SEMI-MARKOV DECISION PROCESSES

SMDP is a generalization of the Markov Decision Process (MDP) where the times between transitions are allowed to be random variables whose distribution may depend upon

- the current state
- the action taken
- (possibly) the next state

Inventory Replenishment: Rather than review the inventory and make a replenishment decision at the end of each day, an automated system might make the decision *after each demand occurs*, an event which can happen at any time during the day.

Taxicab Problem: In the taxi-cab problem used earlier to illustrate MDP, *average reward per trip* was optimized (transitions correspond to passengers).
The *duration of the trips* will vary, depending upon source & destination, and time waiting for the next passenger can depend upon the action (cruising the street, waiting at a taxi stand, waiting for a radio call). More meaningful, therefore, would be optimizing the *average reward per unit time*.

Notation:

τ_i^a = time that the system spends in state i before the next transition, if action a is selected.

$v_i^a \triangleq E\left[\tau_i^a\right]$ = expected duration of the time spent in state i if action a is selected.

p_{ij}^a = probability that the next state is j, given that the current state is i and action a has been selected.

c_i^a = expected total cost if action a is selected in state i.

(Nonlinear) Programming Model for SMDP:

(Average Cost Criterion)

$$\text{Minimize } \frac{\sum_i \sum_a c_i^a x_i^a}{\sum_i \sum_a v_i^a x_i^a}$$

$$\text{subject to } \sum_i \sum_a x_i^a = 1$$

$$\sum_a x_j^a = \sum_i \sum_a p_{ij}^a x_i^a \quad \text{for all states } j$$

$$x_i^a \geq 0 \quad \text{for all states } i \text{ and } a \in A_i$$

As in the case of MDP, we make a

Unichain Assumption:

Every single-stage decision rule R results in a transition probability matrix P^R for which the corresponding *discrete-time Markov chain* has a **single** recurrent set of states and a (possibly empty) set of transient states.

Lemma Let **M** be a matrix and **b** & **d** vectors with the properties

(i) $\left.\begin{matrix} Mx = 0 \\ x \geq 0 \end{matrix}\right\} \Rightarrow x = 0$

(ii) $\left.\begin{matrix} x \geq 0 \\ Mx = b \end{matrix}\right\} \Rightarrow dx > 0$

Make the ***transformation***

$$u = \frac{x}{dx} \quad \text{and} \quad y = \frac{1}{dx}$$

Then there is a ***one-to-one correspondence*** between the solutions of the two systems

$$\begin{cases} Mx = b \\ x \geq 0 \end{cases} \quad \leftrightarrow \quad \begin{cases} Mu = by \\ du = 1 \\ u \geq 0 \end{cases}$$

As a result of this lemma, the ***nonlinear*** (fractional) programming problem

$$\begin{aligned} &\text{Minimize } \frac{cx}{dx} \\ &\text{subject to } Mx = b, \\ &\qquad x \geq 0 \end{aligned}$$

is equivalent to the ***linear*** programming problem

$$\begin{aligned} &\text{Minimize } cu \\ &\text{subject to } Mu = b \\ &\qquad du = 1 \\ &\qquad u \geq 0 \end{aligned}$$

LP model for SMDP: *(Average Cost Criterion)*

$$\text{Minimize } \sum_i \sum_a c_i^a u_i^a$$

$$\text{subject to } \sum_j u_j^a = \sum_i \sum_a p_{ij}^a u_i^a \quad \text{for all states } j$$

$$\sum_i \sum_a v_i^a u_i^a = 1$$

$$u_i^a \geq 0 \quad \text{for all states } i \text{ and actions } a \in A_i$$

Notes:

- *If* $v_i^a \equiv 1$, *then of course this LP model is identical to that of the MDP given earlier, with* $x_i^a = u_i^a$.
- *As in the MDP case, the "steady state" equations above include one redundant constraint which can be eliminated.*

We see, then, that the SMDP may be optimized by a rather small modification to the LP model, replacing x by u and

$$\sum_i \sum_a x_i^a = 1$$

by

$$\sum_i \sum_a v_i^a u_i^a = 1.$$

The objective of optimizing the ***discounted*** *total cost may also be treated in SMDP, but the derivation is more complex and is not treated here.*