

Aplicación de un Algoritmo Genético para la Solución de Modelos de Inventario Estocásticos

Pedro Chahuara Quispe Marcos Rivas Peña

Resumen Generalmente la obtención de políticas óptimas para gestión de inventarios se basa en métodos analíticos en el campo de la investigación de operaciones. Los modelos que representan estas políticas muchas veces pueden llegar a ser tan complejos que su solución mediante métodos analíticos sea difícil e ineficiente. El presente trabajo busca presentar una alternativa para la solución de modelos de inventario estocásticos para los cuales los métodos usualmente aplicados no son factibles. Lo que se plantea es analizar el uso de un algoritmo genético para encontrar mejores soluciones a este problema. Esta técnica computacional del campo de la inteligencia artificial se aplica en este trabajo junto con un simulador de procesos que ayuda a dirigir la búsqueda de una solución satisfactoria.

Palabras Clave: algoritmos genéticos, inventario, simulación, optimización.

Abstract Generally getting optimal policies to inventory management is based on analytical methods from the research operations field. The models that represent these policies can become so complex than their solution by means of analytical model can be difficult and inefficient. This paper proposes an alternative to solve stochastic inventory models for which usually applied methods are not feasible. Here the use of a genetic algorithm to solve this problem is analysed. This computational technique of the artificial intelligence field is applied along with a process simulator that helps to carry out the search of a satisfactory solution.

Key words: genetic algorithms, inventory, simulation, optimization.

1. Introducción

Un inventario es un conjunto de bienes o materiales mantenidos durante un tiempo en un estado inactivo en espera de ser usados. Esto puede darse en la venta de un artículo o durante un proceso de producción donde se demanda algún material para la consecución del proceso productivo. Los tres objetivos principales del control de inventario son:

- Maximizar el servicio al cliente. Esto significa conservar en inventario suficientes artículos de manera que con un alto grado de probabilidad un cliente sea satisfecho haciendo uso del stock.
- Minimizar el costo de almacenamiento. Los gastos de almacenar artículos están en proporción a la cantidad que se almacena. De esto se desprende que tener un exceso en inventario produce problemas de costos demasiado altos.
- Minimizar el costo de ordenar. Este es independiente de la cantidad de artículos ordenados, se produce cada vez que se realiza una orden para adquirir artículos y completar inventario; por lo tanto este costo depende de la frecuencia con que se coloquen las órdenes.

Estos objetivos están en conflicto. Por ejemplo, el servicio al cliente se puede mejorar conservando un alto número de artículos en inventario, pero esto lleva a un alto costo de almacenamiento o si se ordena con frecuencia para disminuir los costos de almacenamiento se aumenta el costo de ordenar.

Los modelos de inventarios planteados por la Investigación Operativa son modelos matemáticos que tiene como propósito encontrar los valores óptimos de las variables de decisión para obtener un costo de inventario mínimo y así encontrar un punto de equilibrio entre los tres objetivos mencionados anteriormente. Estas variables de decisión son:

Cantidad a ordenar (y). Dependiendo del modelo que se utiliza puede ser el tamaño del pedido que se hace al proveedor o una cantidad tal que se alcance un nivel de inventario óptimo.

Instante en el que se debe ordenar (R). Puede ser de acuerdo a un nivel de inventario conocido como punto de reorden o en un periodo fijo de tiempo.

Los costos en los que se incurre en el control de inventario son los siguientes:

- Costo de Pedidos u organización (K). Es el costo por colocar un pedido para reabastecer los

inventarios que es independiente del número de unidades pedidas.

- El costo de compra (C). El costo unitario de cada artículo pedido.
- El costo de conservación (h). Este es el costo por conservar un artículo en un periodo de tiempo.
- Costo de déficit (p). Es el costo de no satisfacer la demanda, es decir, el costo de que se acabe el artículo.

Cada uno de ellos forma parte del costo de inventario total, el cual se quiere minimizar.

Además de los costos, otros componentes de un modelo de inventario son:

Demanda de los Artículos (D). Es la cantidad de pedidos de un artículo que se hacen por unidad de tiempo.

Tiempo de entrega (L). Es el tiempo que transcurre desde el momento en que se coloca una orden al proveedor hasta que se reabastece el inventario.

