Algoritmo Wagner-Whitin: Modelo Dinámico de Dimensionamiento de Lotes

Martin Rojas

9 de noviembre de 2025

Resumen

Este artículo presenta una exposición exhaustiva del algoritmo Wagner-Whitin para la solución óptima del problema de dimensionamiento de lotes con demanda dinámica y determinística. Partiendo del artículo fundacional de Wagner y Whitin (1958), se desarrolla el marco teórico del modelo, se demuestran los teoremas fundamentales que sustentan el algoritmo, se describe el procedimiento computacional y se ilustra con ejemplos numéricos. El análisis revela la conexión entre la programación dinámica y la gestión de inventarios, destacando el concepto de horizonte de planificación como herramienta para reducir la complejidad computacional. Se incluyen definiciones formales de los conceptos clave y se discuten extensiones modernas del modelo.

1. Introducción

El problema de dimensionamiento de lotes (*lot sizing*) constituye uno de los problemas fundamentales en la gestión de inventarios. Mientras que el modelo económico clásico (*Economic Order Quantity - EOQ*) supone una demanda constante a través del tiempo, en entornos reales la demanda frecuentemente exhibe variaciones significativas entre periodos.

El algoritmo Wagner-Whitin, introducido en 1958 por Harvey M. Wagner y Thomson M. Whitin, representa la primera solución óptima para el problema de dimensionamiento de lotes con demanda dinámica y determinística. Este algoritmo emplea programación dinámica para determinar cuándo y cuánto ordenar, minimizando los costos totales de *setup* y mantenimiento de inventario a lo largo de un horizonte finito de planificación.

La relevancia del algoritmo persiste hasta la actualidad, sirviendo como base para numerosas extensiones que incorporan restricciones de capacidad, múltiples productos y condiciones de incertidumbre. Este trabajo busca presentar una exposición integral del algoritmo, desde sus fundamentos teóricos hasta su implementación práctica.

2. Marco Teórico

2.1. Fundamentos de Teoría de Inventarios

Definición 2.1 (Sistema de Inventarios). *Un sistema de inventarios consiste en los procesos de adquisición, almacenamiento y distribución de materiales, con el objetivo de garantizar la disponibilidad de productos mientras se minimizan los costos asociados.*

Definición 2.2 (Costos Relevantes). *En un sistema de inventarios, se consideran típicamente los siguientes costos:*

- ***Costo de ordenar/setup*** *(s_t): Costo fijo incurrido al realizar un pedido o iniciar una producción*
- ***Costo de mantener inventario*** *(h_t): Costo por unidad almacenada por periodo*
- ***Costo de shortage*** *(p_t): Costo por unidad de demanda insatisfecha (no considerado en el modelo Wagner-Whitin clásico)*

Definición 2.3 (Modelo EOQ Clásico). *El modelo de Cantidad Económica de Pedido supone demanda constante D, costo de ordenar S, costo de mantener H por unidad por tiempo. El problema que resuelve el modelo EOQ consiste en encontrar la cantidad de pedido Q que minimice la función de costo total por unidad de tiempo:*

$$TC(Q) = \frac{DS}{Q} + \frac{HQ}{2}$$

donde el primer término representa el costo total de ordenar y el segundo término el costo total de mantener inventario.

La solución óptima a este problema está dada por:

$$Q^* = \sqrt{\frac{2DS}{H}}$$

que produce un costo total mínimo por unidad de tiempo:

$$TC^* = \sqrt{2DSH}$$

Sin embargo, el modelo EOQ resulta inadecuado cuando la demanda varía significativamente entre periodos, lo que motiva el desarrollo de modelos dinámicos como el de Wagner-Whitin.

2.2. Programación Dinámica

El algoritmo Wagner-Whitin se fundamenta en la programación dinámica, técnica desarrollada por Richard Bellman.

Definición 2.4 (Principio de Optimalidad de Bellman)**.** *"Una política óptima tiene la propiedad de que, cualesquiera que sean el estado inicial y la decisión inicial, las decisiones restantes deben constituir una política óptima con respecto al estado resultante de la primera decisión."*

Definición 2.5 (Problema de Optimización en Etapas)**.** *Un problema de optimización en etapas (o de programación dinámica) consiste en encontrar una secuencia de decisiones óptimas a lo largo de un horizonte finito de T etapas. Formalmente, se define mediante los siguientes elementos:*

- ***Horizonte temporal****: $t = 1, 2, \ldots, T$ momentos discretos en los que se toman decisiones. Cada etapa t corresponde a un subproblema.*
- ***Estado del sistema*** *$s_t \in S_t$: Describe la situación del sistema al inicio de la etapa t. El estado s_t resume toda la información relevante para tomar decisiones futuras. S_t es el espacio de estados en la etapa t.*
- ***Decisión o control****: $x_t \in X_t(s_t)$, donde $X_t(s_t)$ es el conjunto de decisiones factibles en el estado s_t*
- ***Función de transición****: $s_{t+1} = g_t(s_t, x_t)$, determina cómo evoluciona el sistema al tomar la decisión x_t en el estado s_t*
- ***Costo inmediato****: $c_t(s_t, x_t)$, costo incurrido en la etapa t*

- ***Costo terminal:*** *$V_{T+1}(s_{T+1})$, costo asociado al estado final (usualmente 0)*

El objetivo es encontrar una política $\pi = (x_1, x_2, \ldots, x_T)$ que minimice el costo total esperado:

$$J_\pi(s_1) = \sum_{t=1}^{T} c_t(s_t, x_t) + V_{T+1}(s_{T+1})$$

Definición 2.6 (Ecuación de Bellman para Gestión de Inventarios)**.** *En el contexto del problema de dimensionamiento de lotes, la ecuación de Bellman describe la política óptima de decisión periodo a periodo. Para cada instante $t = 1, 2, \ldots, N$:*

$$f_t(I) = \min_{x_t \geq 0} \begin{cases} h_{t-1} \cdot I + \delta(x_t) \cdot s_t + f_{t+1}(I + x_t - d_t) & \text{si } I + x_t \geq d_t \\ \infty & \text{en otro caso} \end{cases} \tag{1}$$

donde:

- *$f_t(I)$:* ***Función de Bellman*** *en el periodo t con inventario inicial I, que representa el costo mínimo acumulado desde t hasta el horizonte N*
- *t:* ***Índice temporal*** *que varía en $1 \leq t \leq N$ (periodos discretos)*
- *I:* ***Nivel de inventario*** *al inicio del periodo t*
- *x_t:* ***Cantidad a ordenar*** *en el periodo t (variable de decisión)*
- *h_{t-1}:* ***Costo unitario de mantenimiento*** *del periodo $t-1$ a t*
- *$\delta(x_t)$:* ***Función indicadora*** *de orden ($\delta(x_t) = 1$ si $x_t > 0$, 0 en otro caso)*
- *s_t:* ***Costo fijo de ordenar*** *en el periodo t*
- *d_t:* ***Demanda*** *en el periodo t*

La ***condición terminal*** *completa la definición recursiva:*

$$f_{N+1}(I) = \begin{cases} 0 & \text{si } I = 0 \\ \infty & \text{si } I > 0 \end{cases} \quad \text{para todo } I \geq 0 \tag{2}$$

2.3. Optimización Convexa y Estructuras Especiales

Aunque el problema de Wagner-Whitin incluye costos fijos (no convexos), la estructura especial del problema permite encontrar soluciones óptimas globales.

