Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias Físico Matemáticas
Postgrado en Ciencias Matemáticas

Aleatorización en Programación Dinámica para el análisis del problema de la Dimensionalidad aplicada a Procesos de Decisión de Markov

Tesis

Que para obtener el título de
Maestra en Ciencias (Matemáticas)

Presenta
María Selene Georgina Chávez Rodríguez

Director de Tesis
Dr. Hugo Adán Cruz Suárez

Puebla, Pue. Septiembre 2012

Introducción

El trabajo realizado en la presente tesis está relacionado con la teoría de Procesos de Decisión de Markov (PDMs) a tiempo discreto y con horizonte finito (véase [4], [7] y [10]).

Los PDMs son usados para modelar sistemas que son observados de forma discreta en el tiempo por un controlador en un periodo finito o infinito de tiempo, en el cual el sistema puede presentar una variación en su movimiento. Los PDMs son aplicados en áreas como economía, biología, ingeniería, etc (véase [10], [12] y [15]). Los PDMs se encuentran caracterizados por un modelo conocido como Modelo de Control de Markov (MCM), cuyas componentes permiten describir su desarrollo en el transcurso del tiempo. La evolución de un PDM está dado de acuerdo al siguiente procedimiento. Sea $x_t = x \in X$ un estado al tiempo $t, t = 0, 1, ...$ y $a_t = a \in A(x)$ la acción (control) elegida en ese tiempo, entonces el sistema transita del estado x al estado $x_{t+1} = y \in X$ con probabilidad $Q(y|x, a)$, pagándose un costo $c(x, a)$ (u obteniendo una recompensa $r(x, a)$). Una vez que la transición al siguiente estado ha ocurrido, una nueva acción es elegida y, el proceso es repetido.

A la sucesión de controles que el proceso genera se le conocerá como política. Para evaluar la calidad de cada política se contará con un criterio de rendimiento que medirá la eficiencia de las políticas en función de los costos o recompensas que generan. En la presente tesis se trabajará con el criterio de costo total acumulado. Así, el Problema de Control Óptimo (PCO) consiste en encontrar una política que optimice el criterio de rendimiento. A la política que optimiza el criterio de rendimiento se le llama política óptima y, al criterio de rendimiento evaluado en tal política óptima se le conoce como la función de valores óptimos.

Una manera de resolver el PCO es mediante la técnica conocida como Programación Dinámica (PD) iniciada a mediados de los años 50's por Richard E. Bellman (véase [2]). El principio de Programación Dinámica permite resolver problemas en los que es necesario tomar decisiones en etapas sucesivas que condicionan la evolución futura del sistema, afectando a las situaciones en las que el sistema se encontrará en el futuro (estados), y a las decisiones (acciones) que se plantearán; todo esto se lleva a cabo mediante la Ecuación de Optimalidad de Bellman (EO), la cual permite establecer una forma recursiva que permite resolver el problema a tratar.

La técnica de PD permite determinar tanto a la política óptima como a la función de valor, sin embargo, una desventaja de esta técnica surge cuando el

tamaño del espacio de estados (o acciones) es grande ocasionando que el costo computacional al implementar algoritmos aumente. Esto es conocido en la literatura como *la Maldición de la Dimensionalidad* (*The Curse of Dimensionality*, véase [2], [6] y [9]). La Maldición de la Dimensionalidad plantea que el número de operaciones necesarias para resolver un problema crece exponencialmente con la dimensión del espacio de estados (o controles); la referencia más temprana a dicho término aparece en el texto de Bellman, [2]; y también en [3]. Algunos autores que han trabajado este problema son: [9], [5] y [11].

Así, una importante cuestión es cómo evitar el problema de la Dimensionalidad, mediante la elección de métodos apropiados. De donde, más allá de la determinación de la política óptima o de la función de valor, el problema de la dimensionalidad se convierte en el principal objetivo de este trabajo de tesis. Para ello se propone un método que permite reducir el costo computacional de los algoritmos propuestos.

Algunos métodos que se han propuesto para reducir el problema de la dimensión son: el *Método de la Desigualdad de Bellman*, el cual consiste en aproximar a la función de valor mediante una familia de funciones base, dichas funciones están sujetas a ciertas condiciones, las cuales garantizarán que la aproximación obtenida resulta ser óptima para la verdadera función de valor (véase [13]). Una desventaja de la ecuación de programación dinámica es que debe ser iterada para todos los estados de manera simultánea, así una alternativa es iterar un estado a la vez, mientras se incorpora el cálculo del resultado anterior. Esto es conocido como, el *Método de Gauss-Seidel* (véase [9] y [10]); y, el método de *Simulación de cota superior*, en el cual, para cada estado, en lugar de considerar todo el conjunto de acciones se busca simular las acciones más apropiadas a cada estado (véase [5]). Sin embargo, los métodos anteriores son comúnmente usados cuando el espacio de estados es continuo, así el método que se estudió en esta tesis es conocido como aleatorización, el cual consiste en aproximar a la función de valor mediante la simulación de muestras aleatorias, en este caso, del espacio de estados y evaluando dichas muestras en una Ecuación de Programación Dinámica Aproximada (EPDA), la cual será descrita más adelante (Ecuación 3.17), para determinar una función de valor aproximada. Finalmente se repetirá este procedimiento una cantidad determinada de veces y se procederá a promediar todas las funciones de valor obtenidas, y el resultado de dicho promedio será la aproximación a la función de valor que se busca obtener.

Este trabajo se encuentra motivado en el artículo de Rust (véase [11]), en el cual se estudia un MCM con espacio de estados continuo ($X = [0, 1]$). En este trabajo se presenta una versión adaptada a un MCM con espacio de estados finito. Así, la aportación que se hace con esta tesis es el estudio del método de aleatorización en el caso discreto, así como la prueba de que la aproximación propuesta resulta ser cercana a la función de valor, además de que se desarrollaron algoritmos computacionales en Matlab donde se implementó esta técnica. Note que para todos los métodos propuestos anteriormente se reemplaza la función de valor verdadera por funciones de valor aproximadas, esto es conocido como Programación Dinámica Aproximada (PDA), (véase [9], [5]).

Un método numérico usado en las aproximaciones que se proponen es el

método Monte Carlo, el cual inicialmente se utilizó para evaluar integrales múltiples definidas. El método de Monte Carlo es una técnica numérica para calcular probabilidades y otras cantidades relacionadas, utilizando secuencias de números aleatorios. Para el caso de una sola variable el procedimiento es el siguiente: generar una serie de números aleatorios, $x_1, x_2, \ldots x_n$, uniformemente distribuidos en $[0, 1]$. Usar esta sucesión para producir otra sucesión, $u_1, u_2, \ldots, u_n$, distribuida de acuerdo a la función de distribución de probabilidad en la que se está interesado. Después, se usa la sucesión de valores $\{u_i\}$ para estimar algunas características de la función de distribución de interés, $f(u)$. Los valores de u pueden tratarse para que a partir de ellos se puedan estimar probabilidades o esperanzas.

La tesis está organizada de la siguiente manera: en el Capítulo 1 se presenta la teoría de Procesos de Decisión de Markov (PDMs) a tiempo discreto para plantear el Problema de Control Óptimo (PCO) y presentar la técnica de Programación Dinámica (PD) para su solución, además se dan las condiciones que garantizan su validez. En el Capítulo 2 se presentan dos problemas para ejemplificar la técnica de programación dinámica; en el primero de ellos se observa los costos generados por el proceso de deterioro de una máquina y el objetivo será minimizar dichos costos. El segundo es un problema de inventario en el cual se observan los costos y ganancias generados por el almacenamiento de un producto así, el objetivo del problema será maximizar la ganancia obtenida. En el Capítulo 3 se describe el método de aleatorización estudiado para reducir el problema de dimensionalidad y se probará que las aproximaciones propuestas resultan ser cercanas al valor verdadero y, se presentan los ejemplos del Capítulo 2 pero ahora se ha implementado el método de aleatorización en su solución. Finalmente, se presenta un capítulo de conclusiones y problemas abiertos, así como dos apéndices donde están incluidos resultados auxiliares, los cuales serán utilizados a lo largo de la tesis y los algoritmos computacionales desarrollados.

ÍNDICE GENERAL

Capítulo 1

Procesos de Decisión de Markov

En este capítulo se presenta el problema de control óptimo (PCO) mediante la teoría de Procesos de Decisión de Markov (PDMs). De manera general, un PDM se encarga de modelar un sistema dinámico cuyos estados son observados de manera periódica por un controlador de forma discreta en el tiempo. Un procedimiento de solución para PDM está basado en el principio de optimalidad de Bellman conocido como Programación Dinámica (PD) (véase [2]). La idea de PD es llevar el problema de control óptimo a un problema equivalente, el cual consiste en resolver una ecuación funcional para la función de valores óptimos, conocida como Ecuación de Programación Dinámica (EPD) (véase [7]). También es incluida la teoría referente a los PDMs con horizonte infinito. Además, se presentan condiciones sobre el modelo, las cuales garantizan la validez de PD.

1.1. Procesos de Decisión de Markov con Horizonte Finito

Definición 1.1.1 *Un **Modelo de Control de Markov** (MCM), estacionario, a tiempo discreto, consiste de una quíntupla:*

$$(X, A, \{A(x) \,|x \in X\}, Q, c) \tag{1.1}$$

donde,

a. *X es un espacio de Borel no vacío (véase, Apéndice A, Definición A.0.1), llamado espacio de* estados*;*

b. *A es un espacio de Borel no vacío, llamado espacio de* acciones o controles*;*

c. *$\{A(x) \,|x \in X\}$ es una familia de subconjuntos medibles, no vacíos $A(x)$ de A, donde $A(x)$ denota el conjunto de* acciones admisibles *cuando el*

sistema se encuentra en el estado $x \in X$. El conjunto $\mathbb{K}$ de parejas de estados acciones admisibles, está definido por

$$\mathbb{K} = \{(x, a) \,|x \in X, a \in A\,(x)\}\,,$$

y se supondrá que es un conjunto medible del espacio producto $X \times A$;

d. *Q es un kérnel estocástico definido en X dado $\mathbb{K}$, llamada* ley de transición*, es decir, para cada $(x, a) \in \mathbb{K}$, $Q\,(\cdot\,|x, a\,)$ es una medida de probabilidad en X, y para cada $B \subset X$, medible, $Q\,(B\,|\cdot\,)$ es una función medible;*

e. *$c : \mathbb{K} \to \mathbb{R}$ es una función medible y se llama la* función de costo de un paso*.*

Observación 1.1.2 *Para cierta clase de problemas en lugar de una función de costo c, es más conveniente considerar una* función de recompensa $r : \mathbb{K} \to \mathbb{R}$.

La dinámica que describe a este sistema estocástico funciona de la forma siguiente: si el sistema al tiempo t se encuentra en el estado $x_t = x \in X$, y la acción $a_t = a \in A(x)$ es aplicada; entonces ocurren dos cosas:

a) se paga un costo $c\,(x, a)$ (o se recibe una recompensa $r(x, a)$); y

b) el sistema se traslada a un nuevo estado x_{t+1} mediante la ley de transición $Q\,(\cdot\,|x, a\,)$ sobre X.

Una vez hecha ésta transición a un nuevo estado, se elige una nueva acción y la dinámica anteriormente descrita se repite.

Observación 1.1.3 *Se supondrá que $\mathbb{K}$ contiene la gráfica de una función medible de X a A, es decir, existe $f : X \to A$ medible, tal que $f\,(x) \in A\,(x)$, para toda $x \in X$. El conjunto de estas funciones es denotado por $\mathbb{F}$ y sus elementos son llamados selectores de la multifunción $x \to A\,(x)$.*

1.2. Políticas

Para introducir el concepto de estrategia o política, considérese un MCM y defina $\mathbb{H}_t$, el espacio de las historias observadas del proceso de control hasta el tiempo t, como

$$\begin{aligned} \mathbb{H}_0 &= X, \\ \mathbb{H}_t &= \mathbb{K} \times \mathbb{H}_{t-1} = \mathbb{K}^t \times X, \end{aligned}$$

para $t = 1, 2, \ldots$. Un elemento h_t de $\mathbb{H}_t$ llamado t-historia es un vector de la forma

$$(x_0, a_0, x_1, a_1, \ldots, a_{t-1}, x_t)\,,$$

donde $(x_i, a_i) \in \mathbb{K}$ para $i = 0, \ldots, t-1$ y $x_t \in X$.

Obsérvese que, para cada t, $\mathbb{H}_t$ es un subespacio de $\mathbf{H}_t := (X \times A)^t \times X$ y $\mathbf{H}_0 := X$.

Definición 1.2.1 *Una política es una sucesión $\pi = \{\pi_t\}$ de kérneles estocásticos, donde cada π_t está definido sobre A dado $\mathbb{H}_t$ y satisface que: $\pi_t(A(x_t)|\, h_t) = 1$ para toda $h_t \in \mathbb{H}_t$ y $t = 0, 1, 2, \ldots$. El conjunto de todas las políticas será denotado por Π.*

De acuerdo con ésta definición, una política $\pi = \{\pi_t\}$ puede interpretarse como una sucesión $\{a_t\}$ de variables aleatorias sobre A, tales que, para cada t-historia y $t = 0, 1, 2, \ldots$, la distribución de a_t es $\pi_t(\cdot\,|h_t)$, la cual está concentrada en el conjunto de acciones admisibles $A(x_t)$.

Se denotará a la familia de probabilidades condicionales sobre A dado X, como $\mathcal{P}(A\,|X)$. Sea Φ el conjunto de todas las probabilidades condicionales φ en $\mathcal{P}(A\,|X)$ tales que para toda $x \in X$ se tiene $\varphi(A(x)\,|x) = 1$.

Definición 1.2.2 *Una política $\pi \in \Pi$ es:*

1. ***Markoviana Aleatorizada*** (Π_{RM}). *Si existe una sucesión $\{\varphi_t\} \subset \Phi$ (definidas sobre A dado X), tales que, $\pi_t(\cdot\,|h_t) = \varphi_t(\cdot\,|x_t)$ para toda $h_t \in \mathbb{H}_t$ y $t = 0, 1, 2, \ldots$.*

2. ***Markoviana Aleatorizada Estacionaria*** (Π_{RS}). *Si existe $\varphi \in \Phi$, tal que: $\pi_t(\cdot\,|h_t) = \varphi(\cdot\,|x_t)$ para toda $h_t \in \mathbb{H}_t$ y $t = 0, 1, 2, \ldots$.*

3. ***Determinista*** (Π_D). *Si existe una sucesión $\{g_t\}$ de funciones medibles $g_t : \mathbb{H}_t \to A$, tales que, para cada $h_t \in \mathbb{H}_t$ y $t = 0, 1, 2, \ldots$, se tiene que $g_t(h_t) \in A(x_t)$ y $\pi_t(\cdot\,|h_t)$ está concentrada en $g_t(h_t)$.*

4. ***Determinista Markoviana*** (Π_{DM}). *Si existe una sucesión $\{f_t\}$ de funciones medibles $f_t : X \to A$ (o $f_t \in \mathbb{F}$), tales que, $f_t(x_t) \in A(x_t)$ y $\pi_t(\cdot\,|h_t)$ está concentrada en $f_t(x_t)$ para cada $h_t \in \mathbb{H}_t$ y $t = 0, 1, 2, \ldots$.*

5. ***Determinista Markoviana Estacionaria*** (Π_{DS}). *Si existe una función medible $f : X \to A$ (o $f \in \mathbb{F}$), tal que, $f(x_t) \in A(x_t)$ y $\pi_t(\cdot\,|h_t)$ está concentrada en $f(x_t)$ para cada $h_t \in \mathbb{H}_t$ y $t = 0, 1, 2, \ldots$.*

Observación 1.2.3 *Note que $\Pi_{RS} \subset \Pi_{RM} \subset \Pi$ y $\Pi_{DS} \subset \Pi_{DM} \subset \Pi_D \subset \Pi$.*

Sea $(\Omega, \mathcal{F})$ el espacio medible que consiste del espacio muestral canónico $\Omega := \mathbf{H}_\infty = (X \times A)^\infty$ y $\mathcal{F}$ su correspondiente σ-álgebra producto. Los elementos de Ω son de la forma $w = (x_0, a_0, x_1, a_1, \cdots)$ con $x_t \in X$ y $a_t \in A$ para toda $t = 0, 1, \ldots$, las proyecciones x_t y a_t de Ω sobre X y A son llamados estado y acción, respectivamente. Obsérvese que $\mathbb{H}_\infty = \mathbb{K}^\infty \subset \Omega$ es el espacio de historias $(x_0, a_0, x_1, a_1, \cdots)$ con $(x_t, a_t) \in \mathbb{K}$ para toda $t = 0, 1, \ldots$.

Sean $\pi \in \Pi$ una política arbitraria y ν una medida de probabilidad sobre X, conocida como la distribución inicial. Entonces por el Teorema de Ionescu-Tulcea (véase [1] y Apéndice A, Teorema A.0.7), existe una única medida de probabilidad P_ν^π sobre $(\Omega, \mathcal{F})$. Además, para cada $C \in \mathcal{B}(A)$, cada $B \in \mathcal{B}(X)$, y $h_t \in \mathbb{H}_t$ y $t = 0, 1, 2, \cdots$, se tiene que

$$P_\nu^\pi(x_0 \in B) = \nu(B), \tag{1.2}$$

$$P_\nu^\pi(\, a_t \in C |\, h_t) = \pi_t(C\,|h_t\,), \tag{1.3}$$

$$P_\nu^\pi(\, x_{t+1} \in B |\, h_t, a_t) = Q(B\,|x_t, a_t\,). \tag{1.4}$$

Definición 1.2.4 *El proceso estocástico* $((\Omega, \mathcal{F}, P_\nu^\pi)\,, \{x_t\})$ *es llamado un* Proceso de Control de Markov a tiempo discreto *o* Proceso de Decisión de Markov (PDM). *La esperanza con respecto a* P_ν^π *será denotada por* E_ν^π.

Observación 1.2.5 *Si* ν *está concentrada en un estado inicial* $x \in X$*, entonces se puede escribir* P_ν^π *y* E_ν^π *como* P_x^π *y* E_x^π*, respectivamente.*

De (1.4) se tiene que, la distribución del estado x_{t+1} sólo depende de la pareja estado-acción (x_t, a_t), dicha condición es una propiedad de tipo Markov pero, en general el proceso $\{x_t\}$ no es una cadena de Markov. Sin embargo, si π es una política de clase markoviana, entonces se garantiza que $\{x_t\}$ es un proceso de Markov (véase [7] y [10]).

1.3. Criterio de Rendimiento

Cada PDM estará dotado de una función real, llamada criterio de rendimiento, que medirá de alguna manera la calidad de cada política, a través de la sucesión de costos que genera. A continuación se define el criterio de rendimiento utilizado en este trabajo.

Considérese un Modelo de Control de Markov fijo y un conjunto de políticas Π.

Se define para cada $x \in X$ y $\pi \in \Pi$

$$V\,(\pi, x) = E_x^\pi \left[\sum_{t=0}^{N-1} c\,(x_t, a_t) + c_N\,(x_N) \right]. \tag{1.5}$$

$V\,(\pi, x)$ es conocido como el *Costo Total Acumulado* y $c_N\,(x_N)$ es el costo terminal. Al entero positivo N se le conoce como horizonte del problema, el cual representa el número de etapas en el cual el sistema está operando y puede ser finito o infinito.

Definición 1.3.1 *Para cada* $x \in X$ *se define*

$$V^*\,(x) = \inf_{\pi \in \Pi} V\,(\pi, x)\,,$$

V^* *se le llama funciones de* valores óptimos o valor óptimo.

