

ALGORITMOS GENÉTICOS: UNA SOLUCIÓN ALTERNATIVA PARA OPTIMIZAR EL MODELO DE INVENTARIO (Q, r)

Por

María Andrea Arias Serna
Jorge Iván Londoño Marín

ESCUELA DE CIENCIAS Y HUMANIDADES

DEPARTAMENTO DE CIENCIAS BÁSICAS

MAESTRÍA EN MATEMÁTICAS APLICADAS

MEDELLÍN

Marzo 2009

ALGORITMOS GENÉTICOS: UNA SOLUCIÓN ALTERNATIVA PARA OPTIMIZAR EL MODELO DE INVENTARIO (Q, r)

Trabajo de investigación presentado como requisito para optar al título de
Magíster en Matemáticas Aplicadas

Por
María Andrea Arias Serna
Jorge Iván Londoño Marín

Director

María Eugenia Puerta Yepes

Doctor en Ciencias Matemáticas



ESCUELA DE CIENCIAS Y HUMANIDADES

DEPARTAMENTO DE CIENCIAS BÁSICAS

MAESTRÍA EN MATEMÁTICAS APLICADAS

MEDELLÍN

Marzo 2009

Nota de aceptación

Coordinador de la Maestría

Director del proyecto

Ciudad y fecha (día, mes, año):

Agradecimientos

Queremos expresar nuestra gratitud a todas las personas que contribuyeron a la culminación de la presente tesis, en especial a nuestra directora la Dr. Maria Eugenia Puerta Yepes por su generosa ayuda y consejos, a nuestros esposos, por su paciencia y amorosa dedicación y a todas las personas quienes nos hayan ofrecido su desinteresado apoyo.

Resumen

Suelen encontrarse muchos artículos y textos completos en torno al tópico teoría de inventarios, debido a que es un tema de gran interés al cual se enfrentan diariamente la mayoría de las empresas. Es por esto que día a día surgen generalizaciones de los modelos tradicionales de teoría de inventarios que consideran situaciones más aplicadas a la empresa, como por ejemplo: que hayan varios tipos de productos para ser inventariados [35], [48], varias clases de demandas en un mismo artículo [14], múltiples escalones para analizar, [48],[47], combinación de varias políticas de inventarios [44], [10], etc. En el presente trabajo de grado, será considerado un modelo de inventario con múltiples artículos y para un único escalón, el cual será ajustado a través del modelo de inventarios (Q, r) con aleatoriedad tanto en la demanda como en el tiempo de reposición, y en el cual los pedidos no despachados son considerados pedidos pendientes (no se considera pérdida de ventas). La decisión que será considerada como política de inventario, es minimizar la inversión total del inventario, sujeto a un número máximo de pedidos por período y a un nivel mínimo esperado de satisfacción.

Para hallar las variables de decisión Q y r del modelo, serán consideradas dos situaciones:

En la primera de ellas, se analiza el modelo para el caso en el que no hayan pedidos pendientes en el momento en que llega una nueva orden, para tal consideración, se desarrollan las siguientes tres metodologías de solución: (1) Por medio de *Multiplicadores de Lagrange*: se implementará el algoritmo propuesto en [37] y [48], con el fin de hallar el multiplicador de Lagrange μ , asociado a la restricción correspondiente al nivel de satisfacción, así como también se implementará lo propuesto en [48] para aproximar μ y calcular analíticamente el multiplicador λ asociado a la restricción correspondiente a la frecuencia de pedido, (2) A través de la metodología de *Optimización-Simulación*: la cual será implementada por medio del software de simulación Simul8, el cual tiene incorporado como paquete de optimización el OptQuest, finalmente (3) Se desarrollará el algoritmo de optimización heurística basado en *Algoritmos Genéticos*, el cual representa el método de solución propuesto en el presente trabajo de grado.

La segunda situación que será considerada en el modelo, será el caso en el que hayan pedidos pendientes en el momento en que llega una nueva orden, para dicha situación, se dará solución solo a través de *Algoritmos Genéticos* y de la metodología de *Optimización-Simulación* dado que para dicha situación no es posible considerar una solución a través de multiplicadores de Lagrange, debido a la no convexidad de la función objetivo causada por los pedidos pendientes.

La segunda situación considerada, es decir, que hayan pedidos pendientes en el momento en

que llega una nueva orden ver [55], ha sido poco estudiada, debido a que la función objetivo es compleja y en general no convexa, lo cual ha llevado a muchos autores a considerar sólo el caso en el que no hayan pedidos pendientes en el momento en que llega una nueva orden, como puede verse en [33], [35], [36], [37], [38] y [48]. El propósito en el presente trabajo de investigación es abordar el modelo considerando ambas situaciones, como se describe en los párrafos anteriores.

Situación bajo consideración Metodología de solución	Multiplicadores de lagrange	Optimización-Simulación	Algoritmos Genéticos
Con pedidos pendientes cuando llega una nueva orden		X	X
Sin pedidos pendientes cuando llega una nueva orden	X	X	X

Las pruebas realizadas y que son reportadas en el cuarto capítulo, muestran que el algoritmo de optimización propuesto, basado en Algoritmos Genéticos, es en general, un método eficiente para hallar buenas soluciones para las variables de decisión del sistema de inventarios tanto para el caso en el que hayan pedidos pendientes como para el caso en el que no los hayan.

Índice general

1. Introducción	1
2. Objetivos y Metodología de la investigación	3
2.1. Objetivos	3
2.2. Metodología y distribución de los contenidos	4
3. Temas y conceptos preliminares	5
3.1. El modelo de inventarios (Q, r)	5
3.1.1. Introducción	5
3.1.2. Formulación del modelo Mono-Artículo	8
3.1.3. Formulación del modelo Multi-Artículo	16
3.2. Algunas metodologías de solución para el modelo (Q, r)	18
3.2.1. Multiplicadores de Lagrange	18
3.2.2. Optimización-Simulación	19
3.2.3. Algoritmos Genéticos	26
4. Algoritmo Genético aplicado al modelo (Q, r)	45
4.1. Introducción	45
4.2. Resultados para el problema sin pedidos pendientes	46
4.2.1. Multiplicadores de lagrange	47
4.2.2. Algoritmos Genéticos	49
4.2.3. Optimización-Simulación	50
4.3. Comparación de resultados para el problema sin pedidos pendientes	52
4.4. Resultados para el problema con pedidos pendientes	54
4.4.1. Algoritmos Genéticos	54
4.4.2. Optimización-Simulación	78
4.5. Comparación de resultados para el problema con pedidos pendientes	78

Conclusiones	81
Problemas abiertos	83
Anexos	85
Bibliografía	151

Capítulo 1

Introducción

Existen diversos modelos matemáticos utilizados para describir la gestión de inventarios, entre ellos existen modelos estocásticos y determinísticos, modelos de revisión continua y de revisión periódica, modelos con un único artículo y de múltiples artículos, modelos para artículos perecederos y para artículos no perecederos, etc. En el presente trabajo será considerado el modelo de inventarios conocido como el modelo (Q, r) por ser uno de los más usados en la actualidad, tanto por académicos como por grandes y pequeños empresarios. Este modelo usualmente se emplea cuando se requiere un control constante de las cantidades para saber el nivel exacto en cada momento, y básicamente se aplica en artículos de importancia estratégica para la empresa o artículos de alta rotación.

El propósito, en la presente tesis, es minimizar a través de la implementación de un algoritmo basado en Algoritmos Genéticos, la inversión de un inventario multi-artículos, que es continuamente monitoreado, con demanda Poisson y tiempo de reposición Normal, donde los pedidos no despachados se consideran pedidos pendientes, y que está restringido a un cierto nivel de satisfacción y a un cierto número de pedidos por período, el cual se ha modelado a través de la política de inventario (Q, r) y para el cual se considere tanto el caso en el que hayan pedidos pendientes en el momento en el que llega una nueva orden, como para el caso en el que no los hayan, el cual puede ser descrito como (ver [33], [35], [36] y [37]):

$$\begin{aligned} & \text{mín} \sum_{i=1}^N c_i \left(\frac{Q_i+1}{2} + r_i - \theta_i + B(Q_i, r_i) \right) \\ \text{Sujeto a:} & \sum_{i=1}^N \frac{D_i}{Q_i} \leq F \\ & \frac{1}{D_{tot}} \sum_{i=1}^N D_i S(Q_i, r_i) \geq S \\ & Q_i \geq 1 \\ & r_i \geq 0 \\ & Q_i, r_i \text{ enteros.} \end{aligned} \tag{1.1}$$

Donde F es la frecuencia máxima de pedidos, $S(Q_i, r_i)$ es el nivel de satisfacción esperado del artículo i , S es el nivel de servicio mínimo deseado promedio, $B(Q_i, r_i)$ es el número esperado de pedidos pendientes para el artículo i , D_i es la demanda anual esperada del artículo i , θ_i es la

demanda del artículo i durante el tiempo de reposición del artículo i , c_i es el costo de producción del artículo i , Q_i es la cantidad de pedido del artículo i que se debe comprar o producir en cada ciclo, el punto de pedido r_i , es el nivel del inventario del artículo i en el cual se hace un nuevo pedido y D_{tot} es la suma de todas las demandas anuales promedio.

Como puede verse en [33], [35], [36], [37], [55], [38] y [48] se han desarrollado varias investigaciones alrededor de dicho modelo, encontrando que técnicas de optimización como la Optimización-Simulación y los multiplicadores de Lagrange, han tenido serias dificultades, debido a la no linealidad y a la no convexidad tanto de la función objetivo como de las restricciones del problema, lo cual ha ocasionado que el modelo usualmente sea considerado sólo para el caso en el que la demanda durante el tiempo de reposición sea menor que el punto de pedido, es decir, que no hayan pedidos pendientes en el momento en que llegue una nueva orden.

El objetivo principal al resolver el modelo de inventarios descrito anteriormente a través de la implementación de un algoritmo basado en Algoritmos Genéticos, es optimizar el modelo permitiendo que hayan pedidos pendientes en el momento en que llega una nueva orden y que las soluciones obtenidas sean de buena calidad y se puedan obtener en un tiempo de computo razonable.

Capítulo 2

Objetivos y Metodología de la investigación

2.1. Objetivos

El objetivo principal en el presente trabajo, es minimizar a través del metaheurístico Algoritmo Genético, el cual ha demostrado ser una potente herramienta de optimización, el modelo de inventarios (Q, r) multi-artículos, con demanda y tiempo de reposición aleatorios, restringido a número máximo de pedidos y a cierto nivel de satisfacción, y para el cual se considere tanto el caso en el que hayan pedidos pendientes en el momento en el que llega una nueva orden, como para el caso en el que no los hayan. Para lograrlo, previamente se deben alcanzar los siguientes objetivos específicos:

1. Llevar a cabo un análisis del estado de arte sobre la teoría de inventarios existente para el modelo (Q, r) , así como también la metodología de multiplicadores de Lagrange y las metodologías disponibles sobre Optimización-Simulación, usadas para aproximar la solución del problema descrito en (1.1).
2. Realizar un análisis detallado de la teoría sobre la cual están fundamentados los Algoritmos Genéticos, buscando realizar una correcta implementación del algoritmo.
3. Una vez establecida la base teórica, se procederá a diseñar un algoritmo, basado en la teoría de Algoritmos Genéticos, que calcule el costo mínimo de inversión anual de inventario con el fin de realizar algunas pruebas que permitan verificar o refutar la hipótesis enunciada.
4. Comparar los resultados obtenidos a través de multiplicadores de Lagrange, OptQuest y de Algoritmos Genéticos, para el caso en que el costo de inversión esté restringido a que no hayan pedidos pendientes en el momento en que llega una nueva orden.
5. Comparar los resultados obtenidos a través de las metodologías de Optimización-Simulación y de Algoritmos Genéticos, en el caso en que la función de inversión no tenga ninguna simplificación sobre el término que representa el número de pedidos pendientes.

2.2. Metodología y distribución de los contenidos

Se inicia esta investigación con una búsqueda de información sobre todos los temas relacionados a los que se propone en este escrito, y de una revisión exhaustiva de los métodos de resolución existentes para el modelo de inventarios propuesto, se han incluido una serie de conceptos preliminares además de la notación y definiciones básicas que son necesarias para la comprensión y contextualización de todo el contenido del trabajo; así como también se ha incluido los pseudocódigos de los algoritmos que serán implementados con el fin de dar solución al problema propuesto. Los capítulos guardan un orden en los contenidos, permitiendo con la lectura seguir un desarrollo adecuado.

La presente tesis está dividida en cuatro capítulos. El primer capítulo, **Objetivos y Alcance de la investigación**, expone los objetivos que se pretenden alcanzar en el desarrollo del trabajo, así como también la metodología a seguir para alcanzar los objetivos propuestos. En el capítulo 2, **Objetivos y Metodología de la investigación** se describen los objetivos planteados en la presente tesis y la forma en que se alcanzarán dichos objetivos. En el capítulo 3 **Temas y conceptos preliminares**, se sintetizan los tópicos necesarios para el desarrollo del capítulo central, **Algoritmos Genéticos aplicados al modelo de inventario multi-artículos**, y está dividido en tres secciones fundamentales, en la primera de ellas *El modelo de inventarios* (Q, r) describe el modelo de inventarios en cuestión, tanto para el caso mono-artículo, como para el caso multi-artículo; posteriormente se encuentra la sección de *Optimización-Simulación*, en el cual se describen los fundamentos y conceptos básicos que envuelve dicha metodología, y se trata de manera especial uno de los software (Simul8) utilizados para implementar dicha metodología y finalmente se encuentra la sección *Algoritmos Genéticos*, la cual corresponde a la metodología propuesta para dar solución al problema planteado. El capítulo central, correspondiente al capítulo 4 **Algoritmos Genéticos aplicados al modelo (Q, r) con múltiples artículos** está organizado de la siguiente forma: se comienza con una breve introducción de por qué la metodología de Algoritmos Genéticos es pertinente para resolver el problema de inventario, luego se definen algunos parámetros como: forma y tamaño de elegir la población inicial, el método de cruce y mutación elegido, las probabilidades de cruce y mutación, que en particular se tomarán para la implementación del algoritmo; dicho ajuste de parámetros será reportado a través de algunas gráficas comparativas, que justifiquen el por qué fueron seleccionados dichos parámetros, y al final del capítulo, se realizarán algunos ejemplos numéricos con el fin de realizar las comparaciones pertinentes con las metodologías de Optimización-Simulación, multiplicadores de Lagrange y Algoritmos Genéticos para el caso en que no hayan pedidos pendientes en el momento en que llega un nuevo pedido y con los métodos de Optimización-Simulación y Algoritmos Genéticos para el caso en el que hayan pedidos pendientes en el momento en que llega una nueva orden. Finalmente, en el último capítulo, **Conclusiones**, se exponen las conclusiones que se obtuvieron como resultado de la comparación de las soluciones obtenidas con las metodologías anteriormente mencionadas.