Definición 2.7 (Función Convexa)**.** *Una función* $f : \mathbb{R}^n \to \mathbb{R}$ *es convexa si para todo* $x, y \in \mathbb{R}^n$ *y* $\lambda \in [0, 1]$*:*

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y) \tag{3}$$

Definición 2.8 (Función Cóncava)**.** *Una función* $f : \mathbb{R}^n \to \mathbb{R}$ *es cóncava si* $-f$ *es convexa.*

La formulación de Wagner-Whitin, aunque no estrictamente convexa debido a la presencia de costos fijos discontinuos, exhibe propiedades estructurales que garantizan la optimalidad global de la solución encontrada mediante programación dinámica. Estas propiedades fundamentales son:

1. **Propiedad de Planificación de Cero Inventarios (Zero-Inventory Ordering)**: En una política óptima, la ordenación ocurre únicamente cuando el nivel de inventario es cero. Formalmente, existe una solución óptima que satisface:

$$I_{t-1} \cdot x_t = 0 \quad \text{para todo } t = 1, \ldots, N \tag{4}$$

   donde I_{t-1} es el inventario inicial del periodo t y x_t es la cantidad ordenada.

2. **Estructura de Subproblemas Independientes**: El problema de N periodos se descompone en subproblemas más simples mediante el principio de optimalidad de Bellman. La función de valor $f_t(I)$ satisface:

$$f_t(I) = \min_{x_t \geq 0} \{C_t(I, x_t) + f_{t+1}(I + x_t - d_t)\} \tag{5}$$

   donde $C_t(I, x_t)$ representa el costo inmediato en el periodo t.

3. **Propiedad de Monotonicidad de la Función de Valor**: La función de valor óptimo $f_t(I)$ es no decreciente en el inventario I y no creciente en el tiempo t, lo que permite podar soluciones subóptimas durante la búsqueda.

4. **Existencia de Puntos de Regeneración (Regeneration Points)**: Los periodos en los que el inventario se agota (puntos de regeneración) dividen el problema en subintervalos independientes, reduciendo la complejidad computacional de $O(2^N)$ a $O(N^2)$.

Estas propiedades garantizan que el algoritmo de Wagner-Whitin, basado en programación dinámica, encuentre la solución óptima global a pesar de la no convexidad introducida por los costos fijos. La optimalidad se demuestra mediante inducción hacia atrás sobre el horizonte de planificación, donde en cada etapa t se garantiza que $f_t(I)$ representa el costo mínimo acumulado desde t hasta el final.

2.4. Planificación Horizontal

Un concepto fundamental introducido por Wagner y Whitin es el de *horizonte de planificación*, que permite dividir el problema en subproblemas independientes.

Definición 2.9 (Horizonte de Planificación)**.** *Un periodo t constituye un horizonte de planificación si la solución óptima para los primeros t periodos es independiente de los datos de demanda y costos más allá del periodo t.*

Ejemplo: *Considere un problema con 4 periodos y demandas $d = [50, 60, 70, 80]$. Si al resolver el problema para 3 periodos encontramos que la última orden en la solución óptima ocurre en el periodo 3 (cubriendo d_3), entonces el periodo 3 es un horizonte de planificación. Las decisiones óptimas para los periodos 1-3 permanecen invariantes independientemente de la demanda del periodo 4.*

Definición 2.10 (Propiedad de Extensibilidad)**.** *Un problema de planificación posee la propiedad de extensibilidad si la solución óptima para un horizonte T puede extenderse a un horizonte $T + 1$ sin modificar las decisiones anteriores.*

Ejemplo: *Supongamos que para $T = 3$ periodos la política óptima es:*

- *Ordenar 110 unidades en $t = 1$ (cubre $d_1 = 50$ y $d_2 = 60$)*
- *Ordenar 70 unidades en $t = 3$ (cubre $d_3 = 70$)*

Si al agregar un cuarto periodo con $d_4 = 80$, la política óptima mantiene las mismas decisiones en $t = 1$ y $t = 3$, y simplemente añade una orden en $t = 4$ para d_4, entonces el problema tiene la propiedad de extensibilidad.

Estos conceptos tienen implicaciones prácticas significativas, pues permiten tomar decisiones óptimas sin conocer el futuro completo. En el algoritmo de Wagner-Whitin, la identificación de horizontes de planificación permite:

Ejemplo 2.11 (Aplicación en Wagner-Whitin)**.** *Considere un escenario con:*

$$
\begin{aligned}
d &= [20, 50, 10, 30, 40] \\
s &= [100, 100, 100, 100, 100] \quad \textit{(costos fijos)} \\
h &= [1, 1, 1, 1, 1] \quad \textit{(costos de mantenimiento)}
\end{aligned}
$$

El algoritmo puede determinar que el periodo 2 es un horizonte de planificación, lo que significa que las decisiones óptimas para los periodos 1-2 no dependen de los periodos 3-5. Esto reduce el espacio de búsqueda y simplifica el cómputo.

3. Modelo Matemático y Preliminares

3.1. Definición Formal del Problema

Considérese un horizonte de planificación discreto de N periodos. En cada periodo $t = 1, 2, \dots, N$, se definen los siguientes elementos:

Parámetros del problema (datos conocidos):

- d_t: demanda en el periodo t (unidades)
- s_t: costo fijo de *setup* (ordenar o producir) en el periodo t (\$)
- h_t: costo unitario de mantener inventario del periodo t al $t + 1$ (\$/unidad/periodo)

Variables de decisión:

- x_t: cantidad a ordenar (o producir) en el periodo t (unidades)
- I_t: nivel de inventario al final del periodo t (unidades)

Función auxiliar:

- $\delta(x_t)$: función indicadora que toma valor 1 si $x_t > 0$ y 0 en caso contrario

Definición 3.1 (Problema de Dimensionamiento de Lotes con Demanda Dinámica)**.** *El problema consiste en encontrar una política de pedidos* $\{x_t\}_{t=1}^N$ *que minimice:*

$$\min \sum_{t=1}^{N} [\delta(x_t)s_t + h_t I_t] \tag{6}$$

sujeto a las restricciones de balance de inventario y no negatividad:

$$I_t = I_{t-1} + x_t - d_t, \qquad \forall t = 1, \ldots, N \tag{7}$$

$$I_t \geq 0, \qquad \forall t = 1, \ldots, N \tag{8}$$

$$x_t \geq 0, \qquad \forall t = 1, \ldots, N \tag{9}$$

$$I_0 = 0 \qquad \textit{(inventario inicial cero)} \tag{10}$$

Proposición 3.2 (Análisis de Factibilidad y Existencia)**.** *El problema de dimensionamiento de lotes con demanda dinámica satisface:*

(a) Factibilidad: *El problema siempre es factible. Dada cualquier secuencia de demandas no negativas* $\{d_t\}_{t=1}^N$*, existe al menos una política factible. En particular, la política de lote por lote (ordenar exactamente* $x_t = d_t$ *en cada periodo* t*) es siempre factible.*

(b) Existencia de solución óptima: *Siempre existe una solución óptima finita. Esto se garantiza porque:*

- *El conjunto factible es no vacío y cerrado*
- *Los costos* $s_t > 0$ *y* $h_t \geq 0$ *son acotados*
- *La función objetivo está acotada inferiormente por 0*
- *Existe un número finito de políticas óptimas candidatas debido al* ***Teorema de Wagner-Whitin*** *(ordenar solo cuando el inventario es cero)*

Esbozo de demostración de existencia. Sea $\mathcal{F}$ el conjunto de políticas factibles. Como la política lote-por-lote pertenece a $\mathcal{F}$, tenemos $\mathcal{F} \neq \emptyset$. Además, $\mathcal{F}$ es cerrado pues está definido por igualdades y desigualdades lineales. La función objetivo $f(x, I) = \sum_{t=1}^N [\delta(x_t)s_t + h_t I_t]$ es no negativa y propia. Por el teorema de Weierstrass, al estar minimizando una función semicontinua inferiormente sobre un conjunto cerrado no vacío, existe al menos un minimizador global. □

(c) Caracterización de optimalidad: *Por el Teorema Fundamental de Wagner-Whitin, existe una solución óptima que satisface:*

$$I_{t-1} \cdot x_t = 0 \quad \forall t = 1, \ldots, N \tag{11}$$

es decir, nunca se ordena cuando hay inventario positivo (zero-inventory ordering property).

