

ALGORITMO RECOCIDO SIMULADO–SECCIÓN
DORADA PARA EL PROBLEMA DE
REAPROVISIONAMIENTO MULTIPRODUCTO
CON DEMANDA ESTOCÁSTICA

SIMULATED ANNEALING–GOLDEN SECTION
ALGORITHM FOR THE MULTIPRODUCT
REPLENISHMENT PROBLEM WITH STOCHASTIC
DEMAND

SALVADOR HERNÁNDEZ GONZÁLEZ *

MIGUEL A. GUTIÉRREZ ANDRADE[†]

SERGIO DE LOS COBOS SILVA[‡]

*Received: 18 Nov 2008; Revised: 3 May 2009;
Accepted: 30 Nov 2009*

*Depto. de Investigación de Operaciones, Facultad de Ingeniería, Universidad Nacional Autónoma de México, México. E-Mail: shdezg@yahoo.com.mx

[†]Departamento de Ingeniería Eléctrica, Universidad Autónoma Metropolitana-Iztapalapa, Av. Michoacán y La Purísima s/n, Col. Vicentina, Del. Iztapalapa, México D.F., C.P. 09340, México. E-Mail: gamma@xanum.uam.mx

[‡]Misma dirección que/same address as: M.A. Gutiérrez. E-Mail: cobos@xanum.uam.mx

Resumen

El problema de reaprovisionamiento multiproducto ha sido estudiado por más de 30 años y existen varios algoritmos heurísticos y exactos para determinar la frecuencia de pedidos y el ciclo básico de tiempo. En años recientes se ha considerado el modelo con demanda estocástica; suponiendo que el comportamiento de la demanda se ajusta a una función de probabilidad normal, se puede obtener una ecuación de costo del tipo mixto-entero-no lineal, para la cual solamente ha sido reportada, en la literatura, una técnica heurística. En este trabajo, se implementa un algoritmo de recocido simulado-sección dorada para el problema de reaprovisionamiento multiproducto considerando una demanda con distribución de probabilidad normal y se compara su desempeño contra la técnica heurística reportada en la literatura. Los resultados muestran que el nuevo algoritmo obtiene costos más bajos.

Palabras clave: Problemas de inventario multiproducto, problema de reaprovisionamiento conjunto, recocido simulado, sección dorada.

Abstract

The joint replenishment problem (JRP) has been studied for over 30 years and there are both heuristic and exact algorithms to determine the frequency of orders and fundamental cycle; in recent years it has been considered the model with stochastic demand. If we assume a behavior of normal distribution for the demand, we may obtain a non linear mixed-integer programming for costs, for which only is reported one heuristic solving method. In this paper we propose a simulated annealing algorithm with golden section for one-dimensional search in order to solve the JRP considering a normal distribution demand. Its performance is compared with the reported heuristic method. The results showed that the new algorithm obtains lower costs.

Keywords: Multi-item inventory problem, joint replenishment problem, simulated annealing, golden section.

Mathematics Subject Classification: 90C59, 90B05.

1 Introducción

En la mayoría de las empresas de manufactura, servicios y distribución, el inventario es un factor básico en la medición del desempeño de su rentabilidad. Frecuentemente las necesidades de controlar el inventario van más allá de un producto o de un solo proveedor.

Una tendencia común, hoy en día, consiste en reducir el número de entidades que suministran los recursos en una empresa, de forma que, el

distribuidor proporcione la mayor cantidad de suministros, la adquisición de productos es por grupos o familias en lugar de hacerlo por separado. Otra situación muy similar se presenta cuando es necesario envasar un mismo producto en distintos tipos de contenedores. En ambos casos, se debe determinar la frecuencia de envasado o de pedido de cada producto, así como, el tiempo entre cada solicitud o activación. Este problema se conoce como problema de reaprovisionamiento conjunto (Goyal [7]) y existen dos versiones: determinista o estocástica.

Para el caso del sistema con múltiples productos y demanda estocástica se han desarrollado varios algoritmos para obtener el ciclo básico de tiempo T y la frecuencia de pedido de cada producto para minimizar los costos (Khouja y Goyal [9]), las funciones de probabilidad más empleadas para ajustar la demanda son la de Poisson y la normal. Para el primer caso se pueden mencionar los trabajos de Atkins e Iyogun [1], Pantumsinchai [12], Viswanathan [14] y Fung, Ma y Lau [6].

El caso donde la demanda sigue una función de distribución tipo normal ha sido escasamente estudiado, únicamente se ha reportado una técnica en Eynan y Kropp [4] para obtener la frecuencia de pedido y el ciclo básico de tiempo T ; las soluciones obtenidas son de buena calidad, sin embargo, no se realizó un análisis más profundo del desempeño de dicho procedimiento. Finalmente en Eynan y Kropp [5] se reporta una nueva versión del algoritmo para el caso con nivel de servicio.

Como se ha visto, esta variante del problema ha recibido aún poca atención, al grado de existir solamente un procedimiento heurístico para obtener buenas soluciones; sin embargo, no ha sido comparado en su desempeño con alguna otra técnica.

En este trabajo se desarrolla e implementa un algoritmo de recocido simulado complementada con el algoritmo de sección dorada para resolver el problema de reaprovisionamiento multiproducto con demanda estocástica. Se generan y resuelven varias instancias para evaluar su desempeño y se comparan los resultados con los obtenidos con el procedimiento descrito en Eynan y Kropp [4].

El artículo está organizado de la siguiente manera: en la sección 2 se describe el problema de reaprovisionamiento multiproducto con demanda estocástica, en la sección 3 se muestra el procedimiento de Eynan y Kropp describiendo en forma breve su funcionamiento. En la sección 4 se da una breve introducción sobre el algoritmo de recocido simulado, en la sección 5 se desarrolla la implementación del algoritmo recocido simulado - sección dorada, en la sección 6 se muestran los resultados obtenidos seguido de las conclusiones y las referencias.

