Benemérita Universidad Autónoma de Puebla

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS
LICENCIATURA EN MATEMÁTICAS APLICADAS

TESIS
PARA OBTENER EL GRADO DE
LICENCIADO EN MATEMÁTICAS APLICADAS

Un acercamiento a problemas de paro óptimo vía programación dinámica

Erika Patricia Domínguez Rios

Asesor Hugo Adán Cruz Suárez

2021

A mis padres.

Introducción

Los procesos de decisión o control Markovianos [2, 7, 17, 20, 21] modelan sistemas dinámicos estocásticos, es decir, sistemas cuya evolución está sujeta a factores aleatorios y que puede modificarse por medio de la selección de ciertas variables de decisión o de control, dichos procesos son muy útiles en áreas como Economía [6, 12, 15], Ingeniería [24] o Biología [25], entre otras. El problema de control óptimo consiste en encontrar una política que optimice al criterio de rendimiento, el cual, en este trabajo de tesis será el criterio de costo total esperado, una política es una regla mediante la cual se elige una acción en cada estado de observación del proceso.

Una metodología para resolver un proceso de control de Markov es basado en el principio de optimalidad de Bellman conocido como programación dinámica. Bellman [3] usa el nombre "proceso de decisión de markov" para referirse a un programa dinámico cuyo mecanismo de evolución es aleatorio y satisface una propiedad markoviana o de pérdida de memoria.

El propósito del presente trabajo es dar una introducción a esta clase de procesos y aportar experimentos numéricos para una mejor comprensión de cada problema presentado. En el desarrollo de la tesis se presenta y valida la ecuación de programación dinámica para problemas de control con el criterio de costo total esperado tanta para horizonte finito e infinito, considerando espacio de estados numerables y espacio de acciones finitos. Los problemas de control óptimo con costo total esperado han sido de gran uso desde el año de 1953 con Sharpley [cita], quien analizó este tipo de problemas en el contexto de juegos estocásticos. Posteriormente, existen formulaciones del problema de control vía programación lineal las cual fueron abordadas por D´Epenoux [cita] y De Ghellinck y Eppen[cita]. Recientemente, esta clase de problemas han sido abordados bajo restricciones en el Proceso de Decisión de Markov en las siguientes referencias: poner citas enunciadas al final.

En el primer capítulo se presentan conceptos básicos de procesos estocásticos y un problema de inventario, en el segundo capítulo se da una breve introducción a la

programación dinámica y con ello se prueba uno de los principales teoremas de la teoría, conocido como el teorema de la programación dinámica, para posteriormente extender el problema de inventario [23] del primer capítulo y ejemplificar el teorema probado.

En el tercer y último capítulo se exponen dos problemas de paro óptimo para los cuales se demuestra la validez de la ecuación de programación dinámica, con la cual se resuelve el desarrollo de los problemas. Asimismo, se presentan los pseudocódigos de cada uno de ellos y simulaciones realizadas en Python.

En los anexos se encuentra el código fuente de cada problema del que se realizó una simulación.

Índice general

Introducción v

1 Programación Dinámica **1**
1.1 Preliminares 1
1.2 Ejemplo de inventario 3

2 Teorema de la Programación Dinámica (PD): Horizonte finito **7**
2.1 Introducción 7
2.2 Teorema de PD 9
2.3 Aplicaciones 13

3 Problemas de Paro Óptimo **19**
3.1 Estudio de problemas de paro óptimo vía programación dinámica . . 19
3.2 Aceptando la mejor oferta 29
3.3 Cubriendo distintos tipos de vacantes 35

Conclusión **41**

Anexos **45**

Bibliografía **53**

Índice alfabético **55**

Programación Dinámica

1

En 1953 Richard Bellman propone la programación dinámica como un método de optimización para resolver problemas en procesos de decisión en múltiples pasos, sin embargo, actualmente es más amplia la cantidad de usos que ésta puede tener en problemas tanto prácticos como teóricos. Este capítulo es la médula del trabajo de tesis, en el se presentan dos secciones. La primera sección contiene definiciones y notaciones de uso común y recurrentes a lo largo de la tesis, en la segunda sección se encuentra un ejemplo, el cual será referido en el siguiente capítulo.

1.1 Preliminares

Para la comprensión del trabajo es necesario conocer conceptos básicos de procesos estocásticos, en este capítulo se presentan conceptos o nociones que serán de utilidad, sin suponer que han sido conocidos anteriormente, y posteriormente conceptos fundamentales de programación dinámica.

La siguiente definición fue recuperada de [14].

Definición 1.1. *(i) Un proceso estocástico es una sucesión de variables aleatorias $\{X_t : t \in T\}$ definidas en un espacio de probabilidad $(\Omega, \mathcal{F}, \mathbb{P})$, con $X \subseteq \mathbb{R}$ ($\mathbb{R}$ el conjunto de números reales), donde T es un conjunto de índices y X el rango común de las variables aleatorias (v.a.) a los cuales se les denomina espacio tiempo y espacio de estados, respectivamente.*

(ii) Un proceso de Markov se define como un proceso estocástico que cumple la siguiente propiedad: para $x_0, x_1, \ldots, x_{t-1}, x, y \in X$ y $t \geq 1$,

$$\mathbb{P}\left(X_{t+1} = y | X_0 = x_0, X_1 = x_1, \ldots, X_t = x\right) = \mathbb{P}\left(X_{t+1} = y | X_t = x\right).$$

(iii) Un proceso de Markov se dice homogéneo si las probabilidades de transición son las mismas en cada paso, es decir,

$$\mathbb{P}\left(X_{t+1} = y | X_t = x\right) = \mathbb{P}\left(X_1 = y | X_0 = x\right) \text{ para todo } t \geq 0.$$

Sea $\{X_t : t = 0, 1, 2, \dots\}$ un proceso de Markov homogéneo definido en un espacio de probabilidad $(\Omega, \mathcal{F}, \mathbb{P})$ y con espacio de estados $X \subset \mathbb{Z}$ ($\mathbb{Z}$ el conjunto de números enteros). Se supone que X_0 tiene una distribución conocida π_0, es decir,

$$\pi_0(y) = \mathbb{P}(X_0 = y),\ y \in X,$$

donde π_0 se denomina distribución inicial.

Notación 1.2. *En muchas aplicaciones de las cadenas de Markov el estado inicial es conocido, por ejemplo, suponga que $X_0 = x \in X$, en este caso $\pi_0(x) = 1$, y se dirá que la distribución de X_0 se concentra en el estado x.*

Notación 1.3. *Sean $x, y \in X$, se denota la transición en un paso de x a y del proceso estocástico por:*

$$P_{xy} = \mathbb{P}(X_1 = y | X_0 = x).$$

Definición 1.4. *Sea $c : X \longrightarrow \mathbb{R}^+$ una función de costo en un paso y $\{X_t\}$ un proceso de Markov.*

(i) La función de valor que caracteriza al costo total esperado hasta un horizonte $T \in \mathbb{Z}^+$, se define como:

$$V_T(x) := \mathbb{E}_x \left[\sum_{t=0}^{T-1} c(X_t) \right], x \in X.$$

(ii) Para $0 \leq t \leq T-1$, la función de valor truncado que caracteriza al costo total esperado hasta la etapa t es:

$$V_t(x) = \mathbb{E}_x \left[\sum_{n=0}^{t} c(X_n) \right],\ x \in X. \tag{1.1}$$

Observación 1.5. *Para $t = 0$ en (ii) de la Definición 1.4, para todo $x \in X$*

$$\begin{aligned} V_0(x) &= \mathbb{E}[c(X_0) | X_0 = x] \\ &= c(x). \end{aligned}$$

Para la demostración del próximo teorema será necesaria tener presente identidades de el valor esperado (ver Apéndice).

Teorema 1.6. *La función de valor truncado V_t en (1.1) con $1 \leq t \leq T$ satisface:*

$$V_t(x) = c(x) + \mathbb{E}_x[V_{t-1}(X_1)], x \in X, \tag{1.2}$$

con $V_0(x) = c(x)$ (ver Observación 1.5).

Demostración. Sea $x \in X$ y $t \in \{1, 2, \ldots, T\}$, fijos. Usando (ii) de la Definición 1.4 y propiedades de esperanza condicional (ver Apéndice),

$$\begin{aligned} V_t(x) &= c(x) + \mathbb{E}_x\left[\sum_{n=1}^{t} c(X_n)\right] \\ &= c(x) + \mathbb{E}\left[\mathbb{E}\left[\sum_{n=1}^{t} c(X_n)\middle| X_0 = x, X_1\right]\right] \\ &= c(x) + \sum_{y \in X} \mathbb{E}\left[\sum_{n=1}^{t} c(X_n)\middle| X_0 = x, X_1 = y\right] \mathbb{P}_x(X_1 = y) \end{aligned}$$

Usando la propiedad de Markov y un cambio de variable $k = n - 1$ se siguen las identidades:

$$\begin{aligned} V_t(x) &= c(x) + \sum_{y \in X} \mathbb{E}_y\left[\sum_{k=0}^{t-1} c(X_k)\right] \mathbb{P}_x(X_1 = y) \\ &= c(x) + \sum_{y \in X} V_{t-1}(y)\mathbb{P}_x(X_1 = y) \end{aligned} \tag{1.3}$$

Por lo tanto, usando la definición de esperanza y como x y t son arbitrarios el resultado se sigue. ■

1.2 Ejemplo de inventario

Para reafirmar lo establecido tanto en el Teorema 1.6 como en las definiciones que le preceden, a continuación se muestra un ejemplo de un problema de inventario donde se desea conocer el costo total esperado hasta cierta etapa establecida. Para facilitar el entendimiento de dicho ejemplo se presentan resultados obtenidos en Python.

Ejemplo 1.7. *Considérese X_t el stock de un inventario al tiempo t, el cual sigue la dinámica:*

$$X_{t+1} = (X_t - \xi_{t+1})^+, t = 0, 1, \ldots, \tag{1.4}$$

donde ξ_t representa la demanda en el periodo t, supóngase que la sucesión $\{\xi_t\}$ esta conformada por variables aleatorias independientes e idénticamente distribuidas (iid),

con $p_k = \mathbb{P}(\xi_t = k),\ k \in Rango(\xi_t)$. *Se considera que el almacén tiene una capacidad finita* $M \in \mathbb{Z}^+$, *de esta manera, el espacio de estados es* $X = \{0, 1, 2, \ldots, M\}$, *si la demanda excede la cantidad de producto existente, se paga una penalización y no hay pedidos pendientes.*
Se establece:

- $h \equiv$ *Costo de almacenaje por unidad.*

- $p \equiv$ *Costo de penalización (deuda de producto).*

En este caso se considera la siguiente función de costo, sin olvidar que para todo $a,\ b \in \mathbb{R},\ \text{máx}\{a,\ b\} = \frac{|a-b|+(a+b)}{2}$, *de esta manera,*

$$\begin{aligned} c(x) &= \mathbb{E}[h \cdot \text{máx}\{0, x-\xi\} + p \cdot \text{máx}\{0, -x+\xi\}] \\ &= h \cdot \mathbb{E}\left[\frac{|\xi - x| + x - \xi}{2}\right] + p \cdot \mathbb{E}\left[\frac{|x-\xi| + \xi - x}{2}\right] \\ &= \frac{h}{2} \cdot \Big[\mathbb{E}[|\xi - x|] + x - \mathbb{E}[\xi]\Big] + \frac{p}{2} \cdot \Big[\mathbb{E}[|\xi - x|] + \mathbb{E}[\xi] - x\Big] \\ &= \left(\frac{h-p}{2}\right)(x - \mathbb{E}[\xi]) + \left(\frac{h+p}{2}\right)\mathbb{E}[|\xi - x|], \end{aligned} \tag{1.5}$$

donde el operador esperanza, $\mathbb{E}$, *es con respecto a la variable aleatoria* ξ.
Se requiere el costo total esperado de la Definicion 1.4 para 6 etapas, bajo las condiciones: $M = 5\ ,\ h = 3,\ p = 5$, *por lo que* $X = \{0, 1, 2, 3, 4, 5\}$ *y la distribución de la demanda está dada en la Tabla 1.1.*

k	0	1	2	3	4	5
p_k	0	0.1	0.1	0.3	0.3	0.2

Tabla 1.1: Distribución de la demanda

Haciendo uso del Teorema 1.6 para 6 *etapas, se presenta el Pseudocódigo 1, donde el valor esperado de* ξ *es* 3.4. *Para hacer uso de la ecuación (1.5) es necesario considerar la esperanza* $\mathbb{E}[|\xi - x|]$, *la cual estará representada por* $value2$, *en el cual, cada una de sus entradas significa el valor de* x *en la diferencia, es decir, en la entrada* 2, *se tiene* $\mathbb{E}[|\xi - 2|]$ *y* $\mathbb{E}[\xi] = value$. *Debido a la Notación 1.5 se tiene* $V_0(x) = c(x),$ *para toda* $x \in X$, *por lo que se ejecutan las lineas* 22 *y* 23.