Definición 1.3.2 *Una política* $\pi^* \in \Pi$*, es* óptima*, si*

$$V\,(\pi^*, x) = \inf_{\pi \in \Pi} V\,(\pi, x)\,,$$

$x \in X$.

El *Problema de Control Optimo* consiste en determinar una política que optimice al criterio de rendimiento.

1.4. Programación Dinámica

Una de las herramientas más usadas para resolver el problema de control óptimo, es conocida como Programación Dinámica (PD). Bajo condiciones adecuadas sobre el MCM este procedimiento permite determinar la función de valor óptimo y/o a la política óptima.

El teorema de Programación Dinámica, dado más adelante, tiene como suposición la existencia de selectores $f \in \mathbb{F}$, los cuales minimizan el lado derecho de la Ecuación de Programación Dinámica (EPD, Ecuación 1.7) en cada etapa. Esta suposición es referida como condición de selección medible. La cual se da a continuación

Suposición 1.4.1 *Considere un Modelo de Control de Markov y una función medible $u : X \to \mathbb{R}$ dada, tal que*

$$u^*(x) = \inf_{A(x)} \left\{ c(x,a) + \int_X u(y)\, Q(dy\,|x,a) \right\},$$

$x \in X$ es medible y existe un selector $f \in \mathbb{F}$ tal que la función entre llaves alcanza su mínimo en $f(x) \in A(x)$ para toda $x \in X$; es decir,

$$u^*(x) = c(x, f(x)) + \int_X u(y)\, Q(dy\,|x, f(x)).$$

Si esto ocurre entonces se puede cambiar ínfimo por mínimo.

En muchos problemas la suposición anterior se puede verificar directamente, pero desde un punto de vista teórico, es conveniente tener condiciones generales, mismas que se obtienen de los teoremas de selección medible. Considere las siguientes condiciones sobre el MCM,

Condición 1.4.2 a) *$A(x)$ es compacto para toda $x \in X$;*

b) *La función de costo c es semicontinua inferiormente (l.s.c por sus siglas en inglés, véase Apéndice A, Definición A.0.3) en $A(x)$ para cada $x \in X$;*

c) *La función $v'(x,a) := \int_X v(y)\, Q(dy\,|x,a)$, $(x,a) \in \mathbb{K}$, satisface una de las siguientes condiciones,*

- **(c1)** *$v'(x,\cdot)$ es l.s.c en $A(x)$ para cada $x \in X$ y cada función v continua y acotada sobre X; ó*
- **(c2)** *$v'(x,\cdot)$ es l.s.c en $A(x)$ para cada $x \in X$ y cada función v medible y acotada sobre X.*

Condición 1.4.3 a) *$A(x)$ es compacto para toda $x \in X$;*

b) *La función de costo c es l.s.c y acotada inferiormente;*

c) *La ley de transición Q es:*

(c1) *Débilmente continua (véase Apéndice A, Definición A.0.6); ó*

(c2) *Fuertemente continua (véase Apéndice A, Definición A.0.6).*

Condición 1.4.4 a) *La función de costo c es l.s.c, acotada inferiormente e inf-compacta (véase Apéndice A, Definición A.0.5) sobre* $\mathbb{K}$*;*

b) *La ley de transición Q es:*

(b1) *Débilmente continua; ó*

(b2) *Fuertemente continua.*

El siguiente teorema es conocido como el Teorema de Programación Dinámica y está basado en el principio de optimalidad de Bellman (véase [2]).

Teorema 1.4.5 *Sean* $V_0, V_1, \ldots, V_N$ *funciones sobre* X *definidas por*

$$V_N(x) := c_N(x), \tag{1.6}$$

y para cada $t = N-1, N-2, \ldots, 1, 0,$

$$V_t(x) := \min_{a \in A(x)} \left[c(x,a) + \sum_{y \in X} V_{t+1}(y)\, Q(y \,|x, a) \right]. \tag{1.7}$$

Se supone que estas funciones son medibles y que para cada $t = 0, 1, 2, \ldots, N-1$*, existe un selector* $f_t \in \mathbb{F}$ *con* $f_t(x) \in A(x)$*, tal que*

$$V_t(x) = c(x, f_t(x)) + \sum_{X} V_{t+1}(y)\, Q(y \,|x, f_t(x)). \tag{1.8}$$

Entonces, la política determinista de Markov $\pi^* = \{f_0, f_1, \ldots, f_{N-1}\}$ *es óptima y la función de valor óptimo* V^* *es* V_0*, es decir, para toda* $x \in X$ *se tiene que*

$$V^*(x) = V(\pi^*, x) = V_0(x). \tag{1.9}$$

La relación (1.7) es conocida como Ecuación de Programación Dinámica (EPD) junto con su condición inicial (1.6).

Demostración. Sean $x \in X$ y $\pi = \{\pi_t\}$ una política arbitraria, y sea

$$C_t(\pi, x) := E^{\pi} \left[\sum_{n=t}^{N-1} c(x_n, a_n) + c_N(x_N) \,|x_t = x \right], \tag{1.10}$$

$$C_N(\pi, x) := E^{\pi} [c_N(x_N) \,|x_N = x] = c_N(x), \tag{1.11}$$

$t = 0, 1, 2, \ldots, N-1$. $C_t(\pi, x)$ es llamado el costo total del instante t a $N-1$ cuando se usa la política π y $x_t = x$. En particular note que

$$V(\pi, x) = C_0(\pi, x). \tag{1.12}$$

Para demostrar este teorema, se tiene que mostrar que, para todo $x \in X$ y $t = 0, 1, 2, \ldots, N$, se tiene que

$$C_t(\pi, x) \geq V_t(x), \tag{1.13}$$

cuando $\pi = \pi^*$,

$$C_t(\pi^*, x) = V_t(x). \tag{1.14}$$

En particular, si $t = 0$, se tiene que

$$C_0(\pi, x) = V(\pi, x) \geq V_0(x)$$

y si se utiliza a π^*:

$$C_0(\pi^*, x) = V(\pi^*, x) = V^*(x) = V_0(x),$$

lo cual cumple lo deseado en (1.9). La prueba de las relaciones (1.13)-(1.14) es por inducción hacía atrás. Primero obsérvese que (1.13)-(1.14) se cumplen para $t = N$, ya que, de (1.11) y (1.6)

$$C_N(\pi, x) = V_N(x) = c_N(x).$$

Ahora, suponga que para alguna $t = N-1, \ldots, 0$ se tiene que

$$C_{t+1}(\pi, x) \geq V_{t+1}(x),\ x \in X. \tag{1.15}$$

Entonces, de (1.10) y (1.3)-(1.4),

$$\begin{aligned}
C_t(\pi, x) &= E^\pi\left[\sum_{n=t}^{N-1} c(x_n, a_n) + c_N(x_N) \,|x_t = x\right], \\
&= E^\pi\left[c(x_t, a_t) + \sum_{n=t+1}^{N-1} c(x_n, a_n) + c_N(x_N) \,|x_t = x\right], \\
&= E^\pi\left[c(x_t, a_t) \,|x_t = x\right] + E^\pi\left[\sum_{n=t+1}^{N-1} c(x_n, a_n) + c_N(x_N) \,|x_t = x\right], \\
&= \sum_{a \in A} c(x, a)\, \pi_t(a \,|x) + E^\pi\left[\sum_{n=t+1}^{N-1} c(x_n, a_n) + c_N(x_N) \,|x_t = x\right],
\end{aligned}$$

y por propiedades de la esperanza condicional, se tiene que,

$$\begin{aligned}
&E^\pi\left[\sum_{n=t+1}^{N-1} c(x_n, a_n) + c_N(x_N) \,|x_t = x\right] \\
= \;& E^\pi\left[E^\pi\left(\sum_{n=t+1}^{N-1} c(x_n, a_n) + c_N(x_N) \,|x_{t+1} = y\right) |x_t = x\right].
\end{aligned}$$

Por otro lado,

$$E^{\pi}\left[\sum_{n=t+1}^{N-1} c\left(x_n, a_n\right) + c_N\left(x_N\right) | x_{t+1} = y\right] = C_{t+1}\left(\pi, x\right).$$

Así,

$$\begin{aligned}
C_t\left(\pi, x\right) &= \sum_{a\in A} c\left(x, a\right)\pi_t\left(a\,|x\right) + E^{\pi}\left[C_{t+1}\left(\pi, y\right)|x_t = x\right] \\
&= \sum_{a\in A} c\left(x, a\right)\pi_t\left(a\,|x\right) + \sum_{a\in A}\sum_{y\in X} C_{t+1}\left(\pi, y\right)Q\left(y\,|x, a\right)\pi_t\left(a\,|x\right) \\
&= \sum_{a\in A}\left[c\left(x, a\right) + \sum_{y\in X} C_{t+1}\left(\pi, y\right)Q\left(y\,|x, a\right)\right]\pi_t\left(a\,|x\right) \\
&\geq \sum_{a\in A}\left[c\left(x, a\right) + \sum_{y\in X} V_{t+1}\left(y\right)Q\left(y\,|x, a\right)\right]\pi_t\left(a\,|x\right) \\
&\geq \min_{a\in A(x)}\left[c\left(x, a\right) + \sum_{y\in X} V_{t+1}\left(y\right)Q\left(y\,|x, a\right)\right] \\
&= V_t\left(x\right),
\end{aligned}$$

y bajo la hipótesis de inducción se tiene que

$$C_{t+1}\left(\pi^*, x\right) = V_{t+1}\left(x\right).$$

■

De lo anterior se muestra que $V_t\left(x\right)$ es el óptimo del problema desde el tiempo t a N, es decir,

$$V_t\left(x\right) = \inf_{\pi\in\Pi} C_t\left(\pi, x\right),$$

para toda $x \in X$ y $t = 0, 1, \ldots, N$.

Así, se ha calculado para cada tiempo t el costo óptimo de t en adelante, con esta interpretación de $V_t\left(x\right)$ es posible caracterizar a la ecuación de programación dinámica. Para probar esto, sea $\pi = \{\pi_t, \pi_{t+1}, \ldots, \pi_{N-1}\}$ una política tal que $\pi_t = f \in \mathbb{F}$ es un selector arbitrario y $\{\pi_{t+1}, \ldots, \pi_{N-1}\}$ es una política óptima para el problema de $t+1$ hasta N, entonces

$$\begin{aligned}
C_t\left(\pi, x\right) &= c\left(x, f\right) + \sum_{y\in X} V_{t+1}\left(y\right)Q\left(y\,|x, f\right) \\
&\geq \min_{a\in A(x)}\left\{c\left(x, a\right) + \sum_{y\in X} V_{t+1}\left(y\right)Q\left(y\,|x, a\right)\right\},
\end{aligned}$$

para todo $x \in X$. De aquí, se tiene que

$$V_t\left(x\right) \geq \min_{a\in A(x)}\left\{c\left(x, a\right) + \sum_{y\in X} V_{t+1}\left(y\right)Q\left(y\,|x, a\right)\right\}.$$

Para ver la desigualdad inversa nótese que

$$V_t(x) \leq c(x,f) + \sum_{y \in X} V_{t+1}(y)\, Q(y\,|x,f)$$

lo cual implica que

$$V_t(x) \leq \min_{a \in A(x)} \left\{ c(x,a) + \sum_{y \in X} V_{t+1}(y)\, Q(y\,|x,a) \right\},$$

de donde se prueba lo anteriormente afirmado.

Observación 1.4.6 *Bajo cualquiera de las condiciones (1.4.2)-(1.4.4) se tiene que, para cada función $u : X \to \mathbb{R}$ medible y no negativa, la condición de selección medible (1.4.1) se cumple.*

1.5. Procesos de Decisión de Markov con Horizonte Infinito

A continuación se describirá la técnica de solución para PDMs con horizonte infinito. Considere el criterio de rendimiento de Costo Total Descontado,

$$V(\pi, x) := E_x^{\pi} \left[\sum_{t=0}^{\infty} \alpha^t c(x_t, a_t) \right], \tag{1.16}$$

$\pi \in \Pi, x \in X$. Donde $\alpha \in (0,1)$ es un factor de descuento dado. Una política π^* que satisface

$$V(\pi^*, x) = \inf_{\pi \in \Pi} V(\pi, x) =: V^*(x), \tag{1.17}$$

para toda $x \in X$ será llamada $\alpha-$descontada óptima, y V^* función de valor $\alpha-$descontada óptima.

Se supondrá en esta sección que el costo c es no negativo y se define al n-ésimo costo descontado como

$$V_n(\pi, x) := E_x^{\pi} \left[\sum_{t=0}^{n-1} \alpha^t c(x_t, a_t) \right], \tag{1.18}$$

así, por el teorema de la convergencia monótona, se tiene que

$$V(\pi, x) = \lim_{n \to \infty} V_n(\pi, x). \tag{1.19}$$

Una función medible $v : X \to \mathbb{R}$ es solución de la $\alpha-$descontada Ecuación de Optimalidad ($\alpha-$EO), si satisface que

$$v(x) = \min_{a \in A(x)} \left\{ c(x,a) + \alpha \int_X v(y) Q(dy\,|x,a) \right\}, \tag{1.20}$$

$x \in X$. Se tiene que la función de valor óptimo, V^* es solución de la $\alpha-$EO, es decir,

$$V^*(x) = \min_{a \in A(x)} \left\{ c(x,a) + \alpha \int_X V^*(y) Q(dy\,|x, a\,) \right\}, \tag{1.21}$$

$x \in X$. En efecto, para ello se usaran las funciones de iteración de valores definidas por,

$$v_n(x) := \min_{a \in A(x)} \left\{ c(x,a) + \alpha \int_X v_{n-1}(y) Q(dy\,|x, a\,) \right\} \tag{1.22}$$

para toda $x \in X$ y $n = 1, 2, \ldots$, con $v_0\,(\cdot) \equiv 0$. Note que v_n es la función de valor óptimo para el n-ésimo costo descontado V_n, con costo terminal cero, es decir,

$$v_n(x) = \inf_{\pi \in \Pi} V_n\,(\pi, x)\,, \tag{1.23}$$

$x \in X$. Además se tiene que, para cada $x \in X$

$$V^*(x) = \lim_{n \to \infty} v_n(\pi, x). \tag{1.24}$$

Tomando límite cuando $n \to \infty$ en (1.22) y usando (1.24), se obtiene (1.21), si se cumple el intercambio de límite con el mínimo. Este procedimiento es conocido como *método de aproximaciones sucesivas*, y requiere condiciones de selección medible, las cuales se darán a continuación.

Condición 1.5.1 a) *El costo c es l.s.c, no negativo e inf-compacto en* $\mathbb{K}$;

b) *Q es fuertemente continua.*

Para la parte a), que c sea no negativa es equivalente a que sea acotada inferiormente, ya que, si $c \geq m$ para alguna constante m, entonces $c' = c - m \geq 0$. La siguiente condición permite garantizar que $V^*\,(x)$ es finita para cualquier $x \in X$.

Condición 1.5.2 *Existe una política π tal que, para cada $x \in X$ se tiene que $V\,(\pi, x) < \infty$.*

Observación 1.5.3 *Esta condición claramente se cumple cuando el costo c es acotado, ya que, si $0 \leq c \leq M$ entonces para cada $x \in X$ y $\pi \in \Pi$*

$$\begin{aligned} V(\pi, x) \quad &: \quad = E_x^\pi \left[\sum_{t=0}^{\infty} \alpha^t c(x_t, a_t) \right] \\ &\leq \quad \sum_{t=0}^{\infty} \alpha^t M \\ &= \quad \frac{M}{1-\alpha} < \infty. \end{aligned}$$

Definición 1.5.4 *$\mathcal{M}(X)^+$ denota el conjunto de funciones medibles y no negativas definidas sobre X, y, para cada $u \in \mathcal{M}(X)^+$, Tu es la función sobre X definida como*

$$Tu(x) := \min_{a \in A(x)} \left\{ c(x,a) + \alpha \int_X u(y) Q(dy\,|x,a) \right\}, \tag{1.25}$$

$x \in X$.

Note que, usando el operador T, se pueden escribir la ecuación de optimalidad (1.27) y las funciones de iteración de valores (1.22) como

$$V^* = TV^*, \text{ y } v_n = Tv_{n-1} \tag{1.26}$$

para $n \geq 1$, ($v_0 := 0$), respectivamente. Además, se sabe que el operador T es una contracción módulo α, (véase [7] y [9]).

Teorema 1.5.5 *Suponga que las Condiciones (1.5.1) y (1.5.2) son válidas. Entonces*

(a) *La función de valor óptimo V^* es solución minimal de la $\alpha-EO$, es decir, para cada $x \in X$*

$$V^*(x) = \min_{a \in A(x)} \left\{ c(x,a) + \alpha \int_X V^*(y) Q(dy\,|x,a) \right\}, \tag{1.27}$$

y si u es otra solución de la $\alpha-EO$, entonces, $u(\cdot) \geq V^*(\cdot)$;

(b) *Existe un selector $f_* \in \mathbb{F}$ tal que $f_*(x) \in A(x)$ y alcanza un mínimo en la $\alpha-EO$, es decir, para cada $x \in X$*

$$V^*(x) = c(x, f_*) + \alpha \int_X V^*(y) Q(dy\,|x, f_*), \tag{1.28}$$

(c) *Si π^* es una política tal que $V(\pi^*, \cdot)$ es una solución de la $\alpha-EO$ y satisface que para cada $x \in X$*

$$\lim_{n \to \infty} \alpha^n E_x^{\pi} V(\pi^*, x_n) = 0, \tag{1.29}$$

entonces $V(\pi^*, \cdot) = V^*(\cdot)$; de aquí, π^* es óptima, concluyendo que, si ocurre lo anterior, entonces π^* es una política óptima, si y sólo si, $V(\pi^*, \cdot)$ satisface la $\alpha-EO$.

(d) *Si existe una política óptima, entonces existe una que es determinista estacionaria.*

Demostración. Véase [7]. ■

1.6. Variantes de la Ecuación de Programación Dinámica

A veces es conveniente reescribir la EPD (1.6)-(1.7) en otras formas equivalentes y apropiadas. En está sección, se presentan algunas de las formas que son más frecuentes.

Ecuación en Diferencias. En muchas aplicaciones, la ley de transición Q es inducida por una ecuación en diferencias de la forma siguiente:

$$x_{t+1} = F\left(x_t, a_t, \xi_t\right), \tag{1.30}$$

para $t = 0, 1, 2, \ldots$ y x_0 dado, donde $\{\xi_t\}$ es una sucesión de variables aleatorias independientes e idénticamente distribuidas (i.i.d) que toman valores en algún espacio S con distribución común μ e independiente del estado inicial x_0 y $F : X \times A \times S \to X$ es una función medible conocida. Entonces, la ley de transición Q puede escribirse

$$\begin{aligned} Q\left(B \,|x, a\right) &= \Pr\left(x_{t+1} \in B \,|x_t = x, a_t = a\right) \\ &= \Pr\left(F\left(x_t, a_t, \xi_t\right) \in B \,|x_t = x, a_t = a\right) \\ &= \Pr\left(F\left(x, a, \xi_t\right) \in B\right) \\ &= \mu\left(\{s \in S \,|F\left(x, a, s\right) \in B\}\right) \\ &= \int_S I_B\left[F\left(x, a, s\right)\right] \mu\left(ds\right) \\ &= E\left[I_B\left(F\left(x, a, s\right)\right)\right]. \end{aligned}$$

para todo $B \in \mathcal{B}\left(X\right)$ y $(x, a) \in \mathbb{K}$.