2 Modelo de costo para el problema de reaprovisionamiento multiproducto con demanda tipo normal

En los sistemas en donde se hace una revisión periódica del inventario cada T unidades de tiempo y la orden se activa para que el nivel de inventario alcance un punto preestablecido, el nivel requerido de inventario es el necesario para satisfacer la demanda durante el tiempo transcurrido entre dos órdenes consecutivas. Se ha observado que la función de distribución normal ajusta al comportamiento de la demanda de forma aceptable y esto tiene como ventaja la posibilidad de obtener una función de costo (Eynan y Kropp [4], Silver [13]). Los parámetros de la ecuación de costo son:

- n número total de productos.
- i índice del producto i , $i = 1, 2, \dots, n$.
- a_i costo de activación del producto i .
- h_i costo de acarreo del producto i .
- D_i demanda ajustada mediante una función de distribución Normal del producto i .
- σ_i desviación estándar de la demanda del producto i .
- z_i nivel de confianza de la distribución Normal del producto i .
- t_i tiempo de espera para reaprovisionamiento del producto i .
- k_i frecuencia de pedido del producto i , variable de decisión entera.
- A costo mayor por activación de pedido, independiente del número de productos.
- T ciclo básico de tiempo, variable continua de decisión.
- T_i ciclo individual del producto i , variable de decisión.
- CT costo total anual.

La frecuencia T_i de pedido individual del artículo i se restringe a un múltiplo entero del ciclo básico T :

$$T_i = k_i T. \quad (1)$$

El costo total, es la suma de los costos de activación y los costos de acarreo de inventario de seguridad, el problema de reaprovisionamiento multiproducto planteado como un problema de optimización es el siguiente:

$$\begin{aligned} \min CT(T, k_1, k_2, \dots, k_n) &= \\ &= \frac{1}{T} \left(A + \sum_{i=1}^n \frac{a_i}{k_i} \right) + \sum_{i=1}^n \left(\frac{T}{2} D_i h_i k_i + h_i z_i \sigma_i \sqrt{T k_i + t_i} \right) \end{aligned} \quad (2)$$

sujeto a : $T > 0$
 $k_i \geq 1$, enteros, $i = 1, 2, \dots, n$.

La función objetivo CT es no convexa, sin embargo, al fijar los valores de k_i , la función es convexa en la variable continua T . Para resolver instancias de este modelo sólo se ha desarrollado un método heurístico; una opción viable es emplear una técnica metaheurística complementada con un procedimiento numérico para realizar la búsqueda en la parte continua de la función objetivo. En este trabajo se propone la siguiente estrategia: emplear recocido simulado para obtener el valor de las k_i , a continuación mantenerlas fijas, para obtener el mínimo para T , aprovechando que la función CT es convexa. Es factible que mediante un método de búsqueda unidimensional, como el de sección dorada, se obtenga el mínimo.

3 Algoritmo heurístico de Eynan y Kropp

A continuación se describe la técnica de Eynan y Kropp tomada de [4]:

Paso 1. Calcular

$$T_i^* = \left(\frac{2a_i}{h_i \left(D_i + \frac{z_i \sigma_i}{\sqrt{T_{0i} + t_i}} \right)} \right)^{1/2} \quad (3)$$

donde $T_{0i} = \left(\frac{2a_i}{D_i h_i} \right)^{1/2}$, $i = 1, \dots, n$.

Paso 2. Identifique el producto con la T_i^* mínima, este producto se denota como producto 1 y se asigna $k_1 = 1$.

Paso 3. Calcular

$$T = \left(\frac{2(A + a_1)}{h_1 \left(D_1 + \frac{z_1 \sigma_1}{\sqrt{T_0 + t_1}} \right)} \right)^{1/2} \quad \text{donde} \quad T_0 = \left(\frac{2(A + a_1)}{D_1 h_1} \right)^{1/2}. \quad (4)$$

Paso 4. Hacer $k_i = q$ (entero) tal que:

$$\sqrt{(q-1)q} \leq \frac{T_i^*}{T} \leq \sqrt{q(q+1)}, \quad i = 2, \dots, n. \quad (5)$$

Paso 5. Calcular

$$T = \left(\frac{2 \left(A + \sum \frac{a_i}{k_i} \right)}{\sum k_i h_i \left(D_i + \frac{z_i \sigma_i}{\sqrt{k_i T_0 + t_i}} \right)} \right)^{1/2} \quad (6)$$

$$\text{donde } T_0 = \left(\frac{2(A + \sum \frac{a_i}{k_i})}{\sum k_i h_i D_i} \right)^{1/2}.$$

Paso 6. Repita los pasos 4 y 5 tantas veces como sea necesario hasta que el costo total, ecuación (2), no cambie de valor entre dos iteraciones sucesivas.

Esencialmente el procedimiento calcula los valores de T_i^* en la ecuación (3) de forma iterativa tomando como primer valor a T_{0i} , el calculado para el caso determinista, posteriormente identifica el producto con el T_i^* más pequeño y le asigna una frecuencia $k = 1$, y fija este valor como cota inferior para el ciclo base de tiempo T , el valor óptimo del ciclo base de tiempo T^* debe cumplir con $T_{\min} \leq T^*$. La determinación inicial del ciclo base de tiempo T estocástico se realiza empleando (4) y consiste en el valor de T para el caso determinista, los valores de la frecuencia de pedido k_i de cada producto se obtienen mediante (5). Finalmente obtiene el ciclo base de tiempo con la ecuación (6) substituyendo los valores de k_i .

4 La técnica de recocido simulado

Recocido es un proceso bien conocido en el estudio de materiales: un material con una estructura cristalina dada se lleva a temperatura elevada y posteriormente se deja enfriar lentamente con la finalidad de obtener una estructura cristalina con las características deseadas; existe un algoritmo para simular dicho proceso el cual, en el trabajo de Kirkpatrick et al. ([10]) es adaptado para resolver problemas combinatorios.