En un modelo determinístico se sabe siempre el valor de estas variables, por otro lado un modelo de inventario estocástico es uno en el cual una o más variables no se conocen con certeza; sólo se sabe su distribución probabilística (o tendría que calcularse esta en base a resultados históricos). En situaciones reales la demanda y el tiempo de entrega suelen ser las variables estocásticas. Aunque también se podría tener incertidumbre en el costo de los artículos. Los modelos determinísticos pueden ser resueltos apropiadamente con métodos convencionales; sin embargo, no siempre son aplicables.

Uno de los modelos de inventario más utilizados es el Economic Order Quantity (EOQ) [4] donde se supone que cuando el nivel de inventario alcanza un cierto punto llamado punto de reorden (R) se debe realizar un pedido de tamaño fijo (y). Este es el modelo que servirá como base en el presente trabajo. En la figura 1 se muestra cómo es que varía el inventario en un lapso de tiempo.

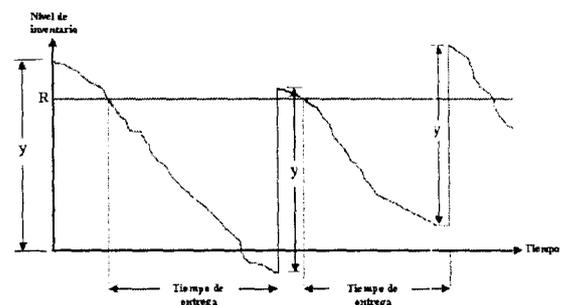


Figura 1. Variación del Inventario a través del Tiempo

La función del costo total de inventario se expresa como:

$$CT(y, R) = \frac{DK}{y} + h\left(\frac{y}{2} + R - E\{x\}\right) + \frac{pD}{y} \int_R^{\infty} (x - R) f(x) dx$$

Donde:

x = Variable aleatoria que representa la demanda durante el tiempo de entrega.

f(x) = función de distribución probabilística de x.

D = Esperanza de la demanda por unidad de tiempo.

Dado que la función del costo total es convexa[3], los valores óptimos y* y R* se obtiene a partir de:

$$\frac{\partial CT}{\partial y} = -\left(\frac{DK}{y^2}\right) + \frac{h}{2} - \frac{pDS}{y^2} = 0$$

$$\frac{\partial CT}{\partial R} = h - \left(\frac{pD}{y}\right) \int f(x) dx = 0$$

Encontrar los valores de las variables de decisión y y R a partir de estas ecuaciones conlleva un alto grado de dificultad, por lo que se utilizan métodos numéricos.

Para resolver este modelo en se describe un algoritmo numérico desarrollado por Hadley y Whitin. Este algoritmo garantiza un resultado óptimo en un número finito de iteraciones.

Limitaciones del modelo EOQ

En un escenario real las suposiciones de este modelo no siempre se cumplirían. Algunas de las razones por las cuales este modelo no siempre es aplicable son las siguientes:

- El modelo considera a la demanda como una variable continua y esta podría ser discreta. Además, para representar mejor al sistema de inventarios se debería utilizar como dato el tiempo entre llegadas de la demanda como variable aleatoria.
- En el modelo EOQ se trabaja únicamente con la esperanza de la demanda por unidad de tiempo pero la variable estocástica es la demanda durante el tiempo de entrega. Sería preferible utilizar directamente la demanda como variable estocástica.
- No se considera un límite de capacidad para el almacén. En la mayoría de casos se tiene un máximo posible de stock y sobrepasarlo acarrearía un costo adicional.

Adecuar la función de costo total para alterar uno de los supuestos haría que el uso de un método analítico para encontrar su mínimo sea inviable dada la complejidad del problema.

2. Algoritmos genéticos

Las técnicas de algoritmos genéticos ofrecen un mecanismo de búsqueda adaptativo basado en el principio darwiniano de reproducción y supervivencia de los más aptos[11]. En ellos se generan una "población de individuos" representando un conjunto de posibles parámetros de entrada para un problema determinado. Cada uno de los individuos es evaluado para que se le asigne un puntaje (fitness, o aptitud) de acuerdo a qué tan bien puede resolver el problema. De esta población se eligen a los mejores individuos y se les da mayores probabilidades de reproducirse mediante "cruce" con otros miembros de la población

Esto produce nuevos individuos, en una siguiente generación, como "descendientes" los cuales comparten características tomadas de sus padres. Los individuos menos aptos tienen una probabilidad menor de reproducirse; y, por lo tanto, tienden a extinguirse. Durante la ejecución del algoritmo genético (A.G.) se crean generaciones consecutivas, que contienen individuos cada vez más aptos (de mayor calidad) que sus sucesores. Lo que se espera es que en las últimas generaciones se obtengan soluciones cercanas al óptimo.