Por el teorema de cambio de variable para funciones medibles sobre X (véase [1]), se tiene que si u es medible sobre X. Entonces

$$\begin{aligned} E\left[u\left(x_{t+1}\right) |x_t = x, a_t = a\right] &= \int_X u\left(y\right) Q\left(dy \,|x, a\right) \\ &= \int_S u\left[F\left(x, a, s\right)\right] \mu\left(ds\right) \\ &= E\left[u\left(F\left(x, a, \xi_0\right)\right)\right]. \end{aligned}$$

Entonces, la EPD (1.6)-(1.7) puede reescribirse como,

$$V_N\left(x\right) = c_N\left(x\right) \tag{1.31}$$

$$\begin{aligned} V_t(x) &= \min_{A(x)} \left\{ c(x,a) + \int_X V_{t+1}(y)\, Q(dy\,|x,a) \right\} \\ &= \min_{A(x)} \left\{ c(x,a) + \int_S V_{t+1}\left[F(x,a,s)\right] \mu(ds) \right\} \\ &= \min_{A(x)} \left\{ c(x,a) + E\left[V_{t+1}\left[F(x,a,\xi_t)\right]\right] \right\} \end{aligned} \tag{1.32}$$

para toda $x \in X$ y $t = N-1, N-2, \ldots, 1, 0$.

Forma hacia adelante de la ecuación de PD. Sea V_t como las funciones en (1.6)-(1.7), y se define $v_t = V_{N-t}$ $(t = 0, 1, \ldots, N)$. Entonces se puede escribir la ecuación de programación dinámica "hacía adelante" de la siguiente manera

$$v_0 = c_N(x) \tag{1.33}$$

$$v_t(x) = \min_{A(x)} \left\{ c(x,a) + \sum_{y \in X} v_{t-1}(y)\, Q(y\,|x,a) \right\} \tag{1.34}$$

si $t = 1, \ldots, N$. Además, si $f_t \in \mathbb{F}$ es como en (1.8), entonces $g_t. = f_{N-t}$ $(t = 1, \ldots, N)$ es un minimizador para (1.34). Entonces, en términos de las funciones v_t la conclusión del Teorema 1.4.5 puede reescribirse como: $\tilde{\pi} = \{g_N, \ldots, g_1\}$ es una política óptima y la función de valor es $V^*(\cdot) = v_N(\cdot) = V(\tilde{\pi}, \cdot)$, es decir,

$$v_n(x) = \inf_{\Pi} V(\pi, x)$$

para $x \in X$ y, donde $V(\pi, x)$ es como se dio anteriormente.

Las funciones v_t en (1.33)-(1.34) son conocidas como las funciones de iteración de valores.

Maximización de Recompensas. En lugar de considerar una función de costo c, considere una función de recompensa r. Entonces el criterio de rendimiento de costo total acumulado, $V(\pi, x)$ se convierte en criterio de recompensa total esperada,

$$V(\pi, x) = E_x^{\pi} \left(\sum_{t=0}^{N-1} r(x_t, a_t) + r_N(x_n) \right),$$

donde r_N es una función de recompensa dada. Así, el problema de control para este caso es encontrar una política que maximice a V, por lo cual la función de valor esta ahora dada por,

$$V^*(x) := \sup_{\Pi} V(\pi, x),$$

$x \in X$. Y, con los cambios necesarios todo lo anteriormente establecido para el caso de costos continúa siendo válido. Por lo tanto, la EPD (1.6)-(1.7) se convierte en,

$$V_N(x) = r_N(x)$$

$$V_t(x) = \max_{A(x)} \left\{ r(x,a) + \sum_{y \in X} V_{t+1}(y)\, Q(y \,|x,a) \right\}$$

para $t = N-1, \ldots, 1, 0$.

Modelo de Control No Estacionario. En lugar de considerar el modelo de control de Markov dado en (1.1), considere un modelo de control no estacionario $(X_t, A_t, \{A_t(x) \,|x \in X_t\}, Q_t, c_t)$, $t = 0, 1, 2, \ldots$. Entonces, se tiene que el la prueba del Teorema 1.4.5 continúa siendo válida si se reemplazan X, A, Q y c por X_t, A_t, Q_t y c_t respectivamente. De donde, (1.7) se convierte, para cada $x \in X_t$ y $t = 0, 1, \ldots N-1$, en

$$V_t(x) = \min_{a \in A_t(x)} \left[c_t(x,a) + \sum_{y \in X_t} V_{t+1}(y)\, Q(y \,|x,a) \right]. \tag{1.35}$$

Costo Descontado. Suponga que, en lugar de el criterio de rendimiento dado en (1.5), el costo esperado es de la forma siguiente,

$$V(\pi, x) := E_x^{\pi} \left[\sum_{t=0}^{N-1} \alpha^t c(x_t, a_t) + \alpha^N c(x_N) \right], \tag{1.36}$$

donde $\alpha > 0$ es un número dado llamado el *factor de descuento*. El correspondiente modelo de control puede ser visto como un modelo no estacionario con X, A, y Q fijos y un costo variando por etapa, $c_t(x,a) := \alpha^t c(x,a)$. Entonces, de (1.36) y (1.6), se obtiene la siguiente ecuación de programación dinámica,

$$V_N(x) = \alpha^N c_N(x),$$

y para $t = N-1, N-2, \ldots, 1, 0$

$$V_t(x) = \min_{a \in A(x)} \left[\alpha^t c(x,a) + \sum_{y \in X} V_{t+1}(y)\, Q(y \,|x,a) \right].$$

Se puede reescribir está ecuación en términos de funciones $J_t(\cdot) := \alpha^{-t} V_t(\cdot)$, $t = 0, \ldots, N$, para obtener

$$J_N(x) = c_N(x), \tag{1.37}$$

y, para $t = N-1, N-2, \ldots, 0$,

$$J_t(x) = \min_{a \in A(x)} \left[c(x,a) + \sum_{y \in X} J_{t+1}(y)\, Q(y \,|x,a) \right], \tag{1.38}$$

y el Teorema 1.4.5 continúa siendo válido cuando las funciones V_t son reemplazadas por J_t. Así, la política $\pi^* = \{f_0, \ldots, f_{N-1}\}$ con $f_t \in F$ un minimizador de (1.38) es óptima para el criterio de costo descontado dado en (1.36) y $V(\pi^*, x) = J_0(x)$.

Capítulo 2

Ejemplos de Programación Dinámica

En este capítulo se darán ejemplos en los cuales se utiliza la técnica de programación dinámica. El primero de éstos está relacionado con la operación eficiente de una máquina (véase [4] y [14]), en el cual se busca minimizar los costos ocasionados por el funcionamiento de la máquina y así, lo que se quiere es decidir en qué instante reemplazar la máquina por una nueva; el segundo es un problema de inventarios (véase [7] y [10]), para este problema se analiza la cantidad de producto que se encuentra en un almacén y en cada etapa se busca decidir la cantidad adecuada de nuevo producto que debe comprarse. En este caso se busca minimizar los costos ocasionados para poder obtener una mayor ganancia.

2.1. Reemplazamiento de Máquinas

Considere un problema de operación de una máquina que es revisada en N periodos y que puede presentar un deterioro de acuerdo a uno de los siguientes niveles: $\{1, 2, \ldots, n\}$. Suponga que encontrarse en el nivel x es mejor que estar en el nivel $x+1$, y el nivel 1 es el de condiciones perfectas en las que la máquina puede operar.

Sea $g : \{1, 2, ..., n\} \to \mathbb{R}_+$ la función de costo de operación por periodo asociado al nivel i, se supone que

$$g(1) \leq g(2) \leq \cdots \leq g(n).$$

Durante un periodo de operación el nivel de la máquina puede empeorar o quedarse igual.

Se considera la probabilidad de transición como:

$$p_{xy} = \begin{cases} p(y|x), & y \geq x, \\ 0, & y < x. \end{cases}$$

Supóngase también que al comenzar cada periodo se conoce el nivel en que la máquina se encuentra y, así se tienen las opciones siguientes:

1. Dejar que la máquina opere un periodo más, en el nivel en que se encuentra.

2. Reparar la máquina hasta que se encuentre en condiciones perfectas, es decir, hasta que se encuentre en el nivel 1, pagando un costo de reparación R (fijo).

Suponga que la máquina, una vez reparada, permanecerá en el estado 1 por al menos un periodo. En los periodos siguientes, puede deteriorarse para los estados $y > 1$ según las probabilidades de transición p_{xy}.

Así el objetivo es decidir sobre el nivel de deterioro en el cual vale la pena pagar el costo de reparación de la máquina, de tal modo que se obtengan costos de operación más bajos en el futuro. Obsérvese que la decisión también está afectada por el periodo en el que se encuentra la máquina, por ejemplo, se estaría menos inclinado por reparar la máquina cuando hay pocos periodos a la izquierda.

Identificando las componentes del MCM, se tiene que,

1. Los estados representan los niveles de deterioro, es decir,

$$X = \{1, 2, \ldots, n\};$$

2. Las acciones y acciones admisibles coinciden y están dados por

$$A = A(x) = \{0, 1\},$$

donde 0 representa permitir que la máquina opere un periodo más y 1, reparar la máquina.

3. La ley de transición está dada por,

$$p_{xy} = p^0_{xy} = \begin{bmatrix} p_{11} & p_{12} & \cdot & \cdot & \cdot & p_{1n} \\ 0 & p_{22} & \cdot & \cdot & \cdot & p_{2n} \\ 0 & 0 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 0 & p_{ii} & \cdot & p_{in} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & 0 & p_{nn} \end{bmatrix}$$

si se decide permitir operar por un periodo más, y

$$p^1_{xy} = \begin{bmatrix} 1 & 0 & \cdot & \cdot & \cdot & 0 \\ 1 & 0 & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & 0 & \cdot & \cdot & \cdot & 0 \end{bmatrix}$$

cuando se decide reparar la máquina.

4. El costo de operación está dado por $g(\cdot)$ si se decide operar por un periodo más la máquina y por $g(\cdot) + R$ si se decide repararla.

Así, considerando el criterio de costo total acumulado,

$$V(\pi, x) := E_x^{\pi}\left[\sum_{t=0}^{N-1} g_t(x_t, a_t) + g_N(x_N)\right].$$

Entonces, se tiene que, la Ecuación de Programación Dinámica es:

$$V_N(x) = 0$$

$$V_t(x) = \min_{a e A(x)} \{g(x,a) + \mathbb{E}(V_{t+1})\}.$$

donde,

$$\begin{aligned}\mathbb{E}(V_{t+1}) &= \sum_{y\in X} p_{xy}^0 V_{t+1}(y) \\ &= \sum_{y=x}^{n} p_{xy} V_{t+1}(y).\end{aligned}$$

en caso de que continúe trabajando la máquina. Y, en el otro caso,

$$\begin{aligned}\mathbb{E}(V_{t+1}) &= \sum_{y\in X} p_{xy}^1 V_{t+1}(y) \\ &= V_{t+1}(1).\end{aligned}$$

En general la EPD es,

$$V_N(x) = g_N(x_N) = 0,$$

$$\begin{aligned}V_t(x) &= \min\left\{g(x) + \sum_{y=x}^{n} p(y|x) V_{t+1}(y), R + g(1) + V_{t+1}(1)\right\}, \\ t &\in \{N-1, N-2, \ldots, 1, 0\}, \; x \in \{1, 2, \ldots, n\},\end{aligned}$$

donde la primera ecuación representa permitir que la máquina continue trabajando por un periodo más y la segunda reparar la máquina.

2.1.1. Ejemplo Numérico

En un caso particular, suponga que $X = \{1, 2, 3, 4, 5\}$ y la máquina es revisada durante dos periodos. La matriz de transición esta dada por

$$p_{xy} = \begin{bmatrix} \frac{2}{10} & \frac{1}{10} & \frac{3}{10} & \frac{2}{10} & \frac{2}{10} \\ 0 & \frac{5}{10} & \frac{3}{10} & \frac{1}{10} & \frac{1}{10} \\ 0 & 0 & \frac{7}{10} & \frac{2}{10} & \frac{1}{10} \\ 0 & 0 & 0 & \frac{4}{10} & \frac{6}{10} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

el costo está definido como $g(x) = x + 1$, para $x = 1, 2, 3, 4, 5$. Y el costo de reparación es $R = \frac{14}{5}$.

Utilizando programación dinámica

$$V_2(x) = 0.$$

En la etapa siguiente

$$\begin{aligned} V_1(x) &= \min\left\{ g(x) + \sum_{y=x}^{5} p_{xy} V_2(y), R + g(1) + V_2(1) \right\} \\ &= \min\left\{ x + 1, \frac{24}{5} \right\}. \end{aligned}$$

De esta forma,

$$V_1(x) = \begin{cases} x + 1, & \text{si } x = 1, 2, 3, \\ \frac{24}{5}, & \text{si } i = 4, 5; \end{cases}$$

es decir, para los tres primeros estados es mejor dejar trabajar un periodo más la máquina mientras que para los restantes es mejor mandar a reparar la máquina.

Para la siguiente etapa

$$\begin{aligned} V_0(x) &= \min\left\{ g(x) + \sum_{y=x}^{5} p_{xy} V_1(y), R + g(1) + V_1(1) \right\} \\ &= \min\left\{ x + 1 + \sum_{y=x}^{5} p_{xy} V_1(x), \frac{34}{5} \right\}. \end{aligned}$$

Ahora, resolviendo para cada estado

$$\begin{aligned} V_0(1) &= \min\left\{ 1 + 1 + \sum_{y=1}^{5} p_{1y} V_1(y), \frac{34}{5} \right\} \\ &= \min\left\{ \frac{291}{50}, \frac{34}{5} \right\} = \frac{291}{50}, \end{aligned}$$

$$\begin{aligned} V_0(2) &= \min\left\{ 2 + 1 + \sum_{y=2}^{5} p_{2y} V_1(y), \frac{34}{5} \right\} \\ &= \min\left\{ \frac{333}{50}, \frac{34}{5} \right\} = \frac{333}{50}, \end{aligned}$$

$$\begin{aligned} V_0(3) &= \min\left\{ 3 + 1 + \sum_{y=3}^{5} p_{3y} V_1(y), \frac{34}{5} \right\} \\ &= \min\left\{ \frac{206}{25}, \frac{34}{5} \right\} = \frac{34}{5}, \end{aligned}$$

$$\begin{aligned} V_0(4) &= \min\left\{4+1+\sum_{y=4}^{5} p_{4y}V_1(y), \frac{34}{5}\right\} \\ &= \min\left\{\frac{49}{5}, \frac{34}{5}\right\} = \frac{34}{5}, \end{aligned}$$

$$\begin{aligned} V_0(5) &= \min\left\{5+1+\sum_{y=5}^{5} p_{5y}V_1(y), \frac{34}{5}\right\} \\ &= \min\left\{\frac{54}{5}, \frac{34}{5}\right\} = \frac{34}{5}. \end{aligned}$$

Así,

$$V_0(x) = \begin{cases} \frac{291}{50}, & \text{si } x = 1, \\ \frac{333}{50}, & \text{si } x = 2, \\ \frac{34}{5}, & \text{si } x = 3, 4, 5. \end{cases}$$

Se concluye que en los dos primeros estados es mejor dejarla trabajar y en los siguientes reparar la máquina.

Ahora, considere $n = 350$; $N = 60$; $R = 26.58$; y $g(x) = 8.16x + 4.27$. Note que se cuenta con una cantidad de estados grande por lo tanto resolver este problema manualmente sería muy laborioso, por lo cual se elaboró un algoritmo en Matlab (véase Apéndice B) que determina a la función de valor óptimo, así se tiene que la función de valor es:

$$V_0(x) = \begin{cases} 1538.10 & \text{si } x = 1, \\ 1543 & \text{si } x = 2, \ldots, 350. \end{cases},$$

de donde se puede observar que es a partir del nivel 2 donde se tiene que reemplazar la máquina. En la siguiente gráfica, figura 1, se representa la función de valor y se aprecia como a partir del estado 2 los costos de operación se mantienen constantes,

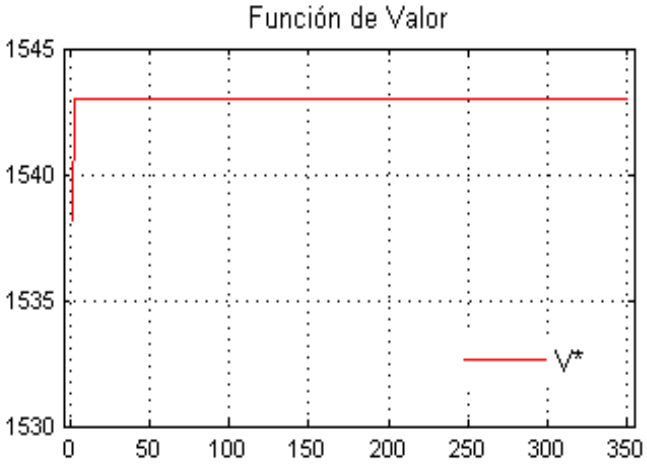


Figura 1. Función de Valor

2.2. Inventarios

Considere un almacén donde cada determinado periodo de tiempo el administrador lleva a cabo un inventario para determinar la cantidad de producto almacenado. Basado en ésta información, él decide ordenar o no cierta cantidad de producto adicional a un proveedor. Al hacer ésto se enfrenta al costo asociado con guardar el producto o las ganancias perdidas por no ser capaz de cubrir la demanda del cliente. Así, el objetivo del administrador es maximizar el beneficio obtenido. Se supondrá que la demanda del producto es aleatoria con distribución de probabilidad conocida. A continuación se darán una serie de condiciones para formular el modelo.

Condición 2.2.1 a) *La decisión de ordenar pedido adicional es hecha al inicio del periodo y se entrega inmediatamente.*

b) *Las demandas de producto se reciben todo el periodo de tiempo pero son cumplidas en el último instante de tiempo del periodo.*

c) *Si la demanda excede al inventario el cliente acude a otra parte por el producto faltante, es decir, no hay pedidos pendientes.*

d) *Los ingresos, costos y la distribución de la demanda no varían con el periodo.*

e) *El producto únicamente es vendido en unidades enteras.*

f) *El almacén tiene capacidad para Z unidades.*

Sea x_t la cantidad de inventario al comienzo del periodo t, a_t el número de unidades ordenadas por el administrador en el tiempo t y, sea D_t la demanda aleatoria en el periodo t. Suponga que la probabilidad de la demanda está dada por $p_y = P(D_t = y)$, $y = 0, 1, 2, \ldots$. El inventario al tiempo de decisión $t+1$, x_{t+1} está relacionado con el inventario en el periodo t, x_t a través del sistema de ecuaciones

$$\begin{aligned} x_{t+1} &= \max\{x_t + a_t - D_t, 0\} \\ &\equiv [x_t + a_t - D_t]^+ . \end{aligned}$$

Debido a que no es permitido pedidos pendientes el nivel del inventario no puede ser negativo. Así, si $x_t + a_t - D_t < 0$, el nivel del inventario en el siguiente periodo de decisión será 0.

A continuación se describen las funciones de valor involucradas en el modelo. El valor del costo por ordenar x unidades en cualquier periodo es $L(x)$ y, se supondrá que está compuesto por un costo $K > 0$ fijo por realizar el pedido y un costo variable $c(x)$ que crece con la cantidad ordenada. Así,

$$L(x) = \begin{cases} K + c(x) & \text{si } x > 0 \\ 0 & \text{si } x = 0 \end{cases} .$$

El valor del costo por mantener un inventario de x unidades por periodo es representado por una función $h(x)$ no decreciente. Finalmente, si la demanda es de

y unidades y el inventario es suficiente para cubrirla entonces, el administrador recibe un ingreso $f(y)$. Se supondrá que $f(0) = 0$.