Recocido simulado (RS) está catalogado como un procedimiento de Búsqueda Aleatoria: a partir de una solución x_i con costo $f(x_i)$ se selecciona un vecino x_j dentro de una vecindad $N(x_i)$ previamente definida y empleando únicamente información local, después se evalúa la calidad de la solución $f(x_j)$. Si el valor es menor o igual mejor, se acepta la nueva solución x_j , en otro caso la solución no se descarta, sino que se evalúa la probabilidad de aceptación dada por la siguiente expresión, que se conoce como el criterio de Metropolis:

$$P[\text{aceptar } x_j] = \begin{cases} 1 & \text{si } f(x_j) < f(x_i) \\ \exp\left(-\frac{f(x_i) - f(x_j)}{c}\right) & \text{si } f(x_j) \geq f(x_i). \end{cases} \quad (7)$$

A c se le conoce como el parámetro de control. Por lo general, se escoge un valor inicial c_0 grande para que se tenga una alta probabilidad de escapar de mínimos locales. A medida que c disminuye, la probabilidad de aceptar soluciones de mala calidad es menor, cuando el valor de c se

aproxima a cero, la probabilidad de aceptar una solución mala se aproxima también a cero. Por regla general, el descenso del valor de c debe ser lento y así se previene que el algoritmo quede atrapado en un óptimo local.

Se deben explorar una cierta cantidad de puntos (o bien transiciones) para un valor dado de c , esta cantidad dependerá del tamaño de la instancia a resolver tomando en cuenta que: si el número de transiciones realizado no es suficiente, el algoritmo dejará zonas sin explorar y esto incrementa la posibilidad de quedar atrapado en un mínimo local, por otro lado, si se fija un número muy elevado de transiciones, se corre el riesgo de que el algoritmo ejecute más iteraciones de las que son necesarias, sin devolver ninguna mejora. Debe encontrarse un buen equilibrio entre la forma de decrementar el parámetro c y el número de transiciones exploradas manteniendo c fijo.

5 Implementación

Implementar el algoritmo de recocido simulado-sección dorada (RSSD) requirió tomar un conjunto de decisiones como la definición de la estructura de vecindades, el número de iteraciones para alcanzar el estado estacionario y el programa de enfriamiento (Johnson et al. [8]). A continuación se describen dichas consideraciones sobre el algoritmo.

5.1 Función de costo

La función de costo empleada, es directamente la función CT dada en el modelo (2).

5.2 Espacio de soluciones para valores de k_i

Para cada producto, la frecuencia mínima de pedido es $k_i = 1$, lo que equivale a solicitar el producto i cada ciclo de tiempo básico T y corresponde a la cota inferior para los valores de k_i , para obtener una cota superior se procedió como sigue: el valor T_{0i} de la ecuación (3) determina el valor del ciclo de tiempo óptimo individual sin tomar en cuenta la parte estocástica y en el problema de reaprovisionamiento multiproducto, el ciclo de tiempo óptimo individual T_i^* debe encontrarse en el intervalo, $T_{\min} \leq T_i^* \leq T_{0i}$ por lo que como cota inferior se selecciona el valor de T_{0i} más pequeño empleando la ecuación (8) para definir un intervalo búsqueda:

$$T_{\min} = \min_i \{T_{0i}\}. \quad (8)$$

En el problema de reaprovisionamiento el ciclo de tiempo individual se restringe a ser un múltiplo entero de T es decir, $T_i = Tk_i$, con la ecuación

(9) se evalúa la cota superior de la frecuencia k_i , en otras palabras, es el valor máximo posible de cada producto i :

$$k_{i,\max} = \left\lfloor \frac{T_{0i}}{T_{\min}} \right\rfloor. \quad (9)$$

Por lo que el intervalo para la frecuencia de pedido de cada producto i será $1 \leq k_i \leq k_{i,\max}$.

5.3 Solución vecina

Una solución vecina $K^* = (k_1^*, \dots, k_l^*, \dots, k_n^*)$ de una solución $K = (k_1, \dots, k_l, \dots, k_n)$ se obtiene al modificar en una unidad el valor de K de un solo producto. Como la elección del índice a perturbar es aleatoria y además la cota superior de k_i no es la misma para todos los productos, conviene establecer un criterio de selección que favorezca la búsqueda en aquellos índices con un espacio de soluciones más amplio. Mediante (10) se obtuvo la probabilidad de selección de cada producto. Así, los índices con un espacio de búsqueda mayor tendrán una probabilidad mayor de ser seleccionados.

$$P(\text{seleccionar } k_i) = \begin{cases} 0 & \text{si } k_{i,\max} = 1 \\ \frac{(k_{i,\max}-1)}{\sum_{\{i|k_{i,\max} \geq 2\}} (k_{i,\max}-1)} & \text{si } k_{i,\max} \geq 2 \end{cases} \quad (10)$$

Para perturbar se permiten dos operaciones: $k_i + 1$ ó $k_i - 1$; y cada una tiene una probabilidad de 0.5. Los índices cuyo $k_{i,\max}$ es 1 no se modifican.

5.4 Búsqueda en la variable continua

Ya que existe un valor de T que minimiza el costo cuando se fijan los valores de k_i , se realiza una búsqueda unidimensional usando sección dorada. Este algoritmo de búsqueda requiere que la función sea cuasi-convexa dentro de un intervalo $[a, b]$. Como se desconoce la localización exacta del mínimo dentro de $[a, b]$ a dicho intervalo se le llama intervalo de incertidumbre y es posible eliminar secciones del mismo para reducirlo (Bazaraa et al. [3]); en esta implementación la búsqueda sobre T se inicia dentro del intervalo $[a, b]$, donde:

$$a = T_{\min}; b = T_{\text{up}} \quad (11)$$

donde T_{up} se calcula empleando el valor T_0 de la ecuación (6), que es la ecuación para el caso determinista. El valor óptimo del ciclo base de tiempo se encuentra en el intervalo $T_{\min} \leq T^* \leq T_{\text{up}}$. Es importante señalar que el ciclo de tiempo estocástico es más pequeño que para el caso determinista, ya que se toma en cuenta la desviación estándar de la demanda.

