



"El saber de mis hijos  
hará mi grandeza"

---

---

# UNIVERSIDAD DE SONORA

DIVISIÓN DE CIENCIAS EXACTAS Y NATURALES

Programa de Posgrado en Matemáticas

Control Óptimo de Sistemas de Inventarios  
con Costo Promedio

## T E S I S

Que para obtener el grado académico de:

**Maestro en Ciencias**  
(Matemáticas)

Presenta:

Joaquín Humberto López Borbón

Director de Tesis: Dr. Oscar Vega Amaya

Hermosillo, Sonora, México,      Mayo del 2007



## SINODALES

Dr. Onésimo Hernández Lerma  
Departamento de Matemáticas, CINVESTAV – I. P. N.

Dr. Fernando Luque Vázquez  
Departamento de Matemáticas, Universidad de Sonora

Dr. Luis Felipe Romero Dessens  
Departamento de Ingeniería Industrial y de Sistemas, Universidad de Sonora

Dr. Oscar Vega Amaya  
Departamento de Matemáticas, Universidad de Sonora



# Contenido

<b>Introducción</b>	<b>vii</b>
<b>1 Modelos de inventarios</b>	<b>1</b>
1.1 Introducción . . . . .	1
1.2 Justificación . . . . .	1
1.3 Componentes principales y tipos de inventarios . . . . .	2
<b>2 Sistemas de inventarios con costo promedio</b>	<b>5</b>
2.1 Introducción . . . . .	5
2.2 Problema de control óptimo de inventarios. . . . .	5
2.3 Estructura de renovación de las políticas $(s, S)$ . . . . .	11
2.4 Propiedades del costo promedio esperado y cotas para $s^*$ y $S^*$ . . . . .	16
<b>3 Algoritmos e implementación computacional</b>	<b>29</b>
3.1 Introducción . . . . .	29
3.2 Algoritmos . . . . .	29
3.3 Complejidad computacional . . . . .	33
3.4 Ejemplos numéricos y sus implementaciones . . . . .	35
3.5 Simulación e implementación . . . . .	42
<b>4 Conclusiones y perspectivas</b>	<b>51</b>



# Introducción

Un inventario de acuerdo a Love [7], es una cantidad de bienes o materiales bajo el control de una empresa que se mantienen por un tiempo en forma improductiva esperando su venta o uso.

La razón fundamental para el control de inventarios se debe a que es poco frecuente que los bienes que se tienen sean justamente los que se demandan. No tener los materiales ni los suministros cuando se necesitan representa pérdidas económicas en el proceso productivo o perder al cliente. Por otra parte, si se tiene en abundancia para protegerse de faltantes, la inversión puede resultar muy grande por tener mucho capital paralizado. Otros motivos para mantener inventarios son: economía de escala, especulación y precaución ([6], [8]). Lo anterior justifica la elaboración de modelos matemáticos con el objeto de minimizar los costos de operación de los inventarios, sujetos a la restricción de satisfacer la demanda y que además den respuestas a las preguntas claves que se requieren para el control óptimo del inventario: ¿Cuándo se ordena? y ¿Cuánto se ordena?

Lee y Nahmias [6] reportan que los primeros trabajos sobre modelos de inventarios se deben a Ford Harris (1913) al crear el modelo *EOQ* (Economic Order Quantity). Más tarde los estudios de R.H. Wilson (1934) causaron interés en círculos académicos. Los artículos destacados fueron publicados al inicio de los 50's por Arrow, Harris & Marshak (1951) y Dvoretzky, Kiefer & Wolfowitz (1952) los cuales pusieron las bases para el posterior desarrollo de la teoría matemática de inventarios. La monografía de Whittin (1957) fue también una importante contribución a la administración de inventarios y al pensamiento de economía clásica. De hecho, fue uno de los primeros tratamientos del modelo  $(Q, r)$  bajo incertidumbre, el cual más tarde se convirtió en la piedra angular para muchos sistemas de inventarios comerciales.

También Lee y Nahmias [6] reportan que durante los 50s Bellman, Glicksberg & Gross (1955) mostraron cómo el método de Programación Dinámica, diseñado por Bellman para describir una clase de problemas de decisión secuencial, podía ser usado para obtener resultados estructurales de una versión simple del problema de inventarios con revisión periódica y demanda estocástica. Posteriormente los modelos matemáticos propuestos por Arrow, Karlin & Scarf (1958) dieron el ímpetu para los siguientes trabajos en esta área. Al mismo tiempo, Wagner & Whittin (1958) encontraron una solución para el modelo *EOQ* no estacionario.

Se han hecho esfuerzos en teoría de inventarios para determinar las condiciones bajo las cuales una política en la clase  $(s, S)$  es óptima en sistemas de un solo producto. Lo anterior, es debido a que las políticas en la clase  $(s, S)$  consisten en colocar una orden para incrementar el nivel de inventario hasta el nivel de reabastecimiento  $S$  tan pronto como el nivel sea menor o igual al nivel de reorden  $s$ . Zheng [13] reporta que los primeros trabajos sobre la optimalidad de estas políticas  $(s, S)$  se deben a Scarf (1960). Iglehart (1963) estableció la existencia de una política  $(s, S)$  bajo la suposición de convexidad en la función de costo. Veinot (1966) extiende el resultado a costos cuasi convexos. Zheng [13] presenta una prueba más intuitiva para la optimalidad de políticas  $(s, S)$ . Beyer y Sethi [3] reconsideran el modelo de demanda continua y llenan algunas lagunas de la demostración de Iglehart.

El objetivo de esta tesis es desarrollar un algoritmo para encontrar una política óptima  $(s, S)$  para un sistema de inventarios con respecto al índice en costo promedio, poniendo especial énfasis en la implementación computacional. Los principales resultados que se presentan provienen de Zheng y Federgruen [12], y Federgruen y Zipkin [4].

Como punto de partida se supone la existencia de una política óptima en la clase  $(s, S)$ ; consultar [13], [3] y [5]. El sistema de inventarios es de un solo producto, en tiempo discreto, en el cual las órdenes son colocadas al inicio de cada periodo. La estructura de costos es estacionaria. Las demandas no satisfechas por no existencia son acumuladas y satisfechas en el momento que haya producto disponible. Se supone además, que las demandas son variables aleatorias independientes e idénticamente distribuidas, con valores enteros no negativos.

Para lograr nuestro objetivo, el sistema de inventarios se plantea como un problema de decisión Markoviano [2] y se aprovechan resultados de teoría de renovación. Bajo una política  $(s, S)$  el sistema de inventario se regenera cada vez que la posición del inventario se incrementa a  $S$ . Denotemos por  $T(j)$  el tiempo esperado hasta que la siguiente orden es colocada cuando se comienza con posición de inventario  $s + j$  unidades. En forma análoga, denotemos  $K(s, y)$  el costo esperado hasta que la siguiente orden es colocada, cuando se comienza con posición de inventario  $y$  unidades. Se sigue de la teoría de procesos regenerativos [9], que el costo promedio esperado  $c(s, S)$  tiene la siguiente forma

$$c(s, S) = \frac{K(s, S)}{T(S - s)},$$

inducida por una política  $(s, S)$ . Aquí obtendremos propiedades de la función  $c(s, S)$  y deduciremos cotas para  $s$  y  $S$ . Estas propiedades y cotas son fundamentales para encontrar una política óptima  $(s, S)$ .

Se implementa un algoritmo el cual usando las propiedades de la función  $c(s, S)$  va obteniendo cotas cada vez más refinadas para los niveles óptimos de reorden  $s^*$  y de reabastecimiento  $S^*$ , hasta converger a una política óptima  $(s^*, S^*)$ . El algoritmo es sencillo y fácil de implementar, su complejidad computacional es solo 2.4 veces mayor que la complejidad de evaluar una política  $(s, S)$  específica y se aplica a sistemas de inventarios con revisión periódica.



La tesis está estructurada en tres capítulos de la siguiente manera. En el Capítulo 1 se presenta la discusión de los diferentes motivos para que una empresa mantenga inventarios, así como las componentes principales y distintos tipos de inventarios. Además, se dan las características de las componentes del sistema de inventarios que se estudia en este trabajo.

En el Capítulo 2, se presentan los fundamentos teóricos de este trabajo. Se dan las hipótesis y ecuaciones de la dinámica del sistema, se definen las políticas en la clase  $(s, S)$  y se plantea el sistema de inventarios como un problema de control óptimo (PCO). Se estudia la estructura de renovación que inducen las políticas  $(s, S)$ , donde la época de renovación es cada vez que el sistema se reabastece hasta el nivel  $S$ . Se deducen propiedades de la función  $c(s, S)$  y cotas para  $s$  y  $S$ .

El Capítulo 3, contiene la parte central de esta tesis. Aquí se desarrollan los siguientes algoritmos y su implementación: el algoritmo para calcular  $T(\cdot)$ , el algoritmo para encontrar las funciones de costo  $K(\cdot, \cdot)$  y  $c(\cdot, \cdot)$  y el algoritmo principal para encontrar una política óptima  $(s, S)$ . Se hace un análisis de la complejidad computacional del algoritmo principal. Se presentan dos ejemplos y los resultados del algoritmo se comparan con los arrojados por la simulación de los sistemas. Los algoritmos y simulación son implementados usando lenguaje R [11].

Finalmente, en el Capítulo 4 se dan las conclusiones y posibles extensiones de este trabajo.



# Capítulo 1

## Modelos de inventarios

### 1.1 Introducción

En este capítulo, se presenta una definición general de inventarios, se discuten algunos motivos que justifican la elaboración de modelos de inventarios. Además, se mencionan las componentes principales de un sistema de inventarios y se especifican las componentes del sistema de inventario que se estudia en este trabajo.

### 1.2 Justificación

El control de sistemas de inventarios es una área en la cual la investigación de operaciones ha tenido un impacto significativo. La teoría matemática es la base de muchos de los sistemas de control de inventarios que se usan actualmente.

Como lo especifica Love [7], un inventario es como un sistema regulador entre los procesos, de oferta y de demanda. En términos generales, un inventario es una cantidad de bienes o materiales que una empresa mantiene por un tiempo en forma improductiva esperando su venta o uso.

La razón fundamental para el control de inventarios se debe a que es poco frecuente que los bienes sean entregados justamente cuando la demanda de ellos ocurre. No tener los materiales ni los suministros cuando se necesitan representa pérdidas económicas en el proceso productivo o perder al cliente. Por otra parte, si se tiene en abundancia para protegerse de faltantes, la inversión puede resultar muy grande y tener mucho capital paralizado. Otros motivos para mantener inventarios son los siguientes ([8], [6]):

1. **Economía de escala.** Se aprovecha el hecho de que en compras al por mayor se reducen los precios. Además, por producir u ordenar grandes lotes se reduce el número de órdenes, disminuyendo así los costos por ordenar.

2. **Especulación.** Cuando se espera que el costo de producción aumente en un futuro inmediato es ventajoso anticiparse al aumento de precio. También cuando se usan productos cuyo valor fluctúa, puede resultar ventajoso tener reserva en inventario. Igualmente, si se incrementa la demanda puede ser más conveniente mantener inventarios grandes que aumentar la capacidad de producción.
3. **Precaución.** Se mantienen inventarios como una protección contra incertidumbre, la cual puede ser motivada por demanda estocástica, incertidumbre en el abastecimiento, incertidumbre en el tiempo de entrega, costos difíciles de predecir, costos de mantenimiento de inventario sujetos a inflación, precios de productos sujetos a variación, etc.

Lo anterior, justifica la elaboración de modelos matemáticos con el objeto de minimizar los costos de operación de los inventarios, sujetos a la restricción de satisfacer la demanda y además que den respuestas a las preguntas claves que se requieren para el control óptimo del inventario: Cuándo se ordena y cuánto se ordena.

### 1.3 Componentes principales y tipos de inventarios

Los sistemas de inventarios se clasifican de acuerdo a las características de tres componentes claves, las cuales determinan la estructura esencial del modelo. Estas son: demanda, costos y aspectos físicos del sistema.

1. **Demanda.** Las suposiciones hechas sobre la demanda son las más importantes en determinar la complejidad del modelo. Usualmente se clasifican de la siguiente manera:
  - (a) Determinista y estacionaria. La demanda es constante y conocida.
  - (b) Determinista y con variación en el tiempo. La demanda es una función del tiempo y es conocida.
  - (c) Aleatoria. La demanda es aleatoria y se conoce su distribución.
  - (d) Desconocida. En este caso la distribución de la demanda es desconocida.
2. **Costos.** Como el objetivo es minimizar costos, las suposiciones acerca de la estructura de costos también son importantes en la determinación de la complejidad del modelo. Los distintos costos que se pueden considerar son los siguientes:
  - (a) Costos por adquirir o producir. El costo por adquirir o producir,  $y$  artículos está dado por la función  $C(y)$  y es más realista incluir un costo fijo  $K_f$  por ordenar, quedando la estructura de costos

$$C(y) + K_f\delta(y).$$

donde

$$\delta(y) := \begin{cases} 1 & \text{si } y > 0 \\ 0 & \text{si } y = 0. \end{cases}$$

La suposición más simple es que  $C(\cdot)$  sea lineal, pero también puede ser convexa, cóncava o de otra naturaleza.

- (b) Costos por mantenimiento. Estos son costos generados por el manejo y mantenimiento de los productos en inventario, como pérdidas por deterioro, obsolescencia, defectos, costos del espacio físico para almacenamiento, costos de promoción, etc.
- (c) Costos de penalización. Son costos causados por no tener en existencia productos para satisfacer la demanda. Estos costos también son conocidos como costos por déficit, los cuales son difíciles de estimar ya que dependen de la reacción del cliente ante la carencia. Aún cuando el cliente acepte que su pedido sea entregado posteriormente, se debe considerar un costo de penalización por no satisfacer al cliente.

La mayoría de los modelos de inventarios consideran los costos invariantes en el tiempo, pero pueden considerarse costos que varían en el tiempo.

3. **Otros aspectos físicos.** Los modelos de inventarios también son caracterizados por suposiciones hechas acerca del tiempo y logística del sistema. Algunas de estas suposiciones son:

- (a) Número de productos. Se consideran modelos con un solo producto o varios productos, en unidades o lotes según el proceso. Los productos pueden ser surtidos de un proveedor o de varios.
- (b) Tiempo de entrega. Es el tiempo que transcurre entre el momento en que se ordena un artículo o se solicita su fabricación y el momento de su entrega. La suposición más simple es que este tiempo sea cero, es decir, la entrega es inmediata. La suposición más común, es que el tiempo de entrega sea constante y fijo. El análisis es más complicado cuando este tiempo es aleatorio.
- (c) Ordenes atrasadas. Suposiciones acerca de la manera que el sistema reacciona cuando la demanda excede al suministro. Lo más simple es registrar el exceso de demanda en una orden atrasada y es representada como nivel de inventario negativo. Otro caso es cuando el exceso de demanda se pierde, muy común en venta al por menor. El otro caso es cuando el cliente espera un tiempo fijo para que su demanda sea satisfecha.
- (d) Proceso de revisión. Revisión continua significa que el inventario es conocido en cualquier tiempo y la decisión de reorden puede hacerse también en cualquier tiempo. El otro caso es revisión periódica, en el cual el nivel de inventario se conoce sólo en tiempos determinados y la decisión de reorden solo puede ser hecha en dichos tiempos los cuales se corresponden con el inicio de cada periodo.

- (e) Cambios que ocurren durante el almacenaje de inventario. Algunos productos como líquidos volátiles o materiales radiactivos decaen exponencialmente y una fracción de inventario se pierde en cada unidad de tiempo. En general la vida útil o caducidad puede también ser determinista o estocástica.
- (f) Horizonte de planeación. Es el tiempo durante el cual se considera necesario planear los inventarios. Este puede ser finito o infinito, entendiendo por infinito un tiempo sumamente largo.

En este trabajo se estudia un sistema de inventarios con revisión periódica para un solo producto que satisface lo siguiente:

1. Las órdenes del producto se hacen al inicio de cada periodo y suponemos que son entregadas instantáneamente.
2. Si se presenta un déficit del producto, este déficit se acumula hasta que se tiene inventario para satisfacer la demanda atrasada.
3. Las demandas en cada periodo toman valores en los enteros no negativos y forman una sucesión de variables aleatorias independientes.
4. La estructura de costo y los parámetros asociados son estacionarios, es decir, no cambian durante la operación del sistema.
5. Horizonte de planeación infinito.

