UNIVERSIDAD DE LA LAGUNA

Métodos Eficientes para Algunas Variantes del Modelo EOQ
Efficient Approaches for Some Extensions of the EOQ Model

Autor: Gutiérrez Expósito, José Miguel

**Directores: Joaquín Sicilia Rodríguez
y Justo Puerto Albornoz**

Departamento de Estadística, Investigación Operativa y Computación

JOAQUÍN SICILIA RODRÍGUEZ, CATEDRÁTICO DE UNIVERSIDAD DEL ÁREA DE ESTADÍSTICA E INVESTIGACIÓN OPERATIVA DE LA UNIVERSIDAD DE LA LAGUNA

JUSTO PUERTO ALBANDOZ, CATEDRÁTICO DE UNIVERSIDAD DEL ÁREA DE ESTADÍSTICA E INVESTIGACIÓN OPERATIVA DE LA UNIVERSIDAD DE SEVILLA

CERTIFICAMOS:

Que la presente memoria, titulada "Métodos Eficientes para Algunas Variantes del Modelo EOQ - Efficient Approaches for Some Extensions of the EOQ Model", ha sido realizada bajo nuestra dirección por el licenciado D. José Miguel Gutiérrez Expósito, y constituye su Tesis para optar al grado de Doctor en Matemáticas por la Universidad de La Laguna.

Y para que conste, en cumplimiento de la legislación vigente y a los efectos que haya lugar, firmamos la presente en La Laguna, a 31 de Marzo de dos mil tres.

Fdo.: Joaquín Sicilia Rodríguez   Fdo.: Justo Puerto Albandoz

A mi familia, especialmente a mi siempre recordado abuelo Hildebrando

He podido comprobar, durante los años que ha llevado el desarrollo de esta memoria, que he contado con el apoyo y aliento de muchas personas, a las que ya considero mis amigos. A todos les estaré eternamente agradecido, y espero que estas palabras sirvan para reconocer mi deuda con ellos.

Es de justicia comenzar mis agradecimientos con los directores de esta tesis, profesores D. Joaquín Sicilia y D. Justo Puerto, que han sabido transmitirme los conocimientos en Investigación Operativa y, en especial, en Control de Inventarios, que han dado como fruto esta memoria. Además, les agradezco la buena disponibilidad que han ofrecido, la paciencia que han mostrado conmigo y sus sabios consejos durante todo este tiempo.

Quiero expresar mi más sincera gratitud a mis colegas y amigos Marcos Colebrook y Antonio Sedeño, que me han prestado apoyo incondicional durante el desarrollo de esta tesis.

Deseo también manifestar mi reconocimiento a los miembros del Departamento de Estadística, Investigación Operativa y Computación de la ULL, que de una forma u otra han contribuido en que esta tesis sea hoy una realidad. En particular, gracias a Sergio Alonso e Inmaculada Rodríguez por sus ánimos y buenos consejos, y a Beatriz Abdul por permitirme participar en su interesante investigación.

Estoy también en deuda con los miembros del Departamento de Estadística e Investigación Operativa de la Universidad de Sevilla, así como con los del Fraunhofer Institut Techno- und Wirtschaftsmathematik (ITWM) en Kaiserslautern (Alemania) por el apoyo prestado y por hacerme agradable mis estancias en ambas instituciones.

Prólogo

En la práctica, las actividades desarrolladas por la mayoría de las organizaciones tienen que ver con el uso, la transformación, la distribución o la venta de algún tipo de artículo o material. Estas tareas no sólo requieren una ubicación física donde almacenar los bienes, sino que además se debe ejercer un cierto control y coordinación sobre el mantenimiento y reposición de las existencias. De manera general, los inventarios se definen como aquellos bienes almacenados con valor económico para los que se prevé una demanda futura. Es por ello, que el control del inventario es una acción común en casi todas las industrias, empresas y organizaciones.

En los paises desarrollados, el capital invertido en el control/gestión de los inventarios representa un porcentaje nada despreciable del producto interior bruto anual. Esta inversión da fe de la importancia que actualmente tiene la gestión de inventarios en cualquier parcela de la economía.

El Control del Inventario es una área relevante de la Investigación Operativa cuyo objetivo es, básicamente, la gestión eficiente de los artículos mantenidos por las empresas para satisfacer la demanda de los clientes, ofreciendo además, información para la toma de decisiones encaminadas a alcanzar objetivos económicos y tácticos. Generalmente, estos objetivos suelen estar en conflicto dentro de la empresa, debido a que las responsabilidades están claramente demarcadas y a los gerentes se les ha animado a suboptimizar independientemente cada uno de sus departamentos. Precisamente, el control de inventarios permite reconciliar estos objetivos para alcanzar beneficios globales a la empresa.

Desde un punto de vista científico, el interés por los problemas de gestión óptima de los inventarios se remonta a los comienzos del siglo veinte, tras la segunda guerra mundial. La postura frente a la gestión del inventario ha cambiado notablemente a lo largo del siglo pasado; desde su inicio cuando se pensaba que lo más conveniente era mantener grandes cantidades de inventario para cubrir fluctuaciones de la demanda, hasta nuestros días en los que se persigue reducir los inventarios a niveles mínimos. Como fruto de este importante desarrollo, podemos encontrar, en revistas y libros especializados, una razonable cantidad de artículos.

Debemos destacar, entre todas las aportaciones a la literatura de gestión de inventarios, los trabajos de Harris (1913) y Wilson (1934), quienes, de manera independiente, desarrollaron el modelo germinal de la teoría de inventarios al que suele referirse como modelo EOQ (Economic Order Quantity). Es ampliamente conocido que la aplicación de este modelo a problemas reales ha dado excelentes resultados y, por ello, no nos debe sorprender que distintas extensiones de este modelo sigan actualmente siendo un tema de investigación. De hecho, esta nutrida cantidad de artículos sobre generalizaciones del modelo EOQ es la que pone de manifiesto la evolución de los sistemas de inventario.

En sintonía con esta evolución, esta tesis contempla nuevas extensiones del modelo EOQ y algoritmos eficientes que las resuelven. De manera más precisa, se abordan las versiones dinámicas de dicho modelo al caso con limitación de inventario y al caso con múltiples escenarios. Además, se analiza la extensión del EOQ al caso de dos niveles ofreciendo un algoritmo para determinar políticas eficientes. Por lo tanto, los métodos propuestos a lo largo de esta memoria representan una recopilación de técnicas eficientes, que pueden servir de ayuda al decisor para diseñar la política más conveniente en términos de minimización de costes.

El resto de esta memoria se ha estructurado como sigue. En el Capítulo 2, hemos abordado la versión dinámica del modelo EOQ admitiendo restricciones de capacidad de inventario, demostrando que dependiendo de la estructura de costes se pueden diseñar distintos algoritmos eficientes. En concreto, los resultados relativos a la caracterización de planes óptimos así como el correspondiente algoritmo para el caso de costes cóncavos se recogen en Gutiérrez et al. [12]. En cambio, en ausencia de costes de setup (activación) y admitiendo que la estructura de costes es lineal , demostraremos, como también se hace en Sedeño-Noda et al. [21], que se puede desarrollar un algoritmo greedy de orden $O(T \log T)$ para obtener políticas óptimas. Además, también propondremos, al igual que en Gutiérrez et al. [13], un algoritmo de orden $O(T \log T)$ basado en una técnica geométrica para el caso en el que las funciones de coste son lineales y se admiten setups. En los Capítulos 3 y 4 se discuten extensiones del modelo EOQ desde la perspectiva de la programación multicriterio. De manera mas específica, en el Capítulo 3, se asume que el valor que toma la demanda en un periodo dado no es conocido sino que se extrae de un conjunto finito de valores discretos, generando así distintos vectores posibles de demanda y dando lugar, por lo tanto, a diferentes escenarios. Resolveremos este problema aplicando un esquema de ramificación y acotación y presentamos un método general para identificar el conjunto de soluciones eficientes. Algunos de los resultados propuestos en este capítulo ya han sido publicados en Gutiérrez et al. [11]. Por último, en el Capítulo 4 se analiza el sistema de Inventario/Distribución (I/D) considerando dos criterios y se desarrolla un método eficiente para caracterizar las soluciones no dominadas. Parte de las contribuciones incluidas en este capítulo se recogen en un trabajo de Gutiérrez et al. [10].

Índice General

Resumen en español

Fundamentos de la Gestión de Inventarios

Introducción

Desde un punto de vista operativo, los inventarios respresentan aquellos bienes almacenados por una organización para los que se prevé una demanda futura. En cambio, desde un punto de vista económico, los inventarios suponen un capital invertido que es recuperado cuando se satisface la demanda para un artículo o servicio específico. En esencia, los inventarios pueden verse como la cantidad de artículos adquirida por la empresa, también conocida por cantidad colchón (buffer), para atenuar la diferencia entre la oferta y la demanda variable. Un objetivo frecuente en control de inventarios es el de gestionar este colchón a mínimo coste. Por lo tanto, el *control de inventario* involucra a todas aquellas actividades y procedimientos empleados para asegurar el mantenimiento de la cantidad correcta de cada artículo. En este sentido, los inventarios juegan un importante papel por la dificultad de sincronizar perfectamente la oferta y la demanda. Esta falta de sincronización es, básicamente, el resultado de cuatro factores: *el tiempo*, *la discontinuidad*, *la incertidumbre* y *la economía* (Tersine [23]).

El interés en el estudio de sistemas de inventario ha crecido notablemente en las últimas decadas, y son numerosas las publicaciones que están dedicadas en exclusiva a este tema. En Hax y Candea [15], Silver et al. [22], Chikán [5], Waters [27], Narasimhan et al. [19], Tersine [23], Plossl [20], Zipkin [32] y Axsäter [3], entre otros, se presentan excelentes revisiones de sistemas de inventario. Además, muchos de los artículos dedicados a este tópico aparecen regularmente en revistas especializadas de impacto como Management Science, Operations Research, Journal of the Operational Research Society, Computers and Operations Research, European Journal of Operational Research, y en muchas otras.

Cabe mencionar que la gestión de inventarios se desarrolló a partir del trabajo germinal de Harris [14] en 1913, quien propuse el modelo EOQ (Economic Order Quantity), aunque este sistema fue introducido también, de manera independiente, por Wilson [29] en 1934. Este modelo básico impone ciertas hipótesis sobre los parámetros como el hecho de que la demanda final para el artículo sea conocida.

Notación y definiciones básicas

Las principales características o componentes que intervienen en los sistemas de inventario son: la *demanda*, la *reposición*, los *costes* y las *restricciones*.

La demanda

De manera general, la demanda no es una variable que pueda ser controlada directamente, ya que depende de las decisiones de gente externa a la organización con el problema de inventario. El *tamaño de la demanda* representa la cantidad necesaria para satisfacer la demanda a través del inventario. Cuando este tamaño no varía con el tiempo diremos que es constante, en otro caso será variable. Los sistemas de inventario en los que el tamaño de la demanda se concoce con antelación reciben el nombre de *sistemas determinísticos*. En tales sistemas, cuando la demanda es constante, es conveniente usar la *tasa de demanda* que se define como el tamaño de la demanda por unidad de tiempo. En ocasiones, es posible identificar distintas maneras de ocurrencia de la demanda. En concreto, si consideramos un periodo de tiempo, la demanda se puede satisfacer al comienzo o al final de este periodo; también puede ser cubierta uniformemente a lo largo del periodo o siguiendo un patrón potencial; etc (ver Naddor [18]). A las distintas maneras en las que la demanda puede acontecer se les da el nombre de patrones de demanda. En adelante, prestaremos atención al caso en el que la demanda ocurre al comienzo del periodo y, también, al caso en el que la demanda adopta un patrón uniforme.

La reposición

La reposición de los sistemas de inventario es controlada, generalmente, por el decisor. La reposición hace referencia a las cantidades que se programan para ser incluidas en el inventario, al instante en el que se toman las decisiones relativas a la reposición de esas cantidades, y al tiempo en el que éstas son realmente añadidas en el inventario. Por lo tanto, se pueden identificar los siguientes elementos en la reposición. El *periodo de planificación* es el intervalo temporal entre reposiciones consecutivas, la *cantidad a reponer* representa la cantidad programada que se debe incorporar al inventario, y por último, el *tiempo de retardo* es el intervalo temporal entre la programación de un pedido y su incorporación al inventario.

Los costes

Representan los componentes económicos más importantes en cualquier modelo de inventario, y se pueden agrupar en distintas categorías. El *coste unitario de compra* representa el coste en el que se incurre cuando se compra una unidad de artículo en caso de abastecimiento externo, o el coste de producir una unidad cuando el artículo es manufacturado por la propia empresa. El coste fijo o coste de activación (setup) indica el gasto fijo de tramitar un pedido a un proveedor externo o de iniciar el proceso de producción. El *coste de reposición/pedido* incluye los gastos variables por tramitar un pedido. El *coste de mantenimiento* incorpora los costes de capital/oportunidad, impuestos, seguros, manipulación, almacenaje, deterioro y obsolesencia y, normalmente, suele ser proporcional al capital invertido en inventarios. Por último, el *coste de rotura/escasez* refleja la consecuencia económica de una mala política de reposición o producción. La rotura externa ocurre cuando la demanda del cliente no es satisfecha, mientras que la escasez interna se produce cuando la demanda, dentro de la organización, no es cubierta. La cuantificación de este coste ha sido, durante mucho tiempo, un problema difícil y no resuelto de manera satisfactoria y, por ello, muchas organizaciones evitan el problema de la estimación de este coste estableciendo niveles de servicio al cliente.

Las restricciones

Están relacionadas con las limitaciones que se imponen sobre los elementos discutidos en las secciones previas, pudiéndose clasificar en: tipo de unidades (discretas o continuas), restricciones sobre la demanda, sobre la reposición y sobre los costes. En el desarrollo de esta memoria, se consideran, en el Capítulo 2, restricciones sobre la cantidad a reponer por capacidades de almacenaje o por la capacidad del vehículo de reparto en el Capítulo 4. Además, en los Capítulos 2 y 3, se obliga a que la cantidad a reponer sea un valor entero.

Es evidente que cualquier *problema de inventario* tiene que ver con la toma de decisiones óptimas que minimicen el coste total de un sistema de inventario. Normalmente, estas decisiones se toman en términos de *tiempo* y *cantidad,* ya que éstas son variables sujetas a control. Según lo anterior, el decisor debe responder a las siguientes cuestiones: *Cuándo se debe tramitar un pedido*? y *Qué cantidad se debe pedir*?. A la primera pregunta se responde con una de las siguientes alternativas:

1. El inventario se debe reponer cuando la cantidad en él sea igual o inferior a s_o[1].

2. El inventario se debe reponer cada t_o unidades de tiempo.

La segunda pregunta suele tener una de las siguientes respuestas:

1. La cantidad a pedir deber ser igual a Q_o unidades.

[1]El subíndice o hace referencia a valor óptimo.

2. La cantidad a pedir debe ser tal que eleve la cantidad de inventario hasta un valor de S_o unidades.

Las cantidades s, t, Q y S reciben el nombre de *punto de reposición*, *periodo de planificación/gestión*, *cantidad de pedido*, y *nivel de inventario*, respectivamente. En Naddor [18], Tersine [23], Plossl [20], Narasimhan et al. [19], Chikán [5], Silver et al. [22] y Axsäter [3], entre otros, se puede encontrar una completa recopilación de estos sistemas de inventario junto con sus métodos solución.

Cuando los parámetros varían con el tiempo, solemos referirnos a los sistemas de inventario como *sistemas dinámicos*. Así, la versión dinámica del modelo EOQ considera un horizonte temporal finito dividido en T periodos, y su objetivo es determinar un plan $\mathbf{Q} = (Q_1, Q_2, \ldots, Q_T)$ con coste mínimo. Dado un periodo i, con $1 \leq i \leq T$, la demanda, el coste de reposición y el de mantenimiento para ese periodo se denotan por d_i, $C_i(Q_i)$ y $H_i(I_i)$, respectivamente. Observe que $C_i(Q_i)$ y $H_i(I_i)$ son funciones de la cantidad a pedir Q_i y de la cantidad final de inventario para ese periodo I_i. De manera más detallada, Q_i representa la cantidad a pedir al comienzo del periodo i, mientras que I_i indica la cantidad de inventario al final del mismo periodo. En ausencia de roturas, el coste total se expresa como la suma de los costes de reposición y mantenimiento para cada periodo, y de lo que se trata es de conseguir un plan de pedidos $\mathbf{Q} = (Q_1, Q_2, \ldots, Q_T)$ que minimice el coste total satisfaciendo la demanda de todos los periodos. Cuando, además, se consideran capacidades de inventario, al problema se le da el nombre de *cantidad de pedido dinámica con limitaciones de almacenaje* o, por simplicidad, EOQ dinámico con capacidad de inventario. Si la capacidad de inventario es fija, ésta se denota por W permaneciendo constante durante todo el horizonte temporal. Por otro lado, si la capacidad varía, la denotaremos por W_i, $i = 1, \ldots, T$.

Variantes del Modelo EOQ

Las extensiones del modelo EOQ que se recogen en esta memoria son la cantidad de pedido dinámica con limitaciones de almacenaje, o EOQ dinámico con capacidades de almacén, el EOQ dinámico con incertidumbre en los datos de entrada y la cantidad de pedido en dos instalaciones, o sistema Inventario/Distribución. Pasamos a comentar, en detalle, cada una de ellas.

EOQ dinámico con capacidad de inventario

Como ya comentamos anteriormente, en este tipo de modelos el horizonte temporal se divide en T periodos, y la demanda se debe satisfacer al comienzo de cada uno de ellos no permitiendo roturas. Este modelo fue introducido por Wagner y Whitin [26], e independientemente por Manne [17]. En su versión original, los costes de mantenimiento eran lineales, los de reposición constantes y se incurría en un coste fijo (setup) cada vez que se hacía un pedido. Además, se asumían niveles de inventario cero al comienzo y al final del horizonte temporal. El objetivo consiste en determinar un vector de pedidos óptimo o plan óptimo satisfaciendo las demandas. Wagner y Whitin establecieron que, entre los planes óptimos, siempre existe uno en el que sólo se pide en un periodo cuando el inventario final del periodo predecesor es cero. Esta condición de optimalidad es conocida como propiedad ZIO (*Zero Inventory Ordering*) y, a partir de ella, se puede desarrollar un algoritmo de orden $\mathrm{O}(T^2)$ basado en programación dinámica para determinar planes óptimos. Esta propiedad es aún válida incluso cuando las funciones de coste son cóncavas en general (ver Veinott [24]). En los últimos años, Federgruen y Tzur [9], Aggarwal y Park [1] y Wagelmans et al. [25] han desarrollado, aplicando distintas técnicas, algoritmos de orden $\mathrm{O}(T\log T)$ para estructura de costes lineales con setup. Zangwill [31] demostró que este problema se podía ver como un problema de flujo en red, donde las soluciones óptimas ZIO se correspondían con flujos acíclicos de la red. Las extensiones al caso con rotura se analizan en Zangwill [30], quien demostró que, entre las soluciones óptimas, siempre había una verificando que, entre dos periodos consecutivos con producción no nula, debería haber al menos un periodo con inventario final igual a cero. Este autor explotó esta propiedad para diseñar un algoritmo de orden $\mathrm{O}(T^3)$.

En el Capítulo 2 de esta memoria se analiza, en detalle, el problema EOQ dinámico con capacidad de inventario. Este modelo fue estudiado originalmente por Love [16], quien desarrolló un algoritmo de orden $\mathrm{O}(T^3)$ basado en programación dinámica para costes cóncavos en general.

Caso I) Considerando que $C_t(\cdot)$ y $H_t(\cdot)$ representan, respectivamente, funciones cóncavas de la cantidad de pedido Q_t y de la cantidad de inventario I_t en el periodo t, $t = 1, \ldots, T$, el problema de Cantidad de Pedido Dinámica con Capacidad de Inventario (o, por simplicidad, P) se puede formular como sigue:

$$\begin{array}{lll}
(P) & \min \sum_{t=1}^{T} \left(C_t(Q_t) + H_t(I_t) \right) & \\
s.a. & & \\
 & I_0 = I_T = 0 & \\
 & I_{t-1} + Q_t - I_t = d_t & t = 1, \ldots, T \\
 & I_{t-1} + Q_t \leq W_t & t = 1, \ldots, T \\
 & Q_t, I_t \in \mathbb{N}_0 & t = 1, \ldots, T
\end{array}$$

donde $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$.

Para esta estructura general de costes demostramos que se puede diseñar un algoritmo de orden $O(T^3)$ que, en la práctica, es casi un 30% mas rápido que el de Love, y que, además, se comporta de manera lineal ($E(T)$) cuando la demanda del periodo t se elige en el intervalo $[0, W_t]$, $t = 1, \ldots, T$. Esta mejora computacional es posible gracias al desarrollo de una nueva caracterización de los planes óptimos que queda reflejada en los siguientes resultados:

Propiedad 1 Entre los planes óptimos para P, existe al menos uno $\mathbf{Q} = (Q_1, \ldots, Q_T)$ tal que para cada periodo i, Q_i satisface:

$$Q_i = \begin{cases} 0 \\ \min\limits_{i < t \leq T+1} \{d_{it} - I_{i-1}, W_i - I_{i-1}\} \end{cases}, i = 1, \ldots, T$$

En otras palabras, para $1 \leq i < t \leq T + 1$, se verifica la siguiente expresión

$$(Q_i + I_{i-1} - W_i)(Q_i + I_{i-1} - d_{it})Q_i = 0$$

Propiedad 2 Siempre se puede encontrar un plan óptimo $\mathbf{Q} = (Q_1, \ldots, Q_T)$ para P tal que si I_{j-1} ($j = 2, \ldots, T$) corresponde a la suma de demandas de los periodos j a k con $j \leq k \leq T$, entonces Q_j es cero.

Caso II) Cuando $C_t(\cdot) = c_t$ y $H_t(\cdot) = h_t$ son funciones lineales y en ausencia de costes de setup, el problem P puede reformularse como un problema P' de flujo de coste mínimo (MCF) de la siguiente manera:

$$
\begin{array}{lll}
(P') & \min \sum\limits_{t=1}^{T} (c_t Q_t + h_t I_t) & \\
s.t. & & \\
 & I_0 = I_T = 0 & \\
 & \sum\limits_{t=1}^{T} Q_t = d_{1,T+1} & \\
 & I_{t-1} + Q_t - I_t = d_t & t = 1, \ldots, T \\
 & 0 \leq I_t \leq w_t & t = 1, \ldots, T-1 \\
 & Q_t \in \mathbb{N}_0 & t = 1, \ldots, T
\end{array}
$$

Basándonos en los resultados de Zangwill [31], podemos establecer la siguiente red para el problema P'. Sea $G(V,A)$ una red dirigida, donde V es el conjunto de $n = T+1$ nodos y A es el conjunto de $m = 2T-1$ arcos. Cada nodo t ($t = 1, \ldots, T$) tiene una demanda igual a $-d_t$, mientras que el nodo 0 (nodo fuente) tiene que satisfacer la demanda en cada nodo con una cantidad de $d_{1,T+1} = \sum_{t=1}^{T} d_t$ unidades. En lo que sigue, $d_{ik} = \sum_{t=i}^{k-1} d_t$.

Podemos distinguir dos tipos de arcos: *arcos de producción/pedido* asociados a las variables de decisión Q_t, y *arcos de inventario* relacionados con las variables de estado I_t, con $t = 1, \ldots, T$. Cada arco $(0, t)$ en la red tiene un coste unitario igual a c_t y una capacidad infinita. Por otro lado, cada arco de inventario $(t, t+1)$ tiene un coste unitario igual a h_t y una capacidad máxima de inventario de $w_t = W_t - d_t$ unidades.

De esta manera, el problema P' se puede resolver empleando cualquier algoritmo de MCF (ver, por ejemplo, Ahuja et al. [2]). Sin embargo, la red generada para este problema es un caso particular de red serie-paralela (ver Duffin [8] para una definición completa de topologías de redes serie-paralelas), y el mejor algoritmo para resolver el problema MCF en este tipo de redes se debe a Booth and Tarjan [4]. La adaptación de las complejidades de este algoritmo al problema P' hace que el procedimiento se ejecute en un tiempo de orden $\mathrm{O}(T \log T)$, requiriendo un espacio de orden $\mathrm{O}(T \log^* T)$.

Sin embargo, proponemos un algoritmo *ad hoc* que explota las características de la red definida anteriormente, requiriendo tiempos de ejecución de orden $\mathrm{O}(T \log T)$ y con una complejidad espacial de orden $\mathrm{O}(T)$, mejorando así, las complejidades correspondientes al procedimiento de Booth and Tarjan [4].

Caso III) Por último, si los costes de mantenimiento h_t y reposición c_t son lineales, y se admiten costes fijos de setup f_t, el problema P' se puede reescribir para dar el problema P''.

$$
\begin{array}{lll}
(P'') & \min \sum_{t=1}^{T} (f_t y_t + c_t Q_t + h_t I_t) & \\
& s.a. & \\
& I_0 = I_T = 0 & \\
& Q_t + I_{t-1} - I_t = d_t & t = 1, \ldots, T \\
& d_{t,T+1} y_t - Q_t \geq 0 & t = 1, \ldots, T \\
& 0 \leq I_t \leq W_t - d_t & t = 1, \ldots, T \\
& Q_t, I_t \in \mathbb{N}_0,\ y_t \in \{0, 1\} & t = 1, \ldots, T
\end{array}
$$

Observe que, como consecuencia de las restricciones de almacenaje, la cantidad máxima a ser producida/pedida en un periodo está acotada. De acuerdo con esto, sea M_t la cantidad máxima a ser producida/pedida en el perido t ($t = 1, \ldots, T-1$), que se puede obtener fácilmente de la expresión: $M_t = \min(M_{t+1} + d_t, W_t)$, donde $M_T = d_T$. Además, denotamos por p_t al periodo máximo accesible cuya demanda puede ser completamente satisfecha con inventario mantenido desde el periodo t ($t = 1, \ldots, T-1$), es decir, $p_t = \max(j : t \leq j \leq T \text{ y } (M_t - d_{t,j+1}) \geq 0)$, con $p_T = T$. Los valores M_t y p_t ($t = 1, \ldots, T-1$) se determinan en $\mathrm{O}(T)$ a partir de los valores de la demanda y la capacidad de almacén.

Para establecer el método solución para este caso es necesario introducir previamente algunas definiciones. Sea $G(t)$ el coste óptimo del subproblema que abarca los periodos t hasta T ($t = 1, \ldots, T$), con $G(T+1) = 0$, y sea $AC_t = c_t + \sum_{i=t}^{T} h_i$ el coste acumulado desde el periodo t al periodo T. Asimismo, sea $\widehat{Q}_t$ la decisión óptima en el periodo t cuando se resuelve el subproblema con periodos t hasta T, y sea $\delta(z)$ una función delta tal que $\delta(0) = 1$ y $\delta(z) = 0$ si $z \neq 0$. Además, denoteremos por $Q^*_{s,t}$ la decisión óptima en el periodo t cuando el subproblema con periodos s hasta T se resuelve, siendo $s < t$.

Los siguientes resultados permiten caracterizar planes óptimos para el problema P''.

Propiedad 3 Si la decisión óptima en el periodo t, $\widehat{Q}_t$, es producir/pedir M_t, entonces la decisión óptima para aquellos periodos $j \in [t+1, p_t+1]$ con $\widehat{Q}_j < M_t - d_{t,j}$ es no pedir, es decir, $Q^*_{t,j} = 0$.

Propiedad 4 Si $\widehat{Q}_t = M_t$, entonces existe un periodo $j \in [t+1, p_t+1]$ con $\widehat{Q}_j \geq M_t - d_{t,j}$ tal que su decisión óptima es $Q^*_{t,j} = \widehat{Q}_j - (M_t - d_{t,j})$.

Propiedad 5 Para el problema P'', se obtiene un plan de reposición/producción óptimo aplicando la siguiente fórmula de recurrencia

$$G(t)=\min\ [\{\min_{t<j\le p_t+1}(f_t+AC_t d_{t,j}+G(j)),\ \text{si } d_t>0,$$
$$\text{o }\min[G(t+1),\min_{t+1<j\le p_t+1}(f_t+AC_t d_{t,j}+G(j))],\ \text{en otro caso}\},$$
$$\min_{\substack{t<j\le p_t+1\\ \widehat{Q}_j\ge M_t-d_{t,j}}}(f_t+AC_t M_t+G(j)-F(t,j))]$$

$$\text{donde} F(t,j)=AC_j(M_t-d_{t,j})+\delta(M_t-d_{t,j}-\widehat{Q}_j)f_j.$$

Es evidente, que una implementación directa de esta fórmula de recursión da lugar a un algoritmo de orden $O(T^2)$. Sin embargo, proponemos un algoritmo $O(T\log T)$ basado en la técnica geométrica de Wagelmans et al. [25]. Estos autores desarrollaron un método geométrico que consistía en el cálculo de la envoltura convexa inferior de los puntos $(d_{t,T}+1, G(t))$, $t=1,\ldots,T+1$. Sea $q(j)$ el perido eficiente más pequeño con pendiente menor que AC_j. Lamentablemente, esta técnica no se puede aplicar directamente en el caso de niveles de inventario acotados. No obstante, podemos adaptar dicha técnica a nuestro modelo de la siguiente manera. Debemos definir dos listas L_E y L_{NE} que contendrán, respectivamente, los periodos eficientes y no eficientes. Cuando se esté evaluando el periodo j, si $q(j)$ es menor que p_j+1, entonces el nuevo procedimiento procederá de la misma manera que el de Wagelmans et al. [25], es decir, produciendo/pidiendo $d_{j,q(j)}$ unidades. En caso que $q(j)$ sea igual a p_j+1, podemos tomar dos decisiones: pedir M_j o pedir d_{j,p_j+1}. Sin embargo, se puede demostrar fácilmente que cuando $AC_j<AC_{q(j)}$ la decisión óptima consiste en pedir/producir M_j, y $d_{j,q(j)}$ en otro caso. Por último, si $q(j)>p_j+1$, el periodo eficiente $q(j)$ no es factible para el subproblema que comienza en el periodo j, y por lo tanto debemos comparar el periodo eficiente con pendiente más pequeña $q_E(j)\le p_j+1$ en L_E con el periodo no eficiente $q_{NE}(j)\le p_j+1$ en L_{NE}. Según esto último, denotemos por $G_E(j)=f_j+AC_j d_{j,q_E(j)}+G(q_E(j))$ y $G_{NE}(j)=f_j+AC_j d_{j,q_{NE}(j)}+G(q_{NE}(j))$ los costes asociados, respectivamente, a los periodos $q_E(j)$ y $q_{NE}(j)$ sucesores de j. Si evaluando ambos costes tenemos que $G_E(j)\le G_{NE}(j)$, entonces el periodo $q_E(j)$ continúa siendo eficiente. En otro caso, es decir, cuando $G_E(j)>G_{NE}(j)$, el periodo $q_{NE}(j)$ debe ser insertado en L_E transfiriendo el resto de elementos en esa lista a L_{NE}. Realmente, esta transferencia entre listas se corresponde con una actualización de la envoltura inferior.

Además, demostraremos que cuando la estructura de costes del problema coincide con la propuesta en Wagner y Whitin [26], siempre se puede obtener un plan óptimo satisfaciendo la propiedad ZIO.

Propiedad 6 Cuando en el problema P'' los costes de producción/reposición son constantes, siempre habrá una política óptima $\mathbf{Q} = (Q_1, \ldots, Q_T)$ tal que $I_{t-1}Q_t = 0$, $t = 1, \ldots, T$.

A diferencia del modelo EOQ dinámico original, en el que los valores de los parámetros de entrada son conocidos con antelación, también consideramos, en el Capítulo 3, la situación en la que estos parámetros pueden tomar valores de un conjunto finito discreto. Como consecuencia de esta característica en los datos de entrada, se producen diferentes escenarios, y distintas políticas eficientes surgen frente a las que el decisor tendrá que elegir aquella (robusta) que se ajuste, más adecuadamente, a sus criterios. Proponemos un método eficiente para determinar el conjunto completo de soluciones no dominadas basado en el esquema de ramificación y acotación (BB). De igual forma, demostraremos que extensiones de la propiedad ZIO se pueden adaptar con éxito al caso con múltiples escenarios.

EOQ dinámico multiescenario

En este caso, se consideran simultáneamente M escenarios o réplicas del sistema, y se asume que sólamente una (robusta) política, perteneciente al conjunto de soluciones no dominadas, va a ser implementada. Estas réplicas modelan la incertidumbre en la estimación de los parámetros, dado que no se conoce a priori ni los verdaderos valores de los datos de entrada ni una distribución de probabilidad para ajustarlos. Es por ello, que buscamos soluciones de compromiso que se comporten aceptablemente bien en cualquiera de los escenarios admisibles.

En adelante, emplearemos la siguiente notación:

$h_i^j(\cdot)$	: coste de mantenimiento para el periodo j en el escenario i.
$c_i^j(\cdot)$	: coste de producir/pedir en el periodo j en el escenario i.
I_i^j	: inventario al final del periodo j en el escenario i.
d_i^j	: demanda para el periodo j en el escenario i.
D	: demanda total ($\sum_{j=1}^T d_i^j = \sum_{j=1}^T d_s^j$, $\forall\ i,\ s \in \{1, \ldots, M\}$).
Q_j	: cantidad de pedido/producción en el periodo j.

Asumiremos, sin pérdida de la generalidad, que $I_i^0 = I_i^T = 0$ para $i = 1, \ldots, M$.

Las siguientes definiciones son necesarias para el planteamiento del problema. Dado un vector de producción/pedido $\mathbf{Q} = (Q_1, \ldots, Q_T) \in \mathbb{N}_0^T$, el vector de niveles de inventario para el escenario i se denota por $I_i(\mathbf{Q}) = (I_i^1, \ldots, I_i^T)$, donde

$$I_i^j = I_i^{j-1} + Q_j - d_i^j,\ j = 1, \ldots, T.$$

De la misma manera, el coste acumulado desde el periodo j hasta el k en el escenario i viene dado por

$$R_i^{j,k}(\mathbf{Q}) = \sum_{t=j}^{k} r_i^t(Q_t, I_i^t)$$

donde $r_i^t(Q_t, I_i^t) = c_i^t(Q_t) + h_i^t(I_i^t)$.

Por lo tanto, el vector de costes totales $R(\mathbf{Q})$ para todos los escenarios, dado un vector de producción/pedido $\mathbf{Q} \in \mathbb{N}_0^T$, es como sigue

$$R(\mathbf{Q}) = \left(R_1^{1,T}(\mathbf{Q}), \ldots, R_M^{1,T}(\mathbf{Q})\right)$$

De ahí que el conjunto de planes de producción/pedido eficientes $\mathcal{P}$ se puede definir como

$\mathcal{P} = \{\mathbf{Q} \in \mathbb{N}_0^T$: no hay otro $\mathbf{Q}' \in \mathbb{N}_0^T : R(\mathbf{Q}') \leq R(\mathbf{Q})$, con al menos una de las inecuaciones siendo estricta$\}$

donde $R(\mathbf{Q}') \leq R(\mathbf{Q})$ indica que $R_i^{1,T}(\mathbf{Q}') \leq R_i^{1,T}(\mathbf{Q})$ para $i = 1, \ldots, M$.

Usando las definiciones previas, podemos establecer el problema EOQ dinámico con múltiples escenarios de la siguiente manera:

$$\begin{array}{ll}
v - \min(R_1^{1,T}(\mathbf{Q}), \ldots, R_M^{1,T}(\mathbf{Q})) & \\
s.a.: & \\
I_i^0 = I_i^T = 0 & i = 1, \ldots, M \\
I_i^{j-1} + Q_j - I_i^j = d_i^j & j = 1, \ldots, T,\ i = 1, \ldots, M \\
Q_j \in \mathbb{N}_0 & j = 1, \ldots, T \\
I_i^j \in \mathbb{N}_0 & j = 1, \ldots, T,\ i = 1, \ldots, M
\end{array}$$

donde $v - \min$ significa mimización vectorial, y el objetivo consiste en determinar el conjunto de soluciones no dominadas con respecto a las M funciones objetivo.

Dado que la versión original con un único escenario se resuelve satisfactoriamente usando programación dinámica, parece razonable aplicar al problema anterior la extensión de esta técnica para el caso multiobjetivo. Para ello, definimos la siguiente fórmula de recurrencia:

$$F(j, (I_1^{j-1}, \ldots, I_M^{j-1})) = v - \min_{Q_j \in \mathbb{N}_0} \left\{ \begin{bmatrix} c_1^j(Q_j) \\ \vdots \\ c_M^j(Q_j) \end{bmatrix} + \begin{bmatrix} h_1^j(I_1^{j-1} + Q_j - d_1^j) \\ \vdots \\ h_M^j(I_M^{j-1} + Q_j - d_M^j) \end{bmatrix} \right.$$

$$\oplus F(j+1,(I_1^j,\ldots,I_M^j))\}$$

con $A \oplus B = \{a+b : a \in A,\ b \in B\}$ para cualesquiera dos conjuntos A y B.

En este caso, el conjunto de soluciones no dominadas para el problema correspondería al conjunto de vectores en $F(1,0,\ldots,0)$. No obstante, debido al inconveniente de la dimensionalidad asociada, normalmente, a este tipo de problemas, proponemos un esquema de ramificación y acotación (BB) para reducir los tiempos de cómputo del método solución. Para ello, introduciremos conjuntos cota superior que serán empleados en el método BB. Recordemos que un conjunto cota superior es un conjunto de vectores que o bien son eficientes o bien están dominados por al menos un punto eficiente.

Para este problema analizaremos los casos con y sin rotura, proponiendo conjuntos cota superior para cada caso.

Caso sin rotura)

Definiremos la propiedad ZIO para el caso con múltiples escenarios como sigue: diremos que un plan $\mathbf{Q}$ es ZIO para el problema EOQ dinámico multiescenario sí y sólo sí

$$Q_j \min\{I_1^{j-1},\ldots,I_M^{j-1}\} = 0 \text{ para } j = 1,\ldots,T.$$

Es evidente que la expresión anterior es una extensión natural de la propiedad ZIO para el caso con un único escenario. Por conveniencia, en el desarrollo de las demostraciones de siguientes resultados, reformularemos el modelo como un problema de flujo en red multicriterio, y se demostrará que las políticas ZIO multiobjetivo representan soluciones extremas y, en consecuencia, se corresponderán con flujos acíclicos en la red. Sea $G = (V, E)$ una red dirigida, donde V representa el conjunto de $n = (T+2)M+1$ nodos, y E el conjunto de $m = 3MT$ aristas. Los nodos se clasifican en: nodo de producción/pedido (nodo 0), nodos de demanda por escenario nd_s, $s = 1,\ldots,M$, y nodos intermedios. Los nodos intermedios se organizan por capas, de tal manera que, en el nivel j habrá M nodos denotados por n_s^j $s = 1,\ldots,M$, $j = 1,\ldots,T+1$. Además, se tienen M arcos desde el nodo 0 a cada nivel con idéntido flujo. Este situación se puede entender como un único flujo que es virtualmente multiplicado por M, de forma que la misma cantidad es enviada a cada nodo de ese nivel. Estos arcos se pueden considerar como un "pipeline" que en cierto momento se transforma en M ramas que reciben el mismo flujo. El arco desde el nodo de producción/pedido al nivel j se asocia a la variable de producción/pedido Q_j en el periodo j. La multiplicación virtual de la producción/reposición se debe a que los distintos escenarios no ocurren simultáneamente en realidad, sino que sólo uno de ellos está sucediendo. El arco desde 0 a n_s^j tiene un coste de $c_s^j(\cdot)$, $s = 1, \ldots, M$ y $j = 1, \ldots, T$.

De la misma manera, se establece un arco entre n_s^j y n_s^{j+1}, $s = 1, \ldots, M$, $j = 1, \ldots, T$, que se corresponde con la variable de estado I_s^j y su coste es de $h_s^j(\cdot)$. Por último, tendremos arcos que dejan cada nodo n_s^j hacia nd_s con valores de flujo d_s^j $s = 1, \ldots, M$ y $j = 1, \ldots, T$.

Para demostrar que las políticas ZIO no dominadas representan soluciones extremas necesitamos introducir la siguiente matriz A de incidencia nodo-arco, en la que las filas se corresponden con los M bloques de $T + 2$ restricciones de este problema.

	Q_1	Q_2	$\cdots$	Q_T	I_1^1	$\cdots$	I_1^{T-1}	I_1^T	$\cdots$	I_M^1	$\cdots$	I_M^{T-1}	I_M^T
	$(0,1)$	$(0,2)$	$\cdots$	$(0,T)$	$(1,2)$	$\cdots$	$(T-1,T)$	$(T,T+1)$	$\cdots$	$(1,2)$	$\cdots$	$(T-1,T)$	$(T,T+1)$
0	1	1	$\cdots$	1	0	$\cdots$	0	0	$\cdots$	0	$\cdots$	0	0
1	-1	0	$\cdots$	0	1	$\cdots$	0	0	$\cdots$	0	$\cdots$	0	0
2	0	-1	$\cdots$	0	-1	$\cdots$	0	0	$\cdots$	0	$\cdots$	0	0
$\vdots$			$\ddots$			$\ddots$			$\ddots$		$\ddots$		
T	0	0	$\cdots$	-1	0	$\cdots$	-1	1	$\cdots$	0	$\cdots$	0	0
$T+1$	0	0	$\cdots$	0	0	$\cdots$	0	-1	$\cdots$	0	$\cdots$	0	0
$\vdots$			$\ddots$			$\ddots$			$\ddots$		$\ddots$		
0	1	1	$\cdots$	1	0	$\cdots$	0	0	$\cdots$	0	$\cdots$	0	0
1	-1	0	$\cdots$	0	0		0	0		1	$\cdots$	0	0
2	0	-1	$\cdots$	0	0	$\cdots$	0	0		-1	$\cdots$	0	0
$\vdots$			$\ddots$			$\ddots$			$\ddots$		$\ddots$		
T	0	0	$\cdots$	-1	0	$\cdots$	0	0	$\cdots$	0	$\cdots$	-1	1
$T+1$	0	0	$\cdots$	0	0	$\cdots$	0	0	$\cdots$	0	$\cdots$	0	-1

A partir de la matriz A y el vector $\mathbf{Q}$, y denotando por $\mathbf{I}$ al vector

$$(I_1^1, \ldots, I_1^T, \ldots, I_M^1, \ldots, I_M^T),$$

se puede establecer, matricialmente, el conjunto de restricciones del problema como sigue

$$(\mathbf{Q}, \mathbf{I})A^t = -(-D, d_1^1, \ldots, d_1^T, 0, \ldots, -D, d_M^1, \ldots, d_M^T, 0).$$

Propiedad 7 La matriz A de restricciones tiene rango $MT + 1$.

Propiedad 8 Cualquier solución básica para este problema satisface que

$$Q_j \min\{I_1^{j-1}, \ldots, I_M^{j-1}\} = 0, \text{ para cualquier periodo } j, j = 1, \ldots, T.$$

Propiedad 9 El conjunto de soluciones no dominadas del problema contiene, al menos, una política ZIO.

Para computar los planes ZIO eficientes, necesitamos introducir notación adicional. Sea $I(j)$ el conjunto de vectores de estado al comienzo del periodo j. Advierta que $I(0) = I(T+1) = (0,\dots,0)$. Además, sea $D_i^{j,k} = \sum_{t=j}^{k-1} d_i^t$ la demanda acumulada desde el periodo j hasta el k en el escenario i y sea $(I_1^{j-1},\dots,I_M^{j-1}) \in I(j)$ un vector de estado dado en el periodo j. De igual manera, admitamos que el vector $(I_1^{j-1},\dots,I_M^{j-1})$ contiene una componente nula, por lo que la variable de decisión Q_j deberá ser distinta de cero para evitar roturas. Así, el conjunto de decisiones factibles correspondientes a un vector de estado $(I_1^{j-1},\dots,I_M^{j-1})$ en el periodo j es dado por

$$\Psi(j,(I_1^{j-1},\dots,I_M^{j-1})) = \begin{cases} 0 & \text{, si } I_i^{j-1} > 0 \text{ para todo } i \\ \max\limits_{1\le i\le M;\, j+1\le k\le T+1} \{0, D_i^{j,k} - I_i^{j-1}\} & \text{en otro caso.} \end{cases}$$

Asumiendo que $(I_1^{j-1},\dots,I_M^{j-1})$ contiene una componente igual a cero, se puede demostrar, fácilmente, que cualquier decisión $Q_j \neq \max\limits_{1\le i\le M} \{0, D_i^{j,j+l} - I_i^{j-1}\}$, $l = 1,\dots,T+1-j$, da lugar a una política no ZIO. De ahí que, dado un periodo j y un vector de inventario $(I_1^{j-1},\dots,I_M^{j-1}) \in I(j)$, el conjunto $F(j,(I_1^{j-1},\dots,I_M^{j-1}))$ de vectores de coste correspondientes a soluciones ZIO eficientes para el subproblema con inventario inicial $(I_1^{j-1},\dots,I_M^{j-1})$ viene dado por:

$$F(j,(I_1^{j-1},\dots,I_M^{j-1})) = \underset{Q_j\in\Psi(j,(I_1^{j-1},\dots,I_M^{j-1}))}{v-\min} \left\{ \begin{bmatrix} c_1^j(Q_j) \\ \vdots \\ c_M^j(Q_j) \end{bmatrix} + \begin{bmatrix} h_1^j(I_1^{j-1} + Q_j - D_1^{j,j+1}) \\ \vdots \\ h_M^j(I_M^{j-1} + Q_j - D_M^{j,j+1}) \end{bmatrix} \right.$$

$$\left. \oplus F(j+1,(I_1^{j-1} + Q_j - D_1^{j,j+1},\dots,I_M^{j-1} + Q_j - D_M^{j,j+1})) \right\}$$

Advierta que el conjunto completo de soluciones eficientes ZIO se obtiene cuando se resuelve $F(1,(0,\dots,0))$.

Propiedad 10 El algoritmo de programación dinámica anterior tiene una complejidad temporal de $O(4^T M^2)$.

Esta claro que desde un punto de vista computacional, el algoritmo anterior es ineficiente y, es por ello, que proponemos un método para obtener un conjunto solución aproximado. Este método consiste en obtener la solución óptima para cada escenario en $O(T^2)$, y en caso de soluciones infactibles, resolver, de nuevo, aquellos escenarios usando un vector de demanda donde cada componente representa la demanda máxima marginal. Es decir, la componente j en este vector vendría dada por la expresión $(\max\limits_{1\le i\le M} \{D_i^{1,j+1}\} - \max\limits_{1\le i\le M} \{D_i^{1,j}\})$. El esfuerzo computacional necesario para llevar a cabo este método es de orden $O(MT^2)$. Además, los planes generados de esta manera serán utilizados como conjunto cota superior inicial en el esquema de ramificación y acotación.

Caso con rotura)

Cuando I_i^j es negativo, representará una rotura de $-I_i^j$ unidades de demanda no satisfecha (acumulada) que deberá cubrirse por la producción/reposición durante los periodos j hasta T.

Asumiremos, por simplicidad, que $h_i^j(I_i^j)$ representa la función de coste de mantenimiento/rotura para el periodo j en el escenario i. Cuando I_i^j no es negativa, $h_i^j(I_i^j)$ continúa siendo el coste de tener I_i^j unidades de inventario al final del periodo j en el escenario i. Cuando I_i^j es negativo, $h_i^j(I_i^j)$ pasa a ser el coste de tener una rotura de $-I_i^j$ unidades al final del periodo j en el escenario i.