5.5 Esquema de enfriamiento

Se aplicó el esquema geométrico empleando la ecuación (12), recomendado cuando no hay antecedentes de otras implementaciones; α es un valor a seleccionar entre 0.9 y 0.99, c_0 se selecciona de tal forma que se acepten alrededor del 80% de las soluciones

$$c_{q+1} = \alpha * c_q \text{ para } q = 0, 1, 2, \dots \quad (12)$$

5.6 Criterio de paro

El parámetro de control desciende hasta que se considera que el sistema está congelado. Para el algoritmo se fija un valor tal que $c_q < \varepsilon$ donde $\varepsilon = 0.01$.

5.7 Transiciones

Ya que se manejaron instancias con distintos tamaños, la búsqueda deberá ser más amplia en aquellas con mayor número de productos, con la ecuación (13) se obtuvo, para cada tamaño de instancia, el número de transiciones con c_q constante.

$$\text{transiciones} = N * n, \quad (13)$$

donde n es el número de productos y N es una constante que varía de acuerdo al tamaño de la instancia. A mayor número de transiciones, se favorece que el ciclo interno de RS alcance un estado estacionario para cada valor fijo de c_q .

5.8 Algoritmo de recocido simulado-sección dorada

En la Figura 1 se da el pseudocódigo del algoritmo de recocido simulado-sección dorada (algoritmo RSSD). Comienza con una Entrada en donde se reciben los parámetros c_0, c_f, α y transiciones (en la sección 6 se describen los valores que se tomaron para dichos parámetros). La solución inicial K se obtiene tomando $k_i = 1$, para toda $i = 1, \dots, n$. El valor de T se calcula mediante el proceso descrito en la subsección 5.4. Para el cálculo del costo total $CT(T, K)$ se usa la ecuación 2. El algoritmo tiene dos ciclos anidados; el ciclo exterior dado por la instrucción **Mientras**, modifica el parámetro de control c del valor inicial c_0 hasta un valor final menor que c_f . El ciclo interior, dado por la instrucción **Para**, se ejecuta desde 1 hasta transiciones; se selecciona una solución vecina K' del conjunto de soluciones vecinas $N(K)$ de K conforme a lo descrito en la subsección 5.3 y se obtienen T' y $CT(T', K')$. La condición **Si**, acepta o rechaza la nueva

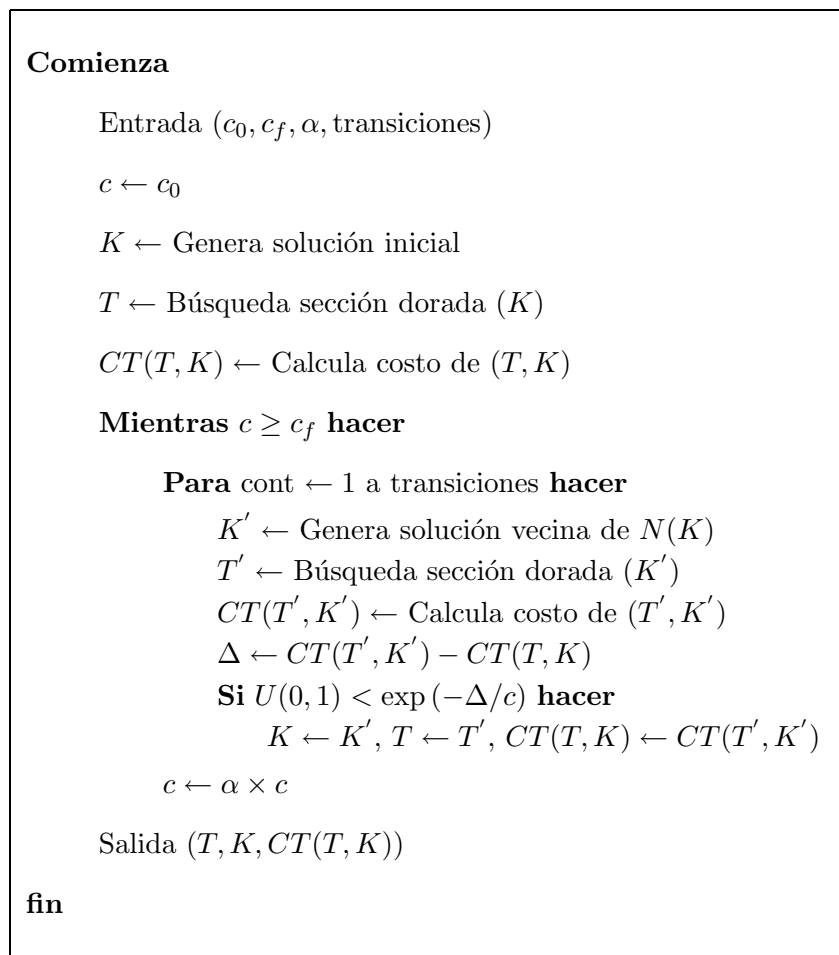


Figura 1: Algoritmo de recocido simulado-sección dorada.

solución K' de acuerdo al criterio de Metropolis (ecuación 7). Finalmente, el algoritmo devuelve la solución final K , y los valores T y $CT(T, K)$ en la última instrucción.

6 Experimentación

Seleccionar los parámetros adecuados de un algoritmo metaheurístico es una tarea que requiere tiempo, sobre todo si se trata de una primera implementación.

Debido a que el algoritmo de Eynan (Eynan y Kropp, (1998) [4]) es de tipo determinístico y no necesita de ningún tipo de ajuste de parámetros, sólo se necesitará encontrar los valores de los parámetros del

algoritmo RSSD para poder realizar la comparación de eficiencia entre estos algoritmos.