2.1. Algoritmo genético estándar

En la figura 2 se muestra las operaciones de un algoritmo genético estándar (A.G.E.)[12][6]. Se mantienen una población x1...xn la cual está formada por un conjunto de n individuos xi. Estos individuos son valores que puede tomar el parámetro de una función objetivo F(xi) que se quiere optimizar para resolver un problema dado. Cada individuo es representado en forma de "cromosoma", lo cual es una cadena definida sobre un alfabeto que codifica el valor del parámetro de la función objetivo. Si esta función objetivo recibiese n parámetros, el cromosoma del individuo codificaría un vector de tamaño n representando estos parámetros. Un individuo tiene asociado a un valor de aptitud que indica su capacidad para optimizar la función objetivo.

Inicialmente se crea una población inicial con valores aleatorios para iniciar la búsqueda de los valores óptimos de los parámetros de la función objetivo. Posteriormente se evalúa cada integrante de la población y se establece, según su valor de aptitud, si alguno de ellos tiene un resultado satisfactorio o si el estado de la búsqueda cumple con una condición de

parada. De no ser así, se selecciona los mejores del grupo y se utilizan para producir la siguiente generación por medio de la combinación de sus cromosomas (cruce). Algunos de los individuos de esta nueva generación son sometidos a un proceso de mutación por el cual sus cromosomas son alterados aleatoriamente.

Las condiciones de parada del algoritmo, suelen ser que el puntaje de desempeño de los individuos (valor de aptitud) converja con un valor especificado (terminación sobre convergencia) o que se llegue a un cierto número de generaciones (terminación sobre generación).

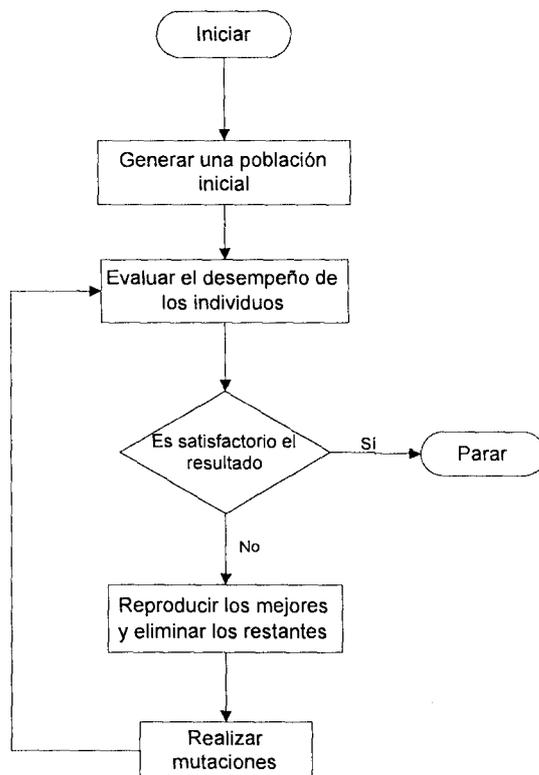


Figura 2. Algoritmo genético estándar

2.2. Exploración y explotación del espacio de búsqueda

Existen dos conceptos importantes asociados al desempeño de un algoritmo genético, la diversidad genética y la presión de selección [5]. El primero de ellos se refiere a la diferencia que existe entre los individuos de una población, si esta diferencia es grande se tiene que los individuos están más esparcidos en el espacio de búsqueda y por lo tanto se tiene menos posibilidades de convergir en un óptimo local. La diversidad genética favorece la exploración del espacio de búsqueda. Una diversidad pequeña sería un indicador que los individuos se encuentran alrededor de un punto.

La exploración de nuevas soluciones depende de la

tasa de mutación y del tamaño de la población.

El concepto de presión de selección se relaciona con el número de descendientes que tienen los individuos. Cuando el mejor individuo genera un gran número de descendientes, estos llegan a ocupar en pocas generaciones la población, y se tiene, por lo tanto, una presión de selección alta, la cual llevaría a concluir la búsqueda en un óptimo local. La presión de selección favorece la explotación del espacio de soluciones, y es un elemento importante porque dirige la búsqueda y acelera la convergencia, sin éste el algoritmo genético se convertiría en una búsqueda aleatoria.

