UNIVERSIDAD CENTROCCIDENTAL
"LISANDRO ALVARADO"

Decanato de Ciencias y Tecnología
Licenciatura en Ciencias Matemáticas

"Estudio de acotamiento para el costo de almacenamiento del problema de dimensionamiento del lote económico con parámetros de costos invariantes en el tiempo"

Trabajo Especial de Grado presentado por

Br. Alexis Andres Yglesias Yajure

como requisito final
para obtener el título de Licenciado
en Ciencias Matemáticas
Área de Conocimiento: Investigación de operaciones.
Tutor: Prof. Alí duin

Barquisimeto, Venezuela. Noviembre del 2016

Universidad Centroccidental
"Lisandro Alvarado"
Decanato de Ciencias y Tecnología
Licenciatura en Ciencias Matemáticas

ACTA
TRABAJO ESPECIAL DE GRADO

Los suscritos miembros del Jurado designado por el Jefe del Departamento de Matemáticas del Decanato de Ciencias y Tecnología de la Universidad Centroccidental "Lisandro Alvarado", para examinar y dictar el veredicto sobre el Trabajo Especial de Grado titulado:

"Estudio de acotamiento para el costo de almacenamiento del problema de dimensionamiento del lote económico con parámetros de costos invariantes en el tiempo"

presentado por el ciudadano Br. Alexis Andres Yglesias Yajure titular de la Cédula de Identidad No. 19.954.959, con el propósito de cumplir con el requisito académico final para el otorgamiento del título de Licenciado en Ciencias Matemáticas.

Luego de realizada la Defensa y en los términos que imponen los Lineamientos para el Trabajo Especial de Grado de la Licenciatura en Ciencias Matemáticas, se procedió a discutirlo con el interesado habiéndose emitido el veredicto que a continuación se expresa:

[1] ____________________

Con una calificación de ____________________ puntos.

En fe de lo expuesto firmamos la presente Acta en la Ciudad de Barquisimeto a los _____ días del mes de ____________________ de __________.

TUTOR	FIRMA
PRINCIPAL	FIRMA
PRINCIPAL	FIRMA

OBSERVACIONES:

____

____

____

[1] Aprobado ó Reprobado

A mis padres, hermanos y mi futura esposa Maria

AGRADECIMIENTOS

La finalización de este trabajo no hubiese sido posible sin el constante apoyo de mis padres, que desde que comencé mis estudios universitarios, siempre han estado presente tanto en los malos como en los buenos momentos. Por eso quiero darles las gracias a mi padre Alexis Rafael Yglesias y a mi madre Iris Yajure, porque sin ustedes no podría haber culminado este proyecto, el cual es necesario para optar por un título de Licenciado.

Además de mis padres, debo agredecer también a Maria Angelica Brown, quien es una persona muy importante en mi vida actual; en ningún momento ha dejado de apoyarme y siempre ha estado presente cuando mas lo he necesitado. A sus padres: Walter Brown y Maria Eugenia Oropeza, por permitirme un espacio en su hogar y tratarme como de la familia, mientras culminaba mis estudios de pregrado.

Agradezco a mis hermanos Arycson Yglesias, Aleiris Yglesias y Andris Yglesias, porque ellos son mi motivación para seguir adelante en el aprendizaje. Asi como también a mi tia Patricia Yajure, ya que en el tiempo que pude vivír con ella logré un gran creciemiento personal. A mis futuros colegas y amigos Mitchel Pargas, Uvencio Gimenez, Andrei Henriquez y Rafael Azuaje a quienes conozco desde el incio de la universidad y aún seguimos siendo muy buenos amigos. Ellos siempre me han apoyado en lo académico y en lo personal.

Finalmente, la terminación de este trabajo fue gracias a las asesorias del profesor Ali Duin, quien a pesar de su reducida disponibilidad de tiempo, siempre encotró un espacio para colaborar con este proyecto. También fue posible gracias a mi futuro colega Aldemar Gómez, quien con sus conocimiento de Latex, ayudó a mejorar los detalles en la transcripción del trabajo. Agradezco a Dios y a todos los mencionados por su ayuda y colaboración.

Resumen

En este estudio, se desea demostrar, hasta que punto la propiedad de perfecto balanceamiento entre el costo de almacenamiento y el costo de preparación, se satisface para el problema ELS, en una solución óptima. Se consideró sin perdida de generalidad un horizonte de pedido que consiste de los periodos $\{1, \ldots, t\}$, y se definió una cota para el costo total de almacenamiento en dicho horizonte de pedido. Esta cota es una cantidad proporcional al costo de preparación. Cabe agregar, que un horizonte de pedido es aquel en donde un único pedido cubre un número entero de periodos consecutivos.

Luego, se define la heurística de Silver-Meal, la cual esta basada en la propiedad de balanceamiento entre el costo de almacenamiento y el costo de preparación del modelo EOQ. Las heurística de Costo Unitario Mínimo (LUC) y la de Balanceamiento de Periodos Parciales (PPB), también estan basadas en esta propiedad. Finalmente, se utiliza la cota obtenida en el capítulo 2 para la construcción de una nueva heurística llamada H^*; y el buen rendimiento de la misma se comprueba al comparar los resultados de la ejecución de H^* con las otras heurísticas.

Índice

Introducción

En el año 1958 Harvey M. Wagner y Thomson M. Whitin publicarón un artículo llamado "Dynamic Version of the Economic Lot Size Model", el cual es una versión del problema del tamaño del lote económico, donde las demandas a lo largo de todo el horizonte de tiempo, son conocidas y variables para cada periodo. Para diferenciar el modelo original de su versión dinámica, se suele denominar al problema de inventario con una tasa de demanda constante, para todo el horizonte de tiempo, como modelo de la "cantidad de pedido enconómico" (EOQ, por sus siglas en inglés); y como modelo de "dimensionamiento del lote económico" (ELS, por sus siglas en inglés) al introducido por Wagner y Whitin en 1958 [4].

En este trabajo se cosidera el problema de dimensionamiento del lote económico (ELS). Este modelo se aplica en aquellas situaciones en donde un artículo presenta una demanda determinista pero variable en un horizonte de tiempo finito y discreto. Así como también costos de preparación y almacenamiento distintos para cada periodo. El problema consiste en determinar las cantidades de dicho producto que deben ordenarse o producirse para que los costos totales del inventario sean minimizados.

En el primer capítulo, se presentan algunas propiedades del problema de la cantidad de pedido económico (EOQ, por sus siglas en inglés). Este modelo posee una tasa de demanda conocida y constaste a lo largo de un horizonte de tiempo infinito y continuo. Además, en él se consideran costos de preparación y almacenamiento constantes durante todo el horizonte de tiempo. En base a estas características, se demuestra que en el modelo EOQ, dichos costos estan perfectamente balanceados en una solución óptima.

En este capítulo, también se estudia el algoritmo presentado por Wagner y Whitin [4], y se comprueba que existe una solución óptima para el problema ELS, que satisface la propiedad de inventario cero; es decir, una orden es realizada solo cuando el nivel del inventario disminuye hasta llegar a cero.

Cabe destacar, que una de las propiedades más interesantes del modelo EOQ, es el perfecto balanceamiento que existe entre el costo de almacenamiento y el costo de preparación en una solución óptima. Esta propiedad se basa en el hecho, de que el costo total del inventario, es una función cóncava de la variable y, donde y representa el nivel del inventario en un determinado punto en el tiempo.

La propiedad de perfecto balanceamiento del problema EOQ, es utilizada como base fundamental para la construcción de muchas heurísticas para el modelo de dimensionamiento del lote económico (ELS). Una de las heurísticas más conocidas, es la denominada Balanceamiento de Periodos Parciales (PPB, por sus siglas en inglés). Esta intenta contruir una solución, donde los costos de preparación y de almacenamiento se encuentren balanceados. Otras heurísticas que buscan explotar esta propiedad, es la conocida Silver-Meal (SM) y la del Costo Unitario Mínimo (LUC, por sus siglas en inglés). En la primera se busca minimizar los costos promedios por periodos y en la segunda se busca minimizar el costo promedio por unidad de artículo.

Son muchas las heurísticas que intentan construir una solución donde los costos esten perfectamente balanceados. Sin embargo, tal como se demostrará en el capítulo 3, estas heurísticas no siempre poseen un buen comportamiento. Esto se debe a que el costo de almacenamiento y el costo de preparación, no necesariamente estan balanceados en una solución óptima del modelo ELS.

En vista de lo anterior, es inevitable realizar la siguiente pregunta: ¿Hasta que punto la propiedad de balanceamiento entre el costo de almacenamiento y el costo de preparación en una solución óptima, se cumple para el modelo ELS?. La respuesta a esta interrogante fue dada por Wilco van den Heuvel y Albert P.M. Wagelmans en un artículo llamado "A holding cost bound for the economic lot-sizing problem with time-invariant cost parameters" [6]. En dicho artículo, ellos demuestran que el costo total de almacenamiento en un horizonte de pedido es acotado superiormente por una cantidad proporcional al costo de preparación.

El objetivo principal de este estudio, es describir los resultados de Heuvel y Wagelmans de forma más detallada y más explícita, ampliando el marco teórico y programando el algoritmo propuesto en el artículo.

Capítulo 1

Preliminares

El objetivo final de un modelo de inventario generalizado es responder a dos sencillas preguntas:

a. ¿Que cantidad de artículos deben perdirse?

b. ¿Cuando debe realizarse el pedido?

Las respuestas a estas dos interrogantes, pueden determinarse minimizando el costo total del inventario, el cual se puede expresar como una función de estas dos variables. Esta función generalmente viene dada de la siguiente forma:

$$\begin{pmatrix} \text{costo total} \\ \text{del inventario} \end{pmatrix} = \begin{pmatrix} \text{costo de} \\ \text{compra} \end{pmatrix} + \begin{pmatrix} \text{costo de} \\ \text{preparación} \end{pmatrix} + \begin{pmatrix} \text{costo de} \\ \text{almacenamiento} \end{pmatrix} + \begin{pmatrix} \text{costo por} \\ \text{faltantes} \end{pmatrix}.$$

Los modelos de inventario se dividen en cuatro tipos dependiendo del patrón de la demanda, estos son:

a. Los modelos deterministas estáticos, donde la demanda es conocida y constante a lo largo del tiempo.

b. Los modelos deterministas pero dinámicos, donde si bien la demanda se conoce para cada periodo (por ejemplo mensualmente), la misma varía de periodo en periodo.

c. Los modelos probabilísticos estacionarios, que son aquellos donde la función de densidad de probabilidad de la demanda se mantiente sin cambio en el tiempo.

d. Los modelos probabilísticos no estacionarios, donde la función de densidad de probabilidad varía con el tiempo.

En este estudio, se van a considerar los modelos de inventario con demandas deterministas, es decir, aquellos en donde las demandas son conocidas a lo largo del tiempo. En particular, se presenta a continuación, el problema clásico de la cantidad de pedido económico, y más adelante se presentará el modelo de dimensionamiento del lote económico (ELS), en el cual se supone una demanda variable para cada periodo.

1.1. Modelo clásico de la cantidad de pedido económico (EOQ)

El problema de la cantidad de pedido económico es uno de los modelos de inventarios mas simples conocidos. En este modelo se supone una demanda de consumo constante a lo largo del tiempo, una reposición de inventario instantánea y no se permiten faltantes. Además, la revisión del inventario se realiza de manera continua, y los costo de compra por unidad permanecen constantes en todo el horizonte de tiempo.