En este modelo la recompensa depende del estado del sistema y del siguiente periodo de decisión; es decir,

$$r_t(x_t, a_t, x_{t+1}) = -L(a_t) - h(x_t + a_t) + f(x_t + a_t - x_{t+1}).$$

Sin embargo es más conveniente trabajar con $r_t(x_t, a_t)$. Para este fin se calculará el valor esperado (al inicio del periodo t), $F_t(x)$, de los ingresos recibidos en el periodo t cuando el inventario previo al ingreso del pedido del cliente es de x unidades. Éste es obtenido de la siguiente manera, si el inventario x excede a la demanda y, el valor del ingreso es $f(j)$. Esto ocurre con probabilidad p_y. En caso contrario, si la demanda excede al inventario, el valor del ingreso es $f(x)$; ésto ocurre con probabilidad $q_x = \sum_{y=x}^{\infty} p_y$. Entonces

$$F(x) = \sum_{y=0}^{x-1} f(y)\, p_y + f(x)\, q_x.$$

A continuación se da la formulación del MCM,

1. Épocas de decisión:

$$T = \{1, 2, \ldots, N\}, \ N < \infty.$$

2. Estados: (la cantidad de inventario al inicio del periodo t)

$$X = \{0, 1, 2, \ldots, Z\}.$$

3. Acciones (la cantidad de producto que puede ordenarse)

$$A = \{0, 1, \ldots, Z\}.$$

4. Acciones admisibles: (la cantidad de producto ordenado en el periodo t)

$$A(x) = \{0, 1, \ldots, Z - x\}.$$

5. Recompensa: (ingreso esperado menos los costos de pedido y mantenimiento)

$$r_t(x, a) = F(x + a) - L(a) - h(x + a),$$

$t = 1, 2, \ldots, N - 1$.

$$r_N(x) = l(x), \ t = N.$$

6. Probabilidades de transición:

$$Q_t(y \,|x, a) = \begin{cases} 0 & \text{si } Z \geq y > x + a \\ p_{x+a-y} & \text{si } Z \geq x + a \geq y, \ y > 0 \\ q_{x+a} & \text{si } Z \geq x + a \text{ y } y = 0 \end{cases},$$

donde q_{x+a} es como se definió anteriormente.

La manera en que se determinó esta probabilidad de transición es la siguiente: si el inventario en existencia al inicio del periodo t es de x unidades y se ordenan a unidades entonces el inventario previo a la demanda externa es de $x + a$ unidades (Suposición 2.2.1 a)). Un nivel de inventario de $y > 0$ al comienzo del periodo $t + 1$ requiere de una demanda de $x + a - y$ unidades en el periodo t, lo cual ocurre con probabilidad p_{x+a-y}. Ya que no se permiten pedidos pendientes (Suposición 2.2.1 c)), si la demanda en el periodo t excede $x + a$ unidades, entonces el inventario al inicio del periodo $t + 1$ es de 0 unidades, lo cual ocurre con probabilidad q_{x+a}. La probabilidad de que el nivel del inventario exceda $x + a$ unidades es 0 ya que, la demanda es no negativa. La Suposición 2.2.1 f) restringe al nivel del inventario a ser siempre menor o igual que Z.

Las reglas de decisión determinan la cantidad de producto a ser ordenado en cada periodo para cada posible posición del nivel de inventario. Una política será una sucesión de tales reglas. Un ejemplo de tal política es el siguiente,

Ordenar suficiente producto para aumentar el nivel del inventario hasta una cierta cantidad de λ unidades, siempre que el nivel del inventario al inicio del periodo sea menor a α unidades. Cuando el nivel del inventario al inicio del periodo sea mayor o igual a α unidades, entonces no se ordena nada. Esta regla de decisión puede ser representada por

$$d_t(x) = \begin{cases} \lambda - \alpha & \text{si } x < \alpha \\ 0 & \text{si } x \geq \alpha \end{cases} .$$

2.2.1. Ejemplo Numérico

A continuación se resolverá este problema con valores numéricos. Sea $K = 6$, $c(x) = 3x$, $l(x) = 0$, $h(x) = 2x$, $Z = 3$, $N = 4$, $f(x) = 10x$, y

$$p_y = \begin{cases} \frac{1}{3} & \text{si } y = 1, 2 \\ \frac{1}{6} & \text{si } y = 0, 3 \end{cases} .$$

Este modelo puede ser interpretado como sigue, el nivel de inventario está limitado a 3 o menos unidades. Todos los costos e ingresos son lineales. El ingreso esperado cuando se tienen x unidades en existencia antes de recibir una orden está dado en tabla 2.1, dada a continuación:

x	$F(x)$
0	0
1	$\frac{25}{3}$
2	$\frac{40}{3}$
3	15

(2.1)

Combinando el ingreso esperado con los costos por pedido y mantenimiento se obtiene la recompensa esperada en el periodo t si el nivel de inventario es x y son ordenadas a unidades. Así, los valores de $r_t(x, a)$ están dados a continuación,

en la tabla 2.2

x/a	0	1	2	3
0	0	$-\frac{8}{3}$	$-\frac{8}{3}$	-6
1	$\frac{19}{3}$	$-\frac{1}{3}$	3	$\times$
2	$\frac{28}{3}$	0	$\times$	$\times$
3	9	$\times$	$\times$	$\times$

(2.2)

donde $\times$ representa una acción no admisible. La matriz de transición, $Q_t\left(y\left|x,a\right.\right)$, está dada por

$x+a/y$	0	1	2	3
0	1	0	0	0
1	$\frac{5}{6}$	$\frac{1}{6}$	0	0
2	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{6}$	0
3	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

(2.3)

Aplicando la técnica de programación dinámica, la ecuación de programación dinámica está dada por,

$$\begin{aligned} V_N\left(x\right) &= 0, \\ V_t\left(x\right) &= \max_{a\in A(x)}\left\{r\left(x,a\right)+\sum_{y\in X}Q\left(y\left|x,a\right.\right)V_{t+1}\left(y\right)\right\}, \end{aligned}$$

$x\in X$ y $t=N-1,\ldots,1$. Note que, debido a que tanto la recompensa como la probabilidad de transición son homogéneas, es decir, no varían con el tiempo no aparece el índice t.

Implementando el algoritmo de inducción hacia atrás, se tiene, que para $t=4$,

$$V_4\left(x\right)=r_4\left(x\right)=0,$$

$x=0,1,2,3$. Ahora, haciendo $t=3$,

$$\begin{aligned} V_3\left(x\right) &= \max_{a\in A(x)}\left\{r\left(x,a\right)+\sum_{y\in X}Q\left(y\left|x,a\right.\right)V_4\left(y\right)\right\} \\ &= \max_{a\in A(x)}\left\{r\left(x,a\right)\right\} \end{aligned}$$

de donde por los valores obtenidos para $r\left(x,a\right)$ dados en la tabla 2.2, se tiene que

x	$a=0$	$a=1$	$a=2$	$a=3$	$V_3\left(x\right)$	$A\left(x\right)_3^*$
0	0	$-\frac{8}{3}$	$-\frac{8}{3}$	-6	0	0
1	$\frac{19}{3}$	$\frac{1}{3}$	-3	$\times$	$\frac{19}{3}$	0
2	$\frac{28}{3}$	0	$\times$	$\times$	$\frac{28}{3}$	0
3	9	$\times$	$\times$	$\times$	9	0

.

Para la siguiente etapa, es decir, cuando $t = 2$,

$$V_2(x) = \max_{a \in A(x)} \left\{ r(x,a) + \sum_{y \in X} Q(y \,|x,a)\, V_3(y) \right\}.$$

Al resolver la ecuación anterior, para cada x, se tiene

$$\begin{aligned} V_2(0) &= \max_{a \in A(0)} \left\{ r(0,a) + \sum_{y \in X} Q(y \,|0,a)\, V_3(y) \right\} \\ &= \max\left\{0, -\frac{29}{18}, 1, \frac{13}{18}\right\} = 1. \end{aligned}$$

$$\begin{aligned} V_2(1) &= \max_{a \in A(1)} \left\{ r(1,a) + \sum_{y \in X} Q(y \,|1,a)\, V_3(y) \right\} \\ &= \max\left\{\frac{133}{18}, 4, \frac{67}{18}\right\} = \frac{133}{18}. \end{aligned}$$

$$\begin{aligned} V_2(2) &= \max_{a \in A(2)} \left\{ r(2,a) + \sum_{y \in X} Q(y \,|2,a)\, V_3(y) \right\} \\ &= \max\left\{13, \frac{121}{18}\right\} = 13. \end{aligned}$$

$$\begin{aligned} V_2(3) &= \max_{a \in A(3)} \left\{ r(3,a) + \sum_{y \in X} Q(y \,|3,a)\, V_3(y) \right\} \\ &= r(3,0) + \sum_{y \in X} Q(y \,|3,0)\, V_3(y) = \frac{283}{18}. \end{aligned}$$

A continuación se resume está información,

x	$a=0$	$a=1$	$a=2$	$a=3$	$V_2(x)$	$A(x)_2^*$
0	0	$-\frac{29}{18}$	1	$\frac{13}{18}$	1	2
1	$\frac{133}{18}$	4	$\frac{67}{18}$	$\times$	$\frac{133}{18}$	0
2	13	$\frac{121}{18}$	$\times$	$\times$	13	0
3	$\frac{283}{18}$	$\times$	$\times$	$\times$	$\frac{283}{18}$	0

.

Ahora, para $t = 1$,

$$V_1(x) = \max_{x \in A(x)} \left\{ r(x,a) + \sum_{y \in X} Q(y \,|x,a)\, V_2(y) \right\}.$$

Resolviendo la ecuación anterior, para cada x, se tiene

$$\begin{aligned} V_1(0) &= \max_{a\in A(0)}\left\{ r(0,a) + \sum_{y\in X} Q(y\,|0,a)\, V_2(y) \right\} \\ &= \max\left\{1, -\frac{65}{108}, \frac{133}{54}, \frac{43}{12}\right\} = \frac{43}{12}. \end{aligned}$$

$$\begin{aligned} V_1(1) &= \max_{a\in A(1)}\left\{ r(1,a) + \sum_{y\in X} Q(y\,|1,a)\, V_2(y) \right\} \\ &= \max\left\{\frac{907}{108}, \frac{295}{54}, \frac{79}{12}\right\} = \frac{907}{108}. \end{aligned}$$

$$\begin{aligned} V_1(2) &= \max_{a\in A(2)}\left\{ r(2,a) + \sum_{y\in X} Q(y\,|2,a)\, V_2(y) \right\} \\ &= \max\left\{\frac{781}{54}, \frac{115}{12}\right\} = \frac{781}{54}. \end{aligned}$$

$$\begin{aligned} V_1(3) &= \max_{a\in A(3)}\left\{ r(3,a) + \sum_{y\in X} Q(y\,|3,a)\, V_2(y) \right\} \\ &= r(3,0) + \sum_{y\in X} Q(j\,|3,0)\, V_2(y) = \frac{223}{12}. \end{aligned}$$

Es decir,

x	$a=0$	$a=1$	$a=2$	$a=3$	$V_1(x)$	$A(x)_1^*$
0	1	$-\frac{65}{108}$	$\frac{133}{54}$	$\frac{43}{12}$	$\frac{43}{12}$	3
1	$\frac{907}{108}$	$\frac{295}{54}$	$\frac{79}{12}$	$\times$	$\frac{907}{108}$	0
2	$\frac{781}{54}$	$\frac{115}{12}$	$\times$	$\times$	$\frac{781}{54}$	0
3	$\frac{223}{12}$	$\times$	$\times$	$\times$	$\frac{223}{12}$	0

En la siguiente tabla se presenta tanto a la función de valor óptimo como a la política óptima

x	$d_1(x)$	$d_1(x)$	$d_1(x)$	$V^*(x)$
0	3	2	0	$\frac{43}{12}$
1	0	0	0	$\frac{907}{108}$
2	0	0	0	$\frac{781}{54}$
3	0	0	0	$\frac{223}{12}$

donde esta información puede ser interpretada como sigue; si al inicio del periodo 1 el inventario es de 0 unidades, entonces se piden 3 unidades de otra manera no

se ordena nada; si en el periodo 2 el inventario es de nivel 2 entonces se ordenan unidades en caso contrario no se ordenada nada y, finalmente en el periodo 3 no se realizan pedidos sin importar el nivel del inventario.

Ahora, considere $K = 19$, $c(x) = 4x$, $g(x) = 0$, $h(x) = 6x$, $Z = 25$, $N = 15$, $f(x) = 17x$, entonces utilizando el algoritmo creado en Matlab (véase Apéndice B) se tiene, que la función de valor óptimo está dada por

x	$V^*(x)$	x	$V^*(x)$	x	$V^*(x)$	x	$V^*(x)$
0	517.8813	7	545.8813	14	592.5488	21	611.6711
1	521.8813	8	553.7006	15	596.8813	22	612.1819
2	525.8813	9	561.1272	16	600.7015	23	612.3891
3	529.8813	10	568.5386	17	603.7991	24	611.3897
4	533.8813	11	575.5625	18	606.5501	25	609.6240
5	537.8813	12	581.8560	19	608.6380		
6	541.8813	13	587.3428	20	610.3504		

en la siguiente gráfica se representa esta información,

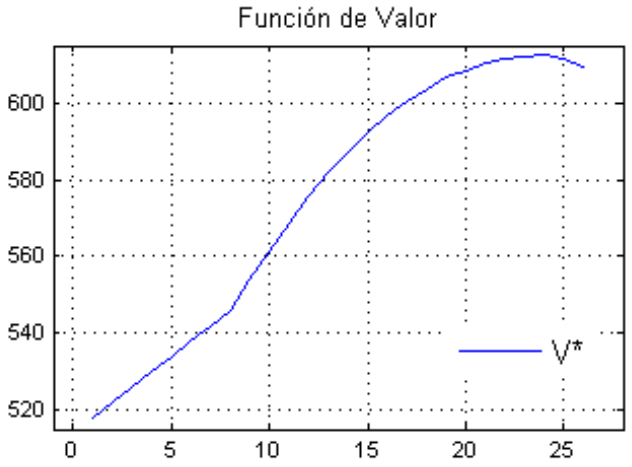


Figura 2. Función de Valor

Capítulo 3

Técnica de Aleatorización en Programación Dinámica

Una desventaja de la técnica de programación dinámica surge cuando la cantidad de estados (o controles) es grande, para lo cual es necesario la implementación de algoritmos computacionales para su solución por lo que podría surgir lo conocido en la literatura como *la Maldición de la Dimensionalidad* (*The Curse of Dimensionality*, véase [2], [6] y [9]). La Maldición de la Dimensionalidad plantea que el número de operaciones necesarias para resolver un problema crece exponencialmente con la dimensión del espacio de estados (o controles).

Por ejemplo, para el problema de reemplazamiento de máquinas que se resolvió anteriormente se tiene que la cantidad de operaciones necesarias para resolver el problema es de $2N\,|X|$ y, para el problema de inventario es de $N\,|X \times A|$, donde $|X|$ denota la cantidad de elementos de X. De esto se tiene que si la cardinalidad de X o la de A es grande llevaría a la implementación de algoritmos computacionales donde podría surgir el problema de la dimensionalidad. Así, en este capítulo se describirá un método que permite evitar esta problemática, el método que se estudió en esta tesis es conocido como aleatorización, el cual consiste en aproximar a la función de valor mediante la simulación de muestras aleatorias, en este caso, del espacio de estados y evaluando dichas muestra en una ecuación de programación dinámica aproximada, la cual será descrita más adelante (véase ecuación 3.17), para determinar una función de valor aproximada. Finalmente, se repetirá este procedimiento una cantidad determinada de veces y se procederá a promediar todas las funciones de valor obtenidas, y el resultado de dicho promedio será la aproximación a la función de valor verdadera que se busca obtener. La base principal de este algoritmo son funciones que permiten aproximar a la función de valor óptimo mediante un método Monte Carlo (véase [8]), el cual será descrito más adelante.

En las secciones siguientes se introducirán definiciones y notación usadas a lo largo de este capítulo, se describirá el método Monte Carlo y se darán

algunos ejemplos de dicho método para finalmente introducir el algoritmo de aleatorización. Finalmente se discutirán algunos ejemplos de dicha técnica.

3.1. Resultados Auxiliares

A continuación se darán definiciones y teoremas necesarios para este capítulo.

Definición 3.1.1 *a) Una sucesión $\{X_n\}$ de variables aleatorias converge en distribución o en ley a la variable aleatoria X, con función de distribución F, si*

$$\lim_{n\to\infty} F_n(x) = F(x)$$

para toda x en la que F es continua, donde F_n representa a la función de distribución de X_n. Se denotará la convergencia en distribución por $\xrightarrow{d}$.

b) Una sucesión $\{X_n\}$ de variables aleatorias converge en probabilidad a la variable aleatoria X, si para cada $\varepsilon > 0$,

$$P(|X_n - X| > \varepsilon) \to 0$$

cuando $n \to \infty$. Se denotará la convergencia en probabilidad por $\xrightarrow{p}$.

c) Una sucesión $\{X_n\}$ de variables aleatorias converge casi seguramente a la variable aleatoria X, si

$$P(\{\omega \,|X_n(\omega) \to X(\omega)\}) = 1$$

cuando $n \to \infty$. Se denotará la convergencia casi seguramente por $\xrightarrow{c.s.}$

Definición 3.1.2 *La media muestral, $\bar{X}$, es el promedio aritmético de los valores en una muestra aleatoria $(X_1, X_2, \ldots, X_n)$ y es definida por,*

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i.$$

El siguiente teorema da algunas características del estadístico $\bar{X}$, las cuales son conocidas en la teoría de probabilidad.

Teorema 3.1.3

a) *Sea $\{X_n\}$ una sucesión de variables aleatorias independientes e idénticamente distribuidas con media μ y varianza finita σ^2. Entonces*

$$\lim_{n\to\infty} P\left(\frac{X_1 + \ldots + X_n - n\mu}{\sigma\sqrt{n}} \leq x\right) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-\frac{y^2}{2}}\, dy.$$

b) *Sea $\{X_n\}$ una sucesión de variables aleatorias independientes e idénticamente distribuidas, de varianza finita. Entonces*

$$\bar{X} \xrightarrow{p} \mu,$$

donde μ es la esperanza común de las variables aleatorias X_1, $X_2, \ldots$.

c) *Sea $\{X_n\}$ una sucesión de variables aleatorias independientes e idénticamente distribuidas, de varianza finita. Entonces*

$$\bar{X} \xrightarrow{c.s.} \mu,$$

donde μ es la esperanza común de las variables aleatorias X_1, $X_2, \ldots$.

Definición 3.1.4 *Se dirá que X_n está acotada en probabilidad, (y se denotará por $X_n = O_p(1)$) si para todo $\varepsilon > 0$ existe una constante R para la cual, $P(|X_n| \geq R) < \varepsilon$ para toda $n \geq 1$. Para una sucesión $\{a_n\}$ se escribirá $X_n = O_p(a_n)$, si $\frac{X_n}{a_n} = O_p(1)$.*

3.2. Método Monte Carlo

En esta sección se revisará una técnica para la simulación de números aleatorios, conocida como el Método Monte Carlo. Como ya se mencionó en la introducción, la base del método es un generador de números aleatorios. El procedimiento que utiliza Monte Carlo es el siguiente: generar una serie de números aleatorios, $x_1, x_2, \ldots x_n$, independientes e idénticamente distribuidos. Usar esta sucesión para producir otra sucesión, $u_1, u_2, \ldots u_n$, distribuida de acuerdo a la función de distribución de probabilidad en la que se está interesado.

