Safest path = cheapest life insurance policy

Stagecoach Problem: Costs

The cost of the standard policy on the stagecoach run from state i to state j denoted by c_{ij} is

	2	3	4
1	2	4	3

	5	6	7
2	7	4	6
3	3	2	4
4	4	1	5

	8	9
5	1	4
6	6	3
7	3	3

	10
8	3
9	4

Which route minimizes the total cost of the policy?

Myopic Approach

- Making the decision which is best for each successive stage need not yield the overall optimal decision
- WHY?

- Selecting the cheapest run offered by each successive stage would give the route 1 -> 2 -> 6 -> 9 -> 10.
- What is the total cost?

- **Observation**: Sacrificing a little on one stage may permit greater savings thereafter.
  - e.g., a cheaper alternative to 1 -> 2 -> 6 is 1 -> 4 -> 6

Is Trial and Error Useful?

- What does it mean to solve the problem (finding the cheapest cost path) by trial and error?
  - What are the trials over? What is the error?

- How many possible routes do we have in this problem?

  Ans: 18

- Is exhaustive enumeration always an option? How does the number of routes scale?

Dynamic Programming Principle

- Start with a small portion of the problem and find optimal solution for this smaller problem
- Gradually enlarge the problem – finding the current optimal solution from the previous one

  … until original problem is solved in its entirety

- This general philosophy is the essence of the DP principle
  - The details are implemented in many different ways in different specialised scenarios

Solving the Stagecoach Problem

- At stage n, consider the decision variable x_n ($n = 1,2,3,4$).
- The selected route is: $1 \rightarrow x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_4$

Which state is implied by x_4?

- Total cost of the overall best *policy* for the *remaining* stages, given that the salesman is in state s and selects x_n as the immediate destination: $f_n(s, x_n)$

$$x_n^* = \arg\min f_n(s, x_n)$$

$$f_n^*(s) = \text{minimum value of } f_n(s, x_n)$$

$$f_n^*(s) = f_n(s, x_n^*)$$

Solving the Stagecoach Problem

- The objective is to determine $f_1^*(1)$
  and the corresponding optimal policy achieving this
- DP achieves this by successively finding $f_4^*(s), f_3^*(s), f_2^*(s)$
  which will lead us to the desired $f_1^*(1)$

- When the salesman has only one more stage to go, his route is entirely determined by his final destination. Therefore,

s	$f_4^*(s)$	x_4^*
8	3	10
9	4	10

Solving the Stagecoach Problem

- What about when the salesman has two more stages to go?

- Assume salesman is at stage 5 – he must next go either to stage 8 or 9 at cost of 1 or 4 respectively
  - If he chooses stage 8, minimum additional cost after reaching there is 3 (table in earlier slide)
  - So, cost for that decision is 1 + 3 = 4
  - Total cost if he chooses stage 9 is 4 + 4 = 8

- Therefore, he should choose state 8

The Two-stage Problem

$$f_3(s, x_3) = c_{sx_3} + f_4^*(x_3)$$

$s \backslash x_3$	8	9	$f_3^*(s)$	x_3^*
5	4	8	4	8
6	9	7	7	9
7	6	7	6	8

Likewise, Three-stage Problem

$$f_2(s, x_2) = c_{sx_2} + f_3^*(x_2)$$

$s \backslash x_2$	5	6	7	$f_2^*(s)$	x_2^*
2	11	11	12	11	5 or 6
3	7	9	10	7	5
4	8	8	11	8	5 or 6

Finally, the Four-stage Problem

$$f_1(s, x_1) = c_{sx_1} + f_2^*(x_1)$$

$s \backslash x_1$	2	3	4	$f_1^*(s)$	x_1^*
1	13	11	11	11	3 or 4

<u>Optimal Solution</u>:
Salesman should first go to either 3 or 4
Say, he chooses 3, the three-stage problem result is 5
Which leads to the two-stage problem result of 8
And, of course, finally 10

Characteristics of DP Problems

The stagecoach problem might have sounded strange, but it is the literal instantiation of key DP terms

DP problems all share certain features:

1. The problem can be divided into **stages**, with a **policy decision** required at each stage
2. Each stage has several **states** associated with it
3. The effect of the policy decision at each stage is to transform the current state into a state associated with the next stage (could be according to a probability distribution, as we'll see next).

Characteristics of DP Problems, contd.