Capítulo 3

Temas y conceptos preliminares

Se comienza el capítulo haciendo una descripción sobre la mecánica básica del modelo de inventarios (Q, r) , dado que sobre dicho modelo se llevará a cabo la implementación del algoritmo propuesto.

3.1. El modelo de inventarios (Q, r)

En el presente trabajo se considerará el modelo de inventarios conocido como el modelo (Q, r) por ser uno de los más usados en la actualidad, tanto por académicos como por grandes y pequeños empresarios. Este modelo usualmente se emplea cuando se efectúa un control constante de las cantidades, es decir, cuando se desea saber el nivel exacto en cada momento y normalmente se emplea en artículos de importancia estratégica para la empresa o artículos de alta rotación.

3.1.1. Introducción

En la teoría de inventarios, dependiendo de los tipos de productos que se producen o se compran, de las épocas en que se demandan los productos, el tipo de demanda (aleatoria o determinística), las políticas de la empresa y de los recursos tanto humanos como físicos; los modelos mediante los cuales se estudian las políticas de inventarios pueden clasificarse en: modelos determinísticos (con demanda y tiempo de reposición conocidos) ó estocásticos (con demanda y/ó tiempo de reposición aleatorios) y modelos de revisión continua (el inventario es permanentemente monitoreado) ó modelos de revisión periódica (el inventario se revisa por periodos, por ejemplo, semanalmente, mensualmente, semestralmente, etc.).

Uno de los principales objetivos en los estudios sobre teoría de inventarios, es minimizar la inversión del inventario, por período (usualmente anual) con restricciones que pueden estar ligadas a ciertas condiciones como por ejemplo: a un nivel de satisfacción, a una capacidad de inversión, a una frecuencia de pedido por período, a la posibilidad de que exista o no déficit de productos, o a un número promedio de pedidos pendientes, etc. En el presente trabajo, se considerará el modelo de inventarios de revisión continua y multi-artículo que lleva a cabo la política (Q, r) , la cual puede describirse como sigue: La demanda del artículo i ocurre aleatoriamente según un proceso Poisson con promedio λ_i ; si la posición del inventario correspondiente al artículo i , alcanza el punto de pedido r_i , se hace un nuevo pedido de tamaño Q_i , el cual se asume que llega completo

y que es recibido después de un tiempo de reposición aleatorio l_i (tiempo que transcurre entre hacer un pedido y recibirlo), el cual se asume que tiene una distribución normal con media μ_{l_i} y desviación estándar σ_{l_i} . Todas las órdenes no satisfechas del inventario disponible se consideran pedido pendiente (no se consideran pérdidas de ventas) y se irán atendiendo según la política FIFO (primero en llegar, primero en salir).

La decisión que será considerada como política de inventarios es determinar los valores apropiados de Q_i y r_i , que minimicen el costo total de inversión sobre el inventario, sujeto a un número de pedidos por período y a cierto nivel de satisfacción. Dicha política, de acuerdo a lo expuesto en [37], puede ser modelada como sigue:

$$\begin{array}{ll} \text{mín} & \text{Inversión de Inventario} \\ \text{Sujeto a:} & \\ & \text{Frecuencia promedio de pedido} \leq F \\ & \text{Nivel de servicio Promedio} \geq S \end{array}$$

Donde F es el valor máximo de pedidos que se pueden hacer durante el período y S es el mínimo nivel de satisfacción esperado durante todo el período.

Revisión de literatura

A continuación se presenta una revisión de la literatura disponible en torno al modelo de inventario (Q, r) , la cual básicamente tiene dos objetivos, el primero de ellos, es citar las metodologías de optimización que han sido empleadas para calcular los parámetros Q y r del modelo, y el segundo, busca describir las diferentes políticas de decisión que han sido consideradas para enfrentar dicho modelo. En la descripción de cada documento se hará claridad en el número de artículos analizados, así como también se enfatizará en si el desarrollo de la solución es hecho considerando pedidos pendientes en el momento en que llega una nueva orden, o si tal caso no es analizado.

Hadley and Whitin [33] derivaron formalmente la función de costo exacta para el caso en que la demanda fuera Poisson y el tiempo de reposición fuera fijo o tuviera una distribución normal. Sin embargo para optimizar analíticamente la función objetivo, debido a que la función de costo es compleja y no siempre convexa, adicionaron algunas restricciones, como suponer que la demanda durante el tiempo de reposición es menor que el punto de pedido, es decir, consideran sólo el caso para el cual no existen pedidos pendientes en el momento en que llega una nueva orden. La decisión que se considera como política para dicho modelo, es minimizar el costo total de mantener el inventario, donde el costo total se define como la suma del costo por ordenar, el costo de almacenamiento y el costo de pedido pendiente.

Hopp et al. [35] consideran un modelo de inventario multi-artículo (Q, r) con demanda Poisson, tiempo de reposición fijo y donde las ventas no satisfechas se consideran pedidos pendientes, la decisión que se considera como política de inventario es minimizar la inversión del inventario restringido a cierta frecuencia de pedido y cierto nivel de satisfacción. El modelo es analizado sólo para el caso en el que no hayan pedidos pendientes en el momento en que llega una nueva orden. Para formular el modelo se desarrollan tres heurísticos, basados en simplificaciones sobre

el nivel de satisfacción, utilizan la teoría de multiplicadores de Lagrange para derivar expresiones simples para los valores de Q y r .

Platt et al. [55] desarrollaron de forma analítica la solución del modelo para un único producto con demanda Poisson, tiempo de reposición aleatorio y donde las ventas no satisfechas se consideran pedidos pendientes. Presentando dos soluciones heurísticas basados en el desarrollo del límite cuando el tamaño del pedido tiende a infinito y para el caso en que el tamaño del pedido tiende a cero, en ambos casos consideran la situación de que hayan pedidos pendiente en el momento en que llega una nueva orden. La decisión que se considera como política de inventario es minimizar la suma de los costos de almacenamiento y preparación, restringido a cierto nivel de satisfacción fijo.

Hopp and Spearman [36] formulan el modelo (Q, r) para un único producto, la decisión que se considera como política de inventario es minimizar el costo total de mantener el inventario. El nivel de satisfacción es aproximado por el nivel de servicio tipo II ($S(Q, r) \approx B(r)$), y el análisis se desarrolla inicialmente para el caso en que el tiempo de reposición es fijo y luego se generaliza a tiempo de reposición aleatorio. Se consideran dos situaciones en la primera de ellas, las ventas no satisfechas se consideren pedidos pendientes, y en la segunda las ventas no satisfechas se consideran perdida de ventas. Los parámetros Q y r del modelo son calculados de forma analítica a partir del criterio de la primera derivada.

Hopp and Spearman [37] consideran el modelo (Q, r) bajo tres perspectivas distintas, en la primera de ellas se aborda el modelo para un único producto con demanda Poisson, tiempo de reposición fijo y donde las ventas no satisfechas se consideran pedidos pendientes, la decisión que se considera como política de inventario es minimizar el costo total de mantener el inventario. Desarrollan un algoritmo iterativo para hallar los parámetros del modelo bajo el supuesto de tiempo de reposición fijo. En la segunda, la decisión que se considera como política de inventario es minimizar la inversión del inventario restringido a cierta frecuencia de pedido y cierto nivel de satisfacción. Restringen el modelo al caso en el que no hayan pedidos pendientes en el momento en que llega una nueva orden y el nivel de satisfacción al nivel de servicio tipo I ($S(Q, r) \approx G(r)$). Calculan las variables de decisión del modelo de forma analítica.

Finalmente, en la tercera perspectiva, generalizan el modelo para el caso multi-artículos con demanda Poisson, tiempo de reposición fijo y donde las ventas no satisfechas se consideran pedidos pendientes y el nivel de satisfacción es aproximado por el nivel de servicio tipo I, la decisión que se considera como política de inventario es minimizar la inversión del inventario restringido a cierta frecuencia de pedido y cierto nivel de satisfacción. Restringen el modelo al caso en el que no hayan pedidos pendientes en el momento en que llega una nueva orden. Presentan un algoritmo basado en la técnica de bisección para estimar los multiplicadores de Lagrange λ y μ , y con base a estos estiman los parámetros del modelo.

Mohammad et al. [48] proponen un algoritmo basado en la técnica de bisección al cual nombran como IHOA para resolver un modelo con dos escalones, en el cual se implementa la política (Q, r) tanto en el almacén como en los minoristas, la decisión que se considera como política de inventario es minimizar la inversión del inventario restringido a cierta frecuencia de pedido y cierto nivel de satisfacción. Utilizan la relajación lagrangiana para optimizar el modelo, calculan

de forma analítica el valor del multiplicador λ y emplean el algoritmo IHOA para hallar el valor del multiplicador μ . Los resultados obtenidos del algoritmo propuesto son comparados con los resultados obtenidos del OptQuest, mostrando que los resultados obtenidos por su algoritmo son superiores tanto en eficiencia y eficacia para sistemas de inventarios a gran escala.

Kopytov et al. [38] emplean el paquete de simulación Extend, para simular el modelo (Q, r) para un único producto, bajo los siguientes supuestos, la demanda de los artículos tiene un proceso Poisson y el tiempo de reposición tiene una distribución normal. En caso de que la demanda durante el tiempo de reposición sea mayor que el punto de pedido, se considera perdida de ventas, la decisión que consideran como política de inventario es minimizar el costo total de mantener el inventario.

3.1.2. Formulación del modelo Mono-Artículo

Supuestos en el modelo

Para desarrollar el modelo formal se asumen los siguientes supuestos, los cuales serán utilizados durante todo el trabajo:

- Los productos son analizados individualmente, es decir, no se considera la interrelación o dependencia con otros artículos.
- La demanda ocurre una a la vez de forma aleatoria y según un proceso Poisson (no hay demanda por lotes).
- La demanda no satisfecha es pedido pendiente.
- El tiempo de reposición es aleatorio con distribución normal.
- Existe una restricción sobre el número de reemplazamientos por año.
- Existe una restricción sobre el nivel de satisfacción del cliente.
- El modelo es analizado sólo para un período.
- Un período tiene una duración de un año.

Notación

Aunque, como se mencionó anteriormente el propósito en la presente tesis, es abordar el modelo de inventario multi-artículo, bajo la política (Q, r) , se comenzará con el desarrollo teórico para un único producto, dado que éste será la base que permita la extensión a varios artículos.

Para el desarrollo teórico del modelo con un único artículo, será empleada la siguiente notación, tomada de [37]:

- D demanda esperada por año (en unidades)
- X demanda (en unidades) durante el tiempo de reposición (en unidades), variable aleatoria

- σ_D desviación estándar de la demanda diaria (en unidades).
- D_t demanda en el día t (en unidades), variable aleatoria. Asumiendo que la demanda es estacionaria sobre el tiempo, así que D_t tiene la misma distribución para cada día t ; también se asume que diariamente las demandas son independientes una de otra.
- $d = E[D_t]$ demanda diaria esperada (en unidades).
- L tiempo de reposición (en número de días), variable aleatoria.
- $l = E[L]$ esperanza del tiempo de reposición (en número de días).
- σ_L desviación estándar del tiempo de reposición (en días).
- $\theta = E[X] = dl$ demanda esperada durante el tiempo de reposición (en unidades).
- σ desviación estándar de la demanda durante el tiempo de reposición.
- $g(x) = P(X = x)$ probabilidad de que la demanda durante el tiempo de reposición sea igual a x (función de densidad de probabilidad). Suponiendo que la demanda es discreta. Pero cuando sea conveniente se aproximará con una distribución continua.
- $G(X) = P(X \leq x) = \sum_{i=0}^x p(i)$ probabilidad de que la demanda durante el tiempo de reposición sea menor o igual a x (función de distribución acumulada)
- c costo de producción por unidad
- Q cantidad de pedido (variable de decisión)
- r punto de pedido (variable de decisión)
- $s = r - \theta$ nivel de stock de seguridad
- F frecuencia de pedido (orden de reposición por año)
- $S(Q, r)$ tasa de satisfacción (porcentaje de órdenes satisfechas de inventario)
- $B(Q, r)$ número promedio de pedidos pendientes no despachados
- $I(R)$ número promedio de inventario a mano

Dado que para describir la función de costo se hacen necesarios algunos conceptos como el nivel promedio de inventario a mano, el nivel de satisfacción y la frecuencia de pedido, se hará una breve descripción de la forma matemática que estas expresiones tomarán a lo largo de la presente tesis.

Nivel de Pedido Pendiente

El número de pedidos pendientes, el cual es denotado por $B(x, r)$ en los cuales se incurre durante un ciclo, según [37], es el número de pedidos que se tienen pendientes para cuando llegue una nueva orden, por lo tanto si la demanda durante el tiempo de reposición es x , el número de pedidos pendientes estará dado por:

$$B(x, r) = \begin{cases} 0 & \text{si } x < r \\ x - r & \text{si } x \geq r \end{cases}$$

Por lo tanto el número esperado de pedido pendiente puede ser calculado promediando sobre los posibles valores de x , es decir:

$$E(B(x, r)) = \bar{B}(r) = \sum_{x=0}^{\infty} B(x, r)p(x) = \sum_{x=r}^{\infty} (x - r)p(x)$$

para el caso discreto;

$$\bar{B}(r) = \int_0^{\infty} B(x, r)p(x)dx = \int_r^{\infty} (x - r)p(x)dx$$

y para el caso continuo.