## Capítulo 2

# Sistemas de inventarios con costo promedio

### 2.1 Introducción

La teoría de los procesos de decisión (o control) Markovianos (PDM) trata de problemas de optimización en sistemas dinámicos conocidos genéricamente como problemas de control óptimo (PCO). El estudio, en sus aspectos básicos, se desarrolla en el contexto de la programación dinámica estocástica y la teoría de procesos de Markov [2].

Uno de los resultados fundamentales en teoría de inventarios es la optimalidad de una política  $(s, S)$  en modelos de un solo producto en un horizonte de planeación infinito. En este capítulo, suponiendo conocida la existencia de una política óptima en la clase  $(s, S)$  (consultar [13], [3] y [5]), se desarrolla la teoría necesaria para encontrar una política  $(s, S)$  óptima para un sistema de inventarios en tiempo discreto con un solo producto. El sistema de inventarios se plantea como un problema de control óptimo (PCO) y se aprovechan resultados de teoría de renovación para deducir propiedades de la función de costo promedio esperado, cotas para el nivel óptimo de reorden  $s^*$  y para el nivel óptimo de reabastecimiento  $S^*$ . Dichas cotas resultan fundamentales en el método para obtener una política óptima  $(s, S)$ .

### 2.2 Problema de control óptimo de inventarios.

En esta sección, se enuncian las hipótesis y ecuaciones de la dinámica del sistema, se definen las políticas en la clase  $(s, S)$  y finalmente se plantea el sistema de inventarios como un problema de control óptimo (PCO).

#### 2.2.1 Dinámica del sistema

Considere un sistema de inventario que evoluciona de acuerdo a la siguiente ecuación recursiva

$$\begin{aligned}x_{n+1} &= x_n + a_n - w_{n+1}, & n \in \mathbb{N}_0 := \{0, 1, 2, \dots\}, \\x_0 &= x \in X,\end{aligned}\tag{2.1}$$

donde:

1.  $x_n$  es el **nivel de inventario** en el periodo  $n \in \mathbb{N}_0$  y toma valores en el **espacio de estados**  $X := \mathbb{Z}$ .

2.  $a_n$  es la **cantidad de productos ordenados** en el periodo  $n \in \mathbb{N}_0$  y toma valores en el **espacio de acciones**  $A := \mathbb{N}_0$  independiente del valor de  $x_n$ . Esto último significa que se considera producción ilimitada en el sistema. Además denotemos por  $\mathbb{K} := X \times A$  al conjunto de **pares admisibles estado-control**.
3.  $y_n := x_n + a_n$  es el **nivel de inventario a la mano** en el periodo  $n \in \mathbb{N}_0$ .
4.  $w_n$  es la **demanda** del producto durante el periodo  $n \in \mathbb{N}_0$ .

**Hiptesis 2.1** *Se supone que se cumplen las siguientes condiciones:*

1. La sucesión  $\{w_n\}$  está formada por variables aleatorias independientes e idénticamente distribuidas, con función de masa de probabilidad  $p_j = P[w_n = j]$  concentrada en  $\mathbb{N}_0$ , esto es,  $\sum_{j=0}^{\infty} p_j = 1$ .
2.  $p_0 < 1$ .
3.  $\mu := E[w_k] < \infty$ .

El **costo en cada periodo** de operación está dado por la función

$$\widehat{C}(x, a) := \begin{cases} K_f + ca + G(x + a) & \text{si } a > 0 \\ G(x) & \text{si } a = 0, \end{cases} \quad (2.2)$$

donde:

1.  $K_f \geq 0$  es el **costo fijo por colocar una orden**.
2.  $c \geq 0$  es el **costo unitario de producción**.
3. La función  $G : \mathbb{Z} \rightarrow \mathbb{R}$  incluye otros costos del sistema. En muchos casos concretos la función  $G$  tiene la forma

$$G(y) = hE_w [\max \{0, y - w\}] + peE_w [\max \{0, w - y\}],$$

donde  $h$  es el **costo unitario por manejo de inventario**,  $pe$  es el **costo unitario por déficit**,  $w$  es la demanda aleatoria,  $E_w$  es el operador esperanza,  $y$  el nivel de inventario a la mano.

**Hiptesis 2.2** *Se supone que la función  $G$  satisface las siguientes condiciones:*

1.  $-G$  es unimodal. Es decir, existe  $y_0^* \in \mathbb{R}$  tal que en  $(-\infty, y_0^*]$   $G$  no es creciente y en  $(y_0^*, \infty)$  es no decreciente.
2.  $\lim_{|y| \rightarrow \infty} G(y) \geq \min_y G(y) + K_f$ .



### 2.2.2 Políticas de control admisibles

Una **política de control admisible** es una sucesión  $\pi = \{\pi_n\}$  de reglas para elegir controles, donde cada  $\pi_n$  puede depender de la historia  $h_n$  del sistema hasta el tiempo  $n$ ,

$$h_n = (x_0, a_0, x_1, a_1, \dots, x_{n-1}, a_{n-1}, x_n).$$

Además de depender de la historia, las reglas  $\pi_n$  pueden ser aleatorizadas. En este caso tendríamos que  $\pi_n(\cdot | h_n)$  es una distribución de probabilidad sobre el espacio de controles  $A$  para cada historia  $h_n$  y  $n \in \mathbb{N}$ . A la familia de las políticas admisibles la denotaremos por  $\Pi$ .

Diremos que la política  $\pi = \{\pi_n\}$  es **estacionaria determinista** si en cada una de los periodos los controles se eligen por medio de una función  $f : X \rightarrow A$ , es decir,  $a_n = f(x_n)$  para cada  $n \in \mathbb{N}_0$ . En este caso, a la política  $\pi$  la identificaremos con la función  $f$  y a la clase de las políticas estacionarias deterministas la denotaremos por  $\mathbb{F}$ .

En este trabajo la atención está dirigida a una clase de políticas estacionarias deterministas denominadas genéricamente como políticas  $(s, S)$ , donde  $s$  y  $S$  son enteros tales que  $s < S$ . Estas políticas se denotan por  $f = (s, S)$  y se definen por

$$f(x) := \begin{cases} S - x & \text{si } x \leq s \\ 0 & \text{si } x > s. \end{cases}$$

Cuando el nivel de inventario es menor o igual a  $s$ , se coloca una orden para incrementar el nivel de inventario hasta  $S = x + f(x)$ , es decir, se ordenan  $S - x$  unidades. Si el inventario es mayor que  $s$  no se ordena. Los parámetros  $s$  y  $S$  se denominan nivel de reorden y nivel de reabastecimiento, respectivamente.

La evolución del sistema (2.1) se puede expresar equivalentemente por medio de probabilidades de transición en un paso. Para mostrar lo anterior suponga que el controlador usa la política de control  $\pi = \{\pi_n\}$  y que el sistema ha realizado las primeras  $n$  transiciones de manera que  $h_n = (x_0, a_0, \dots, x_{n-1}, a_n, x_n)$ ,  $x_n = x \in X$  y  $a_n = a \in A$ . Puesto que  $w_{n+1}$  es independiente de  $h_n$  y  $a_n$ , tenemos que

$$\begin{aligned} \Pr[x_{n+1} = y | h_n, a_n] &= \Pr[w_{n+1} = x + a - y | h_n, a_n = a] \\ &= \Pr[w_{n+1} = x + a - y | x_n = x, a_n = a] \\ &= \Pr[w_{n+1} = x + a - y] \\ &= p_{x+a-y}. \end{aligned}$$

para todo  $n \in \mathbb{N}_0$ . Definimos entonces las probabilidades de transición en un paso  $Q(y|x, a)$  de la siguiente manera:

$$\begin{aligned} Q(y|x, a) &:= \Pr[w_{n+1} = x + a - y | x_n = x, a_n = a] \\ &= p_{x+a-y} \end{aligned}$$

para todo  $x, y \in X, a \in A$ .

Una vez especificada la distribución inicial  $\nu(\cdot)$ , la probabilidad de transición en un paso  $Q(\cdot|\cdot, \cdot)$  y la política de control  $\pi = \{\pi_n\} \in \Pi$  elegida determinan la distribución del proceso controlado  $\{(x_n, a_n)\}$ . Para precisar lo anterior tome  $\Omega :=$

$\mathbb{K}^\infty$  y denote por  $\mathcal{F}$  a la familia de eventos correspondientes. Entonces, para cada política de control  $\pi = \{\pi_n\} \in \Pi$  y cada distribución  $\nu(\cdot)$  sobre  $X$  existe una única medida de probabilidad  $P_\nu^\pi$  sobre el espacio  $(\Omega, \mathcal{F})$  que satisface las siguientes propiedades. Para todo  $x \in X, a \in A, n \in \mathbb{N}$ :

- (a)  $P_\nu^\pi[x_0 = x] = \nu(x)$ ;
- (b)  $P_\nu^\pi[x_{n+1} = y | h_{n-1}, a_{n-1}, x_n = x, a_n = a] = Q(y|x, a)$ ;
- (c)  $P_\nu^\pi[a_n = a | h_n] = \pi_n(a | h_n)$ .

La operación de esperanza con respecto a la distribución de probabilidad  $P_\nu^\pi$  será indicada como  $E_\nu^\pi$ . Si la distribución inicial esta concentrada en algún estado  $x \in \mathbf{X}$ , escribiremos  $P_x^\pi$  y  $E_x^\pi$  en lugar de  $P_\nu^\pi$  y  $E_\nu^\pi$ , respectivamente.

### 2.2.3 El problema de control óptimo

Para una política  $\pi = \{\pi_n\}$  y el estado inicial  $x_0 = x \in X$  se define el **costo esperado en n-pasos** como

$$\hat{J}_n(\pi, x) := E_x^\pi \left[ \sum_{k=0}^{n-1} \hat{C}(x_k, a_k) - c(x_n - w_{n+1}) \right], \quad (2.3)$$

donde  $\hat{C}(x, a)$  es la función en (2.2) y  $E_x^\pi$  indica el operador esperanza cuando se usa la política  $\pi$  dado que el estado inicial es  $x$ .

El **costo promedio esperado** (por unidad de tiempo) es definido como

$$\hat{J}(\pi, x) := \liminf_{n \rightarrow \infty} \frac{1}{n} \hat{J}_n(\pi, x). \quad (2.4)$$

La **función de valor óptimo en costo promedio** es

$$\hat{J}(x) := \inf_{\pi \in \Pi} \hat{J}(\pi, x). \quad (2.5)$$

De esta manera, el **problema de control óptimo** consiste en encontrar una política  $\pi^* \in \Pi$  tal que

$$\hat{J}(x) = \hat{J}(\pi^*, x) \quad \forall x \in X. \quad (2.6)$$

Si tal política existe, se le llama **política óptima en costo promedio**.

El siguiente resultado es el punto de partida para el desarrollo del algoritmo.

**Teorema 2.3** *Si se satisfacen las Hipótesis 2.1 y 2.2, entonces existe una política óptima  $f^* = (s^*, S^*)$ , es decir,*

$$\widehat{J}(x) = \widehat{J}(f^*, x) \quad \forall x \in X.$$

La optimalidad de la clase de las políticas  $f = (s, S)$  se ha demostrado bajo condiciones muy generales (consultar [13], [3] y [5]). En este trabajo partimos de este resultado y nos enfocamos sobre los fundamentos del algoritmo y su implementación.

Por otra parte con el fin de simplificar el análisis, supondremos que el costo de producción  $c$  es igual a cero. Este supuesto no implica pérdida de generalidad como se muestra en el siguiente teorema y su corolario.

**Teorema 2.4** *Para toda política  $\pi = \{\pi_n\}$  y estado inicial  $x_0 = x \in X$ , se cumple lo siguiente:*

$$E_x^\pi \left[ \sum_{k=0}^{n-1} ca_k - c(x_n - w_{n+1}) \right] = -cx + (n+1)c\mu.$$

donde  $\mu := E[w_k]$

**Demostración.** Para verificar este hecho, considere el nivel de inventario a la mano  $y_k := x_k + a_k$  de modo que (2.1) se puede expresar como

$$x_{k+1} = y_k - w_{k+1} \quad k \in \mathbb{N}_0, \quad (2.7)$$

o equivalentemente

$$x_k = y_{k-1} - w_k, \quad k \in \mathbb{N}. \quad (2.8)$$

Note también que de  $y_k := x_k + a_k$  resulta  $a_k = y_k - x_k$ ,  $k \in \mathbb{N}_0$  y de (2.8)

$$a_k = y_k - y_{k-1} + w_k, \quad k \in \mathbb{N}.$$

Así que

$$\sum_{k=0}^{n-1} ca_k - c(x_n - w_{n+1}) = ca_0 + \sum_{k=1}^{n-1} ca_k - c(x_n - w_{n+1})$$

que se puede escribir como

$$\begin{aligned} \sum_{k=0}^{n-1} ca_k - c(x_n - w_{n+1}) &= c(y_0 - x_0) + \sum_{k=1}^{n-1} c(y_k - y_{k-1} + w_k) - c(x_n - w_{n+1}) \\ &= -cx_0 + c \sum_{k=1}^{n-1} w_k + cy_{n-1} - cx_n + cw_{n+1}. \end{aligned}$$

Como  $x_0 = x$  y de (2.8),  $w_n = y_{n-1} - x_n$ , entonces se tiene

$$\begin{aligned} \sum_{k=0}^{n-1} ca_k - c(x_n - w_{n+1}) &= -cx + c \sum_{k=1}^{n-1} w_k + cw_n + cw_{n+1} \\ &= -cx + c \sum_{k=1}^{n+1} w_k. \end{aligned}$$

por lo tanto,

$$E_x^\pi \left[ \sum_{k=0}^{n-1} ca_k - c(x_n - w_{n+1}) \right] = -cx + (n+1)c\mu. \blacksquare$$

**Corolario 2.5** Como consecuencia del Teorema 2.4 anterior se tiene que

$$\widehat{J}(\pi, x) = \liminf_{n \rightarrow \infty} \frac{1}{n} E_x^\pi \sum_{k=0}^{n-1} C(x_k, a_k) + c\mu \quad \forall \pi \in \Pi, \quad x \in X,$$

donde

$$C(x, a) := \begin{cases} K_f + G(x+a) & \text{si } a > 0 \\ G(x) & \text{si } a = 0. \end{cases} \quad (2.9)$$

Definiendo

$$J(\pi, x) := \liminf_{n \rightarrow \infty} \frac{1}{n} E_x^\pi \sum_{k=0}^{n-1} C(x_k, a_k), \quad x \in X, \quad \pi \in \Pi, \quad (2.10)$$

$$J(x) := \inf_{\pi \in \Pi} J(\pi, x), \quad x \in X, \quad (2.11)$$

del corolario anterior se tiene

$$\widehat{J}(\pi, x) = J(\pi, x) + \mu c, \quad \forall x \in X, \quad \pi \in \Pi, \quad (2.12)$$

$$\widehat{J}(x) = J(x) + \mu c, \quad \forall x \in X. \quad (2.13)$$

**Observacin 2.6** En conclusi3n, una pol3tica  $\pi^*$  es 3ptima para  $\widehat{J}(\cdot)$  si y s3lo si es 3ptima para  $J(\cdot)$ . Por esta raz3n, de aqu3 en adelante, nos concentraremos en el problema de control con funci3n de costo (2.9) en un periodo en la cual no se considera el costo unitario de producci3n  $c$ .

### 2.3 Estructura de renovación de las políticas $(s, S)$ .

En esta sección se estudia la estructura de renovación de la función de costo promedio esperado  $J(f, x)$  inducida por las políticas  $f = (s, S)$ , donde la época de renovación es cada vez que el sistema se reabastece hasta el nivel  $S$ . Además, para obtener los resultados de esta sección se mantendrá fija una política  $(s, S)$  que se denotara por  $f$ .