La adaptación de la propiedad del caso con único escenario (ver [30]) al caso multicriterio da como resultado la siguiente condición

$$\text{Si } Q_j > 0 \text{ y } Q_l > 0 \text{ para } j < l, \text{ entonces } I_i^k = 0, \text{ para algún } i \text{ y } k, \ j \leq k < l.$$

A diferencia de la propiedad ZIO para el caso multiescenario, la expresión anterior no da lugar a infactibilidad en los planes que genera, ya que cualquier plan que la satisfaga en un escenario dado va a ser factible para el resto de escenarios. Por lo tanto, se pueden determinar todos los planes cumpliendo la condición anterior de manera independiente en $O(MT^3)$. Podemos usar de nuevo la red introducida previamente para caracterizar los planes eficientes de esta versión del problema EOQ dinámico multiescenario, teniendo en cuenta que, ahora, la demanda para un periodo k se satisface a partir de la producción/reposición en un periodo anterior ($j \leq k$) o en uno posterior ($l \geq k$). Así, en la red subyacente, cada nodo (excepto el nodo productor 0) es accesible desde sólo uno de los siguientes nodos: el nodo productor, el nodo de mantenimiento predecesor o el nodo de rotura sucesor.

Propiedad 11 Cualquier solución básica para el problema EOQ dinámico multiescenario con roturas es acíclica.

Cabe mencionar que no todos los planes pertenecientes al conjunto de soluciones eficientes son básicos, y que el tiempo necesario para determinar el conjunto no dominado aumenta drásticamente con el tamaño de la entrada. Por ello, la obtención de planes eficientes entre los planes extremos no sólo debe verse como una aproximación razonable al verdadero conjunto no dominado, sino como un conjunto cota superior inicial a ser empleado en el esquema de ramificación y acotación.

El conjunto de decisiones factibles para un estado $(I_1^{j-1}, \ldots, I_M^{j-1}) \in I(j)$ es

$$\Phi(j,(I_1^{j-1},\ldots,I_M^{j-1}))=\begin{cases}0, & \text{si } I_i^{j-1}>0\ \forall\, i,\\ \{0\}\cup\{-I_i^{j-1}+D_i^{j,k}\},\ \begin{array}{l}k=j+1,\ldots,T+1\\ i=1,\ldots,M\end{array} & \text{, en otro caso.}\end{cases}$$

y, por lo tanto, podemos determinar el conjunto de vectores de coste eficientes para el estado $(I_1^{j-1},\ldots,I_M^{j-1})$ en el periodo j según la siguiente fórmula de recurrencia

$$F(j,(I_1^{j-1},\ldots,I_M^{j-1}))=\underset{Q_j\in\Phi(j,(I_1^{j-1},\ldots,I_M^{j-1}))}{v-\min}\left\{\begin{bmatrix}c_1^j(Q_j)\\ \vdots\\ c_M^j(Q_j)\end{bmatrix}+\begin{bmatrix}h_1^j(I_1^{j-1}+Q_j-D_1^{j,j+1})\\ \vdots\\ h_M^j(I_M^{j-1}+Q_j-D_M^{j,j+1})\end{bmatrix}+\right.$$

$$\left.\oplus F(j+1,(I_1^{j-1}+Q_j-D_1^{j,j+1},\ldots,I_M^{j-1}+Q_j-D_M^{j,j+1}))\right\}$$

Donde el conjunto $F(1,(0,\ldots,0))$ contiene el conjunto de soluciones eficientes que verifican la adaptación de la propiedad en caso de roturas a la situación con múltiples escenarios

El esquema de ramificación y acotación

Antes de introducir el método solución, se requiere introducir cierta notación adicional. Sea $\mathbf{D}_j\in\mathbb{N}_0^M$ un vector donde cada componente $i=1,\ldots,M$ se corresponde con $D_i^{1,j}$ y, además, denotaremos por $N(j+1,(I_1^j,\ldots,I_M^j))$ el conjunto de vectores asociados a los subplanes que alcanzan el vector de estado $(I_1^j,\ldots,I_M^j)\in I(j+1)$. Es decir,

$$\begin{aligned}N(j+1,(I_1^j,\ldots,I_M^j)) &= \{N(j,(I_1^{j-1},\ldots,I_M^{j-1}))\oplus(r_1^j(Q,I_1^j),\ldots,r_M^j(Q,I_M^j)):Q{\in}\mathbb{N}_0,\\ I_i^{j-1}+Q{-}D_i^{j,j+1} &= I_i^j,\ \text{para todo } i \text{ y } (I_1^{j-1},\ldots,I_M^{j-1})\in I(j)\}\end{aligned}$$

De igual forma, denotemos por $N^*(j+1,(I_1^j,\ldots,I_M^j))$ al conjunto de subplanes no dominados que alcanzan el estado $(I_1^j,\ldots,I_M^j)$. Con ello, podemos establecer el siguiente esquema de ramificación y acotación.

$$P(n,(I_1^n,\ldots,I_M^n))=v{-}\min[\sum_{j=1}^n c_1^j(Q_j)+\sum_{j=1}^{n-1} h_1^j(\sum_{k=1}^j Q_k-D_1^{1,j+1})+h_1^n(I_1^n),\ldots,$$

$$\sum_{j=1}^{n} c_M^j(Q_j)+ \sum_{j=1}^{n-1} h_M^j(\sum_{k=1}^{j} Q_k - D_M^{1,j+1}) + h_M^n(I_M^n)]$$

s.a.:

$$\sum_{j=1}^{k} Q_j \geq D_i^{1,k+1} \qquad k = 1, \ldots, n-1;\ i = 1 \ldots, M$$
$$\sum_{j=1}^{n} Q_j = D_i^{1,n+1} + I_i^n \quad i = 1 \ldots, M$$

Es evidente que $P(n, (I_1^n, \ldots, I_M^n)) = N^*(n+1, (I_1^n, \ldots, I_M^n))$. Ahora podemos determinar los valores eficientes del problema complementario $\overline{P}(n+1, (I_1^n, \ldots, I_M^n))$, es decir, el problema formado por los periodos desde $n+1$ a T con vector de inventario inicial $(I_1^n, \ldots, I_M^n)$, de la siguiente manera

$$\overline{P}(n+1, (I_1^n, \ldots, I_M^n)) = v\text{−}\min[\sum_{j=n+1}^{T} c_1^j(Q_j)+ \sum_{j=n+1}^{T-1} h_1^j(I_1^n+ \sum_{k=n+1}^{j} Q_k - D_1^{n+1,j+1})$$

$$+$$

$$h_1^T(I_1^n+ \sum_{k=n+1}^{T} Q_k - D_1^{n+1,N+1}), \ldots, \sum_{j=n+1}^{T} c_M^j(Q_j)+ \sum_{j=n+1}^{T-1} h_M^j(I_1^n+ \sum_{k=n+1}^{j} Q_k - D_M^{n+1,j+1})$$

$$+$$

$$h_M^T(I_M^n+ \sum_{k=n+1}^{T} Q_k - D_M^{n+1,T+1})]$$

s.a.:

$$\sum_{j=n+1}^{k} Q_j \geq D_i^{n+1,k+1} - I_i^n \quad k = n+1, \ldots, T;\ i = 1 \ldots, M$$
$$\sum_{j=n+1}^{T} Q_j = D_i^{n+1,T+1} - I_i^n \quad i = 1 \ldots, M$$

Cuando se admiten roturas en el modelo, el primer conjunto de restricciones en las formulaciones de P y $\overline{P}$ se debe eliminar. La aplicación del principio de optimalidad da la siguiente ecuación de recurrencia.

$$F(1, (0, \ldots, 0)) = \underset{\substack{(I_1^n, \ldots, I_M^n) \in I(n+1) \\ n = 1, \ldots, T-1}}{v - \min} (P(n, (I_1^n, \ldots, I_M^n)) \oplus \overline{P}(n+1, (I_1^n, \ldots, I_M^n)))$$

Ahora estableceremos los conjuntos cota inferior necesarios para desarrollar el esquema de ramificación y acotación. Diremos que LB es un conjunto cota inferior para un problema vectorial cuando cualquier solución no dominada o bien está incluida en él o es dominada por algún vector en LB. Asumiremos que conocemos a priori conjuntos cota inferior $LB(n+1,(I_1^n,\ldots,I_M^n))$ para cada subproblema $\overline{P}(n+1,(I_1^n,\ldots,I_M^n))$, y cotas superiores globales UB para el problema original $F(1,(0,\ldots,0))$.

Considere $f \in P(n,(I_1^n,\ldots,I_M^n))$ tal que para cualquier $lb \in LB(n+1,(I_1^n,\ldots,I_M^n))$: $f+lb \geq u$ para algún $u \in UB$. Es evidente que la rama generada por f no necesita ser explorada. De hecho, $u \in UB$ y, por lo tanto, existe $\widehat{f}$ eficiente (pudiendo ocurrir que $lb = \widehat{f}$) tal que $\widehat{f} \leq u$. Por lo tanto, $\widehat{f} \leq f+lb \leq f+$ (cualquier subplan posible). Esto implica que ningún subplan que resuelva el subproblema que deja f puede ser eficiente.

Sistema Inventario/Distribución

A diferencia de las variantes anteriores del modelo EOQ, en las que las decisiones se tomaban en relación a una única instalación, en este clase de sistema de inventario contaremos con dos instalaciones.

Este sistema fue ya estudiado por Crowston et al. [7], estableciendo la propiedad de *política de ratio entero,* por la cual la cantidad a pedir en la primera instalación debe ser un múltiplo entero de lo que se pida en la segunda instalación. Lamentablemente, esta propiedad deja de ser válida cuando el sistema cuenta con más de dos instalaciones, ver Williams [28].

Este tipo de sistemas entraña otro aspecto relevante relacionado con los costes de mantenimiento. Ilustraremos este efecto con el siguiente ejemplo: consideremos un sistema con dos niveles formado por un almacén (w) y un minorista (r). De acuerdo con la política de ratio entero tendremos que

$$Q_w = nQ_r \qquad n = 1,2,3,\ldots$$

donde Q_w y Q_r denotan la cantidad a pedir en el almacén y en el minorista, respectivamente.

La Figura 1.5 (Chapter 1) muestra las fluctuaciones del inventario en cada instalación para el caso $Q_w = 3Q_r$. Al reponer el minorista instantáneamente del almacén, el inventario en el almacén no sigue el típico patrón de dientes de sierra. Es por ello, que el inventario en este nivel no pueda determinarse de manera convencional y deba ser calculado a través del concepto de *inventario de nivel* introducido por Clark and Scarf [6]. Este inventario en el nivel j se defiene como el número de artículos en el sistema que actualmente están, o han pasado a través de él pero no han dejado aún el sistema. Usando esta definición, los inventarios en cada instalación siguen un patrón clásico de dientes de sierra.

En el Capítulo 4, se resuelve un sistema de inventario/distribución formado por un almacén y un minorista desde la perspectiva biobjetivo. De manera más específica, en vez de analizar un único criterio (la minimización de costes), se propone un criterio adicional que se corresponde con el número total anual de artículos dañados por manipulación inadecuada.

Introducimos, a continuación, los parámetros necesarios para establecer este modelo.

D	Razón de demanda constante, en unidades/año.
A_r	Coste fijo de hacer un pedido en el minorista, en unidades monetarias.
A_w	Coste fijo de hacer un pedido en el almacén, en unidades monetarias.
$\alpha(Q_r)$	Número de artículos dañados por envío desde el almacén al minorista, que dependerá de la cantidad pedida en el minorista.
h_r	Coste de mantenimiento en el minorista.
h_w	Coste de mantenimiento en el almacén.
J_r	Capacidad de inventario en el minorista, en unidades.
V	Capacidad del vehículo de reparto, en unidades.
Q_0	Cantidad máxima a pedir en el minorista, en unidades ($= \min\{J_r, V\}$)
HOC	Suma total de los costes anuales de mantenimiento y reposición.
DI	Número total de artículos dañados por año.

El objetivo consiste en minimizar ambos criterios (HOC y DI) de manera que se satisfaga la demanda. Las funciones que definen ambos objetivos vienen dadas por

$$HOC(Q_r, n) = \frac{A_r D}{Q_r} + \frac{A_w D}{n Q_r} + \frac{h_r Q_r}{2} + h_w \frac{(n-1)\, Q_r}{2}$$

y

$$DI(Q_r) = \alpha(Q_r)\frac{D}{Q_r}$$

Obviamente $0 \leq \alpha(Q_r) \leq Q_r \leq D$, y parece razonable pensar que a medida que Q_r aumenta también lo haga $\alpha(Q_r)$, pero, por el contrario, asumiremos que el incremento marginal del número medio de artículos dañados por envío disminuye. De esto último, se puede demostrar fácilmente que $\alpha(Q_r)$ es una función cóncava estrictamente creciente en $[0, D]$, con $\alpha(0) = 0$. Además, asumiremos que $DI(Q_r)$ es una función estrictamente decreciente.

El problema de sistema de inventario/distribución bicriterio queda definido por

$$\begin{array}{ll} v-\min & (HOC(Q_r,n), DI(Q_r)) \\ s.a. & Q_r \in (0, Q_0] \\ & n \ \in \mathbb{N} \end{array}$$

y el correspondiente conjunto de soluciones no dominadas por

$\mathcal{P} = \{(Q_r, n)|$ no hay otro $(Q_k^{'}, n^{'})$: $HOC(Q_k^{'}, n^{'}) \leq HOC(Q_r, n)$ y $DI(Q_r^{'}) \leq DI(Q_r)$, con una de las inecuaciones siendo estricta$\}$.

Antes de presentar el método solución, debemos introducir algunas definiciones. La función HOC alcanza su mínimo en el punto (Q_r^*, n^*) con

$$Q_r^* = \sqrt{\frac{2A_rD}{h_r - h_w}}$$

$$n^* = \sqrt{\frac{(h_r - h_w)A_w}{h_wA_r}}$$

Fijado el valor de n, el valor de Q_r que minimiza a $HOC(Q_r, n)$ es

$$\overline{Q}_r(n) = \sqrt{\frac{2D(A_rn + A_w)}{n^2h_w + n(h_r - h_w)}}$$

Por el contrario, si fijamos Q_r, el valor de n que hace a $HOC(Q_r, n)$ mínima es

$$\widehat{n}(Q_r) = \frac{1}{Q_r}\sqrt{\frac{2A_wD}{h_w}}$$

Asumiendo valores reales para n y Q_r, se puede demostrar fácilmente que las funciones $\overline{Q}_r(n)$ y $\widehat{n}(Q_r)$ son estrictamente decrecientes y convexas de n y Q_r respectivamente. A partir de la definición de $\widehat{n}(Q_r)$ se obtiene la función

$$\widehat{Q}_r(n) = \frac{1}{n}\sqrt{\frac{2A_wD}{h_w}}$$

Propiedad 12 Si $n \geq n^*$, entonces $\overline{Q}_r(n)$ es mayor o igual a $\widehat{Q}_r(n)$, y lo contrario cuando $n < n^*$.

Propiedad 13 Para un n dado, $n > 1$, las funciones $HOC(Q_r, n)$ y $HOC(Q_r, n-j)$, $1 \leq j \leq n-1$, se interceptan en un único valor $Q_r^{n,n-j} = \sqrt{\frac{2A_wD}{h_wn(n-j)}}$. Además, $\widehat{Q}_r(n) < Q_r^{n,n-j} < \widehat{Q}_r(n-j)$.

La caracterización de las soluciones eficientes necesita considerar las curvas de nivel de $HOC(Q_r, n)$, que vienen definidas por el conjunto $\mathcal{F}$

$\mathcal{F} = \{\varphi_l(Q_r, n) = 0 : \varphi_l(Q_r, n) = (h_r + h_w(n-1))nQ_r^2 - 2lnQ_r + 2D(A_w + A_rn),$
$l \geq HOC(Q_r^*, n^*)\}$.

Dado que la función $HOC(Q_r, n)$ es convexa, el conjunto $\varphi_l(Q_r, n) \leq 0$ corresponde a un conjunto convexo para cualquier valor $l \geq HOC(Q_r^*, n^*)$. Por otro lado, la caracterización de las soluciones no dominadas dependerá de las posiciones relativas de Q^* y Q_0, por lo que debemos distinguir dos situaciones, a saber, si $Q_0 \leq Q^*$ o lo contrario.

Si $Q_0 \leq Q^*$

Sea $\widehat{n}_0$ el valor entero de $\widehat{n}(Q_0)$ donde la función $HOC(Q_0, n)$ es mínima, es decir, $\widehat{n}_0 = \arg\{\min_{n\in\{\lfloor\widehat{n}(Q_0)\rfloor,\lceil\widehat{n}(Q_0)\rceil\}} HOC(Q_0, n)\}$, donde $\lfloor\widehat{n}(Q_0)\rfloor$ y $\lceil\widehat{n}(Q_0)\rceil$ representan, respectivamente, el menor y mayor valor entero más próximo a $\widehat{n}(Q_0)$. En caso que $HOC(Q_0, \lfloor\widehat{n}(Q_0)\rfloor) = HOC(Q_0, \lceil\widehat{n}(Q_0)\rceil)$, hacemos $\widehat{n}_0 = \lfloor\widehat{n}(Q_0)\rfloor$. De igual forma, asumiendo que $\overline{n}(Q_0)$ es el valor tal que $\overline{Q}_r(n_0) = Q_0$, denotemos por $\overline{n}_0$ al valor entero más cercano a $\overline{n}(Q_0)$. Como $Q_0 \leq Q^*$, se puede demostrar que $\overline{n}_0 \geq \lceil\widehat{n}(Q_0)\rceil$ y, por lo tanto, $\overline{n}_0 \geq \widehat{n}_0$. Además, sea q_l^i el mayor valor de Q_r donde la curva $\varphi_l(Q_r, n) = 0$ intercepta con la recta $n = i$.

Propiedad 14 Cuando $Q_0 \leq Q_r^*$, el conjunto $\mathcal{P}$ de soluciones eficientes, asumiendo que $l_0 = HOC(Q_0, \widehat{n}_0)$, es dado por

1) Si $\overline{n}_0 = \widehat{n}_0$,		: $\mathcal{P} = \{(Q_r, \overline{n}_0) : Q_r \in [\overline{Q}_r(\overline{n}_0), Q_0]\}$
2) Si $\overline{n}_0 = \widehat{n}_0 + 1$,		
	a) Si $\overline{Q}_r(\overline{n}_0) \leq q_{l_0}^{\overline{n}_0} \leq Q_0$	: $\mathcal{P} = \{(Q_r, \overline{n}_0) : Q_r \in [\overline{Q}_r(\overline{n}_0), q_{l_0}^{\overline{n}_0})\}$ $\cup\{(Q_0, \widehat{n}_0)\}$
	b) Si $q_{l_0}^{\overline{n}_0} > Q_0$	: $\mathcal{P} = \{(Q_r, \overline{n}_0) : Q_r \in [\overline{Q}_r(\overline{n}_0), Q_0]\}$
	c) en otro caso	: $\mathcal{P} = \{(Q_0, \widehat{n}_0)\}$
3) Si $\overline{n}_0 > \widehat{n}_0 + 1$,		
	a) Si $q_{l_0}^{\widehat{n}_0+1} = Q_0$	: $\mathcal{P} = \{(Q_0, \widehat{n}_0 + 1), (Q_0, \widehat{n}_0)\}$
	b) en otro caso	: $\mathcal{P} = \{(Q_0, \widehat{n}_0)\}$

Si $Q_0 > Q^*$

En adelante, $\overline{n}$ denotará el valor entero más cercano a n^* que minimiza a $HOC(\overline{Q}_r(n), n)$, es decir, $\overline{n} = \arg\{\min_{n\in\{\lfloor n^*\rfloor,\lceil n^*\rceil\}} HOC(\overline{Q}_r(n), n)\}$, con $\overline{Q}_r(n) \leq Q_0$. En caso que

$HOC(\overline{Q}_r(\lfloor n^*\rfloor),\lfloor n^*\rfloor) = HOC(\overline{Q}_r(\lceil n^*\rceil),\lceil n^*\rceil)$, hacemos $\overline{n} = \lfloor n^*\rfloor$ ya que, por concavidad de $\overline{Q}_r(n)$, el punto $(\overline{Q}_r(\lfloor n^*\rfloor),\lfloor n^*\rfloor)$ está a la derecha de $(\overline{Q}_r(\lceil n^*\rceil),\lceil n^*\rceil)$ y, en consecuencia, mejora el segundo criterio. Observe que, de la definición de $\overline{n}$, se tiene que $\overline{n} \geq \widehat{n}_0$ cuando $Q_0 > Q_r^*$.

Propiedad 15 Cuando $Q_0 > Q_r^*$, aquellos puntos (Q_r, n) con $n < \widehat{n}_0$ o $n > \overline{n}$ no se incluyen en $\mathcal{P}$.

Se puede demostrar que el conjunto $\mathcal{P}$ está formado por la unión de intervalos que están ubicados en distintas rectas n, con $\widehat{n}_0 \leq n \leq \overline{n}$. A partir de este punto, denotaremos por $\mathcal{P}(n)$ al conjunto de puntos no dominados que se encuentran en la recta n. Por lo tanto, el conjunto de soluciones no dominadas es dado por $\mathcal{P} = \bigcup_{n=\widehat{n}_0}^{\overline{n}} \mathcal{P}(n)$.

Propiedad 16 Para todo n con $\widehat{n}_0 < n \leq \overline{n}$, se verifica que $HOC(\overline{Q}_r(n), n) < HOC(\overline{Q}_r(n-1), n-1)$

Propiedad 17 Dadas las rectas n y $n-1$, los conjuntos $\mathcal{P}(n)$ y $\mathcal{P}(n-1)$ de soluciones no dominadas vienen dados por:

1.- Si $\overline{Q}_r(n-1) = \max\{Q_r^{n,n-1}, \overline{Q}_r(n-1)\}$ entonces

$\mathcal{P}(n) = [\overline{Q}_r(n), q_l^n)$, con $l = HOC(\overline{Q}_r(n-1), n-1)$, y $\mathcal{P}(n-1) = [\overline{Q}_r(n-1), a_1]$

2.- Si $Q_r^{n,n-1} = \max\{Q_r^{n,n-1}, \overline{Q}_r(n-1)\}$ entonces

$\mathcal{P}(n) = [\overline{Q}_r(n), Q_r^{n,n-1})$, y $\mathcal{P}(n-1) = [Q_r^{n,n-1}, b_1]$

donde los valores a_1 y b_1 dependerán de los puntos de intercepción con la curva $HOC(Q_r, n-2)$.

Propiedad 18 El conjunto de soluciones eficientes se puede obtener a través de comparaciones de pares de funciones HOC correspondientes a valores consecutivos de n.

El procedimiento solución para determinar el conjunto $\mathcal{P}$ se detalla en el Algoritmo 7.

Conclusiones

A lo largo de esta memoria se han abordado distintas variantes del modelo EOQ, ofreciendo métodos eficientes para resolverlas. Se considera la versión dinámica del modelo EOQ admitiendo restricciones de capacidad de inventario, demostrando que dependiendo de la estructura de costes se pueden diseñar distintos algoritmos eficientes. En concreto, se presenta una caracterización de planes óptimos así como el correspondiente algoritmo para el caso de costes cóncavos. En cambio, en ausencia de costes de setup (activación) y admitiendo que la estructura de costes es lineal , hemos demostrado que se puede desarrollar un algoritmo greedy de orden $O(T \log T)$ para obtener políticas óptimas. Además, hemos propuesto un algoritmo de orden $O(T \log T)$ basado en una técnica geométrica para el caso en el que las funciones de coste sean lineales y se admitan setups. También se han discutido extensiones del modelo EOQ bajo la perspectiva de la programación multicriterio. De manera mas específica, se ha considerado que el valor que toma la demanda en un periodo dado no es conocido, sino que se extrae de un conjunto finito de valores discretos, generando así distintos vectores posibles de demanda y dando lugar, por lo tanto, a diferentes escenarios. Esta variante se ha resuelto aplicando un esquema de ramificación y acotación y se ha presentado un método general para identificar el conjunto de soluciones eficientes. Por último, se ha analizado el sistema de Inventario/Distribución (I/D) considerando dos criterios y se ha desarrollado un método eficiente para caracterizar las soluciones no dominadas.

Bibliografía

[1] A. Aggarwal, J. K. Park. Improved Algorithms for Economic Lot Size Problems. *Operations Research*, 41 (1993), 549-571.

[2] R. K. Ahuja, A. V. Goldberg, J. B. Orlin, R. E. Tarjan. Finding Minimum-Cost Flows by Double Scaling. *Mathematical Programming* 53 (1992), 243-266.

[3] S. Axsäter. *Inventory Control.* Kluwer Academic Publishers, The Netherlands, 2000.

[4] H. Booth, R. E. Tarjan. Finding the Minimum-Cost Maximum Flow in a Series-Parallel Network. *Journal of Algorithm* 15 (1993), 416-446.

[5] A. Chikán (Ed.). *Inventory Models.* Kluwer Academic Publishers, The Netherlands, 1990.

[6] A. J. Clark, H. Scarf. Optimal Policies for a Multi-Echelon Inventory Problems. *Management Science*, 6(4) (1960), 475-490.

[7] W. B. Crowston, H. M. Wagner, J. F. Williams. Economic Lot Size Determination in Multi-Stage Assembly Systems. *Management Science*, 19(5) (1973), 517-527.

[8] R. J. Duffin. Topology of Series-Parallel Networks. *Journal of Mathematical Applications* 10 (1965), 303-318.

[9] A. Federgruen, M. Tzur. A Simple Forward Algorithm to Solve General Dynamic Lot Sizing Models with n Periods in $O(n \log n)$ or $O(n)$ Time. *Management Science*, 37 (1991), 909-925.

[10] J. Gutiérrez, J. Puerto, J. Sicilia. Bicriteria Trade-off Analysis in a Two-echelon Inventory/Distribution System. *Journal of the Operational Research Society*, 53 (2002), 468-469.

[11] J. Gutiérrez, J. Puerto, J. Sicilia. The Multiscenario Lot Size Problem with Concave Costs. To appear in the *European Journal of Operational Research*.

[12] J. Gutiérrez, A. Sedeño-Noda, M. Colebrook, J. Sicilia. A New Characterization for the Dynamic Lot Size Problem with Bounded Inventory. *Computers and Operations Research*, 30(3) (2003), 383-395.

[13] J. Gutiérrez, A. Sedeño-Noda, M. Colebrook, J. Sicilia. Solving the Dynamic Lot Size Problem with Time-varying Storage Capacities in $O(T \log T)$. Submitted to *Optimization Methods and Software*.

[14] F. W. Harris. How Many Parts to Make at Once, Factory, *The Magazine of Management*, 10 (1913), 135-136, 152.

[15] A. C. Hax, D. Candea. *Production and Inventory Management*. Prentice-Hall, Englewood Cliffs, New Jersey, 1984.

[16] S. F. Love. Bounded Production and Inventory Models with Piecewise Concave Costs. *Management Science*, 20(3) (1973), 313-318.

[17] A. S. Manne. Programming of Economic Lot Sizes. *Management Science*, 4 (1958), 115-135.

[18] E. Naddor. *Inventory Systems*. Wiley, New York, 1966.

[19] S. L. Narasimhan, D. W. McLeavey, P. J. Billington. *Production Planning and Inventory Control*, 2nd edition, Prentice-Hall, Englewood Cliffs, New Jersey, 1995.

[20] G. W. Plossl. *Production and Inventory Control*, 2nd edition, Prentice-Hall, Englewood Cliffs, New Jersey, 1985.

[21] A. Sedeño-Noda, J. Gutiérrez, B. Abdul-Jalbar, J. Sicilia. An $O(T \log T)$ Algorithm for the Dynamic Lot Size Problem with Limited Storage and Linear Costs. Submitted to *Computational Optimization and Applications*.

[22] E. A. Silver, D. F. Pyke, R. Peterson. *Inventory Management and Production Planning and Scheduling*, 3rd edition. Wiley, New York, 1998.

[23] R. J. Tersine. *Principles of Inventory and Materials Management*, 4th edition, Prentice-Hall, Englewood Cliffs, New Jersey, 1994.

[24] A. F. Veinott. Unpublished class notes for the Program in Operations Research at Stanford University. 1963.

[25] A. Wagelmans, S. Van Hoesel, A. Kolen. Economic Lot Sizing: An O($n \log n$) Algorithm that Runs in Linear Time in the Wagner-Whitin Case. *Operations Research*, 40 (1992), 145-156.

[26] H. M. Wagner, T. H. Whitin. Dynamic version of the Economic Lot Size Model. *Management Science*, 5(1) (1958), 88-96.

[27] C. D. J. Waters. *Inventory Control and Management.* Wiley, England, 1992.

[28] J. F. Williams. On the Optimality of Integer Lot Size Ratios in Economic Lot Size Determination in Multi-stage Assembly Systems. *Management Science*, 28 (1982), 1341-1349.

[29] R. H. Wilson, Scientific Routine for Stock Control, *Harvard Business Review*, 13 (1934), 116-128.

[30] W. Zangwill. A Deterministic Multi-period Production Scheduling Model with Backlogging. *Management Science*, 13 (1966), 105-119.

[31] W. Zangwill. Minimum Concave Cost Flows in Certain Networks. *Management Science*, 14 (1968), 429-450.

[32] P. H. Zipkin. *Foundations of Inventory Management.* McGraw-Hill, New York, 2000.

Preface

In practice, activities related to most organizations concern with using, transforming, distributing or selling some type of commodity or material. These tasks require not only having a physical location to store the goods, but also to apply a certain coordination and control over inventories. Generally, inventories are defined as those stored commodities with economical value, which are to be required in the future. Accordingly, in almost all industries, firms and organizations of economy, there exists a common activity corresponding to the control and the management of the inventory.

In developed countries, money invested in businesses related to inventory control and management represents a significant percentage of the annual gross inner product. Hence, inventory control and management plays an important role in economy.

Inventory Control is a relevant field in Operations Research that is concerned basically with the efficient management and control of stocks held by firms to satisfy customer demand. Thus, it provides the information needed for the day-to-day decisions required to settle tactical and economical objectives. Reconciling these conflicting objectives in a modern company, where responsabilities have been sharply divided and where managers have been encouraged to suboptimize by their performance measures, becomes a challenging problem; attempting to solve this problem is the primary function of inventory planning and control. Colloquially, the function of inventory control is to reconcile these objectives to meet the overall profit goals of the company.

Interest in the problems of optimal stock management at a scientific level goes back to the start of the 20^{th} century. However, the most important impulse came after World War II when several researches looked into the problem of optimal stocking. The position about inventory management has changed drastically through the past century; from the beginning when it was thought that carrying large inventory levels to cover fluctuations of demand was convenient, until nowadays where the goal consists of reducing inventories to minimum levels. As a result of this development, a huge number of articles in specialized journals and books devoted solely to inventory management and control are periodically published.

Precisely, in works credited to Harris (1913) and Wilson (1934), it was developed the germinal model to obtain the economic order quantity (EOQ) for a particular good. Given that applying this model has provided astonishing results in practice, it is not surprising that currently extensions of the EOQ model keep on being a topic under consideration. Indeed, new generalizations of the EOQ model can be found in literature, which reveal the evolution of inventory systems.

In accordance with this evolution, this monograph deals with extensions of the EOQ model. Accordingly, dynamic versions of the EOQ model, both to the multiscenario case and to the case with limited storage capacity, are addressed. Additionally, we also analyze the extension of the EOQ model to a two-echelon system, providing an algorithm to determine efficient ordering policies. Therefore, it is remarkable that approaches given throughout this work can be seen as a compilation of efficient techniques that can aid the decision maker to determine the more convenient inventory policy in terms of minimization of costs.

Specifically, in Chapter 2, the dynamic version of the EOQ model admitting limitations on the storage capacity is addressed. For this model, we show that depending on the cost structure different efficient algorithms can be applied. In particular, the characterization of optimal plans and its corresponding algorithm for the case of concave cost functions are included in Gutiérrez et al. [46]. In addition, when the cost structure consists of linear holding and reorder costs without setup costs, a greedy algorithm that runs in $O(T \log T)$ time can be devised as Sedeño-Noda et al. [81] pointed out. Moreover, in Gutiérrez et al. [47], it is proven that applying a geometrical technique yields an $O(T \log T)$ algorithm even when setup costs are involved in the linear cost structure. In Chapters 3 and 4, extensions of the EOQ model are discussed from the multicriteria programming viewpoint. In particular, in Chapter 3, the dynamic EOQ problem is studied considering that the demand for each period can take different values from a discrete set, generating so several scenarios. We solve this problem via a branch and bound scheme, providing a general method to characterize the set of efficient solutions. Several results in this chapter have already been published in a paper by Gutiérrez et al. [45]. Finally, Chapter 4 deals with the bicriteria two-echelon Inventory/Distribution (I/D) system. The set of non-dominated solutions for this problem is efficiently characterized. Some contributions in this chapter are compiled in a paper by Gutiérrez et al. [44].

Contents

Chapter 1

Foundations of Inventory Control

1.1 Introduction

From an operational point of view, inventories involve those goods held by an organization which are to be required by customers in the future. On the other hand, from the economic point of view, inventories represent invested money which is returned when the external demand for a specific commodity or facility is satisfied. Essentially, inventories can be seen as a buffer between variable and uncertain supply and demand. A frequent aim of inventory control is to provide this buffer at minimum cost. Therefore, *inventory control* consists of all activities and procedures used to ensure that the right amount of each item is held in stock. In this sense, inventories play an important role since supply and demand are difficult to synchronize perfectly. This lack of synchronization is essentially the result of four factors: *time*, *discontinuity*, *uncertainty* and *economy* (Tersine [88]). The time factor involves the time used in scheduling the production and in-process times of intermediate tasks up to the final product reaches the retailer or the wholesaler. It seems clear that few customers would be patient enough to wait for such an extended period of time on all their purchases. The second factor, that is, discontinuity, permits to manage the different proccess related to firm (retailing, distributing, warehousing, manufacturing and purchasing) in an independent way. The uncertainty factor concerns unpredictable events that modify the original plans of the organization. These events include mistakes of demand forecasting, strikes, equipment breakdown, etc. Finally, the economy allows the firm to get cheaper prices ordering or producing items in economic quantities (e.g. quantity discounts) or reducing transport costs.

Inventories can be classified into several categories depending on its *functionality.* Thus, *fluctuation inventories* are those inventories held in warehouse to cover the fluctuations in demand and supply. These inventories are also called *reserve stocks*, *safety stocks* or *stabilization stocks,* and may be provided in production plan so that production levels do not have to change in order to meet random variations in demand. *Anticipation inventories* represent those inventories ordered in advance of a peak selling season, a marketing promotion program or a plant shutdown period. *Lot-Size inventories* are those inventories obtained in larger quantities than are needed at the moment as a consecuence of the impossibility to manufacture or purchase items at the same rate at which they will be sold. In addition, *transportation inventories* stand for those inventories being moved from one place to another and, therefore, they cannot serve an useful function for plants or customers. Moreover, *speculative inventories* represent those inventories acquired by the organization to take advantage of lower prices.

Besides, inventories can be grouped by their *condition* during processing as well. Accordingly, *raw materials are* those materials needed to make components of the finished products. *Components/spares* are parts or subassemblies ready to be incorporated in the final assembly of a product. Also, *work-in-process inventories* are assumed to be those materials and components being in process or waiting between operations in the factory. Finally, *finished products* are those items carried in inventory at the warehouse or shipped to a customer.

Interest in the study of inventory systems has significantly increased in the last decades, and numerous publications have been devoted solely to this subject. Excellent reviews of inventory systems are given in Hax and Candea [49], Silver et al. [82], Chikán [18], Waters [98], Narasimhan et al. [65], Tersine [88], Plossl [69], Zipkin [110] and Axsäter [6], among others. Many articles on the subject now appear regularly in relevant specialized journals as Management Science, Operations Research, Journal of the Operational Research Society, Computers and Operations Research, European Journal of Operational Research, and many other journals.

It is merit to mention the seminal model from which inventory management has been developed. This model is referred to as EOQ-model (Economic Order Quantity) and it was introduced by Harris [48] in 1913, but the result is often credited to Wilson [101] who independently duplicated the model and marketed the results in the 1930s. The EOQ-model assumes that the demand for the item under study is known. In Section 1.3, a complete analysis of this model is developed.

We introduce below the terminology and basic definitions which are to be used throughout this work.

1.2 Preliminary Concepts and Terminology

As it was mentioned before, inventory control intends to manage buffers of goods, assuring that demand or customer service is covered. The main characteristics or components inherent to inventory systems are the *demand*, the *replenishment*, the *costs* and the *constraints*. We define in detail these features in the following sections.

1.2.1 Demand

Generally, demands cannot be controlled directly. They usually depend on decisions of people outside the organization which has the inventory problem. The *demand size* represents the quantity required to satisfy the demand for inventory. When the demand size is the same from period to period we say that it is constant. Otherwise we refer to it as variable. Inventory systems in which the demand size is known in advance will be referred to as *deterministic systems*. In deterministic systems with constant demand, it is convenient to use the *demand rate* which is the demand size per time unit. Occasionally, it is possible to recognize numerous ways by which quantities are taken out of inventory. Precisely, if we consider a period of time over which the demand occurs, this demand may be withdrawn at the beginning or at the end of the period; it may be withdrawn uniformly over the period or following a power pattern; etc (see Naddor [64]). These different ways by which demand occurs during a period will be referred to as demand patterns. Throughout, we focus our attention to the case where the demand occurs at the beggining of the period and also to the case in which the demand follows an uniform pattern.

1.2.2 Replenishment

The replenishment/reorder of inventory systems are generally controlled by decision makers. Generally speaking, the replenishment refers to the quantities that are scheduled to be put into inventories, to the time when decisions about ordering these quantities are made, and to the time when they are actually added to stock. Accordingly, we can identify the following elements concerning the replenishment. The *scheduling period* is the length of time between consecutive replenishments, and it is not always controllable. Moreover, the *replenishment/reorder* size represents the quantity scheduled for replenishment, and it is usually under control of the decision maker. In analogy to the demand, replenishing can follow different patterns, namely, uniform, instantaneous, power, etc. Finally, the *leadtime* is the length of time between scheduling a replenishment and its actual addition to stock, and it is generally not subject to control.

1.2.3 Costs

Since an inventory problem is basically a minimizing costs problem, it is important to distinguish the costs related to inventory systems. They are the more relevant economic components to any inventory decision model, and they can be grouped in several general costs, which are itemized as follows. The *purchase unit cost* represents the unit purchase price of an item if it is obtained from external source, or the unit production cost when it is produced internally. Furthermore, the fixed charge or *setup cost* stands for the expense of issuing a purchase order to an outside supplier or the internal production setup costs. The setup cost comprises the costs of changing over the production process to produce the ordered item and it depends on the number of orders placed. The *replenishment/reorder cost* includes those charges related to place an order. This cost is usually assumed to vary directly with the size of the order. In addition, the *holding/carrying cost* incorporates the capital/opportunity costs, taxes, insurance, handling, storage, shrinkage, obsolescence, and deterioration. Normally, the holding costs are proportional to the size of inventory investment. Finally, the *stockout/shortages/backlogging cost* reflects the economic consecuence of an external or an internal shortage. The external shortage occurs when the customer's demand is not fulfilled, and the internal one when an order placed within the organization is not filled. The quantification of these costs has long been a difficult and unsatisfactorily resolved issue. For this reason, many organizations avoid the estimating problem by specifying customer service levels.

1.2.4 Constraints

Constraints in inventory systems deal with various properties that in some way place limitations on the components discussed in the previous sections. They can be classified in unit constraints (discrete or continuous units), demand constraints, replenishment constraints and cost constraints. In particular, throughout this monograph we consider limitations on the replenishment size due to storage capacities in Chapter 2 or to the capacity of the vehicle that distributes the commodity in Chapter 4. Moreover, we consider that order quantities should be integer in Chapters 2 and 3.

1.2.5 Inventory Policies

It is clear that an *inventory problem* is a problem of making optimal decisions with respect to an inventory system. In other words, an inventory problem is concerned with the making of decisions that minimize the total cost of an inventory system. Decisions that are made always affect the costs, but such decisions can rarely be made directly in terms of costs. Decisions are usually made in terms of *time* and *quantity*. Consequently, the time and quantity elements are the variables that are subject to control in an inventory system. Accordingly, the decision maker must answer the following questions: "When should an order be placed?" and "How much should be ordered?". Depending on the type of reviewing, periodic or not, and depending on whether the reorder quantity is fixed or not, several *ordering policies* are introduced. The first question is usually answered in one of the following ways:

1. Inventory should be replenished when the amount in inventory is equal to or below s_o[1] quantity units.

2. Inventory should be replenished every t_o time units.

The second question is also usually answered in one of two ways:

1. The quantity to be ordered is Q_o quantity units.

2. A quantity should be ordered so that the amount in inventory is brought to a level of S_o quantity units.

The quantities s, t, Q and S will be referred to as the *reorder point*, *scheduling period*, *lot-size*, and *order level*, respectively, and they are used in static inventory systems, that is, inventory systems where the parameters do not change with time. The stationary inventory systems can be grouped depending on which *inventory policy,* namely, (s, Q), (t, S), (s, S) or (t, Q) is to be applied. In addition, when the leadtime is not negligible the reorder point and the order level will be designated by z and Z, respectively. Consequently, the corresponding policies are (z, Q), (t, Z) and (z, Z). An exhaustive compilation of these inventory systems and their solution methods can be found, among others, in Naddor [64], Tersine [88], Plossl [69], Narasimhan et al. [65], Chikán [18], Silver et al. [82] and Axsäter [6].

The notation above is confined to the case in which the constant lot-size should be determined for a single-item, single-facility and constant input data inventory system. However, when parameters can change with time, the inventory systems are usually referred to as *dynamic* systems. In particular, the *dynamic lot-size* problem assumes that a finite planning horizon is divided into T periods of time and an order plan $\mathbf{Q} = (Q_1, Q_2, \ldots, Q_T)$ should be determined. Given a period i, with $1 \leq i \leq T$, the demand, the replenishment cost and the holding cost for this period are denoted by d_i, $C_i(Q_i)$ and $H_i(I_i)$, respectively. Notice that $C_i(Q_i)$ and $H_i(I_i)$ are functions of the quantity ordered Q_i and the inventory level held I_i, respectively. In particular, Q_i stands for the quantity to be ordered at the beginning of period i, whereas I_i represents the inventory level at the end of that period. It is widely accepted that I_0 and I_T are zero. In case of being I_0 distinct to zero, the demands of the first periods

[1] The subscript o refers to an optimal value.

should be set to zero since they can be fulfilled from the initial inventory. In absence of shortages, the total cost of the systems is commonly expressed as the sum of the replenishment and holding costs. Finally, the goal consists of determining an order plan $\mathbf{Q} = (Q_1, Q_2, \ldots, Q_T)$ such that minimizes the total cost satisfying the demands of all periods. When an inventory capacity is considered, the problem is referred to as the *dynamic lot-size problem with limited storage*. This inventory capacity can be either fixed or variable. In the first case, it is assumed that the capacity of the warehouse W is constant through the planning horizon. Otherwise, the capacity of the warehouse changes with time and it is denoted by W_i, $i = 1, \ldots, T$.

1.2.6 Inventory Problems Classification

The combinations of the characteristics above mentioned result in different classifications of inventory problems. Following the classification of inventory problems due to Tersine [88], we show in Figure 1.1 the main types of problems that can be found in inventory theory.

The perpetual inventory system designated by P_1 in Figure 1.1 is a system in which a order is placed every time the inventory position reaches a prefixed inventory level. The periodic inventory system, P_2, orders stock on a time cycle and, also, decisions on inventory replenishment are only made at discrete points in time. Additionally, the material requirements planning (MRP) system, P_3, is based on planning dependent demand requirements for finished items using a time-phased format. Moreover, the goal in distribution requirements planning (DRP) systems denoted by P_4 consists of ordering stock to satisfy distribution center requirements in multi-echelon networks. Finally, in the single order quantity system or P_5, it is assumed that the same quantity is ordered to meet unique or short-lived requirements.

1.3 The Economic Order Quantity (EOQ) Model

The economic order quantity model (EOQ model) was first published in the book of F. Harris (1913)[48] and it is considered the first classical model of inventory control. This model is included in class of problems P_1, and it assumes that shortages are not allowed and the replenishment is instantaneous. In addition, the leadtime is negligible. The total cost, as a function of the lot-size, consists of a holding cost, which is dependent on the lot-size held in stock and a replenishment cost that depends on the number of replenishments. The goal is to determine the optimal lot-size which minimizes the total cost. One of the characteristics more significant

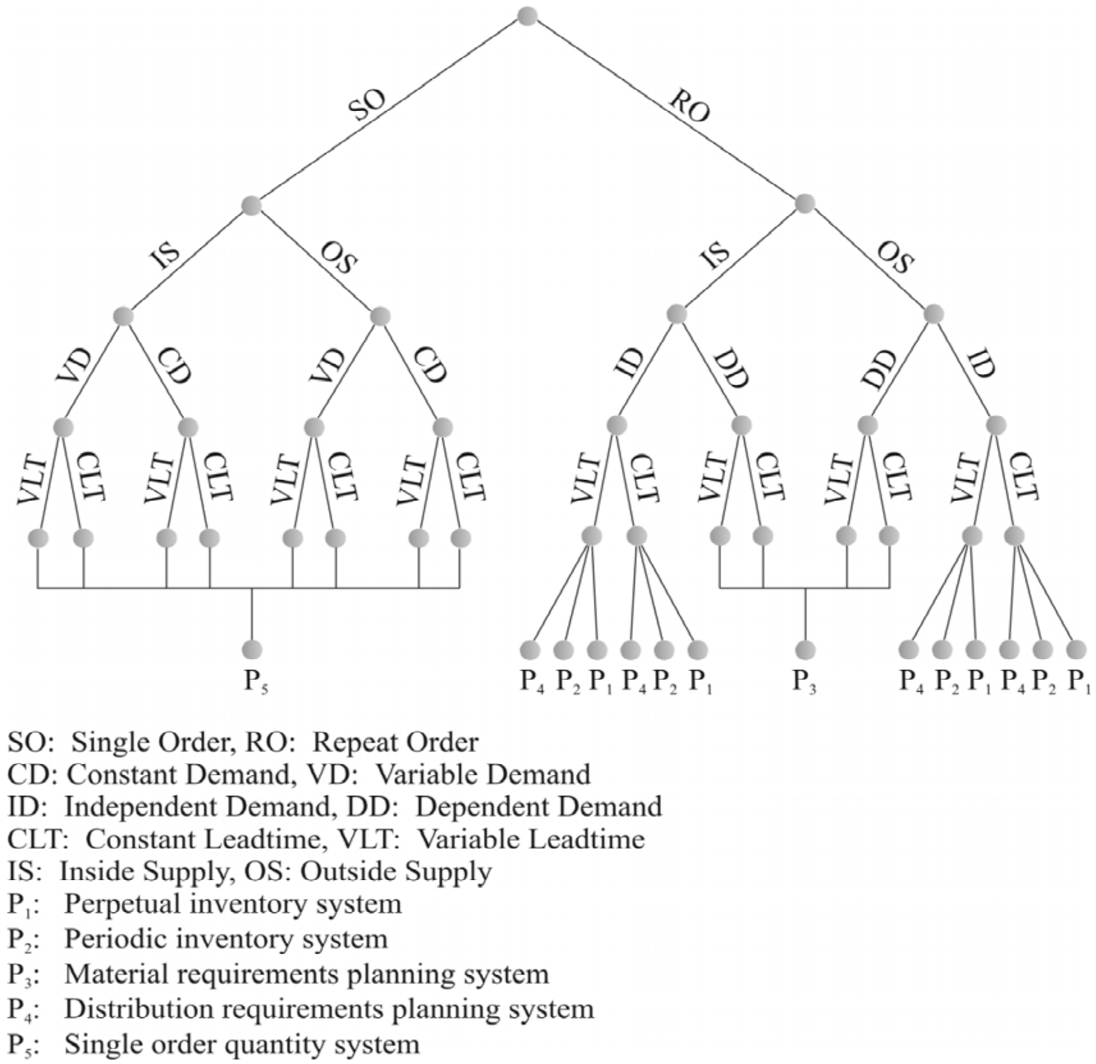

Figure 1.1: Inventory problems classification.

of the EOQ-model is its robustness. This feature justifies the fact that the EOQ-model keeps being successfully used currently. The economic order quantity EOQ is the order amount which balances the costs of inventory holding against those of placing inventory replenishment.

