EE 223: Stochastic Systems: Estimation and Control	Spring 2007
Lecture 20 — April 3	
Lecturer: Venkat Anantharam	*Scribe: Phoebus Chen*

20.1 Example of the use of Discounted Dynamic Programming in a Continuous Time Problem

20.1.1 Poisson Process

Recall the definition of a Poisson Process with rate γ, depicted in Figure 20.1. This is a stochastic process counting the number of arrivals up to time t starting from time 0, $(N_t, t \geq 0)$, where the interarrival times are exponentially distributed with mean $1/\gamma$. we can refer to this as a *Poisson (γ) process*

The number of points in an interval $(a, b]$ is denoted $N(a, b]$.

Proposition 20.1. *If $(a, b]$ and $(c, d]$ are disjoint then $N(a, b]$ and $N(c, d]$ are independent.*

Proposition 20.1 allows one to define a Poisson process on $\mathbb{R}$ by saying $N(a, b]$ is given by a Poisson distribution with parameter $\gamma(b - a)$. The number of points in a disjoint intervals are independent.

For more details, refer to a first or second course on probability such as EE226A.

20.1.2 Queuing Example

Consider a buffer or a queue that can hold packets somewhere in a network. The *arrival process* of packets is modeled as a Poisson process of rate λ. Each packet requires an independent exponentially distributed time to serve, the mean service time being $\frac{1}{\mu}$.

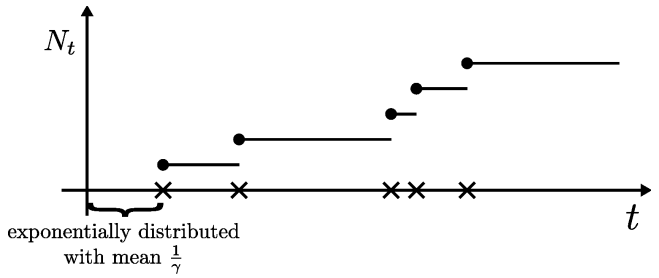

Figure 20.1. Illustration of a Poisson process.

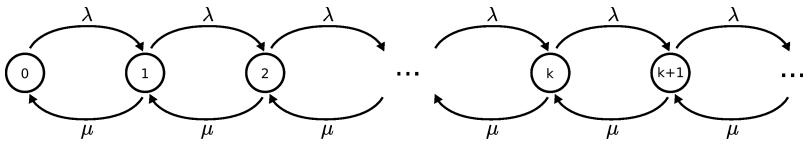

Figure 20.2. Birth Death process.

If all packets were admitted, the queue size would be modeled by a continuous time Markov chain (birth and death process) (See Figure 20.2).

Consider an admission control problem: Suppose the cost of dropping a packet is C and the cost of maintaining a queue of size k is Ak per unit time. There is a tradeoff between admitting and dropping packets, and we wish to find the optimal policy for dropping packets.

While one may be able to guess the answer using intuition, we wish to know arrive at the answer using theory. It turns out that we do not need to resort to using continuous time dynamic programming or stochastic calculus for this; our discrete time theory is good enough. One can often use a technique called *uniformization* to convert a continuous time stochastic control problem whose transitions are driven by Poisson processes into a discrete time control problem.

(1) First note that the services can also be described via a Poisson process (of rate μ) independent of the arrival process. We call this process a *virtual* service process. If there is virtual service when there is no packet in the queue, the virtual service is simply wasted.

(2) Note that Bernoulli thinning of a Poisson (γ) process with a probability of success $0 < \beta < 1$ gives a Poisson ($\beta\gamma$) process.(Bernoulli thinning simply means for each point in the original process we toss a coin with probability of success β and keep the point when the coin toss comes up heads but discard the point when it comes up tails). Further, if $\beta_1 + \beta_2 < 1$ and we toss a 3-sided coin with whose outcomes $(0, 1, 2)$ have probabilities $(1 - \beta_1 - \beta_2, \beta_1, \beta_2)$, then the three resulting Bernoulli thinnings are independent Poisson processes of rates $(1 - \beta_1 - \beta_2)\gamma$, $\beta_1\gamma$, and $\beta_2\gamma$ respectively.