Pseudocódigo 1 Stock de un inventario (a)

```
procedure FUNTION(XI,X,x,bef)
    sum=0
    for y in X do
        a=x-y
        if 0 ≤ a ≤ 5 then
            sum =sum+ bef[y]*XI[a]
        return (sum)

procedure COST(XI,x,value,value2)
    h=3
    p=5
    C1=(h-p)/2·(x-value)
    C2=(h+p)/2·(x-value2[x])
    return(C1+C2)

N=6
X=[0,1,...,M]
XI=[0,0.1,0.1,0.3,0.3,0.2]
value=3.4
value2=[3.4,2.4,1.6,1.0,1.0,1.6]

for x in X do
    V[x]=COST(XI,x,value,value2)

for t in (1,...,N) do
    bef=copyvector(V)
    for x in X do
        V[x]=COST(XI,x,value,value2)+FUNTION(XI,X,x,bef)
    write("Num. etapa: ", t)
    print("V: ", V)
```

Implementando el pseudocódigo en Python, se obtiene la siguiente Tabla.

x	$V_0(x)$	$V_1(x)$	$V_2(x)$	$V_3(x)$	$V_4(x)$	$V_5(x)$
0	17	17	17	17	17	17
1	12	13.7	13.7	13.7	13.7	13.7
2	7.8	10.7	10.87	10.87	10.87	10.87
3	4.4	11.4	11.94	11.957	11.957	11.957
4	3.4	13.32	14.828	14.891	14.8927	14.8927
5	4.8	14.92	18	18.2478	18.2558	18.25597

Tabla 1.2

$V_t(x)$, $0 \leq t < N$, $x \in X$ representa el costo total esperado hasta la etapa t con x productos en stock, por ejemplo, si se considera que cada etapa cambia al finalizar el día, entonces $V_3(4)$ es el costo que debe pagarse al final del día tres dado que se tienen 4 productos en stock y considerando los costos de almacenaje y penalización definidos anteriormente.

Observación 1.8. *Obsérvese que el estado* 0 *corresponde a un estado absorbente, es decir, si para algún* $k \geq 1, X_k = 0$, $\mathbb{P}_x - casi\ seguramente\ (c.s)$ *entonces,* $X_n = 0$, $\mathbb{P}_x - c.s$, $\forall n \geq k$. *Este tipo de comportamiento no ocurre en la práctica, en este caso al modelo (1.4) puede ser modificado del modo siguiente:*

$$X_{t+1} = (X_t + A_t - \xi_{t+1})^+, \tag{1.6}$$

donde A_t *es una variable de control la cual representa los productos solicitados en el stock al tiempo* t. *De este modo (1.6) se denomina proceso de control de Markov para el modelo de inventarios propuestos. En lo subsecuente se presenta la teoría general.*

2 Teorema de la Programación Dinámica (PD): Horizonte finito

La programación dinámica es una técnica simple, pero poderosa, que ha demostrado ser muy útil para la resolución de problemas de decisión multietápicos. La idea fundamental que subyace en el método de la programación dinámica es el principio del invariante encajado, según el cual un problema difícil de resolver, esta encajado en una clase de subproblemas más simples de resolver, lo que permite encontrar una solución al problema original, la segunda sección está enfocada en enunciar y demostrar el Teorema de la Programación Dinámica 2.5 [13], el cual provee un algoritmo para encontrar una función de valor y una política óptima.

2.1 Introducción

Definición 2.1. *Un modelo de control de Markov se define de acuerdo a las siguientes componentes:*

$$(X, A, \{A(x) : x \in X\}, F, c) \tag{2.1}$$

donde, X es un conjunto numerable denominado el espacio de estados; A es un conjunto numerable denominado el conjunto de acciones o controles; $A(x) \subset A$ el conjunto de acciones admisibles para cada estado $x \in X$; $F : \mathbb{K} \times H \longrightarrow X$ la dinámica del sistema dada por

$$X_{t+1} = F(X_t, A_t, \xi_t),\ X_0 = x,\ t = 0, 1, 2, \ldots \tag{2.2}$$

donde $\{\xi_t\}$ es una sucesión de variables aleatorias iid $\mathbb{K} = \{(x, a) : x \in X, a \in A(x)\}$ y $H = Rango(\xi_t)$; $C : \mathbb{K} \longrightarrow \mathbb{R}^+$ representa la función de costo, la cual se supone conocida.

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

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

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

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

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 2.2. *(i). Una política aleatorizada o simplemente política es una sucesión $\pi = \{\pi_t : t = 0, 1, 2, \ldots, T-1\}$ de probabilidades de transición definidas sobre A, dada la historia del proceso $\mathbb{H}_t$ y satisface que: $\pi_t(A(x_t)|h_t) = 1$ para toda $h_t \in \mathbb{H}_t$ y $t = 0, 1, 2, \ldots, T$.*
El conjunto de posibles políticas será denotado por Π.

(ii). Una Política es determinista Markoviana si existe una secuencia $\{f_t\}$ de funciones $f_t \in \mathbb{F}$ tal que $\pi(\cdot|h_t)$ esta concentrada en $f_t(x_t) \in A(x)$ para todo $h_t \in \mathbb{H}_t$ y $t = 0, 1, \ldots,$ es decir, $\pi_t(C|h_t) = I_C[f(x_t)]$ para todo $C \in \mathcal{B}(A)$ donde $\mathbb{F} = \{f : X \longrightarrow A | \, \forall x \in X : f(x) \in A(x)\}$.

(iii). Una política es determinista estacionaria si existe una función $f \in \mathbb{F}$ tal que $\pi(\cdot|h_t)$ esta concentrado en $f(x_t) \in A(x)$ para todo $h_t \in \mathbb{H}_t$ y $t = 0, 1 \ldots$.

Definición 2.3. *Se define el costo total acumulado con horizonte finito para la política $\pi \in \Pi$ y $x \in X$, para N etapas, como:*

$$J(\pi, x) := \mathbb{E}_x^\pi \left[\sum_{t=0}^{N-1} C(X_t, A_t) + c_N(X_N) \right]. \tag{2.3}$$

Donde $\mathbb{E}_x^\pi$ representa el valor esperado cuando se utiliza la política π dado el estado inicial $X_0 = x$ y $c_N : X \longrightarrow \mathbb{R}$ la función de costo en la etapa final.

Definición 2.4. *Se define a J^* la función de valores óptimos como:*

$$J^*(x) := inf\{J(\pi, x) : \pi \in \Pi\}, \, \forall x \in X. \tag{2.4}$$

El problema de control consiste en encontrar una política $\pi^* \in \Pi$ tal que

$$J(\pi^*, x) = J^*(x), \, \forall x \in X. \tag{2.5}$$

Para la siguiente sección será necesario tener presente identidades del valor esperado (ver Apéndice).

2.2 Teorema de PD

Teorema 2.5. *Sean $J_0, J_1, \ldots, J_N$ funciones en X definidas (hacia atrás, desde t=N hasta t=0) por*

$$J_N(x) := c_N(x) \tag{2.6}$$

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

$$J_t(x) := \min_{A(x)} \left[C(x,a) + \sum_{y \in X} J_{t+1}(y) \mathbb{P}(X_{t+1} = y | X_t = x, A_t = a) \right], \tag{2.7}$$

A_t es la variable de control, para minimizar suponemos que para cada $t = 0, \ldots, N-1$, existen selectores $f_t \in \mathbb{F}$ t.q. $f_t(x) \in A(x)$ alcanza el mínimo en (2.7) $\forall x \in X$ y $t = 0, \ldots, N-1$,

$$J_t(x) = C(x, f_t(x)) + \sum_{y \in X} J_{t+1}(y) \mathbb{P}(X_{t+1} = y | X_t = x, A_t = f_t(x)). \tag{2.8}$$

Entonces, la política (determinista markoviana) $\pi^* = \{f_0, \ldots, f_{N-1}\}$ es la óptima, y la función de valor J^* es igual a J_0, i.e.

$$J^*(x) = J_0(x) = J(\pi^*, x), \ \forall x \in X. \tag{2.9}$$

Demostración. Sea $\pi = \{\pi_t\} \in \Pi$ una política arbitraria y $C_t(\pi, x)$ corresponde al costo total esperado del tiempo t al tiempo N, dado el estado $X_t = x$ en el tiempo t, i.e. si $t = 0, 1, \ldots, N-1$,

$$C_t(\pi, x) := \mathbb{E}^\pi \left[\sum_{n=t}^{N-1} C(X_n, a_n) + c_N(X_N) \middle| X_t = x \right] \tag{2.10}$$

$$C_N(\pi, x) := \mathbb{E}^\pi \left[c_N(X_N) | X_N = x \right] = c_N(x). \tag{2.11}$$

Si se usa la política π y $X_t = x$, nótese que por (2.3) y (2.10)

$$J(\pi, x) = C_0(\pi, x). \tag{2.12}$$

Afirmación 2.6. *Para todo $x \in X$ y $t = 0, \ldots, N$,*

$$C_t(\pi, x) \geq J_t(x). \tag{2.13}$$

Donde la igualdad se cumple si $\pi = \pi^*$, es decir,

$$C_t(\pi^*, x) = J_t(x). \tag{2.14}$$

La prueba de la Afirmación 2.6 se realizará por inducción hacia atrás. Con $t = N$ en (2.11) y (2.6), resulta $C_N(\pi, x) = J_N(x) = c_N(x)$.
Ahora como hipótesis de inducción, se asume que para algún $t = N-1, \ldots, 0$,

$$C_{t+1}(\pi, x) \geq J_{t+1}(x), \; x \in X. \tag{2.15}$$

De esta manera,

$$\begin{aligned}
C_t(\pi, x) &= \mathbb{E}^\pi \left[\sum_{n=t}^{N-1} C(X_n, A_n) + c_N(X_N) \middle| X_t = x \right] \\
&= \mathbb{E}^\pi \left[C(X_t, A_t) + \sum_{n=t+1}^{N-1} C(X_n, A_n) + c_N(X_N) \middle| X_t = x \right] \\
&= \mathbb{E}^\pi \left[\mathbb{E}^\pi \left[C(X_t, A_t) + \sum_{n=t+1}^{N-1} C(X_n, A_n) + c_N(X_N) \middle| X_{t+1} = y, A_t = a \right] \middle| X_t = x \right].
\end{aligned}$$

Sea $\beta = \sum_{n=t+1}^{N-1} C(X_n, A_n) + c_N(X_N)$ y resolviendo el valor esperado externo,

$$\begin{aligned}
C_t(\pi, x) &= \sum_{a \in A} \left[\mathbb{E}^\pi \left[C(X_t, A_t) + \beta \middle| X_t = x, X_{t+1} = y, A_t = a \right] \right] \cdot \pi_t(A_t = a | X_t = x) \\
&= \sum_{a \in A} \left[C(x, a) + \mathbb{E}^\pi \left[\beta \middle| X_t = x, X_{t+1} = y, A_t = a \right] \right] \cdot \pi_t(A_t = a | X_t = x) \\
&= \sum_{a \in A} \left[C(x, a) + \mathbb{E}^\pi \left[\mathbb{E}^\pi \left[\beta \middle| X_{t+1} = y \right] \middle| X_t = x, A_t = a \right] \right] \cdot \pi_t(A_t = a | X_t = x).
\end{aligned} \tag{2.16}$$

Notar que $\mathbb{E}^\pi \left[\beta \middle| X_{t+1} = y \right] = C_{t+1}(\pi, y)$, por lo que calculando el valor esperado de (2.16), se obtiene

$$C_t(\pi, x) = \sum_{a \in A} \left[C(x, a) + \sum_{y \in X} C_{t+1}(\pi, y) \mathbb{P}(X_{t+1} = y | X_t = x, A_t = a) \right] \cdot \pi_t(A_t = a | X_t = x). \tag{2.17}$$

