

Algoritmo recocido simulado para el problema de la programación del tamaño del lote económico bajo el enfoque de ciclo básico

Simulated annealing algorithm to solve the economic lot scheduling problem and the basic cycle approach

José Omar Hernández¹ Salvador Hernández¹ Idalia Flores²

Recibido 30 de agosto de 2011, aceptado 19 de diciembre de 2011

Received: August 30, 2011 Accepted: December 19, 2011

RESUMEN

La programación del tamaño del lote económico consiste en determinar la secuencia y la cantidad a fabricar de N productos en un equipo o máquina, la cual tiene una capacidad limitada. Se trata de un problema NP-duro y las propuestas de solución son diversas. En esta investigación se trabaja con el enfoque del ciclo básico planteado por Bomberger, para el cual existen varias propuestas, dentro de las cuales se pueden encontrar hasta el momento sólo algoritmos genéticos en lo que se refiere a la implementación de técnicas metaheurísticas para resolver el problema. En este trabajo se resuelve el problema de muestra de Bomberger mediante la metaheurística recocido simulado; las aportaciones de esta investigación consisten en la forma de obtener un espacio de búsqueda más restringido de las variables, y una estrategia para controlar la exploración del espacio de soluciones que realiza el algoritmo, de tal manera que se realice una búsqueda eficiente. Dado que es una primera implementación de recocido simulado se experimenta con varias combinaciones de parámetros. El algoritmo obtiene los mismos costos en casi todas las pruebas realizadas; sin embargo, en las pruebas donde la relación $\sum \frac{D_i}{P_i}$ es alta, el algoritmo se desempeña mejor, mejorando la solución reportada anteriormente en algunos de los experimentos.

Palabras clave: Producción, lote, inventario, heurísticas, recocido simulado.

ABSTRACT

The problem considered is that of scheduling the production of several different items over the same machine with restricted capacity and on a repetitive basis. The problem is NP-hard and there exist several methods for the problem. In this research we worked with Bomberger's basic cycle approach, for which there are reported several solution proposals, among which are found so far only genetic algorithms in regard to the implementation of metaheuristic techniques to solve the problem. We solved Bomberger's classical example problem, applying simulated annealing metaheuristic. The contributions of this research consist on how to obtain bounds for search space of the variables, and a strategy to control the search along the solution space performed by the algorithm. Because this is a first implementation, we tested several values of the parameters. The algorithm achieves the same cost in almost all tests and the algorithm performs better where the ratio $\sum \frac{D_i}{P_i}$ is high, improving some of the results reported in the literature.

Keywords: Production, lot scheduling, inventory, heuristics, simulated annealing.

¹ Instituto Tecnológico de Celaya, Departamento de Ingeniería Industrial. Antonio García Cubas s/n. CP 38010. Celaya, México. CP 38010. E-mail: blink_leoncampeon@hotmail.com; salvador.hernandez@itcelaya.edu.mx

² División de Estudios de Posgrado, Facultad de Ingeniería, Universidad Nacional Autónoma de México. Circuito Universitario s/n. C.P. 04510. Distrito Federal, México. E-mail: idalia@servidor.unam.mx

INTRODUCCIÓN

La programación del tamaño del lote económico consiste en determinar la secuencia y la cantidad a fabricar de N productos en un equipo o máquina, la cual tiene una capacidad limitada. La solución propuesta consiste en un programa de producción en el que se incurre en costos por producción y de almacenamiento, además debe ser factible, es decir, debe ser una secuencia que se repite de manera indefinida sin interferencias y no sobrepase la capacidad del equipo. Este problema ha recibido mucha atención a partir de las propuestas de Rogers [1] y Bomberger [2]. Este último presenta inclusive un ejemplo numérico que a la fecha ha servido como referencia para probar los procedimientos desarrollados.

Se trata de un problema catalogado como NP-duro, (no se cuenta con un algoritmo que lo resuelva en tiempo acotado por un polinomio), para el que se han propuesto una gran variedad de procedimientos para obtener una solución. De acuerdo a Khouja, Michalewicz y Wilmot [3] existen dos enfoques de estudio para la programación del lote económico:

1. Encontrar una solución al problema restringido, en donde es necesario incorporar restricciones adicionales al modelo. Como ejemplos son el enfoque de ciclo común propuesto por Hanssmann [4], el enfoque de Bomberger [2], citado anteriormente y conocido como ciclo básico o período básico, y el ciclo básico extendido propuesto por Elmaghraby [5].

2. Encontrar una solución al problema original y posteriormente verificar la factibilidad de la solución propuesta, en este caso el tamaño de la corrida de producción puede variar entre cada corrida programación y por esto se le conoce también como Tamaño de Lote variable (Variable-lot-size). Aquí se encuadran, por ejemplo, los trabajos originales de Rogers [1] citado más arriba, Maxwell [6], Delporte y Thomas [7], Dobson [8] por mencionar solamente algunos. Esta línea cuenta con una amplia literatura, incluyendo metaheurísticas, las cuales se han reportado en [9], [10] y [11].

Para dar solución al problema bajo el enfoque de ciclo básico, existe una diversidad de propuestas. En los trabajos de Bomberger [2], Elmaghraby [5] y Axsäter [12] utiliza programación dinámica,

Grznar y Riggle [13] se proponen una técnica de optimización global, Yao y Elmaghraby [14] y Yao [15] proponen métodos de solución basados en potencias de 2. También se han desarrollado procedimientos heurísticos reportados por Madigan [16], Goyal [17] o Doll y Whybark [18]. En cuanto a las técnicas metaheurísticas implementadas, el trabajo más conocido corresponde a Khouja, Michalewicz y Wilmot [3] donde se resuelve el modelo del ciclo básico mediante Algoritmos Genéticos (AG). Una conclusión importante del trabajo es que resuelve problemas donde la relación de la demanda D con respecto a la rapidez de producción $P \sum \frac{D_i}{P_i}$ es alta, en otras palabras, con una elevada utilización de la capacidad del equipo. Hasta el momento, no hay otro estudio que aplique alguna técnica metaheurística sobre el enfoque del ciclo básico original que propuso Bomberger.

Finalmente para el caso del ciclo básico extendido de Elmaghraby se pueden mencionar los trabajos de Park y Yun [19] y Phillipoom, Rees y Taylor [20] basados en enumeración de las soluciones. En lo que respecta a metaheurísticas implementadas en esta línea de investigación se pueden mencionar los AG's de Chatfield [21] y Hainan [22]. Es importante mencionar las revisiones sobre el tema y los enfoques de solución: el trabajo clásico de Elmaghraby [5] hace una crítica de los procedimientos y enfoques de solución propuestos hasta 1978; en Rafiei [11], Chatfield [21], así como la disertación doctoral de Hainan [22], todos ellos enlistan los trabajos reportados a partir de 1978 y abarcan los últimos 30 años.