The formula which expresses the value of the EOQ is also credited to R. H. Wilson (1934)[101]. In the German literature, it is often called the Andler formula since this formula is proposed in the book of K. Andler [5] published in 1929. This model is described also in the books of B. Margansky (1933) [62] and K. Steffanic-Allmayer (1927) [84].

The assumptions of the EOQ model are very restrictive and they are rarely fulfilled in practice. However, the EOQ is still the model most cited and used for an approximated solution in inventory control. The success of the EOQ model lies on the following facts: i) the formula is very simple to implement, to understand and to apply even for people not well trained in operations research; ii) the model is robust with respect to the input data, i.e., an inaccurate estimation of the input data does not result in a significant increase in cost as compared with the optimal solution of the instance with accurately estimated parameters.

The EOQ model corresponds to a lot-size system in which the order quantity should be determined, demands are constant and a (s, Q) policy is applied with $s = 0$, hence, shortages are not permitted. Therefore, the only significant costs are the carrying cost $H(Q)$ and the replenishment/production cost $C(Q)$, and the goal consists of finding the balance between both costs. The inventory fluctuations in the EOQ model are illustrated in Figure 1.2. From this figure, it is clear that the demand occurs at a constant rate of d item units per time unit and leadtime is zero. The objective consists of determining the economic order quantity Q to be replenished/produced every t time units at minimum cost. Hence, variables Q and t are related through the following expression

$$Q = dt$$

Giving that the scheduling period t determines the time interval between consecutive replenishments/productions, it can be easily derived the number of replenishments/productions just computing the quotient $1/t = d/Q$.

Regarding the costs, it is assumed that there is a constant unit carrying cost h with dimension [money unit]/([item unit][time unit]) and a constant unit replenishment cost c with dimension [money unit]. Hence, the total cost TC as a function of Q is given by

$$TC(Q) = \frac{hQ}{2} + \frac{cd}{Q}$$

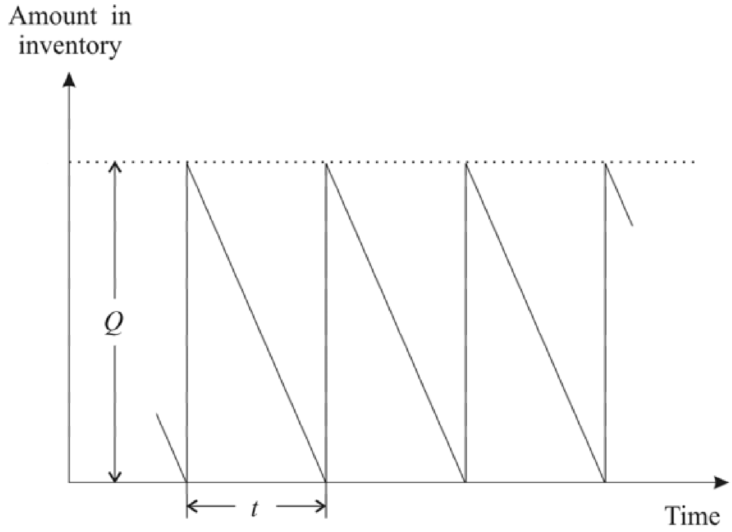

Figure 1.2: Graphical representation of the EOQ model.

Notice that the first term in TC corresponds to the average holding cost whereas the second term is the average replenishing cost. An illustration of these costs is depicted in Figure 1.3.

To obtain the optimal EOQ, it suffices to differenciate $TC(Q)$ with respect to Q and to set this result to zero. Accordingly, the optimal lot-size Q_o is given by

$$Q_o = \sqrt{\frac{2dc}{h}}$$

and, its corresponding optimal cost is $TC_o = \sqrt{2hcd}$.

We have already pointed out that the EOQ model is robust, in the sense that the optimal solution for the problem with not very bad estimated parameters does not significantly differ from the actual optimal solution. This situation arises, for instance, when the decision maker uses estimates for the parameters of the system and these differ from the true values. In particular, suppose now that instead of the optimal Q_o the decision maker use another lot-size Q', which is related to Q_o by

$$Q' = bQ_o \quad b > 0$$

Let TC' designate the total cost of the system when Q' is applied. In addition, let the ratio TC'/TC_o be used as a measure of sensitivity. In Table 1.1, extracted

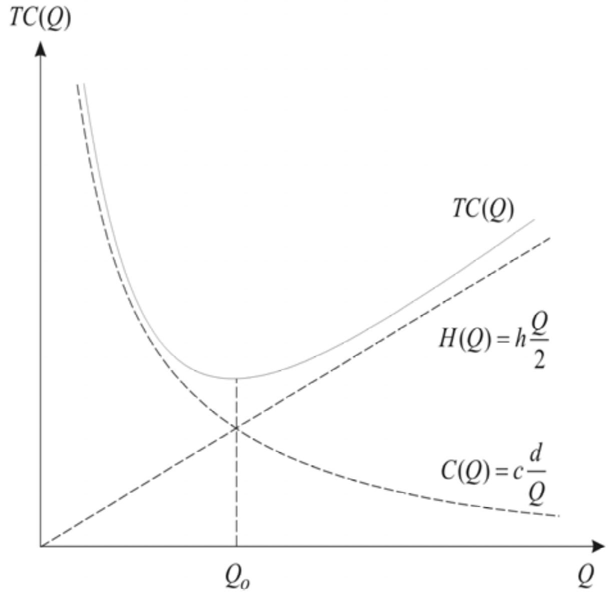

Figure 1.3: Costs of the EOQ model.

b	0.5	0.8	0.9	1.0	1.1	1.2	1.5	2.0	3.0
TC'/TC_o	1.250	1.025	1.006	1.000	1.005	1.017	1.083	1.250	1.667

Table 1.1: Sensitivity of the EOQ model.

from Naddor [64], it is shown the values of TC'/TC_o for various values of b. It is worth noting that underestimating Q_o by 50%, that is, when $b = 0.5$ leads the additional cost to be 25% of the minimum cost. For a more complete analysis of sensitivity of the lot-size system see Naddor [64].

Below, we introduce the extensions of the EOQ model to the time-varying demand and multi-echelon cases and, we present also the most relevant contributions collected in the literature for these topics.

1.4 Extensions of the EOQ Model

Among the different classes of inventory systems, this monograph mainly deals with those concerning with the dynamic lot-sizing and the multi-echelon problem. Due to the applicability of both systems in practice, we think that the results to be introduced in subsequent chapters can play a relevant role in the inventory management of a firm.

1.4.1 Deterministic Dynamic Lot-sizing

This type of inventory systems is included in class of problems P_2 since they assume that the demand for a given item does not follow an uniform pattern, but it can change with time. Therefore, the planning horizon must be divided into T periods and, the demand for each period should be determined and fulfilled at the beginning of that period, not allowing shortages. In the classical version of this model, proposed by Wagner and Whitin [97] and, independently by Manne [61], it is assumed that in each period there is a linear holding cost, a fixed setup cost and a constant production/reorder cost. The total cost is expressed as the sum of the costs associated to each period. In addition, the initial and final inventory levels are assumed to be zero. The solution consists of determining a production/reorder time-phased vector or plan which minimizes the total cost. These authors stated that among the optimal solutions there always exists one such that an order is placed in a period when the inventory at the end of the predecessor period is zero. This optimality condition is usually referred to as ZIO (*Zero Inventory Ordering*) property and the solutions which hold this property are called ZIO policies. Based on this property, an $O(T^2)$ algorithm can be devised to determine an optimal solution even when the cost functions are concave in general (Veinott [91]). Most recently, Federgruen and Tzur [33], Aggarwal and Park [1] and Wagelmans et al. [95] provided $O(T\log T)$ algorithms using very different techniques for the case with time-varying reorder costs, which run in linear time when reorder costs are constant. Zangwill [105] proved that the *production/reorder planning* problem can be seen as a network flow problem where there exists at least one optimal solution (ZIO policy), which corresponds to an acyclic flow of the underlying network. The extension to the case with shortages was addressed by Zangwill [104]. Under this assumption and considering concave costs, Zangwill argued that among the optimal solutions there always exists one satisfying that between two adjacent periods with production/order distinct to zero there is at least one period with inventory level equal to zero. This author exploited this property to devise an $O(T^3)$ algorithm. In addition to these optimal solution methods, there exist others in the literature considering different assumptions with respect to inventory positions and cost structure (see, for instance, Wagner [96], Zabel [103], Eppen et al. [29] and Zangwill [106]).

When in each period the production quantity is limited by the capacity of the resources, namely, manpower and/or material, the model is usually referred to as *Capacitated Lot-Sizing Problem* (CLSP). It is well-known that in most cases the CLSP can not be efficiently solved (see Florian et al.[37], Bitran and Yanasee [12], Chen and Thizy [17] and Baker et al. [7]). Only when the production capacities and setup costs are constant an optimal solution can be determined in $O(T^4)$ (see Florian and Klein [36] and Bitran and Yanasee [12]). Recently, Van Hoesel and Wagelmans [90] developed an $O(T^3)$ algorithm that solves the constant capacities dynamic lot-sizing problem with concave production costs and linear holding costs. There also are a significant number of works concerning with the CLSP, but they are of heuristic nature (see, e.g., Salomon [79]). Unlike the CLSP, which has been extensively studied in the literature, the Dynamic Lot-sizing Problem with Limited Inventory (DLSPLI) problem has been barely considered by researchers. In particular, Love [60] developed an $O(T^3)$ algorithm based on dynamic programming for concave cost functions in general, and Dixon and Poh [24] proposed a heuristic method for when the model involves more than one item.

Assuming that $C_t(\cdot)$ and $H_t(\cdot)$ represent, respectively, concave functions of the order amount Q_t and the inventory level I_t in period t, $t = 1, \ldots, T$, the DLSPLI can be stated as follows:

$$\min \sum_{t=1}^{T} \left(C_t(Q_t) + H_t(I_t)\right)$$

$$\begin{aligned} s.t. \quad & \\ & I_0 = I_T = 0 \\ & I_{t-1} - I_t + Q_t = d_t && t = 1, \ldots, T \\ & x_t, I_t \in \mathbb{N}_0 && t = 1, \ldots, T \end{aligned}$$

where $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$.

In this model, the solution method is based on the ZIO property, which postulates that among the optimal plans there is at least one satisfying

$$I_{t-1} Q_t = 0 \quad t = 1, \ldots, T$$

Accordingly, there exists at least one plan in which the optimal decision consists of producing/ordering the sum of demands from one given period to a subsequent one.

In Chapter 2 of this monograph, we address the single-item DLSPLI considering both general concave cost structure and linear cost functions. In the first case, that is, when the cost structure is defined by general concave cost functions, it is proven that an alternative characterization of the optimal plans is possible. This new approach leads to devise an $O(T^3)$ algorithm that runs in $O(T)$ expected time when the demand for a given period varies between zero and the maximum capacity for that period. On the other hand, in case of linear holding and replenishment costs and considering setup costs, the geometrical technique given in Wagelmans et al. [95] can be extended to develop an $O(T \log T)$ algorithm. Additionally, when setup costs are negligible, one can devise an $O(T \log T)$ algorithm based on the properties of the network underlying to the problem.

In contrast to deterministic lot-sizing problems where the decision maker knows in advance the values of the input data through the planning horizon, stochastic lot-sizing systems assume that the input data can be adjusted to density functions with known parameters. In particular, in Chapter 3, the case is addressed when the demand is discrete and it is uniformly distributed. In other words, the demand can take any value from a finite set of discrete values. Therefore, several scenarios can arise, and hence the efficient solution set should be determined. It is proven that extensions of the ZIO property can be adapted successfully to the multi-scenario case.

1.4.2 Multi-echelon Systems

In contrast to the original EOQ model, where decisions are made for a single location, this type of inventory systems involves more than one installation and, hence they are included in class of problems P_3. In general, multi-echelon systems can be classified into three major categories, namely, *distribution systems* (also called *arborescent systems*), *assembly systems* and *hibrid systems*. In these systems, inventory control is applied across the entire supply chain. Precisely, Silver at al. (1998) [82] define *supply chain management* (SCM) as the management of materials and information from suppliers to component producers to final assemblers to distribution (warehouses and retailers) and, ultimately, to the consumer (see Figure 1.4 for a sketch of a supply chain).

The main inconvenient in the supply chain structure is that decisions on an installation affects to the rest of locations. In particular, overestimated orders at one installation make the lot-sizes increase at predecessor locations. This phenomenon is often called the *bullwhip effect* (see Silver at al. [82] for illustrative real world examples).

The interest for supply chain management and for multi-echelon inventory control in such chains is growing rapidly. Supply chains are not always part of a single company, but often different firms work together to improve the coordination of the total material flow. The *Just In Time* (JIT) philosophy is a particular case where several supply chains corresponding to different companies are coordinated to decrease inventory levels. For further information on JIT systems, the reader is refereed to Silver et al. [82], Tersine [88], Zipkin [110], Narasimhan et al. [65] and

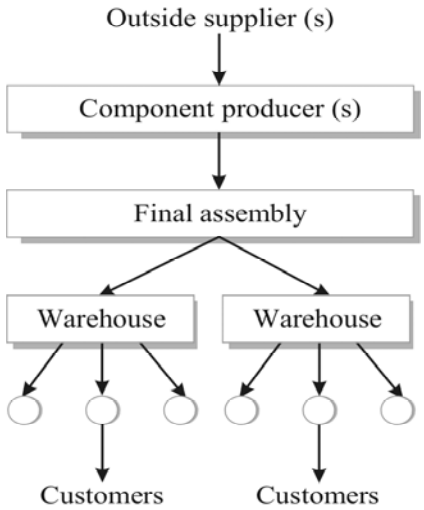

Figure 1.4: A scheme of a Supply Chain.

Waters [98], among others.

Multi-echelon inventory systems are common in both distribution and production. In particular, Figure 1.5 shows a tipical distribution system, which is characterized in that each installation has at most a single immediate predecessor.

It is worth noting that when each installation has also at most one immediate succesor the model is refereed to as *serial system*, which is the simplest fashion within the multi-echelon systems. The two-echelon serial system with one-warehouse and one-retailer was studied by Crowston et al. [21]. These authors introduced the well-known *integer lot-size ratio,* which states that the lot-size at the first location should be an integer multiple of the lot-size at the second installation. Unfortunately, as Williams [100] proved, this property is no longer valid when more than two installations are involved in the system.

Another relevant aspect in multi-echelon systems is that concerning the holding costs. In particular, the difficult arises when the average inventory quantity should be determined. We illustrate this issue with the following example: consider a two-echelon serial system consisting of one warehouse (W) and one retailer (R). According to the integer lot-size ratio, one must have

$$Q_W = nQ_R \qquad n = 1, 2, 3, \ldots$$

where Q_W and Q_R denote the order amount at the warehouse and at the retailer, respectively.

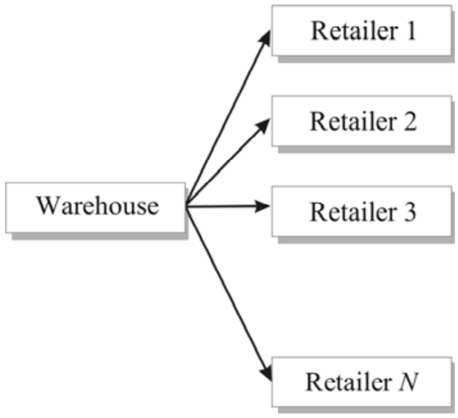

Figure 1.5: Distribution inventory system.

Figure 1.6 shows the behavior of the inventory levels of both installations in such a system for the particular case $Q_W = 3Q_R$. Notice that the retailer instantaneously replenishes from the warehouse and, hence the inventory at the warehouse does not follow the usual sawtooth pattern. It is clear that this inventory could be calculated using conventional definitions of inventories, however, the calculations become quite complex. Instead, it is preferable to use the concept known as *echelon stock* stated by Clark and Scarf [20]. The echelon stock of echelon j is defined as the number of item units in the system that currently are at, or have passed through, echelon j but have not yet left the system. By virtue of this definition and assuming that the demand for the finished item follows an uniform pattern, the inventory at each installation has a sawtooth pattern with time.

Nevertheless, obtaining the total inventory carrying costs cannot be done simply multiplying each echelon stock by its unit holding cost as usual. In case of directly summing these quantities makes the same physical stock units are to be counted in more than one echelon inventory. The way to overcome this difficulty is to value any specific echelon inventory at only the value added at that particular installation. In other words, the *echelon holding cost* actually is the incremental cost of moving the item from the warehouse to the retailer, thus, the echelon cost at the warehouse is $h_W^{'} = h_W$ and at the retailer is $h_R^{'} = h_R - h_W$, where h_W and h_R represent the actual unit holding costs at both locations. In a general multi-echelon system, the echelon holding cost at a particular installation is given by

$$h_j^{'} = h_j - \sum_{i \in Pred(j)} h_i$$

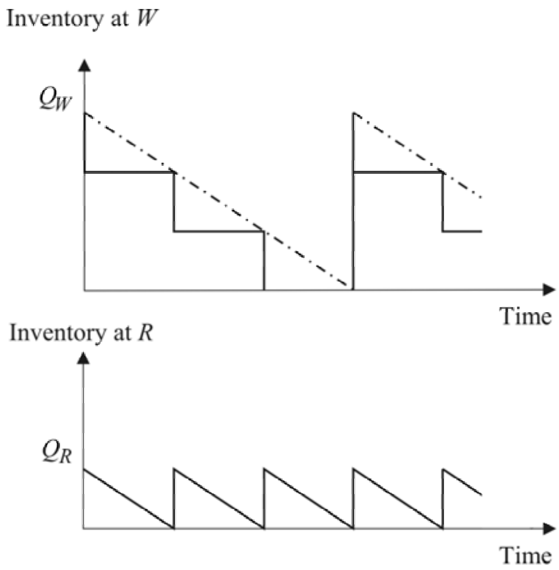

Figure 1.6: Fluctuations of inventory levels at the warehouse and the retailer.

where the set $Pred(j)$ contains all the installations immediately predecessor to echelon j.

In Chapter 4, the two-echelon inventory/distribution system consisting of one warehouse and one retailer is solved from the bi-objective perspective. Specifically, instead of analyzing only one criterion, namely, the annual inventory cost, an additional objective is proposed, which corresponds to the annual total number of damaged items by improper shipment handling from the warehouse to the retailer. The whole Pareto-optimal solution set is determined and, additionally, a previous result due to Bookbinder and Chen [14] that analyzes the same problem with equivalent cost structure is corrected.

1.5 Mathematical Background

We include in this section several topics, which are to be used in subsequent chapters of this work. In particular, the Dynamic Programming (DP) approach and the structure of directed network are used in Chapter 2. Moreover, the extension of the DP approach to the multicriteria case, i.e., Multiobjective Dynamic Programming in combination with a Branch and Bound scheme is exploited in Chapter 3 to determine the whole efficient solution set of the Multiscenario Dynamic Lot-sizing problem. The dynamic determination of the lower envelope of a set of points in $\mathbb{R}^2$ is discussed since it is used as solution method for the dynamic lot-sizing problem with linear and setup costs dealt with in Chapter 2. In addition, multiobjective optimization is commented since problems in Chapters 3 and 4 are of multicriteria nature.

1.5.1 Multicriteria Optimization

In most real-world applications we can find several conflicting criteria. Specifically, in production planning, one could be interested in both maximizing the total net revenue and, on the contrary, minimizing shortages or overtime. Thus, the concept of optimal solution characteristic of the single-objective optimization should be replaced by *efficient solution* (also refereed to as *non-dominated* or *Pareto-optimal* solution). Colloquially, we say that a solution is non-dominated when slight changes of the criteria values do not yield improvements in the objective functions. Formally, a multiple criteria program with k criterion can be stated as follows:

$$\begin{array}{ll} & v - \min(\max)(f_1(\mathbf{x}), f_2(\mathbf{x}) \ldots, f_k(\mathbf{x})) \\ s.t. & \\ & \mathbf{x} \in \mathbf{X} \subseteq \mathbb{R}^n \end{array}$$

where $v - \min(\max)$ (vector minimization(maximization)) is used to differentiate the problem from the minimization(maximization) single-objective problem.

Accordingly, we denote by $\mathcal{P}$ the non-dominated solutions set (or, equivalently, the Pareto-optimal solutions set), which is defined as follows:

$$\begin{aligned} \mathcal{P} = & \{\mathbf{x} \in \mathbf{X} | \text{ there is no } \mathbf{y} \in \mathbf{X} \text{ so that } f_i(\mathbf{y}) \leq f_i(\mathbf{x}),\ i = 1, \ldots, k, \text{ with} \\ & \text{at least one inequality being strict}\} \end{aligned}$$

Further details on methods and applications for multicriteria optimization can be consulted in Steuer [85], Ehrgott [27], Ehrgott and Gandibleux [28], Goicoechea et al. [42] and Zeleny [107], among others.

1.5.2 Dynamic Programming

A *sequential decision process* represents an activity that entails a sequence of actions taken toward some goal. Using the definition of Denardo [23], *Dynamic Programming* is the collection of mathematical tools used to analyze sequential decision process.

In a dynamic programming problem the following concepts are involved: the *state space*, the *functional equation* and the *principle of optimality*. Since dynamic programming handles sequential decision processes, the state of the problem changes when decisions in each stage are made. Therefore, the problem can be characterized by a set of states E. This state set usually contains only one initial state I_0, and, at least, one final state I_T. Applying a decision Q to one state I_j yields this state to transit to another state I_k, and also, to incur a cost/benefit, say t_{jk}, which depends on Q and I_j. A sequence $\mathbf{Q} = Q_1, Q_2, \ldots, Q_T$ of decisions is refereed to as a *plan* or *policy*. It is said that a policy $\mathbf{Q}$ is *feasible* when starting from the initial state, it reachs one of the final states. The cost associated to a policy $\mathbf{Q}$ corresponds with the sum of the costs of each decision contained in $\mathbf{Q}$. We can now state a general dynamic discrete optimization problem in the following way:

$$\begin{aligned} & \min(\max) f(\mathbf{Q}) \\ s.t. & \\ & \mathbf{Q} \in F \end{aligned}$$

where F is the set of all the feasible policies, and f represents the objective function to optimize.

The principle of optimality was originally stated by Bellman and Dreyfus [9] as follows: *A policy is said to be optimal whether for any initial state and decision, the rest of decisions are optimal with respect to the resultant state of the former decision.* This principle is closely related to the functional equation, which characterizes the set of solutions to the optimization problem. Thus, if f_j represents the cost/benefit of the optimal solution to the problem starting with state j, then the functional equation can be stated as follows:

$$f_i = \min_{j \in E} \{t_{ij} + f_j\}$$

These concepts can be extended to the multicriteria framework as Villarreal and Karwan [93] pointed out. Let M be the number of objectives in a multistage decision process where N stands for the number of stages. In addition, let $R_i^{j,k}$ be the return function for criterion i evaluated from stage j to stage k. In what follows, we admit that $R_i^{j,k}$ is obtained applying a $(k-j+1)$-dimensional vector Δ of binary operators. Specifically, we calculate this function as $R_i^{j,k} = \sum_{t=j}^{k} r(Q_t)$, for any i, j and k. Hence, $R_i^{j,k}$ is a stagewise separable function since it can be reconstructed by the iterative use of the corresponding vector of operators Δ. Thus, the serial muticriteria multistage problem would be formultated as follows:

$$
\begin{aligned}
& v - \min(\max) g\{R_1^{1,N}(\mathbf{Q}), R_2^{1,N}(\mathbf{Q}), \ldots, R_M^{1,N}(\mathbf{Q})\} \\
s.t. & \\
& \mathbf{Q} \in F \subseteq \mathbb{R}^N
\end{aligned}
$$

where $g\{R_1^{1,N}(\mathbf{Q}), R_2^{1,N}(\mathbf{Q}), \ldots, R_M^{1,N}(\mathbf{Q})\}$ denotes the multicriteria return function of the decision process. The solution to this problem would be a set of non-dominated policies $\mathcal{P}$.

Assuming that g_j denotes the set of cost/benefit vectors associated to non-dominated solutions of the problem starting with state j, the multicriteria problem above can be formulated using the following functional equation

$$g_i = v - \min_{j \in \overline{E} \subseteq \mathbb{R}^M}(\max) \{T_{ij} \oplus g_j\}$$

In this equation, the set of states $\overline{E}$ contains real-valued M-dimensional vectors and T_{ij} represents the M-dimensional cost/benefit vectors corresponding to all the possible policies that start at state i and reach state j.

Applying dynamic programming in both single and multicriteria frameworks entails the curse of dimensionality. That is, computer time and storage requirements become prohibitive for large problems.

1.5.3 Branch and Bound

This technique is usually applied to solve integer programming (IP) problems. Specifically, *Branch-and-Bound* (BB) methods find the optimal solution to an IP problem by efficiently enumerating the points in a subproblem´s feasible region. It is well-known that whether the solution to the linear programming (LP) relaxation of a pure IP is integer, then this solution is also the optimal solution to the IP. Accordingly, the BB method begins by solving the LP relaxation of the IP. If all the decision variables assume integer values in the optimal solution to the LP relaxation, then the optimal solution to the LP relaxation will be the optimal solution to the IP. On the contrary, the feasible region should be partitioned by arbitrarily choosing a fractional variable, say x_i, in the solution to the LP relaxation. With this in mind, we "branch" on this fractional variable to generate two new subproblems, namely, a subproblem considering the additional constraint that $x_i \leq \lfloor x_i \rfloor$ and the complementary subproblem assuming that $x_i \geq \lceil x_i \rceil$, where operators $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ indicate, respectively, the smaller and greater integer values closest to a given value.

We proceed to branch a subproblem whenever its LP relaxation provides fractional variables. This branching process finishes because either an infeasible integer solution is reached or the converse. In the latter case, the cost associated to this feasible solution represents an *upper bound* when minimizing or a *lower bound* when maximizing to the original IP. Hence, this upper or lower bound can be used to discard subproblems that do not yield better solutions.

1.5.4 Convex Set and Lower Envelope

Let U be a set of points in $\mathbb{R}^n$. We say that U is a *convex set* if for any pair of points $p, q \in U$ the line segment $\overline{pq}$ is completely contained in U. Moreover, the *convex hull* of a set U, $conv(U)$, is defined as the intersection of all convex sets that contain U. Algorithms to determine the convex hull of a set of points can be found, among others, in Preparata and Shamos [70], Berg et al. [10] and Hershberger and Suri [51].

A segment s in $\mathbb{R}^2$ is defined as a function of x whose domain is an open interval $(left(s), right(s))$. If s is a line segment, then this function $s(x)$ is linear. Moreover, we denote by $\mathcal{LE}(U)$ the *lower envelope* of a set U of n segments in $\mathbb{R}^2$, which is a function of x whose domain is the whole real line: $\min(\infty, \{s(x)|s \in U$ and $left(s) < x < right(x)\})$.

An efficient algorithm to calculate the lower envelope of a set of segments is given in Hersberger [50].

1.5.5 Computational Complexity

A relevant aspect in this work concerns the evaluation of the algorithms proposed in the subsequent chapters. The efficiency of an algorithm is measured in terms of quantity of resources that the algorithm needs. That is, as a function of the input size of the instance under consideration. In particular, we are interested in estimating the asymptotic behavior of the *running time* of the algorithm. Formally, let $\mathbb{R}^+$ denote the set of real numbers greater than 0 and let f and g be two functions $f, g : \mathbb{N} \to \mathbb{R}^+$. We say that $f(n) = \mathrm{O}(g(n))$ (f is asymptotically at most g) if there exist positive integers c and n_0 so that for every integer $n \geq n_0$

$$f(n) \leq cg(n)$$

On the contrary, we say that $f(n) = \Omega(g(n))$ (f is asymptotically at least g) if there exist positive integers c and n_0 so that for every integer $n \geq n_0$

$$f(n) \geq cg(n)$$

Finally, we say that $f(n) = \Theta(g(n))$ (f is asymptotically equal to g) if there exist positive integers c, d and n_0 so that for every integer $n \geq n_0$

$$dg(n) \leq f(n) \leq cg(n)$$

For a complete survey on computational complexity, we refer to the books of Garey and Johnson [40], Papadimitriou [68], and Johnson and Papadimitriou [55], among others.

1.5.6 Network Flows

For the sake of simplicity, in Chapters 2 and 3, we will reformulate the original IP problems as *network flow* problems. Accordingly, a *directed graph* $G = (N, A)$ consists of a set N of n nodes and a set A of m arcs whose elements are ordered pairs of distinct nodes. A *directed network* is a directed graph whose nodes or arcs have associated numerical values (typically, costs, capacities, and/or supplies and demands). In Figure 1.7, it is shown an illustration of a simple directed network with $N = \{0, 1, 2, \ldots, T, T+1\}$ and $A = \{(0,1), (0,2), \ldots, (0,T), (1,T+1), (2,T+1), \ldots, (T,T+1)\}$. In general, it is assumed that a flow of x_{ij} units is sent through the arc (i,j), $i = 1,...,T$ and $j = 1,...,T$. Moreover, flows in the network are limited by lower (l_{ij}) and upper (u_{ij}) bounds, namely, $l_{ij} \leq x_{ij} \leq u_{ij}$, $i = 1,...,T$ and $j = 1,...,T$.

In particular, an appropiate assignment of flows in Figure 1.7 to parameters and variables of the Dynamic Lot-sizing problem allows us to reformulate it as a *Minimum Cost Flow* (MCF) problem (Zangwill [105]). Let c_{ij} be the cost of transferring one unit of flow through arc (i,j), and let b_i denote the net supply (outflow-inflow) at node i. Accordingly, the MCF problem can be formally stated as follows

$$\begin{aligned} & \min \sum_{(i,j)\in A} c_{ij} x_{ij} \\ s.t. & \\ & \sum_j x_{ij} - \sum_k x_{ki} = b_i \quad \text{for all } i \in N \\ & l_{ij} \leq x_{ij} \leq u_{ij} \qquad\quad \text{for all } (i,j) \in A \end{aligned}$$

Further information about network flows can be found in Ahuja et al. [2], Dolan and Aldous [25] and Bazaraa et al. [8], among others.

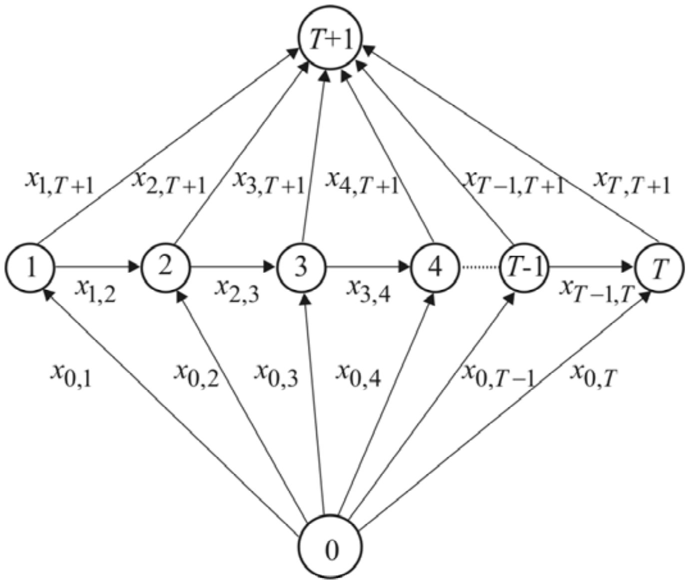

Figure 1.7: Illustration of a directed network.

1.5.7 Multiscenario Analysis

In many real-world applications, it is often needed to consider other information different to that appearing in historic files, e.g. new markets, new products, etc. This lack of information concerns the uncertainty. In such situations, it is appropriate to employ the *scenario analysis* approach where the uncertainty of the parameters of the system is modeled by several replications (subproblems) of the original problem. These subproblems correspond to different scenarios, and the goal of this approach can be either to determine the set of Pareto solutions or to find a *robust* policy among the feasible solutions for all scenarios such that satisfies a particular measure. Further information about scenario analysis can be found in Rockafellar and Wets [75], Wets [99], Escudero and Kamesan [31] [32], Dempster [22] and Birge and Louveaux [11], among others.

We address, in Chapter 3, the Dynamic Lot-sizing problem but admitting uncertainty in the values of parameters. We show that classical properties of the dynamic lot-sizing problem can be rewritten to generate efficient policies. These plans can be helpful to the decision maker in order to determine good policies to be implemented.

Chapter 2

The Dynamic EOQ with Limited Inventory

2.1 Introduction

Dynamic lot size problems describe a relevant class of production/inventory systems which are often met in practice. The goal consists of finding the production/order plan satisfying the demands over a given number of periods at minimum cost. When the lot sizes to be produced are restricted by bounds, the problem is called *capacitated*. On the other hand, when the inventory levels are bounded variables, this problem is usually known as *bounded/limited inventory model*. Although these two latter problems are mathematically related, the principles which characterize the optimal plans in both cases are different.

This chapter is devoted to the dynamic lot-size problem in which the order quantities are restricted by the warehouse capacity. In this type of problems, the planning horizon is divided into T periods. We assume that the demand for each period is known and shortages are not permitted. Moreover, in a given period t, the production/order plus the available inventory at the end of the preceding period must not exceed the storage capacity W_t, $t = 1, \ldots, T$.

The uncapacitated version of this model was introduced by Wagner and Whitin [97] and, independently, by Manne [61]. These authors proposed an $O(T^2)$ algorithm based on the Zero Inventory Order (ZIO) policies to reduce the state space. Later, Veinott [92] approached the model as a minimum cost flow problem considering convex costs. Zangwill [104] proposed a polynomial algorithm to solve the production scheduling problem when backorders are allowed. An excellent survey on capacitated and uncapacitated versions of this model is presented in Wolsey [102]. The algorithm of Wagner and Whitin considering variable production/reorder costs can be improved to run in O(Tlog$T$) as Federgruen and Tzur [33], Wagelmans et al. [95] and Aggarwal and Park [1] shown. Besides, these authors proved that an O(T) algorithm can be devised when costs are of the form like in Wagner and Whitin [97], that is, assuming that production/reorder costs are constant.

The capacitated version of the dynamic lot-size problem has been studied in detail by many authors. These authors have considered distinct assumptions on the cost functions and the boundaries of the production quantities. For instance, Florian and Klein [36] devised a dynamic programming shortest path algorithm for the case of constant capacity and concave costs. Extensions to more general production cost functions were proposed by Jagannathan and Rao [54]. Swoveland [86] considered piecewise concave production and holding-backorder costs. Louveaux [59] presented a formulation, based on Swoveland's results, which assumes concave production costs and holding-backorders costs that are piecewise linear and convex. Later, Baker et al. [7] exploited properties satisfied by the optimal plans in the case of time-varying production capacity constraints, devising an O(2^T) algorithm. Karmarkar et al. [56] extended the model to consider startup costs incurred for switching on the production facility and a separate reservation cost charged for keeping the facility on, whether or not it is used for production. They proposed a dynamic programming algorithm for the uncapacitated case, and a branch and bound approach using Lagrangian relaxation for the capacitated problem. Pseudopolynomial algorithms can be obtained by using dynamic programming as discussed in Florian, Lenstra and Rinnooy Kan [37]. The reader is referred to Bitran and Yanasse [12] and Bitran and Matsuo [13] for a comprehensive discussion of the computational complexity of the capacitated lot-size problem. Chung and Lin [19] developed an O(T^2) algorithm that solves the problem under very specific assumptions of the holding and production costs and the production capacity.

In contrast, the bounded inventory model can be found in few references in the literature. Precisely, Love [60] studied this problem providing an O(T^3) algorithm based on the Dynamic Programming approach. In this chapter, we introduce new properties to determine optimal plans of the dynamic lot-size problem with limited inventory considering general concave costs in Section 2.2, linear costs without setup costs in Section 2.3 and, finally, in Section 2.4 admitting linear and setup costs. Thus, results in Section 2.2 are based on the dynamic programming paradigm. Moreover, the computational results indicate that this new algorithm is almost thirty times faster than Love's procedure. Besides, it can be proved that the algorithm runs in linear expected time when each demand value d_t takes values in the interval $[0, W_t]$. In addition, we also study, in Section 2.4, the model admitting a particular cost structure, namely, linear costs with setup. In particular, we face the problem extending the geometrical technique proposed by Wagelmans et al. [95]. This technique leads to develop an O($T\log T$) algorithm. Finally, Section 2.3 is devoted to the dynamic lot-sizing problem with limited inventory but considering linear costs without setup costs. In this case, the solution method consists of formulating the problem as a network flow problem. Thus, sorting the costs and determining the residual capacities of the network allow to devise an O($T\log T$) algorithm.

2.2 The General Concave Costs Case

In this section we discuss the deterministic single item dynamic lot-size problem with limited inventory (DLSPLI) considering concave reorder and holding costs and nonnegative inventory levels. We introduce an O(T^3) algorithm for this problem which was first solved by Love [60]. Both procedures run in the same worst-case complexity. However, as we will show in a later computational experience, this algorithm is almost thirty times faster than Love's procedure. Moreover, when demands vary in the interval $[0, \frac{2W_t}{k}]$, with $k, t = 1 \ldots, T$, the new algorithm runs in O(Tk^2) expected time for $k \leq \frac{T}{2}$ and it runs in O($kT^2 + k^3$) expected time otherwise. Note that for values of k adequately small with respect to T, this algorithm runs in O(T) expected time. This assertion is proved in an additional computational experiment.

2.2.1 Formulation and Notation

According to the material balance equation, the inventory level I_t is given by

$$I_t = I_0 + \sum_{j=1}^{t} (Q_j - d_j)$$

Thus, the Dynamic Lot Size Problem with Limited Inventory (DLSPLI), or P for short, can be formulated as follows:

$$
\begin{array}{lll}
(P): & \min \sum_{t=1}^{T} \left(C_t(Q_t) + H_t(I_t)\right) & \\
s.t. & & \\
& I_0 = I_T = 0 & \\
& I_{t-1} + Q_t - I_t = d_t & t = 1, \ldots, T \\
& I_{t-1} + Q_t \leq W_t & t = 1, \ldots, T \\
& Q_t, I_t \in \mathbb{N}_0 & t = 1, \ldots, T
\end{array}
\tag{2.1}
$$

As the first constraint in (2.1) suggests, we assume that both the inventory level at the beginning of the first period and the inventory level at the end of the last period are zero. The second group of constraints represents the well-known material balance equations which determine the inventory levels from the previous decisions. The next set of constraints indicates that the sum of the inventory level and the reorder quantity in period t must be smaller than or equal to the storage capacity in items units, and it avoids the reorder quantity exceeding the free storage capacity. The fourth group of constraints in (2.1) forces the reorder quantities and the inventory levels to be nonnegative integers.

Note that the second and third set of constraints in (2.1) can be used to obtain the following new constraints: $d_t + I_t \leq W_t$, that is, $I_t \leq W_t - d_t, t = 1, \ldots, T$. Therefore, the problem P can be reformulated as follows:

$$
\begin{array}{lll}
(P): & \min \sum_{t=1}^{T} \left(C_t(Q_t) + H_t(I_t)\right) & \\
s.t. & & \\
& I_0 = I_T = 0 & \\
& I_{t-1} + Q_t - I_t = d_t & t = 1, \ldots, T \\
& 0 \leq I_t \leq W_t - d_t & t = 1, \ldots, T-1 \\
& Q_t \in \mathbb{N}_0 & t = 1, \ldots, T
\end{array}
$$

Since the inventory levels must be nonnegative and considering the third constraint, the problem reaches a feasible solution whenever $d_t \leq W_t, t = 1, \ldots, T$. Should $d_t > W_t$ for some period t, then the warehouse must contain d_t units of item at the beginning of period t but this would be impossible since the inventory level would exceed the capacity W_t.

Now, we proceed to introduce a functional equation to find an optimal reorder plan by using dynamic programming. Notice that the cost of operating the system during periods t through T depends on the inventory I_{t-1} at the beginning of period t, but neither on prior inventories nor on prior reorder quantities. So, the pair (t, I_{t-1}) constitutes a state. Let $f(t, I_{t-1})$ be the minimum cost of satisfying demand during periods t through T if the inventory is I_{t-1} at the beginning of period t.

The quantity $f(t, I_{t-1})$ is intended to include the holding cost $H_t(I_t)$ that is incurred immediately as well as all future reorder and holding costs. The cost of an optimal inventory plan is $f(1, 0)$.

As we mentioned before, the final inventory must be zero. Therefore, if the inventory level I_{T-1} is below d_T, then the difference must be produced. Inventory levels in excess of d_T are forbidden. Hence,

$$f(T, I_{T-1}) = H_T(I_T) + C_T(d_T - I_{T-1}) \text{ for } I_{T-1} = 0, 1, \ldots, d_T$$

Consider a state (t, I_{t-1}) with $t < T$, and let Q_t denote the quantity ordered in period t. The sum of the inventory at the beginning of period t and the ordered quantity during period t, that is, $I_{t-1} + Q_t$, must be at least as large as the demand d_t during that period. Also, the sum $I_{t-1} + Q_t$ can not exceed the minimum between the warehouse capacity and the total demand during all remaining periods. Thus, the order level Q_t is called feasible for state (t, I_{t-1}) if Q_t is a nonnegative integer satisfying

$$d_t \leq I_{t-1} + Q_t \leq \min\{W_t, d_t + \cdots + d_T\} \tag{2.2}$$

For instance, note that Q_1 can take any integer value in the set $\{d_1, d_1 + 1, d_1 + 2, \ldots, \min\{W_t, d_1 + \cdots + d_T\}\}$. Therefore, the number of possible decisions per period can be enormous.

The optimality principle gives rise to the functional equation

$$f(t, I_{t-1}) = \min_{Q_t \text{ feasible}} \{H_t(I_t) + C_t(Q_t) + f(t + 1, I_{t-1} + Q_t - d_t)\}, t < T \tag{2.3}$$

Taking into account (2.2) and (2.3), a dynamic programming algorithm can be devised. Since the decision states set increases drastically with W_t and/or the sum of demands, searching optimal solutions for this problem becomes an arduous task.

To overcome the above difficulty, in the following sections we reduce significantly the decision states set introducing properties that identify optimal policies.

2.2.2 Love's Approach

As we mentioned before, Love [60] devised an algorithm to identify optimal solutions even when shortages are allowed. He defined an *inventory point* as a period in which the inventory on hand equals its lower bound, zero or its upper bound. Based on this definition, Love [60] proposed a property which states that between any two adjacent inventory points there can be at most one period of nonzero production. Since in problem P shortages are not permitted, the inventories on hand can be equal to either zero or their upper bound. Let I_i^u and I_k^v be the inventories on hand at the end of periods i and k respectively, where u and v represent the two feasible values of these inventories. In other words, u and v represent boolean variables such that $u = 0$ indicates that the inventory on hand at the end of period i is zero, otherwise this inventory is at its upper bound. The same argument is applied to superscript v, which is related to period k. Thus, the values for the inventories on hand of two adjacent inventory points, say i and k, when stockouts are not allowed, along with the quantities to order in each case are shown in Table 2.1

Let f_i^u be the minimum cost in periods $i+1, \ldots, T$, having an inventory level at the end of period i equals I_i^u. Then, setting $f_T^0 = 0$ and $f_T^1 = \infty$, we have

$$f_i^u = \min_{i<j\leq k\leq T; v=0,1} [c_{ijk}^{uv} + f_k^v], \quad i = 0, \ldots, T-1; u = 0, 1$$

where

$$c_{ijk}^{uv} = C_j(Q_j) + \sum_{t=i+1}^{j-1} H_t(I_i^u - \sum_{l=i+1}^{t} d_l) + \sum_{t=j}^{k-1} H_t(I_i^u + Q_j - \sum_{l=i+1}^{t} d_t) + H_k(I_k^v) \quad (2.4)$$

u	v	I_i^u	I_k^v	Quantity to order (Q_j)	j
0	0	0	0	$d_{i+1,k+1}$	$i+1$
0	1	0	$W_k - d_k$	$d_{i+1,k} + W_k$	$i+1$
1	0	$W_i - d_i$	0	$d_{i,k+1} - W_i$	$i < j \leq k$
1	1	$W_i - d_i$	$W_k - d_k$	$d_{ik} + W_k - W_i$	$i < j < k$

Table 2.1: Values for u,v,I_i^u, I_k^v, Q_j and j

where $W_i - d_i$ and $W_k - d_k$ represent the upper bounds for the inventories in periods i and k respectively, according to the formulation in (2.1). For the sake of simplicity, in what follows , we denote by $d_{ik} = \sum_{t=i}^{k-1} d_t$. Remark that, once the accumulated demands $d_{1T}, d_{2T}, \ldots, d_{TT}$ are obtained in O(T), any value d_{ik} is determined in O(1) applying $d_{ik} = d_{iT} - d_{k+1,T}$.

From this point, we will show that this result can be improved using a new approach to identify the extreme plans for the DLSPSC.

2.2.3 Characterizing Optimal Plans

Let us introduce new results which allow us to characterize the optimal plans for problem P. The following theorem identifies reorder quantities for each period, reducing the number of possible decisions to be considered in the dynamic programming approach.

The result below states that there always exists an optimal policy where the reorder quantity for each period must be equal to zero, or to the sum of demands minus the inventory level at the end of the previous period, or to the total capacity minus the inventory level at the end of the previous period.

Theorem 1 *Among the optimal plans for P, there exists, at least, one solution $\mathbf{Q} = (Q_1, \ldots, Q_T)$ such that for each period i, Q_i satisfies:*

$$Q_i = \begin{cases} 0 \\ d_{it} - I_{i-1},\ i < t \leq T+1 \\ W_i - I_{i-1} \end{cases} ,\ i = 1, \ldots, T \tag{2.5}$$

In other words, for $1 \leq i < t \leq T+1$, the following expression holds

$$(Q_i + I_{i-1} - W_i)(Q_i + I_{i-1} - d_{it})Q_i = 0$$

Proof. Let $\mathbf{Q}$ be an optimal plan that does not fullfil (2.5). Then, there exists a period i such that

$$Q_i \neq 0, I_{i-1} + Q_i \neq W_i \text{ and } I_{i-1} + Q_i \neq d_{it}$$

Since $\mathbf{Q}$ is feasible, the following inequalities must hold

$$I_{i-1} + Q_i \leq W_i \text{ and } I_{i-1} + Q_i \geq d_{it} \tag{2.6}$$

The conditions in (2.6) are depicted in Figure 2.1. Plan $\mathbf{Q}$ is characterized by the sequence $[A, B, C, D, E, F]$ in that figure.