3. Optimización de modelos de inventario mediante simulación y algoritmos genéticos

La estrategia que se plantea en el presente trabajo intenta integrar la simulación de procesos [2] en la búsqueda de una solución cercana a la óptima en un problema de optimización. Se trata de que la información necesaria para que la heurística encuentre los mejores valores para las variables de decisión sea obtenida de ejecuciones de simulación del proceso a optimizar.

La optimización basada en simulación es un proceso en el cual se busca los parámetros óptimos de un sistema evaluando sólo algunos de sus valores posibles por medio de simulación. El objetivo es minimizar los recursos computacionales utilizados y a la vez maximizar la información obtenida de un experimento de simulación. En [8] se utiliza este proceso para optimizar modelos de inventario determinísticos.

La Figura 3 muestra un modelo que puede ser aplicado a varios tipos de problemas de optimización. En nuestro caso la simulación será de un sistema de inventarios, utilizando el método de simulación de eventos discretos [9], y la optimización se llevará a cabo a través de algoritmos genéticos.

Inicialmente se debe proveer al sistema de simulación-optimización de los parámetros necesarios, para los inventarios estos son los componentes de costos: de ordenar, de almacenamiento y de faltante; y parámetros que necesita el algoritmo genético como probabilidad de cruce, probabilidad de mutación, tamaño de la población (1). Primero se crea una población inicial, donde cada individuo está formado por una dupla (y,R) cantidad a pedir y punto de reorden, respectivamente, a partir de esta se aplica el operador genético de selección el cual necesita asignar un valor de aptitud a cada individuo. Puesto que no se hace uso de una ecuación ni un método analítico para evaluar a los cromosomas, se decodifican estos en sus valores reales y se envían al simulador de inventarios como

parámetros de entrada y y R (2). Tras simular el sistema de inventarios con estos valores para un número de unidades de tiempo, se retorna al algoritmo genético el costo total obtenido (3), el cual se convierte en el valor de aptitud del individuo; a menor costo, mayor será su desempeño. El proceso se repite para todos los individuos de la población, luego a través de los operadores de cruce y mutación se tendrá una nueva generación, la cual también será evaluada en el simulador. Al completar un número de poblaciones especificadas por el usuario que estudia el sistema se retornará al mejor individuo de la última población como los valores óptimos (4).

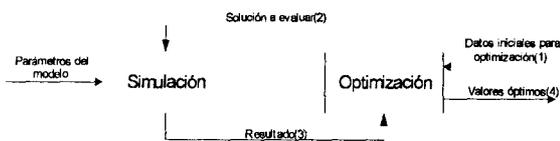


Figura 3. Modelo general de optimización basada en simulación

Diseño del algoritmo genético

Representación de los parámetros de decisión en el algoritmo

Cada individuo de una población en el algoritmo genético representará una posible configuración de los parámetros de decisión del modelo de inventario. De esta manera, un cromosoma será representado por una cadena binaria conteniendo dos sub cadenas, una para la cantidad a pedir (y) y la otra para el punto de reorden (R). En la figura 4 se muestra un ejemplo de cómo es cada individuo.

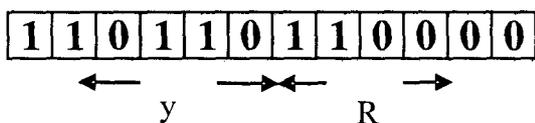


Figura 4. Representación de una solución como un cromosoma

Selección de individuos

Para este operador genético se realizó una variante del método de ranking de selección [7]. Este puede causar que el tiempo de convergencia del algoritmo sea mayor que utilizando el clásico método de la ruleta. Sin embargo, previene que “superindividuos” ocupen la población en pocas generaciones. Si ordenamos los individuos en orden decreciente de acuerdo a su valor de aptitud y asignamos a cada uno un valor entero, $pos[1, m]$, donde m es el tamaño de la población y para el mejor individuo se asume

$pos1=m$, para el segundo mejor $pos1=m-1$ y así sucesivamente; la función que indica cuántas veces se repetirá un individuo x_i en la siguiente generación es:

$$H(x_i) = \left[\frac{pos_i^f \cdot m}{\sum_{j=0}^{m-1} j^f} \right]$$

Donde f , llamado factor de selección, es un parámetro más del algoritmo, nótese que si $f > 1$ la relación entre el puesto de un individuo en el ranking y el número de copias que tiene en la siguiente generación deja de ser lineal.