Teniendo en cuenta la hipótesis de inducción (2.15) y (2.17) se sigue que

$$C_t(\pi, x) \geq \sum_{a \in A} \left[C(x,a) + \sum_{y \in X} J_{t+1}(y) \mathbb{P}(X_{t+1} = y | X_t = x, A_t = a) \right] \cdot \pi_t(A_t = a | X_t = x). \tag{2.18}$$

Con $\alpha = C(x,a) + \sum_{y \in X} J_{t+1}(y) \mathbb{P}(X_{t+1} = y | X_t = x, A_t = a)$, se obtienen las siguientes desigualdades

$$\begin{aligned} C_t(\pi, x) &\geq \sum_{a \in A} [\min_{A(x)} (\alpha)] \cdot \pi_t(A_t = a | X_t = x) \\ &= [\min_{A(x)} (\alpha)] \cdot \sum_{a \in A} \pi_t(A_t = a | X_t = x) \end{aligned} \tag{2.19}$$

Notar que $\sum_{a \in A} \pi_t \left(A_t = a | X_t = x \right) = 1$, lo cual conduce a

$$\begin{aligned} C_t(\pi, x) &\geq \min_{A(x)} \left[C(x,a) + \sum_{y \in X} J_{t+1}(y) \mathbb{P}(X_{t+1} = y | X_t = x, A_t = a) \right] \\ &= J_t(x). \end{aligned}$$

Por lo que queda demostrado (2.13). Para probar (2.14) se realiza el mismo procedimiento hasta (2.18) pues en la hipótesis de inducción habría de suponerse la igualdad, con lo cual $C_t(\pi^*, x)$ es igual a:

$$\sum_{a \in A} \left[C(x,a) + \sum_{y \in X} J_{t+1}(y) \mathbb{P}(X_{t+1} = y | X_t = x, A_t = a) \right] \cdot \pi_t(A_t = a | X_t = x) \tag{2.20}$$

En (2.19) se da la igualdad pues la política es π^*, utilizando el mismo α se siguen las siguientes igualdades

$$\begin{aligned} C_t(\pi^*, x) &= \sum_{a \in A} \left[\min_{A(x)} (\alpha) \right] \cdot \pi_t(A_t = a | X_t = x) \\ &= \left[\min_{A(x)} (\alpha) \right] \cdot \sum_{a \in A} \pi_t(A_t = a | X_t = x) \\ &= \min_{A(x)} \left[C(x,a) + \sum_{y \in X} J_{t+1}(y) \mathbb{P}(X_{t+1} = y | X_t = x, A_t = a) \right] \\ &= J_t(x), \end{aligned}$$

Por lo que se concluye la prueba de la Afirmación 2.6.

Recordando (2.11), $J(\pi, x) = C_0(\pi, x)$ para cualquier política π, entre ellas π^*, por lo que

$$J(\pi^*, x) = C_0(\pi^*, x), \tag{2.21}$$

además, haciendo uso de (2.14) con $t = 0,\ C_0(\pi^*, x) = J_0(x)$, así

$$J(\pi^*, x) = J_0(x). \tag{2.22}$$

Para demostrar la otra igualdad de (2.9), es necesario tener presente la definición

$$J^*(x) = \text{ínf}\{J(\pi, x) : \pi \in \Pi\},$$

con esto ocurre que, para toda $\pi,\ J^*(x) \leq J(\pi, x)$, en particular para π^* y por (2.22)

$$J^*(x) \leq J_0(x). \tag{2.23}$$

Para $t = 0$ en (2.12) y (2.13), $J(\pi, x) = C_0(\pi, x),\ C_0(\pi, x) \geq J_0(x)$ respectivamente, se obtiene

$$J(\pi, x) \geq J_0(x),$$

se sigue que $J_0(x)$ es cota inferior del conjunto $\{J(\pi, x) : \pi \in \Pi\}$, entonces por definición de J^*,

$$J^*(x) \geq J_0(x), \tag{2.24}$$

haciendo uso de (2.23) y (2.24) se llega a la conclusión de que $J^*(x) = J_0(x)$, obteniendo así las igualdades deseadas

$$J^*(x) = J_0(x) = J(\pi^*, x). \tag{2.25}$$

■

Existe una versión de este teorema para el caso de Borel la cual puede ser consultada en [5, 13], en este trabajo no se enuncia dicho caso.

2.3 Aplicaciones

Ejemplo 2.7. *Retomando el Ejemplo 1.7, la variable de acción A_t denota la cantidad pedida al principio del periodo t, en este caso la dinámica del sistema esta dado por la siguiente ecuación en diferencias:*

$$X_{t+1} = (X_t + A_t - \xi_{t+1})^+, \tag{2.26}$$

se agrega un costo de producción, b, por lo que la función de costo queda de la siguiente manera:

$$C(X_t, A_t, \xi_t) = bA_t + h\,\text{máx}\{0, X_t + A_t - \xi_{t+1}\} + p\,\text{máx}\{0, -X_t - A_t + \xi_{t+1}\} \tag{2.27}$$

Se toma la distribución de la demanda dada en el Ejemplo 1.7 , asi como la capacidad de el almacén, M=5, por lo que $X = \{0, 1, 2, 3, 4, 5\}$ y $A(x) = \{0, 1, \ldots, 5 - x\}\,\forall x \in X$, donde X representa el espacio de estados, en este ejemplo se tomará $h = 2$ y $p = 10$. Para ejemplificar el Teorema 2.5, se toma $N = 6$, el costo terminal, $c_N(x) = 0, \forall x \in X$ y $b = 5$, de esta manera, se desea calcular:

$$J_t(x) = \min_{A(x)} \left[C(x, a) + \sum_{y \in X} J_{t+1}(y) \mathbb{P}(X_{t+1} = y | X_t = x, A_t = a) \right], \; x \in X \tag{2.28}$$

para $t = \{5, 4, 3, 2, 1, 0\}$.

Siguiendo la idea del ejemplo anterior:

$$C(x, a) = b \cdot a + \left(\frac{h - p}{2}\right)(x + a - \mathbb{E}[\xi]) + \left(\frac{h + p}{2}\right)(\mathbb{E}\left[|x + a - \xi|\right]).$$

A continuación se muestra el Pseudocódigo 2 referente al ejemplo visto, este consta de dos partes, en la primera se observan las funciones a las cuales se les dará uso en la siguiente página donde se encuentra el código principal.

Pseudocódigo 2 Stock de un inventario (b)

```
procedure FUNTION(aux,XI,X,x,a)
    sum=0
    for y in X do
        xi=x+a-y
        if 0 ≤ xi ≤ M then
            sum =sum+ aux[y]*XI[xi]
        return (sum)

procedure SUMA(XI,x,a)
    sum=0
    for s in (0,...,len(XI)-1) do
        sum= sum+|x+a-s|· XI[s]
    return(sum)

procedure COST(XI,x,a,expxi,M)
    b=5
    h=2
    p=10
    C1=b· a
    C2=(h-p)/2·(x+a-expxi)
    C3=(h+p)/2·SUMA(XI,x,a)
    return(C1+C2+C3)
```

```
N=6
M=5
X=[0,1,...,M]
XI=[0,0.1,0.1,0.3,0.3,0.2]
exp=3.4
print("Tabla de costos: ")
tab1=[M+1,M+1]
for x in X do
    for  a in (0,...,M-x) do
        tab1[x,a]=COST(XI,x,a,espxi,M)
print(tab1)
J=[M+1]
tab1=[ ]
for x in X do
    C=[(M+1)-x]
    for  a in (0,...,M-x) do
        C[a]=COST(XI,x,a,espxi,M)
        posminima=argmin(C)
        J[x]=C[posminima]
print("t: ",N-1)
print("J= ",J)
for t in (N-2,...,0) do
    aux=copy(J)
    for x in X do C=[(M+1)-x]
        for a in (0,...,M) do
            C[a]=COST(XI,x,a,espxi,M)+FUNTION(aux,XI,X,x,a)
            posminima=argmin(C)
            J[x]=C[posminima]
    print("t: ",N-1)
    print("J= ",J)
```

En la Tabla 2.1 se muestran los costos generados, dependiendo del nivel de inventario actual y los productos pedidos en acciones admisibles, recuérdese que la capacidad máxima del almacén es 5*, por ejemplo, si x=3, el conjunto de acciones admisibles es* $A(x) = \{0, 1, 2\}$*, el resto de las acciones no son admitidas y están representadas por "-".*

x \ a	0	1	2	3	4	5
0	34	29	25.2	22.6	23.2	28.2
1	24	20.2	17.6	18.6	23.2	-
2	15.2	12.6	13.6	18.2	-	-
3	7.6	8.6	13.2	-	-	-
4	3.6	8.2	-	-	-	-
5	3.2	-	-	-	-	-

Tabla 2.1: Costos

En la etapa 5, se obtiene:

x	a=0	a=1	a=2	a=3	a=4	a=5	$J_5(x)$	s
0	34	29	25.2	22.6	23.6	28.2	22.6	3
1	24	20.2	17.6	18.6	23.2	-	17.6	2
2	15.2	12.6	13.6	18.2	-	-	12.6	1
3	7.6	8.6	13.2	-	-	-	7.6	0
4	3.6	8.2	-	-	-	-	3.6	0
5	3.2	-	-	-	-	-	3.2	0

Tabla 2.2: Valores para t=5.

Para $t = 4$*:*

x	a=0	a=1	a=2	a=3	a=4	a=5	$J_4(x)$	s
0	34	31.26	29.22	32.4	37.68	42.9	29.22	2
1	26.26	24.22	27.4	32.68	37.9	-	24.22	1
2	19.22	22.4	27.68	32.9	-	-	19.22	0
3	17.4	22.68	27.9	-	-	-	17.4	0
4	17.68	22.9	-	-	-	-	17.68	0
5	17.9	-	-	-	-	-	17.9	0

Tabla 2.3: Valores para t=4.

Para $t = 3$*:*

x	a=0	a=1	a=2	a=3	a=4	a=5	$J_3(x)$	s
0	34.	31.922	30.544	35.71	43.294	50.584	30.544	2
1	26.922	25.544	30.71	38.294	45.584	-	25.544	1
2	20.544	25.71	33.294	40.584	-	-	20.544	0
3	20.71	28.294	35.584	-	-	-	20.71	0
4	23.294	30.584	-	-	-	-	23.294	0
5	25.584	-	-	-	-	-	25.584	0

Tabla 2.4: Valores para t=3.

Para $t = 2$*:*

x	a=0	a=1	a=2	a=3	a=4	a=5	$J_2(x)$	s
0	34	32.054	30.809	36.372	44.552	52.536	30.809	2
1	27.054	25.809	31.372	39.552	47.536	-	25.809	1
2	20.809	26.372	34.552	42.536	-	-	20.809	0
3	21.372	29.552	37.536	-	-	-	21.372	0
4	24.552	32.536	-	-	-	-	24.552	0
5	27.536	-	-	-	-	-	27.536	0

Tabla 2.5: Valores para t=2.

Para $t = 1$*:*

x	a=0	a=1	a=2	a=3	a=4	a=5	$J_1(x)$	s
0	34	32.081	30.862	36.504	44.803	52.939	30.862	2
1	27.081	25.862	31.504	39.803	47.939	-	25.862	1
2	20.862	26.504	34.803	42.939	-	-	20.862	0
3	21.504	29.803	37.939	-	-	-	21.504	0
4	24.803	32.939	-	-	-	-	24.803	0
5	27.939	-	-	-	-	-	27.939	0

Tabla 2.6: Valores para t=1.

De esta manera obtenemos los valores hasta $t = 0$, con lo que se obtiene la Tabla 2.7

x	a=0	a=1	a=2	a=3	a=4	a=5	$J_0(x)$	s
0	34	32.086	30.872	36.531	44.854	53.02	30.872	2
1	27.086	25.872	31.531	39.854	48.02	-	25.872	1
2	20.872	26.531	34.854	43.02	-	-	20.872	0
3	21.531	29.854	38.02	-	-	-	21.531	0
4	24.854	33.02	-	-	-	-	24.854	0
5	28.02	-	-	-	-	-	28.02	0

Tabla 2.7: Valores para t=0.

Donde "s" representa lo que debe producirse.