Observación 3.3. *La formulación con función indicadora $\delta(x_t)$ hace que el problema sea de* ***optimización no lineal con costos fijos****. Sin embargo, la estructura especial del problema garantiza que podemos encontrar la solución óptima global en tiempo polinomial mediante programación dinámica.*

Observación 3.4. *El supuesto de inventario inicial cero ($I_0 = 0$) no es restrictivo. Si $I_0 > 0$, el problema puede transformarse equivalentemente considerando la demanda neta $d'_1 = \text{máx}(0, d_1 - I_0)$ y procediendo con el algoritmo estándar.*

3.2. Formulación como Programación Entera Mixta

El problema de dimensionamiento de lotes puede formularse como un programa lineal entero mixto (MILP) mediante la introducción de variables binarias que linealizan la función indicadora $\delta(x_t)$.

Definición 3.5 (Formulación MILP del Problema Wagner-Whitin)**.** *Definimos las siguientes* ***variables de decisión****:*

- $x_t \geq 0$*: cantidad a ordenar en el periodo t (variable continua)*
- $I_t \geq 0$*: nivel de inventario al final del periodo t (variable continua)*
- $y_t \in \{0, 1\}$*: variable binaria que indica si se realiza un pedido en el periodo t*

El problema de optimización consiste en:

$$\min_{x,I,y} \quad \sum_{t=1}^{N} [s_t y_t + h_t I_t] \tag{9}$$

$$\text{sujeto a:} \quad I_t = I_{t-1} + x_t - d_t, \quad \forall t = 1, \dots, N \tag{10}$$

$$x_t \leq M_t y_t, \quad \forall t = 1, \dots, N \tag{11}$$

$$I_t \geq 0, \quad x_t \geq 0, \quad \forall t = 1, \dots, N \tag{12}$$

$$y_t \in \{0, 1\}, \quad \forall t = 1, \dots, N \tag{13}$$

$$I_0 = 0 \tag{14}$$

donde $M_t = \sum_{i=t}^{N} d_i$ *es una cota superior para* x_t *en cualquier solución óptima.*

Proposición 3.6 (Equivalencia entre Formulaciones)**.** *Las formulaciones original (8) y MILP (9)-(14) son equivalentes, es decir:*

1. *Tienen el* ***mismo valor óptimo*** *de la función objetivo*
2. *Existe una* ***correspondencia biyectiva*** *entre soluciones óptimas*
3. *En toda solución óptima del MILP se cumple* $y_t = \delta(x_t) = \nVdash_{\{x_t>0\}}$

Demostración de Equivalencia. La equivalencia se establece mediante dos transformaciones:

Transformación del problema original al MILP: Dada una solución factible (x_t, I_t) del problema original, definimos:

$$y_t = \delta(x_t) = \begin{cases} 1 & \text{si } x_t > 0 \\ 0 & \text{si } x_t = 0 \end{cases}$$

Esta tripleta (x_t, I_t, y_t) es factible para el MILP porque:

- Las restricciones (10), (12) y (14) se mantienen directamente
- La restricción (11) se satisface: si $x_t > 0$ entonces $y_t = 1$ y $x_t \leq M_t$; si $x_t = 0$ entonces $y_t = 0$ y $x_t = 0 \leq 0$
- El valor objetivo coincide: $\delta(x_t)s_t = y_t s_t$

Transformación del MILP al problema original: Dada una solución factible (x_t, I_t, y_t) del MILP, la pareja (x_t, I_t) es factible para el problema original porque:

- Las restricciones (7)-(10) se satisfacen directamente
- La función objetivo del problema original evaluada en (x_t, I_t) es:

$$\sum_{t=1}^{N}[\delta(x_t)s_t + h_t I_t] \leq \sum_{t=1}^{N}[y_t s_t + h_t I_t]$$

  pues $\delta(x_t) \leq y_t$ (si $y_t = 0$ entonces $x_t = 0$ por (11), luego $\delta(x_t) = 0$; si $y_t = 1$ entonces $\delta(x_t) \leq 1 = y_t$)

Optimalidad y correspondencia: En una solución óptima del MILP, necesariamente se cumple $\delta(x_t) = y_t$ para todo t, pues de lo contrario existiría algún t con $y_t = 1$ pero $x_t = 0$, y podríamos hacer $y_t = 0$ reduciendo el costo sin violar factibilidad. Por lo tanto, en el óptimo las funciones objetivo coinciden exactamente y las soluciones son equivalentes. □

Observación 3.7 (Mecanismo de la Formulación MILP)**.** *La clave de la equivalencia reside en la restricción (11) $x_t \leq M_t y_t$ que:*

- ***Forza la relación lógica****: Si $x_t > 0$ entonces $y_t = 1$ (pues si $y_t = 0$, la restricción impondría $x_t = 0$)*
- ***Permite la minimización****: El algoritmo de optimización puede .^{el}egir”$y_t = 0$ cuando $x_t = 0$ para ahorrar el costo s_t*
- ***Mantiene factibilidad****: No excluye ninguna solución óptima del problema original*

Observación 3.8 (Optimalidad de la Cota M_t)**.** *La elección $M_t = \sum_{i=t}^{N} d_i$ es* ***óptima*** *porque:*

- *Es* ***suficiente****: En cualquier solución óptima, $x_t \leq \sum_{i=t}^{N} d_i$ debido a que ordenar más que la demanda total restante sería subóptimo*
- *Es* ***ajustada****: Existen soluciones óptimas donde $x_t = \sum_{i=t}^{N} d_i$ (política de ordenar todo al inicio)*

- *Evita* ***relajaciones débiles****: Una M_t más grande debilitaría la relajación lineal del problema*

Ejemplo 3.9 (Ilustración de la Transformación)**.** *Considere una solución óptima del problema original para $N = 3$ con:*

$$x_1 = 50, x_2 = 0, x_3 = 40; \quad I_1 = 30, I_2 = 0, I_3 = 0$$

con demandas $d = [20, 30, 40]$.

La solución equivalente en el MILP es:

$$x_1 = 50, x_2 = 0, x_3 = 40; \quad I_1 = 30, I_2 = 0, I_3 = 0; \quad y_1 = 1, y_2 = 0, y_3 = 1$$

Ambas tienen el mismo costo: $s_1 + s_3 + h_1 \cdot 30$.

Observación 3.10 (Complejidad Computacional)**.** *Aunque equivalentes matemáticamente, las formulaciones difieren computacionalmente:*

- ***Formulación original****: Problema no lineal (NP-duro en general)*
- ***Formulación MILP****: Problema lineal entero mixto, pero con N variables binarias*
- ***Algoritmo Wagner-Whitin****: Explota la estructura específica para resolver en $O(N^2)$*

Para problemas grandes, el algoritmo específico sigue siendo más eficiente que resolver el MILP con métodos generales.