La presente investigación se encuadra en la primera línea de trabajo: el enfoque del ciclo básico propuesto por Bomberger [2]. La contribución del trabajo consiste en:

1. Resolver el modelo de ciclo básico de Bomberger utilizando recocido simulado (RS), pero implementando nuevas estrategias para calcular un espacio de soluciones más restringido de las variables, a diferencia por ejemplo de [3].

2. Proponer criterios para dirigir la búsqueda sobre aquellas variables en las que existe un espacio de soluciones más amplio.

Dado que se trata de una primera implementación de recocido simulado para el enfoque de ciclo básico,

se exploran distintas combinaciones de parámetros y las soluciones se comparan con las que se reportan en Khouja, Michalewicz y Wilmot [3], finalmente se realiza un análisis sobre el comportamiento del algoritmo bajo distintos escenarios $\sum \frac{D_i}{P_i}$, en otras palabras, sobre la utilización del equipo.

El artículo consta de las siguientes secciones: primero se muestra el modelo de optimización, describiendo al final el enfoque de Bomberger, a continuación se presenta una breve introducción a la técnica RS, posteriormente se describe la implementación del algoritmo, a continuación se presentan los experimentos y los resultados con su discusión, seguido de las conclusiones y las referencias.

MODELO DE OPTIMIZACIÓN

La notación utilizada en este trabajo se presenta en la Tabla 1.

Tabla 1. Notación.

<p>CT: Costo total (\$/día). i: Índice de producto. CT_i: Costo total individual (\$/día). N: Número de productos. a_i: Costo de preparación del producto i (\$). D_i: Demanda del producto i (unidades /día). h_i: Costos por almacenar el producto i (\$/unidad.día). T: Período Básico, variable continua (días). T_i: Período individual (días). t_i: Tiempo de preparación individual (días). k_i: Frecuencia de producción del producto i, variable entera. P_i: Rapidez de producción (unidades/día). T_{\min}: Cota inferior del período base (días). T_{\max}: Cota superior del período base (días). U: Función de probabilidad uniforme. V: Vecindad de la solución x. ρ: Iteración. μ: Multiplicador de Lagrange. λ: Constante de penalización. ε: Criterio de paro de recocido simulado. l, ζ, γ: Números aleatorios.</p>
--

El problema consiste en construir un programa de producción, en donde se especifique cada cuánto tiempo debe programarse la corrida del producto i con demanda constante [2].

Los supuestos para construir el modelo son:

- (1) Sólo puede fabricarse un producto a la vez.
- (2) La fabricación del producto i implica un tiempo y un costo de preparación del equipo.

- (3) El costo de preparación y el tiempo de preparación son independientes de la secuencia de producción.
- (4) La demanda y la rapidez de producción del producto i son conocidas y constantes.
- (5) Se debe cubrir toda la demanda de cada producto.
- (6) Existe un costo por almacenar el producto i .
- (7) El costo total de producción del producto i para un determinado período es igual a la suma del costo de preparación más el costo de almacenamiento.
- (8) El tiempo requerido para la fabricación del producto i es la suma del tiempo de preparación más el tiempo para la fabricación.

La solución consiste en un vector T_1, T_2, \dots, T_N , donde T_i es el período que transcurre entre dos corridas de fabricación consecutivas del producto i : incluye el tiempo requerido para la preparación, tiempo para la fabricación y el tiempo para permitir la producción de otros productos y satisfaciendo la demanda.

El costo individual de fabricación es la suma del costo de preparación y el costo de almacenamiento dados por la siguiente ecuación:

$$CT_i = \frac{a_i}{T_i} + \frac{D_i h_i}{2} \left(1 - \frac{D_i}{P_i} \right) T_i \quad (1)$$

El costo total es la suma de los costos individuales:

$$CT = \sum_{i=1}^N CT_i = \sum_{i=1}^N \left(\frac{a_i}{T_i} + \frac{D_i h_i}{2} \left(1 - \frac{D_i}{P_i} \right) T_i \right) \quad (2)$$

Derivando (2) con respecto a T_i y despejando la variable de decisión se obtiene (3), con la que se calcula el valor óptimo del ciclo de tiempo individual:

$$T_i = \sqrt{\frac{2a_i}{D_i h_i \left(1 - \frac{D_i}{P_i} \right)}} \quad (3)$$

La solución de esta ecuación es conocida como solución independiente y proporciona una cota inferior del costo, aunque es óptima para el problema sin restricciones [9]. Cabe señalar que por el momento no se ha obtenido una solución para el problema

general [3,16], por lo que se ha optado por agregar restricciones o supuestos al período individual T_i , como son el ciclo común (CC) propuesto por Hanssmann en 1962 [4] o bien restringir el período de tiempo individual a un múltiplo entero de un ciclo básico T propuesto por Bomberger en 1966 [2].

El CC consiste en suponer que los productos comparten el mismo período de tiempo y se fabrican en el mismo orden en cada programa de producción. Además la solución debe ser factible, es decir, el ciclo de tiempo debe ser suficiente para acomodar todos los productos [2]. Bajo estas condiciones el modelo de optimización se puede plantear como:

Min

$$CT = \sum_{i=1}^N CT_i = \sum_{i=1}^N \frac{a_i}{T} + \frac{D_i h_i}{2} \left(1 - \frac{D_i}{P_i}\right) T \quad (4)$$

Sujeto a:

$$\sum_{i=1}^N \left(a_i + T \frac{D_i}{P_i} \right) \leq T \quad (5)$$

La función objetivo dada por (4) debe minimizarse, la restricción (5) asegura que el ciclo es suficiente para acomodar todos los productos y la variable de decisión es el ciclo común T .

Derivando la función de costo, sin tomar en cuenta la restricción se obtiene la siguiente expresión para calcular el valor de T :

$$T = \sqrt{\frac{2 \sum_{i=1}^N a_i}{\sum_{i=1}^N D_i h_i \left(1 - \frac{D_i}{P_i}\right)}} \quad (6)$$

El valor T^* bajo el enfoque de CC, será:

$$T^* = \max [T, T_{\min}] \quad (7)$$

donde $T_{\min} = \frac{\sum_{i=1}^N a_i}{1 - \sum_{i=1}^N \frac{D_i}{P_i}}$

Si los ciclos individuales son muy cercanos entre sí, la solución del enfoque de CC proporcionará una solución muy cercana al óptimo, sin embargo esto no siempre sucede en la práctica [12].

El enfoque de Bomberger

Bomberger propuso una condición para satisfacer tanto la restricción de la capacidad como la de repetitividad y que en principio parece bastante simple y que se conoce como ciclo básico o periodo básico.