So in our example, we first start with a simple Poisson process of rate $\gamma > \lambda + \mu$. Then, we thin the Poisson process with probabilities

$$\Big(\underbrace{\frac{\gamma - \lambda - \mu}{\gamma}}_{\text{marking time (nothing happens)}}, \underbrace{\frac{\lambda}{\gamma}}_{\text{arrival}}, \underbrace{\frac{\mu}{\gamma}}_{\text{virtual service}} \Big) .$$

Now we can think in terms of a discrete time Markov chain (which is "really" running on the points of the underlying rate γ Poisson process). The transitions for the discrete time

Markov chain is depicted as

$$k \longrightarrow \begin{cases} k+1 & \text{with probability } \frac{\lambda}{\gamma} \\ k & \begin{cases} \text{with probability } \frac{1-\lambda-\mu}{\gamma} \text{ if } k>0 \\ \text{and with probability } \frac{1-\lambda}{\gamma} \text{ if } k=0 \end{cases} \\ k-1 & \text{with probability } \frac{\mu}{\gamma} \text{ if } k>0 \end{cases} \tag{20.1}$$

If we now consider the control problem, we can think of either not accepting the transition from k to $k+1$ (paying a cost $E[\tilde{C}\tilde{\alpha}^T]$, where $T \sim exp(\gamma)$) as a control choice in each state k. We also pay a cost $\tilde{A}k \int_0^\infty \tilde{\alpha}^t \gamma e^{-\gamma t} dt$ in state k at each step. In other words, our objective is

$$\min_{\Psi} E[\int_0^\infty \tilde{\alpha}^t \tilde{A} X_t^\Psi dt + \sum_{n\geq 1} \tilde{C}\tilde{\alpha}^{T_n^\Psi}] \tag{20.2}$$

where T_n = time of the n-th discarded packet. Here $\tilde{C}$, $\tilde{A}$, and $\tilde{\alpha}$ can be chosen in terms of C, A, and α to make the discrete time control problem identical to the original continuous time control problem (you should think this through for yourself).

One can prove that the cost to go of the discrete time problem is convex and conclude that the optimal strategy is of the threshold type: There is some threshold $K \geq 0$ such that if the queue size is K or larger when a new packet arrives, we should drop the packet, while if the queue size is $K-1$ or less, we should accept the packet.

20.2 Controlled Markov Chain, Average Cost Dynamic Programming

Let $\mathbb{P}(u) = [p_{ij}(u)]$ be the transition probability matrix and $g(i,u)$ be the one step costs.

Objective

$$\min_{\Psi} \limsup_{N} \frac{1}{N} E[\sum_{k=0}^{N-1} g(X_k^\Psi, U_k^\Psi)] \tag{20.3}$$

Main Result If there exists a number λ and a vector $[h(1) \quad \dots \quad h(d)]^T$ (where $d = |\mathcal{X}|$) such that $\lambda + h(i) = \min_u[g(i,u) + \sum_j p_{ij}(u)h(j)]$ for all i, then λ is the optimum average cost and $\mu^* : \mathcal{X} \mapsto \mathcal{U}$ with $\mu^*(i)$ minimizing state i defines an optimal stationary Markov strategy.

Proof:

$$g(i,u) + \sum_j p_{ij}(u)h(j) \geq \lambda + h(i) \quad \text{for all } (i,u) \tag{20.4}$$

Hence, substituting X_k^Ψ for i and U_k^Ψ for u we get

$$\begin{aligned} g(X_k^\Psi, U_k^\Psi) &\geq \lambda + h(X_k^\Psi) - \sum_j p_{X_k^\Psi j}(U_k^\Psi) h(j) \\ E[g(X_k^\Psi, U_k^\Psi)] &\geq \lambda + E[h(X_k^\Psi)] - \underbrace{E[\sum_j p_{X_k^\Psi j}(U_k^\Psi) h(j)]}_{E[h(X_{k+1}^\Psi)]} \end{aligned}$$

This gives rise to a telescoping sum, so

$$\liminf_N \frac{1}{N} \sum_{k=0}^{N-1} g(X_k^\Psi, U_k^\Psi) \geq \lambda \ . \tag{20.5}$$

Conversely, if use any stationary Markov strategy defined by some $\mu^* : \mathcal{X} \mapsto \mathcal{U}$ where $\mu^*(i)$ is minimizing in state i for all $i \in \mathcal{X}$ we have equality in these equations, and so, for Ψ the corresponding stationary Markov strategy we get

$$\lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} g(X_k^\Psi, U_k^\Psi) = \lambda \ . \tag{20.6}$$

□

We want to lead up to a proof that under the hypothesis that $P_\mu = [p_{ij}(\mu(i))]$ is irreducible for all μ, there exists λ and $[h(1) \quad \dots \quad h(d)]^T$ satisfying the average cost Bellman equation. This involves first showing that the discounted DP problem for discount α close to 1 approaches the average cost DP problem. This approach is known as the *vanishing discount limit* approach.

Definition A stationary strategy is called *Blackwell Optimal* if there is some $\bar{\alpha} < 1$ such that μ is optimal in the α-discounted problem for all $\alpha \in (\bar{\alpha}, 1)$.

Theorem 20.2. *A Blackwell Optimal strategy exists.*

Proof:

The overall cost-to-go for the α-discounted strategy μ, $J_{\alpha,\mu} = (I - \alpha P_\mu)^{-1}$ is a rational function of α. This proves the result, because a polynomial in α can have only finitely many values of α as roots. More details can be found in chapter 4 of Vol.2 of the textbook.