3.3. Formulación como Programación Dinámica

El problema de dimensionamiento de lotes admite una formulación natural mediante programación dinámica, que explota la estructura secuencial de las decisiones a lo largo del horizonte de planificación.

Definición 3.11 (Formulación de Programación Dinámica)**.** *Sea $f_t(I)$ la* ***función de valor óptimo*** *(función de Bellman) que representa el costo mínimo acumulado desde el periodo t hasta el periodo N, dado un nivel de inventario inicial I al comienzo del periodo t.*

La ecuación de Bellman para este problema es:

$$f_t(I) = \min_{\substack{x_t \geq 0 \\ I + x_t \geq d_t}} [C_t(I, x_t) + f_{t+1}(I + x_t - d_t)] \tag{15}$$

*donde el **costo inmediato** $C_t(I, x_t)$ está dado por:*

$$C_t(I, x_t) = h_{t-1}I + \delta(x_t)s_t \tag{16}$$

La condición terminal que cierra la recursión es:

$$f_{N+1}(I) = \begin{cases} 0 & \text{si } I = 0 \\ \infty & \text{si } I > 0 \end{cases} \quad \text{para todo } I \geq 0 \tag{17}$$

Observación 3.12 (Interpretación de la Función de Bellman)**.** *La función $f_t(I)$ encapsula el **principio de optimalidad** de Bellman aplicado a la gestión de inventarios:*

- ***Estado del sistema**: El inventario inicial I al comienzo del periodo t*
- ***Variable de control**: La cantidad a ordenar x_t*
- ***Transición de estado**: $I_{t+1} = I + x_t - d_t$*
- ***Costo inmediato**: $C_t(I, x_t)$ que incluye costos de mantenimiento y setup*
- ***Función de valor futuro**: $f_{t+1}(I + x_t - d_t)$ representa las consecuencias óptimas de la decisión actual*

La ecuación (15) expresa que la política óptima en el estado (t, I) consiste en elegir x_t que minimice la suma del costo inmediato más el costo futuro óptimo.

Proposición 3.13 (Consistencia con el Principio de Optimalidad)**.** *La formulación (15)-(17) satisface el principio de optimalidad de Bellman. Es decir, cualquier subpolítica de una política óptima es a su vez óptima para el subproblema correspondiente.*

Demostración. Sea $\pi^* = (x_1^*, x_2^*, \ldots, x_N^*)$ una política óptima para el problema completo, y sea I_t^* la trayectoria de inventario resultante. Considere el subproblema desde el periodo t hasta N con inventario inicial I_t^*.

Si existiera una política $\pi' = (x_t', \ldots, x_N')$ con costo menor que $(x_t^*, \ldots, x_N^*)$ para este subproblema, entonces la política $(x_1^*, \ldots, x_{t-1}^*, x_t', \ldots, x_N')$ tendría menor costo total que π^*, contradiciendo la optimalidad de π^*. $\square$

Observación 3.14 (Complejidad Computacional de la Implementación Directa)**.** *Aunque la formulación (15)-(17) es conceptualmente correcta, su implementación computacional directa presenta serias dificultades:*

- ***Espacio de estados continuo**: La variable de estado I toma valores en $\mathbb{R}_{\geq 0}$, requiriendo discretización*
- ***Dimensionalidad**: Para cada periodo t, debemos computar $f_t(I)$ para múltiples valores de I*
- ***Complejidad exponencial**: Sin estructura adicional, el número de estados a considerar crece exponencialmente con N*
- ***Problemas de discretización**: La elección del paso de discretización afecta precisión y costo computacional*

En el peor caso, si discretizamos el inventario en K niveles, la complejidad sería $O(K^2 N)$, que puede ser prohibitiva para K grande.

Ejemplo 3.15 (Ilustración de la Complejidad)**.** *Para un problema con $N = 12$ periodos y discretizando el inventario en $K = 100$ niveles:*

- *Número de estados:* $12 \times 100 = 1,200$
- *Evaluaciones de la ecuación de Bellman:* $\approx 100^2 \times 12 = 120,000$
- *Cálculo de mínimos sobre espacios continuos aproximados*

Esto contrasta con la complejidad $O(N^2)$ del algoritmo Wagner-Whitin que explota las propiedades estructurales.

Teorema 3.16 (Reducción del Espacio de Estados)**.** *Gracias a la **propiedad de cero inventarios** (Teorema 1), podemos restringir la atención a estados donde $I = 0$ al comienzo de cada periodo. Esto reduce el espacio de estados de dimensión continua a uno discreto finito.*

Demostración. Por el Teorema 1, existe una solución óptima que satisface $I_{t-1} \cdot x_t = 0$ para todo t. Por lo tanto, en cada periodo t, o bien $I_{t-1} = 0$ (inventario cero) o bien $x_t = 0$ (no se ordena).

En particular, al comienzo de cualquier periodo donde se realice un pedido, debemos tener $I = 0$. Como las decisiones de ordenar ocurren solo en puntos de regeneración, basta con considerar $f_t(0)$ para todo t. □

Observación 3.17 (Simplificación Fundamental)**.** *Los teoremas fundamentales de Wagner-Whitin permiten las siguientes simplificaciones cruciales:*

1. ***Reducción del espacio de estados**: Solo necesitamos considerar $f_t(0)$*

2. ***Patrón de pedidos**: Cada pedido cubre demandas consecutivas*

3. ***Horizontes de planificación**: El problema se descompone en subproblemas independientes*

Estas propiedades reducen la complejidad de $O(K^2N)$ a $O(N^2)$, haciendo el problema computacionalmente tratable incluso para N grande.

Algorithm 1 Programación Dinámica Simplificada por Wagner-Whitin

---

Inicializar $F(0) = 0$ ▷ Costo óptimo hasta el periodo 0
for $t = 1$ to N **do**
    $F(t) = \infty$
    for $j = 0$ to $t - 1$ **do**
        Calcular costo de ordenar en $j+1$ para cubrir periodos $j+1$ hasta t
        $costo = F(j) + s_{j+1} + \sum_{k=j+1}^{t-1} \sum_{l=k+1}^{t} h_l d_k$
        if $costo < F(t)$ **then**
            $F(t) = costo$
            $P(t) = j$ ▷ Registrar punto de regeneración
        end if
    end for
end for

Observación 3.18 (Equivalencia con la Formulación Original)**.** *La formulación simplificada $F(t) = \min_{0 \le j < t}[F(j) + s_{j+1} + H(j,t)]$ es equivalente a resolver la ecuación de Bellman (15) bajo las propiedades estructurales de Wagner-Whitin, donde $H(j,t)$ representa el costo de mantener inventario al cubrir demandas de $j+1$ a t con una orden en $j+1$.*

4. Resultados del articulo base

Los siguientes teoremas, demostrados originalmente por Wagner y Whitin, establecen propiedades estructurales de la solución óptima que reducen drásticamente el espacio de búsqueda.

Teorema 4.1 (Propiedad de Regeneración o Cero-Inventario)**.** *En todo problema de dimensionamiento de lotes con demanda dinámica, existe al menos una solución óptima que satisface la condición de cero inventarios:*

$$I_{t-1} \cdot x_t = 0 \quad \textit{para todo } t = 1, \ldots, N \tag{12}$$

Es decir, en cada periodo t, o bien el inventario inicial I_{t-1} es cero, o bien no se realiza pedido ($x_t = 0$), pero nunca ambos son positivos simultáneamente.