En su modelo, Bomberger establece que el ciclo individual T_i debe ser un múltiplo entero del ciclo básico, planteándolo como $T_i = k_i T$, donde k_i es la frecuencia de producción y T es el ciclo básico [2]. Tomando en cuenta la consideración anterior, se puede plantear la restricción para la capacidad y para la factibilidad de la solución propuesta de la siguiente manera:

$$\sum_{i=1}^N \left(t_i + \left(\frac{D_i}{P_i} \right) k_i T \right) \leq T \quad (8)$$

La restricción expresa que el ciclo básico debe ser suficiente para acomodar la fabricación de los productos sin violar la capacidad del equipo y sin que se presenten interferencias. Si la restricción del ciclo individual se substituye también en (2) se obtiene la siguiente ecuación de costo:

$$CT = \sum_{i=1}^N CT_i = \sum_{i=1}^N \frac{a_i}{Tk_i} + \frac{D_i h_i}{2} \left(1 - \frac{D_i}{P_i}\right) Tk_i \quad (9)$$

La función de costo tiene ahora una variable continua y N variables discretas, obteniéndose una función no-convexa. El modelo de optimización completo es el siguiente:

Min

$$CT = \sum_{i=1}^N \frac{a_i}{Tk_i} + \frac{D_i h_i}{2} \left(1 - \frac{D_i}{P_i}\right) Tk_i \quad (10)$$

Sujeto a:

$$\sum_{i=1}^N \left(\frac{t_i}{T} + \left(\frac{D_i}{P_i} \right) k_i \right) \leq 1 \quad (11)$$

$$T > 0 \quad (12)$$

$$k_i \geq 1 \text{ y entero } \forall i \quad (13)$$

La solución del modelo consiste en determinar el período básico óptimo T^* (variable continua) y la frecuencia de fabricación óptima k_i (variable entera) de los N productos, (10) es la función de costo y debe minimizarse, (11) es la restricción de capacidad y la factibilidad de la solución propuesta, (12) es la restricción para la variable continua y (13) corresponde a la restricción para las frecuencias. Se trata de un problema no-convexo, donde para cada combinación de frecuencias existe un valor T^* que minimiza el costo de la función, por lo que tiene varios mínimos locales y un mínimo global [14, 15].

LA TÉCNICA DE RECOCIDO SIMULADO

Las técnicas metaheurísticas son un conjunto de algoritmos generales para optimización combinatoria; una de sus características es que no están diseñados para un problema concreto, todo lo contrario, son procedimientos muy flexibles y esto permite que sea posible aplicarlos a la gran mayoría de los problemas combinatorios. Las técnicas más populares son los Algoritmos Genéticos, la Búsqueda Tabú y el Recocido Simulado.

El Recocido Simulado (RS) es un procedimiento introducido por Kirkpatrick, Gelatt y Vecchi [23] y ha sido empleado con frecuencia para resolver problemas combinatorios (Figura 1). La idea surge del proceso físico conocido como recocido, en el cual se eleva la temperatura de un sólido hasta el punto que se vuelve líquido, a continuación la temperatura se disminuye de forma paulatina para obtener una estructura cristalina sin defectos (estado basal).

El RS inicia con una solución x , se selecciona a continuación una solución vecina x' dentro de cierta vecindad o región V , posteriormente se evalúa $f(x)$ y $f(x')$.

El algoritmo:

- 1 Repetir
- 2 Repetir (para cierto c)
- 3 Seleccionar aleatoriamente x' en vecindad $V(x)$.
- 4 Sea $\Delta = f(x') - f(x)$
- 5 Si $\Delta < 0$ entonces $x = x'$
- 6 En otro caso generar aleatoriamente $l \in U(0,1)$
- 7 Si $l < \exp(\Delta/c)$ entonces $x = x'$
- 8 Hasta llegar a equilibrio.
- 9 Actualizar temperatura c .
- 10 Hasta condición de parada.

Figura 1. Algoritmo recocido simulado.

Si la solución vecina mejora el valor de la función objetivo se acepta con probabilidad 1, en caso contrario se calcula la probabilidad para aceptarla mediante el criterio de Metrópolis dado por (14).

$$\Pr[\text{aceptar } x'] = \begin{cases} 1 & f(x') < f(x) \\ \exp\left(-\frac{f(x')-f(x)}{c}\right) & f(x') \geq f(x) \end{cases} \quad (14)$$

Al valor c se le conoce como el parámetro de control; el valor inicial c_0 debe ser tal que al inicio las soluciones de mala calidad tengan una probabilidad alta de ser seleccionadas, a medida que c disminuye la probabilidad de aceptar soluciones de pobre calidad disminuye también y el algoritmo se detiene por ejemplo cuando c rebasa un umbral ε , finalmente el número de soluciones vecinas evaluadas a un valor de c debe asegurar una buena exploración del entorno [23, 24].

IMPLEMENTACIÓN DEL ALGORITMO

A continuación se presenta una descripción breve de la implementación del algoritmo RS. La idea de construir un espacio de soluciones tal que la búsqueda heurística se realice de manera eficiente es básica en la implementación de técnicas metaheurísticas; en este sentido la aportación sustancial de este trabajo es, por un lado, la propuesta de intervalos de búsqueda más restringidos para el problema en comparación con los utilizados por ejemplo en [3], y por otro los criterios para dirigir la búsqueda sobre regiones donde existe una mayor cantidad de combinaciones en el caso concreto del algoritmo RS.

Espacio de búsqueda para las variables

Variable continua: Existen varias propuestas para obtener una mejor cota de la variable continua. Desde el punto de vista de las técnicas metaheurísticas, se debe seleccionar un valor del ciclo básico dentro de cierta vecindad o entorno $T' \in V(T)$, el entorno está definido por el intervalo $V=(T_{\min}, T_{\max})$.

Un intervalo natural que se puede utilizar de forma inicial es el que está definido por la solución individual y se calcula con las siguientes fórmulas:

$$T_{\min} = \min_i [T_i] \tag{15}$$

$$T_{\max} = \max [T_i] \tag{16}$$

Se han propuesto nuevas cotas para el intervalo anterior, por ejemplo: en [3] donde se aplican AG, se construye una cota inferior que está por debajo del valor T_i más pequeño, el argumento está basado en el hecho de que a medida que la relación $\sum \frac{D_i}{P_i}$ disminuye, la solución óptima del problema se mueve hacia la solución independiente. Se realizaron algunas pruebas en este sentido y si bien esto es válido, el valor óptimo del ciclo básico no rebasó el valor más pequeño de T_i del problema de Bomberger.