Elitismo

Esta característica es incluida en los algoritmos genéticos para prevenir que los operadores genéticos afecten a los mejores individuos de una población, y de esta manera evitar que se pierdan las mejores soluciones encontradas hasta cierto punto. Para esto se toma un cierto número de individuos de una generación y se copian sin alteración alguna en la siguiente generación. En el algoritmo diseñado sólo se toma en cuenta el mejor individuo. Esto es suficiente para no perder la mejor solución encontrada.

Cruce

Se ha considerado una probabilidad de cruce, P_c , de 0.95 de manera predeterminada. La variante elegida es el cruce en dos puntos, pues este tipo de cruce tiene menos probabilidades de producir trastornos en buenos individuos.

Mutación

Para aplicar este operador se debe tener definida una probabilidad de mutación, P_m , para cada individuo que no esté en el grupo de élite. Este valor debe ser bajo para evitar una búsqueda aleatoria, se ha considerado un valor predeterminado de 0.05. Se genera un número aleatorio por cada gen de los individuos de la población; si este es menor que P_m , se lleva a cabo la mutación en el gen escogido.

Se agregó una estrategia adaptativa a este operador con el fin de disminuir el riesgo que el algoritmo termine la búsqueda en un óptimo local. Si durante un número r (parámetro) de iteraciones el mejor individuo de la población es el mismo, la probabilidad de mutación se incrementa en un valor i (parámetro) en cada generación consecutiva hasta

que se encuentre una solución mejor. Si llegase a encontrarse una solución más cercana a la óptima la tasa de mutación retoma su valor inicial. Esta modificación en el operador de mutación hace que el algoritmo ponga énfasis en la exploración cuando es apropiado.

Diseño del Simulador de inventario

Puesto que el sistema en estudio está representado por un modelo estocástico una única simulación para valores dados de μ y R , puede resultar en una evaluación poco fiable. Por ejemplo, la evaluación para cierta solución daría el costo $C1$ para la simulación $S1$ y en un costo $C2$ para la simulación $S2$, por lo que resulta beneficioso realizar varias evaluaciones y luego promediar el resultado. Cada una de estas simulaciones se debe realizar para un número determinado de unidades de tiempo, ingresado como parámetro. Si la longitud del periodo de evaluación es grande se garantiza un resultado más confiable, a cambio de aumentar el tiempo de respuesta de la solución.

A continuación se listan los componentes del simulador de inventario y en la figura 5 se muestra el flujo que sigue la simulación.

Variables:

- t: instante actual.
- x: nivel de inventario en el tiempo t.
- H: Costo total de mantenimiento hasta el tiempo t.
- O: Costo total de ordenar hasta el tiempo t.
- F: Costo total de faltante hasta el tiempo t.
- pr: Indica si se está esperando un pedido del proveedor(verdadero), o no(falso).
- pend: la cantidad de artículos que se tiene como demanda pendiente.
- t0: Tiempo de llegada de un cliente (variable estocástica).
- t1: Tiempo de llegada de una orden para completar inventario (variable estocástica).
- D: cantidad demandada por un cliente (variable estocástica).
- Media del tiempo entre llegadas de la demanda.

Parámetros:

- h: Costo de almacenamiento por unidad de tiempo.
- f: Costo de faltante por unidad.
- cf= Costo fijo de ordenar.
- cv= Costo de cada artículo al solicitarlo al proveedor.
- $\alpha(x)$: Función del costo de ordenar, donde x es la cantidad de unidades ordenada. Si el modelo no admite descuentos por cantidad está es lineal. Esta función retorna la suma del costo fijo de la orden y el costo variable.
- T: número de unidades de tiempo que se tomarán en cuenta para la simulación.

Parámetros ingresados por el algoritmo genético (solución a evaluar):

y: Cantidad a pedir en cada orden.

R: Punto de reorden.

Eventos:

Los eventos son la llegada de un cliente y la llegada de un pedido para completar inventario. Estos eventos se producen en los tiempos t_0 y t_1 , respectivamente.