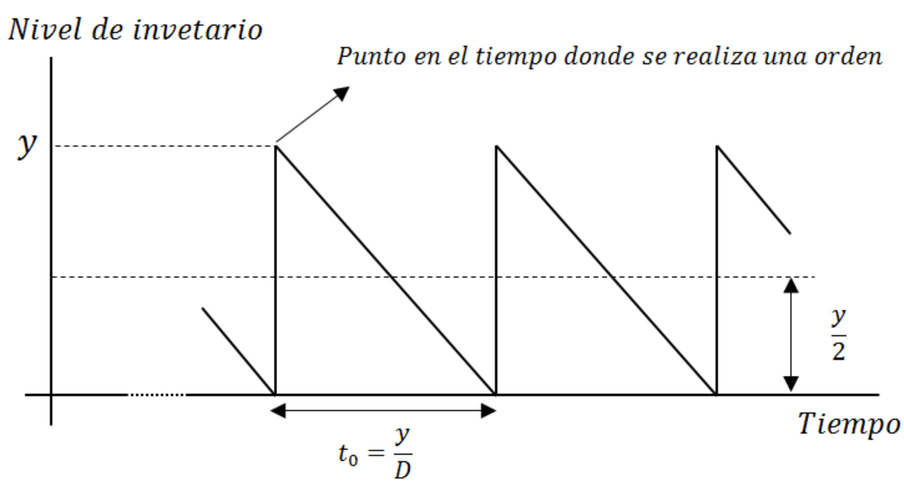

Figura 1.1: Variación del nivel de inventario

En el modelo EOQ, se considera que el nivel de inventario varía a una razón de D unidades por unidad de tiempo. Además, como se muestra en la figura 1.1, el punto mas alto en el nivel del inventario, el cual denotamos por y, ocurre cuando llega una nueva orden.

En este modelo se supone que la cantidad y se repone de manera instantánea y se observa que el tiempo t_0 que tarda en agotarse completamente una orden, luego de haberse realizado un pedido, es de y/D unidades de tiempo.

Por otro lado, si el costo de preparación o el costo por realizar un pedido es K, y el costo de almacenar una unidad de un artículo por una unidad de tiempo (usualmente por un año) es de h, entonces el costo total del inventario por unidad de tiempo (TCU, por sus siglas en inglés) se expresa como una función de la variable y. Es decir;

$$\begin{aligned}
TCU(y) =& \begin{pmatrix} \textit{costo de preparación} \\ \textit{por unidad de tiempo} \end{pmatrix} + \begin{pmatrix} \textit{costo de almacenamiento} \\ \textit{por unidad de tiempo} \end{pmatrix} \\
=& \frac{K}{t_0} + \frac{h(\frac{y}{2})t_0}{t_0} \\
=& \frac{K}{\frac{y}{D}} + h(\frac{y}{2}) \\
TCU(y) =& \frac{KD}{y} + \frac{h}{2}y. \qquad (1.1)
\end{aligned}$$

La ecuación (1.1) representa el costo total del inventario por unidad de tiempo, en función de la variable y. Sin embargo, para obtener esta función, no se consideró el costo por faltantes ni el costo de compra. Esto se debe, al hecho de que en este modelo no se permiten faltantes; de manera que este costo debe ser nulo en la ecuación. Además, como el costo de compra por unidad de inventario permanece sin cambio a lo largo del tiempo, este costo no tiene relevancia al momento de la busquedad de un valor mínimo para TCU.

Partiendo de los supuestos anteriores, una tarea prioritaria, es conseguir el valor y^* que minimiza la función TCU. Para ello, se deriva la función TCU respecto a y y luego se iguala a cero la derivada. Esto es:

$$\frac{\mathrm{d}}{\mathrm{d}y}TCU(y) = 0,$$

lo implica que:

$$-\frac{KD}{y^2} + \frac{h}{2} = 0.$$

Luego, despejando y de esta última ecuación se obtiene:

$$y = \sqrt{\frac{2KD}{h}}$$

.

Así, considerando que la función TCU es convexa en el intervalo $(0, \infty)$, se puede concluir, que el punto que minimiza a TCU es

$$y^* = \sqrt{\frac{2KD}{h}}. \tag{1.2}$$

En otras palabras, la ecuación (1.2) representa la cantidad que debe ordenarse para que el costo total del inventario tenga el mínimo valor posible. Esta cantidad debe ordenarse cada $t_0^* = \frac{y^*}{D}$ unidades de tiempo.

Por último, para finalizar esta sección, se va a deducir la conocida propiedad de balanceamiento perfecto, entre el costo de preparación K y el costo de almacenamiento h en una solución óptima. Para obtener esto, se supone que se realiza un pedido de y^* unidades de un artículo cada $t_0 = \frac{y^*}{D}$ unidades de tiempo. De esta manera, el costo de preparación por unidad de tiempo viene dado como sigue:

$$\begin{aligned}
\frac{K}{t_0^*} =& \frac{K}{\frac{\sqrt{\frac{2KD}{h}}}{D}} \\
=& DK\sqrt{\frac{h}{2KD}} \\
=& DK\sqrt{\frac{h}{2KD}}\sqrt{\frac{2Kh}{2Kh}} \\
=& hD\sqrt{\frac{2K}{hD}} \\
=& \frac{1}{2}hDt_0^*.
\end{aligned}$$

Por lo tanto,

$$\frac{K}{t_0^*} = \frac{1}{2}hDt_0^*. \tag{1.3}$$

La ecuación (1.3), define una propiedad entre el costo de preparación y el costo de almacenamiento, la cual es utilizada en la construcción de algunas heurísticas para el problema de dimensionamiento del lote económico (ELS). Este modelo se va a estudiar en la próxima sección.

Observación 1.1. Para calcular el costo de almacenamiento por unidad de tiempo en la ecuación (1.1), se toma en cuenta el nivel del inventario promedio $\frac{y}{2}$. Por lo tanto, la función TCU, en realidad representa el costo promedio total del inventario.

1.2. Modelo de dimensionamiento del lote económico (ELS)

El problema de dimensionamiento del lote económico (ELS, por sus siglas en inglés), fue presentado por primera vez en el año 1958 por Wagner y Whitin [4]. En el mismo se considera un horizonte de tiempo finito y discreto, digamos de N periodos, en donde cada periodo t, con $t = 1, \ldots, N$, posee una demanda d_t, cuyo valor se conoce de antemano.

En contraste con el modelo EOQ, el problema ELS posee una demanda que varía de periodo en periodo. Además, el instante en donde se realiza un nuevo pedido suele coincidir con el comienzo de cada periodo.

Ante la situación planteada, la finalidad del modelo ELS es minimizar el costo total del inventario, es decir, queremos saber la respuesta a estas dos interrogantes: ¿que cantidad de unidades debe pedirse? y ¿cuándo debe realizarse las ordenes?. A continuación, se presentan los modelos matemáticos que ayudan a responder estas dos preguntas. La dificultad del modelo depende mayormente del tipo de función de costo que se este considerando.

1.2.1. Modelo matemático con funciones de costos arbitrarias

Se presentará el modelo matemático de inventarios, bajo demanda determinista dinámica, un tanto más general que la situación típica para los resultados de acotamiento desarrollados por Van den Heuvel y Albert Wagelmans [6].

Se define para cada $t \in \{1, \ldots, N\}$ los siguientes parámetros:

$$\begin{aligned} d_t =& \textit{cantidad demandada en el periodo } t \\ K_t =& \textit{costo de realizar un pedido en el periodo } t \\ h_t =& \textit{costo de almacenar una unidad desde el periodo } t \textit{ hasta el } t\textit{+1} \\ z_t =& \textit{cantidad ordenada en el periodo } t \\ x_t =& \textit{nivel de inventario al inicio del periodo } t \\ c_t(z_t) =& \textit{función de costo de compra por unidad} \end{aligned}$$

La función de costo de compra para el periodo t viene dada por:

$$C_t(z_t) = \begin{cases} 0, & \text{si} \quad z_t = 0 \\ K_t + c_t(z_t), & \text{si} \quad z_t > 0 \end{cases} \tag{1.4}$$

donde la función $c_t(z_t)$ es de interés, sólo cuando el costo de compra unitario varía de periodo en periodo, o si existen descuentos por cantidad.

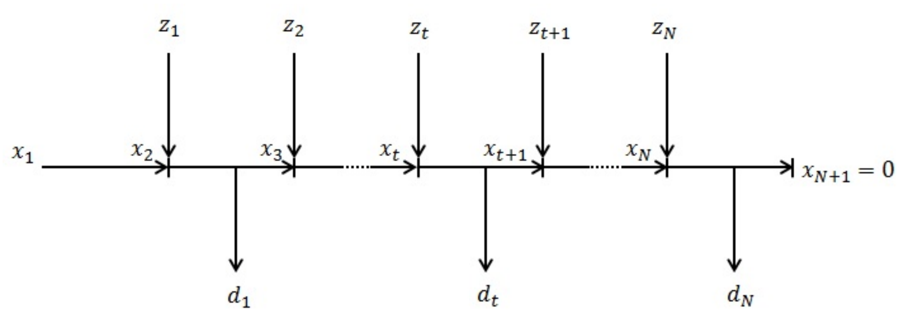

Figura 1.2: Modelo de programación dinámica

Para la solución de este problema, se considera un algoritmo de programación dinámica, el cual se represent gráficamente en la figura 1.2. El mismo consiste en dividir el problema en N subproblemas o etapas, donde en cada etapa se busca minimizar los costos de compra y de almacenamiento, para luego comparar el resultado con la etapa siguiente.

A fin de simplificar, se supone que el costo de almacenamiento esta basado en el inventario final de cada periodo, y además, el costo de almacenamiento se considera proporcional a esa cantidad. Es decir, si la cantidad de inventario que entra en el periodo $t+1$ se define por:

$$x_{t+1} = x_t + z_t - d_t,$$

entonces el costo de mantener la cantidad x_{t+1} desde el periodo t al periodo $t+1$ es de $h_t x_{t+1}$.

Ahora bien, para definir el algoritmo de avance que va a resolver este problema, se tiene en cuenta que cada periodo representará una etapa en el algoritmo de programación dinámica, mientras que la cantidad x_{t+1}, es el estado del sistema en la etapa o periodo t.

La variable x_{t+1} es la cantidad de inventario restante en el periodo t, luego de consumirse la demanda d_t. Debido a esto, el valor mínimo de x_t es cero y puede ser a lo mas $d_{t+1} + \cdots + d_N$ unidades de inventario. Por lo tanto, las distintas alternativas del sistema en la etapa t, se evaluan desde cero hasta $d_{t+1} + \cdots + d_N$.

Sea $f_t(x_{t+1})$ el costo mínimo del inventario para los periodos $1, \ldots, t$, dada la cantidad x_{t+1} de inventario final del periodo t. El algoritmo de avance que resuelve el modelo viene dado por:

$$\begin{aligned} f_1(x_2) &= \min_{z_1 = d_1 + x_2 - x_1} \{C_1(z_1) + h_1 x_2\} \\ f_t(x_{t+1}) &= \min_{0 \le z_t \le d_t + x_{t+1}} \{C_t(z_t) + h_t x_{t+1} + f_{t-1}(x_{t+1} + d_t - z_t)\}, \end{aligned} \tag{1.5}$$

con $t = 2, 3, \ldots, N$.

Observación 1.2. Por un lado, se tiene que en el periodo 1, el valor de z_1 es exactamente $d_1 + x_2 - x_1$ unidades de inventario; sin embargo cuando $t > 1$, la cantidad ordenada z_t puede ser cero, es decir, que no sería óptimo realizar un pedido en el periodo t. Esto se debe a que la demanda d_t podría satisfacerse por una orden anterior.

1.2.2. Modelo matemático con costos marginales constantes o decrecientes

En economía y finanzas, el costo marginal mide la tasa de variación del costo de producción entre la variación de la producción; también se suele expresar como el incremento que sufre el costo cuando se incrementa la producción en una unidad. Matemáticamente, la función de costo marginal se expresa como la derivada de la función de costo total.

El modelo anterior de programación dinámica, es válido para cualquier función de costo. Un caso particular de este modelo, es considerado cuando para un periodo t, el costo de compra y el costo de almacenamiento por unidad, son funciones constantes o decrecientes de las variables z_t y x_t respectivamente. En este caso, la función de costo proporcionará costos marginales constantes o decrecientes.