En lo que respecta a la cota superior, diversos autores obtienen el valor T_{\max} substituyendo $k_i = 1$ para $i = 1, 2, \dots, n$ productos en (17) [3, 12, 15, 17].

$$T = \left[\frac{2 \left(\sum_{i=1}^N \frac{a_i}{k_i} \right)}{\sum_{i=1}^N D_i h_i k_i \left(1 - \frac{D_i}{P_i} \right)} \right]^{\frac{1}{2}} \tag{17}$$

En este trabajo se aplica (17) para obtener el valor T_{\max} . La propuesta que se muestra a continuación sirve para mejorar el intervalo del ciclo básico, para esto se aplicó el procedimiento de Viswanathan [25] utilizado en el problema de reaprovisionamiento de productos múltiples, pero modificándolo para el problema del tamaño económico de lote. Se usó dicho procedimiento para reducir el espacio de búsqueda a un intervalo más pequeño.

Algunas pruebas realizadas mostraron que este procedimiento aplicado para calcular la cota inferior en el problema estudiado aquí dejaba fuera el valor óptimo T^* , por lo que en este trabajo se implementa únicamente la parte del algoritmo para calcular la cota superior y que se muestra a continuación:

Cota superior

1. $\rho \leftarrow 0$
2. Calcule el ciclo básico T_ρ con (17). En la primera iteración, calcule T_0 utilizando $k_i = 1$ para $i = 1, 2, \dots, N$.
3. Calcule los valores de frecuencia óptimos mediante la regla de redondeo propuesta por Goyal [17]:

$$\sqrt{k_i(T_\rho)(k_i(T_\rho)-1)} < \frac{T_i}{T_\rho} < \sqrt{k_i(T_\rho)(k_i(T_\rho)+1)} \tag{18}$$

4. Deténgase cuando no exista un cambio en los valores de frecuencia en dos iteraciones consecutivas y vaya al paso 5, en caso contrario $\rho = \rho + 1$ y vuelva al paso 2.

5. Sea $T_{act} \leftarrow T_\rho$, calcule $T_{\max} = T_{\min} + T_{act}$.

Por lo general, el procedimiento para mejorar la cota superior no requiere más de cuatro iteraciones.

Variable entera: A continuación se detalla la forma en que se obtiene el intervalo de búsqueda de la variable entera, y que se encuentra en la región $V(k)$.

En esta implementación se experimentó con una cota superior más restringida para los valores de frecuencia a diferencia de la utilizada en [3]. Es conocido el hecho de que la cota inferior de la frecuencia es $k_i = 1$, es decir, dicho producto se fabricará en cada corrida de producción, por lo que el valor de la frecuencia se obtiene del intervalo de búsqueda definido como $1 \leq k_i^* \leq k_{i,\max}$. Para calcular $k_{i,\max}$ se procede como sigue:

1. Calcule el valor de ciclo individual con (3).
2. Ordene los productos de manera ascendente. Determine el mínimo con (15).
3. La frecuencia de pedido máxima se calcula con (19):

$$k_{i,\max} = \left\lfloor \frac{T_i}{T_{\min}} \right\rfloor \tag{19}$$

Como se observa a partir de (19), existe al menos un producto que se solicitará con frecuencia 1. Se debe considerar que:

1. La selección del producto i se hace de manera aleatoria, y
2. Existen productos con un espacio de soluciones limitado, por ejemplo, al intervalo de valores de frecuencia $1 \leq k_i \leq 3$ y al mismo tiempo hay productos con intervalo más amplios, por ejemplo $1 \leq k_i \leq 10$.

Al construir el entorno de búsqueda para la variable entera es necesario asegurarse de que la exploración se realice de forma inteligente. En esta implementación, se dirige la selección hacia los productos con un intervalo más amplio con el objeto de que el algoritmo explore regiones con mayor número de combinaciones. En este sentido se desarrolló el siguiente procedimiento para determinar las probabilidades de selección de cada artículo:

1. Suma las frecuencias de aquellos productos que cumplen con $k_{i,\max} > 1$: $\sum_{k_{i,\max} > 1}^N k_{i,\max}$
2. Obtenga la probabilidad de selección mediante:

$$\Pr(x' = i) = \frac{k_{i,\max}}{\sum_{k_{i,\max} > 1}^N k_{i,\max}} \quad (20)$$

Como se puede apreciar, la probabilidad de selección del elemento i es mayor para aquellos en los que el espacio de búsqueda es más amplio, o dicho de otra forma, donde la diferencia entre $k_{i,\min}$ y $k_{i,\max}$ es mayor.

Obtención de la solución vecina

El problema consiste en minimizar una función que consta de una parte continua correspondiente al ciclo básico T y la parte entera representada por las frecuencias (k_1, k_2, \dots, k_N) ; la solución completa está definida por el vector $(T, k_1, k_2, \dots, k_N)$. Dado que son dos tipos de variables se selecciona una a la vez. Para este fin se genera un número aleatorio γ en el intervalo $(0, 1)$ utilizando una distribución uniforme U : si $\gamma \leq 0,5$ se selecciona la variable continua, en otro caso se selecciona la variable entera.

Para acceder a la solución vecina se procede de acuerdo al tipo de variable seleccionada, y esto se describe a continuación.

Variable continua: Una vez definida la vecindad o entorno $V = (T_{\min}, T_{\max})$, y que ya es un intervalo mejorado con el procedimiento descrito más arriba, se selecciona un punto dentro del intervalo de manera aleatoria utilizando una distribución uniforme U .

Variable entera: Una vez que se ha seleccionado el producto i , se perturba su frecuencia para acceder

a la solución vecina siendo las operaciones válidas $[k_i + 1, k_i - 1]$, de las cuales se escoge una aleatoriamente aplicando (21).

$$\Pr(\text{selecc. op.}) = \begin{cases} k_i + 1 & \text{Si } \zeta > 0,5 \\ k_i - 1 & \text{En otro caso} \end{cases} \quad (21)$$

Donde ζ es un número aleatorio generado en el intervalo $(0,1)$ utilizando una distribución uniforme U . Cabe aclarar que en caso de que el valor actual de la frecuencia sea $k_{i,\max}$, la operación $k_i - 1$ se selecciona con probabilidad 1, del mismo modo si la frecuencia actual es 1, entonces la operación $k_i + 1$ se selecciona con probabilidad 1.