Los algoritmos se programaron en Visual Basic 2005. Debido a que no existe una librería de instancias para probar los algoritmos RSSD y de Eynan, fue necesario generarlas usando como referencia para los parámetros de demanda, de costo de activación individual, de costo de acarreo y de costo de activación mayor, los criterios del caso determinista (ver Khouja y Goyal (2008) [9]), pero incorporando los parámetros del tiempo de espera y la desviación estándar. Las instancias se generaron de forma aleatoria para $n = 10, 20, 30, 40$ y 50 productos; la demanda se generó en el intervalo $[100, 100000]$, los costos de acarreo se generaron en el intervalo $[0.5, 5]$ y los costos de activación en el intervalo $[2, 3]$. El tiempo de espera t se generó en el intervalo $[1/40, 1/6]$ unidades de tiempo. Se generaron instancias para valores de costo de activación mayor de $A = 5, 10, 15, 20$ y 30 . Se generaron 100 instancias por cada combinación de n y A , en total se generaron 2,500 problemas, las cuales se resolvieron con ambos algoritmos y se compararon las soluciones obtenidas clasificándolas como sigue: instancias donde $CT_{RS} < CT_{Eynan}$ (problemas tipo I) y las instancias donde $CT_{RS} > CT_{Eynan}$ (problemas tipo II).

Se realizaron corridas considerando valores de $c_0 = 50, 100$, de $\alpha = 0.9, 0.95$ y 0.99 y de $N = 1, 10, 50, 100$ sobre el conjunto de los 2,500 problemas, observándose en todos los casos que se obtenía entre un 95.3 % y un 97.52 % de problemas tipo I, también se observó que para valores de $\alpha = 0.99$ los tiempos de ejecución eran muy grandes (ver Tabla 1). Cabe mencionar que los tiempos de ejecución del algoritmo de Eynan fueron en todos los casos menores a 0.05 segundos, por lo que se tomó la decisión de sólo considerar valores de $\alpha = 0.90$ y 0.95 , puesto que para valores menores se tendría como consecuencia una posible convergencia prematura (a un óptimo local) y para valores mayores como ya se comentó aumenta el tiempo de ejecución.

Combinación	Transiciones	t (seg.)	n
$c_0 = 100$	100	1.45	10
$\alpha = 0.99$	200	4.81	20
$N = 10$	300	10.04	30
	400	17.07	40
	500	26.28	50

Tabla 1: Tiempos de ejecución para RSSD.

Para detectar los puntos de estancamiento (donde el % tipo I ya no aumenta), se resolvieron 500 problemas para cada número de productos,

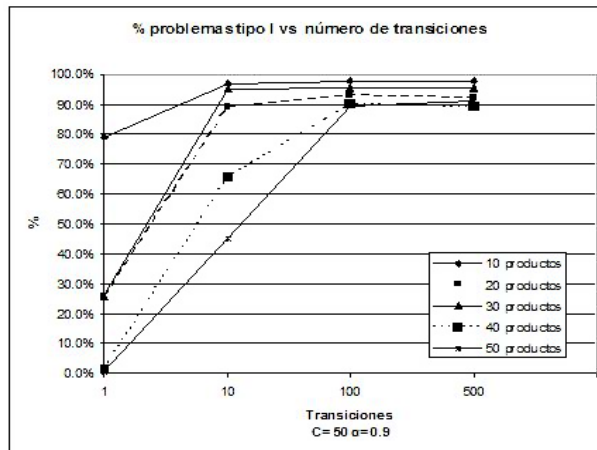


Figura 2: Porcentajes de problemas tipo I.

encontrándose que RSSD alcanza un porcentaje alto de soluciones tipo I (ver Figura 2) a partir de 100 transiciones para problemas de tamaño 10, de 400 transiciones para problemas de tamaño 20 y 30 y de 500 transiciones para problemas de 40 y 50 productos. Estos valores de transiciones afectaron de manera significativa los tiempos de ejecución, encontrándose hasta un ahorro del 79.63 % como se puede observar en la Tabla 2.

n	Transiciones	Tiempo	Transiciones	Tiempo	Mejora
10	500	0.685	100	0.0625	8.76 %
20	1000	0.21	400	0.1090	48.10 %
30	1500	0.44	400	0.1562	64.50 %
40	2000	0.75	500	0.2031	72.92 %
50	2500	1.15	500	0.2343	79.63 %

Tabla 2: Tiempos de ejecución en segs. para RSSD, con $c_0 = 50$ y $\alpha = 0.9$, $N = 50$.

7 Propuesta de un modelo de regresión lineal

Es muy recomendable realizar un análisis estadístico para la exploración de valores adecuados ([2]). En este trabajo nos apoyamos para tal efecto, en un arreglo experimental factorial 2^k con 2 repeticiones, para estimar los valores de las variables que impactan en la respuesta de un sistema ([11]). Después de la experimentación que se ha realizado, se decidió tomar

los valores del parámetro de control $c_0 = 50, 100$, para el parámetro de enfriamiento $\alpha = 0.90, 0.95$ y el factor del tamaño $N = 1, 10$.

Se realizaron dos experimentos para cada combinación de los valores de los parámetros considerados, adicionalmente se agregó un punto central para complementar el análisis, en total se realizaron 17 experimentos para cada una de las 2,500 instancias generadas, las cuales se resolvieron con ambos algoritmos y se compararon las soluciones obtenidas, los resultados se presentan en la Tabla 3.