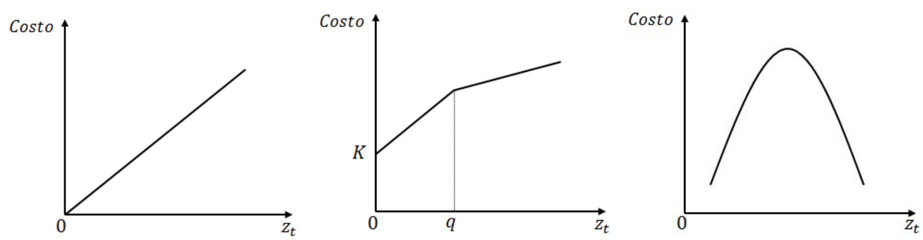

Figura 1.3: Funciones de costos marginales

En la figura 1.3 se observa, en la gráfica de la izquierda, una situación en donde la función de costo total posee un costo marginal constante; esto ocurre en ocasiones donde no existen descuentos por cantidad producida; es decir, el costo de compra por unidad es el mismo, independientemente de la cantidad que se ordene.

Por otro lado, la gráfica situada en el centro de la figura 1.3 muestra una función de costo, donde para un pedido de $z_t > q$ unidades, existe un descuento en el costo por unidad. Por lo tanto, el costo marginal tiende a decrecer. Por último, la gráfica de la derecha en la figura 1.3, ilustra una función de costo cóncava en general.

Dada las condiciones anteriores, se pueden demostrar ciertas propiedades que sirven para reducir los cálculos en la obtención de una solución óptima. A continuación se presenta un teorema que fue introducido por primera vez por Wagner y Whitin en el artículo “dynamic version of the economic lot-size model [4].

Teorema 1.1. *Dado el inventario inicial* $x_1 = 0$*, existe una secuencia óptima* z_1^*, $z_2^*, \ldots, z_N^*$ *que minimiza el costo total del inventario tal que* $x_t^* z_t^* = 0$ *para todo* $t \in \{1, \ldots, N\}$.

Demostración. Primero se considera el caso en el que los costos de compra unitarios y los costos de almacenamiento por unidad, son costantes e idénticos para todo $t \in \{1, \ldots, N\}$. Además, se supone que existe un $t_0 \in \{1, \ldots, N\}$ tal que $x_{t_0}^* z_{t_0}^* > 0$. Esto quiere decir que existe un periodo t_0, donde se realiza un pedido y se trae un inventario del periodo anterior.

Ahora bien, el costo de mantener la cantidad $x_{t_0}^*$ desde el periodo $t_0 - 1$ hasta t_0 es de $hx_{t_0}^*$. Además la cantidad $x_{t_0}^*$ generó un costo de compra en algún periodo $t < t_0$. Así al realizar un aumento de $x_{t_0}^*$ en el pedido $z_{t_0}^*$ tenemos que los costos totales de compra se mantienen sin variación. Por lo tanto, se obtiene un ahorro de $hx_{t_0}^*$ en los costos totales del inventario. Esto contradice el hecho de que z_1^*, $z_2^*, \ldots, z_N^*$ minimice los costos totales del inventario.

Se supone ahora, que los costos de compra y los costos de almacenamiento unitarios, estan dados por funciones cóncavas diferentes para cada periodo, y que $x_{t_0}^* z_{t_0}^* > 0$ para algún periodo $t_0 \in \{1, \ldots, N\}$. Sea b_l el costo de adquirir y almacenar la cantidad $x_{t_0}^*$ desde un periodo $l < t_0$. Se denota b_{t_0}, el costo de ordenar la misma cantidad $x_{t_0}^*$ en t_0.

Si $b_l < b_{t_0}$, entonces es mas económico ordenar una cantidad que satisfaga las demandas desde el periodo l hasta t_0; esto se debe a que las funciones de costos marginales son decrecientes. Pero Si $b_l \geq b_{t_0}$, entonces sería mas económico adquirir $x_{t_0}^*$ en el periodo t_0; esto se debe nuevamente a la particularidad de las funciones de costos marginales. En conclusión, dado que los costos marginales son constantes o decrecientes, la condición $x_t^* z_t^* = 0$ para todo $t \in \{1, \ldots, N\}$ no proporcionará una solución peor. ∎

Dos colorarios siguen del teorema 1.1.

Corolario 1.1. *Existe una secuencia óptima z_1^*, $z_2^* \ldots$, z_N^* tal que para todo $t \in \{1,\ldots,N\}$ se tiene que $z_t^* = 0$ ó $z_t^* = \sum_{i=t}^{k} d_i$ para algún k con $t \leq k \leq N$.*

Corolario 1.2. *Existe una secuencia óptima tal que: si para algún periodo $t_0 \in \{1,\ldots,N\}$, la demanda d_{t_0} es satisfecha por $z_{t_0^*}$ con $t_0^* < t_0$, entonces las demandas $d_{t_0^*}$, $d_{t_0^*+1}, \ldots$, d_{t_0-1} son satisfechas tambien por $z_{t_0^*}$.*

Observación 1.3. Para una demostración de los corolarios 1 y 2 puede verse el artículo [4] publicado por Wagner y Whitin en 1958.

Los corolarios 1.1 y 1.2 indican por un lado que para un periodo t, la cantidad ordenada z_t, si es positiva debe satisfacer un número exacto de periodos consecutivos; mientras que si z_t es cero, la demanda del periodo t debe ser satisfecha totalmente por un pedido anterior.

Gracias a estos resultados, se obtiene un ahorro importante en el número de cálculos del algoritmo de avance. Esto se debe principalmente, a que el estado en el algoritmo de avance esta determinado por las cantidades demandadas en los periodos siguientes. Sin embargo, para el caso donde existen costos marginales constantes o decrecientes, se considera para los estados del algoritmo, el número de periodos subsecuentes y no la cantidad demandada en dichos periodos.

Cabe agregar, que en el teorema 1.1 se dio la condición de que el inventario inicial debe ser cero, sin embargo, si nos encontramos con un problema donde $x_1 > 0$, entonces esta cantidad puede restarse de las demandas posteriores hasta que se agote la existencia. Los periodos donde la demanda se ha satisfecho aún pueden incluirse en el problema, pero dichas demandas deben ser nulas. Esto se ilustrará con un ejemplo.

Ejemplo 1.1. Considere el siguiente modelo de inventario de cuatro periodos

Periodo t	d_t	$K_t(\$)$	$h_t(\$)$
1	76	98	1
2	26	114	1
3	90	185	1
4	67	70	1

El inventario inicial es de 80 unidades y el costo de compra por unidad es constante e igual a \$2.00 para todos los periodos.

Se utilizará el algoritmo de avance para obtener la solución óptima, y se considerarán las propiedades demostradas anteriormente. Dado que $x_1 = 80$, la demanda d_1 es satisfecha por esta cantidad. Además $d_2 = 26 - 4 = 22$. Luego, el nuevo problema es:

Periodo t	d_t	$K_t(\$)$	$h_t(\$)$
1	0	98	1
2	22	114	1
3	90	185	1
4	67	70	1

donde el inventario inicial se considera nulo. En este caso la función de costo se define por:

$$C_t(z_t) = \begin{cases} 0, & \text{si } z_t = 0 \\ K_t + 2z_t, & \text{si } z_t > 0 \end{cases}$$

para todo $t \in \{1, 2, 3, 4\}$.

Etapa 1: $d_1 = 0$, $z_1 = d_1 + x_2 - x_1$.

		$f_1(x_2) = C_1(z_1) + h_1x_2$					
		$z_1 = 0$	22	112	179	Solución óptima	
x_2	h_1x_2	$C_1(z_1)$=0	142	322	456	$f_1(x_2)$	z_1^*
0	0	0	-	-	-	0	0
22	22	-	164	-	-	164	22
112	112	-	-	434	-	434	112
179	179	-	-	-	635	635	179
Pedido en 1 para		1	1,2	1,2,3	1,2,3,4		

Etapa 2: $d_2 = 22$, $0 \leq z_2 \leq d_2 + x_3 = 22 + x_3$.

		$f_2(x_3) = C_2(z_2) + h_2x_3 + f_1(x_3 + d_2 - z_2)$					
		$z_1 = 0$	22	112	179	Solución óptima	
x_3	h_2x_3	$C_2(z_2)$=0	158	338	472	$f_2(x_3)$	z_2^*
0	0	0+164=164	158+0=158	-	-	158	22
90	90	90+434=524	-	428+0=428	-	428	112
157	157	157+635=792	-	-	629+0=629	629	179
Pedido en 2 para		-	2	2,3	2,3,4		

Etapa 3: $d_3 = 90$, $0 \leq z_3 \leq d_3 + x_4 = 90 + x_4$.

		$f_3(x_4) = C_3(z_3) + h_3x_4 + f_2(x_4 + d_3 - z_3)$				
		$z_3 = 0$	90	157	Solución óptima	
x_4	h_3x_4	$C_3(z_3)$=0	365	499	$f_3(x_4)$	z_3^*
0	0	0+428=428	365+158=523	-	428	0
67	67	67+629=696	-	566+158=724	696	0
Pedido en 3 para		-	3	3,4		

Etapa 4: $d_4 = 67$, $x_5 = 0$, $0 \leq z_4 \leq d_4 + x_5 = 67$.

		$f_4(x_5) = C_4(z_4) + h_4x_5 + f_3(x_5 + d_4 - z_4)$			
		$z_4 = 0$	67	Solución óptima	
x_5	h_4x_5	$C_4(z_4)$=0	204	$f_4(x_5)$	z_4^*
0	0	0+696=696	204+428=632	632	67
Pedido en 4 para		-	4		

La solución óptima viene dada de la siguiente forma:

$(x_5 = 0) \longrightarrow \boxed{z_4^* = 67} \longrightarrow (x_4 = 0 + 67 - 67 = 0) \longrightarrow \boxed{z_3^* = 0} \longrightarrow (x_3 = 0 + 90 - 0 = 90) \longrightarrow \boxed{z_2^* = 112} \longrightarrow (x_2 = 90 + 22 - 112 = 0) \longrightarrow \boxed{z_1^* = 0}$.

1.2.3. Modelo con funciones de costos de compras afines

Un caso particular de una función de costo de compra, es cuando ésta es una función lineal o afín; es decir que para cada $t \in \{1, \ldots, N\}$ se tiene que:

$$C_t(z_t) = K_t + c_t z_t,$$

donde $c_{t+1} \geq c_t$, para todo $t \in \{1, \ldots, N-1\}$.

Bajo estas condiciones, el algoritmo de avance se puede modificar, de manera que se obtenga un ahorro significativo en los cálculos. Este nuevo algoritmo modificado, a diferencia del original, define cada etapa t de manera que para el periodo t, la política óptima se determina considerando los pedidos realizados en los periodos anteriores, hasta el mismo periodo t. Matemáticamente el algoritmo modificado se define como sigue:

$$f(t) = min \begin{cases} \underset{1 \leq i < t}{min} \left[C_i + \sum_{k=i}^{t-1} \sum_{p=k+1}^{t} h_k d_p + f(i-1) \right] \\ C_t + f(t-1) \end{cases} \tag{1.6}$$

donde para algún periodo k, la función $f(k)$ representa el costo mínimo total para los periodos $1, 2, \ldots, k$; mientras que C_i es el costo total de realizar un pedido en el periodo i para traer la cantidad $z_i = d_i + d_{i+1} \cdots + d_k$ del periodo i al k.

Observación 1.4. En el modelo original, el número máximo de cálculos que deben realizarse para obtener una solución óptima es de $\frac{N(N+1)+N(N-1)}{2} = N^2$; mientras que en el modelo modificado el número de registros es de $\frac{N(N+1)}{2}$. Sin embargo, este número puede reducirse aún mas, si consideramos el siguiente teorema.

Teorema del horizonte de planeación: Si para un periodo t^* el mínimo en (1.6) ocurre de tal modo que la demanda d_{t^*} es satisfecha desde un periodo $t^{**} < t^*$, entonces para los periodos $t > t^*$, es suficiente buscar la solución óptima considerando solo los periodos t^{**}, $t^{**}+1, \ldots, t$. En particular, si $t^* = t^{**}$, entonces para cualquier periodo futuro $t > t^*$ siempre será óptimo ordenar en t^*. En este caso se dice que t^* marca el comienzo de un horizonte de planeación.