De este modo, gracias al Teorema de PD 2.5 y haciendo uso de las tablas podemos concluir que la estrategia óptima es:

$$f_5(x) = \begin{cases} 0 & si \quad x \geq 3 \\ 3 - x & si \quad x \leq 2 \end{cases}, f_t(x) = \begin{cases} 0 & si \quad x \geq 2 \\ 2 - x & si \quad x < 2 \end{cases}, t = 0, 1, \ldots, 4,$$

$x = 0, 1, \ldots, 5$. Para obtenerla obsérvese la Tabla 2.2, con $t = 5$, dónde los valores de x iguales a $0, 1$ y 2, y los valores de s asignados a los mismos, al ser sumados indican la cantidad de stock que debe mantenerse. Es decir, cuando $x = 0, 1, 2$ deben producirse $3, 2, 1$ respectivamente, de tal modo que se mantenga en 3 el stock. Mientras para $x = 3, 4, 5$, no debe producirse nada.

Problemas de Paro Óptimo

3

En este capítulo se presentan dos problemas de paro óptimo, los cuales son resueltos vía programación dinámica, también se incluye un aporte númerico al tema, razón por la cual son aplicados aquí los conceptos antes enunciados. En la Sección 3.2 se presenta el problema de aceptar la mejor oferta, también conocido como el problema de la secretaria [10, 11]. Este es un problema conocido con una gran cantidad de variantes, que ha sido de gran interés desde los años cincuenta. Aunque su origen no es certero, la primera solución al problema estándar, el cual es objeto de estudio en esta sección, fue dada en 1961, pero en 1963, Dynkin [10] realizó una solución al problema usando Programación Dinámica. En esta sección se presenta la solución al problema, se realizan simulaciones del mismo y se presenta el pseudocódigo para realizar dichas simulaciones. En la Sección 3.3 se presenta el problema de cubrir distintos tipos de vacantes, para el cual son necesarias diversas nociones que son introducidas en la misma sección. El problema, aunque poco conocido, es útil para identificar la diversidad de maneras que hay para trabajar la Programación Dinámica, dotando de un ejemplo dónde los estados no son triviales. De igual manera en esta sección se incluyen simulaciones y pseudocódigos para realizar las mismas.

3.1 Estudio de problemas de paro óptimo vía programación dinámica

En este capítulo se asumirá un espacio de estados contable X y un espacio de acciones finito A. Se continuará con la notación $c : X \longrightarrow \mathbb{R}$ y $C : \mathbb{K} \longrightarrow \mathbb{R}$ de la Definición 1.4 y 2.1 respectivamente. De esta manera se supondrá que cuando una acción a es elegida, entonces se incurre en un costo esperado no negativo, $C(i, a) \geq 0$. El objetivo es minimizar el costo total esperado, pues esto es equivalente al rendimiento total esperado de un problema que tiene una función de recompensa $R(i, a) = [-C(i, a)]$ la cual es no positiva, si esto ocurre se dirá que se esta en el caso negativo.

Para alguna política π, sea

$$V_\pi(i) = \mathbb{E}^\pi \left[\sum_{n=0}^{\infty} C(X_n, A_n) \middle| X_0 = i \right] \tag{3.1}$$

Como $C(i,a) \geq 0,\ V_\pi(i)$ está bien definida. Además, sea

$$V(i) = \inf_{\pi \in \Pi} V_\pi(i),$$

y la política π^* será llamada óptima si

$$V_{\pi^*}(i) = V(i), i \in X.$$

Es posible que $V(i)$ sea infinito, en tal caso todas las políticas serían óptimas. De este modo, en lo subsecuente haremos uso del siguiente supuesto.

Suposición 3.1.

$$V(i) < +\infty,\ \forall\, i \in X.$$

Notación 3.2. *Si X_n denota el estado del proceso al tiempo n y a_n la acción elegida en ese momento, se aplicará en lo subsecuente la siguiente notación:*

$$\mathbb{P}\{X_{n+1} = j | X_n = i, a_n = a\} = P_{ij}(a).$$

Si el proceso está en el estado i, en el tiempo n y la acción a es elegida entonces el siguiente estado del sistema se elige de acuerdo con las probabilidades de transición $P_{ij}(a)$. A continuación se presenta la ecuación de programación dinámica para el caso de horizonte infinito.

Teorema 3.3. *Bajo la Suposición 3.1 para cada $i \in X$ se cumple la siguiente ecuación de programación dinámica:*

$$V(i) = \min_{a \in A} \left[C(i,a) + \sum_{j \in X} P_{ij}(a) V(j) \right].$$

Demostración. Sea $\pi \in \Pi$ y $i \in X$, entonces

$$\begin{aligned}
V_\pi(i) &= \mathbb{E}^\pi \left[\sum_{t=0}^{\infty} C(X_t, A_t) | X_0 = i \right] \\
&= \mathbb{E}^\pi \left[C(X_0, A_0) + \sum_{t=1}^{\infty} C(X_t, A_t) | X_0 = i \right] \\
&= \mathbb{E}^\pi \left[C(X_0, A_0) | X_0 = i \right] + \mathbb{E}^\pi \left[\sum_{t=1}^{\infty} C(X_t, A_t) | X_0 = i \right] \\
&= \sum_{a \in A} \pi_0(A_0 = a | X_0 = i) C(i,a) + \mathbb{E}^\pi \left[\sum_{t=1}^{\infty} C(X_t, A_t) | X_0 = i \right]. \quad (3.2)
\end{aligned}$$

Ahora, realizando un procedimiento análogo a la demostración de Teorema 2.5 se observa que

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^{\infty} C(X_t, A_t)|X_0 = i, \pi\right] &= \mathbb{E}\left[\sum_{t=1}^{\infty} C(X_t, A_t)|X_0 = i, A_0 = a, X_1 = y, \pi_{t-1}\right] \\ &= \sum_{a\in A} \pi_0(A_0 = a|X_0 = i) \sum_j P_{ij}(a)\mathbb{E}_y^{\pi}\left[\sum_{t=1}^{\infty} C(X_t, A_t)\right].\end{aligned} \tag{3.3}$$

Entonces (3.2) puede verse de la siguiente manera

$$V_\pi(i) = \sum_{a\in A} \pi_0(A_0 = a|X_0 = i)\left[C(i,a) + \sum_y P_{ij}(a)\mathbb{E}_y^{\pi}\left[\sum_{t=0}^{\infty} C(X_t, A_t)\right]\right], \tag{3.4}$$

como $\mathbb{E}_y^{\pi}\left[\sum_{t=0}^{\infty} C(X_t, A_t)\right] = V_\pi(y) \geq V(y)$, entonces (3.4) cumple las siguientes desigualdades

$$\begin{aligned}V_\pi(i) &\geq \sum_{a\in A} \pi_0(A_0 = a|X_0 = i)\left[C(i,a) + \sum_y P_{ij}(a)V(y)\right] \\ &\geq \sum_{a\in A} \pi_0(A_0 = a|X_0 = i) \min_{a\in A}\left[C(i,a) + \sum_{j\in X} P_{ij}(a)V(j)\right] \\ &= \min_{a\in A}\left[C(i,a) + \sum_{j\in X} P_{ij}(a)V(j)\right].\end{aligned} \tag{3.5}$$

En otras palabras (3.6) es cota inferior de $V_\pi(i)$ y $V(i)$ es la mayor de las cotas inferiores por lo cual $V(i) \geq \min_{a\in A}\left[C(i,a) + \sum_{j\in X} P_{ij}(a)V(j)\right]$.
Ahora, sea $a^* \in A$ tal que

$$C(i,a^*) + \sum_j P_{ij}(a^*)V(j) = \min_{a\in A}\left[C(i,a) + \sum_{j\in X} P_{ij}(a)V(j)\right] \tag{3.6}$$

Considere la política $\pi \in \Pi$ de tal modo que en el tiempo 0 escoja la acción a^*.
Si el estado siguiente es j, entonces considerando el proceso como uno que comienza en j, se escoge una política π^j que satisfaga que, para $\epsilon > 0$, $V_{\pi^j}(j) \leq V(j) + \epsilon$, y se define que π actúe como la política π^j.

Así,

$$\begin{aligned} V_\pi(i) &= C(i,a^*) + \mathbb{E}\left[\sum_{t=1}^{\infty} C(X_t, A_t)|X_0 = i, \pi\right] \\ &= C(i,a^*) + \sum_{j\in X} P_{ij}(a^*)V_{\pi^j}(j) \text{ por (3.4)} \\ &\leq C(i,a^*) + \sum_{j\in X} P_{ij}(a^*)(V(j)+\epsilon) \\ &= C(i,a^*) + \sum_{j\in X} P_{ij}(a^*)V(j) + \epsilon. \end{aligned}$$

Por definición de $V(i)$ y (3.6),

$$\begin{aligned} V(i) &\leq C(i,a^*) + \sum_{j\in X} P_{ij}(a^*)V(j) + \epsilon \\ &\leq \min_{a\in A}\left[C(i,a) + \sum_{j\in X} P_{ij}(a)V(j)\right] + \epsilon, \end{aligned}$$

como ϵ es arbitrario:

$$V(i) \leq \min_{a\in A}\left[C(i,a) + \sum_{j\in X} P_{ij}(a)V(j)\right] \tag{3.7}$$

Con lo cual se concluye el resultado. ■

Teorema 3.4. *Sea $f \in \mathbb{F}$ la cual satisface:*

$$C(i,f(i)) + \sum_j P_{ij}(f(i))V(j) = \min_a\left[C(i,a) + \sum_j P_{ij}(a)V(j)\right], i \geq 0.$$

Entonces

$$V_f(i) = V(i), i \geq 0.$$

Demostración. Sea $f \in \mathbb{F}$ tal que satisface:

$$V(i) = C(i,f(i)) + \sum_j P_{ij}(f(i))V(j), \tag{3.8}$$

iterando la expresión anterior se obtiene que:

$$\begin{aligned} V(i) &= C(i,f(i)) + \sum_j P_{ij}(f(i))\left[C(i,f(i)) + \sum_k P_{jk}(f(i))V(k)\right] \\ &= C(i,f(i)) + \sum_j P_{ij}(f(i))C(j,f(j)) + \sum_j\sum_k P_{ij}(f(i))P_{jk}(f(i))V(k). \end{aligned} \tag{3.9}$$

Ahora, identificando con valores esperados se tiene:

$$V(i) = \mathbb{E}_i^f\left[C(X_0, A_0)\right] + \mathbb{E}_i^f\left[C(X_1, A_1)\right] + \sum_k P_{ik}^2(f(i))V(k), \tag{3.10}$$

donde P_{ik}^2 representa la probabilidad de transición en 2 pasos de i a k, continuando las iteraciones se puede probar por inducción sobre el número de etapas

$$\mathbb{E}^f\left[\sum_{t=0}^{n-1} C(x_t, A_t)|X_0 = i\right] + \mathbb{E}^f\left[V(X_n)|X_0 = i\right] \quad = \quad V(i), \tag{3.11}$$

como $V(i) \geq 0$ (pues todos los costos son no negativos), obsérvese que

$$\mathbb{E}^f\left[V(X_n)|X_0 = i\right] \geq 0, \tag{3.12}$$

entonces

$$\mathbb{E}^f\left[\sum_{t=0}^{n-1} C(x_t, A_t)|X_0 = i\right] \leq V(i) \tag{3.13}$$

y haciendo $n \longrightarrow \infty$

$$V_f(i) \leq V(i), \tag{3.14}$$

lo cual prueba el resultado. ■

Ahora se supondrá que cuando una acción a es elegida en el estado i, entonces un costo esperado $C(i, a) \geq 0$, es incurrido.

Antes de continuar se debe introducir un poco acerca de los problemas de paro óptimo, se considera el siguiente modelo: cuando se está en el estado i es posible tomar la decisión de detener el proceso o continuar en él, si se detiene el proceso se recibe la recompensa $R(i)$ y el proceso termina, cuando la acción de continuar es elegida se paga un costo $c(i)$, entonces el siguiente estado será j con probabilidad P_{ij}. Lo anterior suponiendo que $c(i)$ y $R(i)$ son no negativos.

Bajo la convención de que el proceso va al estado ∞ cuando se elige la acción de paro, entonces, se tiene un proceso de decisión de dos acciones, donde la acción de paro es representada por el número 1 y la acción de continuar por el número 2.
Con estas consideraciones se presentan las siguientes ecuaciones correspondientes a la probabilidad de transición y los costos que ocurren según las acciones 1 y 2, dependiendo el caso. Bajo este contexto se tiene:

$$\begin{aligned} C(i,1) &= -R(i), & P_{i,j}(2) &= P_{i,j}, \\ P_{i,\infty}(1) &= 1, & P_{\infty,\infty}(a) &= 1, \\ C(i,2) &= c(i), & C(\infty, a) &= 0. \end{aligned} \tag{3.15}$$

Este proceso no encaja en el marco de la programación dinámica negativa por que la recompensa es positiva. Para transformar el proceso en uno que se ajuste a las condiciones necesarias, se asume lo siguiente:

Suposición 3.5.

a) $\hat{C} := inf\{c(i)|i \in X\} > 0$

b) $R := sup\{R(i)|i \in X\} < \infty$.