Demostración. Supongamos que existe una solución óptima $\{(x_t^*, I_t^*)\}_{t=1}^N$ que viola la propiedad para algún periodo t, es decir, $I_{t-1}^* > 0$ y $x_t^* > 0$. Construiremos una solución alternativa con costo menor o igual.

Consideremos transferir una cantidad $\Delta = \text{mín}(I_{t-1}^*, x_t^*)$ del pedido del periodo t al periodo $t-1$. Definamos:

$$\begin{aligned} x'_{t-1} &= x^*_{t-1} + \Delta \\ x'_t &= x^*_t - \Delta \\ I'_{t-1} &= I^*_{t-1} - \Delta \end{aligned}$$

Manteniendo las demás variables iguales. El cambio en el costo total es:

$$\Delta C = -h_{t-1}\Delta + [\delta(x'_{t-1}) - \delta(x^*_{t-1})]s_{t-1} + [\delta(x'_t) - \delta(x^*_t)]s_t$$

Analizamos los casos:
Caso 1: Si $x^*_{t-1} > 0$, entonces $\delta(x'_{t-1}) = \delta(x^*_{t-1}) = 1$, y:

- Si $x^*_t - \Delta > 0$, entonces $\delta(x'_t) = \delta(x^*_t) = 1$, luego $\Delta C = -h_{t-1}\Delta < 0$
- Si $x^*_t - \Delta = 0$, entonces $\delta(x'_t) = 0$, $\delta(x^*_t) = 1$, luego $\Delta C = -h_{t-1}\Delta - s_t < 0$

Caso 2: Si $x^*_{t-1} = 0$, entonces $\delta(x'_{t-1}) = 1$, $\delta(x^*_{t-1}) = 0$, y:

- Si $x^*_t - \Delta > 0$, entonces $\Delta C = -h_{t-1}\Delta + s_{t-1}$

- Si $x_t^* - \Delta = 0$, entonces $\Delta C = -h_{t-1}\Delta + s_{t-1} - s_t$

En todos los casos, podemos elegir Δ tal que $\Delta C \leq 0$. Repitiendo este proceso, eventualmente eliminamos todas las violaciones de la propiedad sin aumentar el costo. $\square$

Ejemplo 4.2 (Ilustración de la Propiedad)**.** *Considere un problema con* $N = 3$*, demandas* $d = [20, 30, 40]$*, costos de setup* $s_t = 100$*, y costos de mantenimiento* $h_t = 1$*.*

Solución que viola la propiedad:

$$\begin{aligned} &x_1 = 90, \quad I_1 = 70 \\ &x_2 = 30, \quad I_2 = 70 \quad \textbf{\textit{VIOLACIÓN:}}\ I_1 > 0 \text{ y } x_2 > 0 \\ &x_3 = 0, \quad I_3 = 30 \end{aligned}$$

Costo total: $100 + 100 + 1 \cdot 70 + 1 \cdot 70 + 1 \cdot 30 = 370$

Solución equivalente que cumple la propiedad:

$$\begin{aligned} x_1 &= 120, \quad I_1 = 100 \\ x_2 &= 0, \quad I_2 = 70 \\ x_3 &= 0, \quad I_3 = 30 \end{aligned}$$

Costo total: $100 + 1 \cdot 100 + 1 \cdot 70 + 1 \cdot 30 = 300$

O alternativamente:

$$\begin{aligned} x_1 &= 50, \quad I_1 = 30 \\ x_2 &= 0, \quad I_2 = 0 \\ x_3 &= 40, \quad I_3 = 0 \end{aligned}$$

Costo total: $100 + 100 + 1 \cdot 30 = 230$ ***(óptimo)***

Proposición 4.3 (Existencia de Múltiples Soluciones Óptimas)**.** *Pueden existir múltiples soluciones óptimas, pero al menos una cumple la propiedad de cero inventarios. Las soluciones que violan la propiedad son* ***dominadas*** *o* ***equivalentes*** *a alguna que la cumple.*

Análisis de Multiplicidad. Consideremos dos casos:

Caso 1: Soluciones estructuralmente diferentes Pueden existir políticas óptimas con patrones de pedidos diferentes pero mismo costo. Por ejemplo, con $d = [10, 10]$, $s_t = 50$, $h_t = 1$:

- Política A: $x_1 = 20$ (cubre ambos periodos), costo: $50 + 1 \cdot 10 = 60$

- Política B: $x_1 = 10$, $x_2 = 10$, costo: $50 + 50 = 100$ **(no óptima)**

Caso 2: Soluciones con misma estructura pero diferentes cantidades Cuando los costos de mantenimiento son cero ($h_t = 0$), múltiples políticas pueden ser óptimas. Por ejemplo, con $d = [10, 10]$, $s_t = 50$, $h_t = 0$:

- Cualquier política que ordene 20 unidades en algún subconjunto de periodos tiene costo 50

- Pero solo las que cumplen cero inventarios evitan soluciones redundantes

□

Observación 4.4 (Implicaciones Computacionales)**.** *La propiedad de cero inventarios reduce drásticamente el espacio de búsqueda:*

- ***Sin la propiedad****: Número exponencial de políticas posibles ($O(2^N)$)*

- ***Con la propiedad****: Solo $O(N^2)$ combinaciones a evaluar*

- ***Poda de soluciones****: Podemos ignorar políticas que violan la propiedad sin perder optimalidad*

Corolario 4.5 (Caracterización de Soluciones Óptimas)**.** *Toda solución óptima puede transformarse en una que cumple la propiedad de cero inventarios mediante reasignación de pedidos sin aumentar el costo. Las soluciones que violan persistentemente la propiedad son estrictamente subóptimas a menos que $h_t = 0$ para todo t.*

Ejemplo 4.6 (Caso Degenerado con $h_t = 0$)**.** *Con $d = [10, 10]$, $s_t = 50$, $h_t = 0$:*

$$\begin{aligned}
&\textit{Solución 1: } x_1 = 20, x_2 = 0 \quad (\textit{cumple propiedad}) \\
&\textit{Solución 2: } x_1 = 15, x_2 = 5 \quad (\textit{viola propiedad}) \\
&\textit{Solución 3: } x_1 = 10, x_2 = 10 \quad (\textit{cumple propiedad})
\end{aligned}$$

Todas tienen costo 50, pero las Soluciones 1 y 3 son preferibles por simplicidad.

Teorema 4.7 (Patrón de Pedidos o Satisfacción Exacta de Demanda)**.** *Existe una solución óptima tal que cada pedido $x_t > 0$ cubre exactamente la demanda de un número entero de periodos consecutivos. Formalmente, para cada t con $x_t > 0$, existe un periodo $k \geq t$ tal que:*

$$x_t = \sum_{j=t}^{k} d_j \tag{13}$$

y el inventario se agota completamente al final del periodo k ($I_k = 0$).

Demostración. Supongamos que existe una solución óptima que viola esta propiedad. Entonces, existe algún periodo t con $x_t > 0$ tal que el pedido no cubre exactamente la demanda de periodos consecutivos. Esto implica que existe al menos un periodo intermedio m (con $t < m \leq N$) donde:

- $I_{m-1} > 0$ (inventario positivo al inicio de m)
- $x_m > 0$ (se realiza un pedido en m)

Pero esta situación viola directamente la **Propiedad de Cero Inventarios** del Teorema 1, ya que tenemos $I_{m-1} \cdot x_m > 0$.