Para  $v > 0$ , se define la función

$$t(v) := \min\{n \geq 1 : x_n \leq s, x_0 = s + v\}. \quad (2.14)$$

Note que  $t(v)$  es el tiempo de espera para que el nivel de inventario sea menor o igual al punto de reorden  $s$ , cuando se comienza con  $v$  unidades arriba de  $s$ . Además, se denota el valor esperado de  $t(v)$  por

$$T(v) = E_{s+v}^f t(v). \quad (2.15)$$

Ahora, considerando,  $y := s + v$ , donde  $v > 0$  se define la función

$$k(s, y) := \sum_{k=0}^{t(v)-1} C(x_k, a_k), \quad y > s \quad (2.16)$$

la cual representa el costo acumulado hasta ordenar, cuando se comienza con un nivel de inventario  $y$  con  $v > 0$  unidades mayor que  $s$ . Es decir, son los costos que se acumulan hasta que el nivel de inventario sea menor o igual al punto de reorden  $s$ , incluyendo el costo fijo por ordenar  $K_f$ . También se denota su valor esperado

$$K(s, y) = E_y^f k(s, y) = E_y^f \sum_{k=0}^{t(v)-1} C(x_k, a_k), \quad y > s \quad (2.17)$$

Por otra parte considérese  $\widehat{K}(s, y)$  como el valor esperado del costo acumulado hasta antes de ordenar, es decir aún no se considera el costo  $K_f$ . Por lo que

$$K(s, y) := \widehat{K}(s, y) + K_f, \quad y > s \quad (2.18)$$

En los siguientes resultados se demostrará que las funciones  $T(\cdot)$  y  $K(s, \cdot)$  satisfacen ecuaciones del tipo de renovación (consultar [9]).

**Teorema 2.7**  $T(\cdot)$  y  $\widehat{K}(s, \cdot)$  son las únicas funciones que satisfacen las ecuaciones

$$T(v) = 1 + \sum_{j=0}^{v-1} T(v-j)p_{jj}, \quad v > 0 \quad (2.19)$$

$$\widehat{K}(s, y) = G(y) + \sum_{j=0}^{y-s-1} \widehat{K}(s, y-j)p_j, \quad y > s. \quad (2.20)$$

**Demostración.** Primero note que  $t(v)$  es tiempo de espera para que la demanda acumulada consuma  $v$  unidades, esto es

$$t(v) = \min\{n \geq 1 : w_1 + w_2 + \dots + w_n \geq v\}.$$

Ahora considere el proceso de conteo y la función de renovación asociada al proceso de demanda  $\{w_k\}$ :

$$N(v) := \begin{cases} 0 & \text{si } w_1 > v \\ \max\{n \geq 1 : w_1 + w_2 + \dots + w_n \leq v\} & \text{si } w_1 \leq v. \end{cases}$$

$$M(v) := E_{s+v}^f N(v).$$

Usando un argumento de renovación se obtiene que

$$E_{s+v}^f [N(v) | w_1 = j] = \begin{cases} 0 & \text{si } j > v \\ 1 + M(v-j) & \text{si } j \leq v. \end{cases}$$

Esto implica que

$$M(v) = \sum_{j=0}^v [1 + M(v-j)]p_j.$$

Luego, observe que

$$t(v) = N(v-1) + 1;$$

así que

$$\begin{aligned} T(v) &= 1 + M(v-1) \\ &= 1 + \sum_{j=0}^{v-1} [1 + M(v-1-j)]p_j \\ &= 1 + \sum_{j=0}^{v-1} T(v-j)p_j, \end{aligned}$$

lo cual prueba (2.19).

Por otra parte para obtener (2.20), recordemos que  $y := s + v$ . donde  $v > 0$  y use otra vez un argumento de renovación para obtener

$$E_y^f [\widehat{k}(s, y) | w_1 = j] = \begin{cases} G(y) & \text{si } j > y - s - 1, \\ G(y) + \widehat{K}(s, y-j) & \text{si } j \leq y - s - 1. \end{cases}$$

Entonces

$$\begin{aligned} \widehat{K}(s, y) &= \sum_{j=y-s}^{\infty} G(y)p_j + \sum_{j=0}^{y-s-1} [G(y) + \widehat{K}(s, y-j)]p_j \\ &= G(y) + \sum_{j=0}^{y-s-1} \widehat{K}(s, y-j)p_j, \end{aligned}$$

lo cual da el resultado deseado. ■

**Corolario 2.8** *Sea  $K(s, y)$  el valor esperado del costo que se acumula hasta que se ordena, aquí ya se incluye el costo  $K_f$ . Entonces*

$$K(s, y) = G(y) + K_f \sum_{j=y-s}^{\infty} p_j + \sum_{j=0}^{y-s-1} K(s, y-j)p_j, \quad y > s, \quad (2.21)$$

**Demostración.** De (2.18) se tiene que

$$K(s, y) := \widehat{K}(s, y) + K_f, \quad y > s.$$

Como  $\widehat{K}(s, y)$  satisface (2.20) se sigue que:

$$\begin{aligned} K(s, y) &= G(y) + \sum_{j=0}^{y-s-1} \widehat{K}(s, y-j)p_j + K_f \\ &= G(y) + K_f \sum_{j=y-s}^{\infty} p_j + \sum_{j=0}^{y-s-1} [\widehat{K}(s, y-j) + K_f]p_j \\ &= G(y) + K_f \sum_{j=y-s}^{\infty} p_j + \sum_{j=0}^{y-s-1} K(s, y-j)p_j. \blacksquare \end{aligned}$$

La siguiente sucesión tiene una gran importancia para encontrar la solución de las ecuaciones de renovación (2.19) y (2.21), es utilizada en el siguiente teorema. Sea

$$m(0) := [1 - p_0]^{-1}, \quad y \quad m(j) := \sum_{k=0}^j m(j-k)p_k, \quad \text{para } j \in \mathbb{N}. \quad (2.22)$$

**Teorema 2.9** *Sea  $\{b(n)\}$  una sucesión fija. Entonces la sucesión*

$$L(n) := \sum_{k=0}^{n-1} b(n-k)m(k), \quad n \in \mathbb{N}, \quad (2.23)$$

*es la única solución de la ecuación de renovación*

$$V(n) = b(n) + \sum_{k=0}^{n-1} V(n-k)p_k, \quad n \in \mathbb{N}. \quad (2.24)$$

**Demostración.** Sea  $n$  un entero positivo fijo. Nótese primero que

$$\begin{aligned}
L(n) &= \sum_{j=0}^{n-1} b(n-j)m(j) \\
&= b(n)m(0) + \sum_{j=1}^{n-1} b(n-j)m(j) \text{ por (2.22)} \\
&= b(n)m(0) + \sum_{j=1}^{n-1} b(n-j) \sum_{k=0}^j m(j-k)p_k \\
&= b(n)m(0) + \sum_{k=0}^{n-1} \left[ \sum_{j=k}^{n-1} b(n-j)m(j-k) \right] p_k - b(n)m(0)p_0.
\end{aligned}$$

A continuación observe que

$$\sum_{j=k}^{n-1} b(n-j)m(j-k) = \sum_{j=0}^{n-k-1} b(n-k-j)m(j) = L(n-k).$$

Entonces

$$L(n) = b(n)m(0) + \sum_{k=0}^{n-1} L(n-k)p_k - b(n)m(0)p_0$$

por lo que

$$L(n) = b(n) + \sum_{k=0}^{n-1} L(n-k)p_k,$$

lo cual prueba que  $L(n)$  satisface la ecuación de renovación (2.24).

La prueba de unicidad se hace por inducción. Suponga que  $\widehat{L}(n)$  es otra solución de la ecuación de renovación. Entonces, considerando  $L(1)$  y  $\widehat{L}(1)$  en (2.24) se verifica que verificar que  $L(1) = \widehat{L}(1)$ .

Luego, supongamos que  $L(k) = \widehat{L}(k)$  para todo  $k \leq n$ . A continuación se prueba que la igualdad se cumple para  $k = n+1$ . En efecto, como ambas sucesiones resuelven la ecuación de renovación, se tiene que

$$\begin{aligned}
L(n+1) - \widehat{L}(n+1) &= \sum_{k=0}^n [L(n+1-k) - \widehat{L}(n+1-k)]p_k \\
&= [L(n+1) - \widehat{L}(n+1)]p_0.
\end{aligned}$$

Puesto que  $p_0 < 1$ , se concluye que  $L(n+1) = \widehat{L}(n+1)$ . ■

A continuación, del Teorema 2.9 se deducen los siguientes corolarios que caracterizan a  $T(v)$  y  $K(s, y)$ , vea (2.19) y (2.21) respectivamente.



**Corolario 2.10** *La única solución de la ecuación de renovación (2.19), es decir de*

$$T(v) = 1 + \sum_{j=0}^{v-1} T(v-j)p_j, \quad v \in \mathbb{N},$$

es la función

$$T(0) := 0 \quad y \quad T(v) = \sum_{j=0}^{v-1} m(j), \quad v \in \mathbb{N}. \quad (2.25)$$

**Demostración.** Haciendo  $V(\cdot) = T(\cdot)$  y  $b(\cdot) = 1$  se observa que (2.19) tiene la estructura de la ecuación de renovación (2.24). Por lo tanto el resultado se concluye del Teorema 2.9. ■

**Corolario 2.11** *La única solución de la ecuación de renovación (2.20), es decir de*

$$\widehat{K}(s, y) = G(y) + \sum_{j=0}^{y-s-1} \widehat{K}(s, y-j)p_j, \quad y > s,$$

es la función

$$\widehat{K}(s, y) = \sum_{j=0}^{y-s-1} G(y-j)m(j), \quad y > s. \quad (2.26)$$

**Demostración.** Como la política  $(s, S)$  es fija, haciendo  $V(\cdot) = \widehat{K}(s, \cdot)$  y  $b(\cdot) = G(\cdot)$  se observa que (2.20) tiene la estructura de (2.24). ■

**Corolario 2.12** *La única solución de la ecuación de renovación (2.21), i.e., de*

$$K(s, y) = G(y) + K_f \sum_{j=y-s}^{\infty} p_j + \sum_{j=0}^{y-s-1} K(s, y-j)p_j, \quad y > s,$$

es la función

$$K(s, y) = K_f + \sum_{j=0}^{y-s-1} G(y-j)m(j), \quad y > s. \quad (2.27)$$

**Demostración.** Recordando que  $K(s, y) = K_f + \widehat{K}(s, y)$  y sustituyendo (2.26) se deduce el resultado. ■

**Observación 2.13** *Bajo una política  $f = (s, S)$  el costo promedio esperado  $J(f, x)$  no depende del inventario inicial  $x \in X$ , ya que por la Hipótesis 2.1  $p_0 < 1$ , el nivel de inventario con probabilidad uno será menor o igual al nivel de reorden  $s$  en un número finito de periodos y en consecuencia el sistema se reabastecerá reiterativamente hasta el nivel  $S$ , con tiempos entre reabastecimientos independientes e idénticamente distribuidos. En lo sucesivo el costo promedio esperado bajo una política  $(s, S)$  será denotado por  $c(s, S)$ .*

Con la teoría desarrollada en esta sección se enuncia el siguiente resultado en teoría de renovación con recompensa su demostración se puede ver en Teorema 3.16, pg 52, [9] y constituye la parte principal del algoritmo para encontrar una política óptima  $(s, S)$ .

**Teorema 2.14** *El costo promedio esperado  $c(s, S)$  tiene la siguiente estructura*

$$c(s, S) = \frac{K(s, S)}{T(S-s)}. \quad (2.28)$$

Donde  $K(s, S)$  esta dado por (2.27) y  $T(S-s)$  por (2.25)

## 2.4 Propiedades del costo promedio esperado y cotas para $s^*$ y $S^*$ .

En esta sección se aprovecharán los resultados de la sección anterior y se obtendrán los hechos necesarios para deducir un algoritmo que determine una política óptima  $(s^*, S^*)$ . Estos hechos incluyen propiedades de la función de costo  $c(s, S)$ , cotas para el nivel de reorden óptimo  $s^*$  y cotas para el nivel de reabastecimiento óptimo  $S^*$ .

Empezaremos por obtener algunos resultados preliminares.

Se define

$$\alpha_v := \frac{T(v)}{T(v+1)}, \quad v \in \mathbb{N}, \quad (2.29)$$

y nótese que  $0 < \alpha_v \leq 1$ .

**Lema 2.15** *Sea  $S$  fijo. Si  $s < S$  y  $v = S - s$ , entonces*

$$c(s-1, S) = \alpha_v c(s, S) + (1 - \alpha_v)G(s). \quad (2.30)$$

Es decir,  $c(s-1, S)$  es una combinación convexa de  $c(s, S)$  y de  $G(s)$ .

**Demostración.** Usando (2.29), (2.27) y (2.28) se obtiene

$$\begin{aligned} \alpha_v c(s, S) + (1 - \alpha_v)G(s) &= \left[ \frac{T(S-s)}{T(S-s+1)} \right] \frac{K_f + \sum_{j=0}^{S-s-1} m(j)G(S-j)}{T(S-s)} \\ &\quad + \left[ 1 - \frac{T(S-s)}{T(S-s+1)} \right] G(s) \end{aligned}$$

Luego de (2.25) resulta

$$\alpha_v c(s, S) + (1 - \alpha_v)G(s) = \frac{K_f + \sum_{j=0}^{S-s-1} m(j)G(S-j)}{T(S-s+1)} + \frac{m(S-s)}{T(S-s+1)}G(s)$$

de donde

$$\begin{aligned}\alpha_v c(s, S) + (1 - \alpha_v)G(s) &= \frac{K_f + \sum_{j=0}^{S-s} m(j)G(S-j)}{T(S-s+1)} \\ &= c(s-1, S). \blacksquare\end{aligned}$$

Para enunciar los resultados que se obtendrán a continuación considérese lo siguiente.

$$\text{Sean } y_1^* = \min\{y : G(y) = \min_x G(x)\} \quad \text{y} \quad y_2^* = \max\{y : G(y) = \min_x G(x)\}.$$

Note que  $-\infty < y_1^* \leq y_2^* < \infty$  ya que por la Hipótesis 2.2 (2)  $\lim_{|y| \rightarrow \infty} G(y) \geq \min_y G(y) + K_f$ .

**Lema 2.16** *Sea  $S$  fijo y  $s < S$ .*

- a1. Si  $G(s) < c(s, S)$ , entonces  $G(s) < c(s-1, S) \leq c(s, S)$ .
- a2. Si  $G(s) = c(s, S)$ , entonces  $G(s) = c(s-1, S) = c(s, S)$ .
- a3. Si  $G(s) > c(s, S)$ , entonces  $G(s) > c(s-1, S) \geq c(s, S)$ .
- b1. Si  $G(s) < c(s-1, S)$ , entonces  $c(s-1, S) \leq c(s, S)$ .
- b2. Si  $G(s) = c(s-1, S)$ , entonces  $c(s-1, S) = c(s, S)$ .
- b3. Si  $G(s) > c(s-1, S)$ , entonces  $c(s-1, S) \geq c(s, S)$ .

**Demostración.** Como  $0 < \alpha_v \leq 1$  de (2.30) se deduce que  $c(s-1, S)$  está entre  $G(s)$  y  $c(s, S)$  siendo posiblemente igual a  $c(s, S)$ . Considerando los diferentes casos respecto a la relación entre los valores de  $G(s)$  y  $c(s, S)$  se concluyen los resultados del lema.  $\blacksquare$

Para un nivel de reabastecimiento  $S$  dado, diremos que un nivel de reorden  $s^0 < y_1^*$  es óptimo si

$$c(s^0, S) = \min_{s < S} c(s, S).$$

El siguiente lema proporciona una caracterización para el nivel de reorden óptimo  $s$ .

**Teorema 2.17** *Sea  $S$  un nivel de reabastecimiento dado:*

1. El nivel de reorden  $s^0 < y_1^*$  es óptimo para  $S$  si

$$G(s^0) \geq c(s^0, S) \geq G(s^0 + 1). \quad (2.31)$$

*Pueden existir varios niveles de reorden  $s$  óptimos para  $S$  fijo si alguna de las desigualdades de (2.31) se satisface con igualdad; esto es,  $s^0 - 1$  ó  $s^0 + 1$  son también niveles de reorden óptimos para  $S$  si*

$$G(s^0) = c(s^0, S) \quad \text{ó} \quad G(s^0 + 1) = c(s^0, S), \quad (2.32)$$

*se satisfacen, respectivamente.*

2. Recíprocamente, si un nivel de reorden  $s^0$  es óptimo para  $S$ , entonces satisface (2.31) y  $s^0 < y_1^*$ .
3. Sean  $s_l^0$  y  $s_u^0$  los niveles de reorden óptimos para  $S$ , más pequeño y más grande respectivamente. Entonces

$$G(s_l^0) > c(s_l^0, S) = c(s_u^0, S) > G(s_u^0 + 1) \quad (2.33)$$

$$y \quad s_l^0 \leq s_u^0 < y_1^*.$$

**Demostración:**

1. supóngase que se cumple (2.31) para algún  $s^0 < y_1^*$ , se demostrará que  $s^0$  es óptimo. Para demostrar que  $s^0$  es óptimo se probará que se cumple: a)  $c(s - 1, S) \geq c(s, S)$  para todo  $s \leq s^0$  y b)  $c(s, S) \leq c(s + 1, S)$  para todo  $s \geq s^0$ . Por otra parte, se añade c) donde se demostrará la existencia de óptimos múltiples.