A continuación se describe este método.

Algoritmo 3.2.1 *Sea $(S, f, \mu, \mathcal{U}, g)$ una quíntupla donde,*

1. *S es un conjunto finito de estados, es decir, los valores de las variables que se desean simular;*
2. *f es una función de S a S y será la función de transición sobre S;*
3. *μ es la distribución sobre S;*
4. *$\mathcal{U}$ será el espacio de salida, es decir, los valores que toman las simulaciones generadas;*
5. *g es una función de S a $\mathcal{U}$, la cual determina como serán generados los números aleatorios;*

Así, el algoritmo tiene la siguiente estructura:

a. *Inicio: Se genera x_0 con distribución μ. Se hace $t = 1$.*

b. Transición: Sea $x_t = f(x_{t-1})$.

c. Salida: $u_t = g(x_t)$.

d. Repetir: Sea $t = t + 1$ *y regresar al paso b.*

Observación 3.2.2 *Para los algoritmos computacionales que se desarrollaron se utilizó el programa de Matlab en el cual la función que genera los números aleatorios es* randint.

La manera en la que será utilizado el método Monte Carlo en el algoritmo de aleatorización es en la simulación de la muestra aleatoria de estados para así obtener funciones de valor aproximadas y una vez que se ha repetido el algoritmo una cantidad determinada de veces promediar estas funciones para obtener la aproximación final deseada.

3.2.1. Simulación Monte Carlo

A continuación se procederá a aplicar el Algoritmo 3.2.1 para ejemplificar el método Monte Carlo a los ejemplos de reemplazamiento de máquinas, problema de inventario y finalmente al problema Lineal Cuadrático (LQ), bajo el supuesto de que se conocen las políticas óptimas.

Ejemplo 3.2.3 *Considere el problema de reemplazamiento de máquinas descrito en la Sección 2.1. Sea* $n = 200$, $N = 25$, $R = 15$, $g(x) = x + 1$ *y* $M = 100$. *A continuación se describirá como simular una muestra aleatoria de estados a partir del Algoritmo 3.2.1. Identificando las componentes de la quíntupla se tiene,*

1. $S = \{1, 2, \ldots, 200\}$;
2. $f = p_{xy}$, *la ley de transición para el ejemplo;*
3. μ *una distribución uniforme en* S;
4. $\mathcal{U} = S = \{1, 2, \ldots, 200\}$;
5. *Para generar una variable aleatoria* X *con distribución* g, *el procedimiento a seguir es el siguiente:*

   a) *Generar una variable aleatoria* $V \sim Unif(0, 1)$,
   b) *Sea* $\hat{X} = g^{-1}(V)$,

   Se probará que $\hat{X}$ *tiene distribución* g,

$$\begin{aligned} g_{\hat{X}}(x) &= P\left(\hat{X} \leq x\right) \\ &= P\left(g^{-1}(v) \leq x\right) \\ &= P\left(g\left(g^{-1}(v)\right) \leq g(x)\right) \\ &= P(V \leq g(x)) \\ &= \int_0^{g(x)} dv = g(v). \end{aligned}$$

Es decir,

$$X \stackrel{d}{=} \hat{X}.$$

Así, el algoritmo, estará dado por

a) Inicio: Sea x_0 *con distribución* μ*. Se hace* $t = 1$*.*

b) Transición: Sea $x_t = f(x_{t-1}) = p_{xy}$*, donde* $x_{t-1} = x$ *y* $x_t = y$*.*

c) Salida: $u_1 = g(x_t)$*.*

d) Repetir: Sea $t = t + 1$ *y regresar al paso b.*

La siguiente gráfica muestra las trayectorias óptimas obtenidas para cuatro simulaciones del algoritmo anterior

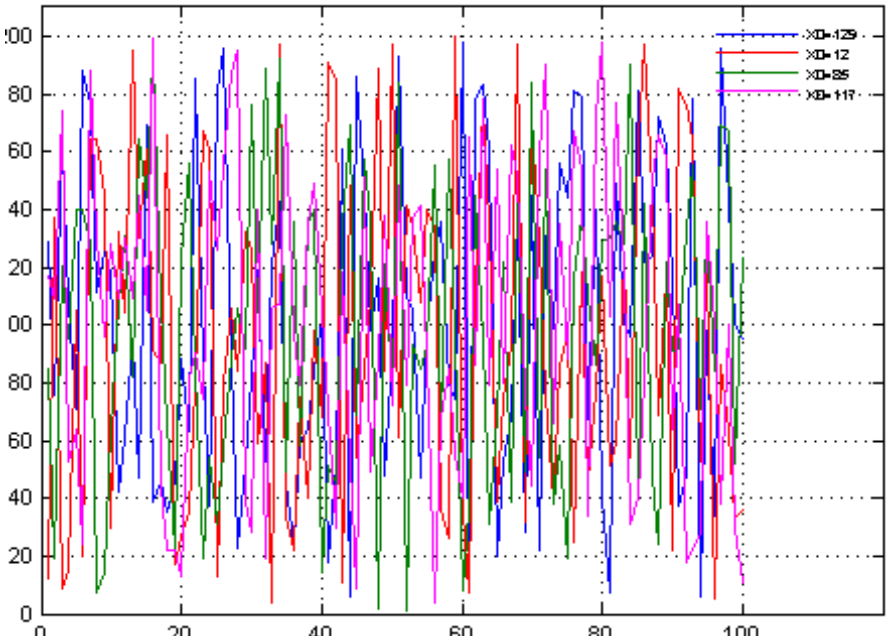


Figura 3. Simulación Monte Carlo

Ejemplo 3.2.4 *Ahora considere el problema de inventario de la Sección 2.2, de manera similar al ejemplo anterior, con un nivel máximo de inventario de* 150 *y un tamaño de muestra de* $M = 75$*, el algoritmo de simulación Monte Carlo estaría dado de la siguiente manera, primero identificando las componentes de la quíntupla se tiene,*

1. $S = \{0, 1, \ldots, 150\}$;
2. $f = Q(y|x, a)$*, la ley de transición para el ejemplo;*

3. μ *una distribución uniforme en* S*;*

4. $\mathcal{U} = S = \{0, 1, \ldots, 150\}$;

5. *La distribución* g *es de la misma forma como se da en el inciso 5 del Ejemplo 3.2.3.*

Así, el algoritmo, estará dado por Inicio: Sea x_0 *con distribución* μ*. Se hace* $t = 1$*. Transición: Sea* $x_t = f(x_{t-1}) = Q(y|x, a)$*, donde* $x_{t-1} = x$ *y* $x_t = y$*. Salida:* $u_1 = g(x_t)$*. Repetir: Sea* $t = t + 1$ *y regresar al paso 2.*

La siguiente gráfica representa cuatro simulaciones de trayectoria óptima

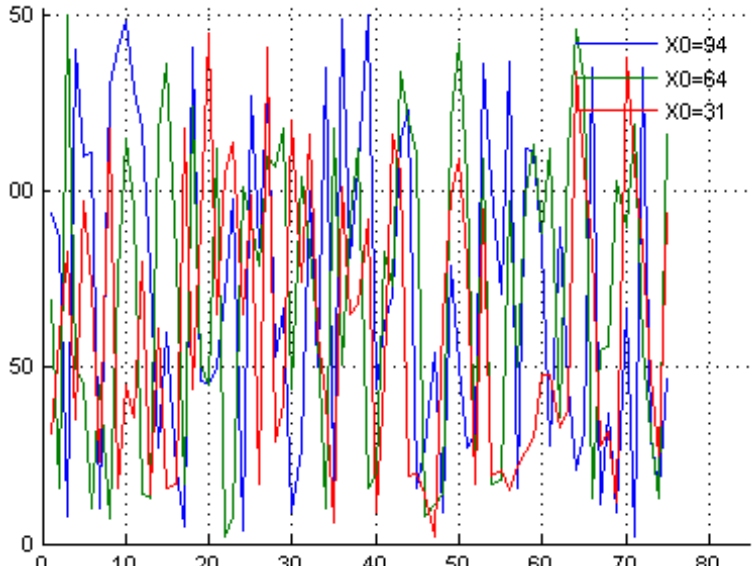


Figura 4. Simulación Monte Carlo

Ejemplo 3.2.5 *Los problemas Lineal Cuadrático (LQ) consisten de un sistema lineal con un costo cuadrático y es uno de los problemas de control más usados en ingeniería, economía y muchos otros campos. Considere un sistema definido por la siguiente ecuación en diferencias*

$$x_{t+1} = \gamma x_t + \beta a_t + \xi_t \tag{3.1}$$

para $t = 0, 1, \ldots$*. Donde* γ *y* β *son constantes reales tales que* $\gamma\beta \neq 0$*. La función de costo está dada por*

$$c(x, a) = qx^2 + ra^2,$$

donde q *y* r *son constantes reales tales que* $q > 0$ *y* $r > 0$*.* $\{\xi_t\}$ *es una sucesión de variables aleatorias i.i.d tomando valores en* $S = \mathbb{R}$*, con función de densidad*

continua Δ, *independientes del estado inicial* x_0, *con media* 0 *y varianza finita,* σ^2, *es decir,*

$$\begin{aligned} E(\xi) &= 0, \\ E(\xi^2) &= \sigma^2 < +\infty. \end{aligned}$$

Sea $X = A = A(x) = \mathbb{R}$. *Bajo las suposiciones anteriores deseamos encontrar una política que minimice el criterio de rendimiento*

$$\begin{aligned} V(\pi, x) &= E_x^{\pi}\left[\sum_{t=0}^{\infty} \alpha^t c(x_t, a_t)\right] \\ &= E_x^{\pi}\left[\sum_{t=0}^{\infty} \alpha^t \left(q x_t^2 + r a_t^2\right)\right], \end{aligned} \tag{3.2}$$

para $\pi \in \Pi$ *y* $x \in X$.

Se tiene que la EPD está dado por,

$$v_0(x) = q_N x^2 = 0 \tag{3.3}$$

$$\begin{aligned} v_n(x) &= \min_{a \in A(x)} \{c(x,a) + \alpha E[v_{n-1}(\gamma x_t + \beta a_t + \xi_t)]\} \\ &= \min_{a \in A(x)} \left\{q x^2 + r a^2 + \alpha E[v_{n-1}(\gamma x + \beta a + \xi)]\right\}, \end{aligned} \tag{3.4}$$

para toda $x \in X$ *y* $n = 1, 2, \ldots$. *Hallando el mínimo de las ecuaciones anteriores, se tiene que, en general*

$$v_n(x) = K_n x^2 + C_n$$

$x \in X$, *con la constante* K_n *dada recursivamente por*

$$\begin{aligned} K_n &= \frac{q\left(r + \alpha\beta^2 K_{n-1}\right) + r\alpha K_{n-1}\gamma^2}{r + \beta^2 \alpha K_{n-1}} \\ &= \frac{qr + K_{n-1}\left(q\alpha\beta^2 + r\alpha\gamma^2\right)}{r + \beta^2 \alpha K_{n-1}} \\ &= \frac{P + K_{n-1}Q}{R + K_{n-1}S}, \end{aligned} \tag{3.5}$$

$n \geq 1$, *y*

$$C_n = \alpha\sigma^2 \sum_{t=1}^{n-1} \alpha^{n-t-1} K_t.$$

Así, la función de valor y política óptima para el problema LQ están dadas por, (véase [7])

$$V^*(x) = K x^2 + C$$

y

$$f^*(x) = \frac{-\gamma\alpha\beta K}{r + \alpha\beta^2 K} x,$$

$x \in X$, *donde*

$$C = \frac{\alpha\sigma^2 K}{1-\alpha}$$

y

$$K = \lim_{n\to\infty} K_n,$$

respectivamente.

Usando el método Monte Carlo, la trayectoria obtenida para $n = 120$ *y* $M = 50$, *es representada en la siguiente gráfica,*

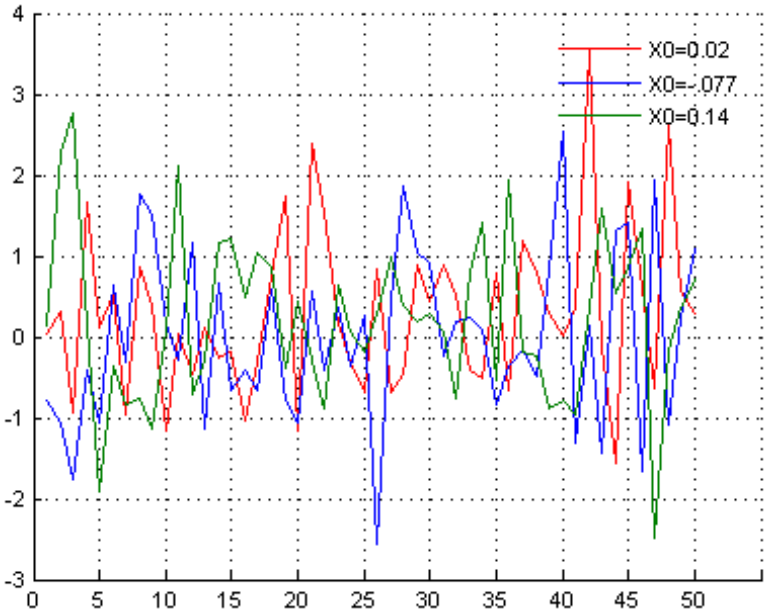


Figura 5. Simulación Monte Carlo

3.3. Método de Aleatorización

Al implementar programación dinámica en problemas de optimización surgen problemas numéricos, en particular, en este trabajo se aborda el problema de la dimensionalidad (Maldición de la Dimensionalidad) [9], [11]. Esta problemática surge en el contexto de control óptimo, cuando la cardinalidad del espacio de estados, acciones y/o el horizonte es "grande". Es decir, al implementar los algoritmos numéricos computacionalmente existen problemas con el tiempo de ejecución y saturación de memoria RAM cuando el número de estados (acciones y/o horizonte) incrementa.

Esta tesis se enfoca al estudio del problema de la dimensionalidad con respecto al espacio de estados. El algoritmo que se implementa consiste en aproximar la función de valor óptimo a partir de un MCM apropiado. Considere un MCM $(X, A, \{A(x) | x \in X\}, Q, c)$ fijo con las siguientes características

Suposición 3.3.1 *1. X y A espacios de Borel finitos.*

2. $A = A(x)$, $x \in X$.

Criterio 3.3.2 *Considere el criterio de rendimiento dado en la Sección 1.5, es decir,*

$$V(\pi, x) = E_x^{\pi}\left[\sum_{t=0}^{\infty} \alpha^t c(x_t, a_t)\right].$$

Para cada $x \in X$ se define

$$V^*(x) = \sup_{\pi \in \Pi} V(\pi, x),$$

V^* se le llama funciones de valores óptimos o valor óptimo. *Una política $\pi^* \in \Pi$, es* óptima, *si*

$$V(\pi^*, x) = \sup_{\pi \in \Pi} V(\pi, x),$$

$x \in X$. Así, el Problema de Control Optimo *consiste en determinar una política que optimice al criterio de rendimiento.*

A partir de este MCM considere el siguiente modelo de control inducido $(S_M, A, \{A(x) | x \in X\}, Q_M, c)$, donde S_M es una muestra aleatoria de X de tamaño M y Q_M está dada por,

$$Q_M(x_k | x, a) = \begin{cases} \frac{Q(x_k|x,a)}{\sum_{j=1}^{M} Q(x_j|x,a)} & \text{si } \sum_{j=1}^{M} Q(x_j | x, a) > 0 \\ 0 & \text{en otro caso.} \end{cases} . \quad (3.6)$$

con $x \in X$.

Criterio 3.3.3 *Ahora, considere el criterio de rendimiento aproximado*

$$\hat{V}(\pi, x) = E_x^{\pi}\left[\sum_{k=1}^{M} \alpha^t c(x_k, a_k)\right].$$

Para cada $x \in X$ se define

$$\hat{V}^*(x) = \sup_{\pi \in \Pi} \hat{V}(\pi, x),$$

$\hat{V}^*$ se le llama función de valor óptimo aproximado. *Una política $\pi^* \in \Pi$, es* óptima, *si*

$$\hat{V}(\pi^*, x) = \sup_{\pi \in \Pi} \hat{V}(\pi, x),$$

$x \in X$. Así, el Problema de Control Optimo Aproximado *consiste en determinar una política que optimice al criterio de rendimiento aproximado.*

Definición 3.3.4 *El **Operador de Bellman**, $T : B(X) \to B(X)$ es una función sobre el espacio de Banach, $B(X)$, de funciones medibles definidas en X (bajo la norma del supremo), definido por*

$$T(W)(x) = \max_{a \in A(x)} \left[c(x,a) + \alpha \sum_{y \in X} W(y)\, Q(y \,|x,a) \right], \tag{3.7}$$

para $x \in X$, $W \in B(X)$ y $\alpha \in (0,1)$.

Definición 3.3.5 *El **Operador Aleatorio de Bellman** $\hat{T}_M : B(X) \to B(X)$, está dado por*

$$\hat{T}_M(W)(x) := \max_{a \in A} \left[c(x,a) + \alpha \sum_{k=1}^{M} W(x_k)\, Q_M(x_k \,|x,a) \right], \tag{3.8}$$

para $x \in X$, $W \in B(X)$, $\alpha \in (0,1)$ y $S_M = \{x_1, \ldots, x_M\}$ es una muestra aleatoria de X y, Q_M está definida por

$$Q_M(x_k \,|x,a) = \frac{Q(x_k \,|x,a)}{\sum_{i=1}^{M} Q(x_i \,|x,a)}.$$

Obsérvese que, $\hat{T}_M(W)$ es una variable aleatoria debido a que depende de $S_M = \{x_1, \ldots, x_M\}$ que es una muestra aleatoria de X, la cual sigue una distribución discreta con media cero y varianza 1.