Demostración. Se considera sin pérdida de optimalidad, un programa óptimo que satisface las características supuestas en el teorema. Supongamos además, que el programa sugiere que se debe considerar un pedido en el periodo $t^{***} < t^{**}$ para el periodo t^*; entonces por el corolario 1.2, la demanda $d_{t^{**}}$ debe ser satisfecha también por un pedido en t^{***}. Luego se tiene un inventario de entrada positivo y un pedido positivo en t^{**}, lo cual contradice el teorema 1.1. ■

El teorema anterior implica dos importantes conceptos:

a. Los cálculos en (1.6) pueden ser realizados solo para $k > t^{**}$. Esto conduce a un ahorro en el número de operaciones del ordenador.

b. En el caso particular, donde $t^{**} = t^*$; además de truncar los cálculos en t^*, los periodos futuros comenzando por t^* serán considerados independientes de todos los periodos anteriores.

En el caso donde $t^{**} < t^*$, la política óptima puede requerir cambios en las demandas futuras. En este caso, t^{**} se denominará el periodo de inicio de un subhorizonte siempre que $t^{**} < t^*$.

Capítulo 2

Cota para el costo total de almacenamiento del modelo ELS

En el capítulo 1, se probó que existe una solución óptima para el problema de dimensionamiento del lote económico (ELS), que satisface la propiedad de inventario cero; esto quiere decir que en una solución óptima, un pedido es realizado cuando el nivel del inventario cae a cero. Esto se demostró partiendo del hecho de que para un periodo t, los costos de compra y de almacenamiento por unidad, son funciones constantes o decrecientes de z_t y x_{t+1} respectivamente.

En este capítulo, se va a deducir una cota superior para el costo de almacenamiento, en un horizonte de orden o horizonte de pedido; donde un *horizonte de orden* se define como el número de periodos en los que las demandas son satisfechas por un único pedido.

Hecha la observación anterior, en este capítulo se va a considerar el caso particular donde los costos de compra, de almacenamiento y de preparación son constantes e indénticos para todos los periodos. Así que el costo de compra deja de tener relevancia en la obtención de una solución óptima.

Ante la situación planteada, se denota por K a el costo de realizar una orden y como h a el costo de almacenamiento del modelo ELS. Se supone además, sin perdida de generalidad, que un horizonte de pedido de una solución óptima contiene los periodos $1, 2, \ldots, t$. Se procederá a deducir una cota para el costo total de almacenamiento en este horizonte de orden.

Sea $r_i = d_{i+1} + \cdots + d_t$ la cantidad de inventario restante para algún periodo $i \in \{1, \ldots, t-1\}$ (como se muestra en la figura 2.1), luego de que se ha consumido la cantidad d_i. Entonces, el costo de almacenar r_i durante el periodo i es de hr_i.

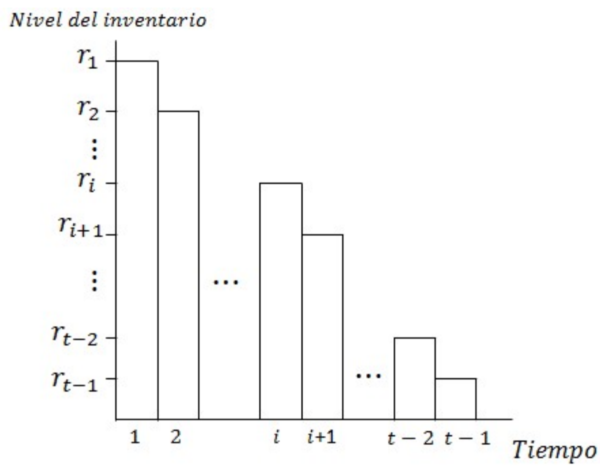

Figura 2.1: Variación del inventario en un horizonte de pedido

Por lo tanto, el costo total de almacenamiento viene dado por:

$$\begin{aligned} H_t &= hr_1 + hr_2 + \cdots + hr_{t-1} \\ &= h(d_2 + \cdots + d_t) + h(d_3 + \cdots + d_t) + \cdots + h(d_i + \cdots + d_t) + \cdots + h(d_t) \\ &= hd_2 + h2d_3 + \cdots + h(i-1)d_i + \cdots + h(t-1)d_t \\ &= h\sum_{i=2}^{t}(i-1)d_i \end{aligned}$$

Luego;

$$H_t = h\sum_{i=2}^{t}(i-1)d_i \tag{2.1}$$

Si se realiza un pedido en algún periodo $p+1$, con $p \in \{2, \ldots, t-1\}$, de tal forma que esta nueva orden modifique la solución óptima, es decir, un pedido realizado en el periodo 1 cubre las demandas desde 1 hasta p; mientras que la cantidad ordenada en $p+1$ debe satisfacer desde $p+1$ hasta el último periodo t.

En la figura 2.2 se ilustra gráficamente esta situación. Se observa que al agregar una preparación en $p+1$, se obtiene un ahorro en el costo total de mantenimiento de la cantidad $d_{p+1} + \cdots + d_t$. Este ahorro es en cada uno de los periodos 1, 2, ... y p. Por lo tanto, como en el horizonte de pedido de la solución óptima, esta cantidad se almacena p veces, el descuento que se obtiene al realizar el pedido en $p+1$ es de

$$hp(d_{p+1} + \cdots + d_t) = ph \sum_{i=p+1}^{t} d_i \tag{2.2}$$

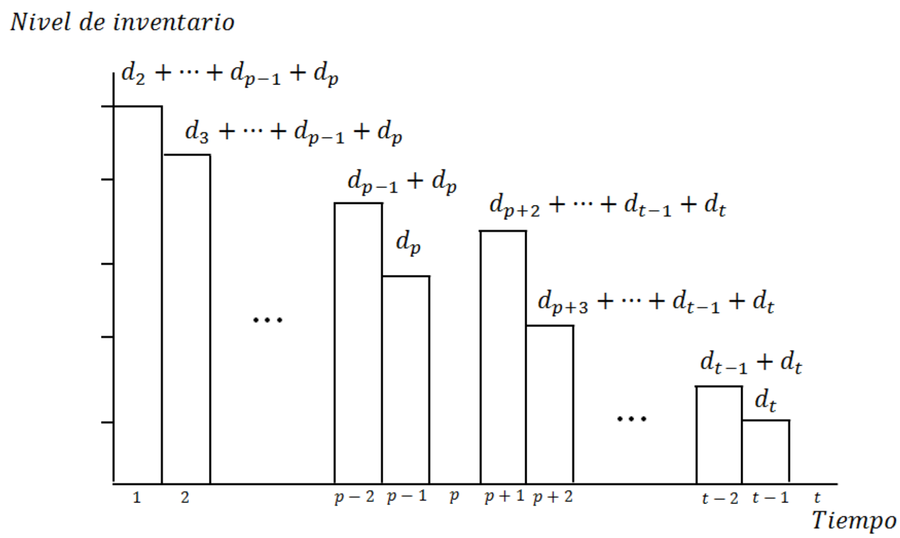

Figura 2.2: Variación del inventario con una orden en $p+1$

Los siguientes lemas que se presentarán a continuación, serán utilizados para demostrar el teorema principal de este estudio, el cual fue introducido en el año 2009 por Wilco Van Den Heuvel y Albert Wagelmans [6]. La idea de la prueba se basa en el hecho, de que ninguna orden adicional en un horizonte de pedido de una solución óptima, conduce a una reducción de los costos totales del inventario.

Lema 2.1. *Supongamos que existe una constante $c \geq 0$ y un periodo $p \in \{1, \ldots, t-1\}$ tal que*

$$cph \sum_{i=p+1}^{t} d_i \geq h \sum_{i=2}^{t} (i-1) d_i. \tag{2.3}$$

Entonces en una solución óptima se tiene que $H_t \leq cK$.

Demostración. Sea el conjunto $\{1, \ldots, t\}$ un horizonte de pedido arbitrario de una solución óptima para el modelo ELS. Entonces, como se demostró anteriormente, al agregar un pedido en el periodo $p+1$ con $p \in \{1, \ldots, t-1\}$, se obtiene un descuento en el costo total de almacenamiento de $ph \sum_{i=p+1}^{t} d_i$ unidades. Además, debido a que se esta considerando un programa óptimo, esta cantidad no debe ser mayor que el costo de realizar una orden en $p+1$. En otras palabras, para todo $p \in \{1, \ldots, t-1\}$ se tiene que

$$ph \sum_{i=p+1}^{t} d_i \leq K.$$

Luego, por (2.2) y (2.3),

$$H_t = h \sum_{i=2}^{t} (i-1) d_i \leq cph \sum_{i=p+1}^{t} d_i \leq cK.$$

■

El Lema 2.1 se interpreta así: si se puede encontrar un $c \geq 0$, tal que la desigualdad (2.3) se cumpla para cualquier secuencia de demandas $\mathbf{d}$=$d_1, \ldots, d_t$, entonces se ha encontrado una cota para el costo total de almacenamiento en un horizonte de pedido de una solución óptima.

Por otro lado, en el horizonte de pedido $\{1, \ldots, t\}$, el mayor incremento que puede sufrir el costo total de almacenamiento es cuando se realiza una orden en cada periodo. Es decir;

$$K + h \sum_{i=2}^{t} (i-1) d_i \leq tK.$$

Así,

$$h \sum_{i=2}^{t} (i-1) d_i \leq (t-1)K.$$

Por lo tanto, una cota superior para la constante c debe ser $t-1$. Esto es, cuando en la desigualdad (2.3) se toma $c = t-1$ y $p = 1$, entonces $H_t \leq cK$.

En la observación de arriba, se ha obtenido una cota trivial para H_t en una solución óptima, sin embargo el objetivo es encontrar una cota superior para el costo total de almacenamiento, dada una secuencia de demandas arbitrarias, que sea menor que esta cota trivial. En el Lema siguiente, se obtiene una cota superior de c, dada una secuencia

$\mathbf{d}$=$d_1, \ldots, d_t$ particular, mientras que en el Lema 2.3 se demuestra que esta cota funciona dada cualquier secuencia de demanda.

Lema 2.2. *Sea* $\mathbf{d}^0 = d_1^0, \ldots, d_t^0$ *una secuencia de demandas definidas por:*

$$d_1^0 > 0, \quad d_i^0 = \tfrac{1}{i(i-1)} = \tfrac{1}{i-1} - \tfrac{1}{i}, \quad \textit{con} \quad i = 2, \ldots, t-1, \quad d_t^0 = \tfrac{1}{t-1}.$$

Entonces, para la secuencia $\mathbf{d}^0$ *se cumple que:*

$$\left(\sum_{i=1}^{t-1} \frac{1}{i}\right) p \sum_{i=p+1}^{t} d_i^0 = \sum_{i=2}^{t} (i-1) d_i^0, \quad \textit{para todo} \quad p = 1, \ldots, t-1.$$

Demostración. Por un lado, tomando en cuenta que la demanda en el i-ésimo periodo es $d_i^0 = \frac{1}{i(i-1)}$, y que en el perido t es $d_t^0 = \frac{1}{t-1}$ se tiene que:

$$\begin{aligned}
\sum_{i=2}^{t} (i-1) d_i^0 &= \sum_{i=2}^{t-1} (i-1) d_i^0 + (t-1) d_t^0 \\
&= \sum_{i=2}^{t-1} (i-1) \frac{1}{i(i-1)} + (t-1) \frac{1}{t-1} \\
&= \sum_{i=2}^{t-1} \frac{1}{i} + 1 \\
&= \sum_{i=1}^{t-1} \frac{1}{i}.
\end{aligned}$$

Por otro lado, para cualquier periodo $p \in \{1, \ldots, t-1\}$ se tiene que

$$\begin{aligned}
p \sum_{i=p+1}^{t} d_i^0 = & \quad p \left[\sum_{i=p+1}^{t-1} \left(\frac{1}{i-1} - \frac{1}{i} \right) \right] \\
= & \quad p \left[\left(\tfrac{1}{p} - \tfrac{1}{t-1} \right) + \tfrac{1}{t-1} \right] = 1.
\end{aligned}$$

Por lo tanto,

$$\left(\sum_{i=1}^{t-1} \frac{1}{i}\right) p \sum_{i=p+1}^{t} d_i^0 = \left(\sum_{i=1}^{t-1} \frac{1}{i}\right) 1 = \sum_{i=2}^{t} (i-1) d_i^0,$$

lo cual demuestra el Lema. ■

Note que si en un horizonte de pedido de una solución óptima, la secuencia de demandas correspondientes a los periodos de este horizonte, estan definida por $\mathbf{d}^0$, entonces cualquier pedido que se agregue en $p+1$, con $p \in \{1, \ldots, t-1\}$ conduce a la misma reducción en el costo total almacenamiento. En otras palabras, cualquier pedido que se haga en $p+1$ conduce a una reducción del costo de una unidad.