Una forma para calcular la expresión anterior de manera simple en hojas de calculo, para el caso en que la demanda tenga una distribución Poisson y también para el caso en que la demanda es aproximada por la distribución normal (continua), como puede consultarse en [35], está dada por:

$$\bar{B}(r) = \theta p(r) + (\theta - r)[1 - G(r)]$$

Finalmente, ver [37], el número esperado de pedidos pendientes por año, al cual denotaremos por $B(Q, r)$, se obtiene multiplicando $\bar{B}(r)$ por el número esperado de ciclos por año, el cual está dado por $\frac{D}{Q}$, es decir

$$B(Q, r) = \frac{D}{Q} \bar{B}(r) = \frac{D}{Q} \int_r^{\infty} (x - r)p(x)dx$$

Nivel de Satisfacción

Suponiendo que la posición del inventario a mano actual es x y que todas las órdenes pendientes llegarán dentro del tiempo de reposición, la probabilidad de deficit, es decir, la probabilidad de que la demanda durante el tiempo de reposición sea mayor que el inventario disponible en cierto momento, está dada por:

$$\begin{aligned} P\{X \geq x\} &= 1 - P\{X < x\} \\ &= 1 - P\{X \leq x - 1\} \\ &= 1 - G(x - 1) \end{aligned}$$

Por lo tanto el nivel de satisfacción cuando la posición de inventario es x es uno menos la probabilidad de déficit, es decir $G(x-1)$, ver [35]. Dado que las Q posibles posiciones de inventario tienen igual probabilidad, el nivel promedio de satisfacción para el sistema completo es calculado como el promedio del nivel de satisfacción sobre las posiciones de inventario entre r y $r + Q$, y está dado por:

$$S(Q, r) = \frac{1}{Q} \sum_{x=r+1}^{r+Q} G(x - 1) = \frac{1}{Q} \sum_{x=r}^{r+Q-1} G(x)$$

Nivel de Inventario

El inventario neto, se define según [33] como el inventario a mano menos el pedido pendiente, es decir, el inventario neto es la diferencia de dos variables aleatorias, por lo tanto el valor esperado del inventario neto es igual al valor esperado del inventario a mano menos el valor esperado del pedido pendiente. De lo anterior, el inventario esperado a mano es igual al valor esperado de inventario neto más el valor esperado de pedido pendiente, donde el inventario neto esperado al comenzar un ciclo es $Q + r - \theta$ y al terminar el ciclo está dado por $r + 1 - \theta$, luego el promedio del inventario neto está dado por:

$$\frac{(Q + r - \theta) + (r + 1 - \theta)}{2}$$

Por lo tanto, el inventario esperado a mano $I(Q, r)$, está dado por:

$$I(Q, r) = \frac{(Q + r - \theta) + (r + 1 - \theta)}{2} + B(Q, r) = \frac{Q + 1}{2} + r - \theta + B(Q, r)$$

Frecuencia de Pedido

Finalmente el número de pedidos por año puede ser calculado como:

$$F(Q, r) = D/Q$$

Con la notación anterior el problema que describe la política de inventario (Q, r) para un único artículo queda descrita matemáticamente como:

$$\text{mín} \quad \frac{Q+1}{2} + r - \theta + B(Q, r)$$

$$\text{Sujeto a:} \quad \frac{D}{Q} \leq F$$

$$S(Q, r) \geq S$$

$$Q \geq 1$$

$$r \geq 0$$

$$Q, r \text{ enteros.}$$

Dado que el número de pedidos pendientes esperado por año está dado por:

$$B(Q, r) = \frac{D}{Q} \int_r^{\infty} (x - r)p(x)dx$$

La expresión que describe la inversión total en el inventario está dada por:

$$c \left(\frac{Q+1}{2} + r - \theta + \frac{D}{Q} \int_r^{\infty} (x - r)p(x)dx \right)$$

En vista a que el número de reemplazamiento por año esta dado por D/Q y que el nivel de servicio esta dado por $\frac{1}{Q} \sum_{x=r}^{r+Q-1} G(x)$.

Se puede formular matemáticamente el problema como sigue:

$$\text{mín } c \left(\frac{Q+1}{2} + r - \theta + \frac{D}{Q} \int_r^\infty (x-r)p(x)dx \right)$$

Sujeto a:

$$\frac{D}{Q} \leq F$$

$$\frac{1}{Q} \sum_{x=r}^{r+Q-1} G(x) \geq S \quad (3.1)$$

$$Q \geq 1$$

$$r \geq 0$$

Q, r enteros.

Como se afirma en [36] no existe una forma práctica para optimizar el problema (3.1), debido a que no es posible garantizar la convexidad de la función objetivo, en el caso en que la demanda durante el tiempo de reposición es mayor que el punto de pedido (es decir, cuando existen pedidos pendientes en el momento en que llega una nueva orden), ya que el término que representa la función de pedido pendiente por año

$$B(Q, r) = \frac{D}{Q} \int_r^\infty (x-r)g(x)dx$$

es convexa (como se probará el siguiente teorema) cuando la demanda durante el tiempo de reposición es menor que el punto de pedido.

Antes de enunciar y demostrar el teorema que caracteriza la función de pedido pendiente, se presenta un teorema que será utilizado en la demostración.

Teorema 3.1.1. *Sea HB la matriz Hessiana de la función de pedido pendiente, si HB es semidefinida positiva en una región del plano, entonces la función de pedido pendiente es convexa en dicha región.*

Así el teorema que describe la región en la cual la función de pedido pendiente es convexa es el siguiente:

Teorema 3.1.2. *Supóngase un número z tal que $g(x)$ es decreciente para $x \geq z$, entonces $B(Q, r)$ es convexa en la región descrita por*

$$R = \{(Q, r) / z \leq r < \infty, 0 < Q < \infty\}$$

Demostración. Sea X la variable aleatoria con función de densidad g , definimos la variable aleatoria

$$X_r = \begin{cases} 0 & \text{si } X < r \\ X - r & \text{si } X \geq r \end{cases}$$

luego

$$E(X_r) = \frac{D}{Q} \int_r^\infty g(x) (x - r) dx$$

Sea $G(y) = P(X > y) = \int_y^{+\infty} g(t) dt$, entonces $P(X_r > x) = P(X - r > x) = P(X > x + r) = \int_{x+r}^{+\infty} g(t) dt = G(x + r)$.

Se puede probar que

$$\int_r^\infty g(x) (x - r) dx = \int_0^\infty G(x + r) dx = \int_r^\infty G(x) dx$$

luego

$$B(Q, r) = \frac{D}{Q} \int_r^\infty G(x) dx$$

Ahora para obtener la matriz Hessiana de la función de pedido pendiente anterior, se efectúan las derivadas parciales respectivas.

$$\frac{\partial B}{\partial r} = \frac{D}{Q} (-G(r)) = -\frac{D}{Q} (G(r)) = -\frac{D}{Q} \int_r^\infty g(x) dx$$

$$\frac{\partial B}{\partial Q} = -\frac{D}{Q^2} \int_r^\infty G(x) dx$$

$$\frac{\partial^2 B}{\partial r^2} = -\frac{D}{Q} (-g(r)) = \frac{D}{Q} (g(r))$$

$$\frac{\partial^2 B}{\partial r \partial Q} = -\frac{D}{Q^2} (-G(r)) = \frac{D}{Q^2} (G(r))$$

$$\frac{\partial B}{\partial Q \partial r} = \frac{D}{Q^2} (G(r))$$

$$\frac{\partial^2 B}{\partial Q^2} = \frac{D}{Q^3} \int_r^\infty G(x) dx$$

Es decir, la matriz Hessiana del $B(Q, r)$ es

$$HB = \begin{bmatrix} \frac{\partial^2 B}{\partial r^2} & \frac{\partial^2 B}{\partial r \partial Q} \\ \frac{\partial^2 B}{\partial Q \partial r} & \frac{\partial^2 B}{\partial Q^2} \end{bmatrix} = \begin{bmatrix} \frac{\lambda}{Q} (g(r)) & \frac{\lambda}{Q^2} (G(r)) \\ \frac{\lambda}{Q^2} (G(r)) & \frac{2\lambda}{Q^3} \int_r^\infty G(x) dx \end{bmatrix}$$

$$\begin{aligned} \det(HB) &= \frac{\lambda}{Q} (g(r)) \frac{2\lambda}{Q^3} \int_r^\infty G(x) dx - \frac{\lambda^2}{Q^4} (G(r))^2 \\ &= \frac{\lambda^2}{Q^4} \left[2g(r) \int_r^\infty G(x) dx - (G(r))^2 \right] \\ &= \frac{\lambda^2}{Q^4} h(r) \end{aligned}$$

Como $\frac{\lambda^2}{Q^4}$ es positivo sólo se debe probar que $h(x) \geq 0$ para $r \geq z$.

Teniendo en cuenta que $G'(r) = -g(r)$ se tiene que la derivada de h está dada por

$$\begin{aligned} h'(r) &= 2g'(r) \int_r^\infty G(x) dx - 2G(r)g(r) - 2G(r)G'(r) \\ &= 2g'(r) \int_r^\infty G(x) dx \end{aligned}$$

Es claro que $\int_r^\infty G(x) dx \geq 0$, como g es decreciente para $r \geq z$, entonces $g'(r) \leq 0$ para $r \geq z$, lo que implica que $h'(r) \leq 0$ para $r \geq z$, donde se sigue que $h(r)$ es decreciente para $r \geq z$. Pero como $\lim_{r \rightarrow +\infty} h(r) = 0$, se concluye entonces que $h(x) \geq 0$ cuando para $r \geq z$. Así la matriz HB es definida positiva. Aplicando el Teorema 3.1.1 se obtiene el resultado. \square

Según [35], al no poderse garantizar la convexidad de la función que representa el número de pedidos pendientes se hace difícil eliminar la existencia de mínimo local el cual difiere del mínimo global, es por esto, que con el fin de poder obtener por lo menos una solución aproximada para (3.1), el problema es restringido al caso en que $r \geq \theta$, como puede verse en [35], [36], [37], [33], [45], [48]. Bajo dicha suposición, el inventario a mano es positivo cuando llega la nueva orden, debido a que la demanda durante el tiempo de reposición es menor que el punto de pedido, y por lo tanto, el número de pedidos pendientes es igual a cero cuando llega el nuevo pedido.

De lo anterior, considerando sólo el caso en que $r \geq \theta$, la función objetivo, que describe el costo de mantener el inventario, está dada por:

$$c \left(\frac{Q+1}{2} + r - \theta \right)$$

Como se puede observar, al considerar sólo el caso en que $r \geq \theta$, se logra obtener una función objetivo que es lineal en sus dos variables Q y r .

Otro problema que se presenta en la formulación anterior es que para calcular el nivel de satisfacción, el cual está dado por

$$S(Q, r) = \frac{1}{Q} \sum_{x=r}^{r+Q-1} G(x) = \frac{1}{Q} \sum_{x=r}^{r+Q-1} \left(\sum_{k=0}^x \frac{e^{-\theta} \theta^k}{k!} \right)$$

se requiere mucho tiempo computacional, debido a la gran cantidad de sumatorias que habría que calcular, es por esto que para simplificar la expresión que describe el nivel de satisfacción, son utilizados algunas aproximaciones, las cuales pueden consultarse en [36]. En éste trabajo, será utilizada sólo una de las tres aproximaciones propuestas en dicho artículo, *Nivel de servicio tipo I*, el cual es visto como la probabilidad de que no ocurra deficit durante el tiempo de reposición, es decir, la probabilidad de tener demanda menor que el punto de pedido, la cual está dada por $G(r)$, esto es

$$S(Q, r) \approx G(r)$$

Como se afirma en [35], la aproximación anterior representa una subestimación del nivel de satisfacción, debido a que la función de distribución $G(x)$ es una función creciente de x . Así que lo que se está tomando es el menor término del promedio, sin embargo, aunque la aproximación anterior puede subestimar seriamente el nivel de servicio, está es muy simple de calcular ya que ésta envuelve sólo a r y no a Q .

Teniendo en cuenta las anteriores simplificaciones, el problema de minimizar la inversión del inventario, como puede consultarse en [37], se puede escribir como:

$$\begin{aligned}
& \text{mín} \quad c\left(\frac{Q+1}{2} + r - \theta\right) \\
& \text{Sujeto a:} \quad \frac{D}{Q} \leq F \\
& \quad \quad \quad G(r) \geq S \\
& \quad \quad \quad Q \geq 1 \\
& \quad \quad \quad r \geq 0 \\
& \quad \quad \quad Q, r \text{ enteros.}
\end{aligned} \tag{3.2}$$

Para optimizar la función $c\left(\frac{Q+1}{2} + r - \theta\right)$, es de notar que la función objetivo varía de acuerdo a como varíen Q y r por lo tanto a menores valores tanto de Q como de r el costo sera menor. Así el menor valor de Q que minimiza la función y que satisface la restricción $\frac{D}{Q} \leq F$ o lo que es equivalente $Q \geq \frac{D}{F}$ está dado por $Q = \frac{D}{F}$. Después de elegir el valor de Q debemos hallar el menor valor de r que satisface que $G(r^*) \geq S$, asumiendo que G es normal podemos escribir $G(r^*) = \Phi\left(\frac{r^* - \theta}{\sigma}\right) \geq S$, donde Φ es la función de distribución acumulada de la distribución normal estándar, luego por el teorema del limite central si la demanda durante el tiempo de reposición tiene una distribución con media θ y desviación σ finitas entonces

$$Z = \frac{X - \theta}{\sigma} \approx N(0, 1)$$

se distribuye normal con media 0 y desviación 1, de donde

$$z = \frac{r^* - \theta}{\sigma}$$

por lo tanto, el punto de pedido, está dado por:

$$r^* = \theta + z\sigma$$

De acuerdo con [35], el primer efecto debido a la variabilidad del tiempo de reposición es incrementar la desviación estándar σ . Para calcular una fórmula para σ que considere variabilidad en el tiempo de reposición, teniendo en cuenta que X representa la demanda durante el tiempo de reposición, entonces

$$X = \sum_{t=1}^L D_t$$

Debido a que las demandas diarias son idénticas e independientemente distribuidas, es posible calcular la demanda esperada durante el tiempo de reposición como

$$E[X] = E[L]E[D_t] = ld = \theta$$

Sin embargo, variabilidad en los tiempos de reposición cambian su varianza. Usando la fórmula estándar para sumas de variables aleatorias independiente e idénticamente distribuidas, podemos calcular

$$Var(X) = E[L]Var(D_t) + E[D_t]^2Var(L) = l\sigma_D^2 + d^2\sigma_L^2$$

Donde la desviación estándar de la demanda del tiempo de reposición es

$$\sigma = \sqrt{\text{Var}(X)} = \sqrt{l\sigma_D^2 + d^2\sigma_L^2}$$

Para tener una fórmula más simplificada consideramos el caso en que la demanda sea Poisson, lo cual implica que $\sigma_D = \sqrt{d}$. Sustituyendo ésto en la ecuación anterior obtenemos

$$\sigma = \sqrt{ld + d^2\sigma_L^2} = \sqrt{\theta + d^2\sigma_L^2}$$

De lo anterior, el valor óptimo de r está dado por:

$$r^* = \theta + z\sigma = \theta + z\sqrt{\theta + d^2\sigma_L^2}$$

y el valor óptimo de Q , como fue calculado anteriormente, está dado por:

$$Q^* = \frac{D}{F}$$

3.1.3. Formulación del modelo Multi-Artículo

Para abordar el problema anterior con varios artículos se usará el subíndice i para representar el artículo i , $i = 1, 2, \dots, N$.