- (a) Para demostrar que se cumple  $c(s - 1, S) \geq c(s, S)$  para todo  $s \leq s^0$ , se demostrará por inducción que

$$G(s) \geq c(s - 1, S) \geq c(s, S) \quad (2.34)$$

se satisface para todo  $s \leq s^0$ .

Para  $s = s^0$  de (2.31) se tiene que

$$G(s^0) \geq c(s^0, S)$$

por el Lema 2.16 (a3)

$$G(s^0) \geq c(s^0 - 1, S) \geq c(s^0, S)$$

se cumple (2.34) para  $s = s^0$ .

Supongase (hipótesis de inducción) que (2.34) se cumple para  $s = y + 1$ , esto es, para algún  $y < s^0$  se supone lo siguiente

$$G(y + 1) \geq c(y, S) \geq c(y + 1, S). \quad (2.35)$$

Note que como

$$y < s^0 < y_1^*,$$

se tiene que

$$G(y) \geq G(y + 1),$$

y en vista de (2.35) y la hipótesis 2.2 (1)  $-G$  unimodal, se tiene que

$$G(y) \geq c(y, S).$$

Luego, aplicando el Lema 2.16. (a3) con  $s = y$ , se tiene

$$G(y) \geq c(y - 1, S) \geq c(y, S)$$

para todo  $y < s^0$ .

Esto demuestra que  $c(s - 1, S) \geq c(s, S)$  para todo  $s \leq s^0$ .

- (b) A continuación, para demostrar que  $c(s, S) \leq c(s+1, S)$  para todo  $s \geq s^0$ , lo haremos en dos partes: 1) para  $s^0 \leq s < y_1^*$ , y 2) para  $y_1^* \leq s < S$ .

i. Para demostrar 1), se prueba por inducción que

$$G(s+1) \leq c(s, S) \leq c(s+1, S), \quad (2.36)$$

para todo  $s^0 \leq s < y_1^*$ . Aplicando el Lema 2.16.(b1) a la segunda desigualdad de (2.31) se deduce que (2.36) se satisface para  $s = s^0$ . Para algún  $s = y - 1$  con  $s^0 \leq y - 1 < y_1^* - 1$  supóngase (hipótesis de inducción) que

$$G(y) \leq c(y-1, S) \leq c(y, S). \quad (2.37)$$

Por demostrar que (2.36) se satisface para  $s = y$ , note, otra vez que como  $y < y_1^*$  se sigue que

$$G(y+1) \leq G(y),$$

lo cual junto con (2.37) se tiene que

$$G(y+1) \leq c(y, S).$$

Aplicando el Lema 2.16.(b1) con  $s = y + 1$ , Se sigue que

$$c(y, S) \leq c(y+1, S)$$

por lo que completa el paso de inducción, quedando demostrado que  $c(s, S) \leq c(s+1, S)$  para  $s^0 \leq s < y_1^*$ .

- ii. Falta demostrar que  $c(s, S) \leq c(s+1, S)$  para  $y_1^* \leq s < S$ .

La demostración, también se hace por inducción, comenzando con  $s = S - 1$ . De (2.28) y (2.27) se tiene que

$$\begin{aligned} c(S-1, S) &= \frac{K_f + m(0)G(S)}{T(1)} \\ &= (1 - p_0)K_f + G(S) \\ &\geq G(S). \end{aligned}$$

Por el Lema 2.16.(b1)

$$G(S) \leq c(S-1, S) \leq c(S, S),$$

es decir (2.36) se cumple para  $s = S - 1$ .

Supóngase (hipótesis de inducción) que (2.36) se satisface para  $y < S - 1$ , esto es

$$G(y+1) \leq c(y, S) \leq c(y+1, S), \quad (2.38)$$

Puesto que  $y > y_1^*$

$$G(y) \leq G(y+1).$$

De (2.38) se tiene que

$$G(y) \leq c(y, S),$$

y aplicando el Lema 2.16.(a1) se obtiene

$$G(y) < c(y-1, S) \leq c(y, S)$$

para todo  $y_1^* \leq y < S - 1$ .

Puesto que también se satisface para  $S-1$ , entonces (2.36) se satisface para  $y_1^* \leq s < S$  y en general para  $s \geq s^0$  cumpliéndose b).

Queda así demostrado que  $s^0$  es óptimo cuando se cumple (2.31).

- (c) A continuación se analizará la ocurrencia de óptimos múltiples cuando al menos una de las desigualdades de (2.31) se satisface con igualdad.

Si la primera desigualdad de (2.31) se cumple con igualdad, es decir,

$$G(s^0) = c(s^0, S),$$

entonces por el Lema 2.16.(a2) se tiene

$$G(s^0) = c(s^0 - 1, S) = c(s^0, S),$$

lo cual significa que  $s^0 - 1$  también es un nivel óptimo de reorden.

En forma análoga, si la segunda desigualdad de (2.31) se cumple con igualdad, esto es si

$$c(s^0, S) = G(s^0 + 1),$$

por el Lema 2.16.(b2)

$$c(s^0, S) = c(s^0 + 1, S),$$

lo cual significa que  $s^0 + 1$  también es un nivel óptimo de reorden.

Por lo tanto, cuando al menos una de las dos desigualdades de (2.31) se cumple con igualdad ocurren óptimos múltiples para  $S$  fijo. Con esto termina la demostración de 1).

2. Sea  $s^0$  un nivel de reorden óptimo para  $S$  fijo, es decir

$$c(s^0, S) = \min_{s < S} c(s, S),$$

por lo que  $s^0$  debe cumplir

$$c(s^0 - 1, S) \geq c(s^0, S) \quad \text{y} \quad c(s^0, S) \leq c(s^0 + 1, S).$$

De (2.30) se tiene que

$$G(s^0) > c(s^0 - 1, S) \geq c(s^0, S) \quad \text{y} \quad G(s^0 + 1) \leq c(s^0, S) \leq c(s^0 + 1, S),$$

de modo que

$$G(s^0) > c(s^0, S) \quad \text{y} \quad G(s^0 + 1) \leq c(s^0, S).$$

Por lo tanto

$$G(s^0 + 1) \leq c(s^0, S) < G(s^0).$$

Es decir, se cumple (2.31), y se tiene que

$$G(s^0 + 1) < G(s^0).$$

pero como además  $y_1^*$  es el mínimo de  $G(\cdot)$  entonces  $s^0 < y_1^*$ , concluyendo con esto la demostración de 2).

3. La demostración se desarrollará en cuatro partes, en las cuales se demostrará que: a)  $s_l^0 < y_1^*$ , b)  $s_u^0 < y_1^*$ , c)  $G(s_l^0) > c(s_l^0, S)$ , d)  $c(s_u^0, S) > G(s_u^0 + 1)$ .

- (a) Por contradicción se demostrará que  $s_l^0 < y_1^*$ . Supóngase que  $s_l^0 \geq y_1^*$ . Entonces

$$G(s_l^0) \leq G(s_l^0 + 1) \leq \dots \leq G(S).$$

Usando lo anterior se tiene

$$\begin{aligned}
c(s_l^0, S) &= \frac{K_f + \sum_{j=0}^{S-s_l^0-1} m(j)G(S-j)}{T(S-s_l^0)} \\
&= \frac{K_f + m(0)G(S) + \dots + m(S-s_l^0-1)G(s_l^0+1)}{\sum_{j=0}^{S-s_l^0-1} m(j)} \\
&\geq \frac{K_f + G(s_l^0) \sum_{j=0}^{S-s_l^0-1} m(j)}{\sum_{j=0}^{S-s_l^0-1} m(j)} \\
&\geq \frac{K_f}{\sum_{j=0}^{S-s_l^0-1} m(j)} + G(s_l^0) \\
&> G(s_l^0).
\end{aligned}$$

Resumiendo esto último tenemos

$$c(s_l^0, S) > G(s_l^0).$$

Por el Lema 2.16.(b1)

$$G(s_l^0) < c(s_l^0 - 1, S) \leq c(s_l^0, S).$$

La última desigualdad contradice la definición de  $s_l^0$ , por lo tanto el supuesto es falso y con esto queda demostrado que  $s_l^0 < y_1^*$ .

- (b) En forma análoga, también por contradicción se demuestra que  $s_u^0 < y_1^*$ . En efecto siguiendo los mismos pasos que en a), se supone que  $s_u^0 \geq y_1^*$  y usando el hecho de que

$$G(s_u^0) \leq G(s_u^0 + 1) \leq \dots \leq G(S),$$

se llega a que

$$c(s_u^0, S) > G(s_u^0)$$

Luego por el Lema 2.16.(b1) se tiene que

$$G(s_u^0) < c(s_u^0 - 1, S) \leq c(s_u^0, S),$$

donde la última desigualdad contradice la definición de  $s_u^0$ . Por lo tanto, el supuesto es falso y con esto queda demostrado que  $s_u^0 < y_1^*$ .

- (c) Se quiere demostrar que  $G(s_l^0) > c(s_l^0, S)$ . También por contradicción, pero ahora sabiendo que  $s_l^0 < y_1^*$  y supóngase

$$G(s_l^0) \leq c(s_l^0, S),$$

se tiene por el Lema 2.16.(a1)

$$G(s_l^0) \leq c(s_l^0 - 1, S) \leq c(s_l^0, S).$$

La última desigualdad contradice la definición de  $s_l^0$ . Por lo tanto

$$G(s_l^0) > c(s_l^0, S),$$

lo cual prueba la primera desigualdad de (2.33).

- (d) La prueba de  $c(s_u^0, S) > G(s_u^0 + 1)$  que es la segunda desigualdad de (2.33) es como sigue:

Por contradicción, se supone que

$$c(s_u^0, S) \leq G(s_u^0 + 1).$$

Luego por el Lema 2.16.(b3)

$$c(s_u^0, S) \geq c(s_u^0 + 1, S),$$

lo cual contradice la definición de  $s_u^0$ . Por lo tanto, el supuesto es falso y la segunda desigualdad de (2.33) es cierta.

Con lo anterior queda totalmente demostrado el Teorema 2.17. ■

El siguiente corolario proporciona una manera eficiente para encontrar un nivel óptimo de reorden para un nivel de reabastecimiento  $S$  dado.

**Corolario 2.18** *Sea  $S$  un nivel de reabastecimiento dado y*

$$s^0 := \max\{y < y_1^* : c(y, S) \leq G(y)\}.$$

*Entonces (2.31) se cumple y  $s^0$  es un nivel de reorden óptimo para  $S$ .*

**Demostración.** Por la definición de  $s^0$

$$G(s^0 + 1) < c(s^0 + 1, S),$$

y del Lema 2.16.(a1)

$$G(s^0 + 1) < c(s^0, S) \leq c(s^0 + 1, S).$$

De nuevo por la definición de  $s^0$

$$c(s^0, S) \leq G(s^0),$$

y se obtiene

$$G(s^0 + 1) < c(s^0, S) \leq G(s^0).$$

Es decir, se satisface (2.31) y por el Teorema 2.17.1  $s^0$  es óptimo. ■

En el siguiente resultado se establecen cotas para un nivel de reorden óptimo  $s^*$ .

**Corolario 2.19** *Cotas para  $s^*$ . Sea  $(s^*, S^*)$  una política óptima*

1. *Si  $s_l^*$  es el menor nivel de reorden óptimo para  $S^*$  óptimo entonces*

$$s_l^* \leq \bar{s} := y_1^* - 1. \quad (2.39)$$

2. *Sea  $s_u^*$  el mayor nivel de reorden óptimo para  $S^*$  óptimo tal que  $s_u^* < y_1^*$ . Si  $s^0$  satisface la ecuación (2.31) para algún nivel de reabastecimiento  $S$  dado, entonces*

$$s^0 \leq s_u^*. \quad (2.40)$$

**Demostración:**



1. Inmediata del Teorema 2.17.2.
2. De la última desigualdad de (2.33)

$$G(s_u^* + 1) < c(s_u^*, S^*),$$

y como  $s^0$  satisface (2.31), se tiene

$$c(s^0, S) \leq G(s^0).$$

Por lo tanto

$$G(s_u^* + 1) < c(s_u^*, S^*) \leq c(s^0, S) \leq G(s^0).$$

Además, como  $G(s)$  no es creciente para  $s \leq s^0 < y_1^*$  concluimos que  $s^0 \leq s_u^*$ . ■

Para un nivel dado  $S$ , la cota inferior  $s^0$  obtenida en (2.31), se puede mejorar conforme se vayan obteniendo nuevos valores del nivel de reabastecimiento  $S$ .

**Teorema 2.20** *Sea  $(s^*, S^*)$  una política óptima. Entonces se cumple lo siguiente:*

1. Cota inferior para  $S^*$ :

$$S^* \geq \underline{S}^* := y_2^*. \quad (2.41)$$

2. Cota superior para  $S^*$ : Sea  $c^*$  el costo promedio esperado óptimo para  $(s^*, S^*)$ . Es decir, si  $c^* := c(s^*, S^*)$  entonces

$$S^* \leq \bar{S}^* := \max\{y \geq y_2^* : G(y) \leq c^*\}. \quad (2.42)$$

3. Si  $c = c(s, S)$  es el costo promedio esperado de una política arbitraria  $(s, S)$ ,  $c > c^*$  y se define

$$\bar{S}_c := \max\{y \geq y_2^* : G(y) \leq c\}. \quad (2.43)$$

entonces  $\bar{S}^* \leq \bar{S}_c$ . Además,  $\bar{S}_{c_1} \leq \bar{S}_{c_2}$  si  $c_1 \leq c_2$ .

**Demostración:**

1. Sea  $S_u^* := \max\{S^* : c(s_u^*, S^*) \text{ es óptimo}\}$

Supóngase que  $S_u^* < y_2^*$ . Como  $G(y)$  es no creciente para  $y \leq y_2^*$ , entonces

$$G(S_u^* - j) \geq G(S_u^* - j + 1) \quad \forall j \geq 0.$$

De lo anterior se obtiene

$$c(s_u^*, S_u^*) = \frac{K_f + \sum_{j=0}^{S_u^* - s_u^* - 1} m(j)G(S_u^* - j)}{T(S_u^* - s_u^*)},$$

$$c(s_u^* + 1, S_u^* + 1) = \frac{K_f + \sum_{j=0}^{S_u^* - s_u^* - 1} m(j)G(S_u^* + 1 - j)}{T(S_u^* - s_u^*)},$$

$$c(s_u^*, S_u^*) \geq c(s_u^* + 1, S_u^* + 1).$$

Esto contradice la definición de  $S_u^*$ , por lo tanto, el supuesto es falso y el inciso (1) del lema queda demostrado.