5. Given the current state, an optimal policy for the remaining stages is **independent** of the policy adopted in previous stages
6. The solution procedure begins by finding the optimal policy for each state of the last stage.
7. Recursive relationship identifies optimal policy for each state at stage n, given optimal policy for each state at stage $n+1$:

$$f_n^*(s) = \min_{x_n}\{c_{sx_n} + f_{n+1}^*(x_n)\}$$

8. Using this recursive relationship, the solution procedure moves backward stage by stage – until finding optimal policy from initial stage

Let us now consider a problem where the transitions may not be deterministic:

(A little bit about) Markov Chains and Decisions

Stochastic Processes

- A *stochastic process* is an indexed collection of random variables $\{X_t\}$
  - e.g., collection of weekly demands for a product
- One type: At a particular time t, labelled by integers, system is found in exactly one of a finite number of mutually exclusive and exhaustive categories or **states**, labelled by integers too
- Process could be *embedded* in that time points correspond to occurrence of specific events (or time may be equi-spaced)
- Random variables may depend on others, e.g.,

$$X_{t+1} = \{ \begin{array}{c} \max\{(3 - D_{t+1}), 0\}, if X_t < 0 \\ \max\{(X_t - D_{t+1}), 0\}, if X_t \geq 0 \end{array}$$

Markov Chains

- The stochastic process is said to have a **Markovian** property if

$$P\{X_{t+1} = j | X_0 = k_0, X_1 = k_1, ..., X_{t-1} = k_{t-1}, X_t = i\} = P\{X_{t+1} = j | X_t = i\}$$

for $t = 0, 1, ...$ and every sequence $i, j, k_0, ..., k_{t-1}$.

- Markovian property means that the conditional probability of a future event given any past events and current state, is *independent* of past states and depends only on present
- The conditional probabilities are **transition probabilities**,

$$P\{X_{t+1} = j | X_t = i\}$$

- These are stationary if time invariant, called p_{ij},

$$P\{X_{t+1} = j | X_t = i\} = P\{X_1 = j | X_0 = i\}, \forall t = 0, 1, ...$$

Markov Chains

- Looking forward in time, n-step **transition probabilities**, $p_{ij}^{(n)}$

$$P\{X_{t+n} = j | X_t = i\} = P\{X_n = j | X_0 = i\}, \forall t = 0, 1, ...$$

- One can write a transition matrix,

$$\mathbf{P}^{(n)} = \begin{bmatrix} p_{00}^{(n)} & \cdots & p_{0M}^{(n)} \\ \vdots & & \\ p_{M0}^{(n)} & \cdots & p_{MM}^{(n)} \end{bmatrix}$$

- A stochastic process is a finite-state Markov chain if it has,
  - Finite number of states
  - Markovian property
  - Stationary transition probabilities
  - A set of initial probabilities $P\{X_0 = i\}$ for all i

Markov Chains

- n-step transition probabilities can be obtained from 1-step transition probabilities recursively (Chapman-Kolmogorov)

$$p_{ij}^{(n)} = \sum_{k=0}^{M} p_{ik}^{(v)} p_{kj}^{(n-v)}, \forall i, j, n; 0 \leq v \leq n$$

- We can get this via the matrix too

$$P^{(n)} = P.P \ldots P = P^n = PP^{n-1} = P^{n-1}P$$

- **First Passage Time**: number of transitions to go from i to j for the first time
  - If $i = j$, this is the **recurrence time**
  - In general, this itself is a random variable

Markov Chains

- n-step recursive relationship for first passage time

$$
\begin{aligned}
f_{ij}^{(1)} &= p_{ij}^{(1)} = p_{ij}, \\
f_{ij}^{(2)} &= p_{ij}^{(2)} - f_{ij}^{(1)} p_{jj}, \\
&\vdots \\
f_{ij}^{(n)} &= p_{ij}^{(n)} - f_{ij}^{(1)} p_{jj}^{(n-1)} - f_{ij}^{(2)} p_{jj}^{(n-2)} \ldots - f_{ij}^{(n-1)} p_{jj}
\end{aligned}
$$

- For fixed i and j, these $f_{ij}^{(n)}$ are nonnegative numbers so that

$$\sum_{n=1}^{\infty} f_{ij}^{(n)} \leq 1$$

What does <1 signify?

- If, $\sum_{n=1}^{\infty} f_{ii}^{(n)} = 1$, state is **recurrent**; If n=1 then it is **absorbing**

Markov Chains: Long-Run Properties

- Consider this transition matrix of an inventory process:

$$P^{(1)} = P = \begin{bmatrix} 0.08 & 0.184 & 0.368 & 0.368 \\ 0.632 & 0.368 & 0 & 0 \\ 0.264 & 0.368 & 0.368 & 0 \\ 0.08 & 0.184 & 0.368 & 0.368 \end{bmatrix}$$

- This captures the evolution of inventory levels in a store
  - What do the 0 values mean?
  - Other properties of this matrix?

Markov Chains: Long-Run Properties

The corresponding 8-step transition matrix becomes:

$$P^{(8)} = P^8 = \begin{bmatrix} 0.286 & 0.285 & 0.264 & 0.166 \\ 0.286 & 0.285 & 0.264 & 0.166 \\ 0.286 & 0.285 & 0.264 & 0.166 \\ 0.286 & 0.285 & 0.264 & 0.166 \end{bmatrix}$$

Interesting property: probability of being in state j after 8 weeks appears independent of *initial* level of inventory.