Let q_i be the minimum between $\{I_{t-1}, W_i - (Q_i + I_{i-1})\}$. Then, following a reasoning similar to that in Denardo [23], we can generate two new plans $\mathbf{Y}$ and $\mathbf{Z}$ from $\mathbf{Q}$ such that:

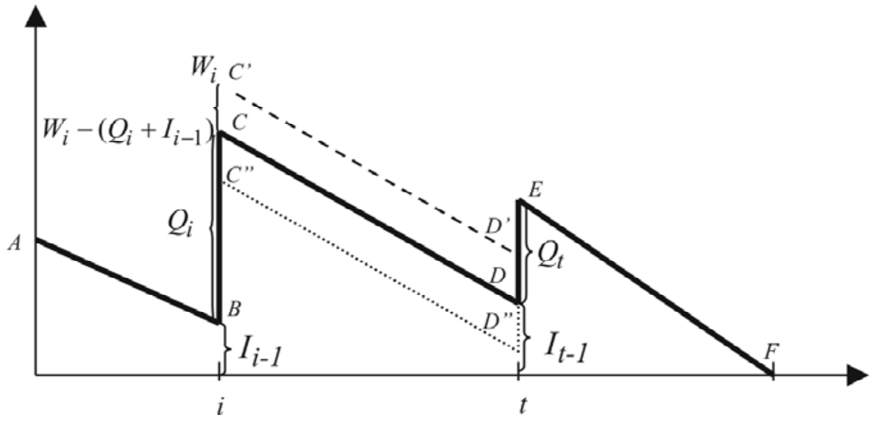

Figure 2.1: Illustration of Theorem 1.

$$\begin{array}{llll} \mathbf{Y} = \mathbf{Q} & \text{except for period} & i: & Y_i = Q_i + q_i \\ & \text{and for period} & t: & Y_t = Q_t - q_i \\ \mathbf{Z} = \mathbf{Q} & \text{except for period} & i: & Z_i = Q_i - q_i \\ & \text{and for period} & t: & Z_t = Q_t + q_i \end{array} \tag{2.7}$$

As one can see, plans $\mathbf{Y}$ and $\mathbf{Z}$ are represented by sequences $[A, B, C', D', E, F]$ and $[A, B, C'', D'', E, F]$ respectively in Figure 2.1.

These perturbed plans cannot decrease the cost below the optimum. Thus

$$C_i(Q_i) + C_t(Q_t) + \sum_{k=i}^{t-1} H_k(I_k) \leq C_i(Q_i + q_i) + C_t(Q_t - q_i) + \sum_{k=i}^{t-1} H_k(I_k + q_i) \tag{2.8}$$

and

$$C_i(Q_i) + C_t(Q_t) + \sum_{k=i}^{t-1} H_k(I_k) \leq C_i(Q_i - q_i) + C_t(Q_t + q_i) + \sum_{k=i}^{t-1} H_k(I_k - q_i) \tag{2.9}$$

Adding (2.8) to (2.9), and rearranging the sum we obtain

$$[C_i(Q_i + q_i) + C_i(Q_i - q_i) - 2C_i(Q_i)] + [C_t(Q_t - q_i) + C_t(Q_t + q_i) - 2C_t(Q_t)]$$

$$+[\sum_{k=i}^{t-1} H_k(I_k+q_i)+\sum_{k=i}^{t-1} H_k(I_k-q_i)-2\sum_{k=i}^{t-1} H_k(I_k)]\geq 0 \tag{2.10}$$

Since the cost functions are concave, each term in brackets in (2.10) corresponds to the sum of nonpositive quantities. Then, inequalities (2.8) and (2.9), of which (2.10) is the sum, must hold as equations. Hence, if $q_i = I_{t-1}$, plan **Z** is also optimal and it holds that $I_{i-1} + Z_i = d_{it}$, verifying (2.5). Otherwise, when $q_i = W_i - (Q_i + I_{i-1})$, plan **Y** is optimal and it holds that $I_{i-1} + Y_i = W_i$, verifying (2.5). ■

Colloquially, the following theorem states that if the final inventory level for a given period is equal to the sum of demands for any latter period, then the optimal decision in the following period is not to order.

Theorem 2 *There always exists an optimal plan* $\mathbf{Q} = (Q_1, \ldots, Q_T)$ *for problem* P *such that if* I_{j-1} $(j = 2, \ldots, T)$ *corresponds to the sum of demands of periods* j *to* t *for some* t, $j \leq t \leq T$, *then* Q_j *is zero.*

Proof. Assume that there exists an optimal plan **Q** such that $I_{j-1} = d_{jt}$ and $Q_j \neq 0$. Since I_{j-1} corresponds to the sum of demands and in view of Theorem 1, the only two possible decisions are: $Q_j = 0$ or $Q_j = W_j - I_{j-1}$. Moreover, note that if $I_{j-1} = d_{jt}$ this is because in a previous period i: $I_{i-1} + Q_i = d_{it}$. Consider $Q_j = W_j - I_{j-1}$, then following a similar argument to that in the proof of Theorem 1, let q_i be the minimum between $\{I_{j-1}, W_i - (I_{i-1} + Q_i)\}$. Again, two new plans **Y** and **Z** can be generated perturbing plan **Q** in the same manner as in (2.7). If $q_i = I_{j-1}$, plan **Z** is also optimal being the new I_{j-1} equal to zero. Thus, this new plan **Z** is collected within the set of plans generated by (2.5). Otherwise, when $q_i = W_i - (I_{i-1} + Q_i)$, plan **Y** is optimal being the new $I_{j-1} = W_i - d_{ij}$, and this new plan is also generated by (2.5). Therefore, the decision of ordering in period j when $I_{j-1} = d_{jt}$ need not be analyzed because this decision has the same cost as one of the plans **Y** or **Z** obtained considering $I_{j-1} = W_i - d_{ij}$ or $I_{j-1} = 0$. That is, it is necessary to evaluate only the decision $Q_j = 0$. ■

Using the above results, the number of decisions per period is reduced with respect to the set of decisions generated by Theorem 1. Accordingly, we devise an efficient dynamic programming algorithm to calculate an optimal plan for problem P.

In the following section we explain in detail the dynamic programming algorithm, and we calculate its theoretical complexity as well.

2.2.4 Algorithm

We introduce the notation required to implement the algorithm. Let M_{ijl} be the set of feasible states obtained aplying Theorem 1, which contains both the sum of demands from period $i+1$ to a period j and the states corresponding to capacities minus the sum of demands from period i down to a period l. That is,

$$M_{ijl} = \{0, d_{i+1}, d_{i+1} + d_{i+2}, \dots, d_{i+1,j+1}, W_i - d_i, W_{i-1} - d_i - d_{i-1}, \dots, W_l - d_{l,i+1}\}$$

It is clear that only those values $d_{i+1,j+1} \leq W_i$ and $W_l - d_{l,i+1} \geq 0$ are allowed. Moreover, given a period i with $i = 0, \dots, T$, let L_i denote the set of feasible states at the end of period i, which can be obtained as follows

$$L_i = \bigcup_{1 \leq l \leq i < j \leq T} M_{ijl}$$

Each state in L_i is characterized by three components: the inventory level I, the minimum cost among all paths that reach this state ($Cost[i, I]$) and the reorder quantity related to this cost ($X[i, I]$). From the formulation of the problem, we know that $L_0 = L_T = \{0\}$. In the worst case, a set L_i contains $T+1$ states at most. Moreover, let $Q_i(I)$ be the set of feasible reorder quantities in the ith period when the inventory level at the beginning of that period is $I \in L_{i-1}$. This means that

$$\begin{aligned} Q_i(I) = \quad & \{Q_i \in \mathbb{N} : I + Q_i = d_{i,j+1},\ i \leq j \leq T \text{ with } I + Q_i < W_i\} \\ & \cup \{Q_i \in \mathbb{N} : I + Q_i = W_i\} \\ & \cup \{0\}. \end{aligned}$$

By virtue of Theorem 2, note that if $I \in L_{i-1}$ corresponds to the sum of demands then $Q_i(I) = \{0\}$. Taking the above reasoning into account, we now outline the algorithm proposed.

In the following section we introduce the theoretical complexity of Algorithm 1 using the worst-case and the average-case analysis.

2.2.5 Complexity of the Algorithm

The following notation is needed to show the complexity of Algorithm 1. For a period i, let A_i denote the number of states with inventory level corresponding to the sum of demands and let B_i be the number of states with inventory level corresponding to W_i minus the sum of demands. Note that the cardinal of L_i equals $A_i + B_i + 1$, where 1 stands for the zero inventory level. Observe that A_i and B_i are at most $T - i$ and i, respectively. Therefore, the maximum number of states in L_i is $T + 1$.

Algorithm 1 Determine an optimal plan $\mathbf{Q} = (Q_1, \ldots, Q_T)$ for problem P

Data: vectors d and W, functions $C_i()$ and $H_i()$, and the number of periods T

Initialize L_i, $i = 0, \ldots, T$
for $i \leftarrow 1$ to T **do**
  for all $I \in L_{i-1}$ **do**
    for all $Q \in Q_i(I)$ **do**
      $J \leftarrow I + Q - d_i$
      if $Cost[i-1, I] + C_i(Q) + H_i(J) < Cost[i, J]$ **then**
        $Cost[i, J] \leftarrow Cost[i-1, I] + C_i(Q) + H_i(J)$
        $X[i, J] = Q$
      end if
    end for
  end for
end for
return$(Cost[T, I_T])$

Proposition 3 *Algorithm 1 runs in $O(T^3)$ time.*

Proof. For a period i with $i = 1, \ldots, T-1$, the number of decisions when $I_{i-1} = 0$ is $A_i + 2$. For each I_{i-1} corresponding to the sum of demands only one decision should be made. In this case, the total number of decisions is A_{i-1}. On the other hand, for each I_{i-1} corresponding to W_i minus the sum of demands, $A_i + 3$ decisions are made. Hence, the total number of decisions for this type of inventory level is $\sum_{j=1}^{B_{i-1}} A_i + 3$. Consequently, the overall number of decisions carried out by the algorithm is

$$\sum_{i=1}^{T-1} (A_i + 2 + A_{i-1} + \sum_{j=1}^{B_{i-1}} A_i + 3) + (A_1 + 2) + (A_{T-1} + B_{T-1} + 1)$$

The above summation leads to the following maximum number of decisions

$$\frac{5}{6}T^3 - T^2 + \frac{13}{16}T + 2$$

Hence, Algorithm 1 runs in $O(T^3)$. ∎

Now, we proceed to analyze the complexity of the algorithm considering the average-case. Throughout, we denote by $E(T)$ the running time of a randomized algorithm which runs in $O(T)$ expected time. For the sake of simplicity, we can admit, without loss of generality that $W_t = W$, $t = 1, \ldots, T$. Let us assume that the demand for a period t ranges in $[0, \frac{2W}{k}]$, $k = 1, \ldots, T$. The value $\frac{2W}{k}$ has been chosen to facilitate the proof of the following proposition, which states the average-case complexity of Algorithm 1.

Proposition 4 *Algorithm 1 runs in $E(Tk^2)$ for $k \leq \frac{T}{2}$, and runs in $E(kT^2 + k^3)$ when $k \geq \frac{T}{2}$.*

Proof. We denote by $\overline{v}$ the expected value of the random variable v. Thus, to determine the average-case complexity, the maximum values of $\overline{A_i}$ and $\overline{B_i}$ are calculated, where A_i and B_i have been previously defined:

$\max \overline{A_i} = \max\limits_{t=i+1,\ldots,T} (t - i : \sum_{j=i+1}^{t} \overline{d_j} \leq W) = \max\limits_{t=i+1,\ldots,T} (t - i : (t-i)\frac{2W}{2k} \leq W) = \max\limits_{t=i+1,\ldots,T} (t - i : (t - i) \leq k) = k$

Since $T - i$ is the maximum value of A_i, then $\max \overline{A_i}$ =min$(T - i, k)$. In addition, as $1 \leq (t - i) \leq T$, only values for k ranging in $[1, T]$ are significant.

$\max \overline{B_i} = \max\limits_{t=1,\ldots,i} (i - t + 1 : W - \sum_{j=t}^{i} \overline{d_j} \geq 0) = \max\limits_{t=1,\ldots,i} (i - t + 1 : W - (i - t + 1)\frac{2W}{2k} \geq 0) = \max\limits_{t=1,\ldots,i} (i - t + 1 : (i - t + 1) \leq k) = k$

Similarly, i represents the maximum value of B_i, then $\max \overline{B_i}$ =min(i, k). Again, as $1 \leq i \leq T$, only values for k ranging in $[1, T]$ are significant.

Observe that for those values of k greater than T the algorithm runs in $O(T^3)$ because $\overline{A_i}$ and $\overline{B_i}$ are at most $T - i$ and i, respectively. Under this assumption, Proposition 3 follows.

For a period i with $i = 1, \ldots, T - 1$, the maximum expected number of decisions when $I_{i-1} = 0$ is $\max \overline{A_i} + 2$. For each I_{i-1} corresponding to the sum of demands only one decision should be made. In this case, the maximum expected number of decisions is $\max \overline{A_{i-1}}$. On the other hand, for each I_{i-1} corresponding to W minus the sum of demands, $\max \overline{A_i} + 3$ decisions, at most, are expected to be made. Hence, the maximum expected number of decisions for this type of inventory level is $\sum_{j=1}^{\max \overline{B_{i-1}}} \max \overline{A_i} + 3$. Consequently, the overall expected number of decisions carried out by the algorithm is

$$\sum_{i=2}^{T-1} (\max \overline{A_i} + 2 + \max \overline{A_{i-1}} + \sum_{j=1}^{\max \overline{B_{i-1}}} \max \overline{A_i} + 3) +$$
$$(\max \overline{A_1} + 2) + (\max \overline{A_{T-1}} + \max \overline{B_{T-1}} + 1)$$

Observe that when $i \leq T-k$ then max $\overline{A_i} = k$, otherwise, max $\overline{A_i} = T-i$. Therefore, the above summation can be separated into two summations:

$$\sum_{i=2}^{T-k} (k+2+k+ \sum_{j=1}^{\max \overline{B_{i-1}}} k+3)+ \sum_{i=T-k+1}^{T-1} (T-i+2+T-i+1+ \sum_{j=1}^{\max \overline{B_{i-1}}} (T-i)+3)$$

Taking into account that max $\overline{B_{i-1}}$ =min$(k, i-1)$, the above expression results in

$$\sum_{i=2}^{T-k} (2k+5+k\min(k,i-1))+ \sum_{i=T-k+1}^{T-1} (2(T-i)+6+(T-i)\min(k,i-1))$$

Therefore, the overall expected number of decisions can be formulated as

$$\sum_{i=2}^{T-k} (2k+5+k\min(k,i-1))+ \sum_{i=T-k+1}^{T-1} (2(T-i)+6+(T-i)\min(k,i-1))+2(k+2) \tag{2.11}$$

To simplify (2.11), we can distinguish two cases: when $k \leq T-k$ or when $k \geq T-k$. In the first case, as i ranges in $[0, T-k]$ in the first summation, if $i \leq k$ then $\min(k, i-1) = (i-1)$. On the other hand, when $k < i \leq T-1$, then $\min(k, i-1) = k$. Accordingly, (2.11) can be written as follows

$$\sum_{i=2}^{k} (2k+5+k(i-1))+ \sum_{i=k+1}^{T-k} (2k+5+k^2)+ \sum_{i=T-k+1}^{T-1} (2(T-i)+6+(T-i)k)+2(k+2)$$

Or, in other words

$$Tk^2 + 2Tk + 5T - k^3 - 2k^2 + 2k - 2$$

Thus, when $k \leq T-k$, that is $k \leq \frac{T}{2}$, Algorithm 1 runs in $E(Tk^2)$.

Considering that $k \geq T-k$, (2.11) can be expressed as

$$\sum_{i=2}^{T-k} (2k+5+k(i-1))+ \sum_{i=T-k+1}^{k} (2(T-i)+6+(T-i)(i-1))$$
$$+ \sum_{i=k+1}^{T-1} (2(T-i)+6+(T-i)k)+2(k+2)$$

That is,

$$kT^2 - k^2T - \frac{T^3}{6} + \frac{T^2}{2} + \frac{14}{3}T + \frac{k^3}{3} + \frac{8}{3}k - 2$$

Therefore, when $k \geq \frac{T}{2}$, Algorithm 1 runs in $E(kT^2 + k^3)$. ■

For values of k adequately small respect to T, Algorithm 1 runs in $E(T)$. If $k \equiv O(\sqrt{\log T})$, then Algorithm 1 runs in $E(T\log T)$; for k varying between $\Omega(\sqrt{\log T})$ and $O(\sqrt{T})$ then the algorithm runs in $E(T^2)$. Otherwise, Algorithm 1 runs in $E(T^3)$.

In particular, when $k = 2$, the demands vary in the interval $[0, W]$ and Algorithm 1 runs in $E(T)$.

2.2.6 Computational results

In this section, we compare the running times of Algorithm 1 with those obtained from the adaptation of Love's algorithm [60] to the case without shortages. We have chosen to compare this method with Algorithm 1 because, to the best of our knowledge, it is the unique procedure which solves problem P. Both, Algorithm 1 and Love's procedure have been coded in C++ using LEDA libraries [58] and were tested in a HP-712/80 workstation.

In the computational experiments the values for T (number of periods) and W (maximum number of items to be stored, i.e., $W_t \in (0, W]$ for $t = 1, \ldots, T$) have been chosen as follows: $T = 25, 50, 75, 100, 150, 200$ and 500 periods and $W = 100, 500$ and 1000. Assuming integer input data, demands have been chosen to be smaller than the capacity for any period. Therefore, the feasibility has been assured. In Table 2.2 we present a comparison of the average running times of Algorithm 1 considering that $k = \frac{T}{2}$, that is, when demands range in $[0, \frac{4W}{T}]$, and Love's procedure. For this computational experiment, we consider linear reorder and holding costs varying randomly in the interval $[0, 100]$. For each combination (T, W) ten replications have been considered.

The first and second columns in Table 2.2 represent the number of periods and the maximum capacity, respectively. The third column gives the average running times of Algorithm 1. The fourth column in Table 2.2 shows the average running times for Love's algorithm. In the last column, the ratio between both running times is presented. As we pointed out at the beginning of this paper, our procedure is almost thirty times faster than that proposed by Love.

Table 2.3 gives the average running times of Algorithm 1 when demands d_t range in $[0, W_t]$. In this case, the values for T and W have been chosen as follows:

	Capacity(W)	*Algorithm 1 average time (sec.)*	*Love's algorithm average time (sec.)*	*Ratio*
$T = 25$				
	100	0.008	0.184	23.00
	500	0.009	0.189	21.00
	1000	0.008	0.190	23.75
$T = 50$				
	100	0.054	1.309	24.24
	500	0.054	1.273	23.57
	1000	0.050	1.268	25.36
$T = 75$				
	100	0.174	3.994	22.95
	500	0.154	3.933	25.53
	1000	0.151	3.907	25.87
$T = 100$				
	100	0.347	8.923	25.71
	500	0.339	8.857	26.12
	1000	0.347	8.923	25.71
$T = 150$				
	100	1.246	28.711	23.04
	500	1.123	28.471	25.35
	1000	1.103	28.496	25.83
$T = 200$				
	100	3.056	66.136	21.64
	500	2.620	66.057	25.21
	1000	2.563	66.524	25.95
$T = 500$				
	100	36.548	999.887	27.35
	500	36.583	999.547	27.32
	1000	36.196	998.760	27.59

Table 2.2: Comparison between Algorithm 1 and Love´s algorithm

$T = 25, 50, 75, 100, 150, 250, 500, 750, 1000$ and 1500 periods and $W = 100, 500$ and 1000. Again, for each combination (T, W) ten replications have been considered.

	Capacity(W)	*Algorithm 1 average time (sec.)*		*Capacity(W)*	*Algorithm 1 average time (sec.)*
$T = 25$			$T = 250$		
	100	0.001		100	0.012
	500	0.001		500	0.010
	1000	0.001		1000	0.010
$T = 50$			$T = 500$		
	100	0.003		100	0.026
	500	0.002		500	0.027
	1000	0.001		1000	0.029
$T = 75$			$T = 750$		
	100	0.003		100	0.041
	500	0.004		500	0.040
	1000	0.005		1000	0.041
$T = 100$			$T = 1000$		
	100	0.005		100	0.058
	500	0.005		500	0.057
	1000	0.005		1000	0.056
$T = 150$			$T = 1500$		
	100	0.008		100	0.084
	500	0.008		500	0.086
	1000	0.007		1000	0.082

Table 2.3: Average times of Algorithm 1 when $k = 2$

From data in Table 2.3, we estimate the CPU time using regression analysis. We obtain the following expression of the running time (with R^2 equal to 0.999):

$$CPU \text{ time} = 6 \cdot 10^{-5} T$$

This result corroborates the fact that when demands range in $[0, W_t]$ the algorithm runs in $E(T)$.

2.3 The Linear Costs without Setup Case

In this section, we address problem P but assuming that $C_t(Q_t) = c_t Q_t$ and $H_t(I_t) = h_t I_t$ for $t = 1, \dots, T$. Hence, we focus our attention on the case where production/reorder and holding costs are linear. Notice that, in this case, setup costs are not included in the cost structure. Considering setup costs is studied apart in a later section. Although considering zero or near zero setup costs seems to be a non realistic assumption, there exist instances in practice where this assumption is admissible. For example, in electronic commerce, setup costs are assumed to be zero as discussed in Röckelein [76]. The same concern can be applied to firms where delivery charges (including shipment and handling of commodities) are assumed to be covered by the suppliers or to firms where fixed charges can be considered negligible in comparison with the unit purchase and holding costs, among other examples. Moreover, the assumption of zero costs leads to instances of economic lot scheduling [52] and lot sizing [30] problems.

It is assumed that inventory levels are to be non-negative. We introduce the network corresponding to this version of P. Furthermore, we devise an $O(T \log T)$ greedy algorithm to obtain optimal policies for this problem and we provide a numerical example to illustrate such an algorithm. Moreover, computational results on a randomly generated problems set are reported.

According to this cost structure, problem P can be reformulated to give problem P'

$$
\begin{array}{lll}
(P'): & \min \sum_{t=1}^{T} (c_t Q_t + h_t I_t) & \\
s.t. & & \\
 & I_0 = I_T = 0 & \\
 & I_{t-1} + Q_t - I_t = d_t & t = 1, \ldots, T \\
 & 0 \leq I_t \leq W_t - d_t & t = 1, \ldots, T-1 \\
 & Q_t \in \mathbb{N}_0 & t = 1, \ldots, T
\end{array}
$$

2.3.1 Greedy Algorithm

As in Zangwill [105], problem P' can be formulated as a Minimum Cost Flow problem. Assuming linear costs, the underlying network for this problem is as follows. Let $G(V,A)$ be a directed network, where V is the set of $n = T + 1$ (there are as much nodes as periods plus 1) nodes and A is the set of $m = 2T - 1$ arcs. Each node t ($t = 1, \ldots, T$) has a demand equals $-d_t$, whereas the node 0 (source node) has to fulfill the demand in each node with an amount equal to $d_{1,T+1}$.

We can distinguish two types of arcs: *production/reorder arcs* which are related to the decision variables Q_t, and the *inventory arcs* associated to the state variables I_t with $t = 1, \ldots, T$. We can assume without loss of generality that $I_0 = I_T = 0$. Each reorder arc $(0, t)$ in the network has an unit cost equals c_t and a capacity equal to infinite. On the other hand, each inventory arc $(t, t+1)$ has an unit cost equal to

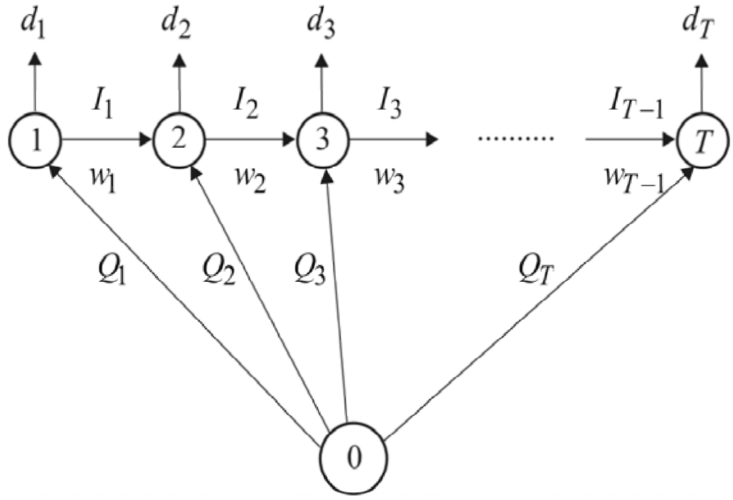

Figure 2.2: Network of problem P'.

h_t and a maximum inventory capacity of $w_t = W_t - d_t$ units. The network for the problem is depicted in Figure 2.2.

Using the previous definitions, we can state problem P' as a Minimum Cost Flow problem with the following mathematical formulation:

$$
\begin{aligned}
& \min \sum_{t=1}^{T} (c_t Q_t + h_t I_t) & \\
s.t. & & \\
& I_0 = I_T = 0 & \\
& I_{t-1} + Q_t - I_t = d_t & t = 1, \ldots, T \\
& 0 \leq I_t \leq w_t & t = 1, \ldots, T-1 \\
& Q_t \in \mathbb{N}_0 & t = 1, \ldots, T
\end{aligned}
$$

As it is well-known, there is an optimal solution to the above problem, which is an integer extreme point as a consequence of both the unimodularity property, inherent in the network flow problems, and the integrality of the input data.

Thus, problem P' can be solved using any Minimum Cost Flow (MCF) algorithm (see, for example, Ahuja et al. [2]). Currently, the best available time bounds for the MCF problem are shown in Table 2.4, where the third column contains the complexities adapted to the network of problem P'. Furthermore, the network depicted in Figure 2.2 for problem P' is also an instance of a series-parallel network. We refer to Duffin [26] for a comprehensive definition of topologies of series-parallel networks. The best known algorithm for solving the MCF problem in such networks is due to Booth and Tarjan [15], which runs in O($m \log m$) time and requires an O($m \log^* m$) space. Adapting these complexities to problem P' leads this procedure to run in O($T \log T$) time with O($T \log^* T$) space.

Authors(year)	*Theoretical complexity*	*Complexity for problem P*
Goldberg and Tarjan [43] (1986)	O($mnlog(n^2/m)lognC$)	O($T^2logTlogTC$)
Ahuja et al. [2] (1992)	O($mnloglogUlognC$)	O($T^2loglogUlogTC$)
Orlin [67] (1993)	O($mlogn(m + nlogn)$)	O(T^2log^2T)

Table 2.4: Complexities of MCF algorithms

However, we propose an *ad hoc* algorithm that exploits the characteristics of the above network. We will show, in a subsequent section, that this new algorithm runs in O($T \log T$) time and requires only O(T) space, improving so the complexity corresponding to the procedure of Booth and Tarjan [15]. In constrast to the data structures (e.g. finger search trees) used by these authors, our procedure needs only vectors which simplifies the implementation of this approach.

The algorithm is devised according to the following idea: for any period t ($t = 1, \ldots, T$), the demand d_t can be satisfied from reorders in any previous period and/or reordering in that period. In the network, the possible paths connecting node 0 and node t depict this situation. Notice that there are t possible paths (one with length 1, one with length 2,..., one with length t). The production/reorder quantity sent through each of those nodes to node t is limited by the capacities of the arcs in the path.

Assuming that the costs associated to the paths connecting node 0 to node t have been evaluated, then let us consider node 1. The demand for this node only can be fulfilled through arc $(0, 1)$. Therefore, the unique decision is to send d_1 flow units through this arc. Now, let us consider node 2. The demand of this node can be met from node 1 and/or from node 0. Thus, assume that the paths to node 2 are sorted in non-decreasing order of their costs. Accordingly, the minimum cost path to node 2 is selected. Then, a flow corresponding to the minimum between the demand d_2 and the minimum remaining capacity of the inventory arcs in such path is sent to node 2. If this quantity is greater than or equal to d_2, then the demand for node 2 is satisfied. Otherwise, the remaining quantity of flow to reach d_2 is sent through the next minimum cost path to node 2. Once the demand for node 2 is fulfilled, we consider node 3 and so until node T is attained. Remark that in this process, if an inventory arc $(t, t + 1)$ becomes *saturated*, that is, $I_t = w_t$, then this arc is not to be considered in future decisions. Such a situation implies that the number of paths to the rest of periods decreases in t units.

The following matrix AC of accumulated costs is required for the statement of the algorithm:

$$AC_{ij} = \begin{cases} c_i & \text{if } i = j \\ c_i + \sum_{k=i}^{j-1} h_k & \text{if } i \neq j \end{cases} \quad 1 \leq i \leq j \leq T$$

where AC_{ij} is the cost of reordering one unit of item in period i and holding to period j. The jth column in AC shows the costs corresponding to the j paths from node 0 to node j. Remark that column j, with $j = 1, \ldots T-1$, can be obtained from column T in AC. That is, $AC_{ij} = AC_{iT} - AC_{jT} + c_j$ with $i = 1, ..., j$. Note that to obtain column j, the same quantity $-AC_{jT} + c_j$ is added to the first j elements of column T. In other words, the order in column T determines the order in the rest of columns. Therefore, only column T must be considered to devise the algorithm. Throughout, we use the notation $AC(t) = AC_{tT}$ with $t = 1, ..., T$. Also, we denote column T in AC by AC^T.

Given that problem P' involves time varying storage capacities, we need to determine the residual capacity RW. This value represents the remaining capacity of the inventory arc $(i, i+1)$ when the accumulated demands from period $i+1$ to j (including both) are sent through this arc. The elements in matrix RW are calculated by the following expression:

$$RW_{ij} = w_i - d_{i+1,j+1} = W_i - d_{i,j+1} \quad 1 \leq i < j \leq T$$

Following a similar argument to that used in matrix AC, any column $j = 1, \ldots T-1$ in RW can be obtained from column T according to $RW_{ij} = RW_{iT} - RW_{j+1,T} + W_{j+1}$ with $i = 1, ..., j-1$. Note that the values of matrix RW can be negative. Moreover, given a column j in RW, the minimum of its values represent the bottleneck of previous inventory arcs to period j. To obtain the minimum, only is necessary to order this column. As in matrix AC, the order in column T determines the order in the rest of columns. Therefore, only column T must be considered to devise the algorithm. Throughout, we use the notation $RW(t) = RW_{iT}$ with $t = 1, ..., T-1$. Also, we denote by RW^T column T in RW.

The algorithm uses two sorted sequences: $SSAC$ and $SSRW$ that represent AC^T and RW^T, respectively. Both sequences are dynamically generated when new reorder periods or inventory arcs are considered. Sequence $SSAC$ contains the visited periods and their corresponding costs to node T. Periods in $SSAC$ are sorted in increasing order of the costs of the paths to node T. In addition, let t be any period in $SSAC$, then any period $tt < t$ with $AC(tt) \geq AC(t)$ is deleted from $SSAC$. This last operation is carried out immediately after period t is inserted in $SSAC$. In sequence $SSRW$ are included the visited inventory arcs and their corresponding residual capacity values associated to the paths to node T. The inventory arcs in $SSRW$ are sorted in increasing order of the residual capacity values. Also, let $(j, j+1)$ be an inventory arc in $SSRW$, then any inventory arc $(i, i+1)$ such that $i < j$ with $RW(i) \geq RW(j)$ is deleted from $SSRW$. Again, this last operation is performed immediately after inventory arc t is inserted in $SSRW$.

Assuming that t is the size of a sorted sequence, the following procedures: insert (IN) and delete-subsequence (DS) require a computational effort of $O(\log t)$. The operations min-key (MINK) and delete-min-key (DMK) are carried out in O(1) time. Finally, the function clear (Clear) runs in $O(t)$ time. The class sorted sequence (sortseq) and its procedures can be found in LEDA User Manual [58].

Taking the above notation and commentaries into account, we introduce the greedy method to solve the problem:

The algorithm starts calculating the accumulated demand from period i, $D_i = d_{i,T+1}$ with $i = 1, \ldots, T$, and the cost of ordering one unit of item in period 1 and holding to period T, that is, $Cost$ equals $AC(1)$. The next step consists of satisfying the demand for the first period updating d_1. Also, in this step, the procedure inserts the pair $\{Cost, 1\}$ in $SSAC$ and sets $Minkey = 1$. In each iteration t of the algorithm, $Minkey$ stores the period with smallest cost and greatest index. Then, within the loop, all periods with demand distinct to zero are chosen. When iteration t starts, the cost associated to period t is added to $SSAC$ and all the elements with position greater than the position of t (pos) in $SSAC$ are deleted ($DS(SSAC, pos+1)$,maxsize($SSAC$)). Also, the residual capacity, $RW(t-1)$, of the inventory arc $(t-1, t)$ is inserted in $SSRW$. Again, all the elements with position greater than the position of $t-1$ in $SSRW$ are removed ($DS(SSRW, (pos-1)+1)$,maxsize($SSRW$)). Now, if the minimum value in $SSAC$ (MINK($SSAC$)) corresponds to period t, then the new reorder period is t and its demand is satisfied from this period. Moreover, the algorithm sets $Minkey = t$ and deletes the sorted sequence $SSRW$(Clear($SSRW$)) because only inventory arcs greater than $(t-1, t)$ must be considered. Notice that in this case $SSAC = \{AC(t), t\}$ and $SSRW$ is empty. If MINK($SSAC$) is not equal to t, two cases can occur. The first case arises when MINK($SSAC$) is greater than $Minkey$. Under this assumption, the algorithm sets $Minkey =$ MINK($SSAC$) and searches in $SSRW$ the inventory arc with smallest residual capacity and index greater than or equal to $Minkey$. In the second case, $Minkey$ is already equal to MINK($SSAC$). In both cases, the inventory arc with minimum residual capacity is calculated and stored in the variable $Capacity$. This inventory arc is the first in $SSRW$ and represents the bottleneck of the reorder quantity. If $Capacity > d_t$, then

Algorithm 2 Determine an optimal plan $\mathbf{Q} = (Q_1, \ldots, Q_T)$ for problem P'

Data: vectors d, c, h and W, and the number of periods T

$D_{T+1} \leftarrow 0$; $D_T \leftarrow d_T$; $Cost \leftarrow c_1$
for $t \leftarrow T-1$ downto 1 **do**
    $D_t \leftarrow D_{t+1} + d_t$; $Cost \leftarrow Cost + h_t$;
end for
$Q_1 \leftarrow d_1$; $d_1 \leftarrow 0$; $MinKey \leftarrow 1$; IN$(SSAC, Cost, 1)$;
for $t \leftarrow 2$ to T **do**
    $Q_t \leftarrow 0$
    IN$(SSRW, W_{t-1} - D_{t-1}, t-1)$; DS$(SSRW, (pos-1)+1$,maxsize(SSRW));
    $Cost \leftarrow Cost - c_{t-1} - h_{t-1} + c_t$;
    IN$(SSAC, Cost, t)$; DS$(SSAC, pos+1$,maxsize$(SSAC))$;
    while $d_t > 0$ **do**
        if MINK$(SSAC) = t$ **then**
            $Q_t \leftarrow Q_t + d_t; d_t \leftarrow 0; Minkey \leftarrow t$; Clear$(SSRW)$;
        else if MINK$(SSAC) > Minkey$ **then**
            $Minkey \leftarrow$ MINK$(SSAC)$;
            while $Minkey >$MINK$(SSRW)$ **do**
                DMK$(SSRW)$;
            end while
        end if
        $Capacity \leftarrow W_{MINK(SSRW)} - D_{MINK(SSRW)} + D_{t+1} + d_t$;
        if $Capacity > d_t$ **then**
            $Q_{Minkey} \leftarrow Q_{Minkey} + d_t; d_t \leftarrow 0$;
        else
            $Q_{Minkey} \leftarrow Q_{Minkey} + Capacity; d_t \leftarrow d_t - Capacity$;
            while MINK$(SSRW) \geq$ MINK$(SSAC)$ **do**
                DMK$(SSAC)$;
            end while
        end if
    end while
end for

d_t units are reordered in period $Minkey$ and the algorithm proceeds to visit period $t+1$. Otherwise, that is, when $Capacity \leq d_t$, $Capacity$ units are reordered in period $Minkey$ and the demand for this period is updated to $d_t - Capacity$. From this point, the reorder period with smallest cost and index greater than MINK($SSRW$) is searched in $SSAC$. The same period t is revisited until d_t is zero. The process finishes when period T is attained.

As one can see, the above method is greedy. Since, for each period, the procedure chooses the remaining paths in non-decreasing cost order and it sends a quantity of items, which equals the minimum between the demand and the minimum residual capacity of the inventory arcs in such paths.

Theorem 5 *The greedy algorithm generates optimal plans for problem P'.*

Proof. It is clear that the algorithm generates feasible solutions to problem P'. Then, we only have to show the optimality of the plans. In each iteration, the demand for a given period is satisfied. Let t be any given period. Obviously, the demands for the previous periods have been fulfilled. Now, let x be the optimal solution for the first $t-1$ periods and let $g(x)$ be its cost. In period t, the algorithm chooses the path from 0 to t with minimum cost, say q. Suppose that this path allows to send d_t units of item. Thus, the partial solution cost is $g(x) + qd_t$. Moreover, assume that x contains a saturated path with cost p from node 0 to a node tt ($tt < t$) and a path with cost r from tt to t (see Figure 2.3). Let us admit that the cost $p + r$ is smaller than q. Now, we can perturb the plan x in such a way that α flow units from path with cost p are sent through an alternative path with cost s. If this alternative path does not exist, then there is no alternative solution. Since the algorithm selects the minimum cost path to node t, then q must be smaller than or equal to $s + r$. Therefore, we can send α flow units with cost $p + r$ and $d_t - \alpha$ with cost q. This new alternative plan will be optimal if

$$g(x) + qd_t > g(x) - \alpha p + \alpha s + \alpha(p + r) + (d_t - \alpha)q$$

However, as one can see, the previous expression yields q to be greater than $s+r$, which contradicts our hypothesis. We can conclude asserting that optimal plans can be obtained choosing the minimum cost path in each period. ■

We bear out the previous result with the following example.

An Example

Let us consider the parameters given in Table 2.5 corresponding to a dynamic lot size problem with time-varying storage capacities:

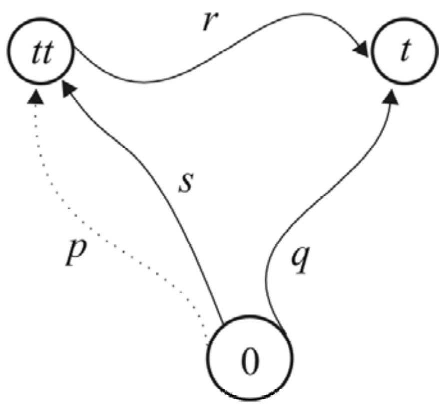

Figure 2.3: Illustration of Theorem 5.

i	1	2	3	4	5
d_i	4	3	4	2	3
c_i	0	4	16	6	9
h_i	1	2	2	1	x
W_i	6	5	7	5	3
D_i	16	12	9	5	3

Table 2.5: Data set for an instance of problem P'

For each iteration ($Iter$) of the algorithm, Table 2.6 shows the period (t), the remaining demand (d_t) and the sorted sequences $SSAC$ and $SSRW$. These sequences contain the period t and the inventory arc $(t-1, t)$, respectively. Also, this table contains variables $Minkey$ and $Capacity$, the decision vector $\mathbf{Q}$, and finally, the updated sequences $SSAC$ and $SSRW$.

From Table 2.6, the optimal decision for period 1 ($Iter$ 2 and 3) consists of ordering the demand for this period plus two units of the demand of the second period. Therefore, the inventory arc (1,2) is saturated. The optimal decision for period 2 ($Iter$ 3 and 4) is to order the remaining demand of this period plus two units of the demand of period 3. In this case, the inventory arc (2,3) is saturated. The remaining demand for period 3 is satisfied in this period ($Iter$ 5). Finally, the demands of periods 4 and 5 are met by reordering in period 4 ($Iter$ 6 and 7).

Iter	*t*	d_t	*SSAC*	*SSRW*	*Minkey*	*Capacity*	**Q**	*SSAC*	*SSRW*
1	1	4	{(6,1)}	-	1	-	4 0 0 0 0	{(6,1)}	-
2	2	3	{(6,1),(9,2)}	-	1	6-16+9+3=2	6 0 0 0 0	{(9,2)}	{(-10.1)}
3	2	1	{(9,2)}	{(-10,1)}	2	-	6 1 0 0 0	{(9,2)}	-
4	3	4	{(9,2),(19,3)}	{(-7,2)}	2	5-12+5+4=2	6 3 0 0 0	{(19,3)}	{(-7,2)}
5	3	2	{(19,3)}	{(-7,2)}	3	-	6 3 2 0 0	{(19,3)}	-
6	4	2	{(7,4)}	{(-2,3)}	4	-	6 3 2 2 0	{(7,4)}	-
7	5	3	{(7,4),(9,5)}	{(0,4)}	4	5-5+0+3=3	6 3 2 5 0	{(9,5)}	{(0,4)}

Table 2.6: Execution of Greedy Algorithm for the example

Complexity of Greedy Algorithm

In this section, we will prove that the complexity of the algorithm under the worst-case hypothesis is $O(T \log T)$.

Theorem 6 *The Greedy algorithm runs in* $O(T \log T)$

Proof. The algorithm iterates as many times as periods are considered. The same period is dealt until its demand is fulfilled. If a period t is visited tt times, then $tt-1$ inventory arcs have been saturated. Therefore, they are not involved in partial paths of subsequent periods to t. Thus, since the number of arcs is $m = 2T-1$ and, at least, one inventory arc is saturated when one period is visited more than once, then the overall number of examinations is at most, $2T-1$, that is, the sum of the number of reorder and inventory arcs.

Furthermore, both routines insert (IN) and delete-subsequence (DS) require an $O(\log t)$ computational effort, the operations min-key (MINK) and delete-min-key (DMK) take $O(1)$ time and function clear (Clear) runs in $O(t)$ time. In a period t, the computational effort of deleting $tt < t$ elements from any sorted sequence, using either the function clear or iteratively the routine delete-min-key, is $O(tt)$. Therefore, at most, $T - tt$ elements can be removed in the rest of periods. Hence, the algorithm runs in $O(\sum_{i=1}^{T} \log i + T)$, which yields the complexity of the algorithm to be $O(T \log T)$. ■

2.3.2 Computational Results

In this section, we report computational results for a set of randomly generated problems. The values for T and the maximum storage capacity (W) have been chosen as follows: $T = 1000, 2000, 3000, 4000,$ and 5000 periods and $W = 100, 500$ and 1000. Assuming integer input data, the storage capacity W_t for a period t ranges in the interval $[0, W]$. The demand d_t ranges in $[0, W_t)$ (feasibility assumption),

T	W	*Greedy Algorithm average CPU time (sec.)*
1000	100	0.080
1000	500	0.079
1000	1000	0.081
2000	100	0.156
2000	500	0.151
2000	1000	0.154
3000	100	0.235
3000	500	0.231
3000	1000	0.233
4000	100	0.311
4000	500	0.303
4000	1000	0.306
5000	100	0.392
5000	500	0.387
5000	1000	0.387

Table 2.7: Average running times for Greedy Algorithm

whereas the unit carrying and reorder costs were randomly generated in the interval $[0, 100]$. For each combination of T and W, ten replications have been considered. Therefore, the number of instances for the experience was 150 (5x3x10).

The greedy algorithm (Greedy) has been coded in C++ using LEDA libraries [58] and they have been tested in a HP-712/80 workstation. The results for the set of problems, considering different periods and capacities, are shown in Table 2.7. This table contains the number of periods (T), the maximum storage capacity (W) and the average running times for the Greedy algorithm. From the results in Table 2.7, it seems that the algorithm is not sensible to the storage capacities.

2.4 The Linear Costs with Setup Case

In this case, we modify the cost structure of problem P' to consider setup costs. Under this assumption, the greedy algorithm introduced in the previous section cannot be longer applied, a different approach should be developed instead. In particular, we generalize the geometrical technique proposed by Wagelmans et al. [95] to deal with storage capacity. Additionally, we also consider the cost structure given in Wagner and Whitin [97], namely, we disallow speculative motives for holding stock making the ordering costs to be constant. In this case, we get a surprising result since among the optimal policies, there exists one satisfying the zero inventory ordering (ZIO) property. Furthermore, we provide an $O(T \log T)$ algorithm to obtain optimal policies for this variant of problem P' along with a numerical example, which illustrates the algorithm. Moreover, we report computational results on a randomly generated problems set.

According to this cost structure, problem P' can be reformulated to give problem P''. Also, the statement of the cost function requires the following variable related to setup costs: $y_t = 1$ if $Q_t > 0$, and $y_t = 0$ otherwise. Then, we can state the dynamic lot size problem with storage capacities, or P'' for short, as follows:

$$
\begin{array}{lll}
(P'') & \min \sum_{t=1}^{T} (f_t y_t + c_t Q_t + h_t I_t) & \\
& s.t. & \\
& I_0 = I_T = 0 & \\
& Q_t + I_{t-1} - I_t = d_t & t = 1, \ldots, T \\
& d_{t,T+1} y_t - Q_t \geq 0 & t = 1, \ldots, T \\
& 0 \leq I_t \leq W_t - d_t & t = 1, \ldots, T \\
& Q_t, I_t \in \mathbb{N}_0,\ y_t \in \{0, 1\} & t = 1, \ldots, T
\end{array}
\quad (2.12)
$$

Notice that constraints in (2.12) coincide with the constraint set in previous models excepting those constraints related to y_t, which are binary variables, and feasibility is assured by the assumption that $d_t \leq W_t$ $(t = 1, \ldots, T)$.

Observe that, as a consequence of the storage constraints, the maximum quantity to be produced/ordered in a period is limited. Accordingly, let M_t be the maximum quantity to be produced or ordered in period t $(t = 1, \ldots, T-1)$, which can be easily derived from the following expression: $M_t = \min(M_{t+1} + d_t, W_t)$, where $M_T = d_T$. We also denote by p_t the maximum reachable period with demand completely satisfied with inventory held from period t $(t = 1, \ldots, T-1)$, that is, $p_t = \max(j : t \leq j \leq T$ and $(M_t - d_{t,j+1}) \geq 0)$, with $p_T = T$. The values M_t and p_t $(t = 1, \ldots, T-1)$ are determined from demand and storage capacity values in $O(T)$.

We introduce below the solution method which determines an optimal plan for problem P'' in $O(T \log T)$.

2.4.1 Solution Method

Let $G(t)$ be the optimal cost of the subproblem consisting of periods t to T $(t = 1, \ldots, T)$, with $G(T+1) = 0$, and for simplicity, let $AC_t = AC_{t,T+1} = c_t + \sum_{i=t}^{T} h_i$

be the accumulated cost from period t to period T. Moreover, let $\widehat{Q}_t$ be the optimal decision in period t when the subproblem consisting of periods t to T is solved, and let $\delta(z)$ denote a delta function such that $\delta(0) = 1$ and $\delta(z) = 0$ if $z \neq 0$. In addition, we denote by $Q^*_{s,t}$ the optimal decision in period t when a subproblem consisting of periods s to T is solved, being $s < t$.