2. De nuevo usaremos contradicción. Supóngase que existe una política óptima  $(s^*, S^*)$  con  $G(S^*) > c^*$ . Note que

$$P[w < S^* - s^*] > 0,$$

ya que si

$$P[w < S^* - s^*] = 0,$$

se tendría

$$m(0) = \frac{1}{1 - p_0} = 1,$$

$$m(j) = \sum_{i=0}^j p_i m(j-i) = 0, \quad \forall j = 1, 2, \dots, S^* - s^* - 1.$$

Usando lo anterior se tiene

$$\begin{aligned} c^* = c(s^*, S^*) &= \frac{K_f + \sum_{j=0}^{S^* - s^* - 1} m(j)G(S^* - j)}{\sum_{j=0}^{S^* - s^* - 1} m(j)} \\ &= K_f + G(S^*) \geq G(S^*), \end{aligned}$$

lo cual no puede ser cierto por el supuesto de que  $G(S^*) > c^*$ . Entonces efectivamente se cumple que

$$P[w < S^* - s^*] > 0.$$

Sea

$$\widehat{\mathbb{W}} := \{\widehat{w} : \widehat{w} < S^* - s^*\} = \{0, 1, 2, \dots, S^* - s^* - 1\}$$

el conjunto de las demandas en el intervalo truncado  $[0, S^* - s^* - 1]$ . Entonces

$$P[\widehat{W} = j \mid w \in \widehat{\mathbb{W}}] = \frac{p_j}{P[w < S^* - s^*]} \quad \forall j = 0, 1, 2, \dots, S^* - s^* - 1.$$

Con esta modificación  $S^* - \widehat{W} > s^*$  se usa como el nivel de reabastecimiento aleatorio en lugar de  $S^*$  (es decir, si el nivel del inventario es menor o igual a  $s^*$  en una realización  $\widehat{w}$  de  $\widehat{W}$  entonces la posición del inventario se incrementa a  $S^* - \widehat{w}$ ).

Bajo esta política el sistema se regenera en cada época de reabastecimiento, de tal manera que el costo promedio esperado está dado por

$$c_* = \frac{K}{T},$$

donde

$$\underline{K} = \sum_{j=0}^{S^*-s^*-1} K(s^*, S^* - j) \frac{p_j}{P[w < S^* - s^*]}$$

$$\underline{T} = \sum_{j=0}^{S^*-s^*-1} T(S^* - j - s^*) \frac{p_j}{P[w < S^* - s^*]}$$

Note además que de las ecuaciones (2.28), (2.21), y (2.19)

$$c^* = \frac{K(s^*, S^*)}{T(S^* - s^*)}$$

$$\begin{aligned} c^* &= \frac{G(S^*) + K_f \sum_{j=S^*-s^*}^{\infty} p_j + \sum_{j=0}^{S^*-s^*-1} p_j K(s^*, S^* - j)}{1 + \sum_{j=0}^{S^*-s^*-1} p_j T(S^* - s^* - j)} \\ &= \frac{G(S^*) + K_f P[w \geq S^* - s^*] + \underline{K} P[w < S^* - s^*]}{1 + \underline{T} P[w < S^* - s^*]} \end{aligned}$$

Como por hipótesis  $G(S^*) > c^*$  y  $K_f P[w \geq S^* - s^*] > 0$ , vemos que

$$c^* > \frac{c^* + c_* \underline{T} P[w < S^* - s^*]}{1 + \underline{T} P[w < S^* - s^*]},$$

es decir

$$c^* + c^* \underline{T} P[w < S^* - s^*] > c^* + c_* \underline{T} P[w < S^* - s^*].$$

Luego

$$c^* > c_*,$$

lo cual contradice la definición de  $c^*$ . Por lo tanto el supuesto es falso y 2) queda demostrado.

3. Sean  $c^* \leq c_1 \leq c_2$ . Como  $G(y)$  es no decreciente para  $y \geq y_2^*$  se sigue que  $S^* \leq \bar{S}_{c_1} \leq \bar{S}_{c_2}$ . Por lo tanto 3) se cumple. ■

El Teorema 2.20.3 se usa para identificar mejores cotas superiores para  $S^*$ , así que cada vez son mejores cotas para las políticas  $(s, S)$ .

El siguiente teorema requiere la siguiente definición: Para cualquier nivel de reabaste-cimiento  $S$  fijo se define

$$c^*(S) := \min_{s < S} c(s, S) \quad (2.44)$$

Se dice que  $S$  es un mejoramiento de  $S^0$  si  $c^*(S) < c^*(S^0)$ .

**Teorema 2.21** Para cualquier nivel de reabastecimiento  $S^0 \geq y_2^*$ , sea  $s^0 < y_1^*$  el nivel de reorden óptimo para  $S^0$ .

1.  $c^*(S) < c^*(S^0)$  si y sólo si  $c(s^o, S) < c(s^o, S^0)$ .
2. Suponga que (2.31) se satisface con  $S = S^0$ . Si  $c(s^o, S') < c(s^o, S^0)$  para algún  $S' \geq y_2^*$ , entonces

$$s' := \min\{y \geq s^o : c(y, S') > G(y + 1)\}, \quad (2.45)$$

es óptimo para  $S'$ ; además  $s' < y_1^*$  y  $G(s') \geq c(s', S') \geq G(s' + 1)$ .

**Demostración:**

1. (a) ( $\Leftarrow$ ) Supongase que  $c(s^o, S) < c(s^o, S^0)$ . Como  $s^o$  es óptimo para  $S^0$ ,

$$c(s^o, S) < c^*(S^0).$$

Luego, como además  $s^o < S$ , se sigue que

$$\min_{s < S} c(s, S) \leq c(s^o, S) < c^*(S^0),$$

con lo cual se llega a

$$c^*(S) < c^*(S^0).$$

- (b) ( $\Rightarrow$ ) Por contradicción, supóngase que

$$c(s^o, S^0) \leq c(s^o, S).$$

Es suficiente demostrar que

$$c(s^o, S^0) \leq c(s, S),$$

lo cual se hará en dos partes 1) para  $s^o < s < y_1^*$  y para 2)  $s < s^o < y_1^*$

- i. Para  $s^o < s < y_1^*$ , considérese

$$\beta = \frac{T(S - s)}{T(S - s^0)}.$$

Note que  $0 < \beta \leq 1$ . De las ecuaciones (2.28) y (2.27)

$$\begin{aligned} c(s^o, S) &= \frac{K_f + \sum_{j=0}^{S-s-1} m(j)G(S-j) + \sum_{j=S-s}^{S-s^0-1} m(j)G(S-j)}{T(S-s^0)} \\ &= \frac{c(s, S)T(S-s) + \sum_{j=S-s}^{S-s^0-1} m(j)G(S-j)}{T(S-s^0)}. \end{aligned}$$

Como  $s^o$  es óptimo para  $S^0$  por el Teorema 2.17.2

$$G(s^0) \geq c(s^o, S^0) \geq G(s^o + 1),$$

y además como  $-G(\cdot)$  es unimodal con valor mínimo en  $y_1^*$  y  $s^o < s < y_1^*$

$$G(s^0 + 1) \geq \dots \geq G(s - 1) \geq G(s).$$

Por lo tanto,

$c(s^0, S^0) \geq G(s^0 + 1) \geq G(s^0 - 1) \geq \dots \geq G(s - 1) \geq G(s)$ ,  
de donde se sigue que

$$c(s^0, S) \leq \frac{c(s, S)T(S - s) + c(s^0, S^0) \sum_{j=S-s}^{S-s^0-1} m(j)}{T(S - s^0)}.$$

De la ecuación (2.25) y de la definición de  $\beta$  se obtiene

$$c(s^0, S) \leq \beta c(s, S) + (1 - \beta)c(s^0, S^0),$$

y considerando la suposición

$$c(s^0, S) \leq \beta c(s, S) + (1 - \beta)c(s^0, S)$$

se obtiene

$$c(s^0, S) \leq c(s, S).$$

De nuevo, considerando la suposición

$$c(s^o, S^0) \leq c(s^0, S) \leq c(s, S),$$

se deduce que

$$c(s^o, S^0) \leq c(s, S) \quad \forall s \text{ en } s^o < s < y_1^*. \quad (2.46)$$

ii. En forma análoga para  $s < s^o < y_1^*$  considérese ahora

$$\beta = \frac{T(S - s^0)}{T(S - s)}.$$

Note que también  $0 < \beta \leq 1$ . De las ecuaciones (2.28) y (2.27),

$$\begin{aligned} c(s, S) &= \frac{K_f + \sum_{j=0}^{S-s^0-1} m(j)G(S-j) + \sum_{j=S-s^0}^{S-s-1} m(j)G(S-j)}{T(S-s)} \\ &= \frac{c(s^0, S)T(S-s^0) + \sum_{j=S-s^0}^{S-s-1} m(j)G(S-j)}{T(S-s)}. \end{aligned}$$

Como  $s^o$  es óptimo para  $S^0$ , por el Teorema 2.17.2 vemos que

$$G(s^0) \geq c(s^0, S^0) \geq G(s^0 + 1).$$

Además, como  $-G(\cdot)$  es unimodal con valor mínimo en  $y_1^*$  y  $s < s^o < y_1^*$  se obtiene

$$G(s^0) \leq G(s^0 - 1) \leq \dots \leq G(s + 1),$$

entonces

$$c(s^0, S^0) \leq G(s^0) \leq G(s^0 - 1) \leq \dots \leq G(s + 1),$$

por lo cual al sustituir

$$c(s, S) \geq \frac{c(s^0, S)T(S - s^0) + c(s^0, S^0) \sum_{j=S-s^0}^{S-s-1} m(j)}{T(S - s)}.$$

de la ecuación (2.25) y de la definición de  $\beta$

$$c(s, S) \geq \beta c(s^0, S) + (1 - \beta)c(s^0, S^0),$$

considerando la suposición  $c(s^0, S^0) \leq c(s^0, S)$

$$\begin{aligned} c(s, S) &\geq \beta c(s^0, S^0) + (1 - \beta)c(s^0, S^0) \\ c(s, S) &\geq c(s^0, S^0), \end{aligned}$$

por lo tanto

$$c(s^0, S^0) \leq c(s, S) \quad \forall s \text{ en } s < s^0 < y_1^* \quad (2.47)$$

De las ecuaciones (2.46) y (2.47)

$$\begin{aligned} c(s^0, S^0) &\leq c(s, S), \quad \forall s \text{ en } s < y_1^* \\ c^*(S^0) &\leq c(s, S), \quad \forall s \text{ en } s < y_1^* \\ c^*(S^0) &\leq c^*(S). \end{aligned}$$

lo cual es una contradicción, por lo tanto el supuesto es falso y la parte 1) del lema queda demostrada.

2. Nótese que  $s'$  está bien definida en vista del Corolario 2.19.2. Por la definición de  $s'$

$$G(s') \geq c(s' - 1, S') \quad \text{y} \quad c(s', S') \geq G(s' + 1).$$

De la primera desigualdad y del Lema 2.16.b3

$$c(s' - 1, S') \geq c(s', S').$$

Por lo tanto

$$G(s') \geq c(s', S') \geq G(s' + 1).$$

Luego, por el Teorema 2.17.1 se tiene que  $s'$  es óptimo para  $S'$ . Y en vista de que  $G(s') \geq G(s' + 1)$  se obtiene  $s' < y_1^*$ .

Con esto queda demostrado el inciso 2) del lema. ■

## Capítulo 3

# Algoritmos e implementación computacional

### 3.1 Introducción

En este capítulo se desarrolla un algoritmo para encontrar una política óptima  $(s^*, S^*)$  para el sistema de inventarios con costo promedio. El algoritmo está basado en las propiedades de la función de costo promedio esperado  $c(s, S)$ , y va obteniendo cotas para  $s^*$  y  $S^*$  (niveles óptimos de reorden y de reabastecimiento respectivamente) cada vez más refinadas, hasta converger a la política óptima. El algoritmo es sencillo y fácil de implementar, y su complejidad computacional es solo 2.4 veces mayor que la complejidad de evaluar una política  $(s, S)$  específica [12]. El algoritmo se aplica a sistemas de inventarios con revisión periódica. Ilustraremos el algoritmo con dos ejemplos y los resultados serán validados mediante simulación.

### 3.2 Algoritmos

En esta sección, usando los resultados del capítulo anterior, se deducen algunos algoritmos para calcular el tiempo esperado hasta la siguiente orden, las funciones de costo esperado y costo esperado promedio en cada renovación. También se deduce el algoritmo principal que calcula una política óptima.

#### 3.2.1 Tiempo esperado hasta la siguiente orden

Las probabilidades se almacenaron en el vector  $P$  para aprovechar la estructura computacional de vector. Puesto que las componentes del vector se numeran computacionalmente desde 1, tomemos  $P[j] := p_{j-1}$ , donde  $j-1$  es el número de unidades demandadas.

Bajo estas consideraciones, la ecuación (2.22) queda expresada como

$$m[1] := 1/(1 - P[1]), \quad \text{y} \quad m[j] := m[1] * \sum_{i=2}^j P[i] * m[j - i + 1]. \quad (3.1)$$

De la misma manera, los tiempos esperados hasta ordenar se almacenan en el vector  $T$ . Utilizando el vector auxiliar  $m$ , la ecuación (2.25) queda transformada en

$$T[1] := 0, \quad \text{y} \quad T[j] := T[j - 1] + m[j - 1]. \quad (3.2)$$

**Algoritmo 3.1** *El algoritmo para obtener los vectores  $m[j]$  y  $T[j]$ , basado en las ecuaciones (3.1) y (3.2), tiene las siguientes instrucciones:*

```

 $m[1] \leftarrow 1/(1 - P[1]);$ 
 $T[1] \leftarrow 0;$ 
Para  $j \leftarrow 2$  hasta  $N$  hacer
     $ms \leftarrow 0;$ 
    Para  $i \leftarrow 2$  hasta  $j$  hacer
         $ms \leftarrow ms + P[i] * m[j - i + 1];$ 
    fin para
     $m[j] \leftarrow m[1] * ms;$ 
     $T[j] \leftarrow T[j - 1] + m[j - 1];$ 
fin para.

```

El parámetro  $N$  representa el valor máximo de la demanda considerada en el sistema,  $ms$  es una variable auxiliar que va acumulando las cantidades  $P[i] * m[j - i + 1]$ .

### 3.2.2 Funciones de costo en cada renovación

El costo por ordenar será denotado por  $K_f$ , para distinguirlo de la función de costo esperado  $K(s, y)$ . Además, para poder utilizar los vectores  $m[j]$  y  $T[j]$  la ecuación (2.27) se transforma en

$$K(s, y) = K_f + \sum_{j=1}^{y-s} G(y - j + 1)m[j], \quad y > s. \quad (3.3)$$



**Algoritmo 3.2** De acuerdo a las ecuaciones (3.3), (3.2) y (2.28) el algoritmo evalúa las funciones  $K(s, y)$  y  $c(s, S)$ .

```

 $KSY \leftarrow K_f;$ 
 $K \leftarrow \text{function}(s, y);$ 
   $N \leftarrow y - s;$ 
  Para  $j \leftarrow 1$  hasta  $N$  hacer
     $KSY \leftarrow KSY + m[j] * G(y - j + 1);$ 
  fin para
   $K(s, y) \leftarrow KSY$ 
fin función  $K$ .
 $c \leftarrow \text{function}(s, S);$ 
   $C(s, S) \leftarrow K(s, S)/T[S - s + 1];$ 
fin función  $c$ .

```

Se ha utilizado  $KSY$  como variable auxiliar para acumular las cantidades  $K_f$  y  $m[j] * G(y - j + 1)$ .

**Algoritmo 3.3** El algoritmo principal calcula una política  $(s^*, S^*)$  óptima y necesita como entrada las funciones  $G(\cdot)$ ,  $c(\cdot, \cdot)$  y el punto mínimo  $y^*$  de  $G(\cdot)$ . A continuación se dan las instrucciones de los dos pasos del algoritmo principal

*Paso 0*

```

 $s \leftarrow y^* - 1;$ 
 $S_0 \leftarrow y^*;$ 
Mientras  $G(s) < c(s, S_0)$  hacer  $s \leftarrow s - 1;$ 
 $s_0 \leftarrow s;$ 
 $c^0 \leftarrow c(s_0, S_0);$ 
 $S^0 \leftarrow S_0;$ 
 $S \leftarrow S^0 + 1;$ 

```

*Paso 1*

```

Mientras  $G(S) \leq c^0$  hacer
  Si  $c(s, S) < c^0$  entonces
     $S^0 \leftarrow S;$ 
    Mientras  $c(s, S^0) \leq G(s + 1)$  hacer  $s \leftarrow s + 1;$ 
     $c^0 \leftarrow c(s, S^0);$ 
  fin si
   $S \leftarrow S + 1;$ 
fin mientras

```

El Paso 0 inicia con nivel de reabastecimiento  $S^0 \leftarrow y^*$ , donde  $y^*$  es un mínimo arbitrario de la función  $G(\cdot)$ . Se encuentra el nivel de reabastecimiento  $s^0$  óptimo

para  $S^0$ , disminuyendo el valor de  $s$  con pasos de tamaño uno desde  $y^*$ , hasta que se obtiene la desigualdad  $c(s, S_0) \leq G(s)$ . La optimalidad de  $s^0$  para  $S^0$  se sigue del Corolario 2.18.