Cabe agregar, que en el Lema 2.2 se demostró que para esta secuencia de demanda en particular, existe una constante $c = \sum_{i=1}^{t-1} \frac{1}{i}$, tal que la igualdad en la inecuación (2.3) se satisface. Esto quiere decir que para las demadas $\mathbf{d}^0$, el valor mas pequeño que puede tomar la costante c de manera que se satisfaga (2.3) es

$$\sum_{i=1}^{t-1} \frac{1}{i}$$

.

Además, se observa que cuando H_t es finito, siempre es posible encontrar un valor de c suficientemente grande que satisfaga (2.3); sin embargo, debido al Lema 2.2, para una demanda arbitraria debemos tener $c \geq \sum_{i=1}^{t-1} \frac{1}{i}$, porque de lo contrario, cuando se tenga un caso particular donde la demanda esta dada por $\mathbf{d}^0$, la desigualdad (2.3) no se cumpliría. Lo cual contradice el Lema 2.2.

El siguiente Lema prueba que la cantidad $\sum_{i=1}^{t-1} \frac{1}{i}$ es una cota superior para el valor de c, cuando (2.3) se satisface para demandas arbitrarias.

Lema 2.3. *Para cualquier secuencia de demandas* $\mathbf{d} = d_1, \ldots, d_t$ *de una solución óptima, existe un periodo* $p \in \{1, \ldots, t-1\}$ *tal que*

$$\left(\sum_{i=1}^{t-1} \frac{1}{i} \right) p \sum_{i=p+1}^{t} d_i \geq \sum_{i=2}^{t} (i-1) d_i \tag{2.4}$$

Demostración. En el caso donde $d_i = 0$ para todo $i \in \{2, \ldots, t\}$, la desigualdad (2.4) se cumple trivialmente. Así que se supone que existe un periodo $i \in \{2, \ldots, t\}$, tal que $d_i > 0$. Entonces, $\sum_{i=2}^{t} (i-1) d_i > 0$.

Sea $\alpha = \dfrac{\sum_{i=2}^{t}(i-1)d_i}{\sum_{i=2}^{t}(i-1)d_i^0}$, entonces $\alpha > 0$, debido a la definición de $\mathbf{d}^0$. Esto implica que:

$$\sum_{i=2}^{t}(i-1)d_i = \alpha\sum_{i=2}^{t}(i-1)d_i^0. \tag{2.5}$$

Se define el vector $\Delta\mathbf{d} = \mathbf{d} - \alpha\mathbf{d}^0$. Entonces por (2.5) se tiene que:

$$\sum_{i=2}^{t}(i-1)\Delta d_i = \sum_{i=2}^{t}(i-1)(d_i - \alpha d_i^0) = 0.$$

Afirmación 2.1. *Existe un perido $p \in \{1,\ldots,t-1\}$ tal que*

$$p\sum_{i=p+1}^{t} d_i \geq p\sum_{i=p+1}^{t}\alpha d_i^0 \Leftrightarrow p\sum_{i=p+1}^{t}\Delta d_i \geq 0. \tag{2.6}$$

En efecto; se supone que (2.6) no se cumple, es decir, para todo $p \in \{1,\ldots t-1\}$ se tiene que

$$p\sum_{i=p+1}^{t}\Delta d_i < 0 \Leftrightarrow \sum_{i=p+1}^{t}\Delta d_i < 0.$$

Entonces

$$0 > \sum_{p=1}^{t-1}\sum_{i=p+1}^{t}\Delta d_i = \sum_{i=2}^{t}\sum_{p=1}^{i-1}\Delta d_i = \sum_{i=2}^{t}(i-1)\Delta d_i = 0.$$

Lo cual es una contradicción.

Ahora bien, debido a la afirmación anterior y al Lema 2.2, existe un periodo $p \in \{1,\ldots,t-1\}$ tal que

$$\begin{aligned}\left(\sum_{i=1}^{t-1}\frac{1}{i}\right)p\sum_{i=p+1}^{t} d_i &\geq \left(\sum_{i=1}^{t-1}\frac{1}{i}\right)p\sum_{i=p+1}^{t}\alpha d_i^0 \\ &= \alpha\sum_{i=2}^{t}(i-1)d_i^0 = \sum_{i=2}^{t}(i-1)d_i.\end{aligned}$$

Por lo tanto el Lema queda demostrado. ■

Puesto que se ha encontrado un cota superior para la constante c, dada cualquier secuencia de demandas arbitrarias, se procederá a enunciar y demostrar el teorema principal.

Teorema 2.1. *Sea $\{1,\ldots,t\}$ el conjunto de periodos en un horizonte de pedido de una solución óptima de un problema con una secuencia de demandas de $d_1, d_2, \ldots, d_t$. Entonces, para el costo total de almacenamiento se cumple que:*

$$H_t = h\sum_{i=2}^{t}(i-1)d_i \leq K\sum_{i=1}^{t-1}\frac{1}{i}. \tag{2.7}$$

Demostración. Por el Lema 2.3, existe un periodo $p \in \{1,\ldots,t-1\}$ tal que

$$\left(\sum_{i=1}^{t-1}\frac{1}{i}\right) ph \sum_{i=p+1}^{t} d_i \geq \sum_{i=2}^{t}(i-1)d_i,$$

Luego si se toma $c = \sum_{i=1}^{t-1}\frac{1}{i}$ en la desigualdad (2.3), se obtiene que

$$H_t = \sum_{i=2}^{t}(i-1)d_i \leq cK = K\sum_{i=1}^{t-1}\frac{1}{i}.$$

Por lo tanto, el teorema queda demostrado. ∎

En el teorema 2.1, se ha encontrado una cota superior para el costo total de almacenamiento en un horizonte de pedido de una solución óptima, para una secuencia de demandas arbitraria.

En un caso particular de N periodos, con costos $K = h = 1$ y una secuencia $\mathbf{d}^0$ definida como en el Lema 2.2. Es decir;

$$d_1^0 > 0, \quad d_i^0 = \tfrac{1}{i(i-1)} = \tfrac{1}{i-1} - \tfrac{1}{i}, \quad \textit{con} \quad i = 2, \ldots, N-1, \quad d_N^0 = \tfrac{1}{N-1}.$$

Se tiene que la solución óptima consiste en realizar un único pedido en el periodo 1 que satisfaga todas las demandas hasta el último periodo N. En efecto; si se agrega una orden en $p+1$, con $p = 1, \dots, N-1$, se obtiene un ahorro en el costo total de almacenamiento de $p \sum_{i=p+1}^{N} d_i^0$ unidades, pero

$$p \sum_{i=p+1}^{N} d_i^0 = p \left[\left(\frac{1}{p} - \frac{1}{N-1} \right) + \frac{1}{N-1} \right] = 1 = K.$$

De modo que, ningún pedido adicional mejorará la solución.

Por otro lado, si se agrega un pedido en un periodo $\widehat{p}+1$, despues de haber agregado previamente una orden en $p+1$, entonces en el costo total de almacenamiento, la cantidad $d_{\widehat{p}+1}^0 + \cdots + d_N^0$ se ahorra un número $\widehat{p} - (p+1) + 1$ de veces. De manera que el ahorro que se obtiene es

$$(\widehat{p} - p) \sum_{i=\widehat{p}+1}^{N} d_i^0.$$

Pero,

$$(\widehat{p} - p) \sum_{i=\widehat{p}+1}^{N} d_i^0 = (\widehat{p} - p) \frac{1}{\widehat{p}} = 1 - \frac{p}{\widehat{p}} < 1 = K,$$

pues $p < \widehat{p}$. Por lo tanto, agregar más de una orden en $\{2, \dots, N\}$ tampoco mejorará la solución. Así que cada vez que se agregue una orden adicional, los costos totales del inventario aumentarán. En conclusión, se ha probado que la mejor solución del problema, es realizar una única orden al incio del intervalo.

En vista de lo anterior, se tiene que la razón entre el costo de almacenamiento y el costo de preparación en el horizonte de pedido $\{1, \dots, N\}$, se hace aribitrariamente grande cuando N se hace grande. Esto es,

$$\frac{H_N}{K} = \sum_{i=1}^{N-1} \frac{1}{i} \to \infty \quad \textit{cuando} \quad N \to \infty$$

Ahora se esta en condiciones de enunciar el siguiente teorema. El cual demuestra que el criterio que se utiliza en la construcción de la heurística PPB no es cierto en general. La heurística PPB se estudiará en el siguiente capítulo.

Teorema 2.2. *Existe una instancia de problema para el cual la razón entre el costo de almacenamiento y el costo de preparación en una solución óptima es arbitrariamente grande.*

Capítulo 3

Heurísticas para el modelo ELS

En este capítulo se definen tres heurísticas para la resolución del problema de dimensionamiento del lote económico (ELS) basadas en la propiedad de perfecto balanceamiento del modelo EOQ. En estas heurísitcas los costos de preparación y de almacenamiento se suponen constante e idénticos para todos los periodos.

Luego, se procede a la construcción de una nueva heurística, la cual estará basada en la propiedad (2.7) deducida en el capítulo dos. Al final del capítulo se muestran los resultados de un pequeño análisis computacional que se realizó usando la herramienta MatLab, y cuyo objetivo fue comparar el rendimiento de esta nueva heurística con las otras tres.

3.1. Heurística de Silver-Meal (SM).

La heurística de Silver-Meal requiere la determinación del costo promedio por periodo como función del número de periodos que el pedido actual cubrirá. Los cálculos se detienen cuando esta función se incrementa.

Se define la función $C(t)$ como el costo total de preparar un pedido en el periodo 1 y mantenerlo durante los periodos $1, 2, \ldots, t$. Esta función viene dada por:

$$C(t) = K + hd_2 + 2hd_3 + 3hd_4 + \cdots + (t-1)hd_t.$$

Si se divide la cantidad $C(t)$ por el número de periodos trascurridos desde que se realizó la orden, se obtiene la función de costo total promedio por periodo $CP(t)$, la cual se define como

$$CP(t) = \frac{K + hd_2 + 2hd_3 + 3hd_4 + \cdots + (t-1)hd_t}{t}.$$

donde CP(1)=K.

El horizonte de pedido se obtiene al verificar en que punto la desigualdad $CP(t) > CP(t-1)$ no se satisface, en ese momento se detienen los cálculos, se programa una orden que satisfaga los periodos $1, 2, \ldots, t-1$ y se comienza nuevamente el procedimiento en el periodo t.

3.2. Heurística de Costo Unitario Mínimo (LUC).

.

La heurística de Costo Unitario Mínimo (LUC, Least Unit Cost), es similar al método de Silver-Meal, excepto que en lugar de dividir el costo para el periodo t entre el número de periodos, se divide entre la cantidad total de unidades demandadas a lo largo del periodo t. Se elige el horizonte de pedido que minimiza el costo por unidad de demanda en lugar del costo por periodo.

Se define $CPU(t)$ como el costo total promedio por unidad para un horizonte de pedido con t periodos. Esto es;

$$CPU(t) = \frac{K + hd_2 + 2h_3 + 3hd_4 + \cdots + (t-1)hd_t}{d_1 + d_2 + \cdots + d_t}$$

donde $CPU(1) = \frac{K}{d_1}$.

Como en el caso de la heurística de Silver-Meal, el cálculo se detiene cuando $CPU(t) > CPU(t-1)$ y se realiza una orden de la cantidad $d_1 + \cdots + d_{t-1}$. Luego, se repite el procedimiento iniciando en el periodo t.