Bajo estas condiciones, considérese un problema relativo en el cual, cuando se está en el estado i, se puede elegir parar y recibir la recompensa terminal $R(i) - R$ (o equivalentemente pagar un costo $R - R(i) \geq 0$), o continuar pagando un costo $c(i)$ para avanzar hacia $j \geq 0$ con probabilidad P_{ij}.

Para cualquier política π considérese V_π y $\bar{V}_\pi$ como las funciones de costo esperado para el problema original y el problema relativo, respectivamente.

Nótese que para cualquier política π que pare con probabilidad 1, se cumple:

$$\bar{V}_\pi(i) = V_\pi(i) + R, \; i = 0, 1, 2, \ldots$$

Además éstas son las únicas políticas que es necesario considerar, debido a que, por Suposición 3.5 $a)$, cualquier política que no se detenga con probabilidad 1, satisface:

$$V_\pi(i) = \bar{V}_\pi(i) = \infty.$$

Así, cualquier política que sea óptima para el proceso original, es óptima para el proceso relativo y viceversa. Por lo cual el proceso relativo es un proceso de decisión de Markov con costos no negativos, en consecuencia, por el Teorema 3.3 y el análogo de las ecuaciones (3.15) para el problema relativo (las cuales resultan ser iguales salvo $C(i, 1) = R - R(i)$), existe una política óptima y la función de costo óptimo $\bar{V}$ satisface:

$$\bar{V}(i) = min\left[R - R(i), c(i) + \sum_j P_{ij}\bar{V}(j)\right], \; i \in X.$$

Además, la política que elige las acciones minimizadoras es óptima puesto que $V(i) = \bar{V}(i) - R(i)$, observe que,

$$V(i) = min\left[-R(i), c(i) + \sum_j P_{ij}V(j)\right], \; i \in X \tag{3.16}$$

y la política así determinada es óptima.

Sea $V_0(i) = -R$, y para $n \geq 1$,

$$V_n(i) = min\left[-R(i), c(i) + \sum_j P_{ij}V_{n-1}(j)\right] \; i \in X. \tag{3.17}$$

Recuérdese que $V_n(i)$ representa el costo total minimo esperado en el que se incurre, si se comienza en el estado i y es permitido un máximo de n etapas antes de detenerse, desde esta interpretación se puede ver que $V_n(i) \geq V_{n+1}(i) \geq V(i)$.
A continuación se probará la primera desigualdad por inducción sobre n. Para el caso $n = 0$, sea $i \in X$,

$$V_1(i) = \text{mín}\{-R(i), c(i) + \sum_j P_{ij}V_0(j)\}.$$

Si $V_1(i) = -R(i)$ se obtiene la igualdad pues $V_0(i) = -R(i)$.
Por otro lado, si $V_1 = c(i) + \sum_j P_{ij}V_0(j)$ significa que ese valor fue el mínimo, o en otras palabras $c(i) + \sum_j P_{ij}V_0(j) \leq -R(i)$, así $V_0(i) \geq V_1(i)$. Supóngase que $V_n(i) \geq V_{n+1}(i)$, entonces ocurre

$$c(i) + \sum_j P_{ij}V_n(j) \geq c(i) + \sum_j P_{ij}V_{n+1}(j) \tag{3.18}$$

y se quiere probar $V_{n+1}(i) \geq V_{n+2}(i)$, es decir,

$$\text{mín}\{-R(i), c(i) + \sum_j P_{ij}V_n(j)\} \geq \text{mín}\{-R(i), c(i) + \sum_j P_{ij}V_{n+1}(j)\}, \tag{3.19}$$

haciendo uso del siguiente resultado,

$$\text{mín}\{a, c\} \geq \text{mín}\{a, b\} \text{ si } b \leq c, \; a, b, c \in \mathbb{R}, \tag{3.20}$$

se obtiene la desigualdad deseada, la demostración de la desigualdad $V_{n+1} \geq V(i)$ es análoga.

Por las desigualdades anteriores se tiene

$$\lim_{n\to\infty} V_n(i) \geq V(i) \tag{3.21}$$

y se dice que el proceso es estable si $\lim_{n\to\infty} V_n(i) = V(i)$.

Proposición 3.6. *Bajo la suposición 3.5. Se tiene que para todo n y toda i,*

$$V_n(i) - V(i) \leq \frac{(R - \hat{C})[R - R(i)]}{(n+1)\hat{C}}.$$

Demostración. Sea $f \in \mathbb{F}$ una política óptima y sea T el tiempo aleatorio en el cual f se detiene. Además, sea f_n la política que elige la misma acción que f al tiempo $0, 1, \ldots, n-1$ pero que se detiene al tiempo n. Haciendo uso de (3.1) para la política f e igualdades del valor esperado (ver Apéndice), donde $\mathcal{S} = \sum_{n=0}^{\infty} C(X_n, A_n)$ de (3.1),

$$V(i) = V_f(i) = \mathbb{E}_i^f(\mathcal{S}|T \leq n)\mathbb{P}(T \leq n) + \mathbb{E}_i^f(\mathcal{S}|T > n)\mathbb{P}(T > n) \tag{3.22}$$

y

$$V_n(i) \leq V_{f_n}(i) = \mathbb{E}_i^f(\mathcal{S}|T \leq n)\mathbb{P}(T \leq n) + \mathbb{E}_i^{f_n}(\mathcal{S}|T > n) \cdot \mathbb{P}(T > n).$$

De esta manera

$$V_n(i) - V(i) \leq \left[\mathbb{E}_i^{f_n}(\mathcal{S}|T > n) - \mathbb{E}_i^f(\mathcal{S}|T > n)\right] \mathbb{P}(T > n) \leq (R - \hat{C})\mathbb{P}(T > n)$$

Para obtener una cota en $\mathbb{P}(T > n)$, haciendo uso de (3.22) se obtiene:

$$\begin{aligned} -R(i) \geq V(i) &\geq -R\mathbb{P}(T \leq n) + \left[-R + (n+1)\hat{C}\right] \mathbb{P}(T > n) \\ &= -R + (n+1)\hat{C}\mathbb{P}(T > n) \end{aligned}$$

entonces

$$\mathbb{P}(T > n) \leq \frac{R - R(i)}{(n+1)\hat{C}}$$

y el resultado se sigue. ■

Observación 3.7. *Por la Proposición se cumple*

$$V_n(i) - V(i) \leq \frac{(R - \hat{C})(R - R(i))}{(n+1)\hat{C}},$$

haciendo $n \longrightarrow \infty$ en ambos lados de la desigualdad, se obtiene:

$$\lim_{n\to\infty} [V_n(i) - V(i)] \leq \lim_{n\to\infty} \frac{(R - \hat{C})(R - R(i))}{(n+1)\hat{C}},$$

notar que $\frac{(R-\hat{C})(R-R(i))}{\hat{C}}$ no depende de n, entonces:

$$\begin{aligned} \lim_{n\to\infty} V_n(i) - V(i) &\leq 0 \\ \lim_{n\to\infty} V_n(i) &\leq V(i) \end{aligned}$$

y haciendo uso de (3.21) $\lim_{n\to\infty} V_n(i) = V(i)$, es decir, cuando se cumplen las condiciones $a)$ y $b)$ el proceso es estable.

Para ver que no todos los problemas de paro son estables considérese el siguiente ejemplo.

Ejemplo 3.8. *Problema de paro no estable.*

Sea el espacio de estados dado por el conjunto de todos los números enteros y sean: $c(i) \equiv 0, R(i) = i, P_{i,i+1} = \frac{1}{2} = P_{i,i-1}$. *Luego* $V_0(i) = -i$, *y puede probarse que* $V_n(i) = -i$, *sin embargo, hasta que el problema se detenga, el estado cambia de acuerdo a una caminata aleatoria simétrica, pero tal cadena de Markov es recurrente nula y por lo tanto, se sabe que, con probabilidad* 1, *cualquier estado* N *dado eventualmente será alcanzado. Por lo que, si se comienza en el estado* i, *se puede garantizar una recompensa final de* N, *usando la política de detenerse cuando el proceso entra a* N. *Como es cierto para toda* N, *se puede notar que* $V(i) = -\infty$. *Así, este problema de paro, el cual no satisface* $a)$ *y* $b)$, *no es estable.*

La siguiente teoría será utilizada en la Sección 3.3.

Sea

$$B = \left\{ i : -R(i) \leq c(i) - \sum_{j=0}^{\infty} P_{ij} R(j) \right\} = \left\{ i : R(i) \geq \sum_{j=0}^{\infty} P_{ij} R(j) - c(i) \right\}.$$

La política que se detiene la primera vez que el proceso entra a un estado en B es llamada, la política de anticipación de una etapa (one-stage look-ahead policy).

Definición 3.9. *Un conjunto de estados* B, *es cerrado si para cualesquiera* $i \in B$ *y* $j \notin B$ *se satisface que* $P_{ij} = 0$.

Se probará un resultado tal que si B es un conjunto cerrado de estados, entonces, bajo el supuesto de estabilidad, la política de una etapa es óptima.

Teorema 3.10. *Si el proceso es estable y* $P_{ij} = 0$ *para* $i \in B$, $j \notin B$, *entonces la política óptima se detiene en* i *si y solo si* $i \in B$.

Demostración. Se mostrará por inducción sobre n que $V_n(i) = -R(i)$, $\forall i \in B$ *y* $\forall n \geq 0$. Por definición, $V_0(i) = -R(i)$. Suponiendo que se cumple para $n-1$, entonces para $i \in B$,

$$
\begin{aligned}
V_n(i) &= \text{mín}\left[-R(i), c(i) + \sum_{j=0}^{\infty} P_{ij} V_{n-1}(j)\right] \\
&= \text{mín}\left[-R(i), c(i) + \sum_{j\in B} P_{ij} V_{n-1}(j)\right] \text{ (pues B es cerrado)} \\
&= \text{mín}\left[-R(i), c(i) - \sum_{j\in B} P_{ij} R(j)\right] \text{ (hip. de inducción)} \\
&= \text{mín}\left[-R(i), c(i) - \sum_{j=0}^{\infty} P_{ij} R(j)\right].
\end{aligned}
$$

Como $i \in B$, $-R(i) \leq c(i) - \sum_{j=0}^{\infty} P_{ij} R(j)$, de esta manera $V_n(i) = -R(i)$, $\forall i \in B$ y $\forall n \geq 0$. Utilizando la hipótesis de estabilidad,

$$
\begin{aligned}
V(i) &= \lim_{n\to\infty} V_n(i) \\
&= \lim_{n\to\infty} -R(i) \\
&= -R(i),
\end{aligned}
$$

se obtiene $V(i) = -R(i)$, $i \in B$. Ahora para $i \notin B$ la política que continúa por exactamente una etapa y se detiene, tiene un costo mínimo total esperado

$$
\begin{aligned}
V_1(i) &= \text{mín}\left[-R(i), c(i) + \sum_{j=0}^{\infty} P_{ij} V_0(j)\right] \\
&= \text{mín}\left[-R(i), c(i) - \sum_{j=0}^{\infty} P_{ij} R(j)\right],
\end{aligned}
$$

como $i \notin B$, se satisface

$$
c(i) - \sum_{j=0}^{\infty} P_{ij} R(j) < -R(i). \tag{3.23}
$$

Por otro lado, por (3.17), para cada j, $V_n(j) \leq -R(j)$, ya que $V_n(j)$ esta definido como un mínimo, entonces

$$
\lim_{n\to\infty} V_n(j) \leq \lim_{n\to\infty} -R(j).
$$

Dado que el proceso es estable $V(j) \leq -R(j)$, con lo cual se obtiene:

$$\begin{aligned} \sum_{j=0}^{\infty} P_{ij} V(j) &\leq -\sum_{j=0}^{\infty} P_{ij} R(j) \\ c(i) + \sum_{j=0}^{\infty} P_{ij} V(j) &\leq c(i) - \sum_{j=0}^{\infty} P_{ij} R(j), \end{aligned}$$

haciendo uso de (3.23)

$$c(i) + \sum_{j=0}^{\infty} P_{ij} V(j) < -R(i),$$

y por (3.16)

$$V(i) = c(i) + \sum_{j=0}^{\infty} P_{ij} V(j)$$

lo cual implica $V(i) < -R(i)$.