Evaluación de la solución vecina propuesta:

Una vez que se ha obtenido la solución vecina (escogiendo el valor de T' o bien perturbando la variable entera), se evalúa su calidad. Para este trabajo se empleó la siguiente función de Costo Total Penalizado (CTP) que evalúa la calidad del plan de producción propuesto:

$$CTP = \sum_{i=1}^N \frac{a_i}{Tk_i} + \frac{D_i h_i}{2} \left(1 - \frac{D_i}{P_i} \right) Tk_i + \lambda_c [\Phi] \quad (22)$$

Donde:

$$\Phi = \max \left[0, \sum_{i=1}^N \left(\frac{t_i}{T} + \frac{D_i}{P_i} k_i \right) - 1 \right] \quad (23)$$

La constante λ_c es el costo adicional en el que se incurre si la solución no cumple con la restricción a un valor c dado, la función Φ es la penalización que se activa para aquellas soluciones que no satisfacen la restricción. En esta implementación la estrategia consistió en una penalización de tipo dinámico, en cada descenso de temperatura se incrementa el valor de la constante λ_c .

$$\lambda_c = \lambda + 100 \quad (24)$$

Es necesario determinar si la solución devuelta por RS es factible. Construyendo la función lagrangeana del problema de optimización se tiene:

$$L = \sum_{i=1}^N \frac{a_i}{Tk_i} + \frac{D_i h_i}{2} \left(1 - \frac{D_i}{P_i} \right) Tk_i + \mu \left[\sum_{i=1}^N \left(\frac{t_i}{T} + \frac{D_i}{P_i} k_i \right) - 1 \right] \tag{25}$$

Fijando los valores de la frecuencia y derivando con respecto al ciclo básico se obtiene la siguiente expresión para calcular el ciclo básico óptimo T^* en función de las frecuencias y donde se toma en cuenta la restricción:

$$T^* = \left[\frac{2 \left(\sum_{i=1}^N \frac{a_i}{k_i} + \mu \sum_{i=1}^N t_i \right)}{\sum_{i=1}^N D_i h_i k_i \left(1 - \frac{D_i}{P_i} \right)} \right]^{\frac{1}{2}} \tag{26}$$

Cuando la solución devuelta por RS es factible (es decir, existe suficiente capacidad para acomodar todos los productos) existirá holgura en la restricción y por las condiciones de optimalidad el valor del multiplicador de Lagrange es cero [26]. En caso contrario, se debe calcular el valor del ciclo básico T^* y el valor de μ que satisfacen la restricción con el procedimiento siguiente:

1. Si $\left(\sum_{i=1}^N \frac{t_i}{T} + \frac{D_i}{P_i} k_i \right) - 1 \geq 0$, perturbe el valor de μ , y actualice el valor de T^* con (26).

2. Repita hasta obtener el valor del multiplicador de Lagrange y del ciclo básico T^* que satisfacen la restricción, calcule el costo.

EXPERIMENTOS Y RESULTADOS

El algoritmo (Figura 2) se codificó en Fortran 90 y se realizaron las pruebas en una computadora con procesador Intel i5 y 4Gb de memoria RAM. Para estudiar el desempeño del algoritmo se emplearon los datos de la Tabla 2, estos corresponden a un problema que ha sido utilizado desde su aparición como referencia para probar los algoritmos desarrollados realizando las evaluaciones bajo tres escenarios de demanda que se calculan mediante la fórmula:

$$D_{i,r} = D_i \times r \tag{27}$$

El algoritmo RS-problema del lote económico:

- 1 Repetir
- 2 Seleccionar variable de decisión (continua, entera).
- 3 Repetir (para cierto c).
- 4 Seleccione aleatoriamente x' en vecindad $V(x)$.
- 5 Sea $\Delta = f(x') - f(x)$.
- 6 Si $\Delta < 0$ entonces $x = x'$.
- 7 En otro caso generar aleatoriamente $l \in U(0,1)$.
- 8 Si $l < \exp(\Delta/c)$ entonces $x = x'$.
- 9 Hasta llegar a equilibrio.
- 10 Actualizar temperatura c .
- 11 Hasta completar enfriado.
- 12 Leer Solución.
- 13 Si Solución viola restricción de capacidad y factibilidad.
- 14 Repetir.
- 15 Perturbe multiplicador de Lagrange.
- 16 Hasta Solución factible.
- 17 Leer solución final.

Figura 2. Algoritmo recocido simulado para el problema del tamaño del lote económico.

Donde D_i es la demanda, $r = 1$ (demanda base), $r = 3$ (demanda intermedia) y $r = 4$ (demanda alta).

Tabla 2. Datos del problema [2].

i	a (\$)	D (u/día)	P (u/día)	t (días)	$h \times 10 \exp 5$ (\$/u. día)
1	15	100	30000	0,125	0,000002708
2	20	100	8000	0,125	0,00007396
3	30	200	9500	0,25	0,00005313
4	10	400	7500	0,125	0,00004167
5	110	20	2000	0,5	0,00116
6	50	20	6000	0,25	0,0001115
7	310	6	2400	1	0,000625
8	130	85	1300	0,5	0,002458
9	200	85	2000	0,75	0,000375
10	5	100	15000	0,125	0,00001667

En la Tabla 3 se muestran los valores de los parámetros del algoritmo utilizados en la primera fase de pruebas, así como el esquema de enfriamiento, el cual en esta implementación es del tipo geométrico.

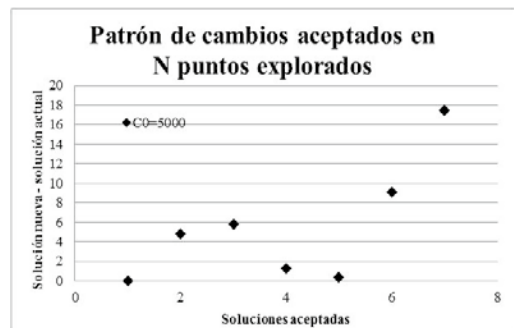
Tabla 3. Parámetros de RS.

Factor	Valor
c_0 inicial	5000
Número de soluciones vecinas exploradas	N
	$10N$
	$100N$
Esquema de enfriamiento	$c = \alpha c$
Rapidez de enfriamiento (α)	0,95
Sistema frío (ϵ)	0,1

En las implementaciones de técnicas metaheurísticas se debe determinar de manera experimental el comportamiento del algoritmo variando los valores de los parámetros. El valor inicial del parámetro de control c_0 debe asegurar que al inicio se acepten la mayor parte de las soluciones de mala calidad, en esta implementación el valor asegura que se aceptan al menos un 50% de los vecinos generados. En la Figura 3 se muestra a modo de ilustración la dispersión de los cambios aceptados para $r = 4$, de acuerdo al número de puntos explorados y $c_0 = 5000$. El número de puntos o soluciones exploradas a un cierto valor c debe ser tal que se asegure una buena evaluación de la vecindad de la solución.