Lema 3.3.6 *Observe que $\hat{T}_M$ es un operador contracción módulo α en $B(X)$.*

Demostración. Sean $U, W \in B(X)$ y suponga que $\hat{T}_M(W) \geq \hat{T}_M(U)$. Para $x \in X$, sean

$$a_w^* \in \arg\max_{a \in A(x)} \left\{ c(x,a) + \alpha \sum_{k=1}^{M} W(x_k)\, Q_M(x_k \,|x,a) \right\},$$

$$a_u^* \in \arg\max_{a \in A(x)} \left\{ c(x,a) + \alpha \sum_{k=1}^{M} U(x_k)\, Q_M(x_k \,|x,a) \right\}.$$

Entonces

$$\begin{aligned} 0 &\leq \hat{T}_M(W)(x) - \hat{T}_M(U)(x) \\ &= c(x, a_w^*) + \alpha \sum_{k=1}^{M} W(x_k)\, Q_M(x_k \,|x, a_w^*) \\ &\quad - \left\{ c(x, a_u^*) + \alpha \sum_{k=1}^{M} U(x_k)\, Q_M(x_k \,|x, a_u^*) \right\} \end{aligned}$$

$$\begin{aligned}
&\leq c\left(x,a_w^*\right)+\alpha\sum_{k=1}^{M}W\left(x_k\right)Q_M\left(x_k\left|x,a_w^*\right.\right)\\
&\quad-\left\{c\left(x,a_w^*\right)+\alpha\sum_{k=1}^{M}U\left(x_k\right)Q_M\left(x_k\left|x,a_w^*\right.\right)\right\}\\
&= \alpha\sum_{k=1}^{M}\left(W\left(x_k\right)-U\left(x_k\right)\right)Q_M\left(x_k\left|x,a_w^*\right.\right)\\
&\leq \alpha\sum_{k=1}^{M}\left\|W-U\right\|Q_M\left(x_k\left|x,a_w^*\right.\right)\\
&= \alpha\left\|W-U\right\|\sum_{k=1}^{M}Q_M\left(x_k\left|x,a_w^*\right.\right)\\
&= \alpha\left\|W-U\right\|.
\end{aligned}$$

Es decir, si $\hat{T}_M\left(W\right)\left(x\right)\geq\hat{T}_M\left(U\right)\left(x\right)$, entonces $\hat{T}_M\left(W\right)\left(x\right)-\hat{T}_M\left(U\right)\left(x\right)\leq\alpha\left|W\left(x\right)-U\left(x\right)\right|$. De manera similar, si $\hat{T}_M\left(W\right)\leq\hat{T}_M\left(U\right)$ se tiene la desigualdad siguiente $\hat{T}_M\left(U\right)-\hat{T}_M\left(W\right)\leq\alpha\left|U-W\right|$. Esto significa que,

$$\left|\hat{T}_M\left(W\right)\left(x\right)-\hat{T}_M\left(U\right)\left(x\right)\right|\leq\alpha\left|W\left(x\right)-U\left(x\right)\right|.$$

para cada $x\in X$. Esto implica que

$$\begin{aligned}
\sup_{x\in X}\left|\hat{T}_M\left(W\right)\left(x\right)-\hat{T}_M\left(U\right)\left(x\right)\right| &= \left\|\hat{T}_M\left(W\right)-\hat{T}_M\left(U\right)\right\|\\
&\leq \alpha\left\|W-U\right\|.
\end{aligned}$$

Es decir, $\hat{T}_M$ es un operador contracción. ■

Observación 3.3.7 *El Lema 3.3.6 garantiza que existe una única solución para el problema de control dado en el Criterio 3.3.3, para cada muestra fija.*

Lo que se probará es que la diferencia entre $\hat{T}_M$ y T se encuentra acotada en probabilidad.

Sean

$$\begin{aligned}
\hat{T}_{a,M}\left(W\right)\left(x\right) &= c\left(x,a\right)+\alpha\sum_{x_k\in S_M}W\left(x_k\right)Q_M\left(x_k\left|x,a\right.\right), \qquad (3.9)\\
T_a\left(W\right)\left(x\right) &= c\left(x,a\right)+\alpha\sum_{y\in X}W\left(y\right)Q\left(y\left|x,a\right.\right).
\end{aligned}$$

para $x\in X$ y $W\in B\left(X\right)$. Entonces, los operadores $\hat{T}_M$ y T pueden ser vistos en función de $\hat{T}_{a,M}$ y T_a en el sentido siguiente,

$$\begin{aligned}
\hat{T}_M\left(W\right)\left(x\right) &= \max_{a\in A}\hat{T}_{a,M}\left(W\right)\left(x\right), \qquad (3.10)\\
T\left(W\right)\left(x\right) &= \max_{a\in A}T_a\left(W\right)\left(x\right),
\end{aligned}$$

$x\in X$, $W\in B\left(X\right)$.

Lema 3.3.8 *Para todo $M \geq 1$ se tiene que*

$$\left\|\hat{T}_M\left(\hat{V}\right) - T\left(V\right)\right\| \leq \sum_{a\in A}\left\|\hat{T}_{a,M}\left(V\right) - T_a\left(V\right)\right\|,$$

donde V y $\hat{V}$ son las funciones de valor óptimo de los problemas de control dados en los Criterios 3.3.2 y 3.3.3, respectivamente.

Demostración. Sea $x \in X$ fijo. Se definen las reglas de decisión f y f_M por

$$\begin{aligned} f\left(x\right) &= \arg\max_{a\in A} T_a\left(V\right)\left(x\right), \\ f_M\left(x\right) &= \arg\max_{a\in A} \hat{T}_{a,M}\left(\hat{V}\right)\left(x\right). \end{aligned} \tag{3.11}$$

De donde,

$$\begin{aligned} T\left(V\right)\left(x\right) &= T_{f(x)}\left(V\right)\left(x\right) \geq T_{f_M(x)}\left(V\right)\left(x\right), \\ \hat{T}_M\left(\hat{V}\right)\left(x\right) &= \hat{T}_{f_M(x),M}\left(\hat{V}\right)\left(x\right) \geq \hat{T}_{f(x),M}\left(\hat{V}\right)\left(x\right). \end{aligned} \tag{3.12}$$

Primero, suponga que $\hat{T}_M\left(\hat{V}\right)\left(x\right) \geq T\left(V\right)\left(x\right)$. Entonces, se tienen las desigualdades siguientes:

$$\hat{T}_{f_M(x),M}\left(\hat{V}\right)\left(x\right) = \hat{T}_M\left(\hat{V}\right)\left(x\right) \geq T\left(V\right)\left(x\right)$$

pero por (3.12),

$$T\left(V\right)\left(x\right) = T_{f(x)}\left(V\right)\left(x\right).$$

Se sigue que

$$\begin{aligned} 0 &\leq \hat{T}_M\left(\hat{V}\right)\left(x\right) - T\left(V\right)\left(x\right) \leq \hat{T}_{f_M(x),M}\left(\hat{V}\right)\left(x\right) - T_{f(x)}\left(V\right)\left(x\right) \\ &\leq \max_{a\in A}\left|\hat{T}_{a,M}\left(\hat{V}\right)\left(x\right) - T_a\left(V\right)\left(x\right)\right|. \end{aligned} \tag{3.13}$$

Usando un argumento similar cuando $T\left(V\right)\left(x\right) \geq \hat{T}_M\left(\hat{V}\right)\left(x\right)$ se obtiene la siguiente desigualdad:

$$\begin{aligned} 0 &\leq T\left(V\right)\left(x\right) - \hat{T}_M\left(\hat{V}\right)\left(x\right) \leq T_{f(x)}\left(V\right)\left(x\right) - \hat{T}_{f_M(x),M}\left(\hat{V}\right)\left(x\right) \\ &\leq \max_{a\in A}\left|T_a\left(V\right)\left(x\right) - \hat{T}_{a,M}\left(\hat{V}\right)\left(x\right)\right| \end{aligned} \tag{3.14}$$

En cualquier caso, se tiene,

$$\left|\hat{T}_M\left(\hat{V}\right)\left(x\right) - T\left(V\right)\left(x\right)\right| \leq \max_{a\in A}\left|\hat{T}_{a,M}\left(\hat{V}\right)\left(x\right) - T_a\left(V\right)\left(x\right)\right|. \tag{3.15}$$

Usando (3.15) se llega a que

$$\begin{aligned}
\left\|\hat{T}_M\left(\hat{V}\right)-T\left(V\right)\right\| &= \sup_{x\in X}\left|\hat{T}_M\left(\hat{V}\right)(x)-T\left(V\right)(x)\right| \\
&\leq \sup_{x\in X}\max_{a\in A}\left|\hat{T}_{a,M}\left(\hat{V}\right)(x)-T_a\left(V\right)(x)\right| \\
&= \max_{a\in A}\sup_{x\in X}\left|\hat{T}_{a,M}\left(\hat{V}\right)(x)-T_a\left(V\right)(x)\right| \\
&= \max_{a\in A}\left\|\hat{T}_{a,M}\left(\hat{V}\right)-T_a\left(V\right)\right\| \\
&\leq \sum_{a\in A}\left\|\hat{T}_{a,M}\left(\hat{V}\right)-T_a\left(V\right)\right\|
\end{aligned}$$

■

Sea

$$G_M\left(W\right)=\frac{1}{d}\sum_{i=1}^{d}\hat{T}_M^i\left(W\right), \tag{3.16}$$

$W\in B\left(X\right)$, el cual representa el promedio de todas las funciones de valor obtenidas para las d muestras simuladas y, donde la notación $\hat{T}_M^i$ representa a la función de valor obtenida para la $i-$ésima muestra aleatoria. Note que el Lema 3.3.8 es válido si $\hat{T}_M\left(\hat{V}\right)$ y $\hat{T}_{a,M}\left(\hat{V}\right)$ son sustituidos por $\hat{T}_M^i\left(\hat{V}\right)$ y $\hat{T}_{a,M}^i\left(\hat{V}\right)$, respectivamente.

El siguiente teorema garantiza que la diferencia entre la función de valor aproximada y la función de valor se encuentra acotada en probabilidad.

Teorema 3.3.9

$$\sqrt{d}\left\|G_M\left(\hat{V}\right)-T\left(V\right)\right\|=O_p\left(1\right).$$

Demostración. Primero note que,

$$\begin{aligned}
\left\|G_M\left(\hat{V}\right)-T\left(V\right)\right\| &= \left\|\frac{1}{d}\sum_{i=1}^{d}\hat{T}_M^i\left(\hat{V}\right)-\frac{1}{d}\sum_{i=1}^{d}T\left(V\right)\right\| \\
&= \left\|\frac{1}{d}\sum_{i=1}^{d}\left(\hat{T}_M^i\left(\hat{V}\right)-T\left(V\right)\right)\right\| \\
&\leq \frac{1}{d}\sum_{i=1}^{d}\left\|\hat{T}_M^i\left(\hat{V}\right)-T\left(V\right)\right\| \\
&\leq \frac{1}{d}\sum_{i=1}^{d}\sum_{a\in A}\left\|\hat{T}_{a,M}^i\left(\hat{V}\right)-T_a\left(V\right)\right\|
\end{aligned}$$

donde la última desigualdad es debida al Lema 3.3.8. Entonces se tiene que $\left\|G_M\left(\hat{V}\right)-T\left(V\right)\right\|$ está acotado superiormente por la suma de los operadores,

$Z^i_{a,M}(W)$ definidos por

$$Z^i_{a,M}(W) := \sum_{a\in A} \left\| \hat{T}^i_{a,M}\left(\hat{V}\right) - T_a(V) \right\|.$$

Entonces, por el teorema central del límite, $Z_{a,M}(W) \xrightarrow{d} Z_a(W)$ donde, $Z_a(W)$ es una variable aleatoria normal estándar. Así, se tiene que

$$\frac{\left\| \frac{1}{d}\sum_{i=1}^{d} Z^i_{a,M}(W) \right\|}{\frac{1}{\sqrt{d}}} \to Z_a(W).$$

Entonces, existe $R > 0$ tal que

$$\lim_{d\to\infty} P\left(\sqrt{d} \left\| \frac{1}{d}\sum_{i=1}^{d} Z^i_{a,M}(W) \right\| > R \right) = 0.$$

pero esto implica que

$$\sqrt{d} \left\| \frac{1}{d}\sum_{i=1}^{d}\sum_{a\in A} \left(\hat{T}^i_{a,M}\left(\hat{V}\right) - T_a(V) \right) \right\| = O_p(1),$$

de lo cual, se tiene

$$\sqrt{d} \left\| G_M\left(\hat{V}\right) - T(V) \right\| = O_p(1).$$

■

En conclusión, el problema de control óptimo dado en el Criterio 3.3.2 puede ser aproximado por el problema de control óptimo dado en el Criterio 3.3.3.

A continuación se describe el algoritmo para determinar la aproximación, dicho algoritmo será llamado algoritmo de aleatorización. Además se describen los pasos seguidos para desarrollar este algoritmo en Matlab (véase Apéndice B),

1. Se simula una sucesión de estados aleatorios, $S_M = \{x_1, x_2, \ldots, x_M\}$, donde M es un entero positivo y denotará el tamaño de la muestra.

2. Se define un nuevo MCM $(S_M, A, \{A(x) | x \in X\} Q_M, c)$ y donde S_M representa la muestra aleatoria y Q_M es como en (3.6).

3. Se evalúa la ecuación de programación dinámica aproximada:

$$\begin{aligned} \hat{V}_t(x) &= \max_{a\in A(x)} \left\{ c(x,a) + \sum_{x_k\in S_M} \hat{V}_{t+1}(x_k)\, Q_M(x_k | x, a) \right\}, \quad (3.17) \\ t &\in \{N-1, N-2, \ldots, 1, 0\}, \end{aligned}$$

$x \in X$.

4. Una vez que se evalúa se obtiene una función de valor para esta muestra aleatoria se regresa al paso 1 y se repite este proceso una cantidad determinada de veces obteniendo una función de valor en cada repetición y finalmente se calcula un promedio de todas ellas y, está función resultante será la aproximación deseada.

La manera en que fue desarrollado este algoritmo en Matlab es la siguiente,

a. Se dan valores iniciales (cantidad de estados, número de etapas, etc);

b. Se fijan el tamaño de la muestra y el número de repeticiones;

c. Se simulan las muestras aleatorias del espacio de estados;

d. Se obtiene la ley de transición para la muestra aleatoria dada;

e. Se calcula la función de valor para las muestras simuladas;

f. Finalmente, se suman todas las funciones de valor obtenidas y se dividen entre el número de repeticiones dado.

3.4. Ejemplos de Programación Dinámica con Aleatorización

En esta sección se retoman los ejemplos vistos en el Capítulo 2, sólo que aquí son resueltos a través del algoritmo propuesto en la Sección 3.3. Se presentan los datos obtenidos para diferentes valores del tamaño de muestra y número de veces que se repite el proceso.

3.4.1. Reemplazamiento de Máquinas

Aplicando el algoritmo descrito anteriormente al problema de reemplazamiento de máquinas, con la ecuación de programación dinámica aproximada dada por

$$\hat{V}_t(x) = \min \left\{ \begin{array}{c} R + g(1) + \hat{V}_{t+1}(1), \\ g(x) + \sum_{y \in S_M} Q_M(y \,|x\,) \hat{V}_{t+1}(y) \end{array} \right\},$$

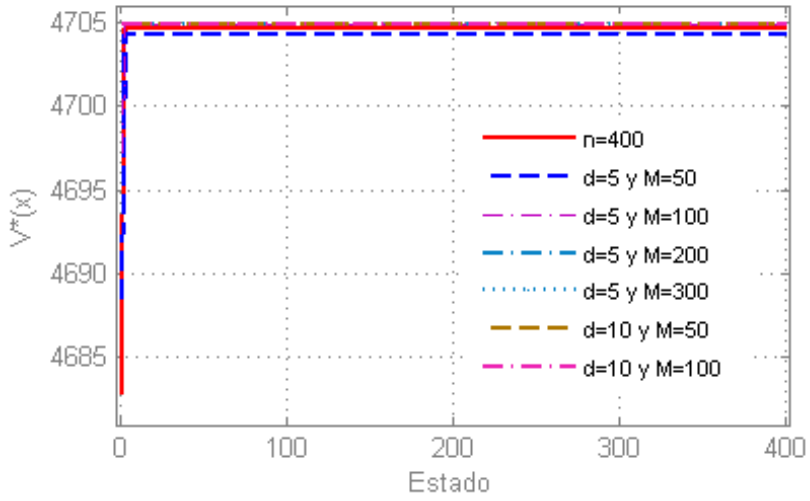
donde S_M es una muestra aleatoria de X de tamaño M. Y, cuando se consideran $n = 400,\ N = 60,\ R = 109\,.\,87$ y $g(x) = 21\,.\,16x + 2\,.\,32$; al implementar la técnica de programación dinámica en un algoritmo en Matlab (véase Apéndice B), se tiene que con un tiempo de ejecución de $144\,.\,833916$ segundos, la función de valor está dada por

$$V^*(x) = \left\{ \begin{array}{ll} 4682\,.\,6 & \text{si } x = 1, \\ 4704\,.\,3 & \text{si } x = 2, \\ 4704\,.\,6 & \text{si } x = 3, \ldots, 400, \end{array} \right. .$$

La siguiente tabla muestra los valores obtenidos cuando se varían el tamaño de la muestra y el número de veces que se repite el proceso,

d	M	$Tiempo$	$\hat{V}(x)$
5	50	5.255437s	4688.1 si $x=1$; 4692.7 si $x=2$ 4704.2 si $x=3,\ldots,400$
	100	8.946899s	4697.3 si $x=1$; 4703.7 si $x=2$; 4704.9 si $x=3,\ldots,400$
	200	84.83887s	4704.9 si $x=1,\ldots,400$
	300	132.507803s	4704.9 si $x=1,\ldots,400$
10	50	38.086780s	4704.9 si $x=1,\ldots,400$
	100	73.797526s	4704.9 si $x=1,\ldots,400$

De la tabla anterior, se tiene que las aproximaciones a la función de valor es cercana y además el tiempo de ejecución del algoritmo disminuyó considerablemente. En la siguiente gráfica se representa a la función de valor representada por la línea roja y las aproximaciones obtenidas las cuales, son representadas por las líneas de diferente color.



Aleatorizacion en ejemplo de reemplazamiento de máquina

3.4.2. Inventarios

Ahora se aplicará el algoritmo de aleatorización al problema de inventario, éste problema servirá para hacer notar que si la condición: $A = A(x)$, $x \in X$ no se cumple entonces, el algoritmo de aleatorización no resulta ser útil ya que, las aproximaciones obtenidas no son cercanas a la función de valor y el tiempo de ejecución resulta ser mayor al obtenido usando la técnica de programación dinámica directamente.

Así, en este caso con la ecuación de programación dinámica aproximada está dada por

$$\hat{V}_t(x) = \max_{a \in A(x)} \left\{ r(x,a) + \sum_{y \in S_M} Q_M(y|x,a)\, \hat{V}_{t+1}(y) \right\},$$

$x \in X$.