- N número total de distintos tipos de partes en el sistema
- D_i demanda esperada por año (en unidades) para el artículo i
- D_{tot} Demanda total
- c_i costo por unidad (en dólares) para el artículo i
- l_i tiempo de reemplazamiento (años) para el artículo i
- $\theta_i = D_i l_i$ demanda esperada durante el tiempo de reposición (en unidades) para el artículo i
- $g_i(x)$ función de densidad $P(X = x)$ probabilidad de la demanda durante el tiempo de reposición para el artículo i
- $G_i(x) = \int_0^x g_i(y) dy$ Función de distribución acumulada de la demanda durante el tiempo de reposición para el artículo i
- S promedio de nivel de servicio deseado
- F promedio deseado de frecuencia de pedido
- Q_i cantidad de pedido del artículo i (variable de decisión)
- r_i punto de pedido del artículo i (variable de decisión)
- $S(Q_i, r_i)$ nivel de satisfacción del artículo i (porcentaje de órdenes satisfechas de inventario)

- $B(Q_i, r_i)$ número promedio de pedidos pendientes no despachados

Para formular matemáticamente el modelo, recordemos que si la cantidad a ordenar para el artículo i es Q_i , entonces el promedio de inversión de inventario para la parte i es $c_i(\frac{Q_i+1}{2} + r_i - \theta + B(Q_i, r_i))$, la frecuencia de pedido para el artículo i es $\frac{D_i}{Q_i}$ y el nivel de satisfacción para el artículo i es $D_i S(Q_i, r_i)$. Por lo tanto, el promedio de inversión de inventario es $\sum_{i=1}^N c_i(\frac{Q_i+1}{2} + r_i - \theta + B(Q_i, r_i))$, el promedio de frecuencia de pedido es $\frac{1}{N} \sum_{i=1}^N \frac{D_i}{Q_i}$ y el promedio de nivel de satisfacción es $\frac{1}{D_{tot}} \sum_{i=1}^N D_i S(Q_i, r_i)$.

De lo anterior, el modelo multi-artículo que lleva a cabo la política (Q, r) , puede describirse como, ver ([35],[36],[37],[48])

$$\begin{aligned}
 & \text{mín} \quad \sum_{i=1}^N c_i \left(\frac{Q_i+1}{2} + r_i - \theta_i + B(Q_i, r_i) \right) \\
 \text{Sujeto a:} \quad & \frac{1}{N} \sum_{i=1}^N \frac{D_i}{Q_i} \leq F \\
 & \frac{1}{D_{tot}} \sum_{i=1}^N D_i S(Q_i, r_i) \geq S \\
 & Q_i \geq 1 \\
 & r_i \geq 0 \\
 & Q_i, r_i \text{ enteros.}
 \end{aligned} \tag{3.3}$$

Con el fin de hallar los valores óptimos de los Q_i y los r_i del problema anterior, nos encontramos, al igual que el caso para un único producto, con la dificultad de que los óptimos locales pueden ser distintos de los óptimos globales, debido a que la función objetivo en general es no convexa, como se demostró en el teorema anterior.

La situación anterior, y el hecho de que las variables de decisión son enteras dificultan la aplicabilidad de la técnica que parecería ser la más indicada por su poco número de restricciones como lo son los multiplicadores de Lagrange, es por ésto, que al igual que para un único artículo, que se hace necesario simplificar el modelo al caso en el que $r_i - \theta_i \geq 0$, y el nivel de satisfacción será aproximada por el nivel de servicio tipo I, es decir, supondremos que en el modelo nunca hay pedidos pendientes. Con las suposiciones anteriores lo que se busca es poder optimizar la función utilizando multiplicadores Lagrangianos y así poder calcular de una manera simple los valores óptimos de Q_i y r_i .

Teniendo en cuenta las simplificaciones mencionadas, el problema de minimizar la inversión del inventario, restringido a un número máximo de pedidos por año y a un cierto nivel de satisfacción, está dado, como puede verse en [37]:

$$\begin{aligned}
& \text{mín} \quad \sum_{i=1}^N c_i \left(\frac{Q_i+1}{2} + r_i - \theta_i \right) \\
& \text{Sujeto a:} \quad \frac{1}{N} \sum_{i=1}^N \frac{D_i}{Q_i} \leq F \\
& \quad \frac{1}{D_{tot}} \sum_{i=1}^N D_i S(Q_i, r_i) \geq S \\
& \quad Q_i \geq 1 \\
& \quad r_i \geq 0 \\
& \quad Q_i, r_i \text{ enteros.}
\end{aligned} \tag{3.4}$$

3.2. Algunas metodologías de solución para el modelo (Q, r)

A continuación serán descritas las metodologías empleadas en el presente trabajo de grado para dar solución al modelo de inventarios en cuestión. Comenzamos con la metodología de Multiplicadores de Lagrange, dado que uno de los objetivos propuestos, es optimizar el modelo de inventarios tanto para el caso en el que hayan pedidos pendientes como para el caso en el que no los hayan, y en vista a que carecemos de una metodología de optimización exacta, en el que no sea necesario hacer suposiciones en la función objetivo, la solución obtenida utilizando multiplicadores de Lagrange, para el caso en el que no se consideran pedidos pendientes, será la forma de validar que la solución obtenida utilizando Algoritmos Genéticos, sea adecuada y así poder extender el problema al caso en el que se permitan pedidos pendientes.

3.2.1. Multiplicadores de Lagrange

Un problema de optimización no lineal restringida (NLP), puede ser descrito matemáticamente como sigue:

$$\begin{aligned}
& \text{mín} \quad z = z(x) \\
& \text{Sujeto a:} \\
& \quad h(x) = 0 \quad (h : \mathbb{R}^n \rightarrow \mathbb{R}^m) \\
& \quad g(x) \leq 0 \quad (g : \mathbb{R}^n \rightarrow \mathbb{R}^k)
\end{aligned} \tag{3.5}$$

Donde la nueva función objetivo está dada por la combinación de todas las restricciones, como la función lagrangiana, incluyendo las variables de decisión μ λ en los espacios respectivos:

$$L(x, \mu, \lambda) = z(x) + (g(x))^T \mu + (h(x))^T \lambda$$

las condiciones de optimalidad, son referidas como las condiciones Karush-Kuhn-Tucker (KKT) y fueron desarrolladas independientemente por Karush (1939) y luego por Kuhn Tucker (1951).

Aquí los valores μ y λ actúan como pesos para balancear las “fuerzas”. Las variables μ y λ también se conocen como variables duales, multiplicadores KKT o multiplicadores de Lagrange. Las condiciones necesarias KKT de primer orden para la optimalidad pueden ser descritas como sigue:

1. Dependencia lineal de los gradientes

$$\nabla xL(x^*, \mu^*, \lambda^*) = \nabla z(x^*) + \nabla(g(x^*))^\tau \mu^* + \nabla(h(x^*))^\tau \lambda^* = 0$$

donde x^* , μ^* y λ^* es solución óptima.

2. Factibilidad de la solución

$$g(x^*) \leq 0, \quad h(x^*) = 0$$

3. Condición de complementariedad

$$(\mu^*)^\tau g(x^*) = 0$$

4. No negatividad de los multiplicadores de la restricción de desigualdad

$$\mu^* \geq 0$$

Cualquier NLP diferenciable puede ser convertido a la forma anterior de tal manera que las condiciones de KKT pueden ser aplicables. La primera condición representa la dependencia lineal de los gradientes y es conocida también como el error Kuhn-Tucker. La segunda condición requiere que el punto satisfaga todas las restricciones. La tercera y la cuarta condición nos habla del hecho de que si la restricción es activa ($g_i = 0$), entonces el multiplicador correspondiente (μ_i) es positivo o si la restricción es inactiva ($g_i < 0$), entonces el multiplicador correspondiente (μ_i) es cero.

Estas condiciones nos reflejan comportamiento local en los puntos x^* , μ^* y λ^* .

3.2.2. Optimización-Simulación

La combinación entre optimización y simulación es conocida sólo desde la década anterior debido en gran parte a la implementación de los métodos heurísticos y metaheurísticos y al desarrollo computacional, el cual ha permitido incorporar software de optimización como Rumrunner, Opt-Quest y RiskOptimizer en paquetes de simulación como el Arena, Cristal Ball y el Simul8 entre otros, lo cual fue posible sólo a partir de 1990.

La Optimización-Simulación según [49] es el proceso de hallar el mejor valor de las variables de decisión de un sistema cuyo desempeño es evaluado con base en las salidas del modelo simulado. El objetivo principal de ésta área es optimizar sistemas complejos, es decir, sistemas en los cuales el tiempo computacional requerido para solucionarlo crece exponencialmente con el tamaño del problema, los cuales son muy frecuentes tanto en la industria como en la ciencia y cuya modelación matemática no es fácil de obtener porque se conoce poco sobre la estructura de la función, debido a múltiples condiciones de no linealidad, a un gran número de variables que además pueden ser de diversos tipos, o porque se deben hacer demasiadas suposiciones simplificadoras deteriorando en muchos casos la representación real del sistema.

Generalmente, el problema de Optimización-Simulación, se plantea matemáticamente, como sigue:

$$\min(\max) \quad f(x) = E(\psi(x, \xi(\omega)))$$

$$\text{s.a.} \quad x \in \Theta;$$

donde $x \in \Theta$ representa el vector de variables de entrada, $\Theta \subseteq \mathbb{R}^d$ es el conjunto factible, el cual puede ser definido explícita o implícitamente (en el caso que Θ sea todo el espacio de búsqueda, se dice que el problema es no restringido) y $f(x)$ es la función a optimizar, la cual debido a su complejidad y naturaleza estocástica no es posible definir a partir de una expresión analítica y por lo tanto debe ser estimada a través de las corridas del modelo simulado $\psi(x, \xi(\omega))$, el cual depende de los parámetros de entrada x , discretos o continuos, en un conjunto factible Θ . Por lo tanto $f(x) = E(\psi(x, \xi(\omega)))$ es el valor esperado de la medida del desempeño del sistema estimada por \hat{f} , la cual se obtiene del desempeño del modelo simulado y $\xi(\omega)$ es un vector aleatorio definido sobre el espacio de probabilidad (Ω, F, P) , ver [17], [21], [49], [57].

En sistemas complejos como los que a menudo se presentan en campos como manufactura, cadena de suministros, inventarios y optimización de portafolios se ha encontrado que métodos de optimización como *Relajación Lagrangiana*, *método Simplex*, *Branch and Bound* u otros métodos encuentran graves dificultades de implementación computacional, debido a que en muchos casos requieren de días, meses, años e incluso siglos para encontrar una solución óptima al problema. En consecuencia, aún cuando los métodos de Optimización-Simulación no proporcionan una solución óptima, sino que más bien sirven como herramienta para el análisis del funcionamiento de un sistema bajo condiciones especificadas por el usuario, la Optimización-Simulación a menudo suministra el único enfoque práctico para enfrentar muchos problemas del mundo real, al permitir diseñar un modelo del sistema real, que tiene elementos matemáticos en gran parte, y que además tiene la ventaja de poderse experimentar sobre este modelo, con el propósito de entender el funcionamiento del sistema, evaluar varias estrategias y tomar nuevas decisiones sin hacer ensayos sobre el sistema real, lo cual sería demasiado costoso y demorado.

Algunos problemas que se han tratado con la metodología de Optimización-Simulación son: Control de inventarios (El modelo (s, S) [21] y el vendedor de periódicos [53]), sistemas de manufacturas [19], cadena de suministro [19], [21], problemas de colas [21], call canter [19], [21], [53] y optimización de portafolio [19], entre otros.

En particular, el problema que será abordado en este trabajo, y sobre el cual será aplicada la metodología de Optimización-Simulación, es optimizar el modelo de inventarios (Q, r) , cuya política de inventario es minimizar la inversión de un inventario de N artículos, con aleatoriedad tanto en la demanda como en el tiempo de reposición, que es continuamente monitoreado, en el cual hayan pedidos pendientes en el momento en que llega una nueva orden y donde la inversión en el inventario está restringida a cierto nivel de satisfacción y cierto número de pedidos por período. Para simular la dinámica de dicho modelo utilizaremos como herramienta de simulación el Simul8, el cual tiene incorporado como paquete de optimización el OptQuest. Los resultados obtenidos en este proceso serán comparados con los resultados obtenidos al utilizar multiplicadores de Lagrange y los resultados del algoritmo de optimización propuesto, basado en

Algoritmos Genéticos.

Técnicas de Optimización-Simulación

Usualmente, son empleadas cuatro metodologías principales para resolver un problema de Optimización-Simulación, las tres primeras de ellas se utilizan cuando los parámetros del problema a ser simulado son continuos, es decir, cuando $\Theta \subseteq \mathbb{R}^d$ es no numerable, y pueden describirse brevemente como sigue:

1. *Métodos de superficie de respuesta (RSM)*: Metodología inicialmente desarrollada por George E. P. Box y que tiene sus raíces en estadística y en el diseño de experimentos, cuyo objetivo principal es explorar la relación entre las variables de entrada y de salidas de la función objetivo. Para lograr esto, usualmente recurre a la construcción de metamodelos, basándose en la aproximación mediante un modelo polinomial que es ajustado utilizando técnicas de regresión, ver [7], [63], [11], [66].
2. *Aproximación estocástica*: En esta metodología, dada una buena solución actual del conjunto de variables de entrada, se debe estimar el gradiente de la función objetivo con el fin de determinar la dirección de la búsqueda y hacer un movimiento en la dirección de la estimación del gradiente, la clave para que esta metodología sea exitosa depende en gran parte de la calidad del estimador del gradiente, ver [39], [6], [59], [45], [40], [65].
3. *Sample path optimization*: En esta metodología primero se ejecutan muchas simulaciones y luego se intenta perfeccionar la estimación resultante \hat{f} de la función objetivo f al realizar n repeticiones de la simulación, es decir, ésta metodología busca optimizar, para un n suficientemente grande, la función determinística \hat{f} , ver [32], [58], [54], [56].

Cuando el problema de Optimización-Simulación está definido a partir de parámetros discretos, es decir, cuando la región factible es un conjunto contable o finitamente contable son usualmente utilizadas técnicas heurísticas como la Búsqueda Aleatoria, y metaheurísticas como Recocido Simulado, Algoritmos Genéticos y Búsqueda Tabú, siendo los metaheurísticos mejoras de las técnicas heurísticas ya que muestran ser algoritmos más robustos y además proporcionan métodos eficientes y eficaces para escaparse de los óptimos locales en los cuales los heurísticos pueden quedar atrapados. Los métodos metaheurísticos a pesar de ser técnicas diseñadas generalmente para optimizar problemas de optimización combinatoria en un contexto determinístico y aunque pueden no garantizar convergencia a un óptimo global, han mostrado ser exitosos cuando son aplicados en Optimización-Simulación.

4. *Metaheurísticos*

Según [51], una definición para el concepto de metaheurísticos es la siguiente: "Los procedimientos metaheurísticos son una clase de métodos aproximados que están diseñados para resolver problemas difíciles de optimización combinatoria, en los que los heurísticos clásicos no son ni efectivos ni eficientes. Los metaheurísticos proporcionan un marco general para crear nuevos algoritmos híbridos combinando diferentes conceptos derivados de la inteligencia artificial, la evolución biológica y mecanismos estadísticos".

Los metaheurísticos combinan de forma diferente dos conceptos clave para el desarrollo de un buen algoritmo de búsqueda: la intensificación y la diversificación. La intensificación consiste en explorar a fondo una región del espacio de búsqueda donde se han encontrado buenas soluciones. Esta intensificación se lleva a cabo, por lo general, perturbando la mejor

solución encontrada hasta el momento para construir soluciones vecinas o cercanas en el espacio de búsqueda. La diversificación, por el contrario, consiste en probar en zonas inexploradas del espacio de búsqueda para evitar que el algoritmo quede atrapado en óptimos locales.