En resumen,

$$\begin{aligned} V(i) &= -R(i),\ i \in B \\ V(i) &< -R(i),\ i \notin B, \end{aligned}$$

es decir, si $i \in B$ la acción sugerida por la política es detenerse, en caso contrario la acción es continuar. ■

3.2 Aceptando la mejor oferta

Suponga que se presentan n ofertas en orden secuencial, con $n \geq 1$, donde los $n!$ órdenes son igualmente probables, la única información con la que se cuenta en cada etapa es el rango relativo de la oferta actual en comparación con las anteriores. Al momento que se presenta una oferta, se puede decidir si se rechaza, y se continúa el proceso, o es aceptada, y el proceso termina. El objetivo es maximizar la probabilidad de seleccionar la mejor oferta.

Se definen:

- i es una oferta candidato, si su valor es mayor o igual que el de las ofertas anteriores.

- $V(i) \equiv$ Probabilidad de elegir la mejor oferta en la posición i.

- $P(i) \equiv$ Probabilidad de que si la i-ésima oferta es aceptada ésta sea la mejor de todas.

- $H(i) \equiv$ Probabilidad de obtener la mejor oferta cuando la i-ésima oferta es rechazada.

Haciendo uso del Teorema (3.3) para maximizar una recompenza se obtiene:

- $V(i) = \text{máx}[P(i), H(i)]$

- $P(i) = \mathbb{P}(\text{la oferta } i\text{-ésima es la mejor de todas} | \text{ la oferta } i\text{-ésima es la mejor de las primeras } i) = \frac{\frac{1}{n}}{\frac{1}{i}} = \dfrac{i}{n}$.

- $H(i) = \frac{1}{i+1}V(i+1) + \frac{i}{i+1}H(i+1)$.

Es posible ver las igualdades anteriores, ya que, si la oferta i es rechazada, la oferta $i+1$ puede ser aceptada o rechazada.

La probabilidad de que la oferta $i+1$ sea mayor que las anteriores es $\frac{1}{i+1}$ y la probabilidad de ganar dado ese evento es, por definición, $V(i+1)$.

La probabilidad de que la oferta $i+1$ no sea mayor que sus antecesoras es

$$1 - \frac{1}{i+1} = \frac{i}{i+1}$$

pero, dado que no es mayor que las anteriores, obtenemos la mejor oferta con probabilidad cero si la aceptamos, por lo que se debe rechazar la oferta $i+1$ y la probabilidad de obtener la mejor oferta dado que fue rechazada la $i+1$ es por definición $H(i+1)$.

V y H son ecuaciones recursivas y para determinarlas hay una condición inicial, $H(n) = 0$ pues la probabilidad de obtener la mejor oferta dado que ya se han rechazado todas las ofertas es cero, con lo cual $V(n) = 1$.

Para ver la forma recursiva de H, se evalúa en la penúltima etapa:

$$\begin{aligned} H(n-1) &= \frac{1}{n} \cdot V(n) + \frac{n-1}{n} H(n) \\ &= \frac{1}{n} \\ &= \left(\frac{n-1}{n}\right)\left(\frac{1}{n-1}\right). \end{aligned}$$

Entonces,

$$\begin{aligned} V(n-1) &= \text{máx}\left\{\frac{n-1}{n}, \frac{n-1}{n}\frac{1}{n-1}\right\} \\ &= \frac{n-1}{n}\cdot \text{máx}\left\{1, \frac{1}{n-1}\right\} \\ &= \frac{n-1}{n}. \end{aligned}$$

Para $n-2$:

$$\begin{aligned} H(n-2) &= \frac{1}{n-1}\cdot V(n-1) + \frac{n-2}{n-1}\cdot H(n-1) \\ &= \left(\frac{1}{n-1}\right)\left(\frac{n-1}{n}\right) + \frac{n-2}{n-1}\left(\frac{1}{n}\right) \\ &= \left(\frac{n-2}{n}\right)\left(\frac{1}{n-2}\right) + \left(\frac{n-2}{n}\right)\left(\frac{1}{n-1}\right) \\ &= \frac{n-2}{n}\left(\frac{1}{n-2} + \frac{1}{n-1}\right). \end{aligned}$$

Lo cual implica

$$\begin{aligned} V(n-2) &= \text{máx}\left\{\frac{n-2}{n}, \frac{n-2}{n}\left(\frac{1}{n-2} + \frac{1}{n-1}\right)\right\} \\ &= \frac{n-2}{n}\cdot \text{máx}\left\{1, \frac{1}{n-2} + \frac{1}{n-1}\right\}. \end{aligned}$$

De manera general se obtiene

$$\begin{aligned} H(i) &= \frac{i}{n}\cdot \sum_{m=i}^{n-1}\frac{1}{m} \\ V(i) &= \frac{i}{n}\cdot \text{máx}\left\{\ 1, \sum_{m=i}^{n-1}\frac{1}{m}\right\}. \end{aligned}$$

Observación 3.11. *Cuando* $\sum_{m=1}^{n-1}\frac{1}{m} \geq 1$*, las ofertas son rechazadas, pues* $V(i) = H(i)$ *y cuando* $\sum_{m=1}^{n-1}\frac{1}{m} \leq 1$ *se considera la opción de aceptar la oferta. Además* $\sum_{m=i}^{n-1}\frac{1}{m}$ *es una función decreciente, siempre que* $n > 2$*,*

$$\sum_{m=1}^{n-1}\frac{1}{m} > 1 \quad y \quad \sum_{m=n-1}^{n-1}\frac{1}{m} < 1.$$

De aquí se deriva la existencia de $j < n$ *tal que:*

$$\sum_{m=j+1}^{n-1}\frac{1}{m} < 1 \leq \sum_{m=j}^{n-1}\frac{1}{m}.$$

H es una función decreciente después de pasar j por lo cual, después de este se debe aceptar la oferta con mayor o igual valor al máximo de las anteriores.

Recuérdese que el estudio se hace sobre las ofertas candidato, i.e., lo que conviene es rechazar las primeras ofertas candidato hasta la oferta j y como H decrece, se acepta la siguiente oferta candidato. Resta encontrar el momento j,

$$\begin{aligned} H(j) &= \frac{j}{n} \cdot \sum_{m=j}^{n-1} \frac{1}{m} && (3.24) \\ &\approx \frac{j}{n} \cdot \int_{j}^{n-1} \frac{1}{x} dx \\ &= \frac{j}{n} \cdot log\left(\frac{n-1}{j}\right) \\ &\approx \frac{j}{n} \cdot log\left(\frac{n}{j}\right) && (3.25) \end{aligned}$$

En (3.24) la suma se aproxima por sumas de Riemman a una integral, mientras que en (3.25) es posible hacer la aproximación ya que la función logaritmo varía muy poco al ser evaluada en valores muy cercanos.

Por otro lado, si

$$g(x) = \frac{x}{n} \cdot log\left(\frac{n}{x}\right),$$

se tiene que:

$$g'(x) = \frac{1}{n} \cdot log\left(\frac{n}{x}\right) - \frac{1}{n},$$

igualando a cero, $log\left(\frac{n}{x}\right) = 1$, entonces $x = \frac{n}{e}$.

La estrategia óptima es rechazar las primeras $\frac{n}{e}$ ofertas y después aceptar la primer oferta candidato que aparezca, la probabilidad de obtener la mejor de todas será $\frac{1}{e}$, pues $g\left(\frac{n}{e}\right) = \frac{1}{e}$.

Recuérdese que, g aproxima a H, i.e., la probabilidad de obtener la mejor oferta dado que se dejan pasar $\frac{n}{e}$ es $\frac{1}{e}$.

A continuación se muestra el Pseudocódigo 3 para realizar la simulación del problema, considerando N como el número total de ofertas presentadas.

Pseudocódigo 3 Aceptando la mejor oferta

```
N = 100
j = int(N/e)
best = 0
aux = 0
for k in (1, 100) do
    oferta[k] =randint(1, 100)
    if oferta[k] > best then
        best = oferta[k]
    if k > j and aux == 0 then
        if oferta[k] ≥ best then
            chosen = oferta[k]
            write("La oferta elegida es: ")
            write(chosen)
            aux = 1
write("La mejor oferta es: ")
write(best)
```

Realizando simulaciones en Python, se generan cien ofertas en orden aleatorio, dentro de un rango de 0 a 100, de las cuales una oferta es elegida siguiendo la estrategia anterior, sin embargo, en algunos casos, puede presentarse una oferta mas alta, después de ya haber elegido alguna otra. El número de ofertas rechazadas para este ejemplo es de $\frac{100}{e} = 36$, en la Figura 3.1 se muestra como la oferta elegida fue de 99, teniendo que la mejor oferta fue de 100.

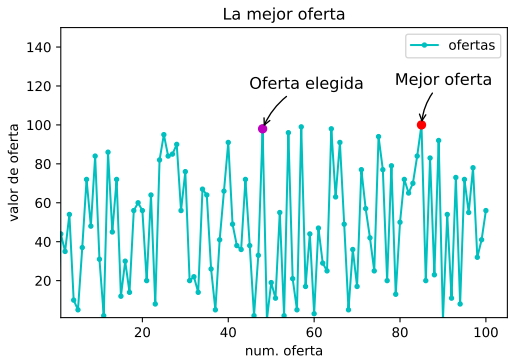

Figura 3.1

Se realizaron cincuenta simulaciones de cien ofertas cada una. En cada simulación se comparó la oferta elegida con la mejor de dicha simulación, si estas coinciden se dice que la diferencia entre ellas es cero, de otra manera, se grafica la discrepancia entre ellas y al final de esto se obtuvo que el porcentaje de coincidencia fue de $62\,\%$, lo cual se muestra en la Figura 3.2

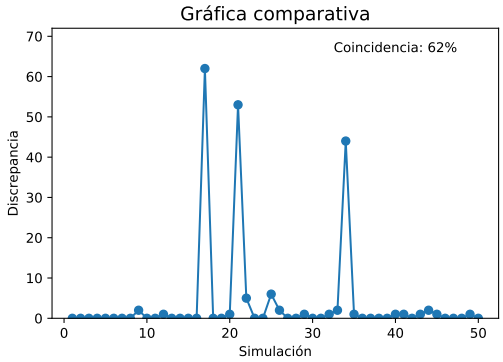

Figura 3.2

Seguido de esto, para las siguientes simulaciones, se conserva la idea anterior, donde la diferencia es el número de ofertas que son rechazadas antes de elegir la oferta candidato, en la primera gráfica de izquierda a derecha de la Figura 3.3 se rechazan las primeras 20 ofertas, donde el porcentaje de coincidencia fue de $40\,\%$, para la segunda gráfica de la Figura 3.3 se rechazan las primeras 80 ofertas y se obtiene un $26\,\%$ de coincidencia.

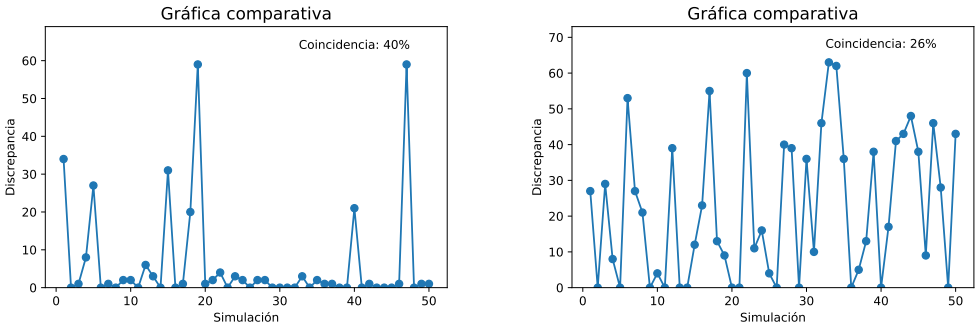

Figura 3.3

Puede observarse de las simulaciones como el porcentaje de coincidencia entre la mejor oferta y la oferta elegida de una simulación es claramente superior al seguir la política óptima, de este modo los ejemplos numéricos corroboran los resultados teóricos con los cuales fue obtenida dicha política. Más aún, la Figura 3.2 y la Figura 3.3 ponen en evidencia que la discrepancia entre oferta elegida y mejor oferta de una simulación es mucho mayor al seguir una política diferente a la óptima. Se ve de esta forma, como resolver un problema vía programación dinámica y como se verifica empíricamente.