En el *Paso 1*, se busca el menor valor de  $S$  mayor que  $S^0$ , que mejora el costo para  $S^0$ . El valor de  $S$  es incrementado de uno en uno, comparando en cada paso  $c(s^0, S)$  y  $c(s^0, S^0)$  para verificar si  $S$  mejora a  $S^0$ , lo anterior es justificado por el Teorema 2.21 (1). En caso de que  $S$  sea una mejora,  $S^0$  se actualiza igualándolo a  $S$  y se obtiene el nuevo nivel óptimo de reorden  $s^0$ , incrementando de uno en uno el valor actual de  $s^0$  hasta que  $c(s, S^0) \leq G(s+1)$ . La existencia de tal nivel de reorden  $s^0$ , su optimalidad (para el nuevo valor de  $S^0$ ) y  $s^0 < y^*$  son garantizados por el Teorema 2.21 (2).

Finalmente, note que siempre que el *Paso 1* es iniciado,  $c^0$  representa una cota superior para  $c^*$  (la mejor cota disponible). En vista del Teorema 2.20 (3) la búsqueda para un mejor valor de  $S$  debe terminar cuando  $G(S) > c^0$ . En la última iteración del algoritmo, cuando  $S^0 \leftarrow S^*$  y  $s^0 \leftarrow s^*$  para alguna política óptima  $(s^*, S^*)$  se tiene que  $c^0 \leftarrow c^*$  y  $S^0 \leftarrow \bar{S}$ , por el Teorema 2.20 (2). La prueba en el ciclo exterior *mientras-fin mientras* del *Paso 1* falla cuando  $S \leftarrow \bar{S}^* + 1$ , de acuerdo a la definición de  $\bar{S}^*$  por el Teorema 2.20 (2).

### 3.2.3 Costo de producción distinto de cero

Los algoritmos anteriores solo han considerado el costo fijo por ordenar  $K_f$  y la función de costo  $G(\cdot)$ , sin tomar en cuenta el costo por unidad de producción  $c$ , ya que de acuerdo a la Observación 2.6 este costo no es necesario para obtener una política óptima.

Para problemas donde el costo por unidad de producción  $c$  es distinto de cero, una política óptima se obtiene con los algoritmos anteriores, solo que al costo promedio esperado  $c(s, S^0)$  hay que añadirle el producto que resulta de la multiplicación de la demanda promedio por el costo por unidad, como lo especifica la ecuación (2.13).

A continuación se da un algoritmo para calcular el valor esperado de la demanda y el costo promedio esperado  $co$ , para el caso con demanda finita.

**Algoritmo 3.4** *Para incluir el costo de producción distinto de cero, se consideran las siguientes instrucciones:*

```

u ← 0;
  Para j ← 1 hasta N hacer
    u ← u + P[j] * (j - 1);
  fin para
co ← C(s, Sc) + c * u;

```

En este caso  $N$  es el mayor valor de la demanda con probabilidad distinta de cero.

### 3.3 Complejidad computacional

Considere el espacio bidimensional entero, con valores de  $s$  en el eje horizontal y valores de  $S$  en el eje vertical. Cada punto  $(s, S)$  con  $s < S$  representa una política. Se supone que  $y^*$  es dada y su determinación depende de la estructura de  $G(\cdot)$ . En las discusiones siguientes  $s_0$  denota el valor de  $s$  obtenido al final del *Paso* 0, el cual es óptimo para  $S = y^*$ .

El *Paso* 1 comienza en el punto  $(s_0, y^*)$ , realizando la búsqueda a través de trayectorias verticales hacia arriba y horizontales a la derecha, finalizando en el punto  $(s^*, \bar{S}^* + 1)$ . Sin tener en cuenta la trayectoria exacta, se realizan  $\bar{S}^* - y^* + 1$  movimientos verticales, en los cuales el valor de  $S$  se aumenta con incrementos de tamaño uno, manteniendo  $s$  fijo. Además requiere  $s^* - s_0$  movimientos horizontales, en los cuales el valor de  $s$  se aumenta con incrementos de tamaño uno, para manteniendo  $S$  fijo.

Cada movimiento vertical (excepto el último el cual solamente involucra una comparación) corresponde a una iteración del ciclo exterior *mientras-fin mientras* del *Paso* 1. Este ciclo consiste de una evaluación de la función  $c(\cdot, \cdot)$ , una suma y a lo más tres comparaciones (dos si  $S^0$  no es actualizado). Cada movimiento horizontal requiere una ejecución adicional de el ciclo interno *mientras-fin mientras* el cual requiere una actualización de la función  $c(\cdot, \cdot)$ , una comparación y una adición. Una evaluación de  $c(\cdot, \cdot)$  en la ecuaciones (2.28) requiere  $2(S - s) + 1$  operaciones elementales ( $(S - s)$  adiciones,  $(S - s)$  multiplicaciones y una división) suponiendo que se dispone de los valores  $G(\cdot)$ ,  $m(\cdot)$  y  $T(\cdot)$ .

Las evaluaciones de la función  $c(\cdot, \cdot)$  en la prueba de la condición *Si* del ciclo exterior *mientras-fin mientras* cuando se realizan movimientos verticales, se efectúan en la ecuación (2.28). Cuando se calcula el numerador se utiliza en la ecuación (2.27) y obtenemos las cantidades

$$A_i(S) = K_f + \sum_{j=0}^i G(y - j)m(j), \quad i = 0, 1, \dots, S - s - 1,$$

como resultados intermedios.

El número de operaciones elementales para movimientos verticales (incluyendo la comparación final para verificar  $S > \bar{S}^*$ ) está acotado por

$$\sum_{S=y^*+1}^{\bar{S}^*} [2(S - s_0) + 1] + 4(\bar{S}^* - y^*) + 1,$$

y la cota para el número de movimientos horizontales es  $3(s^* - s_0)$ .

Por lo tanto, el número total de operaciones elementales que se necesitan en el

*Paso 1* está acotado por

$$\begin{aligned} & \sum_{S=y^*+1}^{\bar{S}^*} [2(S - s_0) + 1] + 4(\bar{S}^* - y^*) + 1 + 3(s^* - s_0), \\ &= (\bar{S}^* + y^* - 2s_0 + 6)(\bar{S}^* - y^*) + 3(s^* - s_0) + 1. \end{aligned}$$

El resto del esfuerzo computacional del algoritmo consiste de las siguientes componentes:

1. Determinación de los  $(\bar{S}^* - s_0 + 2)$  valores de

$$\left\{ G(s_0), G(s_0 + 1), \dots, G(\bar{S}^* + 1) \right\};$$

2. Determinación de  $s_0$  y  $c(s_0, S_0)$ . Esto requiere evaluaciones sucesivas de  $A_i(S_0)$  para  $i = 0, 1, \dots, S_0 - s_0 - 1$ , las cuales pueden ser hechas en  $2(S_0 - s_0)$  operaciones, con  $(y^* - s_0)$  divisiones adicionales para calcular  $c(s, S_0)$  para  $s = y^* - 1, \dots, s_0$  en las comparaciones entre  $c(s, S_0)$  y  $G(s)$ . Dado que  $S_0 = y^*$ , el número total de operaciones elementales requeridas para 2 es entonces dada por

$$2(S_0 - s_0) + 2(y^* - s_0) = 4(y^* - s_0).$$

3. Determinación de los valores  $m(\cdot)$  y  $T(\cdot)$ . El número exacto depende de la trayectoria que sigue el algoritmo pero puede ser acotado por  $(\bar{S}^* - s_0)$ . La ecuación (2.22) con ligeras modificaciones para usarse en la implementación es

$$m(j) := [1 - p_0]^{-1} \sum_{l=1}^j m(l - j)p_l.$$

El cálculo de  $[1 - p_0]^{-1}$  requiere dos operaciones,  $m(j)$  requiere  $2j$  operaciones  $j = 1, 2, \dots$  y una sola suma para obtener  $T(j + 1)$  de  $T(j)$  y  $m(j)$ . El número total de operaciones requeridas para 3 es entonces acotado por

$$\begin{aligned} & 2 + \sum_{j=1}^{\bar{S}^* - s_0 - 1} 2j + (\bar{S}^* - s_0 - 1), \\ &= 2 + (\bar{S}^* - s_0)(\bar{S}^* - s_0 - 1) + (\bar{S}^* - s_0 - 1), \\ &= (\bar{S}^* - s_0)^2 + 1. \end{aligned}$$

El número total de operaciones elementales necesarias para la determinación 2 y 3 es entonces dada por

$$\begin{aligned} B &= (\bar{S}^* + y^* - 2s_0 + 6)(\bar{S}^* - y^*) + 3(s^* - s_0) + 2 + 4(y^* - s_0) + (\bar{S}^* - s_0)^2, \\ &= 2\Delta_1^2 - \Delta_2^2 + 6\Delta_1 - 2\Delta_2 + 3\Delta_3 + 2, \\ &\leq 2(\Delta_1 + 1)^2 + 2(\Delta_1 + 1) - (\Delta_2 + 1)^2 + 3\Delta_3, \end{aligned} \tag{3.4}$$

donde  $\Delta_1 = \bar{S}^* - s_0$ ,  $\Delta_2 = y^* - s_0$  y  $\Delta_3 = s^* - s_0$ .

Note que  $(\Delta_2 + 1)^2 \geq (\Delta_3 + 1)^2 \geq 3\Delta_3$  así que la cota es

$$B \leq 2(\Delta_1 + 1)^2 + 2(\Delta_1 + 1).$$

De la discusión anterior se tiene el siguiente resultado.

**Teorema 3.5**

1. El algoritmo termina con  $(s, S^0)$  siendo una política optima y  $c^0 = c^*$ .
2. Asumiendo  $y^*$  es conocida, el algoritmo requiere a lo más

$$B = 2(\Delta_1 + 1)^2 + 2(\Delta_1 + 1) - (\Delta_2 + 1)^2 + 3\Delta_3,$$

operaciones elementales y  $(\Delta_1 + 2)$  evaluaciones de la función  $G$ .

3. Suponga que  $y^*$  y  $G(s_0 + 1), \dots, G(\bar{S}^* + 1)$  son conocidos. Denotemos por  $R(\Delta_1)$  el producto del número de operaciones elementales necesarias para el algoritmo por el número de operaciones elementales necesarias para una evaluación  $c(s_0, \bar{S}^*)$  entonces

$$R(\Delta_1) \leq 2.4 \quad \text{y} \quad \lim_{\Delta_1 \rightarrow \infty} R(\Delta_1) = 2.$$

**Demostación.** Los incisos 1 y 2 de este teorema ya han sido verificadas.

Para probar el inciso 3, note que para obtener  $m(0), \dots, m(\bar{S}^* - s_0 - 1)$  y  $T(1), \dots, T(\bar{S}^* - s_0)$  se requiere de  $(\bar{S}^* - s_0)^2 + 1 = \Delta_1^2 + 1$  operaciones. El resto de los calculos para una evaluación de  $c(s_0, \bar{S}^*)$  en (2.28) requiere  $2\Delta_1 + 1$  operaciones resultando un total de  $\Delta_1^2 + 2\Delta_1 + 2$  operaciones. La complejidad de la proporción es entonces acotada por

$$R(\Delta_1) = \frac{B}{(\Delta_1 + 1)^2 + 1} \leq \frac{2(\Delta_1 + 1)^2 + 2(\Delta_1 + 1)}{(\Delta_1 + 1)^2 + 1}..$$

Se puede verificar que  $R(\Delta_1)$  es decreciente, así que

$$R(\Delta_1) \leq R(1) \leq \frac{12}{5} = 2.4.$$

Además  $\lim_{\Delta_1 \rightarrow \infty} R(\Delta_1) = 2.$  ■

Evaluando la complejidad de la proporción  $R$  en la parte 3, excluimos el esfuerzo computacional para determinar los valores de  $G(\cdot)$  debido a que depende fuertemente de la estructura de la función  $G(\cdot)$ .

### 3.4 Ejemplos numéricos y sus implementaciones

Con el fin de ilustrar los pasos del algoritmo, se resuelven dos problemas ficticios. En el primer problema con demandas finitas y en el segundo problema con demandas con distribución de Poisson.

### 3.4.1 Problema con demandas finitas.

Se considera un sistema de inventarios con costo de producción  $c = 4$ , costo fijo por ordenar  $K_f = 6$ , costo por mantener en inventario una unidad durante un periodo  $h = 1$ , costo de penalización por unidad de demanda no satisfecha durante un periodo  $pe = 5$ . La demanda en cada uno de los periodos tiene la siguiente distribución

$$p_3 := 0.1, \quad p_4 := 0.2, \quad p_5 := 0.4, \quad p_6 := 0.3 \quad \text{y} \quad p_j := 0, \quad \forall j \neq 3, 4, 5, 6.$$

Determinaremos una política y el costo promedio esperado óptimos cuando se inicia con nivel de inventario inicial igual a cero.

#### Mínimo de la función $G(\cdot)$

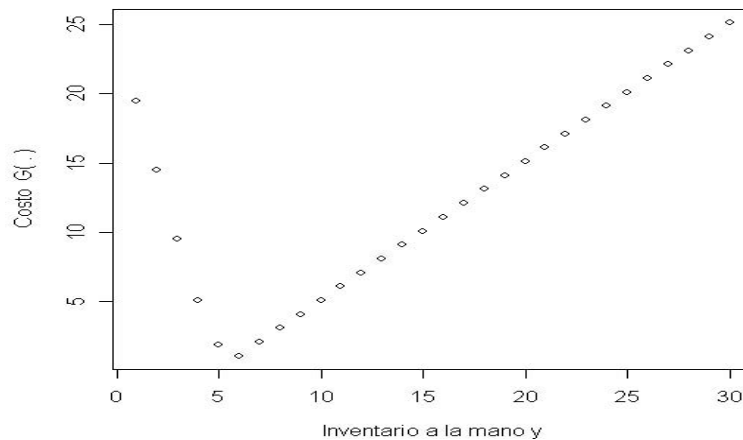
En este problema la función  $G(\cdot)$  incluye los costos por mantener el inventario y el costo de penalización. La expresión de  $G(\cdot)$  es la siguiente:

$$G(y) = 5E_w [\max \{0, w - y\}] + E_w [\max \{0, y - w\}].$$

Expresándola en forma desarrollada queda de la siguiente manera

$$G(y) = \begin{cases} -5y + 24.5, & \text{si } y < 3, \\ 9.5, & \text{si } y = 3, \\ 5.1, & \text{si } y = 4, \\ 1.9, & \text{si } y = 5, \\ 1.1, & \text{si } y = 6, \\ y - 4.9, & \text{si } y > 6, \end{cases}$$

**GRÁFICA DE LA FUNCIÓN  $G(\cdot)$**



En la gráfica se observa que el mínimo está en  $y^* = 6$ .

Recuerde que las probabilidades se almacenaron en el vector  $P$ , donde  $P[j] \leftarrow p_{j-1}$  donde las  $j - 1$  representa el valor de la demanda y  $N$  es el mayor valor de la demanda con probabilidad distinta de cero.

**Algoritmo 3.6** Alimentación de datos de entrada

```

 $N \leftarrow 7;$ 
 $K_f \leftarrow 6;$ 
 $h \leftarrow 1;$ 
 $pe \leftarrow 5;$ 
 $P[4] \leftarrow 0.1;$ 
 $P[5] \leftarrow 0.2;$ 
 $P[6] \leftarrow 0.4;$ 
 $P[7] \leftarrow 0.3$ 

```

**Algoritmo 3.7** Para generar la función  $G(\cdot)$  se enlista las siguientes instrucciones

```

 $G \leftarrow \text{función}(y);$ 
 $PE \leftarrow 0;$ 
 $H \leftarrow 0;$ 
Para  $j \leftarrow 1$  hasta  $N$  hacer
    Si  $y > j - 1$  entonces  $H \leftarrow H + (j - 1 - y) * P[j];$ 
    en otro caso  $PE \leftarrow PE + (j - 1 - y) * P[j];$ 
fin para
 $G(y) \leftarrow h * H + pe * PE;$ 
fin función  $G.$ 

```

Se introducen las variables auxiliares  $H$  y  $PE$  para calcular el valor esperado de los costos por mantener en inventario y déficit respectivamente.