By virtue of Theorem 1, when cost functions are concave, an optimal production/ordering plan $\mathbf{Q} = (Q_1, \ldots, Q_T)$ can always be found in $\mathrm{O}(T^3)$ so that for each production/ordering period t ($t = 1, \ldots, T$), $I_{t-1} + Q_t$ corresponds to either the sum of demands of consecutive periods or the maximum quantity M_t to be produced or ordered in that period. However, when cost functions are of the form as in problem P'', we can exploit the following results to develop an $\mathrm{O}(T^2)$ algorithm.

Lemma 7 *If $\widehat{Q}_t = M_t$ for a given period t, then there is at least one period $k \in [t+1, p_t+1]$ such that $Q^*_{j,k} \geq M_t - d_{t,k}$ for some period $j \in [t+1, k-1]$.*

Proof. By Theorem 1, we know that when the subproblem starting with period k is independently solved (i.e., assuming that $I_{k-1} = 0$), the only two decisions to consider are either $\widehat{Q}_k = d_{k,l}$ for some period $l \in [k+1, p_k+1]$, or $\widehat{Q}_k = M_k$. Moreover, we know that the quantity $\widehat{Q}_t = M_t$ is enough to completely satisfy the demands for periods t to p_t, and to partially satisfy the demand in period $p_t + 1$ (i.e., $M_t - d_{t,p_t+1} < d_{p_t+1}$, or equivalently, $M_t = d_{t,p_t+1} + \lambda d_{p_t+1}$ with $\lambda \in (0, 1)$).

Additionally, taking into account the way in which the values M_i ($i = 1, \ldots, T$) are obtained, it is clear that $M_i - d_{i,l} \leq M_l$ for all $l \in [i+1, p_i+1]$, otherwise there would be a period in $[i+1, p_i+1]$ where the storage constraint is violated.

By contradiction, let us admit that $Q^*_{j,k} < M_t - d_{t,k}$ for all $k \in [t+1, p_t+1]$ with $j \in [t+1, k-1]$, In particular, let us consider period j, so we obtain that $Q^*_{j,j} = \widehat{Q}_j < M_t - d_{t,j} = d_{j,p_t+1} + \lambda d_{p_t+1} \leq M_j$ with $\lambda \in (0, 1)$, and according to Theorem 1, the optimal decision $\widehat{Q}_j$ consists of the sum of demands, i.e., $\widehat{Q}_j \leq d_{j,p_t+1}$. Hence, to prevent a shortage, there should be a period in $[j+1, p_t+1]$, say i, where a quantity at least equal to d_{p_t+1} must be produced/ordered. However, by hypothesis, period i also holds that $Q^*_{j,i} < M_t - d_{t,i} = d_{i,p_t+1} + \lambda d_{p_t+1} \leq M_i$, or equivalently, by Theorem 1, $Q^*_{j,i} \leq d_{i,p_t+1}$, and hence it must be a period in $[i+1, p_t+1]$ such that at least d_{p_t+1} units should be produced/ordered. Following the same argument, we attain period $p_t + 1$, where $Q^*_{j,p_t+1} < M_t - d_{t,p_t+1} = \lambda d_{p_t+1} \leq M_{p_t+1}$, and in accordance with Theorem 1, $Q^*_{j,p_t+1} = 0$. As a result, the demand for period $p_t + 1$ has not been produced/ordered through periods from j to $p_t + 1$ and, therefore, a stockout occurs. Consequently, to avoid this infeasible fact, there must be at least a period $k \in [t+1, p_t+1]$ such that $Q^*_{j,k} \geq M_t - d_{t,k}$ for some period $j \in [t+1, k-1]$. ■

Lemma 8 *If the optimal decision in period t, $\widehat{Q}_t$, is to produce/order M_t, then the optimal decision for those periods $j \in [t+1, p_t+1]$ with $\widehat{Q}_j < M_t - d_{t,j}$ is not to order, i.e., $Q^*_{t,j} = 0$.*

Proof. Let t, j and k be three periods with production/ordering different from zero with $j < k$ and $j, k \in [t+1, p_t+1]$. Furthermore, let $\widehat{Q}_j$ and $Q^*_{j,k}$ be the optimal production/ordering quantities for periods j and k respectively, when the subproblem consisting of periods j to T is solved. Moreover, we assume that j is the first period in $[t+1, p_t+1]$ such that $\widehat{Q}_j < M_t - d_{t,j}$, and let k denote the first period in $[j+1, p_t+1]$ such that $Q^*_{j,k} \geq M_t - d_{t,k}$. The existence of a period $k \in [j+1, p_t+1]$ such that $Q^*_{j,k} \geq M_t - d_{t,k}$ is proved in Lemma 7. In addition, recall from the proof in Lemma 7 that any production/order period $i \in [j, k-1]$ satisfies $Q^*_{j,i} < M_t - d_{t,i} \leq M_i$. Therefore, by virtue of Theorem 1 , the optimal decision $Q^*_{j,i}$ for any production/order period $i \in [j, k-1]$ is either zero or sum of demands (i.e., ZIO subpolicies). Hence, for all period $i \in [j, k]$ such that $Q^*_{j,i} \neq 0$, it holds that $Q^*_{j,i} = \widehat{Q}_i$. In particular, $Q^*_{j,k} = \widehat{Q}_k$.

Accordingly, let A denote the set of indices related to production/order periods in $[j, k]$, and let $B = \{j, j+1, \ldots, k\}$ be the set of indices of all periods between j and k, including both. Additionally, let $q \in [0, \min(M_j - \widehat{Q}_j, \widehat{Q}_k)]$ be a quantity, which can be feasibly produced/ordered through periods j to $k-1$. Moreover, let $B^+ \subset B$ be the set of periods i where an additional amount is added to the optimal quantity $Q^*_{j,i} = \widehat{Q}_i$ for that period. Given that the net result in this process should be null, the same quantity q must be subtracted from the inventory in other periods. Thus, let $B^- \subset B$ denote the set of periods i where an amount is withdrawn of the optimal decision $Q^*_{j,i} = \widehat{Q}_i$. Finally, let $B^= \subset B$ be the set of periods which will not be modified. It is clear that $B = B^+ \cup B^- \cup B^=$, and that $B^+ \cap B^- = B^+ \cup B^= = B^- \cup B^= = \varnothing$. Besides, observe that $k \notin B^+$. Given that, by hypothesis, we are assuming that it is optimal to order $\widehat{Q}_j$ in j, $\widehat{Q}_k$ in k and $\widehat{Q}_i$ in any intermediate production period, then for any feasible combination $(q_j, q_{j+1}, \ldots, q_k)$ such that $\sum_{i \in B^+} q_i = \sum_{i \in B^-} q_i = q$, the following expression holds

$$\sum_{i \in A} (f_i + AC_i \widehat{Q}_i) \leq$$

$$\sum_{i \in B^+} (f_i + AC_i(Q^*_{j,i} + q_i)) + \sum_{i \in A \cap B^-} (f_i + AC_i(\widehat{Q}_i - q_i)) + \sum_{i \in A \cap B^=} (f_i + AC_i \widehat{Q}_i)$$

that is,

$$\sum_{i \in A \cap B^+} (f_i + AC_i \widehat{Q}_i) \leq \sum_{i \in B^+} (f_i + AC_i(Q^*_{j,i} + q_i)) - \sum_{i \in A \cap B^-} AC_i q_i$$

If $A \cap B^+ \neq \varnothing$, then any period in that set $(A \cap B^+)$ satisfies $Q^*_{j,i} = \widehat{Q}_i$. Accordingly,

$$0 \leq \sum_{i \in A \cap B^+} AC_i q_i + \sum_{i \in B^+/A} (f_i + AC_i(Q^*_{j,i} + q_i)) - \sum_{i \in A \cap B^-} AC_i q_i$$

Since the optimal decision of those periods in B^+/A is $Q^*_{j,i} = 0$, then

$$\sum_{i \in A \cap B^-} AC_i q_i \leq \sum_{i \in A \cap B^+} AC_i q_i + \sum_{i \in B^+/A} (f_i + AC_i q_i) \tag{2.13}$$

Note that in (2.13), the sum of q_i on the right-side hand is equal to q and the sum of q_i on the left-side hand in (2.13) is also equal to q.

Assume now, by contradiction, that producing/ordering quantity q through periods in B leads to a cost smaller than solely producing/ordering $\widehat{Q}_t = M_t$ in period t and $Q^*_{j,k} = \widehat{Q}_k - (M_t - d_{t,k-1})$ in period k. That is,

$$f_t + AC_t M_t + \sum_{i \in A \cap B^+} (f_i + AC_i q_i) + \sum_{i \in B^+/A} (f_i + AC_i q_i) - \sum_{i \in A \cap (B^-/\{k\})} AC_i q_i + f_k + AC_k(\widehat{Q}_k - (M_t - d_{t,k-1}) - q_k) < f_t + AC_t M_t + f_k + AC_k(\widehat{Q}_k - (M_t - d_{t,k-1}))$$

or equivalently,

$$\sum_{i \in A \cap B^+} (f_i + AC_i q_i) + \sum_{i \in B^+/A} (f_i + AC_i q_i) < \sum_{i \in A \cap (B^-/\{k\})} AC_i q_i + AC_k q_k$$

Note that if $k \in A \cap B^-$, then the expression above can be reformulated to yield

$$\sum_{i \in A \cap B^+} (f_i + AC_i q_i) + \sum_{i \in B^+/A} (f_i + AC_i q_i) < \sum_{i \in A \cap B^-} AC_i q_i$$

which contradicts (2.13). Otherwise, if $k \notin A \cap B^-$, then $k \in B^=$ and hence $q_k = 0$, obtaining the same expression.

As a consequence of the above result, we can conclude that the optimal decision for those periods $i \in [j, k-1]$ with $\widehat{Q}_i < M_t - d_{t,i-1}$ is not to order (i.e., $Q^*_{t,i} = 0, \forall i \in [j, k-1]$). ■

Lemma 9 *If* $\widehat{Q}_t = M_t$*, then there exists a period* $j \in [t+1, p_t+1]$ *with* $\widehat{Q}_j \geq M_t - d_{t,j}$ *so that its optimal decision is* $Q^*_{t,j} = \widehat{Q}_j - (M_t - d_{t,j})$.

Proof. We know, by Lemma 8, that there exists a period $j \in [t+1, p_t+1]$ such that $\widehat{Q}_j \geq M_t - d_{t,j-1}$, and that the optimal decision for those periods $i \in [t+1, j-1]$ with $\widehat{Q}_i < M_t - d_{t,i-1}$ is $Q^*_{t,i} = 0$. Moreover, since the optimal decision in period t is $\widehat{Q}_t = M_t = d_{t,j-1} + (M_t - d_{t,j-1})$, then we can assert that

$$AC_t(M_t - d_{t,j-1}) \leq AC_j(M_t - d_{t,j-1})$$

even when $M_t - d_{t,j-1} = \widehat{Q}_j$, since $AC_j(M_t - d_{t,j-1}) \leq f_j + AC_j(M_t - d_{t,j-1})$. Otherwise, $\widehat{Q}_t = M_t$ would not have been an optimal decision. ■

Theorem 10 *An optimal production/ordering plan for problem* P'' *is given by the following recurrence formula*

$$\begin{aligned} G(t) = \min \quad & [\{ \min_{t<j\leq p_t+1} (f_t + AC_t d_{t,j} + G(j)), \text{ if } d_t > 0, \\ & \text{or } \min[G(t+1), \min_{t+1<j\leq p_t+1} (f_t + AC_t d_{t,j} + G(j))], \text{ otherwise}\}, \\ & \min_{\substack{t<j\leq p_t+1 \\ \widehat{Q}_j \geq M_t - d_{t,j}}} (f_t + AC_t M_t + G(j) - F(t,j))] \end{aligned} \tag{2.14}$$

where $F(t,j) = AC_j(M_t - d_{t,j}) + \delta(M_t - d_{t,j} - \widehat{Q}_j) f_j$.

Proof. Assuming that $I_{t-1} = 0$ $(t = 1, \ldots, T)$, Theorem 1 states that the production/ordering quantity in period t consists of the sum of demands corresponding to consecutive periods or M_t. The former decision corresponds to the first "min" term within the brackets in (2.14). On the other hand, the latter decision concerns the second "min" term in the same expression. Indeed, when $\widehat{Q}_t = M_t$, only those periods j in $[t+1, p_t+1]$ satisfying $\widehat{Q}_j \geq M_t - d_{t,j}$ must be considered as it is shown in Lemmas 8 and 9. ■

It is clear that a straightforward implementation of this recursion leads to an O(T^2) algorithm, reducing the complexity O(T^3) of both Love's procedure and Algorithm 1, which have been devised for more general cost structures. Nevertheless, a more efficient algorithm can be devised applying a procedure based on the approach proposed by Wagelmans et al. [95]. In particular, these authors argued that only the *efficient periods* should be considered for the determination of $\min_{t<j\leq T+1}\{AC_t d_{t,j} + G(j)\}$ in the uncapacitated case. Accordingly, a period is said to be efficient when it corresponds to a breakpoint of the lower convex envelope of points $(d_{t,T}+1, G(t))$, $t = 1, \ldots, T+1$. The implementation of this technique consists of evaluating the periods from T to 1 and holding the efficient periods in a list L. This list is sorted by ratios which represent the slopes of the line segments joining consecutive efficient periods (breakpoints) of the lower convex envelope. Each time a new period j is considered, the procedure looks for the smallest efficient period $q(j)$ in L with ratio smaller than AC_j, and the lower envelope is updated removing from L the non-efficient periods $j+1$ to the predecessor of $q(j)$ in L.

Unfortunately, this technique can not be used directly when the inventory levels are limited. Unlike the geometrical approach proposed by Wagelmans et al. [95], in our procedure the non-efficient periods can not be discarded since a period that is not efficient for a subproblem consisting of periods j to $p_j + 1$ could be efficient for a subproblem involving periods t to $p_t + 1$, with $j > t$. However, we can adapt this geometrical technique to our model in the following way. We should define two lists L_E and L_{NE} containing, respectively, the efficient and non-efficient periods. When evaluating period j, if $q(j)$ is smaller than $p_j + 1$, then the new procedure proceeds in the same way as the approach in Wagelmans et al. [95], i.e., producing/ordering $d_{j,q(j)}$ units. In case of $q(j)$ equals p_j+1, we can make two decisions, namely, to order either M_j or d_{j,p_j+1}. Nevertheless, it can be easily proved that when $AC_j < AC_{q(j)}$ the optimal decision consists of producing/ordering M_j, and $d_{j,q(j)}$ otherwise. Finally, when $q(j) > p_j + 1$, the efficient period $q(j)$ is not feasible for the subproblem starting in period j, and hence we must compare the efficient period with smallest ratio $q_E(j) \leq p_j + 1$ in L_E with the non-efficient period $q_{NE}(j) \leq p_j + 1$ in L_{NE}. Accordingly, we denote by $G_E(j) = f_j + AC_j d_{j,q_E(j)} + G(q_E(j))$ and $G_{NE}(j) = f_j + AC_j d_{j,q_{NE}(j)} + G(q_{NE}(j))$ the costs associated to, respectively, periods $q_E(j)$ and $q_{NE}(j)$, which are the successors of j. If evaluating both costs we obtain that $G_E(j) \leq G_{NE}(j)$, then period $q_E(j)$ remains to be efficient. Otherwise, the following proposition shows that since $G_E(j) > G_{NE}(j)$, period $q_{NE}(j)$ should be inserted in list L_E and the rest of periods in this list have to be moved to L_{NE}. Actually, this process of transferring periods from one list to the other represents an update of the lower envelope.

Proposition 11 *If evaluating a period j, both $q(j) > p_j + 1$ and $G_E(j) > G_{NE}(j)$ hold, then period $q_{NE}(j)$ should be included in list L_E and the rest of periods in this list must be moved to list L_{NE}.*

Proof. Without loss of generality, we assume that $q(q_E(j)) = q(j)$ and $q(j)$ is the period successor to $q_E(j)$ in L_E. Notice that $G(q_{NE}(j)) + AC_j d_{l,q_{NE}(j)} < G(l)$ for any period l in L_E smaller than or equal to $q_E(j)$, and hence $G(q_{NE}(j)) < G(l)$. Otherwise, $f_j + AC_j d_{j,k} + G(k) < f_j + AC_j d_{j,q_{NE(j)}} + G_{NE}(j)$ for some $k \leq q_E(j)$ in L_E, and therefore $q_E(j) = k$ with $G_E(j) < G_{NE}(j)$, which contradicts the hypothesis. Recall that for a production/reordering period t, $\widehat{Q}_t \in \{M_t, d_{t,q(t)}\}$. In addition, since $q_{NE}(j) < q_E(j)$, the straight line connecting points $(d_{j,T+1}, G_{NE}(j))$ and $(d_{q_{NE}(j),T+1} - (\widehat{Q}_j - d_{j,q_{NE}(j)}), G(q_{NE}(j)) - AC_{q_{NE}(j)}(\widehat{Q}_j - d_{j,q_{NE}(j)-1}))$ intercepts the line segment joining points $(d_{q_E(j),T+1}, G(q_E(j)))$ and $(d_{q(j),T+1} - (\widehat{Q}_{q_E(j)} - d_{q_E(j),q(j)}), G(q(j)) - AC_{q(j)}(\widehat{Q}_{q_E(j)} - d_{q_E(j),q(j)}))$ in a point smaller than $q_E(j)$, and hence the result below follows

$$\frac{G_{NE}(j) - (G(q_{NE}(j)) - AC_{q_{NE}(j)}(\widehat{Q}_j - d_{j,q_{NE}(j)}))}{\widehat{Q}_j} < \frac{G(q_E(j)) - (G(q(j)) - AC_{q(j)}(\widehat{Q}_{q_E(j)} - d_{q_E(j),q(j)}))}{\widehat{Q}_{q_E(j)}}$$

Moreover, given that the term on the right-hand side in the above expression is smaller than the ratio $\frac{G(k)-G(q(k))}{d_{k,q(k)}}$, for any period $k < q_E(j)$ in L_E, these periods are to be dominated by $q_{NE}(j)$. For that reason, these periods should be moved to list L_{NE}. Figure 2.4 shows the case where $\widehat{Q}_j = d_{j,q_{NE}(j)}$. Notice that periods highlighted by the gray line are not accessible from period j. ■

Following a similar argument to that in the previous proposition, we can state the following result

Proposition 12 *If evaluating a period j, it holds that $q(j) > p_j + 1$, $G_E(j) < G_{NE}(j)$ and the ratio related to period $q_E(j)$ is greater than $\frac{G_E(j)-G(q_E(j))}{d_{j,q_E(j)}}$, then every period $k \leq q_E(j)$ in L_E should be moved to list L_{NE}.*

The method outlined above is shown in Algorithm 3, where $pred(j)$ and $succ(j)$ denote, respectively, the period predecessor and succesor of period j in both lists. We also follow the convention that if $d_j = 0$, then the efficient period $j+1$ is replaced by the efficient period j. Regarding the complexity of this procedure, notice that the value $q(j)$ can be obtained by binary search in $O(\log T)$. In case of $q(j) > p_j + 1$, the procedure should inspect by sequential search both L_E and L_{NE} to determine the actual period, q_E or q_{NE}, succesor of j. Specifically, if we are evaluating period j, there would be, at most, $(T - j)$ periods distributed in both lists. Each time the sequental search reaches a period greater than $p_j + 1$, this period is removed from the corresponding list, and it will not be considered in subsequent search processes. Observe that each comparison in any of the two lists, when $q(j) > p_j + 1$, yields a deletion of the corresponding period. Hence, the overall number of comparisons

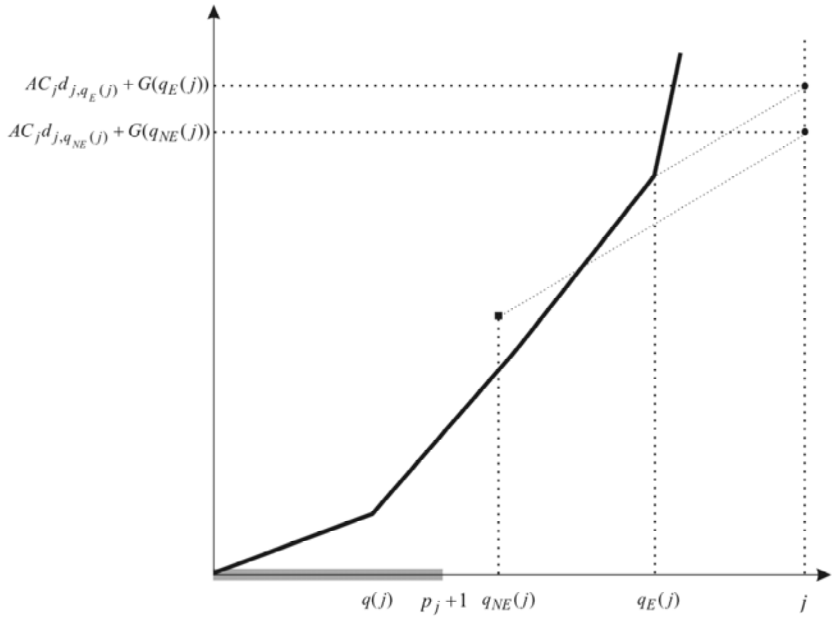

Figure 2.4: Illustration of the case $q(j) > p_j + 1$ and $G_E(j) > G_{NE}(j)$.

is $\mathrm{O}(T)$. Therefore, the process of searching all values $q(j)'s$ $(j = 1, \ldots, T)$ runs in $\mathrm{O}(T \log T)+\mathrm{O}(T)$. Additionally, notice that any period j can be moved between the lists at most two times, and so, the transferring process is $\mathrm{O}(T)$. According to the previous arguments, the algorithm runs in $\mathrm{O}(T \log T)$.

In addition to the case where production/ordering costs are time-varying, we also address the problem admitting that production/ordering costs are constant, i.e., when $c_t = c$ for all t. Under this assumption, the formulation of problem P'' adopts an equivalent form to the one in the Wagner and Whitin model and, hence, $AC_1, AC_2, \ldots, AC_T$ represent a non increasing sequence of values. Therefore, speculative motives for holding stock are not allowed. It is well-known that, under this assumption of the costs and in absence of capacities, the problem admits an optimal plan $\mathbf{Q} = (Q_1, \ldots, Q_T)$ verifying $I_{t-1}Q_t = 0$, for $t = 1, \ldots, T$. This result is commonly refereed to as Zero Inventory Ordering (ZIO) property. Indeed, the ZIO property still holds when the cost functions are concave in general (see Wagner [96] and Zangwill [105]). Moreover, as we show in Proposition 13, the ZIO property holds even when inventory levels are limited. Therefore, the use of the ZIO property is not conditioned to limitations on the inventory levels as the following proposition states.

Proposition 13 *When production/ordering costs in problem P'' are constant, there*

Algorithm 3 Determine an optimal plan $\mathbf{x} = (x_1, \ldots, x_T)$ for problem P''

Data: vectors d, c, h, f, W and the number of periods T

calculate AC_t, M_t and p_t, $t = 1, \ldots, T+1$
$G(T+1) \leftarrow 0$
insert $T+1$ in L_E
for $i \leftarrow T$ downto 1 **do**
  search for $q(i) \leftarrow min[T+1, min\{j \in L_E : \frac{G(j)-G(succ(j))}{\widehat{Q}_j} < AC_i\}]$
  if $(q(i) < p_i + 1)$ or $(q(i) = p_i + 1$ and $AC_i > AC_{q(i)})$ **then**
    $G(i) \leftarrow f_i + AC_i d_{i,q(i)} + G(q(i)); \widehat{Q}_i \leftarrow d_{i,q(i)}$
  else
    if $q(i) = p_i + 1$ **then**
      $G(i) = f_i + AC_i M_i + G(q(i)) - AC_{q(i)}(M_i - d_{i,q(i)}); \widehat{Q}_i \leftarrow M_i$
    else
      $j \leftarrow pred(q(i))$; **while** $j > p_i + 1$ **do** $j \leftarrow pred(j)$
      delete all $k : q_E(i) \leq k < j$ **from** L_E
      $q(i) \leftarrow j$; $G_E(i) \leftarrow f_i + AC_i d_{i,q_E(i)} + G(q_E(i))$; $G_{NE}(i) \leftarrow -1$
      if L_{NE} **is not empty then**
        $j \leftarrow$ **first element in** L_{NE}; **while** $j > p_i + 1$ **do** $j \leftarrow pred(j)$
        delete all $k : 1 \leq k < j$ **from** L_{NE}; $q_{NE}(i) \leftarrow j$
        if $(q_{NE}(i) < p_i + 1)$ **or** $(q_{NE}(i) = p_i + 1$ **and** $AC_i > AC_{q_{NE}(i)})$ **then**
          $G_{NE}(i) \leftarrow f_i + AC_i d_{i,q_{NE}(i)} + G(q_{NE}(i))$; $z \leftarrow d_{i,q_{NE}(i)}$
        else
          $G_{NE}(i) \leftarrow f_i + AC_i M_i + G(q_{NE}(i)) - AC_{q_{NE}(i)}(M_i - d_{i,q_{NE}(i)})$; $z \leftarrow M_i$
        end if
      end if
      if $G_{NE}(i) \geq 0$ **and** $G_{NE}(i) < G_E(i)$ **then**
        $G(i) \leftarrow G_{NE}(i)$; $q(i) \leftarrow q_{NE}(i)$; $\widehat{Q}_i \leftarrow z$
      else
        $G(i) \leftarrow G_E(i)$; $q(i) \leftarrow q_E(i)$; $\widehat{Q}_i \leftarrow z$
      end if
    end if
  end if
  at this point, values $G(i)$ and $\widehat{Q}_i$ have been already determined
  call the routine to update the lower envelope
end for
call the routine to arrange the optimal solution

Algorithm 4 Routine to update the lower envelope

if $q(i) \leq p_i + 1$ or $(q(i) = q_E(i)$ and $\frac{G(i)-G(q(i)}{\widehat{Q}_i} > \frac{G(q(i))-G(succ(q(i)))}{\widehat{Q}_{q(i)}})$ **then**
  if $d_i = 0$ and $G(i+1) < G(i)$ **then**
    $G(i) \leftarrow G(i+1)$; $s \leftarrow succ(i+1)$
  else
    if $d_i > 0$**, then** $s \leftarrow i+1$ **else** $s \leftarrow succ(i+1)$
    while $\frac{G(i)-G(s)}{d_{i,s}} \leq \frac{G(s)-G(succ(s))}{\widehat{Q}_s}$ **and** $s < q(i)$ **do**
      $s \leftarrow succ(s)$
    end while
  end if
  move all $k : i+1 \leq k < s$ from L_E to L_{NE}; insert i in L_E
else
  move all periods in L_E to L_{NE}; insert i in L_E
end if

Algorithm 5 Routine to arrange the optimal solution

$Cost \leftarrow 0$; $i \leftarrow 1$; $x \leftarrow 0$; $Rest \leftarrow 0$
while $i \leq T$ **do**
  if $d_i = 0$ and $G(i) = G(i+1)$ **then**
    $i \leftarrow i+1$
  else
    if $q(i) = p_i + 1$ and $AC_i < AC_{q(i)}$ **then**
      $\widehat{Q}_i \leftarrow M_i - Rest$; $x \leftarrow \widehat{Q}_i + Rest - d_i$; $Rest \leftarrow M_i - d_{i,q(i)}$
    else
      $\widehat{Q}_i \leftarrow d_{i,q(i)} - Rest$; $x \leftarrow \widehat{Q}_i + Rest - d_i$; $Rest \leftarrow 0$
    end if
    if $\widehat{Q}_i = 0$ **then** $f \leftarrow 0$ **else** $f \leftarrow f_i$
    $Cost \leftarrow Cost + f + AC_i\widehat{Q}_i + xh_i$
    for $k \leftarrow i+1$ to $q(i) - 1$ **do**
      $x \leftarrow x - d_k$; $Cost \leftarrow Cost + xh_k$
    end for
    $i \leftarrow q(i)$
  end if
end while
return $Cost$

always exists an optimal policy $\mathbf{Q} = (Q_1, \ldots, Q_T)$ *such that* $I_{t-1}Q_t = 0$, $t = 1, \ldots, T$.

Proof. Let us assume that there exists an optimal plan $\mathbf{Q}$ with at least one period j such that $I_{j-1}Q_j \neq 0$. According to Lemma 9, since $Q_j \neq 0$, there must be a period t, $t < j$, such that $\widehat{Q}_j$ is strictly greater than $M_t - d_{t,j}$, which corresponds to I_{j-1}. Therefore, the following inequality holds

$$f_t + AC_tM_t + G(j) - AC_j(M_t - d_{t,j}) < f_t + C_td_{t,j} + G(j)$$

that is, $AC_t < AC_j$, which contradicts the fact that $AC_t \geq AC_j$. ■

The above proposition allow us to reformulate expression (2.14) as follows

$$G(t) = \begin{cases} \min\limits_{t<j\leq p_t+1} (f_t + AC_td_{t,j} + G(j)) & \text{if } d_t > 0, \\ \min[G(t+1), \min\limits_{t+1<j\leq p_t+1} (f_t + AC_td_{t,j} + G(j))] & \text{if } d_t = 0 \end{cases}$$

which only differs from that proposed by Wagelmans et al. [95] in the range of j. Unfortunately, this result does not imply a computational improvement since each non-efficient period should be sorted in $O(\log T)$ when it is inserted in L_{NE}.

As an illustration of this latter result, we present a numerical example. Assuming that the production/ordering unit costs are equal to zero, the rest of the input data are shown in Table 2.8, where the first column corresponds to the period and the following columns represent, respectively, the demand, the setup cost, the accumulated cost and the storage capacity.

The corresponding trace to the instance introduced in Table 2.8 is shown in Table 2.9. In particular, the rows in this table stand for the iterations (periods), and the second and third columns show the maximum quantity to be produced/ordered and the maximum reachable period for each period, respectively. Additionally, we show in columns four to six the values of $q(j)$, $G(j)$ and $Ratio = (G(j) - G(succ(j)))/\widehat{Q}_j$. Finally, the last two columns contain lists L_E and L_{NE}, where the symbol $\{\o\}$ indicates that the list is empty. Notice that, in absence of capacities, the optimal solution for the example in Table 2.8 is $(22, 0, 0, 24, 0, 22, 0, 0, 8, 0)$ whereas considering capacities yields the optimal plan to be $(5, 10, 7, 9, 15, 12, 0, 10, 8, 0)$.

2.4.2 Computational Experience

We show in Table 2.10 the average running times of Algorithm 3 introduced above, and the average running times of the dynamic programming algorithm developed

j	d_j	f_j	AC_j	W_j
1	5	1	10	10
2	10	30	9	15
3	7	20	8	10
4	9	2	7	20
5	15	40	6	25
6	4	1	5	22
7	8	30	4	10
8	10	25	3	10
9	2	10	2	10
10	6	28	1	10

Table 2.8: Input data for one instance of problem (P'')

from the recurrence formula (2.14). Both algorithms have been implemented using C++ along with LEDA 4.2.1 libraries [58] and were tested in a HP-712/80 workstation. For simplicity, we denote by $T \log T$ the Algorithm 3 and by T^2 the algorithm obtained from (2.14), respectively. The different values for the maximum storage capacity (W) and the number of periods (T) are shown in the first row and column, respectively. For each pair (W, T), we have run thirty instances with d_t varying in $[0, W_t]$, $t = 1, \ldots, T$. Moreover, for each pair (W, T), we show two columns: the first containing the average running times of the dynamic programming algorithm based on the recurrence formula (2.14) and the second showing the average running times of Algorithm 3.

2.5 Conclusions

In this chapter, the dynamic lot size problem with time-varying storage capacities has been studied. This model was solved previously by Love [60] using a different characterization approach. We have provided new properties which identify optimal plans. These new properties allow to devise an efficient algorithm which determines optimal policies over thirty times faster than the procedure proposed by Love [60]. Indeed, we have shown that Algorithm 1 runs in $E(T)$ when demands range in $[0, W]$. Moreover, we have shown that more efficient algorithms can be obtained for more specific cost structures. In particular, we have provided an $O(T \log T)$ greedy algorithm to determine optimal plans in the case of linear costs and in absence of setup costs. Furthermore, we have introduced an efficient recurrence expression for the case with linear and setup costs, which permits to devise an $O(T^2)$ algorithm.

j	M_j	p_j	$q(j)$	$G(j)$	$Ratio$	L_E	L_{NE}
10	6	10	11	34	5.66	{11, 10}	{ø}
9	8	10	11	26	3.25	{11, 9}	{10}
8	10	8	9	81	5.50	{11, 9, 8}	{ø}
7	10	7	8	143	7.75	{11, 9, 8, 7}	{ø}
6	14	7	8	142	5.08	{9, 6}	{8, 7}
5	25	6	6	272	8.66	{9, 6, 5}	{8, 7}
4	20	4	5	337	7.22	{6, 4}	{5}
3	10	3	4	413	10.85	{6, 4, 3}	{5}
2	15	2	3	533	12.00	{6, 3, 2}	{ø}
1	10	1	2	584	10.20	{3, 1}	{2}

Table 2.9: The output related to the instance in Table 2.8

	T									
	25		50		75		100		150	
W	T^2	$TlogT$	T^2	$TlogT$	T^2	$TlogT$	T^2	$TlogT$	T^2	$TlogT$
100	0.016	0.005	0.046	0.012	0.088	0.023	0.138	0.035	0.274	0.066
500	0.015	0.005	0.040	0.013	0.070	0.025	0.103	0.037	0.179	0.069
1000	0.015	0.006	0.041	0.013	0.071	0.023	0.104	0.036	0.180	0.069
2000	0.015	0.006	0.041	0.013	0.069	0.022	0.106	0.037	0.186	0.069
5000	0.016	0.006	0.043	0.014	0.074	0.024	0.108	0.036	0.193	0.070
10000	0.014	0.005	0.044	0.013	0.074	0.024	0.111	0.036	0.193	0.069
100000	0.015	0.005	0.040	0.015	0.071	0.024	0.105	0.038	0.188	0.069

Table 2.10: Average running times in sec. for the dynamic programming algorithm based on (2.14) (T^2) and Algorithm 3 ($TlogT$)

Nevertheless, we have also proven that an adaptation of the geometrical technique of Wagelmans et al. [95] can be exploited to develop an $O(T \log T)$ algorithm. Another relevant aspect introduced in this chapter is that the Zero Inventory Ordering (ZIO) property holds when cost specifications are as in Wagner and Whitin [97].

Future research will focus on the extension of the model to the situation in which shortages are allowed and to the multi-item case.

Chapter 3

The Dynamic EOQ under Uncertainty

3.1 Introduction

Unlike the original dynamic lot size problem introduced by Wagner and Whitin [97], where the demands through the whole horizon are known, in this chapter we consider that the demand vector is unknown rather than the total demand, which is assumed to be a fixed value. Furthermore, for each period, the demand can be chosen from a discrete finite set. As a result, different scenarios can arise combining the different admissible values of the demand per period. One of the most common examples for this problem are the promotions to clear stock. In this case, although we know in advance the total number of items to be sold, we cannot determine an optimal reorder plan because it is impossible to know with certainty how the demand is to occur period by period. Another instance happens when a wholesaler of bricks should satisfy the demands for distinct builders. Despite the wholesaler may know in advance the total demand of bricks needed to carry out the different constructions, he does not know how this total demand is distributed through the planning horizon. However, the decision maker can assume that the demand per period is taken from a discrete finite set. Besides, we allow in our model that the production/reorder and holding cost vectors change from one scenario to another. Taking into account these assumptions, the decision maker can not predict what scenario is to occur. Therefore, this problem deals with the optimization under uncertainty and, it takes place when a firm has to make a decision under variable market conditions. In fact, the uncertainty is present up to a point in almost all the decisions made in the real world.

How to handle the uncertainty in the scenario occurrence is not easy at all. One may want to come up with a unique solution using conservative techniques or the principle of incomplete reason (utilities). On the other hand, one may want to obtain the whole range of solutions that are non-dominated componentwise, as a first step in the analysis of the problem, in order to shed light on the decision process. This set can be seen as a sensitivity analysis of the solutions of the scenario problem for any 'a priori' information on the occurrence of the scenarios. This is the way that we follow in subsequent sections. The former analysis is normative: it prescribes a concrete course of action (based on a utility), the latter is descriptive: it informs on the variability of the solution space. Both analyses have advantages and disadvantages. The final decision should be made according to the goals of the decision maker. Notice that our goal in this chapter is to study the second approach. It is worth remarking that similar analyses have been followed for other scenario problems in the recent literature of Operations Research (see, for instance, Puerto and Fernández [72], Fernández et al. [35], Fernández and Puerto [34], Rodríguez-Chia [77]).

Dantzig [53] mentions the importance of considering uncertainty in the systems. In this sense, the so-called scenario analysis has been developed to deal with the problem of the uncertainty. Assuming that all the different situations of the system can be identified, this approach calculates the non-dominated solutions. These solutions are robust with respect to any possible occurrence because they are non-dominated, componentwise, by any other. Therefore, the approach consists of obtaining the Pareto-optimal solution set.

This chapter is devoted to the problem of determining the Pareto-optimal policies for the multiscenario dynamic lot sizing problem. As in Chapter 2, we assume a planning horizon splitted into T periods for each scenario. Three T-tuple vectors represent the input data for each scenario: a deterministic demand vector, the carrying cost vector and the replenishment cost vector. Also, in the shortages case, a shortage cost vector is considered. As usual, in absence of shortages, the overall cost function consists of the sum of carrying and replenishment costs. The goal is to schedule production/reorder in the various periods of each scenario so as to satisfy demand at minimal cost simultaneously in all the scenarios.

The problem under study fits into the Multi-Objective Combinatorial Optimization (MOCO). MOCO problems are an emergent area of research in many fields of Operations Research (see e.g. Gandibleux et al. [38], Ulungu and Teghem [89]). Nowadays, Multi-Objective Combinatorial Optimization (MOCO) (see Ehrgott and Gandibleux [28]; Ulungu and Teghem [89]) provides an adequate framework to tackle various types of discrete multicriteria problems. Within this research area, several methods are known to handle different problems. Two of them are dynamic programming enumeration (see Villarreal and Karwan [93] for a methodological description and Klamroth and Wiecek [57] for a recent application to knapsack problems) and implicit enumeration (Zionts and Wallenius [109]; Zionts [108]; Rasmussen [74]). In particular, the branch and bound scheme corresponds to an implicit enumeration method and, although it is widely used in the single objective case, only a few papers apply this technique for MOCO since bounds may be difficult to compute (see, e.g., Villarreal et al. [94], Ramesh et al. [73] and Alves and Climaco [4]. The reader is referred to [28] for a complete survey of multiobjective combinatorial optimization methods).

It is worth noting that most MOCO problems are NP-hard and intractable. In most cases, even if the single objective problem is polynomially solvable, the multiobjective version becomes NP-hard. This is the case of spanning tree problems and min-cost flow problems, among others. As we have mentioned, an important tool to deal with these problems is the multi-criteria dynamic programming (MDP) [28]. In the single objective case, Morin and Esoboque [63] exploited the embedded-state recursive equations to overcome many of the problems caused by the curse of the dimensionality (see, for example, Bellman and Dreyfus [9] and Nemhauser [66]). As an extension of the previous result, Villarreal and Karwan [93] introduced a procedure based on the Dynamic Multicriteria Discrete Mathematical Programming (DMDMP) to generate the Pareto-optimal solution set for problems with more than one objective function. We will make use of these techniques to resolve our model. In this context, when time and efficiency become a real issue, different alternatives can be used to approximate the Pareto-optimal set. One of them is the use of general-purpose MOCO heuristics (Gandibleux et al.[38]). Another possibility is the design of *ad hoc* methods based on computing the extreme non-dominated solutions. Obviously, this last strategy does not guarantee that we obtain the whole set of non-dominated solutions. Nevertheless the reduction in computation time can be remarkable.

The rest of this chapter is organized as follows. Section 3.2 introduces the notation and the model. In Section 3.3, we show that when the objective function is concave and shortages are not allowed, the extreme points of the region of feasible production plans satisfy a modified version of the ZIO (Zero Inventory Order) property, and that the Pareto-optimal set will always contain modified ZIO solutions. Therefore, we propose an algorithm to compute this approximated solution set: the non-dominated modified ZIO policies. A subset of such policies will be used later as initial upper bound set in the general algorithm. Furthermore, in Section 3.4, when shortages are allowed, we show that the extreme points of the polyhedron satisfy a modified version of the property for the single scenario case. Again, a subset of the non-dominated policies satisfying the latter property are proposed as the initial upper bound set for the algorithm when shortages are allowed. In Section 3.5, we propose a multicriteria dynamic programming procedure (MDP) that solves the problem and a branch and bound scheme to reduce the computational burden of the MDP algorithm. Also, in Section 3.6, computational results are reported for a set of dynamic multiscenario lot size instances. Finally, Section 3.7 contains conclusions and some further remarks.

3.2 Notation and Problem Statement

It is assumed that M scenarios or replications of that system are to be considered simultaneously and a unique (robust) policy belonging to the Pareto-optimal set is to be implemented. These replications model uncertainty in the parameter estimation, since neither the true values of the parameters of the system nor a probability distribution over them are known before hand. Therefore, we look for compromise solutions which must behave acceptably well in any of the admissible scenarios. This sort of system represents a multiple/serial decision process, since each scenario behaves as a serial multiperiod decision system and each production/reorder decision implies a parallel decision process. A graphical representation of this process is shown in Figure 3.1.

Throughout we use the following notation.

$h_i^j(\cdot)$: holding cost for the jth period in the ith scenario.
$c_i^j(\cdot)$: production/reorder cost for the jth period in the ith scenario.
I_i^j : inventory on hand at the end of the jth period in the ith scenario.
d_i^j : the demand for the jth period in the ith scenario.
D : the total demand ($\sum_{j=1}^T d_i^j = \sum_{j=1}^T d_s^j$ for any i and s in $\{1, \ldots, M\}$).
Q_j : the production/reorder quantity for the jth period.

We assume, without loss of generality, that $I_i^0 = I_i^T = 0$ for $i = 1, \ldots, M$.

The following definitions are required to simplify the formulation of the problem. Given a production/reorder vector $\mathbf{Q} = (Q_1, \ldots, Q_T) \in \mathbb{N}_0^T$, the inventory level vector for a scenario i is denoted by $I_i(\mathbf{Q}) = (I_i^1, \ldots, I_i^T)$, where

$$I_i^j = I_i^{j-1} + Q_j - d_i^j, \; j = 1, \ldots, T. \tag{3.1}$$

In addition, the accumulated cost from period j to period k in scenario i is given by

$$R_i^{j,k}(\mathbf{Q}) = \sum_{t=j}^{k} r_i^t(Q_t, I_i^t) \tag{3.2}$$

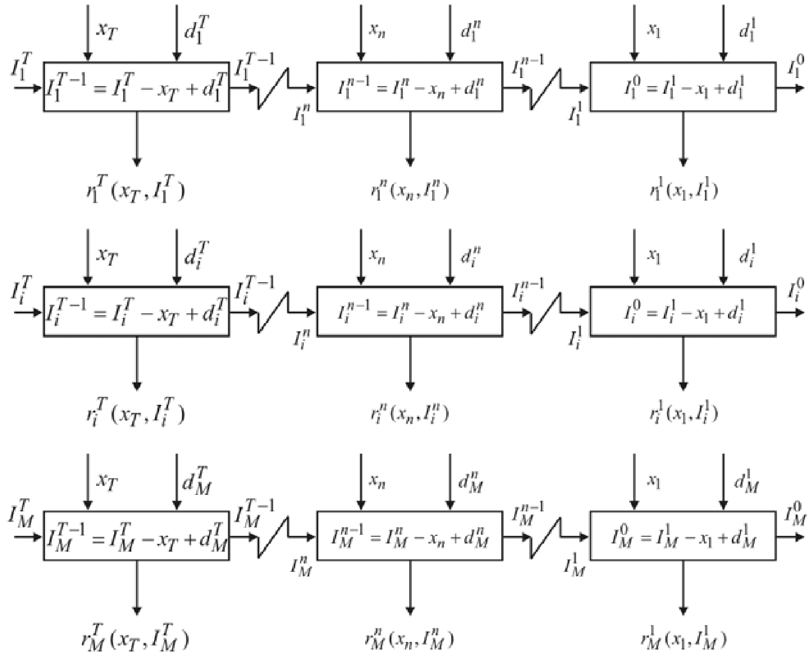

Figure 3.1: The multi-scenario lot-sizing problem scheme.

where $r_i^t(Q_t, I_i^t) = c_i^t(Q_t) + h_i^t(I_i^t)$.

Therefore, the total cost vector $R(\mathbf{Q})$ in all the scenarios for a production/reorder vector $\mathbf{Q} \in \mathbb{N}_0^T$ is as follows

$$R(\mathbf{Q}) = \left(R_1^{1,T}(\mathbf{Q}), \ldots, R_M^{1,T}(\mathbf{Q})\right) \tag{3.3}$$

Then, the Pareto-optimal or non-dominated production/reorder plans set $\mathcal{P}$ can be stated as

$$\begin{aligned} \mathcal{P} \;=\; & \{\mathbf{Q} \in \mathbb{N}_0^T : \text{ there is no other } \mathbf{Q}' \in \mathbb{N}_0^T : R(\mathbf{Q}') \leq R(\mathbf{Q}), \\ & \text{with at least one of the inequalities being strict}\} \end{aligned} \tag{3.4}$$

where $R(\mathbf{Q}') \leq R(\mathbf{Q})$ means that $R_i^{1,T}(\mathbf{Q}') \leq R_i^{1,T}(\mathbf{Q})$ for $i = 1, \ldots, M$.

Using the previous definitions, we can state the Dynamic Multiscenario Lot Size

Problem (DMLSP), or P for short, as follows:

$$
\begin{array}{lll}
(P) & v-\min(R_1^{1,T}(\mathbf{Q}),\ldots,R_M^{1,T}(\mathbf{Q})) & \\
 & s.t.: & \\
 & I_i^0 = I_i^T = 0 & i=1,\ldots,M \\
 & I_i^{j-1} + Q_j - I_i^j = d_i^j & j=1,\ldots,T,\ i=1,\ldots,M \\
 & Q_j \in \mathbb{N}_0 & j=1,\ldots,T \\
 & I_i^j \in \mathbb{N}_0 & j=1,\ldots,T,\ i=1,\ldots,M
\end{array}
\tag{3.5}
$$

where $v-\min$ stands for finding the Pareto-optimal set. Thus, the goal consists of determining the Pareto-optimal solutions with respect to the M objective functions. The first constraint in P forces both the initial and the final inventory level to be zero in all the scenarios. The second constraint set concerns the well-known material balance equation, and hence it states the flow conservation among periods in all the scenarios. The production/reorder quantity must be always a nonnegative integer. Finally, the last set of constraints in P disallows shortages.