Exp.	c_0	α	N	Tipo I	%	Tipo II	%
1	100	0.90	1	2395	95.8	105	4.2
2	50	0.90	1	2396	95.84	104	4.16
3	50	0.95	10	2434	97.36	66	2.64
4	50	0.95	1	2410	96.40	90	3.60
5	100	0.95	10	2438	97.52	62	2.48
6	50	0.90	10	2429	97.16	93	2.84
7	75	0.93	5.5	2430	97.2	70	2.80
8	50	0.90	10	2433	97.32	67	2.68
9	100	0.90	1	2398	95.92	102	4.08
10	100	0.95	10	2434	97.36	66	2.64
11	50	0.95	1	2408	96.32	92	3.68
12	100	0.90	10	2430	97.20	70	2.80
13	100	0.90	10	2433	97.32	67	2.68
14	100	0.95	1	2411	96.44	89	3.56
15	100	0.95	1	2413	96.52	87	3.48
16	50	0.95	10	2434	97.36	66	2.64
17	50	0.90	1	2389	95.56	111	4.46

Tabla 3: Resultados de la experimentación.

El algoritmo RSSD tuvo un excelente desempeño con respecto a la técnica de Eynan; puesto que el porcentaje de problemas tipo I varió entre 95.56% y 97.52 %, lo que muestra la bondad del algoritmo propuesto.

En la Tabla 4 se muestra el tiempo de ejecución de acuerdo al número de productos, se muestran solamente las combinaciones superior e inferior, en el caso del algoritmo de Eynan el tiempo de ejecución fue menor en todos los casos. Es importante señalar que los tiempos de ejecución elevados se deben principalmente a un número de transiciones grande, por lo que debe ponerse atención en la selección de los mismos (Tabla 4).

Combinación	Transiciones	t (seg.)	n
$c_0 = 100$	100	0.8442	10
$\alpha = 0.95$	200	3.22	20
$N = 10$	300	7.614	30
	400	13.483	40
	500	21.53	50
$c_0 = 50$	10	0.044	10
$\alpha = 0.90$	20	0.173	20
$N = 1$	30	0.383	30
	40	0.67	40
	50	1.06	50

Tabla 4: Tiempo de ejecución por número de productos para RSSD.

7.1 Verificación del modelo

Se verificó la validez del experimento realizado, en la Figura 3 se muestran graficados los residuales, donde se observa que los datos se comportan siguiendo una distribución normal, en la Figura 4 se presentan los residuales por corrida, no existe evidencia aparente de un patrón que indicara cambios en las condiciones a lo largo de la experimentación.

DESIGN-EXPERT Plot
% Problemas tipo I

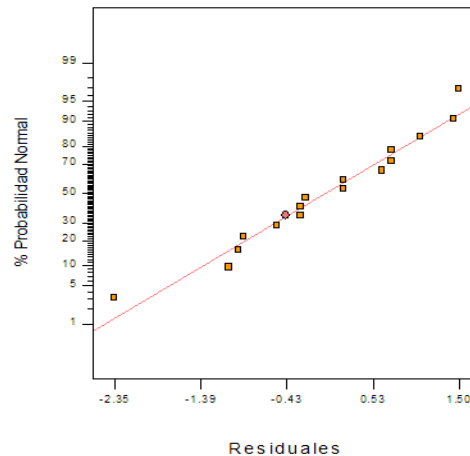


Figura 3: Gráfica de residuales.

Se ajustaron a los datos los diferentes modelos de regresión, los cuales se muestran en la Tabla 5.

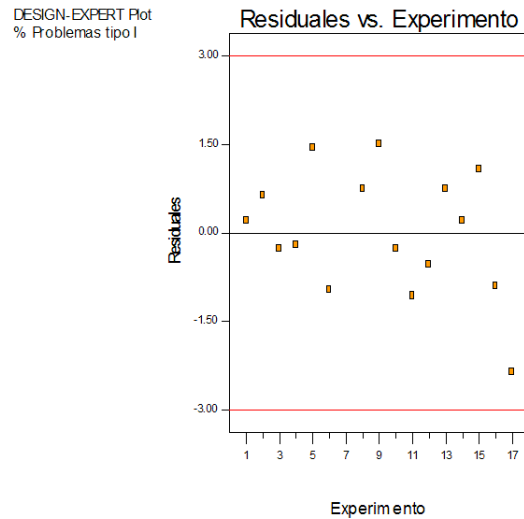


Figura 4: Residuales vs experimento.

Modelo Ajustado	R^2
% de problemas Tipo I = $95.99 + 0.14 N$	83.10
% de problemas Tipo I = $88.49 + 8.11 \alpha + 0.14 N$	92.23
% de problemas Tipo I = $88.35 + 0.002 c_0 + 8.11 \alpha + 0.14 N$	92.74
% de problemas Tipo I = $82.95 + 14.10 \alpha + 1.14 N - 1.09 \alpha N$	95.56

Tabla 5: Modelos ajustados y coeficiente de correlación R^2 .

En cuanto a las variables significativas para el experimento, se considerarán los parámetros α y N , puesto que fueron, según los resultados, los que tienen efecto sobre la respuesta a las condiciones del experimento, además existe una interacción de ambos parámetros y su efecto en la respuesta es similar al de N .

La respuesta, % de problemas tipo I, no requirió ningún tipo de transformación, el modelo tiene un coeficiente de regresión $R^2 = 0.9556$, lo cual indica un buen ajuste, existe curvatura en la región de experimentación, sin embargo la falta de correlación que pudiera deberse a dicha curvatura no es significativa.

La variable de mayor influencia sobre la respuesta es el parámetro de enfriamiento α seguido de la constante N , se consideró además en el modelo de predicción, la interacción entre dichos parámetros, sin embargo su efecto es el menor de los tres (Tabla 5).

7.2 Análisis del modelo y experimentación sobre los parámetros

En la Figura 5 se muestra la predicción de la respuesta, % de problemas tipo I, empleando el modelo obtenido. Se observa que el porcentaje de problemas tipo I, dentro de los límites asignados al experimento es elevado, de hecho con la combinación en los niveles inferiores es factible obtener más del 95% de problemas tipo I. Los valores de los parámetros de RSSD que se seleccionen dentro del modelo propuesto, se esperaría que un porcentaje mayor al 95.00% de los casos devolverán una mejor solución que el procedimiento de Eynan.