### Cálculo de los vectores $m[\cdot]$ y $T[\cdot]$

Para calcular los tiempos y costos promedios esperados hasta ordenar, de acuerdo a las ecuaciones (3.1) y (3.2), se siguen los pasos del Algoritmo 3.1.

```

 $m[1] \leftarrow 1/(1 - P[1]) = 1;$ 
 $T[1] \leftarrow 0;$ 
Para  $j \leftarrow 2$  hasta  $N$  hacer
Primer iteración
 $j \leftarrow 2;$ 
 $ms \leftarrow 0;$ 
Para  $i \leftarrow 2$  hasta 2 hacer  $ms \leftarrow ms + P[2] * m[1] = 0;$ 
 $ms \leftarrow ms + P[2] * m[1] = 0;$ 
fin para
 $m[2] \leftarrow m[1] * ms = 1 * 0 = 0;$ 
 $T[2] \leftarrow T[1] + m[1] = 0 + 1 = 1;$ 

```

### Segunda iteración

```

j ← 3;
ms ← 0;
Para i ← 2 hasta 3 hacer
    i ← 2, j ← 3
    ms ← ms + P[2] * m[2] = 0;
    i ← 3, j ← 3
    ms ← ms + P[3] * m[1] = 0;
fin para
m[3] ← m[1] * ms = 1 * 0 = 0;
T[3] ← T[2] + m[2] = 1 + 0 = 1;

```

**Siguiendo las iteraciones** se obtienen los vectores

$$m = [1.00, 0.00, 0.00, 0.00, 0.10, 0.20, 0.40, 0.31, \dots],$$

$$T = [0.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.30, 1.70, 2.01, \dots].$$

Estos se utilizaran en los siguientes pasos del Algoritmo 3.3

### Algoritmo principal

A continuación se indican los pasos del Algoritmo 3.3

*Paso 0*

$$s \leftarrow y^* - 1 = 6 - 1 = 5;$$

$$S_0 \leftarrow y^* = 6;$$

*Mientras*  $G(s) < c(s, S_0)$  *hacer*  $s \leftarrow s - 1$ ;

*llamada al Algoritmo 3.2 se obtiene*

$$c(s, S_0) \leftarrow c(5, 6) = \frac{K_f + m[1]G(6)}{T[1]} = \frac{6 + (1)(1.1)}{(1)} = 7.1$$

*llamada al Algoritmo 3.7 se obtiene*  $G(s) \leftarrow G(5) = 1.9$

*como*  $G(5) < c(5, 6)$  *hacer*  $s \leftarrow 5 - 1 = 4$

*llamar los Algoritmos 3.7 y 3.2*  $G(s) \leftarrow G(4) = 5.1$

$$c(s, S_0) \leftarrow c(4, 6) = \frac{K_f + m[1]G(6) + m[2]G(5)}{T[2]}$$

$$= \frac{6 + (1)(1.1) + (0)(1.9)}{(1)} = 7.1$$

*como*  $G(4) < c(4, 6)$  *hacer*  $s \leftarrow 4 - 1 = 3$

*llamar los Algoritmos 3.7 y 3.2*  $G(s) \leftarrow G(3) = 9.5$

$$c(s, S_0) \leftarrow c(3, 6)$$

$$= \frac{K_f + m[1]G(6) + m[2]G(5) + m[3]G(4)}{T[3]} = 7.1$$



*termina ciclo Mientras, no se cumple  $G(3) < c(3, 6)$*

$s_0 \leftarrow s = 3;$

$c^0 \leftarrow c(s_0, S_0) = c(3, 6) = 7.1;$

$S^0 \leftarrow S_0 = 6;$

$S \leftarrow S^0 + 1 = 7;$

*Paso 1*

### **Primera iteración**

*Mientras  $G(S) \leq c^0$  hacer*

*Como  $G(7) = 2.1 \leq c^0 = 7.1$  hacer*

*Si  $c(s, S) < c^0$  entonces*

*utilizando el Algoritmo 3.2  $c(3, 7) = 7.8272$*

*fin si, (no se cumple la condición del si)*

$S \leftarrow S + 1 = 7 + 1 = 8;$

### **Segunda iteración**

*Mientras  $G(S) \leq c^0$  hacer*

*Como  $G(8) = 3.1 \leq c^0 = 7.1$  hacer*

*Si  $c(s, S) < c^0$  entonces*

*utilizando el Algoritmo 3.2  $c(3, 8) = 7.93077$*

*fin si, (no se cumple la condición del si)*

$S \leftarrow S + 1 = 8 + 1 = 9;$

### **Tercera iteración**

*Mientras  $G(S) \leq c^0$  hacer*

*Como  $G(9) = 4.1 \leq c^0 = 7.1$  hacer*

*Si  $c(s, S) < c^0$  entonces*

*utilizando el Algoritmo 3.2  $c(3, 9) = 7.429412$*

*fin si, (no se cumple la condición del si)*

$S \leftarrow S + 1 = 9 + 1 = 10;$

### **Cuarta iteración**

*Mientras  $G(S) \leq c^0$  hacer*

*Como  $G(10) = 5.1 \leq c^0 = 7.1$  hacer*

*Si  $c(s, S) < c^0$  entonces*

*utilizando el Algoritmo 3.2  $c(3, 10) = 6.900995$*

*(si se cumple la condición del si)*

$S^0 \leftarrow S = 10;$

*Mientras  $c(s, S^0) \leq G(s + 1)$  hacer  $s \leftarrow s + 1;$*

*$c(3, 10) \leq G(s + 1) = G(4) = 5.1$  no cumple*

*fin mientras*

$$c^0 \leftarrow c(s, S^0) = c(3, 10) = 6.900995;$$

*fin si*

$$S \leftarrow S + 1 = 10 + 1 = 11;$$

### Quinta iteración

Observe que cambio  $c^0 = 6.900995$

*Mientras  $G(S) \leq c^0$  hacer*

*Como  $G(11) = 6.1 \leq c^0 = 6.900995$  hacer*

*Si  $c(s, S) < c^0$  entonces*

*utilizando el Algoritmo 3.2  $c(3, 11) = 6.86$*

*(si cumple la condición)*

$$S^0 \leftarrow S = 11;$$

*Mientras  $c(s, S^0) \leq G(s + 1)$  hacer  $s \leftarrow s + 1$ ;*

$$c(3, 11) \leq G(s + 1) = G(4) = 5.1 \text{ no cumple}$$

*fin mientras*

$$c^0 \leftarrow c(s, S^0) = c(3, 11) = 6.86;$$

*fin si*

$$S \leftarrow S + 1 = 11 + 1 = 12;$$

### Sexta iteración

Observe que cambio  $c^0 = 6.86$

*Mientras  $G(S) \leq c^0$  hacer*

*Como  $G(12) = 7.1 \leq c^0 = 6.900995$  NO CUMPLE*

*fin mientras*

El Algoritmo Principal 3.3 termina con

**Política Óptima (3,11),**

Costo Promedio Óptimo 6.86,

sin incluir todavía el costo de producción  $c$ .

### Calculo del costo promedio considerando el costo de producción $c = 4$ .

Del Algoritmo 3.4 se obtiene que

$$u \leftarrow 0 + 0 * 1 + 0 * 2 + 0.1 * 3 + 0.2 * 4 + 0.4 * 5 + 0.3 * 6 + 0 * 7 = 4.9,$$

$$co \leftarrow C(s, S^c) + c * u = c(3, 11) + 4 * 4.9 = 6.86 + 19.6 = 26.46.$$

Finalmente los algoritmos obtienen la siguiente solución para el problema:

**Política Óptima (3,11),**

**Costo Promedio Óptimo 26.46.**

### 3.4.2 Problema con demandas con distribución de Poisson

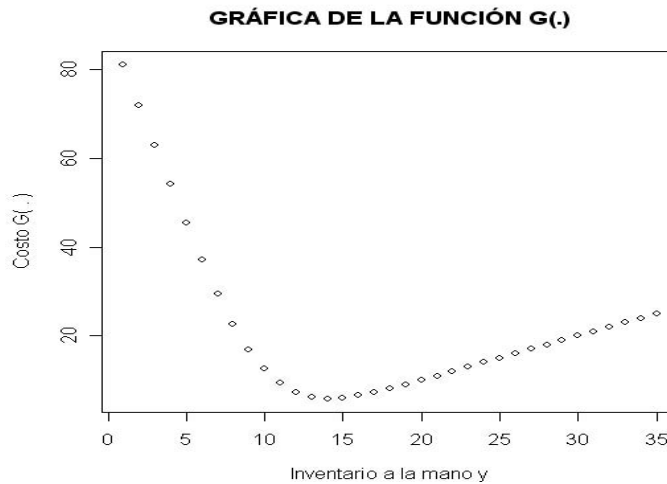
Ahora se considera un sistema de inventarios con costo de producción  $c = 5$ , costo fijo por ordenar  $K_f = 64$ , costo por mantener en inventario una unidad durante un periodo  $h = 1$ , costo de penalización por unidad de demanda no satisfecha durante un periodo  $pe = 9$ . La demanda en cada uno de los periodos tiene una distribución de Poisson con parámetro  $lambda = 10$ .

Determináremos una política y el costo promedio óptimos, cuando se inicia con nivel de inventario cero.

#### Mínimo de la función $G(\cdot)$

Para este problema la función  $G(\cdot)$ , tiene la forma

$$G(y) = 9E_w [\max \{0, w - y\}] + E_w [\max \{0, y - w\}],$$



En la gráfica se observa que el mínimo está en  $y^* = 14$ .

#### Algoritmo 3.8 Alimentación de datos de entrada

$N \leftarrow 100;$

$K_f \leftarrow 64;$

$h \leftarrow 1;$

$lambda \leftarrow 10;$

$P \leftarrow$  Vector de probabilidades Poisson con parámetro  $lambda$ ;

El vector  $P$  es generado por el sistema R. Con estos datos se ejecuta el Algoritmo 3.7 para generar la función  $G(y)$ .

**Calculo de los vectores  $m[\cdot]$  y  $T[\cdot]$** 

Al seguir los pasos del Algoritmo 3.1, se obtiene

$$m[1] \leftarrow 1/(1 - P[1]) = 1/(1 - 4.54 \cdot 10^{-5}) = 1.0000454,$$

$$T[1] \leftarrow 0,$$

*Para  $j \leftarrow 2$  hasta  $N$  hacer*

**Primer iteración**

$$j \leftarrow 2;$$

$$ms \leftarrow 0;$$

*Para  $i \leftarrow 2$  hasta 2 hacer*

$$ms \leftarrow ms + P[2] * m[1] = 0 + 0.000454 * 1.0000454 = 0.000454,$$

*fin para*

$$m[2] \leftarrow m[1] * ms = 1.0000454 * 0.000454 = 0.000454,$$

$$T[2] \leftarrow T[1] + m[1] = 0 + 1.0000454 = 1.0000454,$$

**Siguiendo las iteraciones**

$$m = [1.000045, 0.000454, 0.002270, 0.007569, 0.018930, 0.037888, \dots],$$

$$T = [0.00, 1.000045, 1.000499, 1.002770, 1.010339, 1.029270, \dots],$$

**Algoritmo principal**

El Algoritmo Principal 3.3 termina con

**Política Óptima (6,40),**

**Costo Promedio Óptimo 35.02156,**

sin considerar todavía el costo de producción  $c$ .

Este resultado coincide con el que se reporta en [12]. Los demás resultados reportados también pueden ser obtenidos de igual forma, cambiando el valor adecuado de  $lambda$  en el Algoritmo 3.8.

**Calculo del costo promedio considerando el costo de producción  $c = 5$ .**

Para considerar el costo de producción  $c = 5$ , se necesita añadirle al costo promedio  $c(s, S^0)$  la demanda promedio  $c * lambda = 5 * 10 = 50$ , como lo especifica la ecuación (2.13).

Finalmente el problema, tiene por solución los siguientes resultados:

**Política Óptima (6,40),**

**Costo Promedio Óptimo 85.02156.**

**3.5 Simulación e implementación**

Los datos de los problemas anteriores son ficticios, por lo que se simula la evolución de los sistemas para contrastar resultados [10]. En la simulación la política es dada,

se inicia con nivel de inventario cero y además se utilizan las siguientes variables y vectores:

$n =$  Número de periodos de evolución del inventario.

$SC =$  Suma de costos de producción.

$SH =$  Suma de costos por mantener inventario.

$SPE =$  Suma de costos de penalización.

$CP =$  Costo promedio óptimo.

$X =$  Vector evolución de estados del sistema de inventarios.

$w =$  Vector evolución de las demandas.

$C =$  Vector evolución de costos por producción.

$H =$  Vector evolución de costos por mantenimiento.

$PE =$  Vector evolución de costos por penalización

**Algoritmo 3.9** Lista de instrucciones para simular un inventario con política  $(s, S)$  dada.

Para  $i \leftarrow 1$  hasta  $n$  hacer  
 Si  $X[i] \leq s$  entonces hacer  
      $y \leftarrow S - X[i];$   
      $SC \leftarrow SC + c * y + K_f;$   
      $X[i + 1] \leftarrow S;$   
     si no  $X[i + 1] \leftarrow X[i];$   
 fin si  
 $w[i] \leftarrow d(i) :=$  demanda simulada en la etapa  $i;$   
 $X[i + 1] \leftarrow X[i + 1] - w[i];$   
 Si  $X[i + 1] \leq 0$  entonces hacer  
      $SPE \leftarrow SPE - pe * X[i + 1];$   
     si no  $SH \leftarrow SH + h * X[i + 1];$   
 $C[i] \leftarrow SC;$   
 $H[i] \leftarrow SH;$   
 $PE[i] \leftarrow SPE;$   
 fin para  
 $CP \leftarrow (SC + SH + SPE)/n;$

### 3.5.1 Problema con demandas finitas.

Para la simulación de este problema, además de utilizar el Algoritmo 3.9, es necesario simular las demandas.

La simulación de las demandas se hace con el método de la transformada inversa [10], el cual consiste en lo siguiente: Para simular el valor de la variable aleatoria  $D$

con densidad de probabilidad  $P[D = d_j] = p_j, \forall j = 0, 1, \dots$ , se genera un número aleatorio  $u$  uniformemente distribuido en  $(0,1)$  y se hace

$$D = \begin{cases} d_0 & \text{si } u < p_0, \\ d_1 & \text{si } p_0 \leq u < p_0 + p_1, \\ \vdots & \vdots \\ d_j & \text{si } \sum_{i=1}^{j-1} p_i \leq u < \sum_{i=1}^j p_i, \\ \vdots & \vdots \end{cases}$$

Recuerde que también se requiere alimentar la política óptima (3, 11) y los otros datos del problema:

**Algoritmo 3.10** *Alimentación de datos de entrada del problema*

```

 $K_f \leftarrow 6;$ 
 $c \leftarrow 4;$ 
 $h \leftarrow 1;$ 
 $pe \leftarrow 5;$ 
 $W \leftarrow [3, 4, 5, 6];$ 
 $P \leftarrow [0.1, 0.2, 0.4, 0.3];$ 
 $s \leftarrow 3;$ 
 $S \leftarrow 11;$ 
 $X[1] \leftarrow 0;$ 
 $n \leftarrow 100000;$ 

```

El vector  $W$  almacena las demandas con probabilidad distinta de cero y es apuntador hacia el vector  $P$  que almacena las correspondientes probabilidades.