3.3 Cubriendo distintos tipos de vacantes

Un reclutador es contratado por una empresa, donde hay N tipos de puestos disponibles, cada vez que se entrevista a un candidato, independientemente de los entrevistados con anterioridad, aplicará para el puesto i con probabilidad P_i para $i = 1, \ldots, N$. En cada etapa la empresa puede decidir pagar un costo C al reclutador, al cual solo le es posible entrevistar a lo más a un candidato en dicha etapa, o decidir no hacerlo. Si decide pagar el costo, puede llegar un candidato en busca de empleo con probabilidad α, si decide no continuar pagando al reclutador la empresa recibe la recompensa terminal $R(n)$ generada por los n tipos de puestos cubiertos. Ahora se determinará la política óptima.

El estado en cualquier momento es el conjunto S de los distintos puestos que han sido cubiertos, S_t representa el conjunto de vacantes cubiertas hasta el momento t, $t \geq 0$ y el espacio de estados sigue siendo denotado por X, donde en este caso X es el conjunto potencia de $\{1, \ldots, N\}$. La política de anticipación de una etapa se detendría en S si $S \in B$, donde $|S|$ es la cardinalidad de S (el número de elementos en S).

$$\begin{aligned} B &= \left\{S : R(|S|) \geq \mathbb{E}_{|S|}[R(S_1)] - C\right\} \\ &= \left\{S : R(|S|) \geq \alpha \sum_{i \notin S} P_i R(|S|+1) + \alpha \sum_{i \in S} P_i R(|S|) + (1-\alpha) R(|S|) - C\right\}. \end{aligned}$$

Se denota E al evento donde un candidato es entrevistado y ese puesto ya esta ocupado, es decir, ese puesto se encuentra en S y E_i donde un candidato que aplica para el puesto i es entrevistado, de esta manera:

$$\begin{aligned} \sum_{i \in S} P_i &= \mathbb{P}(\bigcup_{i \in S} E_i) \\ &= \mathbb{P}(E) \\ &= 1 - \mathbb{P}(E^c) \\ &= 1 - \mathbb{P}(\bigcup_{i \notin S} E_i) \\ &= 1 - \sum_{i \notin S} P_i, \end{aligned}$$

entonces

$$\begin{aligned} B &= \left\{S : R(|S|) \geq \alpha \sum_{i \notin S} P_i R(|S|+1) + \alpha R(|S|)\left[1 - \sum_{i \notin S} P_i\right] + R(|S|) - \alpha R(|S|) - C\right\} \\ &= \left\{S : R(|S|) \geq \alpha \sum_{i \notin S} P_i R(|S|+1) - \alpha R(|S|) \sum_{i \notin S} P_i + R(|S|) - C\right\}. \end{aligned}$$

Es posible reescribir a B como:

$$\begin{aligned} B &= \left\{ S : C \geq \Big(R(|S|+1) - R(|S|) \Big) \left[\alpha \sum_{i \notin S} P_i \right] \right\} \\ &= \left\{ S : \ R(|S|+1) - R(|S|) \leq \frac{C}{\alpha \overline{P}(S)} \right\}, \end{aligned}$$

donde $\overline{P}(S) = \sum_{i \notin S} P_i$.
Ahora, suponiendo que $R(n)$ es una función cóncava, se probará que $R(n+1) - R(n)$ decrece en n, sea $g(n) = R(n+1) - R(n)$, quiere probarse que g es decreciente, donde las siguientes desigualdades son equivalentes,

$$\begin{aligned} g(n+1) &\leq g(n) \\ R(n+2) - R(n+1) &\leq R(n+1) - R(n) \\ R(n+2) + R(n) &\leq 2R(n+1), \end{aligned}$$

por suposición R es cóncava, es decir, para todo $x,\ y \in dom(R)$ y $t \in [0,1]$,

$$R(tx + (1-t)y) \geq tR(x) + (1-t)R(y).$$

Sean $x = n,\ y = n+1$ y $t = \frac{1}{2}$,

$$\begin{aligned} R(n+1) &= R(\frac{1}{2}n + \frac{1}{2}(n+2)) \\ &\geq \frac{1}{2}R(n) + \frac{1}{2}R(n+2), \end{aligned}$$

entonces

$$R(n) + R(n+2) \leq 2R(n+1),$$

de esta manera g es decreciente.

Para demostrar que la política es óptima se hará uso del Teorema 3.10, para lo cual es necesario probar el siguiente lema.

Lema 3.12. *El proceso es estable y el conjunto B es cerrado.*

Demostración. La condición $a)$ $inf\{c(|S|) \mid S \in X\} > 0$ se cumple, ya que para $i = 1, \ldots, N, c(i) = C$, positiva, de esta manera $inf\{c(|S|) \mid S \in X\} = C$.
La condición $b)$ $sup\{R(|S|) \mid S \in X\} < \infty$ se cumple, ya que al existir una cantidad finita de recompensas, $sup\{R(|S|) \mid S \in X\} = \text{máx}\{R(|S|) \mid S \in X\}$. Así, por la Observación 3.7 el proceso es estable. Resta ver que B es un conjunto cerrado, para ello, sea $S \in B,\ S' \notin B$, se quiere probar que $P_{SS'} = 0$, suponiendo que esto no

ocurre hay que notar que $S \subseteq S'$ o $S \nsubseteq S'$. Si $S \nsubseteq S'$, entonces existe $i \in S$ tal que $i \notin S'$, en consecuencia $P_{SS'} = 0$, esto debido a que si en el conjunto S se tiene el puesto del tipo i, por la naturaleza del problema, no es posible despedir empleados, es decir, desocupar puestos, por esto no es posible trasladarse de S a S'.

Entonces $S \subseteq S'$, más aún, $S \subsetneq S'$ puesto que $S \in B$ mientras que el otro no. Dado que $S \in B$, cumple $R(|S|+1) - R(|S|) \leq \dfrac{C}{\alpha \overline{P}(S)}$, se sabe que $R(|S|+1) - R(|S|)$ decrece y por la contención $|S| < |S'|$, por consiguiente

$$R(|S'|+1) - R(|S'|) \leq R(|S|+1) - R(|S|). \tag{3.26}$$

Cabe señalar que S' tiene a lo más N elementos y haciendo uso de la contención propia S tiene a lo más $N-1$ elementos, por tanto, existe $j \notin S$ tal que $\overline{P}(S) \geq P_j$, en otras palabras, la suma de las probabilidades de entrevistar a un candidato que aplique para un puesto que no este en S será mayor igual a la probabilidad de entrevistar a un candidato que aplique para un puesto disponible. Si $\overline{P}(S') = 0$,

$$\alpha \overline{P}(S') R(|S'|+1) + \left[1 - \alpha \overline{P}(S')\right] R(|S'|) - C = R(|S'|) - C \leq R(|S'|),$$

lo cual es la condición para pertenecer a B, pero por hipótesis $S' \notin B$, así se afirma que $\overline{P}(S') \neq 0$. Recordando, $\overline{P}(S) = \sum_{i \notin S} P_i = \sum_{i \in S^c} P_i$, de igual manera para S', además $S'^c \subset S^c$, esto es una consecuencia de $S \subset S'$, de ahí que:

$$\begin{aligned}
\overline{P}(S) &= \sum_{i \notin S} P_i \\
&= \sum_{i \in S^c} P_i \\
&= \sum_{i \in S'^c} P_i + R \\
&= \sum_{i \notin S'} P_i + R \\
&= \overline{P}(S') + R
\end{aligned}$$

entonces

$$\overline{P}(S) - \overline{P}(S') = R,$$

con $R = \sum_{i \in S^c \setminus S'^c} P_i$ y $R \geq 0$ pues es suma de probabilidades, esto implica que

$$\overline{P}(S) \geq \overline{P}(S'),$$

de aqui $\overline{P}(S) \neq 0$, entonces

$$\frac{1}{\overline{P}(S)} < \frac{1}{\overline{P}(S')},$$

en síntesis

$$\frac{C}{\alpha\overline{P}(S)} < \frac{C}{\alpha\overline{P}(S')}. \tag{3.27}$$

Haciendo uso de (3.26) y (3.27) se obtiene

$$R(|S'|+1) - R(|S'|) < \frac{C}{\alpha\overline{P}(S')},$$

lo cual es una contradicción pues $S' \notin B$. En conclusión B es cerrado. ■

Por consecuencia del lema anterior, la política es óptima cuando R es cóncava.

Al igual que en la sección anterior, se presenta el Pseudocódigo 4 para realizar una simulación de este problema, usando como función cóncava a: $10 - (\frac{x}{2} - 3)^2$, considerando 50 puestos diferentes y con una probabilidad de que alguien llegue en busca de empleo de 0.7.

Pseudocódigo 4 Cubriendo distintos tipos de vacantes

```
procedure REWARD(x)
    y = 50 − (x/4 − 7)^2
    return y

procedure STOP(type,S,C,alpha,sum1)
    funtion1 =REWARD(|S|)
    funtion2 =REWARD(|S+1|)
    Pbar =suma(type)-sum1
    if (Pbar ≠ 0) then
        if (funtion2 − funtion1 ≤ C/(α·Pbar)) then
            return 1
        else
            return 0
    else
        return 1

procedure FIND(u,alpha,N)
    for k in(1,N) do
        if ((k−1)·alpha/N ≤ u < k·alpha/N) then
            return k
```

```
N = 50
alpha = 0.7
C = 0.5                                   ▷ Se define un costo para continuar
S = [ ]
sum1 = 0
i=N+1
flag = 0
for k in (1, N) do
    type[k] = 1/N

while (k < 80 and flag == 0) do
    u = random(0, 1)
    if (u ≤ alpha) then
        i=FIND(u,alpha,N)
        if (i ∉ S) then
            add i to S
            sum1 = sum1 + type[i]
            aux=STOP(type, S, C, alpha, sum1)
            if (aux == 1 and flag == 0) then
                flag = 1
                chose = |S|
                safe.k = k
                rew =REWARD(|S|)-C
                write("La cardinalidad de S es:")
                write(|S|)
                write("Num. de iteraciones:")
                write(k)
                write(Recompensa: ")
                write(REWARD(|S|)-C)
    C = C + 0.5
```

Haciendo uso del pseudocódigo anterior, se realizaron simulaciones en Python, con $N = 50$ puestos disponibles en alguna empresa, una distribución uniforme para los diferentes tipos de empleos, una probabilidad $\alpha = 0.7$ de que un candidato llegue en busca de empleo y un costo por etapa $C = 0.5$.

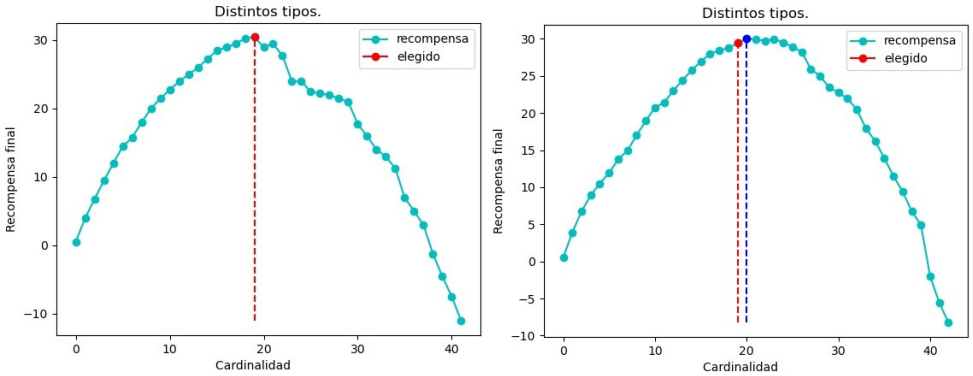

Figura 3.4

En la Figura 3.4 se muestra en color azul claro las diferentes recompensas finales obtenidas cuando la cardinalidad de S varía, pues cada que S crece, un puesto más ha sido cubierto. La empresa genera una ganancia dependiendo del número de puestos cubiertos, a ésta le restamos el costo acumulado, el cual es $k(0.5)$ donde k es el número de etapas que han transcurrido desde el inicio del proceso hasta el puesto cubierto mas reciente. La línea punteada roja, representa nuestra elección basada en la política vista previamente, la línea punteada azul indica la mayor recompensa de toda la simulación, cuando ambas lineas coinciden solo es visible la línea roja.

En estas simulaciones puede observarse como la recompensa terminal está íntimamente relacionada con la función de recompensa, obteniendo con ello una visión más clara del porqué la función de recompensa debe ser cóncava para que la estrategia sea óptima. Para realizar dichas simulaciones la probabilidad P_i fue elegida con una variable uniforme, sin embargo, en los códigos utilizados para este fin, es posible utilizar otra distribución.