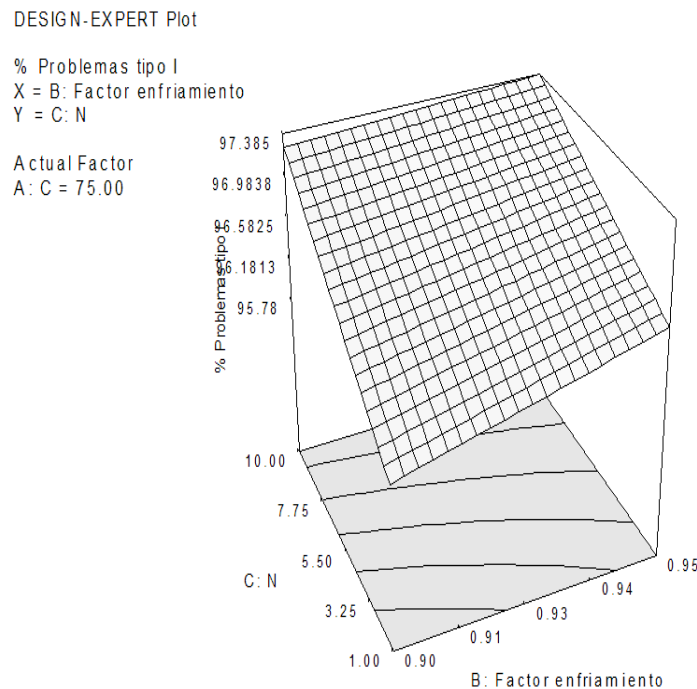


Figura 5: Superficie de respuesta.

Si se desea incrementar este porcentaje, el efecto más pronunciado se obtiene al modificar al mismo tiempo tanto α como la constante N , en la Figura 5 equivale a seguir una ruta que inicia en la intersección dada por la combinación $\alpha = 0.9$ y $N = 1$ hasta la esquina donde se encuentra la combinación $\alpha = 0.95$ y $N = 10$ obteniéndose un valor en el modelo de hasta 97.43% de problemas tipo I.

Es importante resaltar, que para valores mayores en los parámetros del modelo, tendrá un efecto en el que el incremento de problemas tipo I, además se tendrá que considerar que como consecuencia se tendrá un incremento en el tiempo de ejecución.

Se realizaron pruebas para verificar la bondad del modelo y al mismo tiempo estudiar el comportamiento del algoritmo RSSD. El objetivo fue incrementar el % de problemas tipo I, para esto se tomaron primeramente los valores de $N = 1$, $c_0 = 50$ que como ya se mencionó, se obtuvieron resultados similares que con los parámetros en un nivel superior, en contrapartida, con esta combinación el tiempo es menor.

N n	10	20	30	40	50
0.9	98.6	97.4	95.6	93.2	94.4
0.925	98.6	97.4	96.0	94.0	94.8
0.95	98.6	97.8	96.4	94.0	95.2

Tabla 6: Porcentaje de problemas tipo I para distintos valores de α . Fijos $N = 1$ y $c_0 = 50$.

N n	10	20	30	40	50	Promedio
1	98.6	97.4	95.6	93.2	94.4	95.84
5	98.8	97.8	97.4	94.8	96.2	97.00
10	98.8	98.0	97.0	95.4	96.6	97.16

Tabla 7: Porcentaje de problemas tipo I para distintos valores de N . Fijos $c_0 = 50$, $\alpha=0.90$.

Se consideró una rapidez de enfriamiento de $\alpha=0.925$. Al sustituir los valores en el modelo de regresión éste predice que el % de problemas tipo I será 96.13%. Al correr el experimento con estos valores de los parámetros se obtiene un 96.14% de problemas tipo I.

La distribución de los problemas de acuerdo al número de productos se muestra en la Tabla 6 y la Figura 6, se aprecia que dicho porcentaje varía de acuerdo al tamaño de la instancia para los valores de α de 0.9, 0.925 y 0.95. Disminuir el valor de α no es recomendable ya que el efecto de esta variable es muy grande y la posibilidad de quedar atrapado en un óptimo local es alta porque hay que recordar que este valor determina la rapidez con la que se disminuye el valor de c_0 .

A continuación se perturbó el valor de la constante N , realizándose una prueba para un valor $N=5$; el modelo de regresión indica que se obtendrán 96.67% de problemas tipo I, el resultado del experimento es de 97%. La

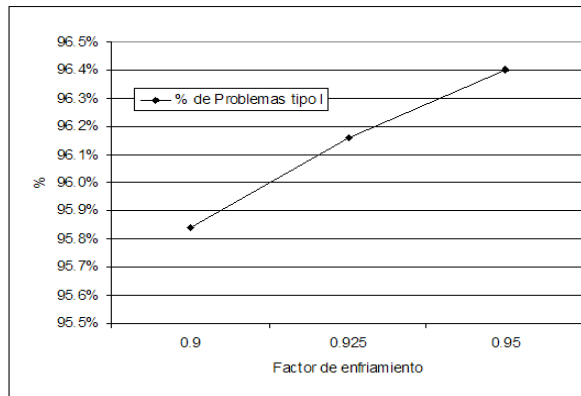


Figura 6: % problemas tipo I. Fijos $N = 1$ y $c_0 = 50$.

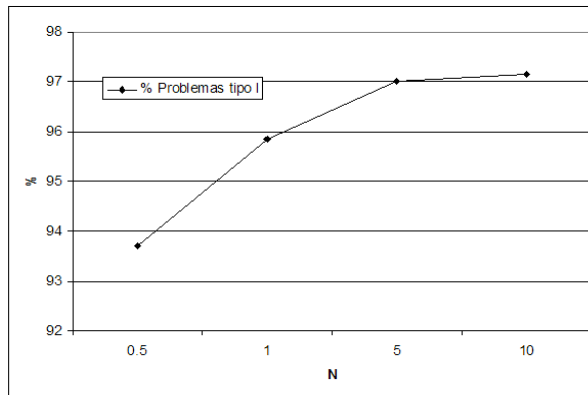


Figura 7: % problemas tipo I. Fijos $\alpha = 0.9$ y $c_0 = 50$.

forma en que se distribuyeron los porcentajes de acuerdo al tamaño de la instancia se muestran en la Tabla 7.