En cada escenario de demanda se realizaron las pruebas variando el número de vecinos: N , $10 \times N$, $50 \times N$ y $100 \times N$ (Tabla 3). En cuanto al sistema de descenso de temperatura, se utilizó el esquema geométrico que es el recomendado para las implementaciones iniciales de RS [23]. Para la constante de la rapidez de enfriamiento α la literatura recomienda un valor tal que no provoque una convergencia prematura hacia un óptimo local, debiéndose seleccionar un valor entre (0,8-0,99), se seleccionó por el momento un valor 0,95, el sistema se considera frío cuando el valor de c es menor a 0,1 [23, 24].

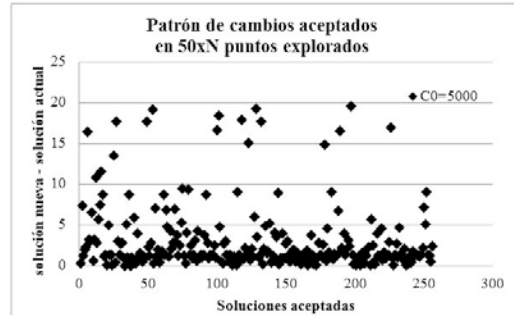
Los resultados se muestran en la Tabla 4 a y b, la cual se describe a continuación: la 1era columna muestra la relación $\sum \frac{D}{P}$, la 2da columna presenta la solución devuelta por el método de programación dinámica de Bomberger, la 3era columna presenta los resultados obtenidos con AG (mejor resultado reportado), la 4ta columna contiene el costo obtenido con RS en las distintas pruebas y la 5ta columna presenta el número de puntos explorados.



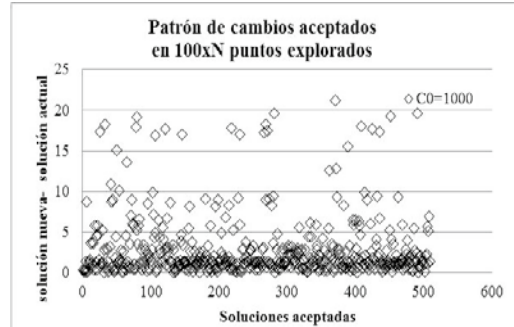
a) N



b) $10 \times N$



c) $50 \times N$



d) $100 \times N$

Figura 3. Dispersión de cambios aceptados vs número de vecinos explorados.

Tabla 4. Costo devuelto por RS.

a) $\sum \frac{D_i}{P_i} = 0,66$

$\sum \frac{D_i}{P_i}$	Bomberger	AG	RS	Ptos. expl.
0,66	7178,4	7024,11	7024,59	10
			7023,87	100
			7026,94	100
			7029,36	500
			7029,18	500
			7024,41	1000
			7027,02	1000

b) $\sum \frac{D_i}{P_i} = 0,88$

$\sum \frac{D_i}{P_i}$	Bomberger	AG	RS	Ptos. expl.
0,88	8796	8782,42	8788	10
			8797	10
			8782,36	100
			8782,34	100
			8781,96	500
			8786,35	500
			8782,49	1000
			8782,97	1000

Para el caso de $\sum \frac{D_i}{P_i} = 0,66$ los costos obtenidos con RS presentan una diferencia promedio de 0,075% con respecto a la mejor solución reportada con AG, siendo la máxima 0,27%; el mejor resultado que devuelve RS tiene un costo de 7023,87, siendo el ciclo básico 30,46. Dicho costo es -0,0034% menor al mejor resultado con AG, el cual reporta un valor de ciclo básico igual a 30,458.

Para las pruebas con $\sum \frac{D_i}{P_i} = 0,88$ se obtiene una diferencia promedio con respecto a los costos reportados con AG de 0,034%; el mejor resultado obtenido en esta serie de pruebas con RS registró un ciclo básico de 38,444 días, donde la diferencia en costo es -0,0052% con respecto al obtenido con AG. El valor del ciclo básico T obtenido con AG es 38,4442.

Igualmente, se muestra en la última columna el número de puntos explorados a un cierto valor de c . En el caso de las pruebas con $\sum \frac{D_i}{P_i} = 0,88$ la diferencia en los resultados se debe al valor del ciclo básico T , ya que los valores de las frecuencias son similares a los reportados con AG.

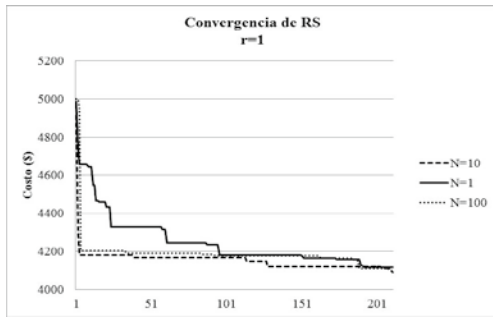
En las Figuras 4a, 4b y 4c se ilustra la manera en que RS converge hacia la solución de dos pruebas en cada escenario de demanda y el número de soluciones exploradas (la prueba con 500 se omite para mejorar la legibilidad de las figuras). Se observa, por ejemplo, que una muestra de N vecinos no asegura una buena exploración del entorno y el resultado es una solución de mala calidad dada la lenta convergencia; de hecho, sería necesario incrementar la constante de la rapidez de enfriamiento para que el descenso fuera más lento, sin embargo, esto repercutirá en el tiempo de ejecución.

En comparación al utilizar $10 \times N$ o bien $100 \times N$ vecinos, se observa que rápidamente el algoritmo va obteniendo soluciones de buena calidad. En cuanto al tiempo de ejecución, este crece de forma lineal y se encuentra entre 20 y 130 milisegundos (Figura 5).

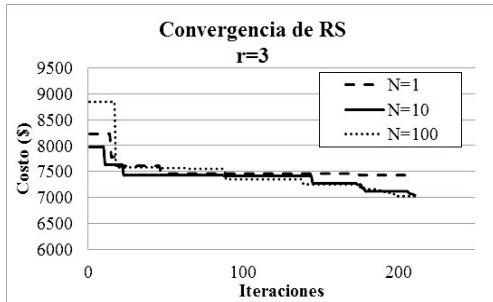
La segunda parte de las pruebas se realizaron variando la relación $\sum \frac{D_i}{P_i}$ tal y como se muestra en [3]; en este trabajo se realizaron pruebas con N , $10 \times N$ y $100 \times N$ para el número de soluciones vecinas exploradas. En esta serie de pruebas el algoritmo RS obtiene resultados que en general se encuentran dentro de una región muy cercana a los reportados anteriormente con AG, por lo que se refuerza el argumento de que en varias de las pruebas se ha encontrado una solución cercana al óptimo [3].