3.3. Heurística de Balanceo de Periodos Parciales (PPB).

Este último método consiste en igualar el horizonte de pedido al número de periodos que más ajuste al costo total de almacenamiento con el costo de preparación en dicho periodo. El horizonte de pedido que iguale exactamente los costos de preparación y almacenamiento, dificilmente será un número entero de periodos, a ello se debe el nombre del método.

3.4. Nueva Heurística H^*

En el capítulo 2, se obtuvo una cota superior para el costo total de almacenamiento en un horizonte de pedido de una solución óptima, lo cual se demostró en el Teorema 2.1. Esto plantea una idea para la construcción de una nueva heurística para el modelo ELS, con parámetros de costos invariantes en el tiempo.

Por el teorema 2.1, se sabe que en cualquier horizonte de pedido de una solución óptima, el costo de almacenamiento satisface la desigualdad (2.7), así la nueva heurística, la cual se denotará como H, consistirá en encontrar horizonte de pedido que cumplan con esta propiedad. En otras palabras, la heurística H va a seleccionar un horizonte de pedido que cubra los periodos $1, \ldots, t$ de tal forma que:

$$h \sum_{i=2}^{t} (i-1) d_i \leq K \sum_{i=1}^{t-1} \frac{1}{i}. \tag{3.1}$$

Por lo tanto, cualquier solución que se encuentre utilizando H cumple con una propiedad que es satisfecha por cualquier solución óptima.

Ejemplo 3.1. Un taller de maquinado desea programar la producción por lotes para cubiertas de computadoras. En las siguiente cinco semanas las demandas de cubiertas son $\mathbf{d} = (18, 30, 42, 5, 20)$. El costo de almacenamiento es de \$2 por cubierta a la semana, y el costo de preparación es de \$80 por tanda de producción. Los datos del problema se resumen en la siguiente tabla.

Periodo t	d_t	$K_t(\$)$	$h_t(\$)$
1	18	80	2
2	30	80	2
3	42	80	2
4	5	80	2
5	20	80	2

Solución: Se utiliza la heurística H para encontrar un programa de producción para este problema. La busquedad consiste en probar hasta que periodo t se satisface (3.1). En el primer periodo se debe realizar una orden obligatoriamente, por lo tanto la heurística empieza la verificación en el segundo periodo.

Para $t = 2$,

$$h\sum_{i=2}^{t}(i-1)d_i = 2(1)30 = 60 \leq 80 = 80(1) = K\sum_{i=1}^{t-1}\frac{1}{i}.$$

Como esta última desigualdad se cumple, se procede a verificar el siguiente periodo.

Para $t = 3$,

$$h\sum_{i=2}^{t}(i-1)d_i = 2((1)30+(2)42) = 228 \leq 120 = 80(1+\frac{1}{2}) = K\sum_{i=1}^{t-1}\frac{1}{i}.$$

Debido a que la desigualdad anterior no es cierta, el primer horizonte de pedido contiene los periodos $\{1,2\}$. Es decir, la heurística H sugiere que se debe realizar un pedido en $t = 1$ que satisfaga las demandas en los periodos 1 y 2. Luego, se realiza un pedido en $t = 3$ y se procede a chequear el periodo $t = 4$. Note que en $t = 3$ comienza un nuevo horizonte de pedido, por lo tanto la heurística empieza la busqueda en $k = 2$, el cual corresponde al segundo periodo del nuevo horizonte de pedido.

Para $t = 4$, $k = 2$ y $\widehat{d_2} = d_4$,

$$h\sum_{i=2}^{k}(i-1)\widehat{d_i} = 2(5) = 10 \leq 80 = 80(1) = K\sum_{i=1}^{k-1}\frac{1}{i}.$$

Para $t = 5$, $k = 3$ y $\widehat{d_3} = d_5$,

$$h\sum_{i=2}^{k}(i-1)\widehat{d_i} = 2((1)5+(2)20) = 90 \leq 120 = 80(1+\frac{1}{2}) = K\sum_{i=1}^{k-1}\frac{1}{i}.$$

Luego, el segundo horizonte de orden contiene los periodos $\{3,4,5\}$.

Finalmente, la solución generada por H consiste en programar la producción de $z_1^* = 48$ unidades para la semana 1 y 2, y de $z_3^* = 67$ para las semanas restantes. Esto generará un costo total C^H de

$$C^H = 2(80) + 2(30) + 2((1)5 + (2)20) = \$310.$$

Note que esta es la solución óptima del problema; sin embargo esta heurística no siempre genera una solución óptima. Mas aún, el siguiente ejemplo muestra que el peor caso del rendimiento de H es arbitrariamente malo.

Ejemplo 3.2. Sea una instancia de problema donde la secuencia de demandas esta definida de la siguiente manera:

$$d_1 > 0, \quad d_N = \tfrac{k}{h(N-1)} \sum_{i=1}^{N-1} \frac{1}{t} \quad y \quad d_t = 0, \quad para\ todo \quad t = 2, \ldots, N-1.$$

Entonces, para este caso, la razón entre el costo de la solución generada por H y el costo de la solución óptima es arbitrariamente grande cuando N se hace arbitrariamente grande.

En efecto; note que para $t = 2, \ldots, N-1$, la desigualdad (3.1) se satisface trivialmente. Pero cuando $t = N$ tenemos que

$$h \sum_{i=2}^{t} (i-1) d_i = h(N-1) \frac{K}{h(N-1)} \sum_{t=1}^{N-1} \frac{1}{t} = K \sum_{t=1}^{N-1} \frac{1}{t} = K \sum_{i=1}^{t-1} \frac{1}{i}.$$

De modo que, la solución generada por H consiste en realizar un único pedido al inicio del horizonte de planeación, obteniendo así un costo total de

$$C^H = K + h \sum_{t=2}^{N} (t-1) d_t = K + h(N-1) \frac{K}{h(N-1)} \sum_{t=1}^{N-1} \frac{1}{t} = K \sum_{t=1}^{N-1} \frac{1}{t}$$

.

Sin embargo, la solución óptima consiste en realizar un pedido en el primer y último periodo, con un costo total de $C^* = 2K$.

Así,

$$\frac{C^H}{C^*} = \frac{K \left(1 + \sum_{t=1}^{N-1} \frac{1}{t} \right)}{2K} \to \infty \quad cuando \quad N \to \infty.$$

En la heurística H, se ha usado el valor $c = \sum_{i=1}^{t-1} \frac{1}{i}$, el cual es el valor mas pequeño que satisface la desigualdad (2.3). Sin embargo, este no es el mejor valor de c que se puede obtener dado un caso particular de un problema. Esto es porque para una instancia en particular $\mathbf{d} = d_1, \ldots, d_t$, la costante c depende principalmente de dicha secuencia.

En vista de lo anterior, la heurística H puede ser mejorada si se actualiza el valor de c cada vez que se verifica si la desigualdad (2.3) se cumple. En otras palabras, la nueva heurística, la cual es denotada por H^*, realiza la busqueda de una solución de la siguiente manera.

Suponga sin perdida de generalidad, que en un problema de inventario se ha llegado en algún periodo t, de tal suerte que el último pedido se realizó en el periodo 1. Entonces, se calcula el valor más pequeño de c que satisface la desigualdad (2.3) y se verifica si

$$h\sum_{i=2}^{t}(i-1)d_i \leq c_t K. \tag{3.2}$$

En caso de que la desigualdad (3.2) se cumpla, se procede con la verificación del perido $t+1$; de lo contrario, debe realizarse una orden que cubra los periodos $\{1,\ldots,t-1\}$, además se hace un pedido en t y se procede a realizar el mismo procedimiento, para luego obtener el próximo horizonte de pedido.

Observación 3.1. El mejor valor de c se obtiene de la siguiente forma: Se verifica para cual de los valores de $p \in \{1,\ldots,t-1\}$ se obtiene un mayor ahorro en el costo de almacenamiento total H_t, cuando se agrega una preparación en el periodo $p+1$. De modo que la constante c que satisface la igualdad en (2.3), es la constante c_t que sugiere la heurística.

Luego si se llega a un periodo t que no satisface (3.2), es decir $h\sum_{i=2}^{t}(i-1)d_i > c_t K$, entonces

$$c_t ph\sum_{i=p+1}^{t} d_i \geq h\sum_{i=2}^{t}(i-1)d_i > c_t K$$

Esto implica que:

$$ph\sum_{i=p+1}^{t} d_i > K.$$

Esto indica que existe un periodo $p \in \{1,\ldots,t-1\}$ tal que al agregar un pedido en $p+1$, se obtiene una reducción de los costos totales. Por lo tanto, H^* selecciona horizontes de pedidos tan largos como sea posible de manera que ningún pedido adicional pueda mejorar la solución. Se usa esta propiedad de H^* para probar el siguiente teorema.

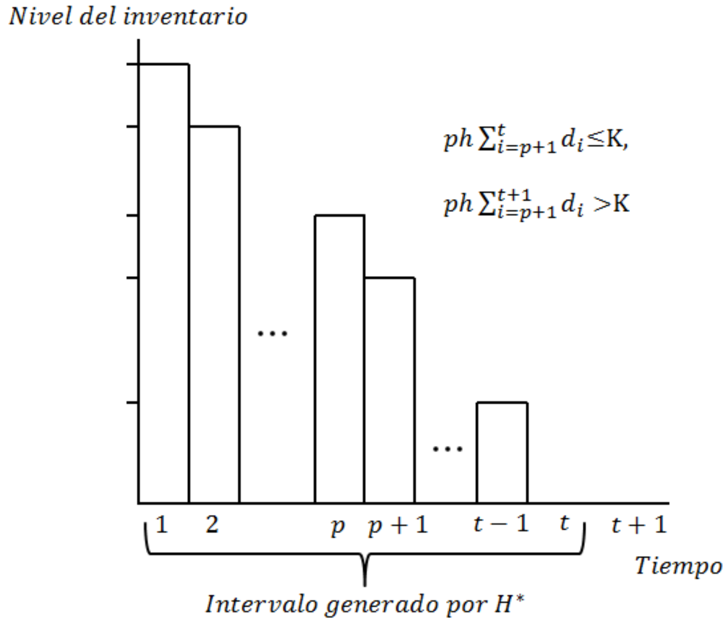

Figura 3.1: Variación del inventario de la solución heurística

Teorema 3.1. *El peor caso de rendimiento de H^* es a lo más 2.*

Demostración. Considere una solución para alguna instancia arbitraria $\mathbf{d}$ generada por la heurística H^*. Suponga que esta solución genera un costo de C^{H^*}. Modifique la solución agregando una orden (siempre que no haya) en cada periodo donde la solución óptima tenga un pedido, y reorganice las cantidades ordenadas, de acuerdo a los pedidos agregados. Denote el costo total de esta nueva solución por $\bar{C}$. Suponga además, que la solución óptima consta de n^* órdenes. Entonces, sabiendo que ambas soluciones deben tener una orden en el primer periodo, se agrega a lo más n^*-1 órdenes en la solución heurística. Luego se tiene que:

$$C^{H^*} \leq \bar{C} \leq C^* + (n^*-1)K \leq 2C^*,$$

donde C^* es el costo total de la solución óptima. La primera desigualdad se cumple debido a la observación 3.1, la cual indica que ningún pedido adicional puede mejorar el costo total del inventario. Así, debido a que la solución que genera un costo de $\bar{C}$, se le ha agregado a lo más n^*-1 órdenes, entonces estos pedidos adicionales pueden desmejorar la solución heurística. La segunda desigualdad se cumple porque la nueva solución tiene a lo más $n^* - 1$ pedidos adicionales y además menor costo de almacenamiento comparado con la solución óptima. Finalmente, la última desigualdad se cumple debido a que $C^* \geq Kn^*$. En conlusión se tiene que:

$$\frac{C^{H^*}}{C^*} \leq 2.$$

■

El siguiente ejemplo confirma que esta cota es ajustada.

Ejemplo 3.3. Considere un modelo de inventario con $N = 2n$ periodos ($n \in \mathbb{N}$), $K = h = 1$, $d_t = 2\varepsilon$ para todo t impar y $d_t = 1 - \varepsilon$ para todo t par. Se utiliza la heurística H^* para encontrar una solución del problema.