- For an irreducible ergodic Markov chain, one has limiting probability

$$\lim_{n\to\infty} p_{ij}^{(n)} = \pi_j$$

Reciprocal gives you recurrence time

$$\pi_j = \sum_{i=0}^{M} \pi_i p_{ij}, \forall j = 0, \dots, M$$

Markov **Decision** Model

- Consider the following application: machine maintenance
- A factory has a machine that deteriorates rapidly in quality and output and is inspected periodically, e.g., daily
- Inspection declares the machine to be in four possible states:
  - 0: Good as new
  - 1: Operable, minor deterioration
  - 2: Operable, major deterioration
  - 3: Inoperable
- Let X_t denote this observed state
  - evolves according to some "law of motion", it is a stochastic *process*
  - Furthermore, assume it is a finite state Markov chain

Markov **Decision** Model

- Transition matrix is based on the following:

States	0	1	2	3
0	0	7/8	1/16	1/16
1	0	3/4	1/8	1/8
2	0	0	1/2	1/2
3	0	0	0	1

- Once the machine goes inoperable, it stays there until repairs
  – If no repairs, eventually, it reaches this state which is absorbing!

- Repair is an **action** – a very simple maintenance **policy**.
  – e.g., machine from from state 3 to state 0

Markov **Decision** Model

- There are costs as system evolves:
  - State 0: cost 0
  - State 1: cost 1000
  - State 2: cost 3000
- Replacement cost, taking state 3 to 0, is 4000 (and lost production of 2000), so cost = 6000
- The modified transition probabilities are:

States	0	1	2	3
0	0	7/8	1/16	1/16
1	0	3/4	1/8	1/8
2	0	0	1/2	1/2
3	1	0	0	0

Markov **Decision** Model

- Simple question (a behavioural property):
  What is the average cost of this maintenance policy?

- Compute the steady state probabilities:

$$\pi_0 = \frac{2}{13}; \pi_1 = \frac{7}{13}; \pi_2 = \frac{2}{13}; \pi_3 = \frac{2}{13}$$

How?

- (Long run) expected average cost per day,

$$0\pi_0 + 1000\pi_1 + 3000\pi_2 + 6000\pi_3 = \frac{25000}{13} = 1923.08$$

Markov **Decision** Model

- Consider a slightly more elaborate policy:
  - When it is inoperable or needing major repairs, replace
- Transition matrix now changes a little bit
- Permit one more possible action: overhaul
  - Go back to minor repairs state (1) for the next time step
  - Not possible if truly inoperable, but can go from major to minor
- Key point about the system behaviour. It evolves according to
  - "Laws of motion"
  - Sequence of decisions made (actions from {1: none,2:overhaul,3: replace})
- Stochastic process is now defined in terms of $\{X_t\}$ and $\{\Delta_t\}$
  - Policy, R, is a rule for making decisions
    - Could use all history, although popular choice is (current) state-based

Markov **Decision** Model

- There is a space of potential policies, e.g.,

Policies	$d_0(R)$	$d_1(R)$	$d_2(R)$	$d_3(R)$
R_a	1	1	1	3
R_b	1	1	2	3
R_c	1	1	3	3
R_d	1	3	3	3

- Each policy defines a transition matrix, e.g., for R_b

States	0	1	2	3
0	0	7/8	1/16	1/16
1	0	3/4	1/8	1/8
2	0	1	0	0
3	1	0	0	0

Which policy is best?
Need costs….

Markov **Decision** Model

- C_{ik} = expected cost incurred during next transition if system is in state i and decision k is made

State	Dec. 1	2	3
0	0	4	6
1	1	4	6
2	3	4	6
3	∞	∞	6

- The long run average expected cost for each policy may be computed using

$$E(C) = \sum_{i=0}^{M} C_{ik}\pi_i$$

R_b is best

So, What is a Policy?

- A "program"
- Map from states (or situations in the decision problem) to actions that could be taken
  - e.g., if in 'level 2' state, call contractor for overhaul
  - If less than 3 DVDs of a film, place an order for 2 more

- A probability distribution $\pi(s,a)$
  - A joint probability distribution over states and actions
  - If in a state s_1, then with probability defined by π, take action a_1

Some Acknowledgements

- Slide 3: https://www.nasa.gov/sites/default/files/thumbnails/image/pia19808-main_tight_crop-monday.jpg
- Slide 4: https://www.nasa.gov/sites/default/files/thumbnails/image/pia19399_msl_mastcammosaiclocations.jpg
- Slide 5: https://ichef.bbci.co.uk/news/624/media/images/55165000/jpg/_55165401_exomarssimulation.jpg
- Core examples are from F.S. Hillier, G.J. Lieberman, Operations Research, 1994. (esp. Ch 6 and 12)