A continuación describiremos brevemente algunos de los metaheurísticos más nombrados en la literatura adaptados al contexto de Optimización-Simulación.

Recocido Simulado: Esta metodología fue introducida por Kirkpatrick, Gelatt y Vecchi en 1983 y se deriva de la analogía termodinámica con el proceso metalúrgico del recocido, el cual puede describirse como un proceso físico en el cual primero se ablanda el sólido, mediante su calentamiento a una temperatura elevada T , y luego lo va enfriando lentamente hasta que las moléculas alcancen su estado fundamental. A medida que disminuye la temperatura la probabilidad de movimiento de las moléculas se ajusta a la función de Boltzmann, $e^{\frac{-\delta}{T}}$, la cual depende de la temperatura. Al hablar de optimización, inicialmente se fija el parámetro T , se produce una perturbación, y el movimiento es aceptado inmediatamente si la solución actual es mejor que la anterior, o si su probabilidad es proporcional a un factor de aceptación. Para evitar que el algoritmo se quede atrapado en un óptimo local el Recocido Simulado permite, con una cierta probabilidad el paso a soluciones peores en los primeros pasos, cuando aún se está lejos del óptimo global, y posteriormente reduciendo la temperatura, se disminuye la posibilidad de cambios a soluciones peores cuando ya se encuentra más cerca al óptimo buscado, ver [3], [60], [23]. El algoritmo del recocido simulado consiste en:

a) *Inicialización:*

Seleccionar una solución inicial x^0 , simular su desempeño $\psi(x^0)$, inicializar $L_T = 0$, $T = T_0$ y fijar un α entre 0.5 y 0.99.

b) *Iteración:*

Seleccionar una solución candidata x^c en la vecindad $N(x^{L_T})$ de la solución actual y simular su desempeño $\psi(x^c)$. Si la solución candidata x^c satisface el criterio de aceptación basado en mejorar el desempeño de la simulación, o si la solución candidata satisface que su probabilidad asociada sea menor a $e^{\frac{-\delta}{T}}$, donde δ es la diferencia entre los desempeños de la solución candidata y la actual, entonces asignamos $x^{L_T+1} = x^c$, en otro caso, $x^{L_T+1} = x^{L_T}$. Actualizar la temperatura $T = \alpha T$.

c) *Finalización:*

Si se satisface el criterio de parada ($T < T_f$) entonces la búsqueda termina; de lo contrario se incrementa L_T a L_{T+1} y regresar al paso 2.

Un metaheurístico que básicamente usa el mismo criterio de aceptación que el del Recocido Simulado, son las **Redes Neuronales Artificiales**, las cuales pretenden simular los procesos que tiene lugar en redes neuronales naturales, tratando de lograr sistemas capaces de aprender de la experiencia y de realizar predicciones. Los modelos de Redes Neuronales más usados para resolver problemas de optimización son: Las máquinas de Boltzmann, las redes de Hopfield, los mapas auto-organizativos, de los cuales las máquinas de Boltzmann se puede considerar como un modelo de implementación paralela al algoritmo del recocido simulado cuya probabilidad está dada por:

$$A_k(u; T) = \frac{1}{1 + e^{\left(\frac{-\delta C_k(u)}{T}\right)}}$$

Donde $A_k(u; T)$ denota la probabilidad de aceptación de la k -ésima configuración, y T denota el parámetro de control. Esto difiere un poco de la probabilidad de aceptación del algoritmo de Recocido Simulado. Ambas probabilidades dan la misma distribución estacionaria y tienen las mismas propiedades de convergencia, ver [28], [42].

Búsqueda Tabú: El término Búsqueda Tabú fue introducido por Fred Glover en 1986 y sus principios fundamentales fueron elaborados en una serie de artículos de finales de los años 80 y principios de los 90, que fueron luego unificados en el libro “Tabu Search” en 1997. La búsqueda Tabú es una estrategia para resolver problemas de optimización combinatoria que se basa en encontrar mejores soluciones cercanas y aleatorias, a través del uso de una estructura de memoria para no quedar atrapado en óptimos locales en los que se puede anclar, al tratar de moverse de una solución a otra. Aquí describimos una forma básica de la Búsqueda Tabú que se caracteriza por incluir una memoria capacitada para recordar las soluciones visitada más recientemente mediante la utilización de una lista (lista Tabú) de tamaño L , impidiendo que éstas sean revisitadas, a menos que se cumplan ciertas condiciones (criterio de aspiración). Para actualizar la lista Tabú usualmente se emplea una estrategia FIFO, en la que en cada iteración los movimientos recientes van sustituyendo a los que se han realizado hace más de L iteraciones. [25], [26], [27]. El algoritmo de Búsqueda Tabú es básicamente:

a) *Inicialización:*

Seleccionar una solución inicial x^0 y simular su desempeño $\psi(x^0)$, inicializar la lista Tabú $H = \Phi$ y $k = 0$.

b) *Iteración:*

Seleccionar una solución candidata x^c en la vecindad $N(x^k)$ de la solución actual y simular su desempeño $\psi(x^c)$. Si la solución candidata x^c satisface el criterio de aceptación basado en el desempeño de la simulación, entonces asignamos $x^{k+1} = x^c$. Actualizar la lista Tabú, H , con las soluciones mas visitadas.

c) *Finalización:*

si se satisface el criterio de parada entonces termina la búsqueda; de lo contrario se incrementa k a $k + 1$ y regresar al paso 2.

Algoritmos Genéticos: El campo de los Algoritmos Genéticos el cual evolucionó principalmente en la década de 1980 y su estructura ha sido diseñada basada en los principios de la selección natural. Su funcionamiento básico puede describirse como sigue: Dos cromosomas de la población son seleccionados para ser padres de una nueva solución mediante cruzamiento para generar hijos los cuales posiblemente mutarán y reemplazarán a los padres menos adaptados. De manera corta esta metodología consiste en:

a) *Inicialización:*

Generar una población inicial de cromosomas (cadenas de bits) creada (usualmente) de manera aleatoria, simular el desempeño y asignar un valor de adaptación a cada individuo relacionado con su valor de desempeño.

b) *Iteración:*

Seleccionar dos individuos de la generación actual de acuerdo a los valores de adaptación para cruzar su material genético con otro individuo igualmente seleccionado. Mutar los descendientes con cierta probabilidad y asignarle su valor de adaptación. Escoger los individuos mejor adaptados entre padres e hijos para formar la nueva generación.

c) *Finalización:*

Si se satisface el criterio de parada entonces la búsqueda termina; de lo contrario regresar al paso 2.

Como se puede notar, los Algoritmos Genéticos se distingue de todos los metaheurísticos anteriores, básicamente por el hecho de que en cada iteración se opera sobre un conjunto de soluciones y no sobre un único punto solución, como los demás métodos, es decir las soluciones sucesoras se obtienen a partir de parejas constituidas con los elementos de la población y no mediante la transformación de la solución actuales.

Una metodología análoga a los Algoritmos Genéticos, en cuanto a que ambos son heurísticos poblacionales que construyen soluciones mediante la combinación de otras soluciones ya existentes, es la **Búsqueda Dispersa**, la cual es una metodología de búsqueda poblacional que opera sobre un conjunto de puntos (que garantizan cierto nivel de diversidad), llamados puntos de referencia, los cuales son combinados para obtener nuevas soluciones; las combinaciones son formas generalizadas de combinaciones lineales, acompañadas de procesos adaptativos para garantizar condiciones de factibilidad. Cada vez que se genera una nueva solución combinada, si ésta es mejor que la peor del conjunto de referencia, se reemplaza y se vuelven a generar los subconjuntos necesarios. Dicha metodología incorpora la Búsqueda Tabú, al hacer uso de su memoria adaptativa la cual proporciona la habilidad de aprovecharse de la historia de la búsqueda para guiar el proceso de la solución.

La diferencia entre estas dos metodologías radica en que en sus implementaciones iniciales los Algoritmos Genéticos escogen un par de padres al azar para producir nuevos descendientes y luego introducir cierta aleatoriedad para determinar como combinar las características de los padres para luego generar una nueva generación. En contraste, la Búsqueda Dispersa no recurre a la aleatoriedad, en el sentido de ser indiferente a cualquier alternativa, ya que el enfoque de ésta búsqueda es generar nuevas soluciones, sin perder la habilidad de producir soluciones diversas, al incluir la generación de nuevos puntos que pueden ser combinaciones convexas o no de los puntos originales, las cuales pueden ingresar inmediatamente al conjunto de referencia durante este proceso, los nuevos puntos pueden contener información que no contienen los puntos de referencia originales. Algunos de los rasgos de Búsqueda Dispersa han sido incorporados en Algoritmos Genéticos después de un período intermedio de aproximadamente una década, ver [28], [42],[29].

En el siguiente capítulo se hará un estudio más profundo sobre Algoritmos Genéticos, dado que ésta será la metodología de optimización utilizada en este trabajo.

Software

Como se mencionó anteriormente la implementación de software de optimización en paquetes de simulación ha venido generando un gran avance en el área de Optimización-Simulación. En la figura 3.1, son recopilados algunos de los software de optimización más utilizados comercialmente.

Table 1: Some Commercial Software Packages

Optimization Package (simulation platform)	Vendor (URL)	Primary Search Strategies
AutoStat (AutoMod)	AutoSimulations, Inc. (www.autosim.com)	evolutionary, genetic algorithms
OptQuest (Arena, Crystal Ball, et al.)	Optimization Technologies, Inc. (www.opttek.com)	scatter search and tabu search, neural networks
OPTIMIZ (SIMUL8)	Visual Thinking International Ltd. (www.simul8.com)	neural networks
SimRunner (ProModel)	PROMODEL Corp. (www.promodel.com)	evolutionary, genetic algorithms
Optimizer (WITNESS)	Lanner Group, Inc. (www.lanner.com/corporate)	simulated annealing, tabu search

Figura 3.1: tomada de [19]

En este trabajo se utilizó el OptQuest para Simul8 ya que nos servirá como soporte para realizar algunas comparaciones, por esta razón dedicamos algunos párrafos al mismo.

De acuerdo con [28] OptQuest es un software comercial diseñado para optimizar sistemas complejos, los cuales son formulados como modelos de la simulación, es un software de optimización autosuficiente que puede interactuar con varios paquetes de simulación comerciales, tales como Arena, Crystal Ball y simul8; el software OptQuest fué desarrollado por Fred Glover, James P. Kelly, y Manuel Laguna de la Universidad de Colorado. El algoritmo implícito en este software incorpora una combinación de estrategias basadas en Búsqueda Dispersa, Algoritmos Genéticos, Búsqueda Tabú y Redes Neuronales. OptQuest está enlazado con simul8 de manera que automáticamente busca la mejor solución posible dado un objetivo particular de entre un conjunto de variables de entradas las cuales cambiarán de corrida a corrida y sobre los cuales la optimización se dirigirá, éstas deben especificarse en la hoja de resumen de resultados del Simul8. También se pueden dar restricciones las cuales permitirán acotar el espacio de búsqueda y determinar que tanto puede cambiar la solución, ver [42], [28].

OptQuest Callable Library (OCL): Es el procesador de optimización del sistema de OptQuest y está diseñado para buscar soluciones óptimas para el tipo de problemas de la forma:

$$\begin{aligned}
 & \min(\max) \quad f(x) \\
 \text{s.a} \quad & Ax \leq b \quad \text{restricciones} \\
 & g_l \leq G(x) \leq g_u \quad \text{requerimientos} \\
 & l \leq x \leq u \quad \text{cotas}
 \end{aligned}$$

donde x es un vector de variables acotadas, las cuales pueden ser parámetros discretos o continuos. La función objetivo $f(x)$ es una función real valuada. El conjunto de restricciones debe ser lineal y los coeficientes de la matriz A como el valor de b son conocidos. Los requerimientos son simplemente cotas las cuales a diferencia de las restricciones pueden ser lineales o no lineales, pero constantes y conocidas, impuestas sobre alguna función que puede ser lineal o no lineal y que permite incorporar algún requisito impuesto en la simulación.

A continuación se describe brevemente el funcionamiento del OptQuest:

Asumiendo que una solución al problema de optimización puede representarse por un vector $x = (x_1, x_2, \dots, x_n)$, donde cada x_i , $i = 1, 2, \dots, n$ es una variable real o entera, las cuales son utilizadas como variables de entrada; y que el valor de la función objetivo $f(x)$ puede ser obtenido al correr el modelo de simulación relacionado. El método de Búsqueda dispersa implementado en OCL comienza generando una población inicial de puntos de referencia la cual incluye las cotas superiores e inferiores de las variables, puntos sugeridos por el usuario y punto medio $z = l + (u - l)/2$, según [41] el tamaño de la población por defecto es 100. Adicionalmente, se crean otros puntos con el fin de crear una población diversa, es decir, que haya una diferencia significativa entre un elemento y otro. Cuando el modelo incluye restricciones lineales, los puntos recientemente creados están sujetos a un test de factibilidad antes de ser evaluados, es decir, antes de evaluar la función objetivo y los requerimientos. El test de factibilidad consiste en verificar (una por una) si las restricciones impuestas por el usuario se satisfacen. El tamaño de la población es automáticamente ajustado por el sistema considerando el tiempo que es requerido para completar una evaluación de $f(x)$ y por el tiempo límite que el usuario permita para que el sistema realice la búsqueda. A mayor número de corridas mayor será la probabilidad de acercarse a las soluciones óptimas.

Una vez la población se genera, comienza el proceso iterativo en busca de mejores soluciones, de la siguiente manera: en cada iteración se seleccionan dos puntos de referencia para crear tres descendientes. Suponiendo que los padres de referencia son x_1 y x_2 , entonces los hijos x_3 , x_4 y x_5 , son hallados como sigue:

$$x_3 = x_1 + d$$

$$x_4 = x_1 - d$$

$$x_5 = x_2 - d$$

Donde $d = r(x_2 - x_1)/2$. La selección de x_1 y x_2 está basada en los valores de $f(x_1)$ y $f(x_2)$ y r es un número aleatorio entre 0 y 1.

Al finalizar una iteración los peores padres son reemplazados por los mejores hijos, dándole un estado tabú-activo al padre superviviente para el número dado de iteraciones. En las iteraciones subsiguientes, la presencia de padres Tabú activos es prohibido. Finalmente, el OptQuest implementa la metodología de Redes Neuronales la cual es utilizada como un modelo de predicción que ayuda a que el sistema acelere la búsqueda evitando que la simulación corra resultados que pueden predecirse como inferiores, ver [28], [42].

En la siguiente sección, el propósito es describir de manera general los conceptos básicos y el funcionamiento del metaheurístico Algoritmos Genéticos, el cual representa la metodología de optimización propuesta en el presente trabajo.

3.2.3. Algoritmos Genéticos

A pesar de que en el capítulo anterior, entre las técnicas utilizadas para resolver problemas de Optimización-Simulación se dieron algunos rasgos del funcionamiento de los Algoritmos Genéticos, en el presente capítulo se hará un estudio más profundo de dicho metaheurístico, dado que

ésta será la metodología de optimización utilizada en la presente tesis.