As we mentioned in Chapter 2, the single objective version for this problem can be solved using a dynamic programming algorithm, hence it seems reasonable to apply MDP for problem P. Accordingly, let $F(j, I_1^{j-1},\ldots,I_M^{j-1})$ be the set of the reachable non-dominated values, which correspond to production/reorder subplans (subpolicies) from the state $(I_1^{j-1},\ldots,I_M^{j-1})$ in period j. Since there are finitely many nonnegative integers Q_j that satisfy (3.1), the principle of optimality gives rise to the following functional equation

$$
F(j,(I_1^{j-1},\ldots,I_M^{j-1})) = v-\min_{Q_j\in\mathbb{N}_0}\left\{\begin{bmatrix} c_1^j(Q_j) \\ \vdots \\ c_M^j(Q_j)\end{bmatrix} + \begin{bmatrix} h_1^j(I_1^{j-1}+Q_j-d_1^j) \\ \vdots \\ h_M^j(I_M^{j-1}+Q_j-d_M^j)\end{bmatrix}\right. \tag{3.6}
$$

$$
\left.\oplus F(j+1,(I_1^j,\ldots,I_M^j))\right\}
$$

where $A\oplus B = \{a+b : a\in A,\ b\in B\}$ for any two sets A, B.

Therefore, the set of Pareto-optimal production/reorder plans of problem P is given by the policies associated with the vectors in the set $F(1,0,\ldots,0)$, and hence MDP algorithms give a solution for our problem. However, due to the inherent curse of the dimensionality of the MDP approach, we introduce a branch and bound scheme to decrease the running times of the solution method. For this reason, before introducing our procedure, we propose two upper bound sets to be applied in the branch and bound algorithm. Recall that an upper bound set is a set of vectors that are either non-dominated or dominated by at least one efficient point. Accordingly, the first upper bound set concerns the case without shortages and the second one represents the upper bound set for when stockouts are allowed.

In the next section, we propose an initial upper bound set assuming that both the carrying and the production/reorder costs are concave and stockouts are not permitted.

3.3 Case without Shortages

In this section we assume that the cost function $R_i^{j,k}(\mathbf{Q})$ is concave in $\mathbf{Q}$ for $i = 1, \ldots, M$, $j = 1, \ldots, T$ and $k \geq j$. Therefore, the following inequality holds:

$$R_i^{1,T}(\mathbf{Q}+1) - R_i^{1,T}(\mathbf{Q}) \leq R_i^{1,T}(\mathbf{Q}) - R_i^{1,T}(\mathbf{Q}-\mathbf{1}) \tag{3.7}$$

where plan $\mathbf{Q} \pm 1$ differs from plan $\mathbf{Q}$ only in two periods where one unit of production/reorder is added or subtracted. In other words, let j and k be the periods (components) where the plan $\mathbf{Q}$ is to be modified, then $\mathbf{Q}+1$ is equal to $\mathbf{Q}$ except for period j where one more production/reorder unit is added and in period k where one production/reorder unit is subtracted. On the other hand, plan $\mathbf{Q}-1$ is equal to $\mathbf{Q}$ excepting in period j in which one production/reorder unit is subtracted and in period k where one production/reorder unit is added.

Recall that the single objective model given in [97] can be formulated as a network flow problem (see [105]). Moreover, for each partition over the state set, there is always a representative plan fulfilling the ZIO property. Therefore, we can use an O(T^2) algorithm (see [97]) to determine a minimum cost plan via pairwise comparison.

We define now the ZIO property for the multiscenario case as follows: a plan $\mathbf{Q}$ is said to be ZIO for P if

$$Q_j \min\{I_1^{j-1}, \ldots, I_M^{j-1}\} = 0 \text{ for } j = 1, \ldots, T. \tag{3.8}$$

It is worth noting that this modification is the natural extension of the corresponding property in the scalar case. As it will be shown later on, efficient ZIO policies play an important role in the determination of the Pareto set because they represent the set of basic solutions, namely, extreme solutions of P. For the sake of simplicity, we formulate problem P as a multicriteria network flow problem since efficient ZIO plans correspond to acyclic flows in the network as well. Accordingly, assuming non-negative concave costs, the underlying network for this problem, depicted in Figure 3.2, is as follows. Let $G = (V, E)$ be a directed network, where V stands for the set of $n = (T + 2)M + 1$ nodes, and E represents the set of $m = 3MT$ edges. The nodes are classified in: production/reorder node (node 0), demand per scenario nodes nd_s, $s = 1, \dots, M$, and intermediate nodes. The intermediate nodes are organized per layers. Thus, in layer j, there are M nodes denoted by n_s^j $s = 1, \dots, M$, $j = 1, \dots, T + 1$.

There are M arcs from node 0 to each layer. The flow entering these arcs is the same. It can be seen as a single flow that is virtually multiplied M times so that the same amount is directed to each one of the nodes in this layer. These arcs can be considered as a pipeline that at a certain point is transformed into M branches. Each one of these branches receives exactly the same flow that the one that enters through the initial node of the arc. The arc from production/reorder node 0 to layer j is related to the production/reorder variable Q_j in period j. The virtual multiplication of the production/reorder is because the different scenarios do not occur simultaneously in reality. Actually, only one of them is to occur, and we are considering simultaneous (parallel) network flow problems with the same kind of input. The arc from 0 to n_s^j has a cost $c_s^j(\cdot)$, $s = 1, \dots, M$ and $j = 1, \dots, T$.

In addition, there are also arcs from n_s^j to n_s^{j+1} $s = 1, \dots, M$ and $j = 1, \dots, T$. Each arc in this category is an inventory arc associated to the state variable I_s^j and its cost is $h_s^j(\cdot)$. Finally, there are arcs leaving each node n_s^j towards nd_s with flow values d_s^j $s = 1, \dots, M$ and $j = 1, \dots, T$.

We proceed now to show that non-dominated ZIO policies represent the set of extreme solutions of problem P. Previously, let us consider first the explicit representation of the multicriteria node-arc incidence matrix A in which the rows correspond to the M blocks of $T + 2$ constraints of problem P.

	Q_1	Q_2	$\cdots$	Q_T	I_1^1	$\cdots$	I_1^{T-1}	I_1^T	$\cdots$	I_M^1	$\cdots$	I_M^{T-1}	I_M^T
	$(0,1)$	$(0,2)$	$\cdots$	$(0,T)$	$(1,2)$	$\cdots$	$(T-1,T)$	$(T,T+1)$	$\cdots$	$(1,2)$	$\cdots$	$(T-1,T)$	$(T,T+1)$
0	1	1	$\cdots$	1	0	$\cdots$	0	0	$\cdots$	0	$\cdots$	0	0
1	-1	0	$\cdots$	0	1	$\cdots$	0	0	$\cdots$	0	$\cdots$	0	0
2	0	-1	$\cdots$	0	-1	$\cdots$	0	0	$\cdots$	0	$\cdots$	0	0
$\vdots$			$\ddots$			$\ddots$			$\ddots$		$\ddots$		
T	0	0	$\cdots$	-1	0	$\cdots$	-1	1	$\cdots$	0	$\cdots$	0	0
$T+1$	0	0	$\cdots$	0	0	$\cdots$	0	-1	$\cdots$	0	$\cdots$	0	0
$\vdots$			$\ddots$			$\ddots$			$\ddots$		$\ddots$		
0	1	1	$\cdots$	1	0	$\cdots$	0	0	$\cdots$	0	$\cdots$	0	0
1	-1	0	$\cdots$	0	0		0	0		1	$\cdots$	0	0
2	0	-1	$\cdots$	0	0	$\cdots$	0	0		-1	$\cdots$	0	0
$\vdots$			$\ddots$			$\ddots$			$\ddots$		$\ddots$		
T	0	0	$\cdots$	-1	0	$\cdots$	0	0	$\cdots$	0	$\cdots$	-1	1
$T+1$	0	0	$\cdots$	0	0	$\cdots$	0	0	$\cdots$	0	$\cdots$	0	-1

Notice that each block of $T + 2$ rows represents a scenario and the columns are divided in two groups: the first T columns are related to the arcs from the producer

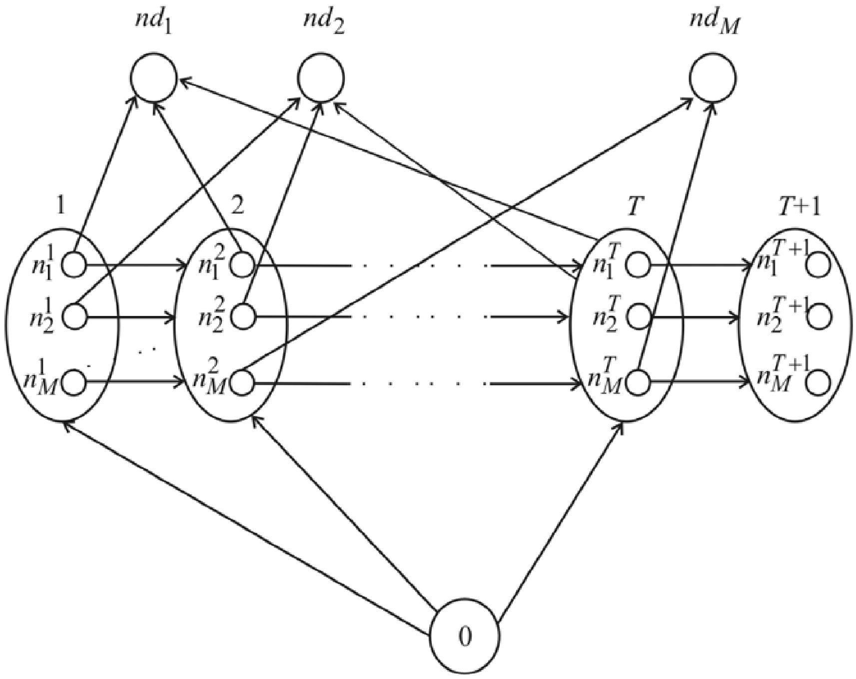

Figure 3.2: The network of problem P.

node to the T periods, and the rest of columns concern the inventory holding between two consecutive periods for each scenario. Using matrix A and vector $\mathbf{Q}$, and denoting the vector $(I_1^1, \ldots, I_1^T, \ldots, I_M^1, \ldots, I_M^T)$ by $\mathbf{I}$, it is straightforward that we get the set of constraints of problem P as follows:

$$(\mathbf{Q}, \mathbf{I})A^t = -(-D, d_1^1, \ldots, d_1^T, 0, \ldots, -D, d_M^1, \ldots, d_M^T, 0).$$

Proposition 14 *The constraint matrix A for problem P has rank $MT + 1$.*

Proof. Indeed, each block of $T+2$ rows has one row (e.g. the last one) being linearly dependent since the sum by blocks equals zero. According to this argument, the rank is, at most, $M(T+1)$. In addition, in the remaining matrix the row corresponding to node 0 appears M times (one per block), hence $(M-1)$ of them could be removed resulting in a matrix with $MT + 1$ rows.

Now, removing the last constraint in each block and using the columns corresponding to $Q_T, I_1^1, \ldots, I_1^T, \ldots, I_M^1 \ldots, I_M^T$, a triangular matrix is obtained with elements in the diagonal equal to one.

	$(0,T)$	$(1,2)$	$\cdots$	$(T-1,T)$	$(T,T+1)$	$\cdots$	$(1,2)$	$\cdots$	$(T-1,T)$	$(T,T+1)$
0	1	0	$\cdots$	0	0	$\cdots$	0	$\cdots$	0	0
1	0	1	$\cdots$	0	0	$\cdots$	0	$\cdots$	0	0
2	0	-1	$\cdots$	0	0	$\cdots$	0	$\cdots$	0	0
$\vdots$			$\ddots$			$\ddots$		$\ddots$		
T	-1	0	$\cdots$	-1	1	$\cdots$	0	$\cdots$	0	0
$\vdots$			$\ddots$			$\ddots$		$\ddots$		
1	0	0		0			1	$\cdots$	0	0
2	0	0	$\cdots$	0			-1	$\cdots$	0	0
$\vdots$			$\ddots$			$\ddots$		$\ddots$		
T	-1	0	$\cdots$	0		$\cdots$	0	$\cdots$	-1	1

(3.9)

Therefore, since a submatrix with rank $MT + 1$ exists the result follows. ■

The following theorem states that the basic solutions for our problem fulfill that the demand in each period is satisfied from either the production/reorder in that period or the units carried in the inventory, but not from both simultaneously. Thus, in the underlying network of the problem, each node (excepting the production/reorder node) is attainable either from the production/reorder node or from the predecessor holding node, but never from both. Hence, the graph associated to the non-null variables of any feasible basic solution verifies for any period j : either $Q_j = 0$ or $\min\{I_1^{j-1}, \ldots, I_M^{j-1}\} = 0$.

Theorem 15 *Any basic solution of problem P fulfills that* $Q_j \min\{I_1^{j-1}, \ldots, I_M^{j-1}\} = 0$ *for any period* j, $j = 1, \ldots, T$.

Proof. Assume without loss of generality that the variables Q_1, Q_2 are non-null. Let us consider the columns that correspond with these variables and the inventory carrying variables from period 1 to 2, i.e. $I_1^1, \ldots, I_M^1$. The matrix has two columns $(0,1)$ and $(0,2)$, for the variables Q_1 and Q_2; and M columns, one per scenario, for the I_s^1 variables $s = 1, \ldots, M$.

$$\begin{bmatrix}
Q_1 & Q_2 & I_1^1 & I_2^1 & \cdots & I_M^1 \\
(0,1) & (0,2) & (1,2) & (1,2) & \cdots & (1,2) \\
+ & - & + & + & \cdots & + \\
1 & 1 & 0 & 0 & \cdots & 0 \\
-1 & 0 & 1 & 0 & \cdots & 0 \\
0 & -1 & -1 & 0 & \cdots & 0 \\
\vdots & & & & & \vdots \\
0 & 0 & 0 & 0 & \cdots & 0 \\
1 & 1 & 0 & 0 & \cdots & 0 \\
-1 & 0 & 0 & 1 & \cdots & 0 \\
0 & -1 & 0 & -1 & \cdots & 0 \\
\vdots & & & & & \vdots \\
0 & 0 & 0 & 0 & \cdots & 0 \\
1 & 1 & 0 & 0 & \cdots & 0 \\
-1 & 0 & 0 & 0 & \cdots & 1 \\
0 & -1 & 0 & 0 & \ldots & -1 \\
 & & & & \cdots & \\
0 & 0 & 0 & 0 & \cdots & 0
\end{bmatrix}$$

It is easy to see that the linear combination of columns with coefficients $+1$, -1, $+1, \cdots, +1$ gives the null vector. Therefore, all the considered variables cannot be part of any basic solution. Hence, the condition holds. ■

For linear cost problems this result implies that there is always a non-dominated ZIO policy. However, for general concave cost problems this result must be proven.

Proposition 16 *The Pareto-optimal solution set of problem P contains, at least, one ZIO policy.*

Proof. By contradiction, assume that all ZIO policies are dominated. Let $\mathbf{Z}$ be a non extreme efficient point such that $\mathbf{Z}$ makes the function $R_i^{1,T}(\cdot)$ minimal. That is, $\mathbf{Z}$ is a plan with cost smaller than or equal to the rest of non-dominated policies in the ith scenario. We can assert that $\mathbf{Z}$ exists, otherwise, the efficient point that minimizes $R_i^{1,T}(\cdot)$ would be an extreme point and the theorem would follow. Furthermore, assume $\mathbf{Q}$ being a feasible extreme point such that the following inequality holds

$$R_i^{1,T}(\mathbf{Z}) < R_i^{1,T}(\mathbf{Q})$$

We can also guarantee that $\mathbf{Q}$ always can be found, otherwise, $R_i^{1,T}(\mathbf{Z}) = R_i^{1,T}(\mathbf{Q})$ for all the extreme points $\mathbf{Q}$, that is, the ith component of the cost vector of $\mathbf{Q}$ is equal to the minimal value for this component and $\mathbf{Z}$ could have been taken an extreme point.

Also, by concavity of the cost functions, the following expression must be fulfilled

$$R_i^{1,T}(\theta\mathbf{Z} + (1-\theta)\mathbf{Q}) \geq \theta R_i^{1,T}(\mathbf{Z}) + (1-\theta)R_i^{1,T}(\mathbf{Q})$$

where θ is a scalar that ranges in [0,1].

In addition, let $\mathbf{P}$ be a point on a facet of the feasible set such that $\mathbf{P}$ is aligned with $\mathbf{Z}$ and $\mathbf{Q}$, and $\mathbf{Z}$ can be expressed as a convex combination of $\mathbf{P}$ and $\mathbf{Q}$. Hence, the following inequality holds

$$R_i^{1,T}(\theta\mathbf{Q} + (1-\theta)\mathbf{P}) \geq \theta R_i^{1,T}(\mathbf{Q}) + (1-\theta)R_i^{1,T}(\mathbf{P})$$

Since $\mathbf{Z}$ is minimal for $R_i^{1,T}(\cdot)$

$$R_i^{1,T}(\mathbf{Z}) \leq R_i^{1,T}(\mathbf{P})$$

Taking $\widehat{\theta}$ such that $\mathbf{Z} = \widehat{\theta}\mathbf{Q} + (1-\widehat{\theta})\mathbf{P}$, the following expression holds

$$R_i^{1,T}(\widehat{\theta}\mathbf{Q} + (1-\widehat{\theta})\mathbf{P}) = R_i^{1,T}(\mathbf{Z}) \geq \widehat{\theta} R_i^{1,T}(\mathbf{Q}) + (1-\widehat{\theta})R_i^{1,T}(\mathbf{P})$$

Notice that $R_i^{1,T}(\mathbf{Z}) < R_i^{1,T}(\mathbf{Q})$ and $R_i^{1,T}(\mathbf{Z}) \leq R_i^{1,T}(\mathbf{P})$, then we have that

$$R_i^{1,T}(\mathbf{Z}) \geq \widehat{\theta} R_i^{1,T}(\mathbf{Q}) + (1-\widehat{\theta})R_i^{1,T}(\mathbf{P}) > \widehat{\theta} R_i^{1,T}(\mathbf{Z}) + (1-\widehat{\theta})R_i^{1,T}(\mathbf{Z}) = R_i^{1,T}(\mathbf{Z})$$

That is, $R_i^{1,T}(\mathbf{Z}) > R_i^{1,T}(\mathbf{Z})$, which is a contradiction. ■

Since we know that there exist Pareto policies satisfying the ZIO property and the procedure in (3.6) that computes the complete Pareto set has a large complexity, we are now interested in determining the Pareto policies within the ZIO plans. This may be considered in some cases as an approximation to the actual Pareto set (indeed, ZIO plans coincide with extreme solutions as Theorem 15 shows). The fact is that the non-dominated ZIO policies represent an initial upper bound set to be used in the branch and bound algorithm.

In order to compute the Pareto ZIO plans, we need to introduce some notation. Let $I(j)$ denote the set of state vectors at the beginning of period j. Notice that $I(0) = I(T+1) = (0, \ldots, 0)$. In addition, let $D_i^{j,k} = \sum_{t=j}^{k-1} d_i^t$ be the accumulated demand from period j to k in scenario i and let $(I_1^{j-1}, \ldots, I_M^{j-1}) \in I(j)$ be a given state vector in period j. Moreover, let us admit that there is a null component in $(I_1^{j-1}, \ldots, I_M^{j-1})$, hence the decision variable Q_j should be distinct to zero to prevent shortages. Thus, the set of feasible decisions corresponding to a state vector $(I_1^{j-1}, \ldots, I_M^{j-1})$ in period j is given by

$$\Psi(j, (I_1^{j-1}, \ldots, I_M^{j-1})) = \begin{cases} 0 & \text{if } I_i^{j-1} > 0 \text{ for all } i, \\ \max_{1 \le i \le M; j+1 \le k \le T+1} \{0, D_i^{j,k} - I_i^{j-1}\} & \text{otherwise.} \end{cases}$$

Assuming that $(I_1^{j-1}, \ldots, I_M^{j-1})$ contains a component equals zero, it can be easily proved that any decision $Q_j \neq \max_{1 \le i \le M} \{0, D_i^{j,j+l} - I_i^{j-1}\}$, $l = 1, \ldots, T+1-j$, results in a non ZIO policy.

Accordingly, given a period j and an inventory vector $(I_1^{j-1}, \ldots, I_M^{j-1}) \in I(j)$, the set $F(j, (I_1^{j-1}, \ldots, I_M^{j-1}))$ of cost vectors corresponding to Pareto ZIO subpolicies for the subproblem with initial inventory vector $(I_1^{j-1}, \ldots, I_M^{j-1})$ is as follows:

$$F(j, (I_1^{j-1}, \ldots, I_M^{j-1})) = \underset{Q_j \in \Psi(j,(I_1^{j-1},\ldots,I_M^{j-1}))}{v-\min} \left\{ \begin{bmatrix} c_1^j(Q_j) \\ \vdots \\ c_M^j(Q_j) \end{bmatrix} + \begin{bmatrix} h_1^j(I_1^{j-1} + Q_j - D_1^{j,j+1}) \\ \vdots \\ h_M^j(I_M^{j-1} + Q_j - D_M^{j,j+1}) \end{bmatrix} \right.$$

$$\left. \oplus F(j+1, (I_1^{j-1} + Q_j - D_1^{j,j+1}, \ldots, I_M^{j-1} + Q_j - D_M^{j,j+1})) \right\} \qquad (3.10)$$

Notice that the whole set of Pareto ZIO policies for P is determined when $F(1, (0, \ldots, 0))$ is achieved.

Proposition 17 *The multicriteria dynamic programming algorithm for problem (3.10) runs in* $O(4^T M^2)$.

Proof. Given an initial inventory vector $(I_1^{j-1}, \ldots, I_M^{j-1}) \in I(j)$, it is clear that Q_j can only take values in $\Psi(j, (I_1^{j-1}, \ldots, I_M^{j-1}))$ to satisfy property (3.8). Thus, if $I_i^{j-1} \neq 0$ for all i, the unique decision is $Q_j = 0$, otherwise, the number of decisions for state $(I_1^{j-1}, \ldots, I_M^{j-1})$ is at most $T - j + 1$. Each different decision leads to a new state vector in the following period, hence the maximum number of states at the beginning of stage $j + 1$ is $T - j + 1$ as well. Remark that the computational effort to make up the accumulated demands matrix $\overline{D}_{M\text{x}T} = \{\overline{d}_{i,j} = D_i^{j,T+1}\}$ is $\mathrm{O}(MT)$, and also $\mathrm{O}(M(T - j) + 1)$ comparisons must be carried out to obtain the maximum values. Hence, the determination of $\Psi(j, (I_1^{j-1}, \ldots, I_M^{j-1}))$ requires of $\mathrm{O}(M(T-j)+1)$ operations.

By virtue of the ZIO property, there are at most two vectors reaching one state in period 2 and, at most, four vectors can achieve any state in period 3. In general, in one state of period j there are at most 2^{j-1} vectors to be evaluated via pairwise comparison. Therefore, the number of comparisons for one state of period j is given by $\mathrm{O}(\frac{2^{j-1}(2^{j-1}-1)}{2}M)$. Accordingly, the number of comparisons in period j is $\mathrm{O}((\frac{2^{j-1}(2^{j-1}-1)}{2}M)(M(T - j) + 1))$. Thus, the procedure carries out $\mathrm{O}(M \sum_{j=2}^{T} 2^{j-2}(2^{j-1} - 1)(M(T - j) + 1))$ comparisons, and hence the complexity is $\mathrm{O}(4^T M^2)$. ■

As Proposition 17 states, the implicit enumeration process of the whole set of efficient ZIO policies for P requires a number of operations which grows exponentially with the input size. This is not a surprising result since the multicriteria network flow problem, which is in general *NP*-hard (Ruhe [78]), can be reduced to the problem we deal with.

From the computational point of view, the algorithm based on (3.10) is inefficient, hence we propose a different approach to obtain an approximated solution set. This method consists of obtaining the optimal solution for each scenario in $\mathrm{O}(T^2)$. Notice that, as a consecuence of disallowing shortages, some of these solutions could be infeasible for problem P. In this case, all the scenarios with infeasible solutions are solved again using a demand vector where each component corresponds to the marginal maximum demand, namely, the jth value in this vector coincides with $(\max\limits_{1 \leq i \leq M} \{D_i^{1,j+1}\} - \max\limits_{1 \leq i \leq M} \{D_i^{1,j}\})$. Remark that the demand vector obtained in this way is a ZIO plan and, hence, is feasible for P. Moreover, the computational effort to determine this set of policies is $\mathrm{O}(MT^2)$. In addition, these plans can also be used as the starting upper bound set of the branch and bound scheme when shortages are not permitted.

We proceed below to analyze the case when both the carrying and the production/reorder costs are concave and shortages are allowed.

3.4 Case with Shortages

This section is devoted to the case in which inventories on hand are not restricted to be nonnegative. When I_i^j is negative, it now represents a shortage of $-I_i^j$ units of unfilled (backlogged) demand that must be satisfied by production/reorder during periods j through T.

We assume, for simplicity, that $h_i^j(I_i^j)$ represents the holding/shortage cost function for period j in scenario i. When I_i^j is nonnegative, $h_i^j(I_i^j)$ remains equal to the cost of having I_i^j units of inventory on hand at the end of period j in scenario i. When I_i^j is negative, $h_i^j(I_i^j)$ becomes the cost of having a shortage of $-I_i^j$ units of unfilled demand on hand at the end of period j in scenario i.

In the single scenario version, there exists at least one period with inventory on hand equal to zero between two consecutive periods with production/reorder different from zero (see [104]). That is, if $Q_j > 0$ and $Q_l > 0$ for $j < l$, then $I^k = 0$ for at least one k so that $j \leq k < l$. This idea is exploited to develop an $\mathrm{O}(T^3)$ algorithm that determines an optimal policy.

Assuming that inventory levels are unconstrained, we can adapt the previous property to the multiscenario case as follows:

$$\text{If } Q_j > 0 \text{ and } Q_l > 0 \text{ for } j < l, \text{ then } I_i^k = 0, \text{ for some } i \text{ and } k,\ j \leq k < l. \quad (3.11)$$

In contrast to the ZIO property for the multiscenario case, the above expression allows us to obtain all the plans satisfying (3.11) independently. In other words, any plan satisfying (3.11) for one scenario is to be feasible for the rest of scenarios, hence a straightforward approach to generate the whole plans set is to determine each set (one per scenario) separately. Again, these plans play a relevant role for obtaining the Pareto set of problem P with stockouts, since, as Theorem 18 shows, they represent the extreme points of the feasible set.

We can use again the network previously introduced to characterize the extreme solutions of P with shortages. Accordingly, the following theorem states that such extreme points represent acyclic policies. That is, demand in a period k is satisfied from the production/reorder either in a previous period ($j \leq k$) or in a successor period ($l \geq k$). Therefore, in the underlying network of the problem, each node (excepting the production/reorder node) is attainable from only one of the following nodes: the production/reorder node, the predecessor holding node or the succesor backlogging node.

Theorem 18 *Any basic solution for problem P with shortages is acyclic*

Proof. Following a similar argument to that in Theorem 15, let us select, for each block (scenario), any two columns corresponding to production/reorder arcs in (3.9), e.g., columns j and l. Moreover, we select, for each scenario, the columns related to periods j up to l. It is easy to see that a linear combination of these columns with coefficients $+1, -1, +1, \ldots, +1$ respectively, gives the null vector. Therefore, any basic solution is acyclic. ■

Proposition 19 *The Pareto-optimal set of problem P with shortages contains, at least, one plan satisfying property (3.11).*

Proof. Similar to that in Proposition 16. ■

Notice that not all the basic plans belong to the Pareto-optimal set and, the solution time required to determine the whole set of non-dominated solutions increases with the input size. Therefore, obtaining the efficient plans among the extreme plans seems to be a reasonable approach, not only as approximation to the real Pareto-optimal set but also as an upper bound set to be used in the branch and bound scheme. Thus, taking into account that the set of feasible decisions verifying (3.11) for one state $(I_1^{j-1}, \ldots, I_M^{j-1}) \in I(j)$ is as follows

$$\Phi(j, (I_1^{j-1}, \ldots, I_M^{j-1})) = \begin{cases} 0 & \text{if } I_i^{j-1} > 0 \text{ for all } i, \\ \{0\} \cup \{-I_i^{j-1} + D_i^{j,k}\}, \ \begin{matrix} k = j+1, \ldots, T+1 \\ i = 1, \ldots, M \end{matrix} & \text{, otherwise.} \end{cases}$$

we can determine the set of non-dominated cost vectors for the state $(I_1^{j-1}, \ldots, I_M^{j-1})$ in period j according to the following functional equation

$$F(j, (I_1^{j-1}, \ldots, I_M^{j-1})) = \underset{Q_j \in \Phi(j,(I_1^{j-1},\ldots,I_M^{j-1}))}{v-\min} \left\{ \begin{bmatrix} c_1^j(Q_j) \\ \vdots \\ c_M^j(Q_j) \end{bmatrix} + \begin{bmatrix} h_1^j(I_1^{j-1} + Q_j - D_1^{j,j+1}) \\ \vdots \\ h_M^j(I_M^{j-1} + Q_j - D_M^{j,j+1}) \end{bmatrix} + \right.$$

$$\left. \oplus F(j+1, (I_1^{j-1} + Q_j - D_1^{j,j+1}, \ldots, I_M^{j-1} + Q_j - D_M^{j,j+1})) \right\} \qquad (3.12)$$

Remark that when $F(1, (0, \ldots, 0))$ is evaluated, the non-dominated solutions set satisfying (3.11) is achieved.

Proposition 20 *The multicriteria dynamic programming algorithm for the problem (3.12) runs in $O(\frac{M(MT+1)^{2T}}{2(MT)^2})$.*

Proof. In period j, Q_j can take values from $\Phi(j,(I_1^{j-1},\dots,I_M^{j-1}))$. Accordingly, the maximum number of states in any period is $M(T-1)+1$. Also, in one state of period j there are, at most, $(MT+1)^{j-1}$ vectors. Therefore, at most, $\frac{M(MT+1)^{j-1}((MT+1)^{j-1}-1)}{2}$ comparisons have to be made. Consequently, the total number of comparisons is $O(M\sum_{j=2}^{T}\frac{(MT+1)^{j-1}((MT+1)^{j-1}-1)}{2})$, and hence the procedure runs in $O(\frac{M(MT+1)^{2T}}{2(MT)^2})$. ■

Since the implementation of the algorithm based on (3.12) involves a number of operations, which increases exponentially with the input size, we propose a different approach to obtain an approximated solution set. This method consists of obtaining the optimal solution for each scenario in $O(T^3)$ using the procedure proposed by Zangwill [104]. In contrast to the case without shortages, all the single scenario solutions are to be feasible for problem P. Therefore, the computational effort to determine the set of optimal solutions for each scenario is $O(MT^3)$, and the non-dominated plans in this set are proposed as the initial upper bound set of the branch and bound scheme when shortages are allowed.

Once the initial upper bound sets for both shortages and not shortages situations have been introduced, we present in the following section the branch and bound scheme, as well as an initial lower bound set to determine the Pareto-optimal set.

3.5 The Solution Method

Before introducing the solution method, we need some additional notation. Let $\mathbf{D}_j \in \mathbb{N}_0^M$ be a vector where each component $i=1,\dots,M$ corresponds to $D_i^{1,j}$ and, also, let $N(j+1,(I_1^j,\dots,I_M^j))$ denote the set of cost vectors associated to subplans that attain the state vector $(I_1^j,\dots,I_M^j)\in I(j+1)$. That is,

$$\begin{aligned} N(j+1,(I_1^j,\dots,I_M^j)) &= \{N(j,(I_1^{j-1},\dots,I_M^{j-1}))\oplus(r_1^j(Q,I_1^j),\dots,r_M^j(Q,I_M^j)) : Q\in\mathbb{N}_0, \\ I_i^{j-1}+Q-D_i^{j,j+1} &= I_i^j, \text{ for all } i \text{ and } (I_1^{j-1},\dots,I_M^{j-1})\in I(j)\} \end{aligned}$$

Since we are interested in calculating the non-dominated policies that reach the state $(0,\dots,0)\in I(T+1)$, we must determine the efficient plans among those in $N(T+1,(0,\dots,0))$ via pairwise comparison. As Villarreal and Karwan [93] pointed out, a necessary condition for a Pareto-optimal point is that it must contain, as its first $n-1$ components, an efficient solution to an $(n-1)$-stage problem, hence the previous process must be applied in all the attainable states. Thus, the efficient subplans should be selected in every attainable state. Therefore, we define $N^*(j+1,(I_1^j,\ldots,I_M^j))$ to be the set of non-dominated subplans that attain the state $(I_1^j,\ldots,I_M^j)$.

Moreover, the interval for the decision variable Q can be calculated according to the following argument: the lot size for the state $(I_1^j,\ldots,I_M^j)$ must be at least equal to zero or $\max\limits_{1\le i\le M}\{0,D_i^{j+1,j+2}-I_i^j\}$, respectively, depending on whether shortages are permitted or not. On the other hand, the upper bound for the interval corresponds to the remaining quantity to reach the total demand, hence Q ranges in $[0,\max\limits_{1\le i\le M}\{0,D_i^{j+1,T+1}-I_i^j\}]$ in case of allowing shortages or in $[\max\limits_{1\le i\le M}\{0,D_i^{j+1,j+2}-I_i^j\},\max\limits_{1\le i\le M}\{0,D_i^{j+1,T+1}-I_i^j\}]$, otherwise. In addition, given a period j, let s be the scenario so that $D_s^{1,j+1}=\max\limits_{1\le i\le M}\{D_i^{1,j+1}\}$. Then, we consider as initial state vector in $I(j)$ either vector $(D_s^{1,j+1}-D_1^{1,j+1},\ldots,D_s^{1,j+1}-D_M^{1,j+1})$, if shortages are not allowed, or vector $(-D_1^{1,j+1},\ldots,-D_M^{1,j+1})$ otherwise. Thus, the rest of vectors in $I(j)$ are obtained just augmenting one unit each component as many times as $D-(D_s^{1,j+1}-D_i^{1,j+1})$ or $D-(-D_i^{1,j+1})$ for any i, respectively.

Taking into account that $I(1)=I(T+1)=(0,\ldots,0)$, we can now outline the Multicriteria Dynamic Programming (MDP) algorithm.

Algorithm 6 Determine the Pareto-optimal set for problem P

Data: matrices d_i^j, c_i^j, h_i^j, numbers M and T, and sets $I(j)$, $j=1,\ldots,T+1$

for $j\leftarrow T$ downto 1 **do**
  for all state $(I_1^j,\ldots,I_M^j)\in I(j+1)$ **do**
    for all state $(I_1^{j-1},\ldots,I_M^{j-1})\in I(j)$ **do**
      if $I_i^j-I_i^{j-1}+d_i^j\ge 0$ and $I_i^j-I_i^{j-1}+d_i^j=I_s^j-I_s^{j-1}+d_s^j$ for $i\neq s$ **then**
        $Q_j=I_i^j-I_i^{j-1}+d_i^j$
        insert Q_j and its cost vector in state $(I_1^{j-1},\ldots,I_M^{j-1})$ and update $N^*(j,(I_1^{j-1},\ldots,I_M^{j-1}))$
      end if
    end for
  end for
end for
return $N^*(1,(0,\ldots,0))$

Example 21 *For the sake of completeness, we present the following numerical example to illustrate the previous results for the case without shortages.*

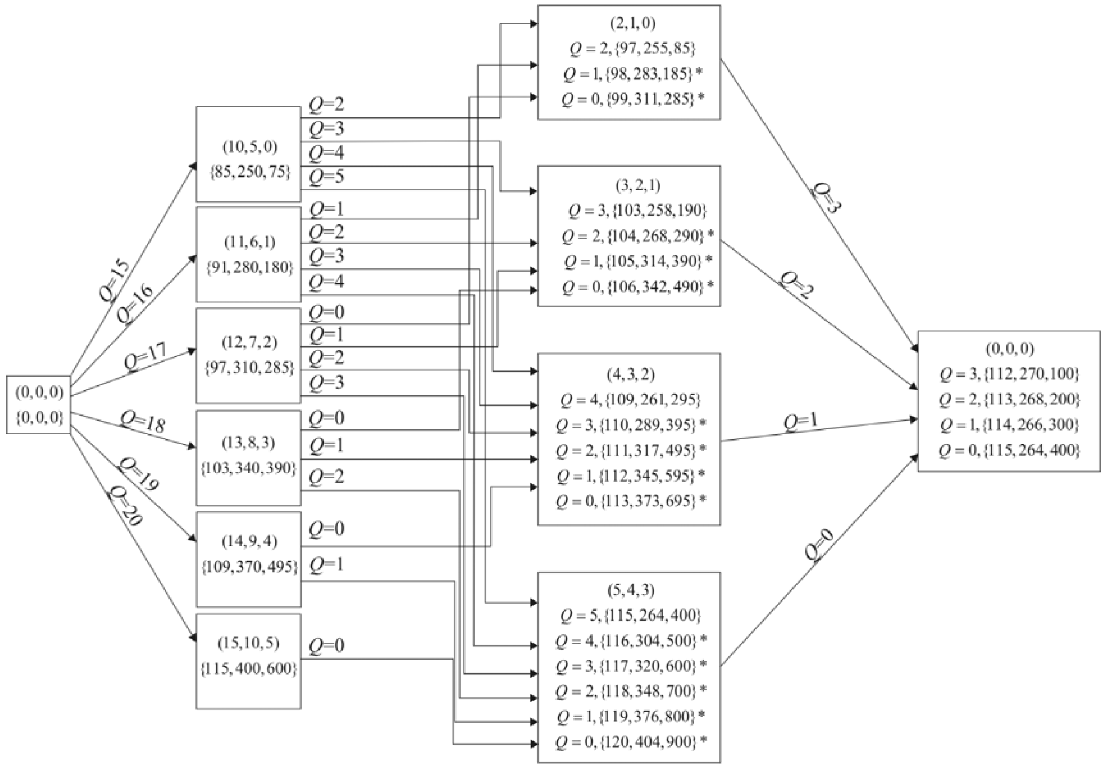

Figure 3.3: The MDP graph of Example 21.

	d_i^j			c_i^j			h_i^j		
	$j=1$	$j=2$	$j=3$	$j=1$	$j=2$	$j=3$	$j=1$	$j=2$	$j=3$
$i=1$	5	10	5	5	5	5	1	1	0
$i=2$	10	6	4	10	2	5	20	1	0
$i=3$	15	2	3	5	5	5	100	100	0

As one can see, all possible plans are collected in the graph depicted in Figure 3.3. In this graph, each node represents one state that is identified by its inventory level vector (in parenthesis). Also, within each node, the partial cost vectors (in brackets) associated to subplans that attain this node are shown. Those subplans which are dominated by any other subplan in the same node are marked with an asterisk. For each node, the leaving arcs (arrows) represent the possible decisions for this node. The right-most node contains the set of non-dominated solutions.

Figure 3.3 illustrates also the case where a non-ZIO plan dominates a ZIO plan, namely, the ZIO plan (17,0,3) with cost vector {114,326,300} is dominated by the non-ZIO plan (15,3,2) with cost vector {113,268,200}.

Since Algorithm 6 becomes intractable as the difference $(D - \max_{1\leq i\leq M} \{d_i^1\})$ increases, a branch and bound approach is proposed. We first focus our attention on the case without shortages. The other case is commented later on. We should reformulate problem P without shortages in a more appropriate way. Accordingly, we denote by $(I_1^n, \ldots, I_M^n) \in I(n+1)$ a state vector at the beginning of period $n+1$, and let $P(n, (I_1^n, \ldots, I_M^n))$ be the set of Pareto-values of the subproblem consisting of periods 1 to n with final inventory vector $(I_1^n, \ldots, I_M^n)$. Therefore, we can now state the problem as follows

$$P(n, (I_1^n, \ldots, I_M^n)) = v\text{-}\min[\sum_{j=1}^{n} c_1^j(Q_j) + \sum_{j=1}^{n-1} h_1^j(\sum_{k=1}^{j} Q_k - D_1^{1,j+1}) + h_1^n(I_1^n), \ldots,$$

$$\sum_{j=1}^{n} c_M^j(Q_j) + \sum_{j=1}^{n-1} h_M^j(\sum_{k=1}^{j} Q_k - D_M^{1,j+1}) + h_M^n(I_M^n)]$$

s.t.:

$$\sum_{j=1}^{k} Q_j \geq D_i^{1,k+1} \qquad k = 1, \ldots, n-1;\ i = 1 \ldots, M$$

$$\sum_{j=1}^{n} Q_j = D_i^{1,n+1} + I_i^n \quad i = 1 \ldots, M$$

It is worth noting that $P(n, (I_1^n, \ldots, I_M^n)) = N^*(n+1, (I_1^n, \ldots, I_M^n))$. Now, one can determine the Pareto values of the complementary problem $\overline{P}(n+1, (I_1^n, \ldots, I_M^n))$, i.e., the problem consisting of periods $n+1$ to T with initial inventory vector $(I_1^n, \ldots, I_M^n)$, as follows

$$\overline{P}(n+1, (I_1^n, \ldots, I_M^n)) = v\text{-}\min[\sum_{j=n+1}^{T} c_1^j(Q_j) + \sum_{j=n+1}^{T-1} h_1^j(I_1^n + \sum_{k=n+1}^{j} Q_k - D_1^{n+1,j+1})$$

$$+$$

$$h_1^T(I_1^n + \sum_{k=n+1}^{T} Q_k - D_1^{n+1,N+1}), \ldots, \sum_{j=n+1}^{T} c_M^j(Q_j) + \sum_{j=n+1}^{T-1} h_M^j(I_1^n + \sum_{k=n+1}^{j} Q_k - D_M^{n+1,j+1})$$

$$+$$

$$h_M^T(I_M^n + \sum_{k=n+1}^{T} Q_k - D_M^{n+1,T+1})]$$

s.t.:

$$\sum_{j=n+1}^{k} Q_j \geq D_i^{n+1,k+1} - I_i^n \quad k = n+1, \ldots, T;\ i = 1\ldots, M$$
$$\sum_{j=n+1}^{T} Q_j = D_i^{n+1,T+1} - I_i^n \quad i = 1\ldots, M$$

Remark that when shortages are allowed, the first set of constraints in both formulations P and $\overline{P}$ should be removed. Again, the optimality principle gives rise to the following recursive equation which provides the Pareto-optimal set for P.

$$F(1,(0,\ldots,0)) = \underset{\substack{(I_1^n,\ldots,I_M^n)\in I(n+1)\\ n=1,\ldots,T-1}}{v-\min} (P(n,(I_1^n,\ldots,I_M^n)) \oplus \overline{P}(n+1,(I_1^n,\ldots,I_M^n)))$$

These equations along with the upper and lower bound sets allow us to introduce the branch and bound scheme into the dynamic programming heap. In particular, we say that LB is a lower bound set for a vector-valued problem when any non-dominated solution either belongs to LB or it is dominated by some vector in LB. In addition, recall that all the vectors in an upper bound UB set are either non-dominated or dominated by at least one efficient point.

Assume that we know both lower bounds $LB(n+1,(I_1^n,\ldots,I_M^n))$ for each sub-problem $\overline{P}(n+1,(I_1^n,\ldots,I_M^n))$ and also global upper bounds UB for the original problem $F(1,(0,\ldots,0))$.

Consider $f \in P(n,(I_1^n,\ldots,I_M^n))$ such that for any $lb \in LB(n+1,(I_1^n,\ldots,I_M^n))$: $f + lb \geq u$ for some $u \in UB$. It is straightforward that the branch generated by f needs not being explored. Indeed, $u \in UB$ and, therefore, there exists $\widehat{f}$ efficient (it may occur that $lb = \widehat{f}$) so that $\widehat{f} \leq u$. Hence, $\widehat{f} \leq f + lb \leq f+$ (any feasible completion). This implies that no completion of f can be efficient.

Once the branch and bound scheme has been outlined, the following step consists of determining how the UB and LB sets are initialized. We set UB to the non-dominated ZIO policies which are obtained in previous sections. On the other hand, different LB sets can be determined depending on the cost functions type. In case of linear costs, we propose two sets. The first concerns with the continuous relaxation of the problem. The second approach consists of determining the optimal policies for each scenario using the Wagelmans et al. algorithm [95] and applying, for each pair of optimal plans, a procedure to calculate the lower envelope.

When the cost functions are concave, we can exploit the notion of linear minorant to obtain an LB set for problem P. Specifically, for a given scenario i, we say that a linear function $L_i(\mathbf{Q})$ is a linear minorant of function $R_i^{1,T}(\mathbf{Q})$ if $L_i^{1,T}(\mathbf{Q}) \leq R_i^{1,T}(\mathbf{Q})$ holds for all feasible plan $\mathbf{Q}$. Accordingly, $L(\mathbf{Q}) = (L_1^{1,T}(\mathbf{Q}),\ldots,L_M^{1,T}(\mathbf{Q}))$

is called a linear minorant vector of vector $(R_1^{1,T}(\mathbf{Q}), \ldots, R_M^{1,T}(\mathbf{Q}))$ if $L_i^{1,T}(\mathbf{Q})$ is a linear minorant of $R_i^{1,T}(\mathbf{Q})$ for any scenario i. We can now formulate the following linear multiobjective (LM) problem

$$
\begin{array}{lll}
LM & v\text{-}\min(L_1^{1,T}(\mathbf{Q}), \ldots, L_M^{1,T}(\mathbf{Q})) & \\
s.t.: & & \\
& I_i^0 = I_i^T = 0 & i = 1, \ldots, M \\
& I_i^{j-1} + Q_j - I_i^j = d_i^j & j = 1, \ldots, T,\ i = 1, \ldots, M \\
& I_i^j \geq 0,\ Q_j \text{ integer} & j = 1, \ldots, T,\ i = 1, \ldots, M
\end{array}
$$

Let $\mathbf{X}$ be a Pareto-optimal solution to *LM*. As Geoffrion [41] proved, $\mathbf{X}$ minimizes a scalarization of LM with suitable positive weights adding up to 1. If $\mathbf{X}$ is the unique optimal solution to such scalarization, then it must be an extreme point of the feasible polyhedron, and thus by Theorem 15 it must satisfy the modified ZIO property. Moreover, the following result shows that obtaining a linear minorant vector reduces to the problem of finding an LB set for the original problem.

Theorem 22 *The Pareto-optimal solution set to LM is an LB set for problem P .*

Proof. Let us denote $LB = L(E(L_1^{1,T}, \ldots, L_M^{1,T}))$ where $E(L_1^{1,T}, \ldots, L_M^{1,T})$ is the set of Pareto-optimal solutions of *LM*. Furthermore, we denote by $E(R_1^{1,T}, \ldots, R_M^{1,T})$ the Pareto-optimal set of the original problem P. Accordingly, if $\mathbf{Q} \in E(R_1^{1,T}, \ldots, R_M^{1,T})$ then either $\mathbf{Q} \in E(L_1^{1,T}, \ldots, L_M^{1,T})$ or $\mathbf{Q} \notin E(L_1^{1,T}, \ldots, L_M^{1,T})$. In the first case, $L(\mathbf{Q}) = (L_1^{1,T}(\mathbf{Q}), \ldots, L_M^{1,T}(\mathbf{Q})) \in LB$ and hence $L(\mathbf{Q}) \leq R(\mathbf{Q})$, where $R(\mathbf{Q})$ was defined in (3.3). In the second case, it must exist $\mathbf{Q}'$ such that $\mathbf{Q}' \in E(L_1^{1,T}, \ldots, L_M^{1,T})$ and $L(\mathbf{Q}') \underset{\neq}{\leq} L(\mathbf{Q})$. Thus, $L(\mathbf{Q}') \in LB$ and $L(\mathbf{Q}') \leq R(\mathbf{Q})$. Therefore, LB is an actual lower bound for problem P. ■

3.6 Computational Experience

This section is divided into two parts. In the first part, the Pareto-optimal set for ten randomly generated problems is reported. On the other hand, the second part is devoted to test the efficiency of the two algorithms, the MDP procedure and the Branch and Bound (BB) approach, as a function of both the number of scenarios and the number of periods.