Conclusión

En este trabajo de tesis se abordó la teoría de procesos de decisión de Markov, se trabajaron problemas a tiempo discreto con horizonte finito. El criterio de rendimiento utilizado para evaluar la calidad de las políticas fue el costo total esperado, este es presentado en el Capítulo 1.

La teoría de procesos de decisión de Markov fue presentada en el Capítulo 2, además se provee una técnica de solución para los problemas de control óptimo que consisten en encontrar una política que optimice el criterio de rendimiento, lo cual se ejemplificó en el problema de inventario, donde las estrategias óptimas fueron caracterizadas usando el teorema de la programación dinámica.

En el Capitulo 3, se presentaron los problemas de paro óptimo, los cuales son resueltos vía programación dinámica, asimismo se presentaron dos ejemplos de dichos problemas, en ambos ejemplos se mostró la política óptima y los cálculos para obtenerla, posteriormente se realizó un pseudocódigo para cada ejemplo a partir de su política correspondiente, haciendo uso de éste se realizaron simulaciones para mostrar de manera numérica los resultados. Los pseudocódigos se implementaron en Python lo cual permite resolver los problemas para un número mayor de estados.

Algunos problemas cuyo análisis posterior puede ser realizado, a consecuencia de este trabajo son los siguientes:

i. Resolver problemas de manera numérica que cumplan las condiciones de alguno de los expuestos en esta tesis, únicamente cambiando los parámetros del código correspondiente.

ii. Plantear y resolver el ejemplo de vacantes para un número mayor de acciones, estados y periodos.

iii. Modelar mediante procesos de decisión de Markov problemas de aplicación cotidiana.

Apéndice

Para una mejor comprensión del texto son necesarias algunas identidades aquí enunciadas. Si $\mathcal{F}_1, \mathcal{F}_2$ son $\sigma - álgebras$ y $\mathcal{F}_1 \subseteq \mathcal{F}_2$:

$$\mathbb{E}\left[\mathbb{E}[X|\mathcal{F}_1]|\mathcal{F}_2\right] = \mathbb{E}\left[\mathbb{E}[X|\mathcal{F}_2]|\mathcal{F}_1\right] = \mathbb{E}[X|\mathcal{F}_1]. \tag{3.28}$$

Sean $X,\ Y$ variables aleatorias, entonces

$$\mathbb{E}[X] = \mathbb{E}[\mathbb{E}[X|Y]], \tag{3.29}$$

cuando Y es discreta la igualdad anterior se puede expresar como:

$$\mathbb{E}[X] \sum_y \mathbb{E}[X|Y = y]\mathbb{P}(Y = y). \tag{3.30}$$

Nótese que (3.29) es un caso particular de (3.28) al tomar las σ-álgebras $\mathcal{F}_1 = \{\Omega, \varnothing\}$ y $\mathcal{F}_2 = \sigma(Y)$.

Las identidades fueron recuperadas de [4] y [18].

Anexos

Anexo 1

```
import numpy as np
from random import random

def matriz(M,V,x):
    for i in np.arange(0,6,1):
        M[i,x]=V[i,0]
    return M

def Costo(XI,x,espxi,exp2):
    h=3
    p=5

    C1=((h-p)/2)*(x-espxi)
    C2=((h+p)/2)*exp2[x,0]
    return (C1+C2)

def Sumatoria(aux,XI,X,x):

    sumatoria=0
    for y in X:
        a=x-y
        if(0<=a<=(len(XI)-1)):
            sumatoria =sumatoria+ aux[y,0]*XI[a,0]
    return sumatoria

M=5
X=np.arange(0,M+1,1) #espacio de estados
print("El espacio de estados es:")
print(X)
```

```
N=int(input("N?"))#numero de etapas

XI=np.array([[0],[0.1],[0.1],[0.3],[0.3],[0.2]])
espxi=3.4
exp2=np.array([[3.4],[2.4],[1.6],[1.0],[1.0],[1.6]])

V=np.zeros((M+1,1)) #representa V_0
for x in X:
    V[x,0]=Costo(XI,x,espxi,exp2)

aux=np.zeros((M+1,1))
M=np.zeros((M+1,N))
M=matriz(M,V,0)

for t in np.arange(1,N,1):
     for x in X:
        V[x,0]=(Sumatoria(aux,XI,X,x)+Costo(XI,x,espxi,exp2))
    M=matriz(M,V,t)#organiza los V_t en una matriz
    print("V␣es:␣")
    print(V)
print(M)
```

Anexo 2

```
import numpy as np
from random import randint
import math
N=100#de ofertas
ele=0
best=0
ban=0

j=int(N/math.exp(1))
pre=np.zeros((1,j))
post=np.zeros((1,N-j))

#hay 100 ofertas pues arange llega hasta N-1
for k in np.arange(0,N,1):

    if(k<j):
        #se genera la oferta i de 0 a 100
        pre[0,k]=randint(1,100)
    else:
        post[0,k-j]=randint(1,100)

#determina la mayor oferta presentada
if(int(np.amax(pre))>int(np.amax(post))):
    best=int(np.amax(pre))
else:
    best=int(np.amax(post))

#para determinar que oferta que se elige
#solo es necesario recorrer post,
#pues la mayor oferta de pre es la oferta
#candidato hasta ese momento
bigpre=int(np.amax(pre))
print(bigpre)
elegida=0
for k in np.arange(0,N-j,1):
    if(k==(N-j-1)):
        elegida=post[0,k]

    if(post[0,k]>=bigpre):
```

```
        elegida=post[0,k]
        break

print("pre=",pre)
print("post= ",post)
print("the best= ",best)
print("elegida= ",elegida)
```

Anexo 3

```
import numpy as np
import random
import math
import matplotlib.pyplot as plt

def recompensa(x):
    y=((x/4)-7)**2
    return (50-y)

def paro(pi,S,alpha,sumS):
    C=0.5#C es fijo

    fun1=recompensa(len(S))
    fun2=recompensa(len(S)+1)
    Pbarra=np.sum(pi)-sumS
    num=alpha*Pbarra
    beta=fun2-fun1

    if(Pbarra!=0):
        if (beta<=(C/num)):
            return 1#debe de pararse
        else:
            return 0
    else:
        return 1# pues si la suma es cero debe detenerse

def buscar(u,alpha,N):
    for k in np.arange(1,N+1):
        if((u>=((k-1)*alpha)/N)and (u<(k*alpha/N))):
            return k

N=50
X=np.arange(1,N+1,1)
print("ESPACIO DE ESTADOS\n")
print(X)#espacio de estados


C=0
alpha=0.7
```

```
print("alpha=␣")
print(alpha)

pi=np.zeros((1,N+1))#hacemos 31 espacios para dejar el primero vacio
#se tienen tipos del 1 hasta N
#la primera entrada es con k=0

for k in np.arange(1,N+1):
    pi[0,k] = 1/N

S=[]
rew=[]#servira para graficar
rew.append(recompensa(len(S))-0.5)
#la recompensa cuando la cardinalidad es cero

aux=0
ban=0
#esta suma acumulara cada probabilidad
#del tipo de empleo que fue cubierto
sumS=0
i=N+1
elegido=0
safek=0
tope=101
ganancia=0

k=0
while(k<tope):
    C=C+0.5
    u=random.random()

    #simulacion bernoulli
    if(u<=alpha):
        i=buscar(u,alpha,N)

        if(S.count(i)==0):
            S.append(i)
            sumS=sumS+pi[0,i]
            aux=paro(pi,S,alpha,sumS)

            if(aux==1 and ban==0):
                ban=1
```

```
                    elegido=len(S)
                    print(S)
                    sakek=k
                    recom=recompensa(len(S))-C

                rew.append(recompensa(len(S))-C)

    k=k+1

print("elegido= ",elegido)
print("recom: ",recom)
print("rew18: ",rew[18])
print("rew19: ",rew[19])
print("rew20: ",rew[20])
ejeX=np.arange(0,len(S)+1,1)

plt.plot(ejeX,rew,'c-o',label='recompensa')
plt.plot(elegido,recom,'r-o',label='elegido')
plt.plot([elegido,elegido],[np.min(rew),recom],'r--')

if(np.max(rew)>recom):
    plt.plot(rew.index(np.max(rew)),np.max(rew),'b-o')
    plt.plot([rew.index(np.max(rew)),rew.index(np.max(rew))],
    [np.min(rew),np.max(rew)],'b--')

plt.ylabel('Recompensa final')
plt.xlabel('Cardinalidad de S')
plt.title("Simple Plot")
plt.legend()
plt.show()
```

Bibliografía

[1] Altman, E., Boularouk, S., & Josselin, D. (2019, March). Constrained Markov decision processes with total expected cost criteria. In Proceedings of the 12th EAI International Conference on Performance Evaluation Methodologies and Tools (pp. 191-192).

[2] Bather, J. (2000). Decision Theory: An Introduction to Dynamic Programming and Sequential Decisions, John Wiley & Sons, Inc., New York.

[3] Bellman, R. (1957). A markovian decision process, Journal of Mathematics and Mechanics, 679–684.

[4] Caballero, M. E., Hernández, N. S., Rivero, V. M., Bravo, G., & Velarde, C. (2017). Cadenas de Markov Un enfoque elemental. Gráficos digitales avanzados.

[5] Chávez, S. (2010). Estudio sobre un Problema de Reemplazamiento de Máquinas vía Procesos de Decisión de Markov y Teoría de Juegos, Benemérita universidad Autónoma de Puebla, Puebla, Mexico.

[6] Cruz, H., Montes-de-Oca, R., & Zacarías, G. (2011). A consumption-investment problem modelled as a discounted markov decision process. Kybenetika, 47(6), 909–929.

[7] DeNardo, E. (2003). Dynamic Programming: Models and Applications (Illustrated ed.). Dover Publications.

[8] Dufour, F., Horiguchi, M., & Piunovskiy, A. B. (2012). The expected total cost criterion for Markov decision processes under constraints: a convex analytic approach. Advances in Applied Probability, 44(3), 774-793.

[9] Dufour, F., & Piunovskiy, A. B. (2013). The expected total cost criterion for Markov decision processes under constraints. Advances in Applied Probability,

45(3), 837-859.

[10] Freeman, P. (1983). The Secretary Problem and Its Extensions: A Review, International Statistical Review,(51)2, 189-206.

[11] García, E. (2016). Estudio de variantes del problema de la Secretaria, Universidad de Chile - Facultad de Ciencias Físicas y Matemáticas, Santiago, Chile. http://repositorio.uchile.cl/handle/2250/137963

[12] Heer, B., & Maussner, A. (2011). Dynamic General Equilibrium Modeling: Computational Methods and Applications (2nd 2009. Corr. 4th Printing 2009 ed.). Springer.

[13] Hernández, O. (1996). Discrete-time Markov control processes: basic optimality criteria, Springer Science & Business Media, New York.

[14] Hoel, P., Port, S., Stone, C. (1987). Introduction to Stochastic Processes, Waveland Press, Inc.

[15] Jaśkiewicz, A. y Nowak, A. S. (2011). Discounted dynamic programming with unbounded returns: application to economic models, Journal of Mathematical Analysis and Applications, 450-462.

[16] Kallenberg, L. C. (1980). Linear Programming and Finite Markovian Control Problems. na.

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

[18] Rincon, L. (2006). Curso Intermedio de Probabilidad. Plaza Prometeo, [Online]. Available: www.matematicas.unam.mx/lars.

[19] Rincon, L. (2015). Introducción a la Probabilidad. Prensas de Ciencias, México, D.F.

[20] Ross, S. (1992). Applied Probability Models with Optimization Applications (Reprint ed.). Dover Publications.

[21] Ross, S. (1983). Introduction to Stochastic Dynamic Programming, Academic Press, New York.

[22] Ross, S. (1996). Stochastic Processes, John Wiley, New York.

[23] Salgado, G. D., Cruz, H., Zacarías, J. & Velasco, F. (2017). Un Modelo de Inventarios con Demanda Estocástica. Memorias de la 10a. Semana Internacional de la Estadística y la Probabilidad. FCFM-BUAP, Puebla, México.

[24] Sheskin, T. J. (2019). Markov Chains and Decision Processes for Engineers and Managers (1.a ed.). CRC Press.

[25] Wilkinson, J. (2010). Parameter inference for stochastic kinetic models of bacterial gene regulation: a Bayesian approach to systems biology. Oxford University Press. Newcastle University, UK