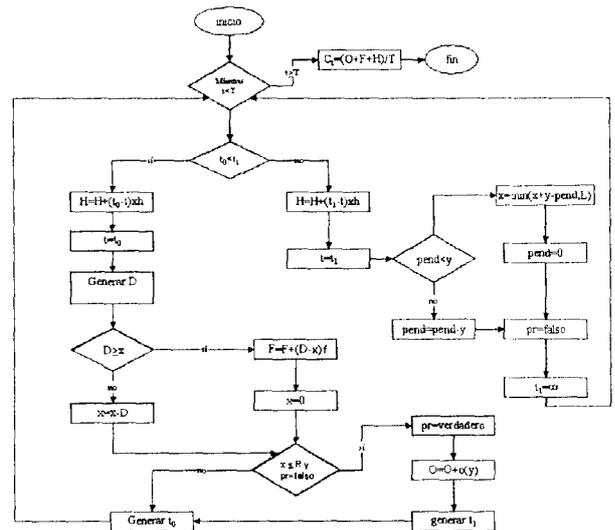


Figura 5. Flujo de la simulación de Inventario

Análisis de los datos de salida del simulador

La evaluación de cada individuo del algoritmo genético se hace mediante un proceso de simulación. Al simular un proceso se debe especificar el periodo de tiempo que se quiere representar; este periodo se divide en unidades de tiempo que se consideran como horas, días, semanas, etc., según sea el caso. La variable de interés es el costo de inventario por unidad de tiempo, puesto que se busca la configuración de parámetros que corresponda al mínimo costo. Denominaremos a esta como: CT, los valores que se obtienen a lo largo de una corrida de simulación serán CT1,CT2...CTn, donde cada uno de estos corresponde al costo total de inventario en una unidad de tiempo. El costo obtenido en dos unidades de tiempo será distinto puesto que se está tratando con modelos estocásticos y las variables del sistema se generan mediante número aleatorios. De esta manera, al considerar un promedio del costo total(CT) en toda la corrida como la medida para evaluar un individuo del algoritmo genético; mientras más larga sea la corrida tendremos un resultado más confiable para asignar al individuo y, por otro lado, el desempeño del algoritmo disminuirá porque tendrá q hacerse una corrida larga para todos los individuos de una generación.

El método de media por lotes[10] se utiliza en el presente trabajo para encontrar un tamaño apropiado para la corrida de simulación de manera que se guarde un equilibrio entre el tiempo de convergencia del algoritmo y el grado de confianza al asignar un valor de aptitud a un individuo.

Al dividir los resultados obtenidos en cada unidad de tiempo en k lotes de tamaño b , la media de cada lote se calcula como:

$$CTL_i(b) = \frac{1}{b} \sum_{j=1}^b CT_{(i-1)b+j}$$

Con esto se determina un intervalo de confianza de la forma:

$$P\left\{CT \in \left[\overline{CTL} - t_{k-1,\alpha/2} \frac{\delta}{\sqrt{k}}, \overline{CTL} + t_{k-1,\alpha/2} \frac{\delta}{\sqrt{k}} \right] \right\} = 1 - \alpha$$

donde:

CTL = Media de los lotes.

δ = desviación estándar de los lotes.

$t_{k-1,\alpha/2}$ = valor de la distribución t-student con $k-1$ grados de libertad y un nivel de significancia de $\alpha/2$.

Para la realización de los experimentos se determinó un tamaño de lote de $b=50$. El valor de k dependerá del valor escogido para α , el cual se ha fijado en 0.05 en todos los casos, y la amplitud del intervalo

$$[CTL - e, CTL + e]$$

A medida que el valor de e disminuya, se requiere un mayor número de lotes. Si se considera $e = \lambda \overline{CTL}$, entonces la amplitud del intervalo de confianza dependería del valor de la constante λ . La siguiente secuencia permite hallar el valor adecuado para k :

Paso 0: Seleccionar un valor inicial para k (este será 5 de manera predeterminada).

Paso 1: Efectuar una corrida de tamaño kb (unidades de tiempo) y dividir los resultados en lotes de tamaño b .

Paso 2: Basándose en las medias de cada lote, calcular CTL , δ y

$$E = t_{(n-1),\alpha/2} * \frac{\delta}{\sqrt{k}}$$

Paso 3: Si $e < \lambda CTL$, detener el proceso. En otro caso continuar la corrida con un lote más, hacer $k=k+1$, y regresar al paso 2.

1. Resultado Aplicando el algoritmo en un modelo de inventario EOQ

En esta prueba se tomó un modelo de inventario complejo. Aquí se considera, además del costo fijo de

realizar una orden al proveedor del inventario, un costo variable por los items solicitados, que puede tener un descuento según la cantidad pedida. En la bibliografía utilizada no se encontraron métodos matemáticos para hallar los parámetros óptimos, por lo cual, antes de hallar una solución al modelo probabilístico con el algoritmo genético, se convirtieron a determinísticos sus componentes probabilísticos para tener un resultado teórico como referencia para calcular el espacio de búsqueda en el cual el algoritmo genético realice la búsqueda.