Para $t = 2$, se tiene que $p = 1$. Así,

$$ph \sum_{i=p+1}^{t} d_i = (1)(1)(d_2) = 1 - \varepsilon.$$

Luego, $c_2 = \dfrac{(1)(2-1)d_2}{1-\varepsilon} = 1$.

Ahora se verifica si se satisface la desigualdad (3.2) para $c_2 = 1$. Es decir; queremos saber si se cumple que:

$$(1)(2-1)d_2 = 1 - \varepsilon \leq 1 = c_2(K)$$

Para un ε suficientemente pequeño, la ecuación de arriba se satisface. Por lo tanto, se procede a la verificación del siguiente periodo.

Para $t = 3$, se tiene los posibles valores $p = 1$ y $p = 2$. Así, para $p = 1$

$$ph \sum_{i=p+1}^{t} d_i = (1)(1)(d_2 + d_3) = 1 - \varepsilon + 2\varepsilon = 1 + \varepsilon,$$

y para $p = 2$

$$ph \sum_{i=p+1}^{t} d_i = (2)(1)(d_3) = 2(2\varepsilon) = 4\varepsilon.$$

Luego,

$$c_3 = \frac{(1)(d_2 + 2d_3)}{4\varepsilon} = \frac{1 - \varepsilon + 4\varepsilon}{4\varepsilon} = \frac{1 + 3\varepsilon}{4\varepsilon}.$$

Ahora se verifica si satisface (3.2) para c_3,

$$(1)(d_2 + 2d_3) = 1 + 3\varepsilon \leq \frac{1 + 3\varepsilon}{4\varepsilon}.$$

La desigualdad de arriba no se satisface para ningún ε. Por lo tanto, se debe realizar un pedido en 1 que cubra las demandas d_1 y d_2. Luego, se ordena en el periodo 3 y se comienza la búsqueda de un nuevo horizonte de pedido.

Como las cantidades demandadas en $t = 2k - 1$ y $t = 2k$ son las mismas que en los periodos $t = 1$ y $t = 2$ respectivamente, para todo $k \in \{2, \ldots, n\}$. Entonces, la heurística H^* generará una solución con órdenes en los periodos $1, 3, \ldots, 2n - 1$ con un costo total de

$$C^{H^*} = nK + nh(1 - \varepsilon) = 2n - n\varepsilon.$$

Por otro lado, una solución alternativa del problema es realizar pedidos en los periodos $1, 2, 4, \ldots, 2n$ generando un costo total de

$$C^A = (n + 1)K + h(n - 1)2\varepsilon = (n + 1)K + 2\varepsilon(n - 1).$$

Si se supone que C^* es el costo total de la solución óptima, entonces

$$\frac{C^{H^*}}{C^*} \geq \frac{C^{H^*}}{C^A} = \frac{2n - n\varepsilon}{(n + 1) + 2(n - 1)\varepsilon} \rightarrow 2$$

para $\varepsilon = \frac{1}{n}$ y $n \rightarrow \infty$.

El ejemplo 3.2 y el teorema 3.1 demuestran que la razón del peor caso de la heurística H^* es 2. Los siguientes teoremas demuestran otras propiedades de H^*.

Teorema 3.2. *El número de pedidos en una solución generada por la heurística H^* es a los más el número de órdenes en cualquier solución óptima.*

Demostración. Sea el conjunto $\{r, \ldots, s - 1\}$ un horizonte de pedido de una solución óptima. Es suficiente demostrar que la heurística H^* no genera una solución con mas de una preparación en este horizonte. Asi que se supone que H^* genera una solución con dos órdenes en $\{r, \ldots s-1\}$. Es decir, si v y w son dos órdenes consecutivas en la solución heurística, entonces ocurre que $r \leq v < w < s$.

En vista de que en el periodo w la desigualdad (3.2) no se satisface (esto es por la definición de H^*), se tiene por la observación 3.1 que existe un periodo $p \in \{v+1, \ldots, w\}$ tal que una orden en dicho periodo conduce a una reducción de los costos totales, es decir;

$$\begin{aligned} K + h \sum_{i=v+1}^{p-1} (i-v)d_i + K + h \sum_{i=p+1}^{w} (i-p)d_i & \\ < K + h \sum_{i=v+1}^{w} (i-v)d_i \quad &= \quad K + h \sum_{i=v+1}^{p-1} (i-v)d_i + h \sum_{i=p}^{w} (i-v)d_i. \end{aligned}$$

Esto implica que:

$$K + h \sum_{i=p+1}^{w} (i-p)d_i < h \sum_{i=p}^{w} (i-v)d_i. \tag{3.3}$$

En la figura 3.2, se representa gráficamente la variación del nivel de inventario en el horizonte de pedido $\{v, \ldots, p-1\}$. Se observa que para algún periodo $v+k$, con $k \in \{1, \ldots, p-1-v\}$, la cantidad d_{v+k} se ha almacenado durante $v+k-v = k$ periodos, así el costo total por almacenar esta cantidad en este horizonte de orden es de $hk(d_{v+k})$. Luego, el costo total de almacenamiento, el cual se va denotar por $H_{v,p-1}$, se obtiene al sumar todas las cantidades hkd_{v+k}. Esto es;

$$H_{v,p-1} = h \sum_{k=1}^{p-1-v} kd_{v+k}.$$

Haciendo el cambio de variable $i = v+k$, se tiene que $k = i-v$, y por lo tanto,

$$H_{v,p-1} = h \sum_{i=v+1}^{p-1} (i-v)d_i.$$

Si se realiza una orden en r que satisfaga la demanda en el horizonte de orden $\{r, \ldots, p-1\}$, entonces siguiendo un procedimiento similar al que se utilizó para deducir

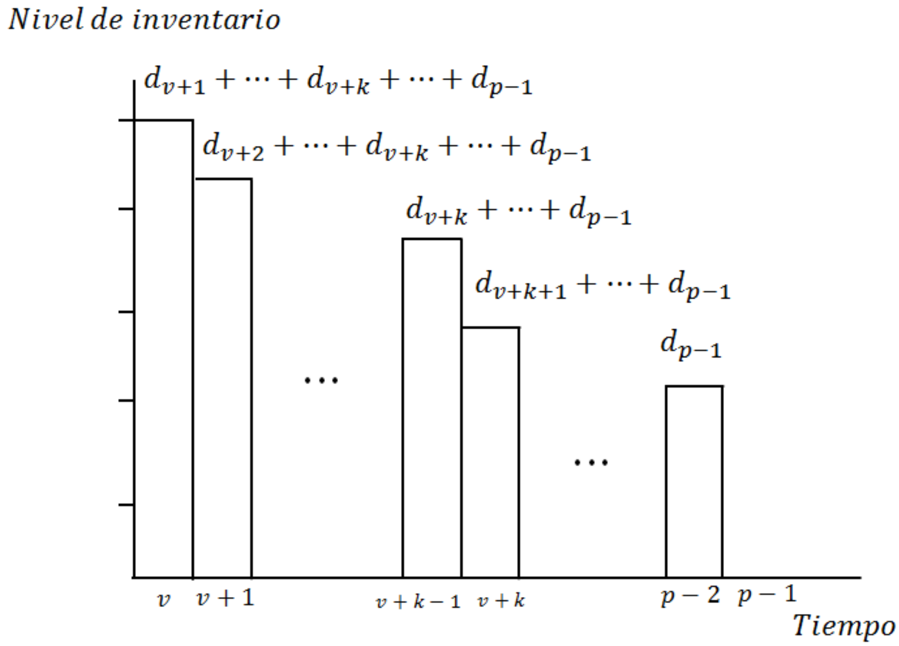

Figura 3.2: Variación del inventario en los periodos v,...,p-1

$H_{v,p-1}$ se tiene que el costo de almacenamiento total en este horizonte, el cual se denotará por $H_{r,p-1}$, viene dado por:

$$H_{r,p-1} = h \sum_{i=r+1}^{p-1} (i-r)d_i. \tag{3.4}$$

Finalmente, usando (3.3) y (3.4) se tiene que:

$$\begin{aligned} K + h \sum_{i=r+1}^{p-1} (i-r)d_i \quad &+ \quad K + h \sum_{i=p+1}^{w} (i-p)d_i \\ &< \quad K + h \sum_{i=r+1}^{p-1} (i-r)d_i + K + h \sum_{i=p}^{w} (i-v)d_i \\ &< \quad K + h \sum_{i=r+1}^{p-1} (i-r)d_i + K + h \sum_{i=p}^{w} (i-r)d_i, \ (r < v) \\ &= \quad K + h \sum_{i=r+1}^{w} (i-r)d_i. \end{aligned} \tag{3.5}$$

Pero si se agrega una orden en p, en la solución óptima, el costo total de almacenamiento en el horizonte de pedido $\{r, \ldots, s-1\}$ es de

$$\begin{aligned}
K + h \sum_{i=r+1}^{p-1} d_i \quad + & \quad K + h \sum_{i=p+1}^{s-1} (i-p)d_i \\
= & \quad K + h \sum_{i=r+1}^{p-1} (i-r)d_i + K + h \sum_{i=p+1}^{w} (i-p)d_i \\
& \quad + h \sum_{i=w+1}^{s-1} (i-p)d_i \\
< & \quad K + h \sum_{i=r+1}^{w} (i-r)d_i + h \sum_{i=w+1}^{s-1} (i-p)d_i \\
\leq & \quad K + h \sum_{i=r+1}^{s-1} (i-r)d_i,
\end{aligned}$$

donde la última desigualdad se debe a que $p > r$. Esto significa que se ha encontrado una mejor solución que la óptima, lo cual es una contradicción. Por lo tanto, el teorema queda demostrado. ∎

Teorema 3.3. *Sea el conjunto $\{r, \ldots, s-1\}$ algún horizonte de pedido generado por H^*. Entonces una solución óptima con un mínimo número de órdenes tiene a lo más dos órdenes en este intervalo.*

Demostración. Sea el conjunto $\{r, \ldots, s-1\}$ un horizonte de pedido generado por la heurística H^*. Sean u, v y w órdenes consecutivas de una solución óptima con un mínimo número de órdenes, tal que $r \leq u < v < w \leq s-1$.

El costo total de almacenamiento en el horizonte $\{u, \ldots, w\}$ viene dado por:

$$H_{u,w-1} = h \sum_{i=u+1}^{w-1} (i-u)d_i.$$

Luego, al agregar un pedido en el periodo v se obtiene un ahorro en $H_{u,w-1}$ de $(v-1-u+1)h \sum_{i=v-1+1}^{w-1} d_i = (v-u)h \sum_{i=v}^{w-1} d_i$. Pero, como la solución posee una orden en v, se tiene que

$$(v-u)h\sum_{i=v}^{w-1} d_i > K.$$

Esta desigualdad es estricta debido a que la solución óptima es una una solución con el mínimo de órdenes. Ahora como $r \leq u$,

$$(v-r)h\sum_{i=v}^{w-1} d_i \geq (v-u)h\sum_{i=v}^{w-1} d_i > K.$$

Esta última desigualdad significa que al agregar una orden en el periodo v, en la solución generada por H^*, se obtiene una reducción de los costos, y por lo tanto H^* debe tener una orden en el periodo $w-1 < s-1$ o antes de este periodo. Esto contradice la hipotesis del teorema y por lo tanto el teorema queda demostrado.

■

Los teoremas anteriores implican el siguiente corolario.

Corolario 3.1. *Sea n^* el número de pedidos en una solución óptima y sea n el número de preparaciones generado por H^*. Entonces*

$$n \leq n^* \leq 2n \quad \text{ó} \quad \frac{1}{2}n^* \leq n \leq n^*.$$

Demostración. El teorema 3.2 dice que el número de órdenes en una solución generada por la heurística es a lo mas n^*, mientras que el teorema 3.3 muestra que el número de órdenes de una solución óptima es a lo más $2n$. Por lo tanto, $n \leq n^* \leq 2n$. La otra desigualdad es consecuencia de la primera. ■

3.5. Análisis computacional del rendimiento de H^*

Para finalizar el estudio, se presentan los resultados de un pequeño estudio computacióonal que se ha realizado utilizando la herramienta de software MatLab. En este estudio se observa el buen rendimiento de la heurística H^* comparandola con las heurísticas PPB, LUC y Silver-Meal. Se ha considerado para esta prueba una cantidad de 1000 modelos de inventarios de $N = 100$ periodos cada uno, cuya secuencia de demandas, costos de preparación y costos de compra se obtienen de manera aleatoria (en el Apéndice A se ilustran los algoritmos que se utilizarón para realizar el estudio). Las conclusiones del estudio estan resumidas en la siguiente tabla.