To simplify the computational experiment, we have chosen the cost functions to be linear and the inventory levels to be non-negative. Taking into account these assumptions, the instances have been solved using the procedure given in the previous section.

In this part, Tables 3.1 and 3.2 show the input data for ten instances and the non-domimated plans with their overall cost vectors, respectively. Table 3.1 is organized as follows: the first column indicates the number of the problem, the rows represent the scenarios and the rest of columns give for the different periods the values for the demand, unit reorder cost and unit holding cost, respectively. Given that the final inventory vector is null, the unit carrying cost for the last period does not affect the optimal solution, and hence it is denoted by x. This computational experience involves instances with two scenarios and four periods up to instances with five scenarios and five periods. In Table 3.2, for each problem, the efficient plans with their respective costs are located in consecutive cells of the same row.

The MDP solution procedure was coded in C++ using LEDA libraries [58]. The main difficulty to implement this code was the storage requirement which increases with the difference $(D - \max_{1\leq i\leq M} \{d_i^1\})$. This difficulty, known as curse of dimensionality, was already discussed by Villarreal and Karwan [93]. These authors argued that as the number of objective functions increases so does the solution time. The instances proposed in Table 3.1 were solved in a workstation HP 9000-712/80. Another interesting aspect of the problem concerns its sensitivity. After several samples, we notice that slight changes in the input data make the Pareto-optimal set to vary drastically.

The BB scheme has been incorporated to the MDP procedure as follows: for each subproblem $\overline{P}(n+1, I_1^n, \ldots, I_M^n)$, the LB set is obtained from calls to the ADBASE code developed by Steuer [85]. This code gives the supported non-dominated solutions for continuous linear multicriteria problems. As a consecuence of both the input to and the output from the ADBASE code is file typed, conversions of the form matrix(C++)-file(ADBASE) and file(ADBASE)-matrix(C++) are required. Moreover, since all the parameters are integer and the constraints matrix is unimodular, the extreme solutions given by ADBASE are integer-valued as well, i.e., feasible for P. Hence, the non-dominated solutions associated to the first subproblem are also considered as the initial UB for the original problem $F(1, (0, \ldots, 0))$.

Now, we provide, in Table 3.3, the average running times for different instances of this problem. For each pair (M, T) ten instances were run. The parameters have been generated according to the following values: the total demand D ranges in the interval [1,1000], the unit carrying and reorder costs vary between 1 and 100. The burden in the computational experience arises as a consequence of the ADBASE limitations. As the number of scenarios or periods increases so does the number of rows and columns in the constraint matrix of the linear multiobjective problem and the problem becomes intractable. Therefore, only some (M, T) combinations can be

		d			c			h		
		1	2	3	1	2	3	1	2	3
P1	Scenario 1	6	3	3	3	7	5	1	2	x
	Scenario 2	7	2	3	2	3	2	6	5	x
P2	Scenario 1	7	4	4	2	7	8	1	1	x
	Scenario 2	3	7	5	3	4	4	1	5	x
	Scenario 3	7	3	5	7	3	4	1	1	x
P3	Scenario 1	6	7	2	2	6	5	1	2	x
	Scenario 2	5	7	3	6	2	1	3	3	x
	Scenario 3	6	6	3	5	4	5	2	4	x
	Scenario 4	7	7	1	1	3	7	4	5	x

		d				c				h			
		1	2	3	4	1	2	3	4	1	2	3	4
P4	Scenario 1	5	7	5	3	5	5	7	5	1	1	1	x
	Scenario 2	7	5	3	5	7	5	5	5	1	1	1	x
P5	Scenario 1	5	6	5	4	1	5	5	3	2	1	1	x
	Scenario 2	4	5	6	5	6	4	2	2	3	3	2	x
	Scenario 3	6	4	4	6	2	1	2	3	5	4	3	x
P6	Scenario 1	3	9	7	5	7	3	5	6	4	1	2	x
	Scenario 2	7	5	6	6	5	4	4	5	4	3	3	x
	Scenario 3	7	5	5	7	7	5	5	2	5	5	4	x
	Scenario 4	8	4	4	8	3	4	5	4	3	3	5	x
P7	Scenario 1	5	2	7	7	6	7	2	3	1	1	2	x
	Scenario 2	10	5	4	2	7	7	6	1	3	1	4	x
	Scenario 3	6	6	4	5	4	4	5	3	4	1	1	x
	Scenario 4	11	3	4	3	2	8	6	7	1	1	2	x
	Scenario 5	9	2	6	4	3	5	7	6	1	2	2	x

		d					c					h				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
P8	Scenario 1	8	2	6	5	4	8	7	5	7	6	4	3	3	1	x
	Scenario 2	5	5	5	5	5	1	6	7	5	6	1	2	2	2	x
	Scenario 3	4	4	5	6	6	2	2	3	2	1	5	6	7	6	x
P9	Scenario 1	9	5	6	2	3	7	5	2	7	6	5	6	1	1	x
	Scenario 2	10	3	5	3	4	8	3	6	4	2	2	1	4	3	x
	Scenario 3	7	4	7	4	3	6	4	5	5	4	4	3	5	2	x
	Scenario 4	8	5	4	3	5	5	6	4	6	5	1	2	7	5	x
P10	Scenario 1	5	3	2	2	3	2	8	6	7	5	2	1	2	1	x
	Scenario 2	7	3	2	1	2	6	3	5	5	2	5	3	2	4	x
	Scenario 3	6	6	1	1	1	5	4	8	6	6	1	1	4	6	x
	Scenario 4	8	1	3	1	2	4	8	7	6	5	4	2	5	3	x
	Scenario 5	5	2	3	3	2	5	4	7	7	6	1	3	3	2	x

Table 3.1: Parameter values for ten randomly generated instances

P1	{7,2,3} (51, 26)	{8,1,3} (48, 31)	{9,0,3} (45, 36)		
P2	{7,4,4} (74, 62, 78)	{8,3,4} (70, 62, 83)	{9,2,4} (66, 62, 88)	{10,1,4} (62, 62, 93)	{11,0,4} (58, 62, 98)
	{12,0,3} (54, 67, 103)	{13,0,2} (50, 72, 108)	{14,0,1} (46, 77, 113)	{15,0,0} (42, 82, 118)	
P3	{7,7,1} (64, 69, 78, 35)	{8,6,1} (61, 76, 81, 37)	{9,5,1} (58, 83, 84, 39)	{10,4,1} (55, 90, 87, 41)	{11,3,1} (52, 97, 90, 43)
	{12,2,1} (49, 104, 93, 45)	{13,1,1} (46, 111, 96, 47)	{14,0,1} (43, 118, 99, 49)		
P4	{7,5,5,3} (112, 116)	{7,6,4,3} (111, 117)	{7,7,3,3} (110, 118)	{7,8,2,3} (109, 119)	{7,9,1,3} (108, 120)
	{7,10,0,3} (107, 121)				
P5	{6,5,5,4} (70, 88, 49)	{7,4,5,4} (68, 93, 55)	{8,3,5,4} (66, 98, 61)	{9,2,5,4} (64, 103, 67)	{10,1,5,4} (62, 108, 73)
	{11,0,5,4} (60, 113, 79)	{12,0,4,4} (59, 123, 88)	{13,0,3,4} (58, 133, 97)	{14,0,2,4} (57, 143, 106)	{15,0,1,4} (56, 153, 115)
	{16,0,0,4} (55, 163, 124)				
P6	{8,4,7,5} (153, 116, 134, 110)	{8,5,6,5} (152, 119, 139, 112)	{8,6,5,5} (151, 122, 144, 114)	{8,7,4,5} (150, 125, 149, 116)	{8,8,3,5} (149, 128, 154, 118)
	{8,9,2,5} (148, 131, 159, 120)	{8,10,1,5} (147, 134, 164, 122)	{8,11,0,5} (146, 137, 169, 124)		
P7	{11,4,4,2} (132, 134, 112, 95, 107)	{12,3,4,2} (132, 137, 116, 90, 106)	{13,2,4,2} (132, 140, 120, 85, 105)	{14,1,4,2} (132, 143, 124, 80, 104)	{15,0,4,2} (132, 146, 128, 75, 103)
	{16,0,3,2} (138, 151, 132, 73, 102)	{17,0,2,2} (144, 156, 136, 71, 101)	{18,0,1,2} (150, 161, 140, 69, 100)	{19,0,0,2} (156, 166, 144, 67, 99)	{20,0,0,1} (163, 180, 151, 66, 101)
	{21,0,0,0} (170, 194, 158, 65, 103)				
P8	{8,2,6,5,4} (167, 118, 117)	{9,1,6,5,4} (172, 114, 122)	{10,0,6,5,4} (177, 110, 127)	{11,0,5,5,4} (187, 107, 137)	{12,0,4,5,4} (197, 104, 147)
	{13,0,3,5,4} 207, 101, 157	{14,0,2,5,4} 217, 98, 167	{15,0,1,5,4} 227, 95, 177	{16,0,0,5,4} 237, 92, 187	
P9	{10,4,6,2,3} (139, 154, 159, 160)	{10,5,5,2,3} (148, 152, 161, 164)	{10,6,4,2,3} (157, 150, 163, 168)	{10,7,3,2,3} (166, 148, 165, 172)	{10,8,2,2,3} (175, 146, 167, 176)
	{10,9,1,2,3} (184, 144, 169, 180)	{10,10,0,2,3} (193, 142, 171, 184)	{10,4,7,1,3} (135, 160, 164, 165)	{10,4,8,0,3} (131, 166, 169, 170)	{10,4,9,0,2} (129, 177, 177, 181)
	{10,4,10,0,1} (127, 188, 185, 192)	{10,4,11,0,0} (125, 199, 193, 203)			
P10	{8,4,1,1,1} (84, 89, 78, 96, 105)	{9,3,1,1,1} (80, 97, 80, 96, 107)	{10,2,1,1,1} (76, 105, 82, 96, 109)	{11,1,1,1,1} (72, 113, 84, 96, 111)	{12,0,1,1,1} (68, 121, 86, 96, 113)
	{8,5,0,1,1} (87, 90, 75, 99, 105)	{9,4,0,1,1} (83, 98, 77, 99, 107)	{10,3,0,1,1} (79, 106, 79, 99, 109)	{11,2,0,1,1} (75, 114, 81, 99, 111)	{12,1,0,1,1} (71, 122, 83, 99, 113)
	{13,0,0,1,1} (67, 130, 85, 99, 115)				

Table 3.2: Pareto-optimal sets for the ten instances in Table 3.1

Scenarios (M)	*Periods (T)*	*Average time (MDP)*	*Average time (BB)*
2	3	7.08	4.98
2	4	8.90	0.66
2	5	24.67	12.80
3	3	19.93	13.25
3	4	11.23	1.24
3	5	2.76	0.63
4	3	10.70	4.65
4	4	15.94	5.90
4	5	22.85	1.46
5	3	20.54	5.00
5	4	76.47	13.15
5	5	17.06	11.28

Table 3.3: Comparison of running times (in sec.)

tested.

Our computational experiments show that the BB scheme outperforms the MDP approach in all cases. The small difference in some instances between the average running times of both procedures is due to each subproblem in the BB calls the ADBASE code. Therefore, the bottleneck of the BB procedure is just the time required to obtain the LB set for each subproblem. In spite of this difficulty, the BB results in CPU times smaller than the MDP method.

3.7 Conclusions

In this chapter, we introduce different algorithms to solve the multiscenario lot size problem considering concave costs. The solution procedures for this case have been implemented using the DMDMP approach and exploiting the dynamic lot size problem's properties. Moreover, a BB procedure has been implemented with a reasonably good behavior in most cases. We are interested in improving this procedure by finding LB sets that are not obtained from external routines, which will decrease much more the running times of the BB versus MDP.

Chapter 4

The Bicriteria I/D Problem

4.1 Introduction

We deal with two-echelon Inventory/Distribution (I/D) systems, where it is appropriate to coordinate the control of different stock keeping units. We look at the case of an item being stocked at two locations with resupply being made between them. In particular, we consider the situation of a two-echelon system consisting of one warehouse and one retailer.

The retailer outlet is replenished from the warehouse which is supplied from an outside supplier. In such a situation, coordinated control makes sense in that, for example, replenishment decisions at the retailer outlet impinge as demand on the warehouse. We consider that the demand at the warehouse is dependent on the deterministic demand (and stocking decisions) of the customers. We refer to this as a dependent demand situation in contrast to classical demands for different stock keeping units, which are considered as being independent. We assume that, at each location, a continuous review (s, Q) is used, and stockouts are not permitted.

The decision involves the choice of a lot size for each facility (warehouse and retailer) which minimizes the inventory cost, that is, the sum of the holding cost plus the ordering cost at both the warehouse and retailer. Determination of the optimal policy for a two-echelon serial I/D system is not obvious, mainly because of the complex interactions between echelons (see Schwarz [80]). However, it is possible to model a multi-echelon system using the concept of echelon stock, first introduced by Clark and Scarf [20]. They defined the echelon stock of echelon j (in a general multi-echelon system) as the number of units in the system that are at, or have passed through, echelon j but have as yet not been specifically committed to outside customers (when backorders are permitted the echelon stock can be negative). With this definition and uniform end-item demand, each echelon stock has a sawtooth pattern with time.

Taking into account the integer-ratio policy proposed by Taha and Skeith [87] (its optimality was proved by Crowston et al. [21] and Williams [100]), it is simple to compute the average value of an echelon stock and the echelon holding costs. This policy tells us that an optimal set of lot sizes exists such that the lot size at each facility is a positive integer multiple of the lot size at its successor facility. This fact was used by Crowston et al. [21] in the development of a dynamic programming approach for determining optimal lot sizes. Some other interesting models about multi-echelon systems are also studied in Silver et al. [82].

Traditional approaches for multi-echelon I/D systems usually have one global objective, cost minimization, typically optimized in an unconstrained manner. Whereas the conventional methods study multi-echelon systems with only one objective, that of inventory cost minimization, in recent years a number of new approaches considering different objectives in multi-echelon systems have been developed. In particular, these objectives involved in inventory management concern the reduction of the inventory cost, the minimization of the transportation cost, the reduction of the expected number of shortages per year (customer service), among others.

A remarkable number of researchers in inventory management have made notable efforts to deal with more than one performance measure or objective. Starr and Miller [83] determined a trade-off between two measures: the number of annual orders (i.e. workload) and the average investment in inventory. They developed the concept of an optimal policy curve, where the points on this curve represent policies between which the decision maker is indifferent. Points off the curve are either infeasible or sub-optimal, but can be improved by moving back to the curve. Gardnet and Dannenbring [39] extended the above concept to a three-dimensional optimal policy surface by adding the performance measure of customer service when they analyse a probabilistic multi-item distribution system. Brown [16] also derived an exchange curve between two performance measures such as workload, investment in inventory or customer service for both deterministic and probabilistic inventory problems. Zeleny [107] discussed Star and Miller´s work in the sense that the optimal policy curve (or surface) is equivalent to non-dominated solutions in Multiple Criteria Decision Making (MCDM). Recently, Puerto and Fernández [71] also analyzed some inventory models from the MCDM perspective using the level curves approach.

Bookbinder and Chen [14] applied the MCDM methodology to a two-echelon serial inventory/distribution system. They discussed different models with deterministic and probabilistic demand, and they assumed that marginal inventory costs were known. Three non-linear multiobjetive programming models and corresponding solution approaches were presented to obtain non-dominated inventory policies achieving trade-offs among objectives such as customer service, inventory investment and transportation cost. Their results were MCDM generalizations of Brown´s exchange curve, Starr and Miller´s optimal policy curve and Gardner and Dannenbring´s optimal policy surface.

In this chapter, we show a new MCDM approach for determining all the admissible lot sizes for a two-echelon inventory/distribution system considering deterministic demand. This problem can be seen as a two-objective non-linear mixed-integer programming model. The first objective consists of minimizing the annual inventory cost, i.e., the sum of the total holding and ordering costs both at the warehouse and at the retailer. The second objective concerns the minimization of the total average number of damaged items by improper shipment handling, which is assumed to be proportional to the number of shipments per year and to the order quantity at the retailer. Thus, as the annual number of shipments increases so does the number of items which could be damaged due to negligence of the personnel handling the items. The minimization of this latter objective is mainly justified when fragile goods are handled. In addition, two constraints are considered: the first concerns the inventory capacity at the retailer and the capacity of the vehicle for delivery, and the second one is related to the restriction of the integer-ratio policy previously commented. We solve with exactitude the problem by finding the complete set of non-dominated policies by means of an exhaustive case analysis of the model.

Notice that the cost structure of the problem under study is similar to that presented in Bookbinder and Chen [14]. Therefore, as it could be expected, their solution approach should give the set of non-dominated solutions for our problem as well. However, as we will prove further on, their solution method for the deterministic case is not correct since it provides no good solutions generating dominated policies.

The rest of the chapter is organized as follows. In Section 4.2 we introduce some notation and we state the model. We continue, in Section 4.3, introducing some preliminary results, which simplify the determination of the Pareto solution set. In Section 4.4, the form of the non-dominated solution set is studied. This form is not unique, depending on the case studied. Several results show us how this set will be with respect to both objectives. In addition, we use our solution method, in Section 4.5, to show that the approach proposed by Bookbinder and Chen to calculate efficient policies is not correct. Some computational results are reported in Section 4.6. We conclude in Section 4.7 with a summary and a brief discussion of implications of the model.

4.2 Notation and Problem Statement

We consider a two-echelon inventory/distribution system where a single item is provided by an outside supplier, stocked at the warehouse and distributed to customers through one retailer.

It is assumed throughout that the demand is known with certainty. Perhaps, this is admittedly an idealization, but it is important to study for two reasons. First, the model may reveal the basic interactions among replenishment quantities at the different echelons. Second, we could choose, where possible, the pragmatic route of developing replenishment strategies based on deterministic demand, and then, conditional on these results, establishing safety stocks to provide appropriate protection against uncertainties.

If there are delays in moving between echelons, the delays are constant and not a function of lot size. No stockouts are permitted in the system.

Let us introduce some preliminary notation. Let Q_r and Q_w denote the variables of the problem, which correspond to the order quantity at the retailer (in units) and the order quantity at the warehouse (in units), respectively. In addition, we present below the parameters of the model.

D	Constant deterministic demand rate, in units/year.
A_r	Fixed ordering cost of a replenishment at the retailer, in money units.
A_w	Fixed ordering cost of a replenishment at the warehouse in money units.
$\alpha(Q_r)$	Number of damaged items per shipment from the warehouse to the retailer, which depends on the order quantity at the retailer.
h_r	Inventory holding cost rate at the retailer, in money/unit/year.
h_w	Inventory holding cost rate at the warehouse, in money/unit/year.
J_r	Inventory capacity at the retailer, in units.
V	Vehicle capacity, in units.
Q_0	Maximum quantity to order at the retailer, in units (i.e., $\min\{J_r, V\}$)
HOC	Sum of the total holding and ordering costs per year.
DI	Total number of damaged items per year.

The goal is the minimization of the criteria so that all the demand is satisfied and no backorders occur. Two general criteria are considered. The first objective (HOC) represents the sum of costs, which are assumed to depend upon the echelon (warehouse or retailer), there being a fixed charge for ordering, and a linear installation inventory holding cost per unit. The second one (DI) represents the annual total number of damaged items which is proportional to the number of shipments from the warehouse to the retailer and to the order quantity at the retailer.

The two controllable (or decision) variables are the replenishment sizes Q_r and Q_w. We have to take into account that the optimality of the integer-ratio policy (the lot size at a given echelon is an integral multiple of the lot size at its successor echelon) was proved for two-echelon systems (see Crowston et al. [21] and Williams [100]). Therefore, we follow this integer-ratio policy and set

$$Q_w = nQ_r$$

where n is a positive integer.

The HOC objective corresponds to the annual inventory cost which depends on the two decision variables Q_r and n. Thus, this cost will be

$$HOC(Q_r, n) = \frac{A_r D}{Q_r} + \frac{A_w D}{nQ_r} + \frac{h_r Q_r}{2} + h_w \frac{(n-1)\,Q_r}{2} \tag{4.1}$$

The DI objective is a function of the variable Q_r, i.e.,

$$DI(Q_r) = \alpha(Q_r)\frac{D}{Q_r} \tag{4.2}$$

Obviously $0 \leq \alpha(Q_r) \leq Q_r \leq D$, and it seems reasonable to think that as Q_r increases so does $\alpha(Q_r)$, however, we assume that the marginal increment of the average number of damaged items per shipment decreases. Hence, it can be easily proved that $\alpha(Q_r)$ is a strictly increasing concave function on $[0, D]$, with $\alpha(0) = 0$. In addition, we assume that $DI(Q_r)$ is a strictly decreasing function.

There are two constraints for the problem. The first concerns the maximum quantity to order at the retailer, which depends on the minimum between the inventory capacity at the retailer and the capacity of the vehicle for delivery. The second constraint restricts n to be a positive integer. Thus, the problem consists of finding Q_r and an integer n that minimize (4.1) and (4.2), subject to $0 < Q_r \leq Q_0$.

It is worth noting that this problem is a two-objective non-linear mixed-integer programming problem. Unfortunately, these kinds of problems are not easy to solve. Continuous multiobjective problems can be solved using scalarization results or constrained parametric optimization which in most times is a tedious task. Integer or combinatorial multiobjective problems are very complex enumeration problems that can be tackled using techniques such as dynamic programming, branch and bound and branch and cut, among others, to obtain a formal approach to the optimal solution set. Non-linear mixed-integer multiobjective problems have all the difficulties inherent in the two former families of problems. In fact, it is not possible to use any of the tools valid either for the continuous or the discrete multiobjective problems and therefore, it is necessary to develop specific approaches for each new problem. In spite of their difficulty, we have found an appropriate way to solve the considered model performing a complete case analysis of the problem and identifying the complete set of non-dominated solutions. These results are presented in the following sections.

4.3 Preliminary Results

As we commented previously, the one warehouse one retailer system fits into a two-objective non-linear mixed integer programming model. To deal with this problem it is appropriate to use the multiple criteria decision making (MCDM) methodology. The goal is to find the set of non-dominated solutions, or Pareto-optimal solution set, of the Bicriteria Biechelon Inventory/Distribution (BBID) problem given by:

$$\begin{array}{lll} BBID: & v-\min & (HOC(Q_r,n), DI(Q_r)) \\ & s.t. & Q_r \in (0, Q_0] \\ & & n \ \in \mathbb{N} \end{array} \tag{4.3}$$

Thus, the Pareto-optimal solution set $\mathcal{P}$ is defined as

$\mathcal{P} = \{(Q_r, n)|$ there is no other $(Q_k^{'}, n^{'}) : HOC(Q_k^{'}, n^{'}) \leq HOC(Q_r, n)$ and $DI(Q_r^{'}) \leq DI(Q_r)$, with at least one of the inequalities being strict$\}$.

Before characterizing the set $\mathcal{P}$, some specific properties of the objective functions are stated. First, it is clear that function $HOC(Q_r, n)$ is convex over the region: $K = \{(Q_r, n) : 0 < n < \infty, 0 < Q_r \leq B(n)\}$, where

$$B(n) = \frac{1}{n}\sqrt{\frac{2A_wD}{h_w}[2\sqrt{1+\frac{A_r}{A_w}n} - 1]}, \tag{4.4}$$

and $HOC(Q_r, n)$ reaches its global minimum at (Q_r^*, n^*), where

$$Q_r^* = \sqrt{\frac{2A_rD}{h_r - h_w}}, \tag{4.5}$$

$$n^* = \sqrt{\frac{(h_r - h_w)A_w}{h_wA_r}}. \tag{4.6}$$

Furthermore, for a fixed n, the value of Q_r which minimizes function $HOC(Q_r, n)$ is given by

$$\overline{Q}_r(n) = \sqrt{\frac{2D(A_r n + A_w)}{n^2 h_w + n(h_r - h_w)}}, \tag{4.7}$$

On the contrary, when Q_r is fixed, the value of n which makes function $HOC(Q_r, n)$ minimal can be obtained by

$$\widehat{n}(Q_r) = \frac{1}{Q_r}\sqrt{\frac{2A_w D}{h_w}} \tag{4.8}$$

Assuming that n and Q_r are real-valued variables, both $\overline{Q}_r(n)$ and $\widehat{n}(Q_r)$ are strictly decreasing convex functions of n and Q_r, respectively. This statement is easily proved since the first derivatives of both functions exist and they are increasing with respect to n and Q_r, respectively. In addition, it can be easily shown that functions $\overline{Q}_r(n)$ and $\widehat{n}(Q_r)$ coincide at the point (Q_r^*, n^*) given by (4.5) and (4.6).

In order to show when function $\overline{Q}_r(n)$ is greater than $\widehat{n}(Q_r)$ or vice-versa, let us define the following expression derived from (4.8).

$$\widehat{Q}_r(n) = \frac{1}{n}\sqrt{\frac{2A_w D}{h_w}} \tag{4.9}$$

Lemma 23 *If $n \geq n^*$, then $\overline{Q}_r(n)$ is greater than or equal to $\widehat{Q}_r(n)$, and the reverse holds when $n < n^*$.*

Proof. If $n \geq n^*$, then $n^2 \geq \frac{(h_r - h_w)A_w}{h_w A_r}$ and, hence $2Dh_w A_r n^2 \geq 2D(h_r - h_w)A_w$. Thus, adding $2DA_w n h_w$ and multiplying by n both terms of the previous expression, we obtain that

$$\frac{2D(A_r n + A_w)}{n^2 h_w + n(h_r - h_w)} \geq \frac{2A_w D}{n^2 h_w}$$

or, in other words, $\overline{Q}_r(n) \geq \widehat{Q}_r(n)$ (see Figure 4.1).

Otherwise, if $n < n^*$, it is clear that $\overline{Q}_r(n) < \widehat{Q}_r(n)$. ■

Lemma 24 *For a fixed n, $n > 1$, the functions $HOC(Q_r, n)$ and $HOC(Q_r, n-j)$, $1 \leq j \leq n-1$, intercept in a unique value $Q_r^{n,n-j} = \sqrt{\frac{2A_w D}{h_w n(n-j)}}$. Besides, $\widehat{Q}_r(n) < Q_r^{n,n-j} < \widehat{Q}_r(n-j)$.*

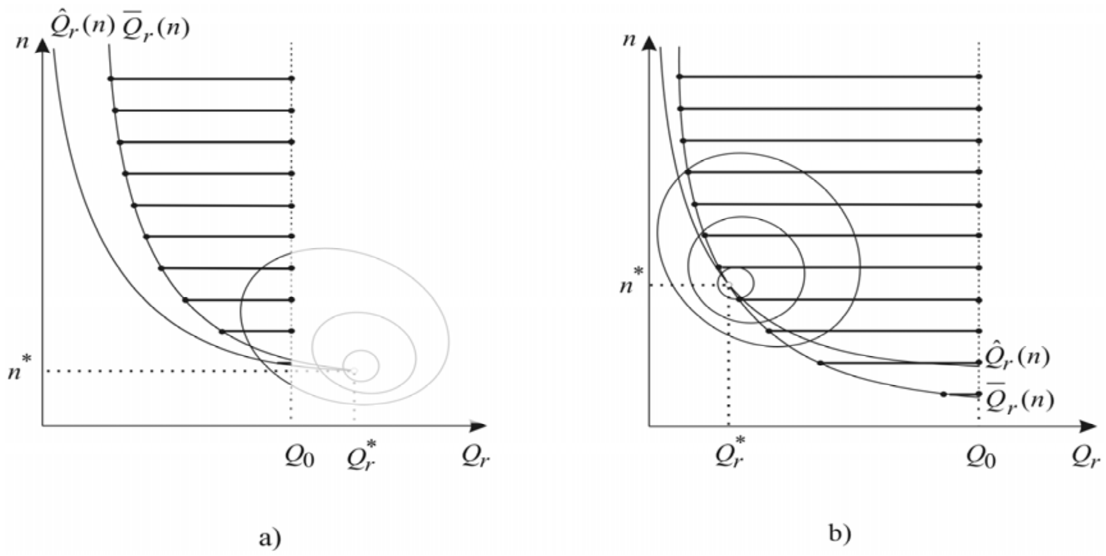

Figure 4.1: Illustration of $\overline{Q}_r(n)$, $\widehat{Q}_r(n)$ and some level curves in $\mathcal{F}$: a) situation when $Q_0 \leq Q_r^*$, and b) when $Q_0 > Q_r^*$.

Proof. Just computing $HOC(Q_r, n) = HOC(Q_r, n-j)$ and, taking into account that $\widehat{Q}_r(n) = \sqrt{\frac{2A_wD}{n^2h_w}} < Q_r^{n,n-j}$ and $\widehat{Q}_r(n-j) = \sqrt{\frac{2A_wD}{(n-j)^2h_w}} > Q_r^{n,n-j}$, the result follows. ■

Characterizing the non-dominated solutions set $\mathcal{P}$ requires to consider the level curves of function $HOC(Q_r, n)$. Accordingly, let us denote the family $\mathcal{F}$ of level curves by

$$\mathcal{F} = \{\varphi_l(Q_r, n) = 0 : \varphi_l(Q_r, n) = (h_r + h_w(n-1))nQ_r^2 - 2lnQ_r + 2D(A_w + A_rn), \\ l \geq HOC(Q_r^*, n^*)\}.$$

Notice that these curves are the level curves of $HOC(Q_r, n)$, i.e., they are sets of the form $\{(Q_r, n) \in \mathbb{R}^2 : HOC(Q_r, n) = l\}$. Since $HOC(Q_r, n)$ is convex in K, set $\varphi_l(Q_r, n) \leq 0$ corresponds to a convex set for any value $l \geq HOC(Q_r^*, n^*)$ (see Figure 4.1).

4.4 Characterizing Pareto-optimal Solutions

We start this section discarding those points in $\mathbb{R}^2$ which are not to be included in $\mathcal{P}$ with certainty. The following lemmas reduce the admissible set of candidate points to be Pareto solutions to those that belong to a given region.

Lemma 25 *The non-dominated solution set $\mathcal{P}$ is included inside region R, characterized by*

$$R = \{(Q_r, n) : \overline{Q}_r(n) \leq Q_r \leq Q_0, \; n \in \mathbb{N}\} \tag{4.10}$$

Proof. By contradiction, let us assume that point (Q_r, n) is a feasible solution which is not in R, i.e., $Q_r < \min\{Q_0, \overline{Q}_r(n)\}$. Then (Q_r, n) is dominated by $(\min\{Q_0, \overline{Q}_r(n)\}, n)$ since both criteria would be improved by convexity of HOC and because DI is strictly decreasing with Q_r. ■

Since the characterization of the Pareto solution set depends on the relative positions of Q^* and Q_0, we should distinguish two possible cases, namely, if $Q_0 \leq Q^*$ or the reverse. The candidate points to be Pareto-optimal solutions are plotted as bold lines in Figure 4.1.

4.4.1 When $Q_0 \leq Q^*$

We need to introduce the following notation before characterizing the non-dominated solution set. Let $\widehat{n}_0$ denote the integer value of $\widehat{n}(Q_0)$ where function $HOC(Q_0, n)$ reaches its minimum, i.e., $\widehat{n}_0 = \arg\{\min_{n \in \{\lfloor \widehat{n}(Q_0) \rfloor, \lceil \widehat{n}(Q_0) \rceil\}} HOC(Q_0, n)\}$, where $\lfloor \widehat{n}(Q_0) \rfloor$ and $\lceil \widehat{n}(Q_0) \rceil$ represents, respectively, the floor and the ceiling of $\widehat{n}(Q_0)$. In case of $HOC(Q_0, \lfloor \widehat{n}(Q_0) \rfloor) = HOC(Q_0, \lceil \widehat{n}(Q_0) \rceil)$, then we set $\widehat{n}_0 = \lfloor \widehat{n}(Q_0) \rfloor$. Furthermore, assuming that $\overline{n}(Q_0)$ stands for the value such that $\overline{Q}_r(n_0) = Q_0$, let $\overline{n}_0$ be the closest integer value greater than $\overline{n}(Q_0)$. Since $Q_0 \leq Q^*$, by virtue of Lemma 23, it is clear that $\overline{n}_0 \geq \lceil \widehat{n}(Q_0) \rceil$ and hence $\overline{n}_0 \geq \widehat{n}_0$.

Lemma 26 *Those points (Q_r, n) in R with $n > \overline{n}_0$ or $n < \widehat{n}_0$ are not included in $\mathcal{P}$.*

Proof. By contradiction, we assume that (Q_r, n) is an efficient point with $n > \overline{n}_0$. Let (Q_0, n_1) be the point where the straigth line joining points (Q_r, n) and (Q_r^*, n^*) intercepts with line Q_0 (see Figure 4.2 a)). Since function HOC is convex, $HOC(Q_0, n_1)$ is smaller that $HOC(Q_r, n)$ and, also, $DI(Q_r) > DI(Q_0)$ because $Q_r < Q_0$. Therefore, (Q_r, n) is dominated by (Q_0, n_1). Moreover, by Lemma 23 and by convexity of function $\widehat{Q}_r(n)$, point (Q_0, n_1) is even dominated by $(Q_0, \overline{n}_0)$. Therefore, point (Q_r, n) cannot be an efficient point.

Following a similar argument, it can be shown that any point (Q_r, n) with $n < \widehat{n}_0$ is dominated by point $(Q_0, \widehat{n}_0)$. ■

We can now use the level curves of function HOC introduced before to simplify the characterization of set $\mathcal{P}$. Accordingly, let q_l^i denote the greatest value of Q_r where curve $\varphi_l(Q_r, n) = 0$ intercepts with line $n = i$. In particular, let $q_{l_0}^{\overline{n}_0}$ be the greatest value, if exists, on the straight line $\overline{n}_0$ with $l_0 = HOC(Q_0, \widehat{n}_0)$, i.e., the greatest value such that $HOC(q_{l_0}^{\overline{n}_0}, \overline{n}_0) = HOC(Q_0, \widehat{n}_0)$. Additionally, let $q_{l_0}^{\widehat{n}_0+1}$ denote the greatest value, if exists, on the straight line $\widehat{n}_0 + 1$ such that $HOC(q_{l_0}^{\widehat{n}_0+1}, \widehat{n}_0 + 1) = HOC(Q_0, \widehat{n}_0)$. The following theorem uses these values to identify the non-dominated solutions set in the case of $Q_0 \leq Q_r^*$.

Theorem 27 *When $Q_0 \leq Q_r^*$, the Pareto solutions set $\mathcal{P}$, assuming that $l_0 = HOC(Q_0, \widehat{n}_0)$, is given as follows*

1) if $\overline{n}_0 = \widehat{n}_0$, : $\mathcal{P} = \{(Q_r, \overline{n}_0) : Q_r \in [\overline{Q}_r(\overline{n}_0), Q_0]\}$

2) if $\overline{n}_0 = \widehat{n}_0 + 1$,
   - a) if $\overline{Q}_r(\overline{n}_0) \leq q_{l_0}^{\overline{n}_0} \leq Q_0$: $\mathcal{P} = \{(Q_r, \overline{n}_0) : Q_r \in [\overline{Q}_r(\overline{n}_0), q_{l_0}^{\overline{n}_0})\} \cup \{(Q_0, \widehat{n}_0)\}$
   - b) if $q_{l_0}^{\overline{n}_0} > Q_0$: $\mathcal{P} = \{(Q_r, \overline{n}_0) : Q_r \in [\overline{Q}_r(\overline{n}_0), Q_0]\}$
   - c) otherwise : $\mathcal{P} = \{(Q_0, \widehat{n}_0)\}$

3) if $\overline{n}_0 > \widehat{n}_0 + 1$,
   - a) if $q_{l_0}^{\widehat{n}_0+1} = Q_0$: $\mathcal{P} = \{(Q_0, \widehat{n}_0 + 1), (Q_0, \widehat{n}_0)\}$
   - b) otherwise : $\mathcal{P} = \{(Q_0, \widehat{n}_0)\}$

Proof. By virtue of Lemma 26, the candidate points to be Pareto solutions are of the form (Q_r, n) with $\widehat{n}_0 \leq n \leq \overline{n}_0$. In particular, when $\overline{n}_0 = \widehat{n}_0$ or $\overline{n}_0 = \widehat{n}_0 + 1$, these points lie on the line $\overline{n}_0$ from $\overline{Q}_r(\overline{n}_0)$ to the value corresponding to $\min\{q_{l_0}^{\overline{n}_0}, Q_0\}$ and, also point $(Q_0, \widehat{n}_0)$, which is represented by the largest black dot in Figures 4.2 a) and b). Moreover, the fact that $q_{l_0}^{\overline{n}_0}$ does not exist implies that there is not any point on $\overline{n}_0$ with HOC cost equal to $l_0 = HOC(Q_0, \widehat{n}_0)$. Indeed, what this result indicates is that the HOC value of any point on line $\overline{n}_0$ is greater than $HOC(Q_0, \widehat{n}_0)$, since any point on this line can be seen as the interception point between $\varphi_l(Q_r, n) = 0$ and $\overline{n}_0$, with $l > HOC(Q_0, \widehat{n}_0)$. Therefore, when either $q_{l_0}^{\overline{n}_0}$ does not exist or $q_{l_0}^{\overline{n}_0}$ is Q_0, the Pareto solution set is of the form: $\mathcal{P} = \{(Q_0, \widehat{n}_0)\}$. On the contrary, if $q_{l_0}^{\overline{n}_0}$ exists, two cases can arise, namely, $\overline{Q}_r(\overline{n}_0) \leq q_{l_0}^{\overline{n}_0} \leq Q_0$ or $Q_0 < q_{l_0}^{\overline{n}_0}$. Notice that, by Lemma 25, the case $\overline{Q}_r(\overline{n}_0) > q_{l_0}^{\overline{n}_0}$ leads to non-dominated solutions.

Thus, when $\overline{Q}_r(\overline{n}_0) \leq q_{l_0}^{\overline{n}_0} \leq Q_0$, there exists a point $(q_{l_0}^{\overline{n}_0}, \overline{n}_0)$, depicted as a diamond in Figure 4.2 b), with the same value of HOC cost than point $(Q_0, \widehat{n}_0)$ but with worst value for the second criterion. Hence, $(q_{l_0}^{\overline{n}_0}, \overline{n}_0)$ is dominated by $(Q_0, \widehat{n}_0)$. In addition, since function $\overline{Q}_r(n)$ reachs its minimum at $(\overline{Q}_r(\overline{n}_0), \overline{n}_0)$ when $n = \overline{n}_0$, all the points in $[\overline{Q}_r(\overline{n}_0), q_{l_0}^{\overline{n}_0})$ have smaller HOC value than point $(q_{l_0}^{\overline{n}_0}, \overline{n}_0)$ and,

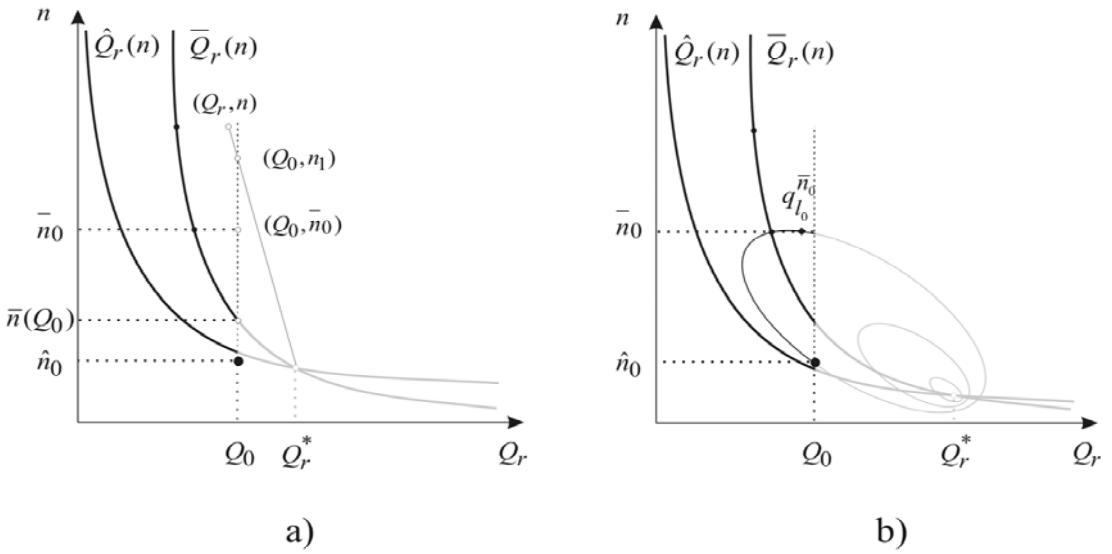

Figure 4.2: a) Illustration of Lemma 26, and b) Illustration of Theorem 27 when $\overline{n}_0 = \widehat{n}_0 + 1$.

therefore, they are non-dominated solutions. Accordingly, the Pareto solutions set is as follows: $\mathcal{P} = \{(Q_r, \overline{n}_0) : Q_r \in [\overline{Q}_r(\overline{n}_0), q_{l_0}^{\overline{n}_0})\} \cup (Q_0, \widehat{n}_0)$.

On the contrary, if $\overline{Q}_r(\overline{n}_0) < Q_0 < q_{l_0}^{\overline{n}_0}$, we can also exploit the fact that $\overline{Q}_r(n)$ is strictly convex to guarantee that points $[\overline{Q}_r(\overline{n}_0), Q_0]$ on line $\overline{n}_0$ have smaller HOC values than $(Q_0, \widehat{n}_0)$ and, hence, $\mathcal{P} = \{(Q_r, \overline{n}_0) : Q_r \in [\overline{Q}_r(\overline{n}_0), Q_0]\}$.

When $\overline{n}_0 > \widehat{n}_0 + 1$, the unique non-dominated solution is point $(Q_0, \widehat{n}_0)$ unless $q_{l_0}^{\widehat{n}_0+1} = Q_0$, in such a case, the Pareto solution set contains points $(Q_0, \widehat{n}_0 + 1)$ and $(Q_0, \widehat{n}_0)$. ■

4.4.2 When $Q_0 > Q_r^*$

From now on, let $\overline{n}$ denote the closest integer value to n^* which minimizes $HOC(\overline{Q}_r(n), n)$, that is, $\overline{n} = \arg\{\min_{n \in \{\lfloor n^* \rfloor, \lceil n^* \rceil\}} HOC(\overline{Q}_r(n), n)\}$, with $\overline{Q}_r(n) \leq Q_0$, and where $\lfloor n^* \rfloor$ stands for the closest integer value smaller than n^*, and $\lceil n^* \rceil$ is the closest integer value greater than n^*. In case of $HOC(\overline{Q}_r(\lfloor n^* \rfloor), \lfloor n^* \rfloor) = HOC(\overline{Q}_r(\lceil n^* \rceil), \lceil n^* \rceil)$, we set $\overline{n} = \lfloor n^* \rfloor$ since, by convexity of $\overline{Q}_r(n)$, point $(\overline{Q}_r(\lfloor n^* \rfloor), \lfloor n^* \rfloor)$ is on the right of $(\overline{Q}_r(\lceil n^* \rceil), \lceil n^* \rceil)$ and, therefore, the second criterion is improved. Observe that, from definition of $\overline{n}$, $\overline{n} \geq \widehat{n}_0$ when $Q_0 > Q_r^*$.

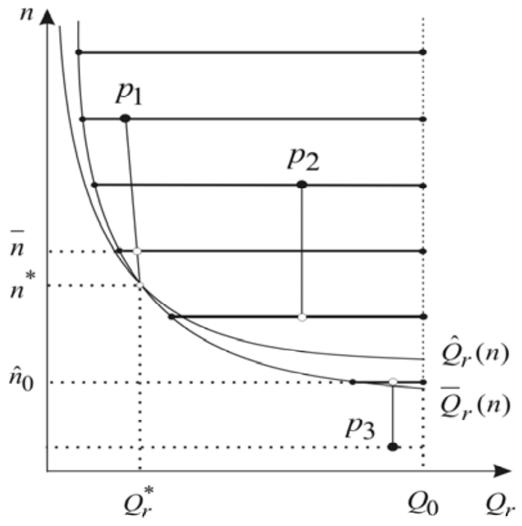

Figure 4.3: Illustration of Lemma 28.

The admissible set of candidate points to be Pareto solutions can be more specifically charaterized, according to the following lemmas.

Lemma 28 *When $Q_0 > Q_r^*$, those points (Q_r, n) in R with $n < \widehat{n}_0$ or $n > \overline{n}$ are not to be included in $\mathcal{P}$.*

Proof. Let (Q_r, n) be an efficient point with $n > \overline{n}$. To show, by contradiction, that (Q_r, n) cannot be a non-dominated point we should distinguish two cases, namely, when $Q_r \leq Q_r^*$ and $Q_r > Q_r^*$. We first focus our attention on the case $Q_r \leq Q_r^*$, accordingly, let p_1 denote point (Q_r, n). Since function HOC is strictly convex, point p_1 is dominated by that point corresponding to the interception point (plotted by a white dot in Figure 4.3) of the segment line joining points p_1 and (Q_r^*, n) with straight line $\overline{n}$. On the other hand, when $Q_r > Q_r^*$, let p_2 denote point (Q_r, n). In this case, since function $\widehat{Q}_r(n)$ provides the point with minimum HOC cost for a fixed Q_r, it is easy to see that point p_2 is dominated by point $(Q_r, \lceil \widehat{n}(Q_r) \rceil)$ depicted by a white dot in Figure 4.3. Therefore, in both cases, point (Q_r, n) is dominated.

Moreover, applying the same reasoning, namely, that function $\widehat{Q}_r(n)$ provides the point with minimum HOC cost for a fixed Q_r, it can be easily shown that any point $p_3 = (Q_r, n)$ with $n < \widehat{n}_0$ is dominated by point $(Q_r, \widehat{n}_0)$. ■

As a result of Lemma 28, the maximum number of intervals containing non-dominated solutions is $k = \overline{n} - \widehat{n}_0 + 1$.

We show below that the Pareto solution set $\mathcal{P}$ consists of union of intervals, which are located on diferent lines n, with $\widehat{n}_0 \leq n \leq \overline{n}$. In what follows, we denote by $\mathcal{P}(n)$ the set of non-dominated points on line n. Therefore, the Pareto solutions set is given by $\mathcal{P} = \bigcup_{n=\widehat{n}_0}^{\overline{n}} \mathcal{P}(n)$. Previously, we need to show that $HOC(\overline{Q}_r(\overline{n}), \overline{n})$, $HOC(\overline{Q}_r(\overline{n}-1), \overline{n}-1), \ldots,$ $HOC(\overline{Q}_r(\widehat{n}_0), \widehat{n}_0)$ represent a sequence of increasing values.