**Algoritmo 3.11** *Para poder generar una demanda y utilizarla en cada etapa del sistema, se hace mediante la estructura de función.*

```

 $d \leftarrow \text{function}(x)$ 
   $u \leftarrow \text{número aleatorio uniformemente distribuido en } (0, 1)$ 
   $p \leftarrow P[1];$ 
   $j \leftarrow 1;$ 
  Mientras  $u > p$  hacer
     $j \leftarrow j + 1;$ 
     $p \leftarrow p + P[j];$ 
  fin mientras
   $d(x) \leftarrow W[j];$ 
fin función  $d$ 

```

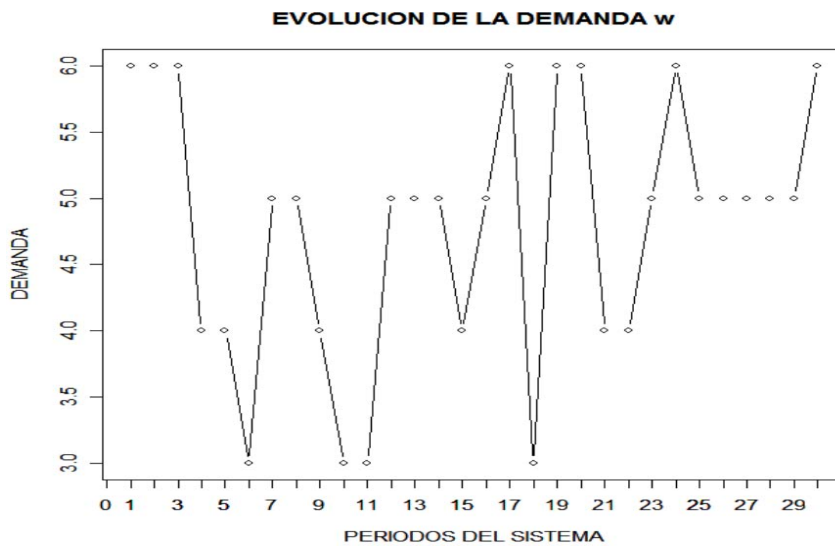
El Algoritmo 3.9 utiliza los Algoritmos 3.10 y 3.11 para poder simular el problema.

En una corrida del algoritmo considerando  $n = 100,000$  periodos del inventario, se obtuvieron los siguientes resultados.

$$X = [ 0 \ 5 \ -1 \ 5 \ 1 \ 7 \ 4 \ -1 \ 6 \ 2 \ 8 \ 5 \ \dots ]$$

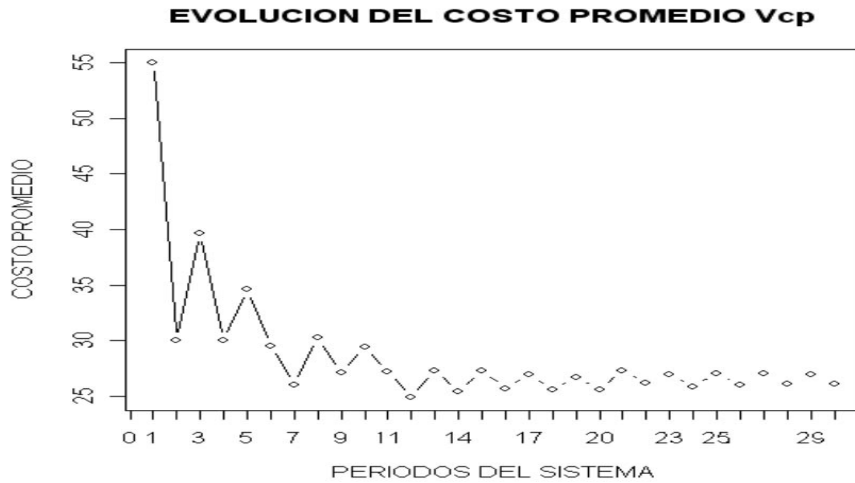


$$w = [ 6 \ 6 \ 6 \ 4 \ 4 \ 3 \ 5 \ 5 \ 4 \ 3 \ 3 \ 5 \ \dots ]$$



$$\begin{aligned}
 C &= [50 \ 50 \ 104 \ 104 \ 150 \ 150 \ 150 \ 204 \ 204 \ 246 \ 246 \ 246 \ \dots] \\
 H &= [5 \ 5 \ 10 \ 11 \ 18 \ 22 \ 22 \ 28 \ 30 \ 38 \ 43 \ 43 \ \dots] \\
 PE &= [0 \ 5 \ 5 \ 5 \ 5 \ 5 \ 10 \ 10 \ 10 \ 10 \ 10 \ 10 \ \dots]
 \end{aligned}$$

$$V_{cp} = [55 \quad 30 \quad 39.67 \quad 30 \quad 34.60 \quad 29.50 \quad 26 \quad 30.25 \quad 27.11 \quad 29.40 \quad \dots]$$



Al realizar varias corridas también de  $n = 100,000$  periodos se obtienen los siguientes costos promedios esperados

$$26.45015 \quad 26.46852 \quad 26.46109 \quad 26.46602 \quad 26.4793 \quad 26.47514$$

$$26.46767 \quad 26.45723 \quad 26.45744 \quad 26.45708 \quad 26.46639 \quad 26.47571$$

Recuerde que el costo promedio óptimo es  $c(3,11) = 26.46$ . Como se puede observar la simulación corrobora el resultado obtenido por el algoritmo propuesto en este trabajo de tesis.

Simulando otras políticas se puede observar que (3,11) es efectivamente la de menor costo promedio esperado.

$c(0,13)$	$c(1,13)$	$c(2,13)$	$c(3,13)$	$c(4,13)$	$c(5,13)$	$c(6,13)$
28.72	28.15	27.59	27.43	27.70	28.06	28.21

$c(0,12)$	$c(1,12)$	$c(2,12)$	$c(3,12)$	$c(4,12)$	$c(5,12)$	$c(6,12)$
28.89	27.88	27.14	26.97	27.10	27.21	28.40

$c(0,11)$	$c(1,11)$	$c(2,11)$	$c(3,11)$	$c(4,11)$	$c(5,11)$	$c(6,11)$
28.55	27.29	26.61	26.46	26.48	27.43	29.43

$c(0,10)$	$c(1,10)$	$c(2,10)$	$c(3,10)$	$c(4,10)$	$c(5,10)$	$c(6,10)$
28.06	26.97	26.55	26.50	26.81	28.49	29.85

$c(0,9)$	$c(1,9)$	$c(2,9)$	$c(3,9)$	$c(4,9)$	$c(5,9)$	$c(6,9)$
28.14	27.50	27.33	27.04	27.76	28.87	29.70

### 3.5.2 Problema con demandas con distribución de Poisson

En este problema la simulación de las demandas se hace directamente con una instrucción, la cual genera números aleatorios con distribución de Poisson y es utilizada en el Algoritmo 3.9, en la instrucción



$$w[i] \leftarrow d(i) := \text{demanda simulada en la etapa } i$$

En este problema, la alimentación de los datos, junto con la política (6,40) se realiza con las siguientes instrucciones:

$$K_f \leftarrow 64;$$

$$c \leftarrow 5;$$

$$h \leftarrow 1;$$

$$pe \leftarrow 9;$$

$$w \leftarrow \text{demanda Poisson}$$

$$s \leftarrow 6;$$

$$S \leftarrow 40;$$

$$X[1] \leftarrow 0;$$

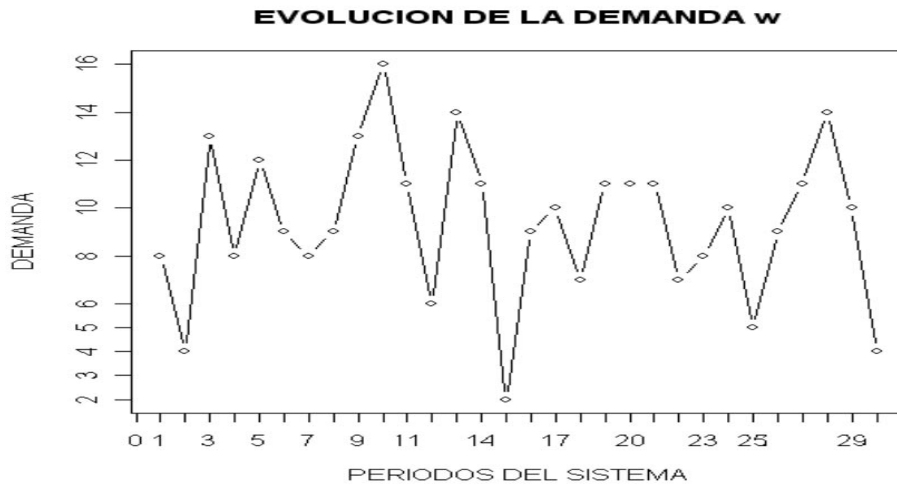
$$n \leftarrow 100000;$$

En una corrida del Algoritmo 3.9 considerando  $n = 100,000$  periodos del inventario, se obtienen los siguientes resultados.

$$X = [ 0, 32, 28, 15, 7, -5, 31, 23, 14, 1, 24, 13, 7, -7, 29, 27, 18, 8, 1, 29, 18, 7, 0, \dots ]$$



$$w = [ 8, 4, 13, 8, 12, 9, 8, 9, 13, 16, 11, 6, 14, 11, 2, 9, 10, 7, 11, 11, 11, 7, 8, 10, \dots ]$$



$$V_{cp} = [ 296, 162, 113, 86.5, 78.2, 118.5, 104.85, 93.50, 83.22, 103.20, 95.00, 87.67, 85.77, 103.07, 98.00, 93.00, 88.00, 83.17, 93.94, 90.15, 86.19, 82.27, \dots ]$$



La evolución de los otros costos es como sigue:

$$\begin{aligned}
 C &= [ 264 \quad 264 \quad 264 \quad 264 \quad 264 \quad 553 \quad 553 \quad 553 \quad 553 \quad 812 \quad 812 \quad \dots ] \\
 H &= [ 32 \quad 60 \quad 75 \quad 82 \quad 82 \quad 113 \quad 136 \quad 150 \quad 151 \quad 175 \quad 188 \quad \dots ] \\
 PE &= [ 0 \quad 0 \quad 0 \quad 0 \quad 45 \quad 45 \quad 45 \quad 45 \quad 45 \quad 45 \quad 45 \quad \dots ]
 \end{aligned}$$

Se realizan varias corridas también de  $n = 100,000$  periodos y se obtienen los siguientes costos promedios esperados

$$\begin{aligned}
 &85.06527 \quad 85.0265 \quad 85.03992 \quad 85.02532 \quad 85.03723 \quad 84.97543 \\
 &84.9977 \quad 84.99914 \quad 85.02090 \quad 84.98499 \quad 84.98267 \quad 85.0408
 \end{aligned}$$

Recuerde que el costo promedio óptimo es  $c(6, 40) = 85.02156$ . En este caso se observa también que las simulaciones corroboran los resultados obtenidos por el Algoritmo 3.3.

Haciendo simulaciones con otras políticas también para 100,000 periodos, se puede observar que (6, 40) es efectivamente la de menor costo promedio esperado.

$c(3, 41)$	$c(4, 41)$	$c(5, 41)$	$c(6, 41)$	$c(7, 41)$	$c(8, 41)$	$c(9, 41)$
85.7790	85.3003	85.1009	85.0795	85.1706	85.6688	86.0903

$c(3, 40)$	$c(4, 40)$	$c(5, 40)$	$c(6, 40)$	$c(7, 40)$	$c(8, 40)$	$c(9, 40)$
85.6226	85.2925	85.1192	85.0375	85.0841	85.4308	86.0430

$c(3, 39)$	$c(4, 39)$	$c(5, 39)$	$c(6, 39)$	$c(7, 39)$	$c(8, 39)$	$c(9, 39)$
85.7856	85.3784	85.1842	85.0485	85.1587	85.3853	86.0771



## Capítulo 4

### Conclusiones y perspectivas

En este trabajo se han desarrollado los fundamentos teóricos y la implementación de un método para encontrar una política  $(s, S)$  que minimiza el costo promedio esperado de un sistema de inventario con revisión periódica para un solo producto.

El sistema de inventario fue modelado como un problema de control Markoviano (PCM) y la base fundamental del método fue la estructura de renovación del costo promedio esperado inducida por una política  $(s, S)$ . La teoría aquí desarrollada para el PCM se basó en [1] y [2]; la teoría de procesos regenerativos en [9], el algoritmo implementado en los trabajos [12] y [4], la simulación en [10]; la bibliografía restante sirvió de apoyo.

La importancia del método radica en que trabaja bajo hipótesis muy generales, por lo que se puede modelar una amplia variedad de sistemas de inventarios.

La implementación computacional del algoritmo permite acortar la brecha entre teoría y práctica en sistemas de inventarios. La implementación fue desarrollada en lenguaje R y soporta cualquier distribución discreta para la demanda, ya sea generada por el sistema del lenguaje R o por el usuario. El Algoritmo 3.10 simula la demanda para el caso finito, el cual es muy frecuente en sistemas de inventarios.

Se analizaron dos ejemplos de sistemas de inventarios con datos ficticios. La solución óptima arrojada por el método propuesto fue contrastada con los resultados obtenidos mediante simulación, lográndose en ambos ejemplos resultados satisfactorios.

Desde el punto de vista pedagógico la implementación del algoritmo y simulación del sistema puede ser de utilidad, ya que el estudio de sistema de inventarios es un tema curricular en escuelas de ingeniería industrial y escuelas de administración.

Algunas posibles líneas de extensión de este trabajo son las siguientes:

1. Este trabajo se enfocó a sistema de inventarios con revisión periódica. Se reporta en [12] que con algunas modificaciones el algoritmo aquí implementado se aplica a sistemas con revisión continua. Entonces, una natural extensión sería la implementación del algoritmo para sistemas con revisión continua.
2. Una limitación del algoritmo implementado es que considera que la entrega es inmediata. En la práctica generalmente los tiempos de entrega son variables aleatorias. Zipkin [14] reporta que el tiempo de entrega aleatoria puede ser incorporado en el costo  $G(\cdot)$  como tiempo de entrega fijo, sin violar las condi-

ciones de la hipótesis aquí consideradas. Entonces, una segunda posibilidad consistiría en extender el algoritmo a esta nueva situación.

3. En el ámbito comercial las funciones de costo generalmente son cóncavas, ya que los costos unitarios van disminuyendo en la compra de grandes cantidades. Por lo tanto, otra extensión sería el estudio de sistemas de inventarios con funciones de costos de producción cóncava.
4. En este trabajo se consideró el índice de costo promedio. En algunas situaciones se requiera considerar el criterio de costo descontado, lo cual plantea otra línea de trabajo.

## Bibliografía

- [1] D. Bertsekas (1987), *Dynamic Programming: Deterministic and Stochastic Models*, Prentice-Hall, New York
- [2] D. Bertsekas (1995), *Dynamic Programming and Optimal Control Stochastic, Vols 1 and 2*, Athenea Scientific, Belmont, MA.
- [3] D. Beyer and S.P. Sethi (1999), *The classical average-cost inventory models of Iglehart and Veinott-Wagner, revisited*, Journal of Optimization Theory and Applications: Vol. 101, No. 3, pp 523-555.
- [4] A. Federgruen and P. Zipkin (1984), *An efficient algorithm for computing optimal  $(s, S)$  policies*, Oper. Res: Vol. 32, No. 6, pp 1268-1285.
- [5] E.A. Feinberg and M.E. Lewis (2005), *Optimality inequalities for average cost Markov decision processes and optimality of  $(s,S)$  policies*. dirección electrónica: <http://www.ams.sunysb.edu/~feinberg/public/feinberg-lewis.pdf>.
- [6] H. L. Lee and S. Nahmias (1993), *Single-product, single-location models*, in Handbooks in OR & MS: Vol. 4, Eds. S. C. Graves et al., Elsevier Science Publishing, North Holland.
- [7] S.F. Love (1979), *Inventory Control Theory*, Mc. Graw-Hill.
- [8] E. L. Porteus (1990), *Stochastic inventory theory*, in Handbooks in OR & MS, Vol. 2, Eds. D. P. Heyman, M. J. Sobel, Elsevier Science Publishing B. V., North Holland.
- [9] S. M. Ross (1970), *Applied Probability Models with Optimization Applications*, Holden-Day, San Francisco.
- [10] S. M. Ross (2002), *Simulation*, Academic Press, New York.
- [11] W.N. Venables, D. M. Smith and the R Development Core Team, *An Introduction to R* (Copyright c 1999–2005) R Development Core Team.
- [12] Y. S. Zheng and A. Federgruen (1991), *Finding optimal  $(s, S)$  policies is about as simple as evaluating a simple policy*, Oper. Res: Vol 39, No 4, pp 654-665.
- [13] Y. S. Zheng (1991), *A simple proof for optimality of  $(s, S)$  policies in infinite-horizon inventory systems*, J. Appl. Prob: Vol 28, pp 802-810.
- [14] P. Zipkin (1986), *Stochastic leadtimes in continuous-time inventory models*. Naval Res. Logist. Quart: Vol 33, pp 763-774