A continuación se presenta una tabla con la información obtenida al implementar aleatorización en Matlab para el problema de inventario con los siguientes costos e ingresos, $K = 36$, $c(x) = 14x$, $g(x) = 0$, $h(x) = 21x$, $Z = 180$, $N = 65$, $f(x) = 56x$, donde lo que representa es la función de valor para cada estado, con un tiempo de ejecución de 28.505302 segundos,

x	$V(x)$	x	$V(x)$	x	$V(x)$	x	$V(x)$
0	56602	46	57246	92	57926	138	58286
1	56616	47	57260	93	57939	139	58286
2	56630	48	57274	94	57953	140	58287
3	56644	49	57288	95	57966	141	58286
4	56658	50	57302	96	57978	142	58286
5	56672	51	57316	97	57991	143	58285
6	56686	52	57330	98	58004	144	58284
7	56700	53	57344	99	58016	145	58282
8	56714	54	57358	100	58028	146	58281
9	56728	55	57372	101	58040	147	58279
10	56742	56	57386	102	58052	148	58276
11	56756	57	57400	103	58064	149	58274
12	56770	58	57414	104	58075	150	58271
13	56784	59	57428	105	58086	151	58267
14	56798	60	57442	106	58097	152	58263
15	56812	61	57456	107	58107	153	58259
16	56826	62	57470	108	58117	154	58254
17	56840	63	57484	109	58127	155	58249
18	56854	64	57498	110	58137	156	58244
19	56868	65	57512	111	58147	157	58239
20	56882	66	57526	112	58156	158	58233
21	56896	67	57540	113	58165	159	58226
22	56910	68	57554	114	58174	160	58220
23	56924	69	57568	115	58182	161	58213
24	56938	70	57582	116	58190	162	58206
25	56952	71	57596	117	58198	163	58199
26	56966	72	57610	118	58206	164	58191
27	56980	73	57628	119	58213	165	58183
28	56994	74	57645	120	58220	166	58175
29	57008	75	57663	121	58226	167	58167
30	57022	76	57680	122	58232	168	58158
31	57036	77	57697	123	58237	169	58149

32	57050	78	57714	124	58243	170	58139
33	57064	79	57730	125	58248	171	58130
34	57078	80	57747	126	58252	172	58120
35	57092	81	57763	127	58257	173	58110
36	57106	82	57779	128	58261	174	58099
37	57120	83	57795	129	58265	175	58087
38	57134	84	57810	130	58268	176	58076
39	57148	85	57825	131	58272	177	58064
40	57162	86	57840	132	58275	178	58051
41	57176	87	57855	133	58277	179	58039
42	57190	88	57869	134	58280	180	58025
43	57204	89	57883	135	58282		
44	57218	90	57898	136	58284		
45	57232	91	57912	137	58285		

Al implementar la técnica de aleatorización en un algoritmo en Matlab para diferentes valores de tamaño de muestra y número de repeticiones se dan a continuación las funciones de valor aproximado obtenidas para los mismos valores dados arriba. Se puede notar que debido a que el problema de inventario no cumple con la condición de $A = A(x)$, $x \in X$, se tiene que tanto las aproximaciones como el tiempo de ejecución del algoritmo resultar no ser tan buenas como en el caso del ejemplo de reemplzamiento de máquina

Así, considerando que el tamaño de muestra es de $M = 50$ y que el proceso es repetido $d = 5$ veces, se obtiene la siguiente función de valor, con un tiempo de ejecución de 157.757136,

x	$\hat{V}(x)$	x	$\hat{V}(x)$	x	$\hat{V}(x)$	x	$\hat{V}(x)$
0	70719	46	71363	92	72026	138	72221
1	70733	47	71377	93	72021	139	72218
2	70747	48	71391	94	72061	140	72230
3	70761	49	71405	95	72085	141	72242
4	70775	50	71419	96	72065	142	72231
5	70789	51	71433	97	72074	143	72250
6	70803	52	71447	98	72089	144	72192
7	70817	53	71461	99	72117	145	72203
8	70831	54	71475	100	72116	146	72253
9	70845	55	71489	101	72130	147	72237
10	70859	56	71503	102	72144	148	72191
11	70873	57	71517	103	72194	149	72202
12	70887	58	71531	104	72150	150	72196
13	70901	59	71545	105	72148	151	72189
14	70915	60	71559	106	72189	152	72177
15	70929	61	71573	107	72151	153	72153
16	70943	62	71587	108	72179	154	72166
17	70957	63	71601	109	72196	155	72158
18	70971	64	71615	110	72166	156	72139

19	70985	65	71629	111	72209	157	72147
20	70999	66	71643	112	72192	158	72131
21	71013	67	71657	113	72231	159	72152
22	71027	68	71671	114	72217	160	72141
23	71041	69	71685	115	72179	161	72117
24	71055	70	71699	116	72229	162	72116
25	71069	71	71713	117	72240	163	72091
26	71083	72	71727	118	72243	164	72072
27	71097	73	71741	119	72229	165	72115
28	71111	74	71755	120	72221	166	72070
29	71125	75	71769	121	72233	167	72039
30	71139	76	71783	122	72243	168	72055
31	71153	77	71797	123	72261	169	72031
32	71167	78	71811	124	72240	170	72018
33	71181	79	71825	125	72264	171	72027
34	71195	80	71839	126	72243	172	71997
35	71209	81	71860	127	72257	173	72011
36	71223	82	71867	128	72307	174	71959
37	71237	83	71898	129	72255	175	71988
38	71251	84	71917	130	72261	176	71970
39	71265	85	71909	131	72243	177	71961
40	71279	86	71949	132	72241	178	71935
41	71293	87	71938	133	72245	179	71901
42	71307	88	71977	134	72279	180	71943
43	71321	89	71965	135	72268		
44	71335	90	71979	136	72271		
45	71349	91	72015	137	72231		

Ahora, sean $d = 5$, $M = 100$ y con un tiempo de ejecución de $315\,.\,759360$ segundos se obtuvo la siguiente función de valor,

x	$\hat{V}(x)$	x	$\hat{V}(x)$	x	$\hat{V}(x)$	x	$\hat{V}(x)$
0	62133	46	62777	92	63400	138	63683
1	62147	47	62791	93	63433	139	63688
2	62161	48	62805	94	63419	140	63687
3	62175	49	62819	95	63445	141	63670
4	62189	50	62833	96	63463	142	63695
5	62203	51	62847	97	63475	143	63673
6	62217	52	62861	98	63463	144	63658
7	62231	53	62875	99	63497	145	63646
8	62245	54	62889	100	63474	146	63659
9	62259	55	62903	101	63503	147	63664
10	62273	56	62917	102	63511	148	63655
11	62287	57	62931	103	63540	149	63693
12	62301	58	62945	104	63552	150	63647
13	62315	59	62959	105	63550	151	63663
14	62329	60	62973	106	63548	152	63650

15	62343	61	62987	107	63576	153	63642
16	62357	62	63001	108	63563	154	63658
17	62371	63	63015	109	63584	155	63642
18	62385	64	63029	110	63564	156	63596
19	62399	65	63043	111	63591	157	63603
20	62413	66	63057	112	63589	158	63593
21	62427	67	63078	113	63611	159	63603
22	62441	68	63085	114	63619	160	63616
23	62455	69	63099	115	63617	161	63588
24	62469	70	63113	116	63653	162	63588
25	62483	71	63131	117	63624	163	63573
26	62497	72	63144	118	63630	164	63578
27	62511	73	63157	119	63655	165	63578
28	62525	74	63170	120	63640	166	63555
29	62539	75	63198	121	63690	167	63556
30	62553	76	63219	122	63651	168	63552
31	62567	77	63211	123	63661	169	63514
32	62581	78	63227	124	63678	170	63528
33	62595	79	63242	125	63648	171	63518
34	62609	80	63261	126	63656	172	63492
35	62623	81	63280	127	63671	173	63500
36	62637	82	63281	128	63679	174	63476
37	62651	83	63301	129	63665	175	63494
38	62665	84	63309	130	63681	176	63490
39	62679	85	63323	131	63673	177	63469
40	62693	86	63373	132	63687	178	63466
41	62707	87	63341	133	63699	179	63428
42	62721	88	63374	134	63685	180	63461
43	62735	89	63361	135	63683		
44	62749	90	63380	136	63687		
45	62763	91	63409	137	63698		

Considerando, que el tamaño de muestra es de $M = 150$ y el número de repeticiones es de $d = 5$ se obtuvo la siguiente función de valor con un tiempo de ejecución de 466.935032 segundos.

x	$\hat{V}(x)$	x	$\hat{V}(x)$	x	$\hat{V}(x)$	x	$\hat{V}(x)$
0	60920	46	61564	92	62244	138	62530
1	60934	47	61578	93	62246	139	62521
2	60948	48	61592	94	62244	140	62526
3	60962	49	61606	95	62284	141	62535
4	60976	50	61620	96	62270	142	62531
5	60990	51	61634	97	62292	143	62530
6	61004	52	61648	98	62304	144	62517
7	61018	53	61662	99	62308	145	62511
8	61032	54	61676	100	62336	146	62537

9	61046	55	61690	101	62345	147	62509
10	61060	56	61704	102	62352	148	62514
11	61074	57	61718	103	62381	149	62487
12	61088	58	61732	104	62369	150	62496
13	61102	59	61746	105	62383	151	62506
14	61116	60	61760	106	62382	152	62484
15	61130	61	61774	107	62390	153	62494
16	61144	62	61788	108	62426	154	62476
17	61158	63	61802	109	62409	155	62460
18	61172	64	61816	110	62447	156	62462
19	61186	65	61830	111	62448	157	62443
20	61200	66	61844	112	62439	158	62445
21	61214	67	61858	113	62470	159	62456
22	61228	68	61872	114	62454	160	62436
23	61242	69	61886	115	62455	161	62451
24	61256	70	61900	116	62474	162	62407
25	61270	71	61914	117	62474	163	62442
26	61284	72	61930	118	62485	164	62418
27	61298	73	61942	119	62476	165	62397
28	61312	74	61956	120	62495	166	62392
29	61326	75	61970	121	62516	167	62382
30	61340	76	61984	122	62496	168	62374
31	61354	77	62001	123	62516	169	62377
32	61368	78	62012	124	62490	170	62334
33	61382	79	62026	125	62510	171	62352
34	61396	80	62063	126	62503	172	62335
35	61410	81	62062	127	62513	173	62325
36	61424	82	62087	128	62532	174	62310
37	61438	83	62096	129	62528	175	62310
38	61452	84	62106	130	62512	176	62305
39	61466	85	62128	131	62535	177	62287
40	61480	86	62130	132	62517	178	62273
41	61494	87	62172	133	62525	179	62267
42	61508	88	62165	134	62526	180	62270
43	61522	89	62178	135	62524		
44	61536	90	62210	136	62531		
45	61550	91	62220	137	62512		

Ahora, considerando a $M = 50$ y $d = 10$ se obtuvo la siguiente función de valor, con un tiempo de ejecución de $316\,.\,999579$ segundos,

x	$\hat{V}(x)$	x	$\hat{V}(x)$	x	$\hat{V}(x)$	x	$\hat{V}(x)$
0	76077	46	76721	92	77376	138	77562
1	76091	47	76735	93	77415	139	77540
2	76105	48	76749	94	77404	140	77540
3	76119	49	76763	95	77383	141	77555

4	76133	50	76777	96	77410	142	77547
5	76147	51	76791	97	77438	143	77539
6	76161	52	76805	98	77438	144	77534
7	76175	53	76819	99	77463	145	77539
8	76189	54	76833	100	77460	146	77516
9	76203	55	76847	101	77457	147	77515
10	76217	56	76861	102	77470	148	77514
11	76231	57	76875	103	77488	149	77495
12	76245	58	76889	104	77534	150	77499
13	76259	59	76903	105	77523	151	77493
14	76273	60	76917	106	77506	152	77472
15	76287	61	76931	107	77541	153	77463
16	76301	62	76945	108	77538	154	77441
17	76315	63	76959	109	77521	155	77470
18	76329	64	76973	110	77554	156	77441
19	76343	65	76987	111	77560	157	77410
20	76357	66	77001	112	77542	158	77427
21	76371	67	77015	113	77579	159	77406
22	76385	68	77029	114	77546	160	77397
23	76399	69	77043	115	77587	161	77403
24	76413	70	77057	116	77578	162	77359
25	76427	71	77071	117	77583	163	77359
26	76441	72	77085	118	77595	164	77375
27	76455	73	77099	119	77581	165	77332
28	76469	74	77113	120	77578	166	77329
29	76483	75	77127	121	77572	167	77315
30	76497	76	77141	122	77586	168	77308
31	76511	77	77155	123	77580	169	77304
32	76525	78	77169	124	77591	170	77261
33	76539	79	77183	125	77620	171	77268
34	76553	80	77197	126	77566	172	77244
35	76567	81	77215	127	77593	173	77224
36	76581	82	77251	128	77562	174	77232
37	76595	83	77254	129	77573	175	77209
38	76609	84	77253	130	77618	176	77195
39	76623	85	77274	131	77574	177	77183
40	76637	86	77297	132	77578	178	77160
41	76651	87	77321	133	77587	179	77129
42	76665	88	77314	134	77571	180	77115
43	76679	89	77336	135	77589		
44	76693	90	77346	136	77597		
45	76707	91	77351	137	77577		

Suponga que el número repeticiones es de $d = 10$ y el tamaño de las muestras es de $M = 100$, entonces con un tiempo de ejecución de 637.043960 segundos

se obtuvo la siguiente función de valor,

x	$\hat{V}(x)$	x	$\hat{V}(x)$	x	$\hat{V}(x)$	x	$\hat{V}(x)$
0	71240	46	71884	92	72550	138	72769
1	71254	47	71898	93	72566	139	72770
2	71268	48	71912	94	72573	140	72753
3	71282	49	71926	95	72581	141	72741
4	71296	50	71940	96	72570	142	72746
5	71310	51	71954	97	72620	143	72738
6	71324	52	71968	98	72610	144	72743
7	71338	53	71982	99	72630	145	72742
8	71352	54	71996	100	72640	146	72720
9	71366	55	72010	101	72657	147	72727
10	71380	56	72024	102	72671	148	72713
11	71394	57	72038	103	72669	149	72715
12	71408	58	72052	104	72697	150	72720
13	71422	59	72066	105	72689	151	72684
14	71436	60	72080	106	72684	152	72674
15	71450	61	72094	107	72707	153	72674
16	71464	62	72108	108	72709	154	72666
17	71478	63	72122	109	72738	155	72661
18	71492	64	72136	110	72738	156	72667
19	71506	65	72150	111	72733	157	72649
20	71520	66	72164	112	72737	158	72618
21	71534	67	72178	113	72732	159	72619
22	71548	68	72192	114	72753	160	72617
23	71562	69	72206	115	72771	161	72617
24	71576	70	72220	116	72758	162	72605
25	71590	71	72234	117	72761	163	72566
26	71604	72	72248	118	72757	164	72573
27	71618	73	72263	119	72774	165	72549
28	71632	74	72276	120	72769	166	72547
29	71646	75	72290	121	72779	167	72552
30	71660	76	72312	122	72759	168	72517
31	71674	77	72318	123	72757	169	72509
32	71688	78	72341	124	72768	170	72492
33	71702	79	72364	125	72743	171	72504
34	71716	80	72374	126	72774	172	72489
35	71730	81	72396	127	72781	173	72453
36	71744	82	72416	128	72777	174	72433
37	71758	83	72434	129	72780	175	72434
38	71772	84	72449	130	72775	176	72411
39	71786	85	72456	131	72778	177	72390
40	71800	86	72468	132	72786	178	72389
41	71814	87	72489	133	72773	179	72380
42	71828	88	72494	134	72786	180	72354

43	71842	89	72500	135	72772
44	71856	90	72536	136	72773
45	71870	91	72537	137	72775

Sean $d = 20$ y $M = 50$, con un tiempo de ejecución de $657\,.\,143881$, se obtuvo la siguiente función de valor,

x	$\hat{V}(x)$	x	$\hat{V}(x)$	x	$\hat{V}(x)$	x	$\hat{V}(x)$
0	74550	46	75194	92	75870	138	76080
1	74564	47	75208	93	75873	139	76095
2	74578	48	75222	94	75889	140	76076
3	74592	49	75236	95	75892	141	76081
4	74606	50	75250	96	75907	142	76080
5	74620	51	75264	97	75920	143	76056
6	74634	52	75278	98	75933	144	76081
7	74648	53	75292	99	75933	145	76055
8	74662	54	75306	100	75961	146	76066
9	74676	55	75320	101	75951	147	76045
10	74690	56	75334	102	75995	148	76036
11	74704	57	75348	103	75992	149	76028
12	74718	58	75362	104	76009	150	76033
13	74732	59	75376	105	76006	151	75999
14	74746	60	75390	106	75984	152	76014
15	74760	61	75404	107	76006	153	75977
16	74774	62	75418	108	76008	154	75981
17	74788	63	75432	109	76046	155	75969
18	74802	64	75446	110	76059	156	75965
19	74816	65	75460	111	76060	157	75965
20	74830	66	75474	112	76061	158	75942
21	74844	67	75488	113	76068	159	75949
22	74858	68	75502	114	76093	160	75935
23	74872	69	75516	115	76087	161	75925
24	74886	70	75530	116	76104	162	75916
25	74900	71	75544	117	76082	163	75883
26	74914	72	75560	118	76080	164	75877
27	74928	73	75572	119	76085	165	75884
28	74942	74	75586	120	76093	166	75854
29	74956	75	75600	121	76112	167	75853
30	74970	76	75614	122	76128	168	75822
31	74984	77	75638	123	76066	169	75834
32	74998	78	75642	124	76105	170	75829
33	75012	79	75667	125	76095	171	75780
34	75026	80	75682	126	76095	172	75788
35	75040	81	75703	127	76122	173	75768
36	75054	82	75718	128	76115	174	75749

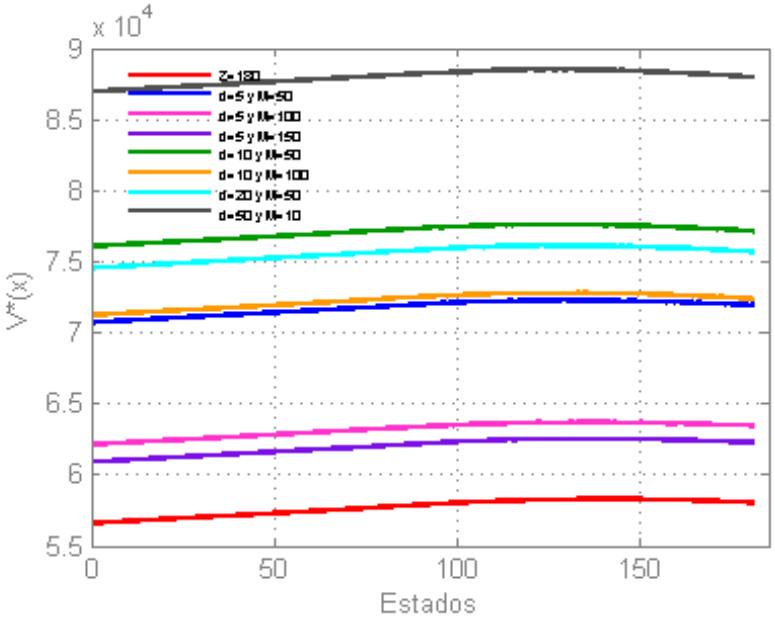
37	75054	83	75748	129	76099	175	75747
38	75068	84	75729	130	76105	176	75727
39	75096	85	75747	131	76112	177	75705
40	75110	86	75763	132	76105	178	75710
41	75124	87	75765	133	76118	179	75691
42	75138	88	75785	134	76086	180	75660
43	75152	89	75812	135	76097		
44	75166	90	75843	136	76069		
45	75180	91	75837	137	76091		

Finalmente, cuando $d = 50$ y $M = 10$ se obtuvo la siguiente función de valor, con un tiempo de ejecución de 442.701393 segundos,

x	$\hat{V}(x)$	x	$\hat{V}(x)$	x	$\hat{V}(x)$	x	$\hat{V}(x)$
0	86961	46	87605	92	88250	138	88465
1	86975	47	87619	93	88249	139	88445
2	86989	48	87633	94	88262	140	88451
3	87003	49	87647	95	88279	141	88442
4	87017	50	87661	96	88272	142	88449
5	87031	51	87675	97	88322	143	88431
6	87045	52	87689	98	88311	144	88437
7	87059	53	87703	99	88333	145	88418
8	87073	54	87717	100	88333	146	88411
9	87087	55	87731	101	88358	147	88402
10	87101	56	87745	102	88355	148	88387
11	87115	57	87759	103	88352	149	88388
12	87129	58	87773	104	88377	150	88360
13	87143	59	87787	105	88380	151	88342
14	87157	60	87801	106	88394	152	88345
15	87171	61	87815	107	88404	153	88326
16	87185	62	87829	108	88430	154	88324
17	87199	63	87843	109	88472	155	88311
18	87213	64	87857	110	88448	156	88307
19	87227	65	87871	111	88460	157	88288
20	87241	66	87885	112	88451	158	88263
21	87255	67	87899	113	88480	159	88277
22	87269	68	87913	114	88452	160	88256
23	87283	69	87927	115	88463	161	88241
24	87297	70	87941	116	88469	162	88219
25	87311	71	87955	117	88462	163	88180
26	87325	72	87969	118	88474	164	88187
27	87339	73	87983	119	88508	165	88186
28	87353	74	87997	120	88468	166	88165
29	87367	75	88011	121	88496	167	88168
30	87381	76	88025	122	88478	168	88133
31	87395	77	88039	123	88483	169	88116

32	87409	78	88053	124	88533	170	88107
33	87423	79	88067	125	88482	171	88109
34	87437	80	88081	126	88496	172	88101
35	87451	81	88095	127	88522	173	88065
36	87465	82	88109	128	88487	174	88045
37	87479	83	88123	129	88493	175	88035
38	87493	84	88137	130	88520	176	88028
39	87507	85	88151	131	88501	177	87995
40	87521	86	88166	132	88499	178	88007
41	87535	87	88215	133	88487	179	87964
42	87549	88	88190	134	88492	180	87979
43	87563	89	88203	135	88496		
44	87577	90	88253	136	88483		
45	87591	91	88244	137	88448		

En la siguiente gráfica se muestran las funciones de valor obtenida, donde la línea roja representa la función de valor verdadera y las demás líneas las aproximaciones obtenidas.