Proposition 29 *For all n with $\widehat{n}_0 < n \leq \overline{n}$, it holds that $HOC(\overline{Q}_r(n), n) < HOC(\overline{Q}_r(n-1), n-1)$*

Proof. Without loss of generality, consider values n^*, $\overline{n}$ and $\overline{n}-1$. By contradiction, let us admit that $HOC(\overline{Q}_r(\overline{n}), \overline{n}) \geq HOC(\overline{Q}_r(\overline{n}-1), \overline{n}-1)$. Since (Q_r^*, n^*) represents the point where function HOC reachs the minimum, it holds that $HOC(\overline{Q}_r(n^*) = Q_r^*, n^*) < HOC(\overline{Q}_r(\overline{n}), \overline{n})$. Therefore, there should be two interception points $a = (Q^1, \overline{n})$ and $b = (Q^2, \overline{n})$ on $\overline{n}$, with $Q^1 < Q^2$ and $l = HOC(\overline{Q}_r(\overline{n}-1), \overline{n}-1)$ (see Figure 4.4). Accordingly, the HOC value in points a and b on $\overline{n}$ coincides with $HOC(\overline{Q}_r(\overline{n}-1), \overline{n}-1)$, so points a, b and $(\overline{Q}_r(\overline{n}-1), \overline{n}-1)$ are included in the same level curve $\varphi_l(Q_r, n) = 0$. However, this result contradicts the fact that $(\overline{Q}_r(\overline{n}), \overline{n})$ is the unique point that minimizes function HOC for $\overline{n}$, and hence, inequality $HOC(\overline{Q}_r(\overline{n}), \overline{n}) \geq HOC(\overline{Q}_r(\overline{n}-1), \overline{n}-1)$ is not feasible. ■

According to Proposition 29 and taking into account that $\overline{Q}_r(n) < \overline{Q}_r(n-1)$ since $\overline{Q}_r(n)$ is a strictly decreasing function, the only two combinations of HOC values for consecutive values of n are depicted in Figure 4.5. Therefore, it is clear that the Pareto set is updated adding a new interval on line $n-1$, which starts from point $\max\{Q_r^{n,n-1}, \overline{Q}_r(n-1)\}$ for $\widehat{n}_0 < n \leq \overline{n}$. The following corollary sheds light on the determination of efficient solutions when only two consecutive values of n are considered.

Corollary 30 *Given lines n and $n-1$, the sets of non-dominated solutions $\mathcal{P}(n)$ and $\mathcal{P}(n-1)$ on these lines are given as follows:*

1.- If $\overline{Q}_r(n-1) = \max\{Q_r^{n,n-1}, \overline{Q}_r(n-1)\}$ then

$\mathcal{P}(n) = [\overline{Q}_r(n), q_l^n)$, with $l = HOC(\overline{Q}_r(n-1), n-1)$, and $\mathcal{P}(n-1) = [\overline{Q}_r(n-1), a_1]$

2.- If $Q_r^{n,n-1} = \max\{Q_r^{n,n-1}, \overline{Q}_r(n-1)\}$ then

$\mathcal{P}(n) = [\overline{Q}_r(n), Q_r^{n,n-1})$, and $\mathcal{P}(n-1) = [Q_r^{n,n-1}, b_1]$

where values a_1 and b_1 depend on the interception points with curve $HOC(Q_r, n-2)$.

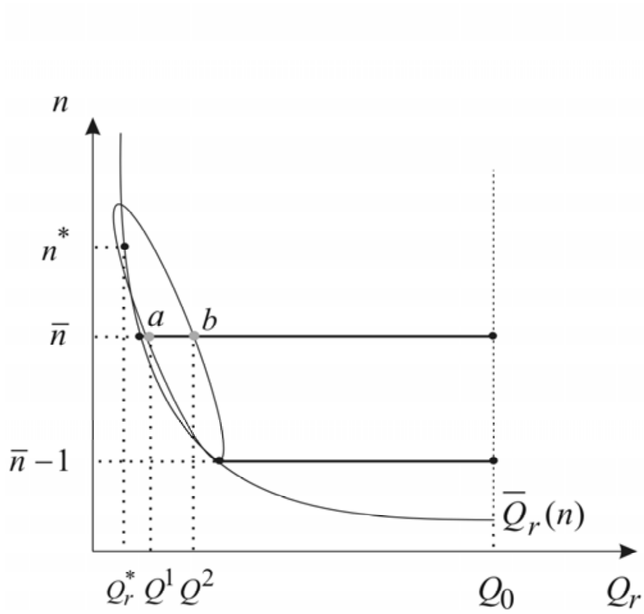

Figure 4.4: Illustration of Proposition 29.

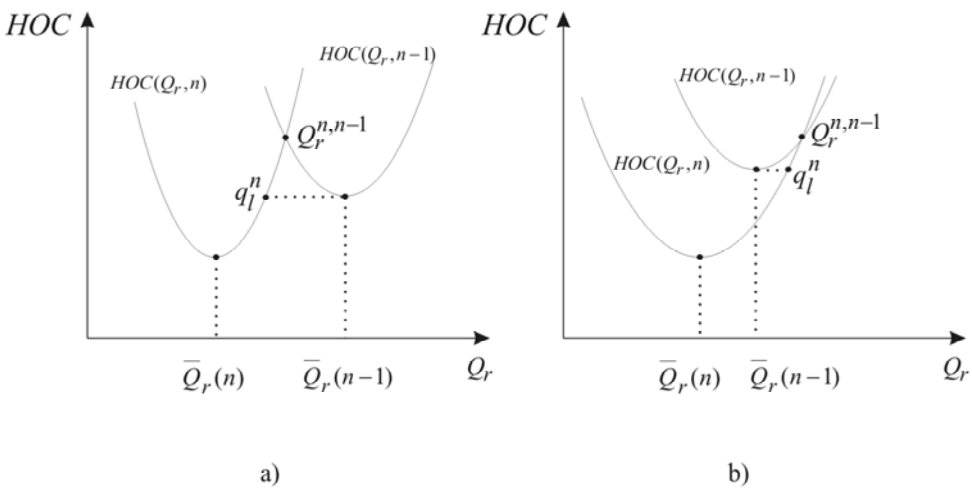

Figure 4.5: Feasible cases when two HOC functions corresponding to two consecutive values of n are faced.

The following result shows that the determination of $\mathcal{P}$ is reduced to successively evaluate functions HOC corresponding to two consecutive values of n.

Theorem 31 *The Pareto-optimal solution set can be computed via pairwise comparison of functions HOC corresponding to consecutive values of n*

Proof. Assume that the efficient points related to lines n and $n-1$ have been already determined. In accordance with Corollary 30, only two cases are feasible, namely, **1** and **2**. Moreover, consider that we analyze line $n-2$ in the process of detetermination of $\mathcal{P}$. Hence, the possible relationships between curves $HOC(Q_r, n)$, $HOC(Q_r, n-1)$ and $HOC(Q_r, n-2)$ are of the form **1-1**, **1-2**, **2-1**, **2-2** (see Figure 4.6). We proceed to evaluate each combination separately:

a) When the pairs of curves $HOC(Q_r, n)$, $HOC(Q_r, n-1)$ and $HOC(Q_r, n-1)$, $HOC(Q_r, n-2)$ are both of the form **1**, it is easily proved that $\mathcal{P}(n) = [\overline{Q}_r(n), q_l^n)$, with $l = HOC(\overline{Q}_r(n-1), n-1)$, $\mathcal{P}(n-1) = [\overline{Q}_r(n-1), q_{l'}^{n-1}]$, with $l' = HOC(\overline{Q}_r(n-2), n-2)$ and the interval associated to $\mathcal{P}(n-2)$ begins at point $\overline{Q}_r(n-2)$ (see Figure 4.6 a)). Therefore, case **1-1** is reduced to independently analyze two cases of the form **1** since adding curve $HOC(Q_r, n-2)$ does not alter the efficient solutions corresponding to curves $HOC(Q_r, n)$ and $HOC(Q_r, n-1)$.

b) When the combination of curves $HOC(Q_r, n)$ and $HOC(Q_r, n-1)$ is of the form **1** and the pair of curves $HOC(Q_r, n-1)$ and $HOC(Q_r, n-2)$ corresponds to the type **2**, it can be easily shown that $\mathcal{P}(n) = [\overline{Q}_r(n), q_l^n)$, with $l = HOC(\overline{Q}_r(n-1), n-1)$, $\mathcal{P}(n-1) = [\overline{Q}_r(n-1), Q_r^{n-1,n-2}]$ and the interval associated to $\mathcal{P}(n-2)$ begins at point $Q_r^{n-1,n-2}$ (see Figure 4.6 b)). Again, the inclusion of curve $HOC(Q_r, n-2)$ does not affect the efficient solutions corresponding to curves $HOC(Q_r, n)$ and $HOC(Q_r, n-1)$, and hence, case **1-2** can be considered separately.

c) When the combination of curves $HOC(Q_r, n)$ and $HOC(Q_r, n-1)$ is of the form **2** and the pair of curves $HOC(Q_r, n-1)$ and $HOC(Q_r, n-2)$ corresponds to the type **1**, two different situations can arise. In particular, we must distinguish two cases, namely, if $HOC(\overline{Q}_r(n-2), n-2) > HOC(Q_r^{n,n-1}, n-1)$ or $HOC(\overline{Q}_r(n-2), n-2) \leq HOC(Q_r^{n,n-1}, n-1)$. The latter case, depicted in Figure 4.7, is not feasible since $q_{l'}^{n-1} < q_{l'}^n < \overline{Q}_r(n-2)$ with $l' = HOC(\overline{Q}_r(n-2), n-2)$, which contradicts the fact that the lower level set $\varphi_{l'}(Q_r, n) \leq 0$ is a convex set containing level sets $\varphi_l(Q_r, m) \leq 0$, with $n < m \leq \overline{n}$ and $l = HOC(\overline{Q}_r(m), m)$. Thus, the unique valid alternative is that $HOC(\overline{Q}_r(n-2), n-2) > HOC(Q_r^{n,n-1}, n-1)$ (see Figure 4.6 c)), and hence, combination 2-1 can be analyzed separately to give $\mathcal{P}(n) = [\overline{Q}_r(n), Q_r^{n,n-1})$, $\mathcal{P}(n-1) = [Q_r^{n,n-1}, q_{l'}^{n-1})$ and the interval $\mathcal{P}(n-2)$ starting from $\overline{Q}_r(n-2)$.

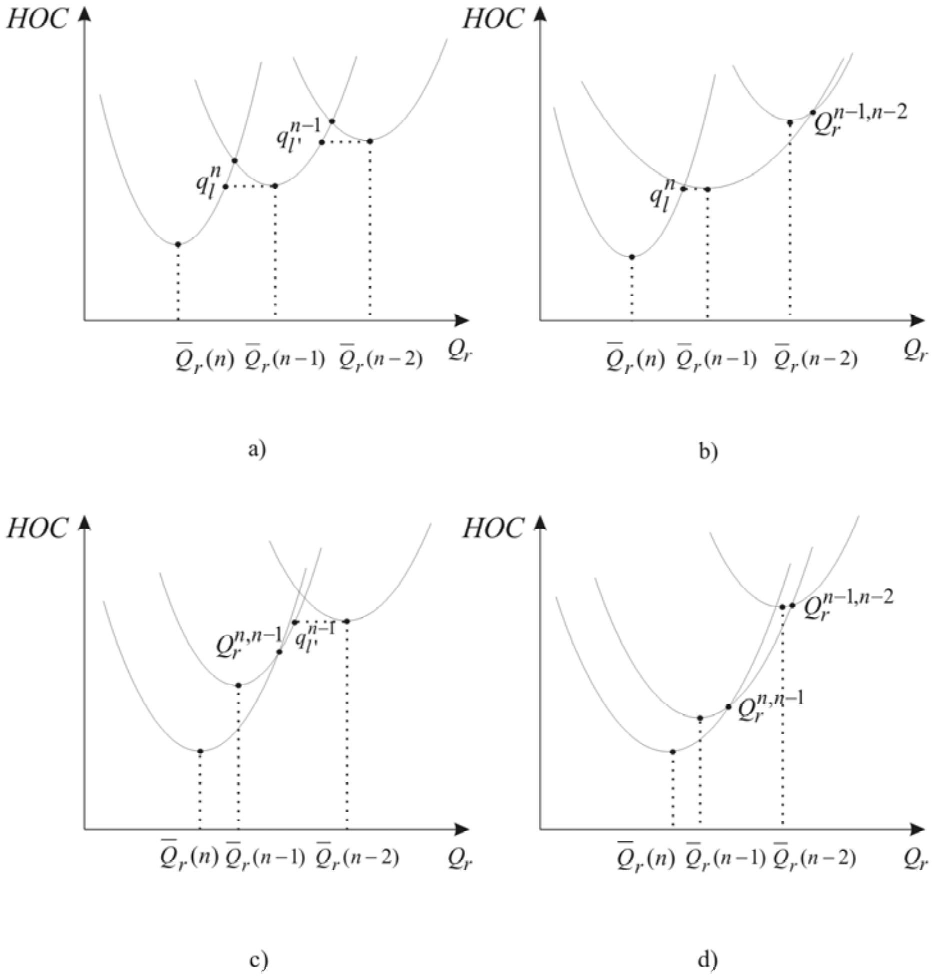

Figure 4.6: Admissible cases when three consecutive HOC functions are compared.

d) When the pairs of curves $HOC(Q_r, n)$, $HOC(Q_r, n-1)$ and $HOC(Q_r, n-1)$, $HOC(Q_r, n-2)$ are both of the form **2** (see Figure 4.6 d)), combination **2-2** is reduced to independently evaluate two consecutive cases of type **2** to give $\mathcal{P}(n) = [\overline{Q}_r(n), Q_r^{n,n-1})$, $\mathcal{P}(n-1) = [Q_r^{n,n-1}, Q_r^{n-1,n-2})$ and $\mathcal{P}(n-2)$ starting from $Q_r^{n-1,n-2}$.

Concluding, any feasible combination between two pairs of consecutive curves HOC is reduced to consider each pair separately. ■

The procedure to determine the whole Pareto-optimal solution set, which is based on the previous results, is sketched in Algorithm 7.

In the next section we use this characterization of $\mathcal{P}$ to show that the approach

Algorithm 7 Procedure to determine the Pareto-optimal set for problem $BBID$

Data: D, A_r, A_w, h_r, h_w, Q_0 and function α

Determine Q_r^*
Calculate $\widehat{n}_0$
if $Q_0 \leq Q_r^*$ **then**
    Calculate $\overline{n}_0$
    Determine $\mathcal{P}$ according to Theorem 26
else
    Calculate $\overline{n}$
    $n \leftarrow \overline{n}$
    $\mathcal{P}(n) \leftarrow \emptyset$
    $\mathcal{P} \leftarrow \emptyset$
    $Q \leftarrow \overline{Q}_r(n)$
    while $Q < Q_0$ and $n - 1 \geq \widehat{n}_0$ **do**
        if $\overline{Q}_r(n-1)$ =max$\{Q_r^{n,n-1}, \overline{Q}_r(n-1)\}$ **then**
            $\mathcal{P}(n) = [Q,\min\{Q_0, q_l^n\})$, with $l = HOC(\overline{Q}_r(n-1), n-1)$
            $Q = \min\{Q_0, \overline{Q}_r(n-1)\}$
        else
            $\mathcal{P}(n) = [Q,\min\{Q_0, Q_r^{n,n-1}\})$
            $Q = \min\{Q_0, Q_r^{n,n-1}\}$
        end if
        $\mathcal{P} \leftarrow \mathcal{P} \cup \mathcal{P}n)$
        $n \leftarrow n - 1$
    end while
    if $n - 1 < \widehat{n}_0$ **then**
        $\mathcal{P} \leftarrow \mathcal{P} \cup \mathcal{P}(n) = [Q, Q_0]$
    end if
end if
return $\mathcal{P}$

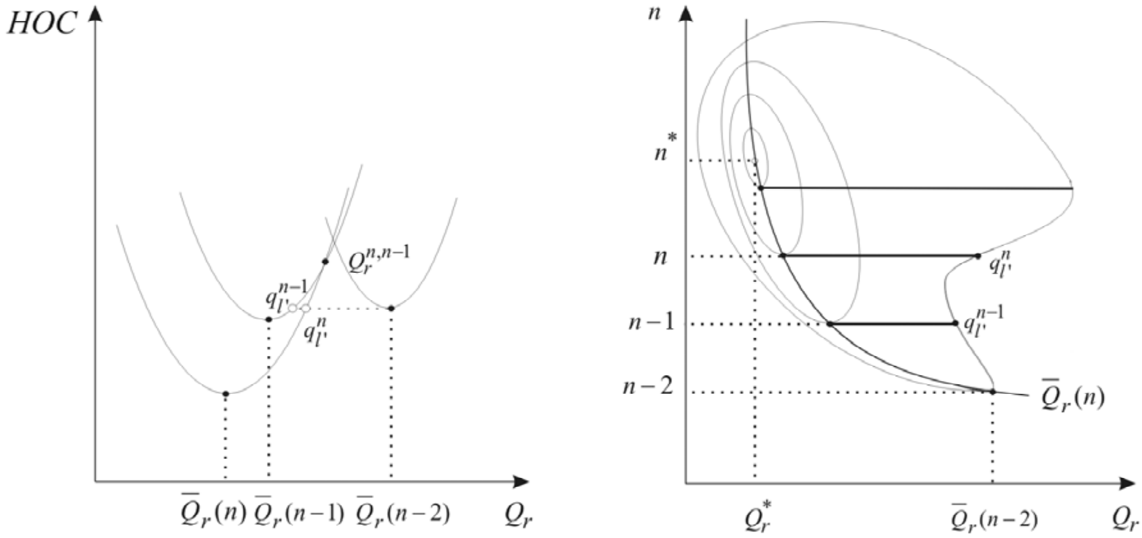

Figure 4.7: Infeasible case when $HOC(\overline{Q}_r(n-2), n-2) \leq HOC(Q_r^{n,n-1}, n-1)$.

proposed in Bookbinder and Chen [14] to solve their bicriteria problem is not correct.

4.5 Bookbinder and Chen's Approach

Bookbinder and Chen [14] likewise addressed a bicriteria two-echelon inventory / distribution system. In their problem, the first criterion coincides with our function HOC, and the second one concerns the annual transportation cost (TC). However, even though both objectives, their cost TC and our criterion DI, are conceptually different, they are characterized by the same type of function, namely, a strictly decreasing function in Q_r. Therefore, it seems reasonable to think that our solution method and their approach should provide the same solution for the same instance. Nevertheless, as we show below, their approach does not always provide good solutions. In particular, when demand is assumed to be known, their solution set consists of either the point (Q_0, n^*) if $Q_0 \leq Q_r^*$, or otherwise, an infinite number of points (Q_r, n), with $n = n^*$ and $Q_r \in [Q_r^*, Q_1]$, where $Q_1 = \min\{B(n^*), Q_0\}$. Moreover, when $Q_0 \leq Q_r^*$, they claim that the problem has its global minimum at (Q_0, n^*), (see their theorem on page 710 [14]). This assertion is wrong. As our Theorem 27 states, the problem formulated in (4.3) does not have a unique solution. Moreover, different solutions can be reached depending on the input data.

We consider example 1 described in Bookbinder and Chen [14], and change only the demand $D = 10000$ by $D = 90000$, leaving the same values for the rest of parameters. That is, $A_r = 30, A_w = 20$, (transportation unit cost) $T_r = 100, h_r = 1, h_w = 0.5$ and $Q_0 = J_r = V = 1000$. According to (4.5), $Q_r^* = 600\sqrt{30} = 3286.33$, then we have $Q_r^* \geq Q_0$. Following Bookbinder and Chen's method, the global minimum is achieved at (Q_0, n^*). Since $Q_0 = 1000$ and $n^* = 0.816$, their solution is given by $(Q_0, \lceil n^* \rceil) = (1000, 1)$ with $HOC = 5000$ and $TC = \frac{T_r D}{Q_r} = 9000$.

However, the solution above is not good, since we can find a new point that dominates the former. Thus, following our method, we obtain that $\overline{n}_0 = 11$ and $\hat{n}_0 = 3$. Therefore, $\overline{n}_0 > \hat{n}_0 + 1$, and according to Theorem 27, the optimal solution is $(Q_0 = 1000, n = 3)$ with costs $HOC = 4300$ and $TC = 9000$, respectively.

Secondly, when $Q_r^* \leq Q_0$, Bookbinder and Chen pointed out (see assertion (2) of their theorem on page 710) that the problem has non-dominated solutions (Q_r, n), with $n = n^*$ and $Q_r \in [Q_r^*, Q_1]$, where $Q_1 = \min\{B(n^*), Q_0\}$. $B(n^*)$ is given in (4.4) and represents an upper bound necessary to guarantee that the function $HOC(Q_r, n)$ has its global minimum at (Q_r^*, n^*). Nevertheless, as it has been shown in previous sections, non-dominated solutions are arranged at different intervals, changing the n integer value in each interval.

To show this effect, we consider the same example 2 proposed in [14], where the parameters are given as: $A_r = 100, A_w = 200, T_r = 400, D = 10000, h_r = 3, h_w = 1, J_r = 1500$ and $V = 2000$. Their procedure yields the following results: $n^* = 2, Q_r^* = 1000$ and $Q_0 = 1500$. Since $Q_r^* \leq Q_0$, Bookbinder and Chen asserted that the problem has an infinite number of non-dominated solutions with $n = 2$ and $1000 \leq Q_r \leq 1500$. Also, they even showed some of these solutions in their Table 1 on page 711. Again, the authors have been wrong, because point $(Q_r = 1500, n = 2)$, with $HOC = 4333$ and $TC = 2667$, was proposed as non-dominated solution in that table. However, this point is dominated by $(Q_r = 1500, n = 1)$, with $HOC = 4250$ and $TC = 2666.66$.

Therefore, the non-dominated solution set $\mathcal{P}$ should be determined according to Algorithm 7. First, we must calculate $\overline{n} = 2$ and $\widehat{n}_0 = 1$, hence $k = \overline{n} - \widehat{n}_0 + 1 = 2$. Moreover, $\overline{Q}_r(1) = \max\{Q_r^{2,1}, \overline{Q}_r(1)\} = 1000\sqrt{2}$, $l = HOC(\overline{Q}_r(1), 1) = 6000/\sqrt{2}$ and thus $q_l^2 = 1000\sqrt{2}$. Hence, $\mathcal{P}(2) = [\overline{Q}_r(2), q_l^2)$ and we proceed to evaluate $n = 1$ with $Q = \overline{Q}_r(1)$. Since $n - 1 = 0 < \widehat{n}_0$, the algorithm finishes determining $\mathcal{P}(1) = [\overline{Q}_r(1), Q_0]$, therefore, the Pareto-solution set contains two intervals, namely,

$$\begin{aligned} \mathcal{P} = & \ \{(Q_r, 2) : Q_r \in [\overline{Q}_r(2), \overline{Q}_r(1))\} \cup \{(Q_r, 1) : Q_r \in [\overline{Q}_r(1), Q_0]\} \\ = & \ \{(Q_r, 2) : Q_r \in [1000, 1000\sqrt{2}]\} \cup \{(Q_r, 1) : Q_r \in [1000\sqrt{2}, 1500]\} \end{aligned}$$

Hence, for $Q_r \geq 1000\sqrt{2}$, all those solutions proposed by Bookbinder and Chen's method are not efficient and are dominated by points $(Q_r, n = 1)$.

4.6 Computational Results

The procedure corresponding to Algorithm 7 was implemented in C++ using LEDA libraries [58] on a HP-712/60 workstation. In order to check the efficiency of this algorithm, multiple instances were randomly generated. The input data were obtained from uniform distributions on intervals, where the minimum and maximum values were different random numbers. In Table 4.1, thirty instances are shown.

The Pareto-optimal solution sets for the instances in Table 4.1 are shown in Table 4.2.

The efficiency of our procedure has been tested. This test consists of generating one thousand random points for each instance. Then, we choose among them those which are non-dominated by using an enumerative comparison algorithm. We compare our Pareto solution set with the non-dominated randomly generated points. For each non-dominated generated point, we have to determine whether the point is included in the Pareto-optimal solution set proposed, or it is dominated by a point in that set. In all the instances, the considered points either belong to our solution set or they are dominated by points in our Pareto-optimal solution set.

4.7 Conclusions

In this chapter, we have studied a non-linear biobjective optimization model for a two-echelon serial inventory/distribution system with deterministic demand. We have characterized the non-dominated optimal solution set and proposed an algorithm to generate it.

A similar model was studied by Bookbinder and Chen [14], but unfortunately their solution method is not correct as we have shown in a previous section. The complete analysis of the problem requires a detailed study of the model. In this analysis, it is not possible to use the classical tools in multiobjective optimization because the problem is a mixed-integer non-linear two-objective optimization model, where neither the tools of continuous nor discrete optimization are directly applicable. We have performed this analysis decomposing the problem and integrating the solutions obtained in each subproblem into the final solution set. Two goals have been achieved in this chapter: to study a mixed-integer non-linear two-objective optimization model, which is completely resolvable and to correct the solution of a model already proposed in the literature.

Further research could be carried out to analyze the biechelon inventory/distribution system here studied, but considering more than two criteria. Also, it could be in-

	Ar	**Aw**	**hw**	**hr**	**Q0**	**D**
P1	4.70	1.98	3.79	5.24	90.90	256.56
P2	3.26	7.86	0.14	0.33	44.34	38.95
P3	5.89	5.57	4.04	7.42	28.13	918.46
P4	5.70	7.31	0.14	1.50	60.30	77.23
P5	7.13	5.13	0.18	1.68	64.10	19.76
P6	1.00	4.60	0.41	0.79	77.33	63.40
P7	9.05	3.86	0.14	1.12	68.65	45.32
P8	8.17	8.83	6.60	6.79	52.36	652.69
P9	3.37	8.09	0.63	1.72	6.36	4.11
P10	9.39	9.49	1.04	4.83	60.05	611.55
P11	8.07	2.21	0.01	0.06	75.25	18.22
P12	4.17	4.25	1.20	4.32	38.81	587.49
P13	1.98	5.49	0.43	0.55	99.49	92.27
P14	8.81	7.44	1.43	5.75	97.99	416.01
P15	5.91	8.33	0.22	1.31	38.02	69.45
P16	6.52	5.47	2.24	9.26	48.30	758.28
P17	8.25	5.39	2.32	5.94	17.48	97.77
P18	2.04	4.99	0.81	1.43	13.26	26.43
P19	9.08	8.42	0.18	1.93	52.90	52.47
P20	7.94	3.13	1.08	6.16	29.23	689.42
P21	1.07	7.53	0.08	0.10	62.92	43.00
P22	7.59	6.46	0.04	0.12	13.35	76.87
P23	9.90	2.12	2.75	5.73	26.70	166.27
P24	5.34	5.90	0.36	1.85	29.17	44.79
P25	9.37	6.52	0.05	1.37	37.24	97.04
P26	7.51	8.53	0.20	0.77	68.50	94.92
P27	1.05	4.98	0.26	1.67	19.85	73.75
P28	4.45	6.17	0.11	0.26	42.89	12.90
P29	1.87	6.52	0.62	1.07	39.12	97.09
P30	7.48	8.01	0.97	0.99	58.16	96.30

Table 4.1: Thirty randomly generated instances of the BBID problem.

Problem	Variables	Pareto Optimal Solution Set		
P1	Qr	[25.57,90.90]		
	n	1		
P2	Qr	[34.52,44.34]		
	n	2		
P3	Qr	[28.13,28.13]		
	n	2		
P4	Qr	[36.66,60.30]	[26.57,36.66)	[24.60,25.37)
	n	2	3	4
P5	Qr	[23.73,64.10]	[14.35,23.73)	
	n	1	2	
P6	Qr	[29.98,77.33]	[18.67,26.12)	
	n	1	2	
P7	Qr	[35.34,68.65]	[28.10,35.34)	
	n	1	2	
P8	Qr	[52.36,52.36]		
	n	1		
P9	Qr	[5.09,6.36]		
	n	2		
P10	Qr	[54.27,60.05]		
	n	2		
P11	Qr	[75.25,75.25]		
	n	1		
P12	Qr	[36.60,38.81]		
	n	2		
P13	Qr	[50.06,99.49]		
	n	1		
P14	Qr	[48.49,97.99]	[38.10,46.31)	
	n	1	2	
P15	Qr	[30.24,38.02]	[26.25,29.54)	
	n	2	3	
P16	Qr	[44.31,48.30]	[34.93,42.94)	
	n	1	2	
P17	Qr	[17.48,17.48]		
	n	1		
P18	Qr	[13.26,13.26]	[10.34,12.21]	
	n	1	2	
P19	Qr	[49.54,52.90]	[28.60,49.54)	[23.33,28.60)
	n	1	2	3
P20	Qr	[29.23,29.23]		
	n	2		
P21	Qr	[48.06,62.92]		
	n	2		
P22	Qr	[13.35,13.35]		
	n	12		
P23	Qr	[26.41,26.70]		
	n	1		
P24	Qr	[27.09,29.17]	[18.33,27.09)	
	n	1	2	
P25	Qr	[37.24,37.24]		
	n	4		
P26	Qr	[63.62,68.50]	[48.00,63.62)	
	n	1	2	
P27	Qr	[15.34,19.85]	[11.88,15.34)	[10.55,11.88)
	n	3	4	5
P28	Qr	[32.46,42.89]		
	n	1		
P29	Qr	[39.02,39.12]	[24.27,29.23)	
	n	1	2	
P30	Qr	[54.89,58.16]		
	n	1		

Table 4.2: Pareto-optimal solution sets for the problems shown in Table 4.1.

teresting to extend the results of this non-linear biobjective optimization model to multi-echelon serial inventory/distribution systems or, in general, to multi-echelon systems.

Bibliography

[1] A. Aggarwal, J. K. Park. Improved Algorithms for Economic Lot Size Problems. *Operations Research*, 41 (1993), 549-571.

[2] R. K. Ahuja, A. V. Goldberg, J. B. Orlin, R. E. Tarjan. Finding Minimum-Cost Flows by Double Scaling. *Mathematical Programming* 53 (1992), 243-266.

[3] R. K. Ahuja, T. L. Magnanti, J. B. Orlin. *Network Flows, Theory, Algorithms, and Applications*. Prentice-Hall, Englewood Cliffs, New Jersey, 1993.

[4] M. J. Alves, J. Climaco. An Interactive Reference Point Approach for Multiobjective Mixed-Integer Programming using Branch-and-Bound. *European Journal of Operational Research*, 124 (2000), 478-494.

[5] K. Andler. *Rationalisierung der Fabrikation und Optimale Losgrösse*. Oldenburg, Munich, 1929.

[6] S. Axsäter. *Inventory Control*. Kluwer Academic Publishers, The Netherlands, 2000.

[7] K. R. Baker, P. Dixon, M. J. Magazine, E. A. Silver. An Algorithm for the Dynamic Lot-size Problem with Time-varying Production Capacity Constraints. *Management Science*, 24(16) (1978), 710-720.

[8] M. S. Bazaraa, J. J. Jarvis, H. D. Sherali. *Linear Programming and Network Flows*, 2nd edition. Wiley, New York, 1990.

[9] R. Bellman, S. E. Dreyfus. *Applied Dynamic Programming*. Princeton University Press, Princeton, New Jersey, 1961.

[10] M. Berg, M. Van Kreveld, M. Overmars, O. Schwarzkopf. *Computational Geometry Algorithms and Applications*. Springer, Berlin-Heidelberg, 1997.

[11] J. Birge, F. V. Louveaux. *Introduction to Stochastic Programming*. Springer, Berlin, 1997.

[12] G. B. Bitran, H. H. Yanasse. Computational Complexity of the Capacitated Lot Size Problem. *Management Science*, 28(10) (1982), 1174-1185.

[13] G. B. Bitran, H. Matsuo. Approximation Formulations for the Single-product Capacitated Lot Size Problem. *Operations Research*, 34(1) (1986), 63-74.

[14] J. H. Bookbinder, V. Y. X. Chen. Multicriteria Trade-Offs in a Warehouse/Retailer System. *Journal of the Operational Research Society*, 43(7) (1992), 707-720.

[15] H. Booth, R. E. Tarjan. Finding the Minimum-Cost Maximum Flow in a Series-Parallel Network. *Journal of Algorithm* 15 (1993), 416-446.

[16] R. G. Brown. *Decision Rules for Inventory Management*. Holt, Rinehart and Winston, New York, 1967.

[17] W. H. Chen, J. M. Thizy. Analysis of Relaxations for the Multi-item Capacitated Lotsizing Problem. *Annals of Operations Research*, 26 (1990), 29-72.

[18] A. Chikán (Ed.). *Inventory Models*. Kluwer Academic Publishers, The Netherlands, 1990.

[19] C. S. Chun, C. H. M. Lin. An O(T^2) Algorithm for the NI/G/NI/ND Capacitated Lot Size Problem. *Management Science*, 3 (1988), 420-425.

[20] A. J. Clark, H. Scarf. Optimal Policies for a Multi-Echelon Inventory Problems. *Management Science*, 6(4) (1960), 475-490.

[21] W. B. Crowston, H. M. Wagner, J. F. Williams. Economic Lot Size Determination in Multi-Stage Assembly Systems. *Management Science*, 19(5) (1973), 517-527.

[22] M.A.H. Dempster. On Stochastic Programming II: Dynamic Problems under Risk. *Stochastics,* 25 (1988), 15-42.

[23] E. V. Denardo. *Dynamic Programming: Models and Applications*. Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

[24] P. S. Dixon, C. L. Poh. Heuristic Procedures for Multi-item Inventory Planning with Limited Storage. *IIE Transactions*, 22(2) (1990), 112-123.

[25] A. Dolan, J. Aldous. *Networks and Algorithms An Introductory Approach*. Wiley, Chichester, 1993.

[26] R. J. Duffin. Topology of Series-Parallel Networks. *Journal of Mathematical Applications* 10 (1965), 303-318.

[27] M. Ehrgott. *Multicriteria Optimization.* Springer, Berlin, 2000.

[28] M. Ehrgott, X. Gandibleux. An Annotated Bibliography of Multicriteria Combinatorial Optimization. *OR-Spektrum*, 22 (2000), 425-460.

[29] G. D. Eppen, F. J. Gould, B. P. Pashigian. Extensions of the Planning Horizon Theorem in the Dynamic Lot Size Model. *Management Science*, 15(5) (1969), 268-277.

[30] S. Erenguc, M. Mercan. On the Joint Lot-Sizing Problem with Zero Setup Costs. *Decisions Sciences* 20(4) (1989), 669-676.

[31] L.F. Escudero, P.V. Kamesan. *MRP Modeling via Scenarios.* In: T. A. Ciriani, R. C. Leachman (Eds.), Optimisation in Industry. Wiley, New York, 101-111, 1993.

[32] L.F. Escudero, P.V. Kamesan. On Solving Stochastic Production Planning Problems via Scenario Modelling. *TOP*, 3 (1995), 69-96.

[33] A. Federgruen, M. Tzur. A Simple Forward Algorithm to Solve General Dynamic Lot Sizing Models with n Periods in $O(n \log n)$ or $O(n)$ Time. *Management Science*, 37 (1991), 909-925.

[34] E. Fernández, J. Puerto. The Multiobjective Solution of the Uncapacitated Plant Location Problem. *European Journal of Operational Research*, 145(3) (2002), 509-529.

[35] F. R. Fernández, S. Nickel, J. Puerto, A. Rodríguez-Chía. Robustness of the Pareto-solution for the Multicriteria Minisum Location Problem. *Journal of Multicriteria Decision Analysis*, 10 (2001), 191-203.

[36] M. Florian, M. Klein. Deterministic Production Planning with Concave Costs and Capacity Constraints. *Management Science*, 18(1) (1971), 12-20.

[37] M. Florian, J. K. Lenstra, A. H. G.Rinnooy Kan. Deterministic Production Planning: Algorithms and Complexity. *Management Science*, 26 (1980), 12-20.

[38] X. Gandibleux, A. Jaszkiewicz, A. Freville, R. Slowinski (Eds.). Multiobjective Methaheuristics. *Journal of Heuristics*, 6(3) (2000), 291-431, special issue.

[39] E. S. Gardnet, D. G. Dannenbring. Using Optimal Policy Surfaces to Analyze Aggregate Inventory Trade-Offs. *Management Science*, 25 (1979),

[40] M. R. Garey, D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York, 1979.

[41] A. M. Geoffrion. Proper Efficiency and the Theory of Vector Maximization. *Journal of Mathematical Analysis and Application*, 22 (3) (1968), 618-630.

[42] A. Goicoechea, D. R. Hansen, L. Duckstein. *Multiobjective Decision Analysis with Engineering and Business Applications*. Wiley, New York, 1982.

[43] V. Goldberg, R. E. Tarjan. Solving Minimum Cost Flow Problem by Successive Approximation. *Mathematics of Operations Research* 15 (1990), 430-466.

[44] J. Gutiérrez, J. Puerto, J. Sicilia. Bicriteria Trade-off Analysis in a Two-echelon Inventory/Distribution System. *Journal of the Operational Research Society*, 53 (2002), 468-469.

[45] J. Gutiérrez, J. Puerto, J. Sicilia. The Multiscenario Lot Size Problem with Concave Costs. To appear in the *European Journal of Operational Research*.

[46] J. Gutiérrez, A. Sedeño-Noda, M. Colebrook, J. Sicilia. A New Characterization for the Dynamic Lot Size Problem with Bounded Inventory. *Computers and Operations Research*, 30(3) (2003), 383-395.

[47] J. Gutiérrez, A. Sedeño-Noda, M. Colebrook, J. Sicilia. Solving the Dynamic Lot Size Problem with Time-varying Storage Capacities in $O(T \log T)$. Submitted to *Optimization Methods and Software*.

[48] F. W. Harris. How Many Parts to Make at Once, Factory, *The Magazine of Management*, 10 (1913), 135-136, 152.

[49] A. C. Hax, D. Candea. *Production and Inventory Management*. Prentice-Hall, Englewood Cliffs, New Jersey, 1984.

[50] J. Hershberger. Finding the Upper Envelope of n Line Segments in $O(n \log n)$ Time. *Information Processing Letters*, 33(4) (1989), 169-174.

[51] J. Hershberger, S. Suri. Off-Line Maintenance of Planar Configurations. *Journal of Algorithms*, 21 (1996), 453-475.

[52] D. J. Hockstra. Simple Policies in the Economic Lot Scheduling Problem with Zero Setup Costs. *Production Planning and Control* 8(1) (1997), 25-30.

[53] P. Horner. Planning under Uncertainty, Question & Answers with G. Dantzig. *OR/MS Today*, October (1999), 26-30.

[54] R. Jagannathan, M. R. Rao. A Class of Deterministic Production Planning Problems. *Management Science*, 19(11) (1973), 1295-1300.

[55] D. Johnson, C. H. Papadimitriou. Computational Complexity. In *The Traveling Salesman Problem. A guided tour of Combinatorial Optimization*, E. Lawler, J. Lenstra, A. H. G. Rinnooy Kan, D. B. Shmoys (Eds.). Wiley, Chichester, 1985.

[56] U. Karmarkar, S. Kekre, S. Kekre. The Dynamic Lot-sizing Problem with Setup and Reservation Costs. *Operations Research*, 35 (1987), 389-398.

[57] K. Klamroth, M. Wievceck. Dynamic Programming Approach to the Multiple Criteria Knapsack Problem. *Naval Research Logistic*, 47 (2000), 57-76.

[58] LEDA User Manual 4.2.1 at www.mpi-sb.mpg.de/LEDA/MANUAL/MANUAL.html.

[59] F. Louveaux. An Algorithm for Deterministic Production Planning with Concave Costs and Capacity Constraints. CORE discussion paper 7401 (1974), Université Catholique de Louvain, Louvain, Belgium.

[60] S. F. Love. Bounded Production and Inventory Models with Piecewise Concave Costs. *Management Science*, 20(3) (1973), 313-318.

[61] A. S. Manne. Programming of Economic Lot Sizes. *Management Science*, 4 (1958), 115-135.

[62] B. Margansky. *Wirtschaftliche Lagerhaltung*. Berlin.1933.

[63] T. L. Morin, A. M. O. Esoboque. The Imbedded State Space Approach to Reducing Dimensionality in Dynamic Programs of Higher Dimensions. *Journal of Mathematical Analysis and Applications*, 48 (1974), 801-810.

[64] E. Naddor. *Inventory Systems*. Wiley, New York, 1966.

[65] S. L. Narasimhan, D. W. McLeavey, P. J. Billington. *Production Planning and Inventory Control*, 2nd edition, Prentice-Hall, Englewood Cliffs, New Jersey, 1995.

[66] G. L. Nemhauser. *Introduction to Dynamic Programming*. Wiley, New York, 1966.

[67] J. B. Orlin. A Faster Strongly Polynomial Minimum Cost Flow Algorithm. *Operations Research* 41(2) (1993), 338-350.

[68] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[69] G. W. Plossl. *Production and Inventory Control*, 2nd edition, Prentice-Hall, Englewood Cliffs, New Jersey, 1985.

[70] F. P. Preparata, M. I. Shamos. *Computational Geometry An Introduction*. Springer, New York, 1985.

[71] J. Puerto, F. R. Fernández. Pareto-optimality in Classical Inventory Models. *Naval Research Logistic*, 45 (1998), 83-98.

[72] J. Puerto, F. R. Fernández. Multicriteria Weber Location Problems. *Journal of the Multicriteria Decision Analysis*, 18 (1999), 268-280.

[73] R. Ramesh, S. Zionts, M. Karwan. A Class of Practical Interactive Branch and Bound Algorithms for Multicriteria Integer Programming. *European Journal of Operational Research*, 26 (1986), 161-172.

[74] L. M. Rasmussen. Zero-one Programming with Multiple Criteria. *European Journal of Operational Research*, 26 (1986), 83-95.

[75] R. T. Rockafellar, R. J. -B Wets. Scenario and Policy Aggregation in Optimisation under Uncertainty. *Mathematics of Operations Research*, 16 (1991), 119-147.

[76] W. Röckelein. Systems for Purchase on the Internet Requirements and Evaluation. Presented in the *Fifth Symposium on Research and Teaching in Electronic Commerce*, Bled, Slovenia, 1995.

[77] A. M. Rodríguez-Chia, J. Puerto. A Multicriteria Competitive Markov Decision Process. *Mathematical Methods of O.R.*, 55 (2002), 359-369.

[78] G. Ruhe. Complexity Results for Multicriteria and Parametric Network Flows using a Pathological Graph of Zadeh. *Zeitschrift für Operations Research*, 32 (1988), 9-27.

[79] M. Salomon. *Deterministic Lotsizing Models for Production Planning*. Springer, Berlin, 1991.

[80] L. B. Schwarz (Ed.). *Multi-level Production/Inventory Control Systems: Theory and Practice*. North-Holland, New York, 1981.

[81] A. Sedeño-Noda, J. Gutiérrez, B. Abdul-Jalbar, J. Sicilia. An $O(T \log T)$ Algorithm for the Dynamic Lot Size Problem with Limited Storage and Linear Costs. Submitted to *Computational Optimization and Applications.*

[82] E. A. Silver, D. F. Pyke, R. Peterson. *Inventory Management and Production Planning and Scheduling*, 3rd edition. Wiley, New York, 1998.

[83] M. K. Star, D. W. Miller. *Inventory Control: Theory and Practice.* Prentice-Hall, Englewood Cliffs, New Jersey, 1962.

[84] K. Steffanic-Allmayer. Die Güngstigste Bestellunge beim Einkauf. *Sparwirtschaft*, Vienna, 1927.

[85] R. E. Steuer. *Multiple Criteria Optimization: Theory, Computation and Application.* Wiley, New York, 1986.

[86] C. Swoveland. A Deterministic Multi-period Production Planning Model with Piecewise Concave Production and Holding-Backorder Costs. *Management Science*, 21(9) (1975), 1007-1013.

[87] H. A. Taha, R. W. Skeith. The Economic Lot Sizes in Multistage Production Systems. *AIIE Transactions*, 2 (1970), 157-162.

[88] R. J. Tersine. *Principles of Inventory and Materials Management*, 4th edition, Prentice-Hall, Englewood Cliffs, New Jersey, 1994.

[89] E. L. Ulungu, J. Teghem. Multi-objective Combinatorial Optimization Problems: A Survey. *Journal of Multicriteria Decision Analysis*, 3 (1994), 83-104.

[90] C. P. M. Van Hoesel, A. P. M. Wagelmans. An $O(T^3)$ Algorithm for the Economic Lot-sizing Problem with Constant Capacities. *Management Science*, 42(1) (1996), 142-150.

[91] A. F. Veinott. Unpublished class notes for the Program in Operations Research at Stanford University. 1963.

[92] A. F. Veinott. Production Planning with Convex Costs: A Parametric Study. *Management Science*, 10 (1964), 441-460.

[93] M. Villarreal, H. Karwan. Multicriteria Dynamic Programming with an Application to the Integer Case. *Journal of Optimization Theory and Applications*, 38 (1) (1982), 43-69.

[94] M. Villarreal, H. Karwan, S. Zionts. *An Interactive Branch and Bound Procedure for Multicriterion Integer Linear Programming*, in: G. Fandel, T. Gal (Eds.), Multiple Criteria Decision Making Theory and Application, Lecture Notes in Economics and Mathematical Systems 177, Springer, Berlin, 198, 448-467.

[95] A. Wagelmans, S. Van Hoesel, A. Kolen. Economic Lot Sizing: An O($n \log n$) Algorithm that Runs in Linear Time in the Wagner-Whitin Case. *Operations Research*, 40 (1992), 145-156.

[96] H. M. Wagner. A Postcript to Dynamic Problems in the Theory of the Firm. *Naval Research Logistic Quarterly*, 7 (1960), 7-12.

[97] H. M. Wagner, T. H. Whitin. Dynamic version of the Economic Lot Size Model. *Management Science*, 5(1) (1958), 88-96.

[98] C. D. J. Waters. *Inventory Control and Management*. Wiley, England, 1992.

[99] R. J.-B. Wets. *Large-scale Linear Programming Techniques*. In: Y. Ermoliev, R. J.-B. Wets (Eds.), Numerical Techniques for Stochastic Optimization. Springer, Berlin, 65-93, 1988.

[100] J. F. Williams. On the Optimality of Integer Lot Size Ratios in Economic Lot Size Determination in Multi-stage Assembly Systems. *Management Science*, 28 (1982), 1341-1349.

[101] R. H. Wilson, Scientific Routine for Stock Control, *Harvard Business Review*, 13 (1934), 116-128.

[102] L. A. Wolsey. *Integer Programming*. Wiley, New York, 1998.

[103] E. Zabel. Some Generalizations of an Inventory Planning Horizon Theorem. *Management Science*, 10(3) (1964), 465-471.

[104] W. Zangwill. A Deterministic Multi-period Production Scheduling Model with Backlogging. *Management Science*, 13 (1966), 105-119.

[105] W. Zangwill. Minimum Concave Cost Flows in Certain Networks. *Management Science*, 14 (1968), 429-450.

[106] W. Zangwill. A Backlogging Model and a Multiechelon Model of a Dynamic Economic Lot Size Production System - A Network Approach. *Management Science*, 15(9) (1969), 506-527.

[107] M. Zeleny. *Multiple Criteria Decision Making*. McGraw-Hill, New York, 1982.

[108] S. Zionts. A Survey of Multiple Criteria Integer Programming Methods. *Annals of Discrete Mathematics*, 5 (1979), 389-398.

[109] S. Zionts, J. Wallenius. Identifying Efficient Vectors: Some Theory and Computational Results. *Operations Research*, 23 (1980), 785-793.

[110] P. H. Zipkin. *Foundations of Inventory Management*. McGraw-Hill, New York, 2000.