Para construir una solución que cumpla la propiedad, consideremos cualquier pedido $x_t > 0$ en la solución óptima. Sea k el último periodo cuya demanda es satisfecha total o parcialmente por x_t. Entonces, podemos redefinir:

$$x'_t = \sum_{j=t}^{k} d_j$$

y ajustar los pedidos posteriores en consecuencia. Este reajuste no aumenta el costo total porque:

1. **Costos de setup**: El número de pedidos permanece igual o disminuye
2. **Costos de mantenimiento**: Eliminamos situaciones de inventario positivo con pedidos, reduciendo posiblemente los costos de almacenamiento
3. **Factibilidad**: Se sigue satisfaciendo toda la demanda

Por lo tanto, existe una solución óptima que satisface el patrón de satisfacción exacta de demanda. □

Ejemplo 4.8 (Ilustración del Teorema). *Considere un problema con $N = 4$, demandas $d = [20, 30, 40, 25]$, $s_t = 100$, $h_t = 1$.*

Solución que viola el patrón:

$$\begin{aligned} x_1 &= 50, \quad I_1 = 30 \\ x_2 &= 45, \quad I_2 = 45 \quad \textbf{\textit{VIOLACIÓN}} \\ x_3 &= 20, \quad I_3 = 25 \\ x_4 &= 0, \quad I_4 = 0 \end{aligned}$$

Aquí $x_2 = 45$ no cubre exactamente la demanda de periodos consecutivos.

Transformación a solución que cumple el patrón:

$$\begin{aligned} x_1 &= 50, \quad I_1 = 30 \\ x_2 &= 70, \quad I_2 = 70 \quad \textit{(cubre periodos 2-3)} \\ x_3 &= 0, \quad I_3 = 30 \\ x_4 &= 25, \quad I_4 = 0 \end{aligned}$$

O mejor aún:

$$\begin{aligned} x_1 &= 115, \quad I_1 = 95 \quad \textit{(cubre todos los periodos)} \\ x_2 &= 0, \quad I_2 = 65 \\ x_3 &= 0, \quad I_3 = 25 \\ x_4 &= 0, \quad I_4 = 0 \end{aligned}$$

Costo: $100 + 1 \cdot 95 + 1 \cdot 65 + 1 \cdot 25 = 285$

Proposición 4.9 (Consecuencias del Patrón de Pedidos). *El teorema implica que en una solución óptima:*

1. *Los* ***puntos de regeneración*** *(periodos con $I_t = 0$) dividen el horizonte en intervalos independientes*

2. *Cada intervalo $[t, k]$ se abastece completamente con un solo pedido en t*

3. *El problema de N periodos se reduce a decidir en qué periodos ordenar*

Demostración Alternativa por Optimalidad Local. Consideremos cualquier política óptima. Fijemos un periodo t con $x_t > 0$. Sea k el último periodo tal que $I_k > 0$ y $I_{k+1} = 0$ (el próximo punto de regeneración).

Si $x_t \neq \sum_{j=t}^{k} d_j$, entonces existen dos casos:

Caso 1: $x_t > \sum_{j=t}^{k} d_j$. Esto implica que sobra inventario después de k, pero $I_{k+1} = 0$ por definición de k, contradicción.

Caso 2: $x_t < \sum_{j=t}^{k} d_j$. Esto implica que necesitamos pedidos adicionales entre t y k, pero entonces existiría algún $m \in (t, k]$ con $I_{m-1} > 0$ y $x_m > 0$, violando el Teorema 1.

Por lo tanto, debe cumplirse $x_t = \sum_{j=t}^{k} d_j$. $\square$

Observación 4.10 (Implicaciones Computacionales)**.** *Este teorema reduce drásticamente el espacio de búsqueda:*

- ***Sin el teorema****: Número exponencial de combinaciones de cantidades a ordenar*
- ***Con el teorema****: Solo $O(N^2)$ posibles políticas (elegir puntos de regeneración)*
- ***Algoritmo eficiente****: Permite el desarrollo del algoritmo Wagner-Whitin con complejidad $O(N^2)$*

Ejemplo 4.11 (Caso con Múltiples Soluciones Óptimas)**.** *Considere $d = [10, 10, 10]$, $s_t = 50$, $h_t = 1$. Existen múltiples soluciones óptimas:*

Solución 1: *Un pedido al inicio*

$$x_1 = 30, \quad I_1 = 20, \quad I_2 = 10, \quad I_3 = 0$$
$$\textit{Costo: } 50 + 1 \cdot 20 + 1 \cdot 10 = 80$$

Solución 2: *Dos pedidos estratégicos*

$$x_1 = 20, \quad I_1 = 10, \quad I_2 = 0$$
$$x_3 = 10, \quad I_3 = 0$$
$$\textit{Costo: } 50 + 50 + 1 \cdot 10 = 110 \quad \textit{(no óptima)}$$

Solución 3: *Pedidos en cada periodo*

$$x_1 = 10, \quad I_1 = 0$$
$$x_2 = 10, \quad I_2 = 0$$
$$x_3 = 10, \quad I_3 = 0$$
$$\textit{Costo: } 50 + 50 + 50 = 150 \quad \textit{(no óptima)}$$

Todas las soluciones óptimas cumplen el patrón de satisfacción exacta.

Corolario 4.12 (Formulación Recursiva Simplificada)**.** *El costo mínimo $F(t)$ para los primeros t periodos satisface:*

$$F(t) = \min_{0 \leq j < t} \left[F(j) + s_{j+1} + \sum_{k=j+1}^{t-1} \sum_{l=k+1}^{t} h_l d_k \right]$$

donde j representa el último punto de regeneración antes de t.

Teorema 4.13 (Horizonte de Planificación)**.** *Si al resolver el problema para los primeros t periodos se encuentra que el último pedido ocurre en el periodo $j^* \leq t$, entonces la solución óptima para el problema completo puede obtenerse concatenando la solución óptima para los primeros $j^* - 1$ periodos (considerados independientemente) con la solución para los periodos restantes.*

Demostración. Sea $F(t)$ el costo mínimo para los primeros t periodos. Por la propiedad de regeneración, existe un último periodo de pedido $j^* \leq t$ que satisface la demanda hasta t. El costo total sería $F(j^* - 1) + s_{j^*} + \sum_{k=j^*}^{t-1} \sum_{l=k+1}^{t} h_l d_k$. Si este es el mínimo global para el subproblema de t periodos, cualquier extensión a periodos futuros no modificará la optimalidad de esta decisión para los primeros $j^* - 1$ periodos. □

Corolario 4.14 (Formulación Recursiva Simplificada)**.** *El costo mínimo para los primeros t periodos puede expresarse como:*

$$F(t) = \min_{0 \leq j < t} \left[F(j) + s_{j+1} + \sum_{k=j+1}^{t-1} \sum_{l=k+1}^{t} h_l d_k \right] \tag{18}$$

con $F(0) = 0$.

Demostración. Usaremos los Teoremas 2 y 3, que establecen la existencia de una solución óptima con puntos de regeneración donde el inventario se agota.

Paso 1: Identificación de la estructura óptima

Por el **Teorema 3** (Patrón de Pedidos), en una solución óptima cada pedido $x_{j+1} > 0$ cubre exactamente la demanda de periodos consecutivos $j + 1$ hasta k para algún $k \geq j + 1$.

Para el subproblema de los primeros t periodos, sea j el último punto de regeneración antes de t (es decir, $I_j = 0$). Entonces, el último pedido ocurre en el periodo $j+1$ y cubre los periodos desde $j+1$ hasta t.