Cabe señalar que en un caso se observó una diferencia significativa con respecto al resultado previo de la literatura. Los resultados se muestran en las Tablas 5a, 5b y 5c y se describen a continuación: la 1era columna muestra la relación $\sum \frac{D_i}{P_i}$, la segunda columna muestra la solución independiente, la 3era contiene el mejor resultado obtenido con AG y reportadas en [3], la 4ta es el resultado devuelto con RS, la 5ta presenta el valor de T , finalmente la última columna es el número de puntos o soluciones vecinas exploradas.

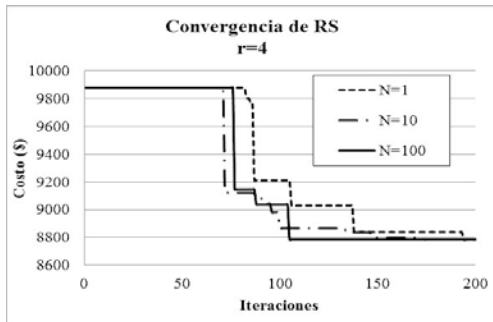
Para el caso $\sum \frac{D_i}{P_i} = 0,99$ la mejora es de 14,638% con respecto al resultado obtenido con AG y es la mejora más importante obtenida de toda la implementación, para $\sum \frac{D_i}{P_i} = 0,98$ la mejora es de 0,083%, con $\sum \frac{D_i}{P_i} = 0,97$ se alcanza una mejora de 0,021% y finalmente para $\sum \frac{D_i}{P_i} = 0,95$ la mejora es de 0,881%.



a) $r=1, \sum \frac{D_i}{P_i} = 0,22$



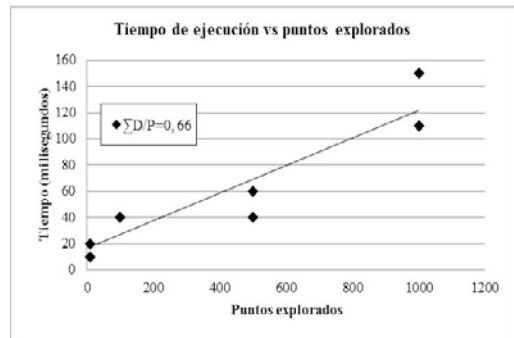
b) $r=3, \sum \frac{D_i}{P_i} = 0,66$



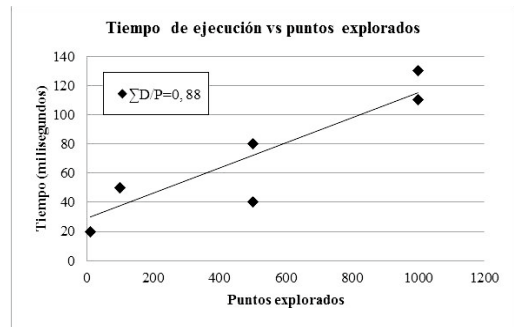
c) $r=4, \sum \frac{D_i}{P_i} = 0,88$

Figura 4. Convergencia de RS para los escenarios de demanda propuestos por Bomberger, $c_0=5000$.

El algoritmo RS devuelve soluciones con costos muy cercanos a los reportados anteriormente en los casos $\sum \frac{D_i}{P_i} = 0,86, 0,83, 0,80, 0,65, 0,60, 0,55$ y $0,50$; en las pruebas con $\sum \frac{D_i}{P_i} = 0,92, 0,89$ y $0,75$ y $0,70$, RS obtiene soluciones que si bien son muy cercanas a la solución óptima, quedan por arriba de las reportadas anteriormente: para las pruebas con $\sum \frac{D_i}{P_i} = 0,92$ la mejor solución que devuelve RS se aproxima a un $0,019\%$, con $\sum \frac{D_i}{P_i} = 0,89$ RS se aproxima a $0,0207\%$ del costo reportado anteriormente, para la prueba con $\sum \frac{D_i}{P_i} = 0,75$ RS



a) $\sum \frac{D_i}{P_i} = 0,66$



b) $\sum \frac{D_i}{P_i} = 0,88$

Figura 5. Tiempo de ejecución (milisegundos) vs puntos explorados a distintos valores de $\sum \frac{D_i}{P_i}$.

se aproxima a $0,0871\%$ del costo obtenido con AG y con $\sum \frac{D_i}{P_i} = 0,70$ RS se aproxima a $0,0407\%$ del mejor resultado obtenido con AG.

CONCLUSIONES

En esta investigación se implementó el algoritmo Recocido Simulado al problema de la programación del tamaño económico de lote, estudiado desde hace varias décadas. La aportación de este trabajo es la obtención de cotas más restringidas para ambas variables, también se muestra una estrategia para una selección eficiente de las variables enteras, y realizar una exploración con un mejor control.

El algoritmo RS obtiene resultados similares a los reportados en [3] con AG reforzando la conclusión de que se ha obtenido la solución óptima; en algunas de las pruebas RS devuelve soluciones con un mejor costo al reportado con anterioridad. En los

Tabla 5. Resultados de las pruebas.

a) $\sum \frac{D_i}{P_i}$ alto

$\sum \frac{D_i}{P_i}$	S.I.	AG	RS	T	Ptos. expl.
0,99	7936,17	55544,57	47414	373,912	10
					100
					1000
0,98	7905,51	24533,82	24513,4	187,97	10
					100
					1000
0,97	7874,53	17143,83	17140,3	125,03	10
					100
					1000
0,95	7811,61	12018,08	11912,3	74,55	10
					100
					1000
0,92	7714,73	9745,8	9763,25	54,197	10
			9757,96	54,11	100
			9747,67	53,94	1000
0,89	7614,76	8874,55	8890	42,31	10
			8878,78	41,9	100
			8876,39	41,82	1000

b) $\sum \frac{D_i}{P_i}$ intermedio

$\sum \frac{D_i}{P_i}$	S.I.	AG	RS	T	Ptos. expl.
0,86	7511,59	8553,31	8566,13	36,29	10
			8490,445	38,707	100
			8485,33	30,44	1000
0,83	7405,09	8250,29	8262,79	36,9	10
			8250,83	35,5	100
			8250,06	35,27	1000
0,8	7295,11	8096,01	8095,93	36,07	10
			8100,38	33,74	100
			8096,54	36,19	1000
0,75	7103,67	7789,63	7797,26	31,69	10
			7798	33,4	100
			7796,42	32,79	1000

experimentos donde la relación $\sum \frac{D_i}{P_i}$ es alta, RS obtiene mejoras significativas con respecto a lo reportado anteriormente con AG.