Los datos del problema son los siguientes:

- $D \sim N(33,3)$ por día
- $k = \$100$
- $h = 0.066$ por día
- $p = \$10$
- $t_{ll} \sim N(1,0.01)$ días
- $L = \text{Uniforme en el intervalo } [0,3]$

$$o = \begin{cases} 2, & 0 < y < 350 \\ 1.7, & 350 \leq y < 700 \\ 1.4, & 700 \leq y < \infty \end{cases}$$

Donde:

D: Demanda.

H: Costo de almacenamiento por unidad de tiempo.

k: Costo fijo de realizar un pedido al proveedor.

o: Costo de solicitar una unidad al proveedor.

t_{ll} : Tiempo entre llegadas de la demanda.

L: Tiempo que transcurre entre el instante en que se coloca una orden y el reabastecimiento de inventario.

Solución del modelo determinístico

En esta sección se utilizó el método para resolver modelos de inventarios determinísticos EOQ con descuentos por cantidad que figura en las referencias. Los elementos probabilísticos del modelo anteriormente propuesto se cambiaron de la siguiente manera para adecuarlos al modelo teórico:

$D=33$ unidades por día (demanda continua).

$L=1.5$ días.

De este modo los valores óptimos son:

$y=700, R=49.5$

No es factible realizar una comparación del resultado obtenido con el algoritmo genético y el resultado teórico porque al pasar de un modelo probabilística a uno determinístico se trata de resolver dos problemas distintos. Sin embargo, se puede utilizar este resultado para determinar el espacio de búsqueda para el algoritmo genético. Ambos parámetros a optimizar se buscarán en el intervalo $[0,1024]$.

Aplicación de la heurística

En la tabla 1 se presenta el mejor costo por generación para tres ejecuciones independientes del algoritmo genético.

En la figura 6 se muestra el mejor costo total por día obtenido de realizar cinco ejecuciones independientes del algoritmo y promediar sus resultados en cada generación.

6	1			2			3		
	Y*	R*	Costo	Y*	R*	Costo	y*	R*	Costo
1	482	91	81.4302	784	98	77.9021	834	91	79.2089
5	715	84	75.1552	772	97	75.6424	701	91	75.0414
10	715	84	75.1552	772	97	75.6424	701	91	75.0414
15	731	86	74.8314	772	97	75.6424	733	91	74.6837
25	731	92	74.7342	708	97	75.4511	733	91	74.6837
30	730	92	74.4207	730	101	74.9989	733	91	74.6837
35	730	92	74.4207	730	98	74.8480	733	91	74.6837
40	730	92	74.4207	709	98	74.6319	733	91	74.6837
45	730	92	74.4207	709	98	74.6319	733	91	74.6837
50	730	92	74.4207	709	98	74.6319	733	91	74.6837
55	730	92	74.4207	709	98	74.6319	733	91	74.6837
60	730	92	74.4207	709	98	74.6319	733	91	74.6837
65	730	92	74.4207	709	98	74.6319	733	91	74.6837
70	730	92	74.4207	709	98	74.6319	733	91	74.6837
75	730	92	74.4207	709	98	74.6319	733	91	74.6837
80	730	92	74.4207	709	98	74.6319	733	91	74.6837
85	730	92	74.4207	709	98	74.6319	733	91	74.6837
90	730	92	74.4207	709	98	74.6319	726	98	74.6767
95	730	92	74.4207	709	98	74.6319	706	91	74.4121
100	730	92	74.4207	709	98	74.6319	706	91	74.4121

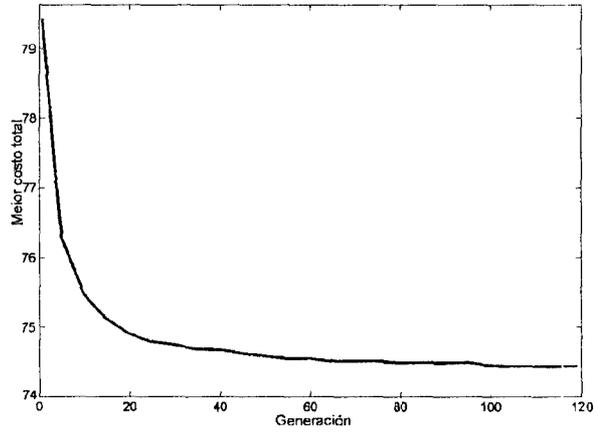


Figura 6. Mejor costo total encontrado (sobre 10 ejecuciones).