Se comienza éste capítulo con una breve introducción, sobre el origen y la analogía entre los Algoritmos Genéticos y la evolución natural, para luego continuar con un listado de ventajas y desventajas, con las ventajas lo que se pretende es mostrar las razones de por qué entre todos los metaheurísticos los Algoritmos Genéticos es el más utilizado, siendo muchas de éstas razones las motivadoras para el desarrollo de nuestro trabajo. El resto del capítulo consiste en describir los conceptos básicos necesarios para la implementación de dicho metaheurístico, los tipos de algoritmos disponibles, las desventajas de implementar dicha metodología, así como también las razones del por qué funciona dicho algoritmo.

Introducción

Los principios básicos de los Algoritmos Genéticos fueron establecidos originalmente por John Holland [34] y por sus compañeros del grupo de investigación Logic of Computers, de la Universidad de Michigan, durante las décadas de 1960 y 1970 y que posteriormente fueron descritos en el libro *Adaptación en sistemas naturales y artificiales*, escrito por Holland en 1975, el cual presenta los Algoritmos Genéticos, como una abstracción de la teoría de la Selección Natural desarrollada por Charles Darwin y publicada en 1859 en el libro *El Origen de las Especies*. Más adelante en la década de 1980 las ideas de John Holland fueron formalizadas y sistematizadas por uno de sus estudiantes David Goldberg, quien definió los Algoritmos Genéticos como técnicas de búsqueda y optimización basadas en la mecánica de la evolución biológica, bajo dicha mecánica las poblaciones evolucionan en la naturaleza con los principios de la selección natural y la supervivencia de los más adaptados.

Los Algoritmos Genéticos emulan la evolución natural para explorar con eficiencia el espacio de búsqueda, con el supuesto de que individuos con ciertas características son aptos para sobrevivir y transmitir estas características a su descendencia [60]. Básicamente el algoritmo comienza, generando un conjunto de soluciones (población) al problema, usualmente de manera aleatoria. Se supone que los individuos (posibles soluciones del problema), pueden representarse como un conjunto de parámetros (que se denominan genes), cada uno de los cuales representa una solución factible a un problema específico. Cada una de estas soluciones (individuos) están codificadas en cadenas de caracteres (originalmente en números binarios). Haciendo una relación con la evolución biológica, dichas cadenas representan un cromosoma y cada elemento de la cadena, un gen. A cada individuo se le asigna un valor de adaptación, relacionado con la calidad de dicha solución. En la naturaleza esto equivaldría al grado de adaptación de un organismo para competir por unos determinados recursos como: agua, comida y techo. Mientras mayor sea la adaptación del individuo, mayor será la probabilidad de ser seleccionado como padre y cruzar sus genes con otro padre igualmente seleccionado, para crear nuevos hijos con características similares a las de sus padres y que harán parte de la nueva población. La transformación de los individuos de cada generación se lleva a cabo mediante la definición de una serie de operadores genéticos, de los cuales, los de selección, cruce, y mutación son los más frecuentes. Mediante la aplicación de estos operadores sobre los miembros de la población, es posible obtener individuos que hereden una mayor cantidad de buenas características de sus predecesores.

Algunas Aplicaciones de los Algoritmos Genéticos como se puede observar en [22] son muy frecuentes en: Optimización, Programación automática, Aprendizaje máquina, Economía, Sistemas inmunes, Ecología, Genética de poblaciones, Sistemas sociales, Evolución y aprendizaje, etc.

Antes de comenzar con el desarrollo teórico que envuelve esta poderosa metodología de optimización, daremos un listado bastante amplio, de por qué según [52] dentro de los metaheurísticos, los Algoritmos Genéticos es el más empleado.

Ventajas y Desventajas

Ventajas

- Algunos métodos de optimización tales como Recocido Simulado y búsqueda Tabú se mueven a través de un único punto en el espacio de búsqueda, usando algún movimiento para el siguiente vecino, este tipo de movimiento de punto a punto a menudo puede quedar encapsulado en óptimos locales. En contraste, los Algoritmos Genéticos trabajan simultáneamente con una población de soluciones factibles, reduciendo la probabilidad de quedar atrapado en óptimos locales.
- Son los algoritmos que hacen una barrida mayor al subespacio de posibles soluciones válidas. De hecho, se considera que, de todos los algoritmos de optimización estocásticos, los algoritmos genéticos son de los más exploratorios disponibles.
- Trabajan con la codificación (en forma de cromosoma) de las variables de decisión y no directamente con las variables de decisión
- Algunos métodos tales como la relajación lagrangiana necesitan que la función objetivo y/o sus restricciones satisfagan condiciones tales como continuidad, diferenciabilidad, convexidad u otras sobre el espacio de búsqueda. A diferencia, los Algoritmos Genéticos sólo requieren la función objetivo.
- Dentro de las técnicas de computación evolutiva, los Algoritmos Genéticos son los que poseen una mayor base teórica.
- Son flexibles y adaptables a una gran cantidad de problemas diferentes, pertenecientes a distintas áreas y permiten ser combinados con otras técnicas (hibridación).
- Cuando se usan para problemas de optimización, resultan menos afectados por los óptimos locales que las técnicas tradicionales (es decir, son métodos robustos).
- Poseen una gran versatilidad pues son los que necesitan menos conocimiento específico del problema para su resolución.
- Es posible implementarlos en computadores con capacidad media obteniendo resultados muy aceptables.
- Son algoritmos de búsqueda múltiple, capaces de manejar varias variables de decisión simultáneamente.
- A diferencia de otros algoritmos, cuya convergencia y resultado final son fuertemente dependientes de la solución inicial, en los algoritmos genéticos (salvo poblaciones iniciales

realmente degeneradas, en los que el operador de mutación va a tener mucho trabajo) la convergencia es poco sensible a la población inicial si esta se escoge de forma aleatoria y es lo suficientemente grande.

- Es una búsqueda paraméricamente robusta. Eso quiere decir que hemos de escoger realmente mal los parámetros del algoritmo para que no converja. Con tasas razonables, va a converger (mejor o peor) en una solución razonablemente buena si la representación es la adecuada.
- Por último, los algoritmos genéticos son intrínsecamente paralelos. Esto significa que, independientemente de que lo hayamos implementado de forma paralela o no, buscan en distintos puntos del espacio de soluciones de forma paralela.

Desventajas

- Pueden tardar mucho tiempo en converger, o pueden no converger, (esta última situación se describe usualmente como la tendencia del Algoritmos Genético a engañar, aunque en realidad lo que está pasando es que el Algoritmo Genético está haciendo lo que se le pidió hacer, no lo que sus creadores pretendían que hiciera), esto se debe en gran medida a los valores de los parámetros que se utilicen, como por ejemplo: al tamaño de la población, al número de generaciones, a las probabilidades de cruce y mutación, al criterio de parada, etc.
- Pueden converger prematuramente, provocando que el algoritmo converja hacia un óptimo local.

Conceptos Básicos

Algunos elementos indispensables para plantear un Algoritmo Genético son los siguientes:

Cromosoma: Cadena de caracteres que codifica un individuo de la población. Son los encargados de transmitir la información genética de padres a hijos.

Gen: Cada uno de los caracteres que forman un cromosoma. El gen es fundamentalmente una unidad de herencia.

Gametos: Son las células que llevan información genética de los padres con el propósito de efectuar la reproducción sexual (esperma y óvulos en el ser humano).

Alelo: A cada mínimo elemento que conforma el cromosoma, en Algoritmos Genéticos, se lo denomina Alelo. En representación binaria, esta representación sería el bit.

Genotipo y Fenotipo: El genotipo es la codificación (por ejemplo, binaria) de los parámetros que representan una solución del problema a resolverse; el fenotipo es la decodificación del cromosoma, es decir los valores obtenidos de la representación (binaria). Por ejemplo si se utiliza una representación en base 2 se tendrá genotipo 1010 y su fenotipo será 10.

Explotación: Es el proceso de utilizar la información obtenida de puntos del espacio de búsqueda previamente visitados para determinar los puntos que conviene visitar a continuación. La explotación involucra movimientos finos y es un mecanismo provechoso para que un algoritmo encuentre óptimos locales.

Exploración: Proceso de visitar nuevas regiones del espacio de búsqueda para tratar de encontrar soluciones prometedoras. La exploración involucra grandes saltos en el espacio de búsqueda y es un mecanismo útil para evitar que un algoritmo quede atrapado en óptimos locales.

Valor de Adaptación: El valor de adaptación de un individuo evalúa su capacidad de adaptación a las condiciones de su entorno. Se relaciona con la probabilidad de que el individuo sobreviva para reproducirse, y tiene dependencia directa con su número de descendientes.

Función de Adaptación: La función de Adaptación corresponde a una forma de medir qué tan bien un determinado individuo se adapta al medio, o bien cuál es el valor de dicho individuo. Se debe tener en consideración dos aspectos: la calidad de la función objetivo y la no factibilidad del individuo. La no factibilidad del individuo debe ser penalizada en la función de ajuste proporcionalmente al grado de no factibilidad del mismo. La influencia de la función de adaptación es fundamental para determinar los individuos candidatos a sobrevivir (aquellos individuos a los que se aplicarán los operadores evolutivos).

Para definir la función de Adaptación se deben tener en cuenta los siguientes aspectos:

- El tipo de problema y el criterio de optimización.
- Que puede cambiar con el tiempo a medida que el Algoritmo Genético procede.
- Debe considerar las restricciones del problema.
- Puede definir objetivos múltiples (función vectorial) o incorporar sub-objetivos (para problemas multi-objetivo).

Una de las principales dificultades que se puede presentar a la hora de definir la función de Adaptación, es la generación de individuos no factibles durante la evolución. Para resolver dicha dificultad existen cuatro mecanismos para tratar los individuos no factibles:

1. Evitarlos en la codificación: En general no es un procedimiento sencillo, complica los procesos de codificación y decodificación.
2. Descartarlos: Aquellos individuos que no verifican las restricciones, no son considerados como tales, y se siguen efectuando cruces y mutaciones hasta obtener individuos válidos, o bien, a dichos individuos se les asigna una función objetivo igual a cero. Es la opción más simple, pero conduce a la pérdida de características que podrían ser útiles para resolver el problema.
3. Reconstruirlos: Reconstruir aquellos individuos que no verifican las restricciones. Dicha reconstrucción suele llevarse a cabo por medio de un nuevo operador que se acostumbra a denominar reparador.

4. Penalizarlos: La idea general consiste en alterar la función objetivo del individuo por una cantidad (la penalización) que guarda relación con las restricciones que dicho individuo viola. Para problemas complejos puede ser difícil, al requerir estudios teóricos y empíricos para definir un modelo de penalización adecuado.

Población

La población es el conjunto de individuos que forman cada generación, en adelante se denotará por P la población, P_0 la población inicial y M el tamaño de la población, el cual puede permanecer constante en cada población o variar de población a población. Es importante resaltar que tanto la calidad, como la efectividad del proceso de solución final en los Algoritmos Genéticos, depende en gran parte del tamaño de la población y de cómo la población inicial fue construida; la población inicial de individuos puede ser generada aleatoriamente o utilizando alguna heurística de construcción, sin embargo, al utilizar un heurístico en ocasiones se han encontrado situaciones como que el algoritmo se queda atrapado en un óptimo local o que existe el riesgo de considerar un subconjunto del espacio de soluciones [24], sin embargo, también se podría tener soluciones en un tiempo menor.

Tamaño de la población

Aunque el tamaño adecuado de la población es un valor que aún no se ha logrado determinar con certeza, algunos autores como De Jong y Grefenstette J. han sugerido valores como 100 individuos y 30 individuos, respectivamente, y otros autores como Alander J. sugiere que un tamaño de población comprendido entre l y $2l$, donde l es la longitud del cromosoma. Se ha encontrado que poblaciones pequeñas corren el riesgo de no cubrir adecuadamente el espacio de búsqueda y por lo tanto converger prematuramente, mientras que trabajar con poblaciones de gran tamaño puede conducir a excesivos costos computacionales. Usualmente el tamaño de la población permanece constante en todas las generaciones, sin embargo, una generalización sería el considerar que el tamaño depende de la generación, es decir $\lambda_t = |P_t|$, donde P_λ denota el conjunto de poblaciones de tamaño λ . Diversas formas de generar una población inicial pueden encontrarse en [31].

Reemplazo de la Población

Otro aspecto importante en la ejecución de los Algoritmos Genéticos es la formación de la siguiente generación. En esta construcción se deben tener en cuenta aspectos importantes, como por ejemplo: si se deben eliminar todos los padres y formar la nueva población sólo con hijos, o si se debe dejar un porcentaje de padres y el porcentaje restantes de hijos, o si lo mejor es escoger de entre padres e hijos los mejor adaptados. Esta operación se puede hacer de diversas formas, pero en general existen cuatro métodos fundamentales para realizar el reemplazo:

- Cuando el número de individuos llega a un cierto número, se elimina un subconjunto de la población conteniendo a los individuos menos adaptados.
- Cada vez que se crea un nuevo individuo, en la población se elimina el menos adaptado para dejar su lugar al nuevo individuo.

- Cada vez que se crea un nuevo individuo, en la población se elimina aleatoriamente una solución, independientemente de su adaptación.
- Formar la nueva población por los individuos mejor adaptados, de entre padres e hijos.

El procedimiento de reemplazo se puede representar a través de las funciones de extensión y reducción como sigue:

La función de extensión, f_{ext} , crea a partir de dos poblaciones $x, z \in P_\lambda$, una población $n \in P_{2\lambda}$:

$$f_{ext} : (x, z) \rightarrow n$$

Denotando por N_i con $i = 1, \dots, 2\lambda$ el i -ésimo individuo en n , por X_k , con $k = 1, \dots, \lambda$ el k -ésimo individuo en x , y por Z_j con $j = 1, \dots, \lambda$ el j -ésimo individuo en z , se tendría:

$$N_i = \begin{cases} X_i & \text{si } i \leq \lambda \\ Z_{i-\lambda} & \text{si } i > \lambda \end{cases}$$

La función de reducción global, f_{red} , convierte una población $n \in P_{2\lambda}$ en una población $r \in P_\lambda$

$$f_{red} : n \rightarrow r$$

Donde r denota la población de individuos en el tiempo $t + 1$.

La función de reducción es elitista de grado λ si la población en el tiempo $t + 1$ está constituida por los mejores λ individuos, de entre los λ individuos que constituyen la población en el tiempo t y los descendientes derivados de ellos.

La función de reducción se denomina simple si la población en el tiempo $t + 1$ está formada por los descendientes derivados de la población en el tiempo t .

Pruebas de hipótesis

Para efectos de ajuste de parámetros del Algoritmo Genético, se utilizan pruebas de hipótesis para promedios y varianzas, tomando como base lo expuesto en [4], por lo tanto introduciremos algunos preliminares teóricos sobre este tema.