Aleatorización en el problema de inventarios

Se puede observar que las aproximaciones obtenidas no son tan buenas como en el ejemplo de reemplazamiento de máquinas, pero esto es debido a la dependencia del estado de acciones sobre el espacio de estados, abriendo así el camino para el estudio de este tipo de problemas y la búsqueda de algoritmos que funcionen mejor que el método de aleatorización.

Conclusiones

En la tesis se trabajó con la teoría de Procesos de Decisión de Markov (PDMs) a tiempo discreto con horizonte finito y espacio de estados y acciones finito. Para el análisis de los PDMs se estudió la técnica de Programación Dinámica (PD), la cual permite determinar a las funciones de valor óptimo. Una de las desventajas de PD surge cuando el espacio de estados, horizonte y/o acciones incrementan en cardinalidad, llevando así a la implementación de algoritmos computacionales. Sin embargo, de nueva cuenta la gran cantidad de estados ocasiona que el costo computacional crezca exponencialmente; esto es conocido como La Maldición de la Dimensionalidad. En la tesis se estudió un algoritmo con la finalidad de evitar el problema de la dimensionalidad, el cual consistió en aproximar a la función de valor mediante la simulación de muestras aleatorias sobre el espacio de estados y en evaluar en dichas muestras una ecuación de programación dinámica aproximada para obtener funciones de valor aproximadas a partir de las cuales se obtiene una aproximación a la verdadera función de valor mediante el promedio de ellas.

Se revisaron dos problemas, el primero de ellos fue el de reemplazamiento de una máquina y el segundo fue un problema de inventario; ambos problemas fueron resueltos primero mediante el uso de programación dinámica y después implementando el algoritmo de aleatorización de estados. Para ambos casos, usando programación dinámica y aleatorización, se realizaron algoritmos en Matlab y, a través de ellos se presentan ejemplos numéricos.

Mediante el problema de inventario se hizo notar que este método no es tan bueno cuando el espacio de acciones depende del espacio de estados. Así, un trabajo futuro sería el estudio de está clase de problemas y la búsqueda de algoritmos que permitan evitar el problema de la dimensionalidad sin la restricción de $A = A(x)$ para cada $x \in X$.

También, se probó que la diferencia entre la función de valor verdadera y la función de valor aproximada se encuentra acotada en probabilidad y se incluyó una sección donde es descrito el método Monte Carlo, el cual fue utilizado para las simulaciones de la muestra de estados, además de que se incluyeron ejemplos donde se describe este método.

Posibles consecuencias y trabajos futuros que podrían derivar de este trabajo son: el estudio de métodos para reducir el problema de la maldición pero ahora aplicados al espacio de acciones o al horizonte del problema; el estudió de los PDMs en el caso continuo y con horizonte infinito y la búsqueda de algoritmos

para tales casos.

Apéndice A

Resultados Auxiliares

A continuación se dan una serie de resultados auxiliares utilizados los cuales son utilizados en el desarrollo de la tesis.

Definición A.0.1 *X es un espacio de Borel, si X es un subconjunto de Borel de un espacio métrico, separable y completo.*

Definición A.0.2 *Sea (X,τ) un espacio topológico, la mínima $\sigma-$álgebra que contiene a τ es la $\sigma-$álgebra de Borel, es decir, la $\sigma-$álgebra generada por τ. Será denotada por $\mathcal{B}(X)$.*

De aquí en adelante, cuando se hable de conjuntos o funciones medibles, se entenderán como Borel medibles.

Definición A.0.3 *Sean (X,d) un espacio métrico y $v : X \to \mathbb{R} \cup \{+\infty\}$ una función tal que $v(x) < \infty$ para al menos un punto $x \in X$, se dirá que la función v es semicontinua inferiormente (l.s.c) en x, si*

$$\liminf_{n\to\infty} v(x_n) \geq v(x)$$

para cualquier sucesión $\{x_n\}$ en X que converge a x. La función v es llamada inferiormente semicontinua (l.s.c) si es l.s.c para cada x en X.

Observación A.0.4 *La función v es semicontinua superiormente (u.s.c), si $-v$ es l.s.c. Más aún, v es continua, si y solo si, es l.s.c y u.s.c.*

Definición A.0.5 *Una función $v : \mathbb{K} \to \mathbb{R}$ se llama inf-compacta sobre $\mathbb{K}$, si para toda $x \in X$ y $r \in \mathbb{R}$, el conjunto $\{a \in A(x) \,|\, v(x,a) \leq r\}$ es compacto.*

Definición A.0.6 *La función $v'(x,a) := \int v(y)\, Q(dy\,|x,a)$ sobre $\mathbb{K}$ se dirá que es*

a) Débilmente continua si, $v'(x,a)$ es continua y acotada en $\mathbb{K}$ para cada función continua y acotada, v sobre X; ó

b) Fuertemente continua si, $v'(x,a)$ es continua y acotada en $\mathbb{K}$ para cada función medible y acotada, v sobre X.

Teorema A.0.7 *(de Ionescu-Tulcea). Sean $X_0, X_1, \ldots,$ una sucesión de espacios de Borel y, para $n = 0, 1, \ldots,$ se define $Y_n := X_0 \times X_1 \times \cdots \times X_n$ y $Y := \prod_{n=0}^{\infty} X_n$. Sea v una medida de probabilidad arbitraria sobre X_0 y para cada $n = 0, 1, \ldots, P_n(dx_{n+1}|y_n)$ es una probabilidad condicional sobre X_{n+1} dada Y_n. Entonces existe una única medida de probabilidad, P_v sobre Y tal que, para cada rectángulo medible $B_0 \times B_1 \times \cdots \times B_n$ en Y_n,*

$$\begin{aligned} P_v(B_0 \times B_1 \times \cdots \times B_n) &= \int_{B_0} v(dx_0) \int_{B_1} P_0(dx_1|x_0) \\ &\quad \int_{B_2} P_1(dx_2|x_0, x_1) \cdots \\ &\quad \int_{B_n} P_{n-1}(dx_n|x_0, x_1, \ldots, x_{n-1}). \end{aligned}$$

Además, para cualquier función u, medible y no negativa sobre Y, la función

$$x \to \int u(y) P_x(dy)$$

es medible en X_0, donde P_x representa a P_v cuando v es la probabilidad concentrada en $x \in X_0$.

Demostración. Véase ([1]). ■

Sean X y A espacios de Borel.

Una multifunción ψ de X a A es una función tal que para toda $x \in X$ su imagen $\psi(x)$ es un subconjunto no vacío de A.La gráfica de ψ es el subconjunto de $X \times A$ definido por,

$$Gr(\psi) := \{(x,a) | x \in X, a \in \psi(x)\}. \tag{A.1}$$

En el trabajo, se considera a $\psi(x)$ como $A(x)$ y la gráfica $Gr(\psi)$ como $\mathbb{K}$. Para cada subconjunto B de A, sea $\psi^{-1}[B] := \{x \in X | \psi(x) \cap B \neq \emptyset\}$.

Definición A.0.8 *Una multifunción, $\psi : X \to A$, se dice,*

1. *Borel medible, si $\psi^{-1}(G)$ es Borel medible en X para cualquier conjunto abierto $G \subset A$;*

2. *Semicontinua superiormente (u.s.c), si $\psi^{-1}(F)$ es cerrado en X para todo conjunto cerrado $F \subset A$;*

3. *Semicontinua inferiormente (l.s.c), si* $\psi^{-1}(G)$ *es abierto en* X *para todo conjunto abierto* $G \subset A$;

4. *Continua si, es ambas, u.s.c y l.s.c.*

5. *Cerrada, si* $\psi(x)$ *es cerrado para toda* $x \in X$*;*

6. *Compacta, si* $\psi(x)$ *es compacta para toda* $x \in X$*.*

Se supondrá que la multifunción ψ es Borel medible, $v : Gr(\psi) \to \mathbb{R}$ es una función medible y para cada $x \in X$.

$$v^*(x) . = \inf_{a \in \psi(x)} v(x, a) .$$

Además, si $v(x, \cdot)$ alcanza su mínimo en algún punto de $\psi(x)$, se puede escribir min en lugar de inf.

Proposición A.0.9 *Suponga que* ψ *es compacta.*

a) Si $v(x, \cdot)$ *es l.s.c sobre* $\psi(x)$ *para cada* $x \in X$*, entonces existe un selector* $f \in \mathbb{F}$ *tal que*

$$v(x, f(x)) = v^*(x) = \min_{\psi(x)} v(x, a)$$

para cada $x \in X$ *y* v^* *es medible.*

b) Si ψ *es u.s.c y* v *es l.s.c y acotada inferiormente sobre* $Gr(\psi)$*, entonces existe un selector* $f \in F$ *tal que la relación anterior se cumple y* v^* *es l.s.c y acotada inferiormente en* X*.*

Apéndice B

Algoritmos

En este apéndice se presentan los algoritmos desarrollados en MATLAB que permitieron resolver los problemas planteados en los capítulos 2 y 3. Primero se presentará el algoritmo para determinar a la función de valor verdadera seguido del algoritmo desarrollado para disminuir el problema de la Maldición.

B.1. Reemplazamiento de Máquinas

Algoritmo B.1.1 *(Función verdadera)*

```
clear all tic;
n=500; N=35;
R=25.3; g=zeros(1,n);
P=zeros(n,n);
V=zeros(N+1,n);
c=zeros(1,n); A=rand(n);
B=triu(A);
for i=1:n
c(i)=sum(B(i,:));
g(i)=20*i+1;
end
for i= 1:n
for j=1:n
P(i,j)=B(i,j)/c(i);
end
end
for k=2:N+1
for i=1:n
for j=i:n
V(k,i)=min(R+g(1)+ V(k-1,1), g(i)+ sum(P(i,:).*V(k-1,:)));
end
end
```

```
end
V(N,:)
toc
```

Algoritmo B.1.2 *(Utilizando aleatorización)*

```
tic;
n=500;
N=35;
R=25.3;
d=100;
prom=10;
sj=zeros(prom,d);
g=zeros(1,n);
P=zeros(n,n);
VA=zeros(N+1,n); }
VP=zeros(N+1,n,prom);
vp=zeros(N+1,n);
c=zeros(1,n);
A=rand(n);
B=triu(A);
for i=1:n
c(i)=sum(B(i,:));
g(i)=20*i+1;
end for i= 1:n
for j=1:n
P(i,j)=B(i,j)/c(i);
end
end
for i=1:prom
sj(i,:)=randint(1,d,[1,n]);
end
deno=zeros(prom,n);
for y=1:prom
for i=1:n
for t=1:d
for j=sj(y,t)
deno(y,i)= deno(y,i)+ P(i,j);
end
end
end
end
p2=zeros(n,n,prom,d);
for y=1:prom
for i=1:n
for t=1:d
j=sj(y,t);
```

```
if (deno(y,i)==0)
p2(i,j,y,t)=0;
else p2(i,j,y,t)=P(i,j)/deno(y,i);
end
end
end
end
for y=1:prom
for k=2:N+1
for i=1:n
q=0;
for t=1:d
j=sj(y,t);
q = q + p2(i,j,y,t).*VA(k-1,j);
end
VA(k,i)=min(R + g(1)+ VA(k-1,1), g(i)+ q);
end
end VP(:,:,y)=VA(:,:);
end
for i=2:N+1
for j=1:n
vp(i,j)=(1/prom)*sum(VP(i,j,:));
end
end
vp(N,:)
toc
```

B.2. Inventarios

Algoritmo B.2.1 *(Función Verdadera)*

```
clear all
tic
M=100;
N=30;
K=4;
P=zeros(M+1,M+1);
u=zeros(M+1,M+1,N);
F=zeros(1,M+1);
c=zeros(1,M+1);
f=zeros(1,M+1);
h=zeros(1,M+1);
p=zeros(1,M+1);
q=zeros(1,M+1);
o=zeros(1,M+1);
A=rand(1,M+1);
```

```
r=zeros(M+1,M+1);
for i=1:M+1
p(i)=A(i)/sum(A);
end
q(1)=1;
for i=2:M+1
for j=1:i-1
q(i)=q(i)+ p(j);
end
q(i)=1-q(i);
end
for i=1:M+1
h(i)=i-1;
c(i)=2*(i-1);
f(i)=8*(i-1);
end
for i=2:M+1
for j=1:i-1
F(i)=F(i) + f(j)*p(j);
end
F(i)=F(i) + f(i)*q(i);
o(i)=K + c(i);
end
for i=1:M+1
for j=1:M+2-i
r(i,j)=F(i+j-1)-o(j)-h(i+j-1);
end
end
for i=1:M+1
for j=1:M+1
if j>i
P(i,j)=0;
elseif (i>=j && j>1)
P(i,j)=p(i-j+1);
elseif j==1
P(i,j)=q(i+j-1);
end
end
end
for t=2:N
y=zeros(1,M+1);
for i=1:M+1
y(i)=max(u(i,:,t-1));
end
for i=1:M+1
for j=1:M+2-i
```

```
u(i,j,t)= r(i,j) + P(i+j-1,:)*y(:);
end
end
end
V=zeros(1,M+1);
for i=1:M+1
V(i)=max(u(i,:,N));
end
V;
toc
```

Algoritmo B.2.2 *(Usando Aleatorización)*

```
clear all
tic
M=100; %Estados
N=30; %Etapa
d=20; %Tamaño de muestra
prom=25; %Total de iteraciones
sj=zeros(prom,d); %Vector de estados aleatorios
K=4; %Constante de costo
P=zeros(M+1,M+1); %Matriz de transicion
up=zeros(M+1,M+1,prom);
F=zeros(1,M+1); %Ingreso esperado
c=zeros(1,M+1); %Costo por unidad ordenada
f=zeros(1,M+1); %Ganancia por unidad vendida
h=zeros(1,M+1); %costo de almacenaje
p=zeros(1,M+1); %Probabiidad de demanda
q=zeros(1,M+1);
o=zeros(1,M+1);
A=rand(1,M+1);
r=zeros(M+1,M+1); %Matriz de recompensa
for i=1:M+1
p(i)=A(i)/sum(A);
end
q(1)=1;
for i=2:M+1
for j=1:i-1
q(i)=q(i)+ p(j);
end
q(i)=1-q(i);
end
for i=1:M+1
h(i)=3*(i-1);
c(i)=6*(i-1);
f(i)=8*(i-1);
end
```

```
for i=2:M+1
for j=1:i-1
F(i)=F(i) + f(j)*p(j);
end
F(i)=F(i) + f(i)*q(i);
o(i)=K + c(i);
end
for i=1:M+1
for j=1:M+2-i
r(i,j)=F(i+j-1)-o(j)-h(i+j-1);
end
end
for i=1:M+1
for j=1:M+1
if j>i
P(i,j)=0;
elseif (i>=j && j>1)
P(i,j)=p(i-j+1);
elseif j==1
P(i,j)=q(i+j-1);
end
end
end
sj=zeros(prom,d);
for i=1:prom %Crea la muestra de estados
sj(i,:)=randint(1,d,[1,M+1]);
end
deno=zeros(prom,M+1);
for y=1:prom
for i=1:M+1
for t=1:d
c=sj(y,t);
deno(y,i)= deno(y,i)+ P(i,c);
end
end
end
pd=zeros(M+1,M+1,prom,d);
for s=1:prom
for i=1:M+1
for j=1:M+2-i
for k=1:d
c=sj(s,k);
if (deno(s,i+j-1)==0)
pd(i,j,s,k)=0;
else
pd(i,j,s,k)=P(i+j-1,c)/deno(s,i+j-1);
```

```
end
end
end
end
end
for s=1: prom
u1=zeros(M+1,M+1,N);
for t=2:N
y=zeros(1,M+1);
for i=1:M+1
y(i)=max(u1(i,:,t-1));
end
ap=zeros(M+1,M+1,prom);
for i=1:M+1
for j=1:M+2-i
for k=1:d
c=sj(s,k);
ap(i,j,s)= ap(i,j,s) + pd(i,j,s,k)*y(c);
end
end
end
for i=1:M+1
for j=1:M+2-i
u1(i,j,t)= r(i,j) + ap(i,j,s);
end
end
end
up(:,:,s)=u1(:,:,N);
end
VA=zeros(M+1,M+1);
for i=1:M+1
for j=1:M+1
VA(i,j)=(1/prom)*sum(up(i,j,:));
end
end
U=zeros(1,M+1);
for i=1:M+1
U(i)=max(VA(i,:));
end
U
toc
```

Bibliografía

[1] Ash R. B, Doléans-Dade C. A, Probability and Measure Theory. Academic Press Elsevier (1999).

[2] Bellman R, Dynamic Programming. Dover (2003).

[3] Bellman R, Dreyfus S, Applied Dynamic Programming. Princeton University Press (1962).

[4] Bertsekas D, Dynamic Programming: Deterministic and Stochastic Models. Prentice-Hall, Inc. (1987).

[5] Chang H. S, Fu M. C, Hu J, Marcus S. I, Simulation-based Algorithms for Markov Decision Processes. Springer. (2007).

[6] Feinberg E. A, Shwartz A, Handbook of Markov Decision Processes: Methods and Applications. International Series in Operations Research & Management Science (2000).

[7] Hernández-Lerma O, Lasserre J. B, Discrete-Time Markov Control Processes Basic Optimality Criteria. Springer (1996).

[8] Kroese D. P, Taimre T, Botev Z. I, Handbook of Monte Carlo Methods. Wiley Series in Probability and Statistics (2011).

[9] Powell, W, Approximate Dynamic Programming: Solving the Curses of Dimensionality. Wiley Series in Probability and Statistics (2007).

[10] Puterman M. L, Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley, New York (1994).

[11] Rust J, Using Randomization to Break the Curse of Dimensionality. Econometria, Vol. 65, No. 3 (May, 1997), 487-516.

[12] Stokey N. L, Lucas R. E., Jr. with Prescott E. C, Recursive Methods in Economic Dynamics. Harvard University Press (1989).

[13] Wang Y, Boyd S, Approximate Dynamic Programming via Iterated Bellman Inequalities.

[14] Zacarías Espinoza G, Procesos de Decisión de Markov Descontados. Tesis Licenciatura, BUAP (2007).

[15] Zacarías Espinoza G, Cruz Suárez H, Venegas Pérez L, Control óptimo de dos máquinas usando políticas de reemplazo, Novena Conferencia Iberoamericana en Sistemas, Cibernética e Informática CISCI 2010, Orlando, Florida EE.UU., del 29 de Junio al 2 de Julio, Volumen 3, pp. 159-163 (2010).