Paso 2: Descomposición del costo

El costo total para los primeros t periodos se descompone en:

1. **Costo óptimo hasta el periodo** j: $F(j)$

2. **Costo de setup del pedido en** $j+1$: s_{j+1}

3. **Costo de mantenimiento de inventario**: Para cada periodo k desde $j+1$ hasta $t-1$, la demanda d_k debe almacenarse desde el periodo $k+1$ hasta t

Paso 3: Cálculo del costo de mantenimiento

Para cada $k \in \{j+1, \ldots, t-1\}$, la demanda d_k incurre en costos de mantenimiento en los periodos $l = k+1, \ldots, t$. El costo unitario de mantenimiento en el periodo l es h_l.

Por lo tanto, el costo total de mantenimiento para la demanda d_k es:

$$d_k \sum_{l=k+1}^{t} h_l$$

Sumando sobre todos los k desde $j+1$ hasta $t-1$, obtenemos:

$$\sum_{k=j+1}^{t-1} \left(d_k \sum_{l=k+1}^{t} h_l \right) = \sum_{k=j+1}^{t-1} \sum_{l=k+1}^{t} h_l d_k$$

Paso 4: Formulación recursiva

Dado que j puede ser cualquier punto de regeneración entre 0 y $t-1$, debemos minimizar sobre todas las posibles elecciones de j:

$$F(t) = \min_{0 \leq j < t} \left[F(j) + s_{j+1} + \sum_{k=j+1}^{t-1} \sum_{l=k+1}^{t} h_l d_k \right]$$

Paso 5: Condición inicial

Para $t = 0$, no hay periodos que planificar, por lo que $F(0) = 0$.

Paso 6: Verificación de optimalidad

Esta formulación garantiza la optimalidad porque:

- Considera todos los posibles puntos de regeneración j
- Cada término en el mínimo corresponde a una política factible
- Por el principio de optimalidad de Bellman, la solución óptima debe tener esta estructura

□

Ejemplo 4.15 (Aplicación de la Formulación Recursiva)**.** *Considere* $N = 3$, $d = [20, 30, 40]$, $s_t = 100$, $h_t = 1$.

Cálculo de $F(1)$***:***

$$F(1) = \min_{0 \le j < 1} [F(j) + s_{j+1} + \textit{costo mantenimiento}]$$

Solo $j = 0$*:*

$$F(1) = F(0) + s_1 + 0 = 0 + 100 + 0 = 100$$

Cálculo de $F(2)$***:***

$$F(2) = \min \begin{cases} j = 0 : F(0) + s_1 + \sum_{k=1}^{1} \sum_{l=2}^{2} h_l d_k = 0 + 100 + 1 \cdot 20 = 120 \\ j = 1 : F(1) + s_2 + 0 = 100 + 100 + 0 = 200 \end{cases}$$

$F(2) = 120$

Cálculo de $F(3)$***:***

$$F(3) = \min \begin{cases} j = 0 : F(0) + s_1 + \sum_{k=1}^{2} \sum_{l=k+1}^{3} h_l d_k = 0 + 100 + (1 \cdot 20 + 1 \cdot 20 + 1 \cdot 30) = 170 \\ j = 1 : F(1) + s_2 + \sum_{k=2}^{2} \sum_{l=3}^{3} h_l d_k = 100 + 100 + 1 \cdot 30 = 230 \\ j = 2 : F(2) + s_3 + 0 = 120 + 100 + 0 = 220 \end{cases}$$

$F(3) = 170$

Observación 4.16 (Interpretación de los Índices)**.** *En la doble sumatoria:*

- k*: Periodo cuya demanda* d_k *se almacena*
- l*: Periodos en los que* d_k *permanece en inventario*
- *El rango* $k = j + 1$ *hasta* $t - 1$ *excluye el periodo* t *porque su demanda no se almacena*

- *El rango $l = k+1$ hasta t captura todos los periodos de almacenamiento*

Proposición 4.17 (Equivalencia con la Formulación Original)**.** *La formulación recursiva (18) es equivalente a la ecuación de Bellman original bajo las propiedades de Wagner-Whitin, pero con complejidad reducida de $O(N^2)$ en lugar de $O(2^N)$.*

Demostración. Sea $f_t(I)$ la función de Bellman original. Por el Teorema 1, en una solución óptima $I_t = 0$ en los puntos de regeneración. Por el Teorema 2, entre puntos de regeneración consecutivos j y t, el pedido en $j+1$ cubre exactamente la demanda de $j+1$ a t.

La formulación (18) explota estas propiedades considerando solo los estados donde $I = 0$, reduciendo el espacio de estados de dimensión continua a uno discreto finito.

La optimalidad se mantiene porque cualquier política que viole estas propiedades puede transformarse en una que las cumple sin aumentar el costo. $\square$

5. Algoritmo Wagner-Whitin: Implementación Computacional

Tras haber establecido los fundamentos teóricos y demostrado las propiedades estructurales que caracterizan las soluciones óptimas del problema de dimensionamiento de lotes, presentamos ahora el algoritmo Wagner-Whitin en su forma canónica. Este algoritmo representa la materialización práctica de los teoremas fundamentales demostrados, transformando un problema de optimización combinatorio complejo en un procedimiento computacional eficiente y elegante.

El algoritmo Wagner-Whitin constituye el núcleo de esta investigación, sintetizando los conceptos de programación dinámica, puntos de regeneración y horizontes de planificación en un método sistemático que resuelve el problema en tiempo polinomial. Su importancia histórica radica en ser el primer algoritmo exacto para el problema de dimensionamiento de lotes con demanda dinámica, estableciendo un paradigma que ha inspirado numerosas extensiones durante más de seis décadas.

5.1. Formulación Recursiva y Estructura del Algoritmo

El algoritmo se fundamenta en la formulación recursiva derivada del Corolario 1, que explota las propiedades de cero inventarios y satisfacción exacta de demanda:

$$F(t) = \min_{0 \leq j < t} \left[F(j) + s_{j+1} + \sum_{k=j+1}^{t-1} \sum_{l=k+1}^{t} h_l d_k \right] \tag{19}$$

donde $F(t)$ representa el costo mínimo acumulado para los primeros t periodos.

La elegancia del algoritmo reside en su interpretación geométrica: el horizonte temporal se divide en intervalos definidos por puntos de regeneración consecutivos, donde cada intervalo $[j+1, t]$ es abastecido por un único pedido en $j+1$ que cubre exactamente la demanda acumulada del intervalo.

Algorithm 2 Algoritmo Wagner-Whitin

Require: Demandas d_t, costos de setup s_t, costos de mantener h_t para $t = 1, \dots, N$

Ensure: Política óptima de pedidos $\{x_t\}_{t=1}^{N}$ y costo total mínimo $F(N)$

Inicializar $F(0) = 0$, $P(0) = 0$ ▷ $P(t)$ almacena el último punto de regeneración
for $t = 1$ to N **do**
  $F(t) \leftarrow \infty$ ▷ Inicializar con valor grande
  for $j = 0$ to $t - 1$ **do**
    Calcular costo de mantener: $H \leftarrow \sum_{k=j+1}^{t-1} \sum_{l=k+1}^{t} h_l d_k$
    $costo \leftarrow F(j) + s_{j+1} + H$
    if $costo < F(t)$ **then**
      $F(t) \leftarrow costo$
      $P(t) \leftarrow j$ ▷ Registrar que el pedido en $j+1$ cubre hasta t
    end if
  end for
end for
$\{x_t\}_{t=1}^{N} \leftarrow$ reconstruirPolítica(P, d) ▷ Backtracking
return $F(N)$, $\{x_t\}_{t=1}^{N}$

Observación 5.1 (Interpretación del Algoritmo). *El algoritmo opera en dos fases fundamentales:*

1. ***Fase forward**: Calcula iterativamente $F(t)$ para $t = 1, \dots, N$, almacenando en $P(t)$ la decisión óptima del último punto de regeneración.*

2. ***Fase de backtracking**: Reconstruye la política óptima recorriendo $P(t)$ desde $t = N$ hasta $t = 0$.*

Cada evaluación en el lazo interno corresponde a probar si el último pedido antes de t debería ubicarse en $j + 1$, cubriendo los periodos $j + 1$ hasta t.