Si bien RS obtuvo resultados similares, aún quedan líneas por explorar: realizar pruebas con instancias generadas de manera aleatoria, un esquema de enfriamiento distinto o bien un nuevo esquema de obtención de las soluciones vecinas más eficiente.

c) $\sum \frac{D_i}{P_i}$ bajo

$\sum \frac{D_i}{P_i}$	S.I.	AG	RS	T	Ptos. expl.
0,7	6901,34	7395,46	7579,79	34,14	10
			7398,84	33,67	100
			7398,475	33,62	1000
0,65	6687,13	6914,7	6924,22	30,026	10
			6925,108	30,21	100
			6911,809	30,74	1000
0,6	6459,91	6621,75	6601,01	29,43	10
			6602,75	29,43	100
			6607,72	30,61	1000
0,55	6218,25	6328,67	6343,09	28,773	10
			6326,507	29,56	100
			6323,42	28,02	1000
0,5	5960,45	6038,41	5993,68	28,247	10
			6018,57	26,579	100
			6018,776	30,17	1000

De igual forma, sería interesante aplicar el algoritmo resolviendo una aplicación real. También queda por explorar el uso de diseños experimentales para estudiar el efecto de los parámetros del algoritmo y su posterior calibración, en aras de obtener un algoritmo eficiente y robusto.

AGRADECIMIENTOS

Los autores agradecen al árbitro sus valiosas observaciones, las cuales enriquecieron y mejoraron este trabajo.

REFERENCIAS

- [1] J. Rogers. "A computational approach to the economic lot scheduling problem". Management Science. Vol. 4, Issue 3, pp. 264-291. April, 1958. ISSN 0025-1909.
- [2] E. Bomberger. "A dynamic approach to a lot size scheduling problem". Management Science. 12, Issue 11, pp. 778-784. July, 1966. ISSN 0025-1909.
- [3] M. Khouja, Z. Michalewicz, W. y M. Wilmot. "The use of genetic algorithms to solve the economic lot size scheduling problem". European Journal of Operational Research. Vol. 110, Issue 3, pp. 509-524. November, 1998. ISSN 0377-2217.

- [4] F. Hanssmann. "Operation Research in Production and Inventory". Wiley, New York. 1962.
- [5] S.E. Elmagraby. "The economic lot scheduling problem (ELSP): review and extensions". *Management Science*. Vol. 24, Issue 6, pp. 587-598. February, 1978. ISSN 0025-1909.
- [6] W.L. Maxwell. "The scheduling of economic lot sizes". *Naval Research Logistics Quarterly*. Vol. 11, Issue 2, pp. 89-124. June, 1964. ISSN 1520-6750.
- [7] C. Delporte and L. Thomas. "Lot sizing and sequencing for N products and one facility". *Management Science*. Vol. 23, Issue 10, pp. 1070-1079. June, 1977. ISSN 0025-1909.
- [8] G. Dobson. "The economic lot-scheduling problem: achieving feasibility using time-varying lot sizes". *Operations Research*. Vol. 35, Issue 5, pp. 764-771. September, 1987. ISSN 0030-364X.
- [9] S.A. Raza, A. Akgunduz and M.Y. Chen. "A tabu search algorithm for solving lot scheduling problem". *Journal of Heuristics*. Vol. 12, Issue 6, pp. 413-426. July, 2006. ISSN. 13811231.
- [10] S.A. Raza and A. Akgunduz. "A comparative study of heuristic algorithms on economic lot scheduling problem". *Computers and Industrial Engineering*. Vol. 55, Issue 1, pp. 94-109. January, 2008. ISSN. 0360-8352.
- [11] F.M. Rafiei. "A metaheuristic algorithm for solving economic lot scheduling problem". *Proceedings of the 2010 International Conference on Industrial Engineering and Operations Management*. Dhaka, Bangladesh. 2010.
- [12] S. Axsäter "Inventory control". Springer, New York. 2006.
- [13] J. Grznar and C. Riggle. "An optimal algorithm for the basic period approach to the economic lot scheduling problem". *OMEGA*. Vol. 25, Issue 3, pp. 355-364. June, 1997. ISSN 0305-0483.
- [14] M.J. Yao and S.E. Elmagraby. "The economic lot scheduling problem under Power-of-Two policy". *Computers and Mathematics with Applications*. Vol. 41, Issue 10, pp. 1379-1393. May-June, 2001. ISSN 0898-1221.
- [15] M.J. Yao. "The economic lot scheduling problem without capacity constraints". *Annals of Operations Research*. Vol. 133, Issue 1-4, pp. 193-205. January, 2005. ISSN 0254-5330.
- [16] S. Madigan "Scheduling a multiproduct single machine system for an infinite planning period". *Management Science*. Vol. 14, Issue 11, pp. 713-719. July, 1968. ISSN 0025-1909.
- [17] S.K. Goyal. "Scheduling a multiproduct single machine problem". *Operational Research Quarterly*. Vol. 24, Issue 2, pp. 261-269. June, 1973. ISSN 1473-2858.
- [18] C.L. Doll and D.C. Whybark. "An interactive procedure for the single-machine multi-product lot scheduling problem". *Management Science*. Vol. 20, Issue 1, pp. 50-55. September, 1973. ISSN 0025-1909.
- [19] P. Park and D. Yun. "A stepwise partial enumeration algorithm for the economic lot scheduling problem". *IIE Transactions*. Vol. 16, Issue 4. pp. 363-370. April, 2004. ISSN 0740-817X.
- [20] P. Phillipoom, L.P. Rees and B.W. Taylor. "Solving the economic lot-scheduling problem using the method of prime periods". *Decision Sciences*. Vol. 20, Issue 4. pp. 794-809. December, 1989. ISSN 1540-5915.
- [21] D.C. Chatfield. "The economic lot scheduling problem: A pure genetic search approach". *Computers and Operations Research*. Vol. 34, Issue 10, pp. 2865-2881. October, 2007. ISSN. 0305-0548.
- [22] S. Hainan. "On the economic lot scheduling problem". *Doctoral Dissertation*, National University of Singapore. 2007. Date of visit: August 12, 2011. URL: <http://scholarbank.nus.edu.sg/handle/10635/13242>.
- [23] S. Kirkpatrick, C.D. Gellat, and M.P. Vecchi. "Optimization by simulated annealing". *Science*. Vol. 220, Issue 4598. pp. 671-680. May 1983. ISSN 0036-807.
- [24] H. Hoos and T. Stützle. "Stochastic local search: foundations and applications". 1^a ed. Springer-Verlag. San Francisco, CA. 2005.
- [25] S. Viswanathan. "A new optimal algorithm for the joint replenishment problem". *Journal of the Operational Research Society*. Vol. 47, Issue 7, pp. 936-934. July, 1996. ISSN 01605682.
- [26] M. Bazaraa, H. Sherali and C.M. Shetty. "Nonlinear programming". John Wiley and Sons. New Jersey. 2006.