El término, *prueba de hipótesis* se referirá al proceso de intentar decidir la verdad o falsedad de dichas hipótesis sobre la base de evidencias experimentales.

Definición 3.2.1. *La región crítica para una prueba de hipótesis es el subconjunto del espacio muestral que corresponde al rechazo de la hipótesis nula*

Se manejan dos tipos de errores en el rechazo aceptación de la hipótesis nula, ellos son:

- Error tipo I: Rechazar un H_0 verdadero
- Error tipo II: Falla rechazar un H_0 falso (Aceptar un H_0 falso)

Se adopta la siguiente notación para las probabilidades de estos errores:

- $p[\text{Error tipo I}] = p[\text{TI}] = \alpha$
- $p[\text{Error tipo II}] = p[\text{TII}] = \beta$

Definición 3.2.2. Para una hipótesis nula simple H_0 , la probabilidad de rechazar a H_0 verdadera, $\alpha = p[\text{TI}]$, se refiere al Nivel de significancia de la prueba. Para una hipótesis nula compuesta H_0 , el tamaño de la prueba o (Tamaño de la región crítica) es la probabilidad máxima de rechazar H_0 cuando H_0 es verdadera (maximizando sobre los valores bajo el parámetro H_0)

Nótese que para un H_0 simple el nivel de significancia es también el tamaño de prueba

Definición 3.2.3. La función potencia $\pi(\theta)$, de una prueba de H_0 es la probabilidad de rechazar H_0 cuando el valor verdadero del parámetro es θ

Pruebas para dos muestras

Es posible construir pruebas de hipótesis para las varianzas de dos distribuciones normales, tal como $H_0 : \sigma_2^2/\sigma_1^2 = d_0$, basada en el estadístico F , en particular considerar la prueba estadística

$$F_0 = \frac{S_1^2}{S_2^2} d_0$$

donde $F_0 \sim F(n_1 - 1, n_2 - 1)$

El siguiente teorema presenta la prueba de hipótesis para comparar las varianzas de dos muestras las cuales deben provenir de distribuciones normales

Teorema 3.2.1. Supóngase que x_1, x_2, \dots, x_{n_1} y y_1, y_2, \dots, y_{n_2} son valores observados de muestras aleatorias independientes de $N(\mu_1, \sigma_1^2)$ y $N(\mu_2, \sigma_2^2)$, respectivamente, y sea

$$f_0 = \frac{S_1^2}{S_2^2} d_0$$

1. Un tamaño α de la prueba $H_0 : \sigma_2^2/\sigma_1^2 \leq d_0$ versus $H_a : \sigma_2^2/\sigma_1^2 > d_0$ es rechazar H_0 si $f_0 \leq 1/f_{1-\alpha}(n_2 - 1, n_1 - 1)$
2. Un tamaño α de la prueba $H_0 : \sigma_2^2/\sigma_1^2 \geq d_0$ versus $H_a : \sigma_2^2/\sigma_1^2 < d_0$ es rechazar H_0 si $f_0 \geq f_{1-\alpha}(n_1 - 1, n_2 - 1)$
3. Un tamaño α de la prueba $H_0 : \sigma_2^2/\sigma_1^2 = d_0$ versus $H_a : \sigma_2^2/\sigma_1^2 \neq d_0$ es rechazar H_0 si $f_0 \leq f_{1-\alpha/2}(n_2 - 1, n_1 - 1)$ o $f_0 \geq f_{1-\alpha/2}(n_1 - 1, n_2 - 1)$

Si las varianzas son desconocidas pero iguales, entonces pruebas de hipótesis para las medias tales como $H_0 : \mu_2 - \mu_1 = d_0$ pueden ser construidas basadas en la distribución t . En particular, sea

$$t_0 = \frac{\bar{x} - \bar{y} - d_0}{s_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

donde

$$S_p^2 = \frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2}$$

El siguiente Teorema presenta la prueba de hipótesis para comparar las medias de dos muestras las cuales deben provenir de distribuciones normales con varianzas iguales.

Teorema 3.2.2. *Supóngase que x_1, x_2, \dots, x_{n_1} y y_1, y_2, \dots, y_{n_2} son valores observados de muestras aleatorias independientes de $N(\mu_1, \sigma_1^2)$ y $N(\mu_2, \sigma_2^2)$, respectivamente, donde $\sigma_1^2 = \sigma_2^2 = \sigma^2$*

1. *Un tamaño α de la prueba $H_0 : \mu_2/\mu_1 \leq d_0$ versus $H_a : \mu_2/\mu_1 > d_0$ es rechazar H_0 si $f_0 \leq 1/f_{1-\alpha}(n_2 - 1, n_1 - 1)$*
2. *Un tamaño α de la prueba $H_0 : \sigma_2^2/\sigma_1^2 \geq d_0$ versus $H_a : \sigma_2^2/\sigma_1^2 < d_0$ es rechazar H_0 si $f_0 \geq f_{1-\alpha}(n_1 - 1, n_2 - 1)$*
3. *Un tamaño α de la prueba $H_0 : \sigma_2^2/\sigma_1^2 = d_0$ versus $H_a : \sigma_2^2/\sigma_1^2 \neq d_0$ es rechazar H_0 si $f_0 \leq f_{1-\alpha/2}(n_2 - 1, n_1 - 1)$ o $f_0 \geq f_{1-\alpha/2}(n_1 - 1, n_2 - 1)$*

Operadores Genéticos

Los Operadores Genéticos son los encargados de alterar la composición de los individuos de la población a través de las generaciones, buscando con esto obtener individuos lo suficientemente adaptados como para ser soluciones óptimas al problema en cuestión. Los operadores genéticos más comúnmente utilizados son: el operador de selección, el operador cruce, y el operador mutación. A continuación se dará una descripción corta de cada uno de ellos.

Operador de Selección

El operador de selección es el encargado de preseleccionar una subpoblación de la población actual con el fin de transmitir y conservar a través de las generaciones, las mejores características de las soluciones.

Durante la primera fase del algoritmo, se seleccionan n individuos de la población inicial P_0 , los cuales serán los candidatos a ser padres de la siguiente generación, posteriormente se cruza el material genético de los padres seleccionados, produciendo con esto, algunos descendientes los cuales una vez mutados constituirán parcial o totalmente la siguiente población de individuos. La selección de padres se efectúa usualmente al azar, usando un procedimiento que favorezca a los individuos mejor adaptados. Algunas de las técnicas disponibles para efectuar dicha selección son las siguientes:

- Selección directa: De entre los M individuos de la población actual, se escogen los n individuos más adaptados.
- Selección aleatoria: De entre los M individuos iniciales se eligen aleatoriamente n individuos.
- Selección estocástica: A cada individuo de la población inicial se le asigna una probabilidad p de ser seleccionado en función de su adaptación y respecto a la adaptación de toda la población. Las técnicas de selección estocásticas más utilizadas son: el sorteo, muestreo por restos, por torneos, y la más utilizada de ellas el muestreo por ruleta sesgada, ver [31] en la cual a cada individuo se le asigna una probabilidad, la cual es representada por un sector de la ruleta.

Uno de los procedimientos más utilizados para calcular dichas probabilidades, en el caso que el objetivo sea maximizar la función, es la *función de selección proporcional a la función objetivo*, en la cual cada individuo tiene una probabilidad de ser seleccionado como padre directamente proporcional al valor de su función objetivo, la cual está dada por

$$P(x_i) = \frac{g(x_i)}{\sum_{i=1}^M g(x_i)}$$

Donde x_i denota el individuo i -ésimo de la población P , $g(x_i)$ su valor de adaptación y $P(x_i)$ su probabilidad de ser seleccionado.

Después de calcular la probabilidad de cada individuo, dichas probabilidades se ordenan de mayor a menor con el fin de calcular sus probabilidades acumuladas, formando con estos valores, una serie de intervalos cuyo tamaño es proporcional a la adaptación de cada individuo. Para seleccionar un individuo se genera un número aleatorio entre 0 y 1 y se selecciona al individuo cuyo intervalo contiene dicho número. En ésta forma de selección, los individuos mejor adaptados tienen una mayor probabilidad de ser elegidos como padres, sin embargo dado que es un número aleatorio el que decide que padre será seleccionado, todos los individuos tienen alguna probabilidad de ser seleccionados, incluso los menos adaptados.

Ésta forma de realizar la selección tiene la desventaja de converger prematuramente cuando un individuo tiene una probabilidad muy grande en comparación a los demás individuos, por lo cual, en otros casos la selección de padres se efectúa a través de la *selección proporcional al rango* del individuo, con la cual se produce una selección en la cual todos los individuos tienen una probabilidad más uniforme, en cuanto a que la diferencia a que un individuo sea seleccionado es siempre constante e igual para todos. En esta metodología los elementos se ordenan de acuerdo al valor de la función objetivo g , del más adaptado al menos adaptado.

Denotemos por $ran(g(x_i))$ el rango de la función del individuo x_i , así, el mejor individuo tiene rango M y el peor individuo tiene rango 1, y por $p(x_i)$ la probabilidad de que el individuo (x_i) sea seleccionado, la cual está definida por:

$$p(x_i) = \frac{ran(g(i))}{M(M+1)/2}$$

Otra de las técnicas utilizadas para la selección es la denominada *criterio elitista*, en la cual se selecciona el mejor individuo de la población actual (solución élite), el cual obligatoriamente hará parte de la nueva población. Según [15] el empleo de estrategias elitistas permite una exploración más profunda del espacio de soluciones en el entorno de la elite, pudiendo compararse con una intensificación local de la búsqueda, es decir, tienen tendencia a hacer la búsqueda más explotadora. Si bien es cierto que su empleo puede mejorar la convergencia del algoritmo, ya que evita perder la mejor solución encontrada. También es cierto que cuando se presentan funciones fuertemente multimodales (con muchos óptimos locales) su eficiencia disminuye aumentando el riesgo de convergencia prematura, es decir, es posible que este método conduzca rápidamente a un óptimo local.

La función de selección, denotada por f_{sel} , selecciona al azar y con reemplazamiento una colección de n individuos $y \in P_\lambda$ a partir de una población $x \in P_\lambda$, donde P_λ : es el conjunto de poblaciones

de tamaño λ . El operador de selección puede representarse en forma funcional de la siguiente manera:

$$f_{sel} : (\alpha, x) \rightarrow y$$

donde α es un vector de dimensión λ constituido por valores escogidos aleatoriamente.

Operador de Cruce

El operador de cruce es el encargado de intensificar la búsqueda al rededor de las mejores soluciones obtenidas hasta cierta iteración t y combinar dichas soluciones para crear nuevos individuos. El cruce entre dos individuos consiste en escoger dos miembros de la población seleccionada, fijar una probabilidad de cruce, la cual comúnmente está en el intervalo $[0.6, 0.95]$ y si el individuo tiene una probabilidad de cruce más alta que la fijada, entonces el material genético de dicho individuo es intercambiado con otro individuo igualmente seleccionado. Habitualmente el operador de cruce se encarga de intensificar la búsqueda explorando las áreas más prometedoras del espacio de búsqueda alrededor de las mejores soluciones visitadas hasta el momento. Según John Holland, el cruce es el principal operador genético hasta el punto que se puede decir que, no es un algoritmo genético si no tiene cruce, y sin embargo puede serlo perfectamente sin mutación, según descubrió John Holland el teorema de los esquemas confía en él para hallar la mejor solución a un problema.

Los operadores de cruces más utilizados son las siguientes, ver [64], [15]:

- *Cruce básico o Monopunto:* Bajo este operador de cruce, aleatoriamente se genera un número entero x entre 1 y $l - 1$, con el fin de dividir los cromosomas de cada padre en dos cadenas de genes, la primera desde el gen 1 hasta el gen x y la segunda desde el gen $x + 1$ hasta el gen l , para luego intercambiar las cadenas que se encuentran a la derecha del gen en la posición x , ver [64], [15]. El cruce basado en un punto se ilustra de la siguiente manera:

Sean P_1 y P_2 los padres a cruzarse en la posición $x = 5$

$$P_1 = [00011010101]$$

$$P_2 = [11110000001]$$

generando los hijos h_1 y h_2

$$h_1 = [00011000001]$$

$$h_2 = [11110010101]$$

- *Cruce multipunto:* igual que el cruce básico, sólo que estableciendo más de un punto de cruce. De Jong realizó algunos trabajos en los cuales analizó el desempeño del operador de cruce basado en múltiples puntos, obteniendo como resultado que para poblaciones grandes el mejor es el cruce basado en dos puntos, en el cual, aleatoriamente se generan dos números enteros x e y entre 1 y $l - 1$ (x menor que y) con el fin de dividir los cromosomas de cada padre en tres cadenas de genes, la primera desde el gen 1 hasta el gen x , la segunda desde el gen $x + 1$ hasta el gen y y la tercera desde el gen $y + 1$ hasta el gen l , [13].

Sean P_1 y P_2 los padres a cruzarse en la posición $x = 5$ y $y = 7$

$$P_1 = [00011010101]$$

$$P_2 = [11110000001]$$

generando los hijos h_1 y h_2

$$h_1 = [00011000101]$$

$$h_2 = [11110010001]$$

- *Cruce segmentado*: Consiste en un cruce multipuntos en el cual varía el número de puntos de cruce, conforme se va formando la nueva cadena del descendiente, en cada gen, se verifica si ahí se va producir un cruce.
- *Cruce uniforme*: En el denominado operador de cruce uniforme cada gen en la descendencia se crea copiando el correspondiente gen de uno de los dos padres, escogido de acuerdo a una máscara de cruce, la cual es un cromosoma del mismo tamaño del que se piensa cruzar, generado aleatoriamente. Cuando existe un 1 en la máscara de cruce, el gen es copiado del primer padre, mientras que cuando exista un 0 en la máscara, el gen es copiado del segundo padre, [62].

Existen otros operadores de cruce los cuales han surgido ha medida que han sido surgiendo distintos problemas de aplicación, entre ellos se tienen: Operador de cruce basado en una correspondencia parcial (PMX), [30], Cruces sobre alfabetos de alta cardinalidad, [8], Operador de cruce basado en ciclos (CX), [50], Operador de cruce basado en el orden 1 (OX1), [12], Operador de cruce basado en el orden 2 (OX2), [62], Operador de cruce basado en la posición (POS),[62], Operador de cruce basado en la alternancia de las posiciones (AP),[43].

Asociada al operador de cruce existe la función de reproducción global, la cual puede describirse como sigue, ver [60]: La función de reproducción global, f_{prod} , produce una población de descendientes $z \in P_\lambda$ a partir de individuos seleccionados $y \in P_\lambda$ por medio de un operador de cruce:

$$f_{prod} : (\beta, y) \rightarrow z$$

donde β es un vector de dimensión $\lambda \div 2$ constituido por valores escogidos aleatoriamente entre 1 y $l - 1$.