Heurística	Soluciónes no óptimas	PP(%)	Instancias consideradas
H^*	39	0,0327	1000
SM	43	0,0205	1000
LUC	375	18,3743	1000
PPB	583	1,2477	1000

Tabla 3.1: Resumen de nuestro estudio computacional

En la tabla 3.1 se observa que la heurística H^* proporciona un mayor número de soluciónes óptimas si se consideran instancias aleatorias. Sin embargo, la heurística Silver-Meal tiene un mejor valor PP, donde PP es el porcentaje promedio para el cual el costo generado por la heurística correspondiente excede el costo de la solución óptima.

Conclusiones

La propiedad de balanceamiento, entre el costo de almacenamiento y el costo de preparación, en una solución óptima del modelo EOQ, sirvió como una base para el desarrollo de algunas heurísticas para el modelo ELS. Las mas conocidas son las heurísticas de Silver-Meal (SM), la de Balanceamiento de Periodos Parciales (PPB) y la de Costo Unitario Mínimo (LUC). A pesar de que, en una solución óptima, la propiedad de balanceamiento no siempre se cumple, se pudo demostrar que el costo de almacenamiento en un horizonte de pedido, es acotado por una cantidad proporcional al costo de preparación.

Gracias a esta cota, fue posible la construcción de una nueva heurística, denotada como H^*, cuyo peor rendimiento puede ser a lo mas de dos, tal como se demostró en el teorema 3.1. Esto quiere decir, que en caso de que H^* proporcione una solución diferente de la óptima, esta nunca va a generar un costo mayor al doble del costo mínimo total. Además, en vista de que el número de preparaciones que genera H^*, es a lo más el número de preparaciones en una solución óptima, no tendremos que preocuparnos de que la heurística pudiera generar demansiadas preparaciones en una solución.

Cabe destacar, que gracias al teorema 3.3, podemos asegurar que la cantidad de pedidos que genera H^*, es cuando menos la mitad de los pedidos de una solución óptima, con un número mínimo de órdenes. El estudio computacional que se ha realizado comprueba que esta heurística posee un buen comportamiento con respecto a las otras tres heurísticas.

Apéndice A

Algoritmos utilizados para el análisis computacional del rendimiento de la heurística H^*. Los mismos fueron ejecutados en la conocida herramienta de software MatLab.

.1. Algoritmo Wagner-Whitin en MatLab.

El siguiente código cálcula una solución óptima para el modelo de dimensionamiento del lote económico basado en el algoritmo de avance definido por Wagner y Whitin en [4].

```
function [z,x,J,arg_J,C] = wargner_whitin(d , K , c , h);

%
% Wagner-Whitin Algorithm
%
% Build production plans versus time cycles k =1,...,N responding / production queries dk
% and minimimg the sum of following production costs
%
% i) Fix production cost K_k
% ii) Unitary production cost c_k
% iii) Stockage unitary cost h_k
%
% Assumption : initial and final stocks are null.
%
%
% Inputs
```

```
% ------
%
% d Queries of production througth time cycles (1 x N)
% K Fix production cost (1 x N)
% c Unitary production cost (1 x N)
% h Stockage unitary cost (1 x N)
%
%
% Outputs
% ------
%
%
% z Optimal cost at the end of the N cycles
% x Quantities to product (1 x N)
% J Partial politics (1 x N)
% arg_J Arguments of the partial politics (1 x N)
% C Cost transition matrix (N x N)
% Author : Sbastien PARIS (sebastien.paris@lsis.org) 08/30/2001.
% ------
%

N = length(d);

%------------------- Initialization ----------------------%

C = zeros(N);

FF = C;

J = zeros(1 , N + 1);

arg_J = J ;
```

```
ZN = zeros(1 , N);

ON = ones(1 , N);

vect_N = (1:N);


%------- Cost matrix C = {c(k,l)}, k,l = 1,...,N -----%

mat_tp = triu(vect_N(ON , :));

K1 = K(:);

c1 = c(:);

h1 = h(:);

h1 = h1(1:end - 1);

KK = triu(K1(: , ON));

CC = triu(c1(: , ON));

indice = (mat_tp ~= 0);

FF(indice) = d(mat_tp(indice));

YY = cumsum(FF , 2);

SS = YY(2:end , 1:end);

```

```
HH1 = triu(h1(: , ON));

EE = HH1.*SS;

ZZ = cumsum(EE(end : -1 : 1 , :) , 1);

ZZ = ZZ(end : -1 : 1 , :);

C = KK + (CC.*YY) + ([ZZ ; ZN]);


%------------------------ Dynamic Programming --------------------%

J(end) = 0;

arg_J(end) = 0;

for k = N : -1 : 1

    tp = ZN;

    tp(:) = NaN;

    tp(k :end) = C(k , k:end ) + J(k + 1 : end);

    [J(k) arg_J(k) ] = min(tp);

end

z = J(1);

%------------------------ Back-Tracking --------------------%
```

```
k = 1;
x = ZN;
while(k <= N)
    x(k) = sum(d(k : arg_J(k)));
    k = arg_J(k) + 1;
end
J = J(1:end - 1);
```

.2. Algoritmo Silver-Meal (SM) en Matlab.

EL siguiente código cálcula una solución para el modelo de dimensionamiento del lote económico basado en la heurística propuesta por Silver y Meal.

```
% Heuristica de Silver-Meal con costos de preparacion y almacenamiento
% constantes e identicos para cada periodo.
% Entradas:
% d= vector de demandas,
% h= costo de almacenamiento,
% K= costo de preparacion.
% Salidas:
% C= costo total generado por la solucin.

% Autor: Alexis Yglesias alexisandres1990@hotmail.com

N=length(d);
```

```
Z=zeros(1,N); Z(1)=1;
k=1;
CPt=K;
for t=(k+1):N;
    %Calcula es costo de almacenamiento desde k hasta t.
    Ht=0;
    for i=(k+1):t;
        Ht=Ht+h*(i-k)*d(i);
    end
    x=(K+Ht)/(t-k+1);
    if x<=CPt
        CPt=x;
        continue
    else k=t; Z(t)=t; CPt=K;
    end
end
% Esto calcula el costo total de la solucion generada por la /
    heuristica
C=K;
w=1;
for j=(w+1):N;
    if Z(j)<=0;
        C=C+h*(j-w)*d(j);
    else C=C+K; w=j;
    end
end

disp('El costo total es')
disp(C)
```

.3. Algoritmo LUC en MatLab.

Con estas sentencias se optiene una solución para el problema ELS basado en la heurística de Costo Mínimo Unitario.

```
% Heuristica de Costo Unitario Minimo (LUC)
% Entradas:
% d= vector de demandas
% h= costo de alamcenamiento
% K=costo de preparacion
% El costo de compra se considera constante e identico para cada periodo.
% Salida:
% C= costo total
% Z= vector de pedidos

% Autor: Alexis Yglesias alexisandres1990@hotmail.com

N=length(d);
Z=zeros(1,N); Z(1)=1;
k=1;
CPt=(K/d(1));
for t=(k+1):N;
    %Calcula es costo de almacenamiento desde k hasta t.
    Ht=0;
    for i=(k+1):t;
        Ht=Ht+h*(i-k)*d(i);
    end

    x=(K+Ht)/(sum([d((k):t)]));
    if x<=CPt
        CPt=x;
        continue
```

```
    else k=t; Z(t)=t; CPt=(K/d(t));
    end
end
% Esto calcula el costo total de la solucion generada por la heuristica
C=K;
w=1;
for j=(w+1):N;
    if Z(j)<=0;
        C=C+h*(j-w)*d(j);
    else C=C+K; w=j;
    end
end

disp('El costo total es')
disp(C)
```

.4. Algoritmo PPB en MatLab

Una solución para el modelo de dimensionamiento del lote económico es obtenida con el siguiente código, basado en la heurística de Balanceamiento de Periodos Parciales.

```
% Heuristica de Balanceo de Periodos Parciales con costos de preparacion
% y alamcenamiento constantes e identicos para cada periodo.
% Entradas:
% d= vector de demandas,
% h= costo de almacenamiento,
% K= costo de preparacion.
% Salidas:
% C= costo total generado por la solucion.
% Z= vector de pedidos
```

```
% Autor: Alexis Yglesias alexisandres1990@hotmail.com

N=length(d);
Z=zeros(1,N); Z(1)=1;
k=1;

for t=(k+1):N;
    %Calcula es costo de almacenamiento desde k hasta t.
    Ht=0;
    for i=(k+1):t;
        Ht=Ht+h*(i-k)*d(i);
    end

    if Ht<=K
        continue
    else
        h1=0;
        for i=(k+1):(t-1);
        h1=h1+h*(i-k)*d(i);
        end
        h2=0;

        for i=(k+1):t;
        h2=h2+h*(i-k)*d(i);
        end

        m=min(K-h1,h2-K);
        if (K-h1)<=(h2-K)
           k=t; Z(t)=t;
        else
             if t<N; %Esto resuelve el problema que hay cuando t=N
```

```
              k=t+1; Z(t+1)=t+1;
             else
                 k=k; Z(k)=k;
             end
           end
    end
end
% Esto calcula el costo total de la solucion generada por la /
    heuristica
C=K;
w=1;
for j=(w+1):N;
    if Z(j)<=0;
        C=C+h*(j-w)*d(j);
    else C=C+K; w=j;
    end
end

disp('El costo total es')
disp(C)
```

.5. Algoritmo para el análisis del rendimiento de H^*

Se obtiene con las siguientes lineas una matriz que contiene las soluciones generadas por cada uno de los algoritmos anteriores, para una cantidad de 1000 modelos de inventarios.

```
% Autor: Alexis Yglesias alexisandres1990@hotmail.com

N=100;
costos=zeros(1000,5);% Matriz 100x5 que contiende los costos generados
                     % por WW, H, SM, LUC y PPB respectivamente.
```

```
optimas=zeros(1,4);
for l=1:1000;
d = ceil(N*rand(1 , N));
K = ceil(N*rand(1 , 1));
h = ceil(N*rand(1 , 1));
K1=linspace(K,K,N);
h1=linspace(h,h,N);
c=zeros(1,N);
O=wargner_whitin(d,K1,c,h1);
costos(l,1)=O;
heuristicaH;
costos(l,2)=C;
if C<=O
    optimas(1,1)=optimas(1,1)+1;
end
Silver_Meal;
if C<=O
    optimas(1,2)=optimas(1,2)+1;
end
costos(l,3)=C;
HeuristicaLUC;
if C<=O
    optimas(1,3)=optimas(1,3)+1;
end
costos(l,4)=C;
HeuristicaPPB;
if C<=O
    optimas(1,4)=optimas(1,4)+1;
```

```
end
costos(l,5)=C;
end
```

Referencias Bibliográficas

[1] Florian Maas. Performance of on-line lot sizing heuristics with a worst case ratio of two, July 2013.

[2] Steven Nahmias. *Production and Operations Analysis*. Mcgraw-Hill College, Mexico, D.F., 5^{th} edition, Julio 2004.

[3] Hamdy A. Taha. *Investigación de Operaciones*. Alfaomega, Mexico, D.F., 5^{a} edition, 1995.

[4] Harvey M. Wagner and Thomson M. Whitin. Dynamic version of the economic lot size model. *Management Science*, 5(1):89–96, 1958.

[5] Van Den Heuvel Wilco and Albert P.M. Wagelmans. Worst case analysis for a general class of on-line lot sizing heuristics. *Operations Research*, 58(1):59–67, 2008.

[6] Van Den Heuvel Wilco and Albert P.M. Wagelmans. A holding cost bound for the economic lot-sizing problem with time-invariant cost parameters. *Operations Research Letters*, 37(1):102–106, 2009.