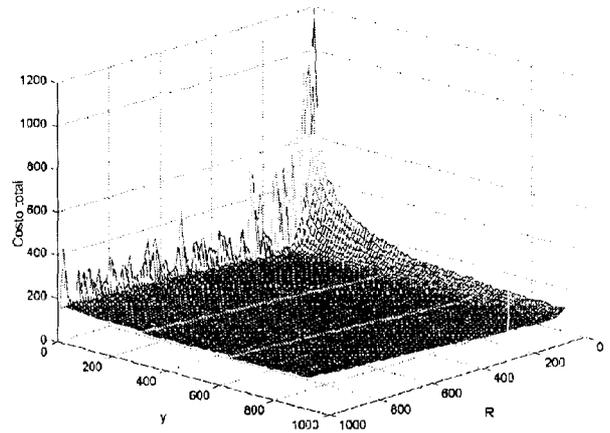


Figura 7. Espacio de soluciones para el problema

De la tabla se puede apreciar que existe variación entre los resultados de las ejecuciones del algoritmo, esto debido a la naturaleza probabilística del problema. Se puede notar claramente que a partir de la generación 25 el estado de la búsqueda, en cuanto al mejor individuo, no mejora. Para tener una idea del espacio de búsqueda sobre el cual trabaja el algoritmo genético se creó una matriz de tamaño 100x100 con valores de y y R variando entre 0 y 1000. Para cada solución (y,R) se realizó una ejecución de simulación y se tomó el costo total resultante. La Figura 7 muestra la superficie resultante. Esta figura puede explicar el porqué de que a partir de un momento de la búsqueda no se encuentra un mejor resultado hasta el fin, allí se aprecia que una gran parte del espacio de soluciones es una superficie casi plana donde existe variaciones al no tratarse de un modelo determinístico. Al final los resultados encontrados en las tres ejecuciones son bastante similares.

5. Conclusiones

- Los problemas de naturaleza estocástica son apropiados para los algoritmos genéticos dado el ruido de su espacio de soluciones.
- La optimización basada en simulación permite establecer un límite entre el modelo a optimizar y el componente de optimización haciendo independientes estos dos elementos.
- El análisis de la salida de simulación es un factor clave en la optimización basada en simulación. Si no se tiene un método que establezca un intervalo de confianza para la salida de la simulación, la búsqueda a través de la heurística puede llevar a resultados inexactos.
- Los recursos computacionales utilizados son proporcionales a la varianza de los parámetros estocásticos. Por ejemplo, si en las pruebas realizadas la varianza de la demanda fuese mayor, habría que realizar un mayor número de corridas de simulación.

- Es más sencillo cambiar un modelo de simulación que las ecuaciones que resuelven un problema de optimización. Como se vio en el segundo problema, cambiar las ecuaciones del modelo EOQ para que se considere un descuento por cantidad pedida puede ser una tarea complicada; sin embargo, en el modelo a simular no fue difícil realizar estos cambios.
- En los casos en los que sea posible aplicar métodos matemáticos de optimización es preferible evitar usar heurísticas puesto que estas requieren de un alto grado de recursos computacionales y conducen a resultados sub óptimos.

Referencias

1. Alberto Delgado (1998). Inteligencia Artificial y mini robots. Ecoe.
2. David Ríos Insua y Sixto Ríos Insua (2000). Simulación, Métodos y Aplicaciones. Alfaomega.
3. Handy A. Taha (1998). Investigación de operaciones. Pearson.
4. Hillier y Lieberman (2002). Investigación de operaciones. Mc Graw Hill.
5. Lance Chamber (2001). The practical handbook of genetic algorithms.
6. Marco Antonio Guimarães Dias (2000). Selection of Alternatives of Investment in Information for Oilfield Development Using Evolutionary Real Options Approach. PUC Brasil.
7. Melanie Mitchell (2003). An introduction to Genetic Algorithms. MIT Press.
8. Roger Ericsson (1996). Applying cooperative coevolution to inventory control parameter optimization. Tesis. Universidad de Skövde, Suecia.
9. Sheldon M. Ross (1999). Simulación. Limusa. Academic Press.
10. Stewart Robinson (2004). Simulation The practice of model.
11. Ueli Rutishauser (2002). The Genetic Algorithm - An example of an evolutionary algorithm. University of Applied Sciences, Suecia.
12. Ulrich Bodenhofer (2003). Genetic Algorithms: Theory and Applications.