El valor de N aún se puede disminuir, cabe señalar que en este caso sería un valor fuera del intervalo estudiado por lo que no es recomendable realizar una predicción con el modelo de regresión propuesto. Se realizó una prueba empleando un valor $N=0.5$ obteniéndose un 93.72% de problemas tipo I. En este caso ya no se determinó la distribución de los porcentajes por número de productos, únicamente se muestra el resultado global junto con las otras pruebas en la Figura 7, los cambios son visibles ya que se requiere un número de transiciones al menos igual al tamaño de la instancia para obtener al menos 95% de problemas tipo I.

8 Conclusiones

El algoritmo de recocido simulado complementado con sección dorada se implementó para resolver el modelo del problema de reaprovisionamiento conjunto con demanda estocástica; el cual es un problema de optimización, no convexo.

Se realizó una extensa experimentación, en ella se comparó el algoritmo propuesto RSSD con otra técnica desarrollada por Eynan [4]. Los resultados obtenidos mostraron que en un porcentaje elevado, el algoritmo RSSD propuesto en este trabajo, obtuvo soluciones mejores comparadas con el algoritmo de Eynan [4]. Con los parámetros asignados al algoritmo RSSD y explorados en el diseño experimental, no menos del 95% del total de 2500 instancias son resueltas con mejores costos con el algoritmo propuesto en este trabajo.

El modelo de regresión obtenido indica que para el intervalo de valores propuesto, el mayor efectos sobre el % de problemas tipo I corresponde al parámetro de la rapidez de enfriamiento α , seguido de N y a continuación la interacción de ambas variables.

Posteriormente, se realizó una perturbación de los valores de los parámetros en el modelo de regresión y se realizaron corridas experimentales para conocer mejor su efecto sobre la respuesta y contrastarla respecto al valor de predicción del modelo propuesto. Se encontró que los cambios en cuanto al % de problemas tipo I variaron ligeramente para α , se observa que para esta variable, cualquier incremento significaría un incremento muy pequeño en el % de problemas tipo I a costa del tiempo de ejecución, además no es recomendable disminuir este valor ya que implica un descenso muy rápido en el valor de c_0 . Por otro lado, la perturbación de la constante N mostró que un incremento aumenta muy poco el % de problemas tipo I, sin embargo, al realizar una prueba empleando un valor de 0.5 se observó que el % de problemas tipo I cae a 93.6 %, al observar la gráfica se aprecia que se requiere al menos un número de transiciones igual o mayor al tamaño de la instancia para alcanzar sobre un 95% de problemas tipo I.

El modelo que se propone es un modelo lineal de regresión (MLR), lo que no significa que no exista algún modelo no lineal que mejor ajuste y que considere todos los parámetros. Ahora bien, bajo el MLR y considerando el criterio *ceteris paribus* se podría decir que, independientemente de los valores de los parámetros, el % de problemas tipo I será de al menos del 82.95%, lo cual muestra la bondad del algoritmo RSSD propuesto.

9 Agradecimientos

Los autores desean agradecer de manera muy especial a los árbitros anónimos, cuyas observaciones permitieron una mejora substancial del presente trabajo.

Referencias

- [1] Atkins, D.; Iyogun, P. (1988) “Periodic versus “can-order” policies for coordinated multi-item inventory systems”, *Management Science* **34**(6): 791-796.
- [2] Barr, R.; Golden, B. L.; Kelly, J.; Resende, G.C; Stewart, W. R. JR. (1995) “Designing and reporting on computational experiments with metaheuristics”, *Journal of Heuristics* **1**: 9–32.
- [3] Bazaraa, M.; Sherali, H.; Shetty, C.M. (2006) *Nonlinear Programming*. John Wiley and Sons, New York.
- [4] Eynan, A.; Kropp, D. (1998) “Periodic review and joint replenishment in stochastic demand environments”, *IEEE Transactions* **30**(11): 1025–1033.
- [5] Eynan, A.; Kropp, D. (2007) “Effective and simple EOQ-like solutions for stochastic demand periodic review systems”, *European Journal of Operational Research* **180**(31): 1135–1143.
- [6] Fung, R. Y. K.; Ma, X.; Lau, H. C. W. (2001) “(T, S) Policy for coordinated inventory replenishment systems under compound poisson demand”, *Production Planning and Control* **12**(6): 575–583.
- [7] Goyal, S. K. (1974) “Determination of optimum packaging frequency for items jointly replenished”, *Management Science* **21**: 436–443.
- [8] Johnson, D.; Aragon, C.; McGeoch, L.; Schevon, C. (1989) “Optimization by simulated annealing: an experimental evaluation; part I, graph partitioning”, *Operations Research* **37**(6): 865–892.
- [9] Khouja, M.; Goyal, S. (2008) “A review of the joint replenishment problem literature: 1989-2005”, *European Journal of Operational Research* **186**(1): 1–16.
- [10] Kirkpatrick, S.; Gellat, C. D.; Vecchi, M. P. (1983) “Optimization by simulated annealing”, *Science* **220**: 671–680.

-
- [11] Myers, R.H.; Montgomery, D C. (2002), *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. John Wiley and Sons, New York.
- [12] Pantumsinchai, P.A. (1992) “Comparison of three joint ordering inventory policies”, *Decision Sciences***23**: 111–127.
- [13] Silver, E.; Peterson, R. (1985) *Decision Systems for Inventory Management and Production Planning*. John Wiley and Sons, New York.
- [14] Viswanathan, S. (1997) “Periodic review (s,S) policies for joint replenishment inventory systems”, *Management Science* **43**(10): 1447–1454.