5.2. Análisis de Complejidad Computacional

La implementación básica del algoritmo tiene complejidad $O(N^3)$, pero mediante precomputación inteligente es posible alcanzar $O(N^2)$.

Teorema 5.2 (Optimización de Complejidad). *El algoritmo Wagner-Whitin puede implementarse con complejidad temporal $O(N^2)$ mediante precomputación de la matriz de costos de mantenimiento.*

Demostración. Definamos la matriz de costos de mantenimiento $H(j, t)$ como:

$$H(j,t) = \sum_{k=j}^{t-1} \sum_{l=k+1}^{t} h_l d_k$$

Esta matriz puede precomputarse en $O(N^2)$ usando la relación de recurrencia:

$$H(j,t) = H(j,t-1) + d_{t-1} \sum_{l=t}^{t} h_l \quad \text{para } j < t$$

con condiciones iniciales $H(j, j) = 0$ para todo j.

Una vez precomputada H, cada evaluación del costo en el algoritmo principal toma tiempo constante, reduciendo la complejidad total de $O(N^3)$ a $O(N^2)$. □

Proposición 5.3 (Reconstrucción de la Política Óptima). *La política óptima se reconstruye mediante el siguiente procedimiento de backtracking:*

```
function RECONSTRUIRPOLÍTICA(P, d)
    t ← N, x ← [0, ..., 0]
```

Algorithm 3 Algoritmo Wagner-Whitin Optimizado

Require: Demandas d_t, costos s_t, h_t para $t = 1, \ldots, N$
Ensure: Política óptima y costo mínimo

```
Precomputar H(j,t) para 0 ≤ j < t ≤ N                ▷ O(N^2)
F(0) ← 0, P(0) ← 0
for t = 1 to N do
    F(t) ← ∞
    for j = 0 to t − 1 do
        costo ← F(j) + s_{j+1} + H(j + 1, t)
        if costo < F(t) then
            F(t) ← costo, P(t) ← j
        end if
    end for
end for
return reconstruirPolítica(P, d), F(N)
```

```
    while t > 0 do
        j ← P(t)
        x_{j+1} ← Σ_{i=j+1}^{t} d_i              ▷ Pedido en j + 1 cubre hasta t
        t ← j                ▷ Retroceder al punto de regeneración anterior
    end while
    return x
end function
```

5.3. Ejemplo Computacional Detallado

Para ilustrar el funcionamiento del algoritmo, consideremos el ejemplo original de Wagner y Whitin (1958) con $N = 12$ periodos, cuyos datos se presentan en la Tabla 1.

Asumiendo $h_t = 1$ para todo t, la aplicación del algoritmo produce:

Cuadro 1: Datos del problema original de Wagner-Whitin (1958)

Mes t	d_t	s_t	Mes t	d_t	s_t
1	69	85	7	34	105
2	29	102	8	67	86
3	36	102	9	45	119
4	61	101	10	67	110
5	61	98	11	79	98
6	26	114	12	56	114

Ejemplo 5.4 (Ejecución Paso a Paso)**. *Cálculo de* $F(1)$ *a* $F(3)$*:***

$$F(1) = F(0) + s_1 + H(1,1) = 0 + 85 + 0 = 85$$

$$F(2) = \text{mín} \begin{cases} F(0) + s_1 + H(1,2) = 0 + 85 + 29 = 114 \\ F(1) + s_2 + H(2,2) = 85 + 102 + 0 = 187 \end{cases} = 114$$

$$F(3) = \text{mín} \begin{cases} F(0) + s_1 + H(1,3) = 0 + 85 + (29 + 36) = 150 \\ F(1) + s_2 + H(2,3) = 85 + 102 + 36 = 223 \\ F(2) + s_3 + H(3,3) = 114 + 102 + 0 = 216 \end{cases} = 150$$

Continuando este proceso hasta $t = 12$, obtenemos la política óptima completa.

Cuadro 2: Política óptima para el problema de 12 periodos

Periodo de pedido	Periodos cubiertos	Cantidad x_t	Costo acumulado
1	1-2	98	114
3	3-4	97	254
5	5-7	121	473
8	8-9	112	635
10	10	67	772
11	11-12	135	864

5.4. Análisis de la Solución Óptima

La política óptima obtenida ilustra perfectamente las propiedades estructurales demostradas:

- **Propiedad de cero inventarios**: Cada pedido ocurre cuando el inventario es cero
- **Satisfacción exacta**: Cada x_t cubre exactamente la demanda de bloques consecutivos
- **Eficiencia computacional**: El algoritmo evalúa solo $O(N^2)$ combinaciones en lugar de $O(2^N)$

El costo total óptimo es $F(12) = 864$, que representa una reducción significativa comparada con políticas ingenuas como *lote por lote* (costo: $85 + 102 + 102 + 101 + 98 + 114 + 105 + 86 + 119 + 110 + 98 + 114 = 1234$).

5.5. Extensiones y Limitaciones Prácticas

Aunque el algoritmo Wagner-Whitin resuelve óptimamente el problema básico, en la práctica surgen extensiones importantes:

- **Restricciones de capacidad**: Límites en x_t que pueden hacer el problema NP-duro
- **Plazos de entrega**: Tiempos entre la orden y la recepción
- **Descuentos por cantidad**: Costos unitarios variables
- **Múltiples productos**: Interacciones en el espacio de almacenamiento
- **Incertidumbre**: Versiones estocásticas del problema

Sin embargo, el algoritmo original sigue siendo fundamental como punto de referencia teórico y como componente básico en sistemas más complejos de planificación de requerimientos de materiales (MRP).

Observación 5.5 (Vigencia del Algoritmo)**.** *Más de seis décadas después de su publicación, el algoritmo Wagner-Whitin mantiene su relevancia:*

- ***Educación**: Enseña principios fundamentales de programación dinámica*
- ***Investigación**: Sirve como benchmark para nuevos algoritmos*
- ***Práctica industrial**: Se utiliza en sistemas de gestión de inventarios*
- ***Extensiones modernas**: Inspira algoritmos para problemas más complejos*

Referencias

Referencias

[1] Wagner, H. M. & Whitin, T. M. (1958). Dynamic Version of the Economic Lot Size Model. *Management Science*, 5(1), 89-96.

[2] Richter, K. & Sombrutzki, M. (2001). The reverse Wagner/Whitin model with variable manufacturing and remanufacturing cost. *International Journal of Production Economics*, 71(1-3), 1-12.

[3] Bellman, R. (1957). *Dynamic Programming*. Princeton University Press.

[4] Bravo, F. & Vidal, C. J. (2023). Stochastic Lot-Sizing: Recent Advances and Future Directions. *European Journal of Operational Research*, 305(2), 501-520.