La función de reproducción se dice que está basada en un punto si los padres $I^i = (s_1, \dots, s_l)$ y $I^j = (b_1, \dots, b_l)$ producen hijos $CH^{i,j;1} = (c_1, \dots, c_l)$ y $CH^{i,j;2} = (d_1, \dots, d_l)$ verificándose que:

$$c_j = \begin{cases} s_j & \text{si } j \leq m \\ b_j & \text{si } j > m \end{cases}$$

$$d_j = \begin{cases} s_j & \text{si } j \leq m \\ b_j & \text{si } j > m \end{cases}$$

donde m es un número entero escogido al azar según una distribución uniforme discreta definida sobre el conjunto $1, \dots, l - 1$.

Operador de Mutación

El operador de Mutación es el encargado de diversificar la búsqueda hacia regiones alejadas del espacio de búsqueda actual y proporciona un pequeño elemento de aleatoriedad en la vecindad de los individuos de la población. En la evolución, la mutación es un suceso debido a un cambio, usualmente aleatorio, del material genético de un individuo. Similarmente a como sucede en la naturaleza, en los Algoritmos Genéticos es un proceso en el cual, por lo general, aleatoriamente se cambia el valor del alelo para algunos de los genes de los cromosomas, usualmente se lleva a cabo generando un número aleatoriamente y si éste es menor que la probabilidad de mutación, usualmente en el intervalo $(0, 0.1]$, el bit cambiará, de lo contrario se dejará como está.

Es recomendado no abusar de la mutación, porque aunque es cierto que es un mecanismo generador de diversidad, permitiendo la exploración de la búsqueda hacia zonas alejadas del espacio de soluciones actuales, logrando con esto que el Algoritmo Genético no se quede estancado, también es cierto que puede reducir el Algoritmo Genético a una búsqueda aleatoria. De Jong recomienda la utilización de una probabilidad de mutación del bit de l^{-1} . La naturaleza del cambio depende de la composición de los bloques de los cromosomas. Si cada bloque es un bit (en la codificación binaria), el único cambio posible es invertir su valor. Si los bloques son números reales, la modificación podría ser la suma o sustracción de un pequeño valor aleatorio.

Algunas de técnicas de mutación [64], son citadas a continuación.

- *Mutación de bit*: existe una única probabilidad de que se produzca una mutación de algún bit, de producirse, el algoritmo toma aleatoriamente un bit, el bit cambiará de 0 a 1 o de 1 a 0 de acuerdo al caso.
- *Mutación multibit*: cada bit tiene una probabilidad de mutarse o no, que es calculada en cada pasada del operador de mutación multibit.
- *Mutación de gen*: igual que la mutación de bit, solamente que, en vez de cambiar un bit, cambia un gen completo.
- *Mutación multigen*: igual que la mutación de multibit, solamente que, en vez de cambiar un conjunto de bits, cambia un conjunto de genes.
- *Mutación de intercambio*: existe una probabilidad de que se produzca una mutación. De producirse, toma dos bits/genes aleatoriamente y los intercambia.
- *Mutación de barajado*: existe una probabilidad de que se produzca una mutación. De producirse, toma dos bits o dos genes aleatoriamente y baraja de forma aleatoria los bits o genes, según se hubiera escogido, comprendidos entre los dos.

Cabe anotar que en la mayoría de la literatura existente, las probabilidades de cruce y mutación son fijas y constantes durante todas las iteraciones del algoritmo, sin embargo, en algunos artículos como: [1], [8], [16] y [46], han cambiado la probabilidad de mutación a medida que aumenta el número de iteraciones obteniendo mejores resultados.

La función que describe el proceso de mutación puede verse como: La función de mutación individual $f_{ind-mut}$, aplicada a $I = (s_1, \dots, s_l)$, genera otro individuo $MI = (sm_1, \dots, sm_l)$, es

decir $f_{ind-mut} = MI$, tal que $\forall j \in 1, \dots, l P(sm_j = s_j) = 1 - p_m$, donde p_m es la probabilidad de mutación.

Otros operadores menos empleados son:

- **Cromosomas de Longitud Variable:** Se utilizan cromosomas cuyo tamaño de la longitud varía de generación a generación.
- **Operadores de Nicho:** Estos operadores están encaminados a mantener la diversidad genética de la población, de forma que cromosomas similares sustituyan sólo a cromosomas similares.
- **Operadores Especializados:** En una serie de problemas hay que restringir las nuevas soluciones generadas por los operadores genéticos, pues no todas las soluciones generadas van a ser válidas, sobre todo en los problemas combinatorios y en los problemas con restricciones. Por ello, se aplican operadores que mantengan la estructura del problema.

Algoritmos

El Algoritmo Genético Canónico

A continuación se muestra el pseudocódigo de el Algoritmo Genético Simple, también conocido como algoritmo Genético Canónico, el cual es de gran importancia, ya que es la base sobre la cual está fundamentado el teorema de los esquemas.

El pseudocódigo de un algoritmo Genético, es el siguiente:

```

BEGIN /* Algoritmo Genético Simple */
    Generar una población inicial.
    Evaluar la función de adaptación de cada individuo.
    WHILE NOT Terminado DO
        BEGIN /* crear una nueva generación */
            FOR Tamaño población/2 DO
                BEGIN /*Ciclo Reproductivo */
                    Seleccionar dos individuos de la anterior generación, para el cruce (probabilidad de
                    selección proporcional a la función de evaluación del individuo).
                    Cruzar con cierta probabilidad los dos individuos obteniendo dos descendientes.
                    Mutar los dos descendientes con cierta probabilidad.
                    Evaluar la función de evaluación de los dos descendientes mutados.
                    Insertar los dos descendientes mutados en la nueva generación.
                END
            END
        END
    END
    IF la población ha convergido THEN

```

```

    Terminado := TRUE
  END
END

```

Para implementar el Algoritmo Genético simple, es necesario iniciar con una codificación del problema. Además se requiere una función de adaptación al problema, la cual asigne un número real a cada posible solución codificada. Durante la ejecución del algoritmo, los dos padres deben ser seleccionados para la reproducción con una probabilidad de selección proporcional a la función de evaluación del individuo, luego los padres seleccionados se cruzarán generando dos hijos, los cuales después de mutados formarán la nueva población.

El Algoritmo Genético Abstracto

Un algoritmo que presenta una modificación del algoritmo genético canónico, en cuanto a que la selección no tiene que ser en base a la probabilidad de selección proporcional a la función de evaluación y en cuanto a que el reemplazo de la población no se efectúa de forma inmediata si no que escoge los individuos mejor adaptados entre padres e hijo para formar la nueva generación, es el siguiente:

```

BEGIN /* Algoritmo Genético Abstracto */
  Generar una población inicial.
  Evaluar la función de adaptación de cada individuo.
  WHILE NOT Terminado DO
    BEGIN /* Crear nueva generación */
      FOR Tamaño población/2 DO
        BEGIN /*Ciclo Reproductivo */
          Seleccionar dos individuos de la anterior generación, para el cruce.
          Cruzar con cierta probabilidad los dos individuos obteniendo dos descendientes.
          Mutar los dos descendientes con cierta probabilidad.
          Evaluar la función de adaptación de los dos descendientes mutados.
          Extender la población añadiendo los hijos.
          Reducir la población extendida.
        END
      IF la población ha convergido THEN
        Terminado := TRUE
      END
    END
  END
END

```

Criterio de Parada

En cuanto al criterio de parada, generalmente el algoritmo termina cuando un porcentaje alto de la población converge a un valor o después de un número fijo de evaluaciones en la función de aptitud o también viene determinado por criterios tales como, un número máximo de generaciones o un tiempo máximo de resolución, o más eficientemente por estrategias relacionadas con indicadores del estado de evolución de la población, como por la pérdida de diversidad dentro de la población o por no haber mejora en un cierto número de iteraciones, siendo por lo general una condición mixta lo más utilizado, es decir, limitar el tiempo de ejecución a un número de iteraciones y tener en cuenta algún indicador del estado de la población para considerar la convergencia antes de alcanzar tal limitación. Usualmente se dice que un gen ha convergido cuando al menos el 95 % de los cromosomas tienen el mismo alelo y cuando al menos un β % (usualmente alto) de los genes han convergido se dice que la población converge.

Tomando como base lo expuesto en [60], en la siguiente sección se estudiará el por qué funcionan los Algoritmos Genéticos, es decir se estudiará porque dicho algoritmo es capaz de encontrar el óptimo de la función durante el proceso de búsqueda; con este fin se desarrollará el denominado **Teorema de los esquemas**, para el caso del Algoritmo Genético Canónico, en el cual se utiliza el alfabeto binario, la creación de la población inicial es aleatoria, la selección está basada en la ruleta, el reemplazo es inmediato, el cruce se considera monopunto, la mutación es gen a gen y no se considera la aparición de individuos no factibles.

Teorema de los Esquemas

El teorema de los esquemas de John Holland caracteriza la evolución de una población en un Algoritmo Genético en términos del número de elementos que representa cada esquema.

El teorema está basado en el concepto del esquema, el cual puede definirse como una cadena de bits formada por los elementos del conjunto $\{1, 0, *\}$. El orden del esquema H , se denota por $o(H)$, y se define como el número de posiciones fijas en el esquema. La longitud del esquema H , se denota por medio de $\delta(H)$, y se define como la distancia entre la última y la primera posición fijas de la cadena. Por ejemplo, el esquema $H = 0*10$ representa el conjunto de cadenas $\{0010, 0110\}$, su orden $o(H) = 3$ y su distancia $\delta(H) = 3$.

Teorema de los Esquemas: Sea $m(H, g)$ el número de individuos de la población en un generación g que se asocian a un esquema H , después de efectuar la selección, el cruce y la mutación en un Algoritmo Genético Canónico, se tiene que:

$$m(H, g + 1) \geq \frac{g(H)}{\bar{g}} \cdot m(H, 0) \cdot (1 + c)^g \cdot \left[1 - p_c \frac{\delta(H)}{l - 1}\right] \cdot [1 - p_m \cdot o(H)]$$

Demostración:

Primero se estudiará el efecto de la selección basada en la ruleta sesgada. A continuación se verá el efecto del cruce en dicho número esperado, para finalmente tratar el efecto de la mutación, obteniéndose una cota inferior para $m(H, g + 1)$.

Efecto de selección

Dado que la probabilidad de que el individuo i sea seleccionado para cruzarse está dada por

$$P(x_i) = \frac{g(x_i)}{\sum_i g(x_i)}$$

Se tiene que la probabilidad de que un individuo seleccionado para cruzarse se empareje con el esquema H , se calculará por medio de

$$P(H) = \frac{g(H)}{\sum_i g(x_i)}$$

Por lo tanto, el número esperado de individuos que a partir del anterior individuo seleccionado van a asociarse con H , será

$$\frac{g(H)}{\sum_i g(x_i)} m(H, g)$$

Al efectuarse M selecciones, se tendrá que:

$$m(H, g + 1) = M \frac{g(H)}{\sum_i g(x_i)} m(H, g)$$

Ahora, si denotamos por \bar{g} el promedio de aptitud de la población entera el cual está dado por

$$\bar{g} = \frac{\sum_i g(x_i)}{M}$$

Entonces

$$m(H, g + 1) = \frac{g(H)}{\bar{g}} m(H, g)$$

Si se asume que la evaluación del esquema supera a la evaluación media extendida a toda la población en un $\epsilon\%$, es decir si $g(H) = \bar{g} + \epsilon \cdot \bar{g}$ la ecuación anterior puede simplificarse a

$$m(H, g + 1) = (1 + \epsilon) m(H, g)$$

Iniciando en $g = 0$, y asumiendo un valor estacionario para ϵ , se tiene que

$$m(H, g + 1) = (1 + \epsilon)^g m(H, 0)$$

Efecto de cruce

Iniciamos esta descripción con un par de ejemplos, buscando con esto ofrecer una mayor claridad. Considerese el esquema $H_1 = [1 * 0 * **]$ cuya longitud está dada por $\delta(H_1) = 2$. Asumiendo que el cruce utilizado es el cruce Monopunto, se tienen 2 posibilidades en las cuales se puede cortar y destruir el esquema, de lo contrario el esquema permanece intacto, por lo tanto si denotamos por $p_d(H_1)$ la probabilidad de que el esquema H_1 sea destruido entonces $p_d(H_1) = \frac{2}{5}$, luego la probabilidad de que sobreviva, denotada por $p_s(H_1)$ está dada por $p_s(H_1) = \frac{3}{5}$, es decir $p_s(H_1) = 1 - p_d(H_1)$. De lo anterior, podemos generalizar diciendo que:

$$p_d(H) = \frac{\delta(H)}{l - 1}$$

En realidad como el operador de cruce se lleva a cabo con una cierta probabilidad p_c , se tiene

$$p_d(H) = \frac{\delta(H)}{l-1} p_c$$

De donde

$$p_s(H) = 1 - \frac{\delta(H)}{l-1} p_c$$

En realidad la fórmula anterior proporciona una cota para la probabilidad de que sobreviva el esquema H . Ello es debido a que, puede ocurrir que aunque el punto de corte se encuentre entre los símbolos * comprendidos entre el primer y el último elemento distintos de *, el esquema resultante no desaparezca. De ahí que en realidad se tiene que:

$$m(H, g+1) \geq 1 - \frac{\delta(H)}{l-1} p_c$$

Efecto de cruce y selección

Asumiendo independencia entre selección, la estimación del número de un tipo particular de un esquema H esperado en la próxima generación dado el efecto combinado entre selección y cruzamiento está dado por el producto de las expresiones anteriores, es decir

$$m(H, g+1) \geq \frac{g(H)}{\bar{g}} (1 + \epsilon)^g m(H, 0) \left[1 - \frac{\delta(H)}{l-1} p_c\right]$$

Efecto de Mutación

La mutación introduce cambios aleatorios para un único bit. Para que un esquema sobreviva, ninguno de los $o(H)$ bits fijos debe ser afectado por la mutación. Si p_m es la probabilidad de mutación de un cierto bits, entonces su probabilidad de supervivencia es $1 - p_m$. Ya que la mutación de un individuo es estadísticamente independiente de los demás bits, un esquema particular sobrevivirá si todas sus $o(H)$ posiciones fijas sobreviven. Así, la probabilidad de que un esquema H sobreviva a la mutación, lo cual denotaremos por p_{sm} está dada por la multiplicación de las probabilidades de supervivencia de cada cada uno de sus bits, es decir

$$p_{sm} = (1 - p_m)^{o(H)}$$

Al ser $p_m \ll 1$, la anterior probabilidad de sobrevivir puede ser aproximada por

$$p_{sm} = 1 - p_m \cdot o(H)$$

Efecto de Selección, Cruce y Mutación

Asumiendo independencia entre selección y cruce la estimación del número esperado de copias de un esquema H en la próxima generación, dado el efecto combinado entre selección y cruce y mutación está dado por el producto de las probabilidades anteriores, lo cual nos proporciona el denominado *Teorema de los esquemas*.

$$m(H, g+1) \geq \frac{g(H)}{\bar{g}} m(H, 0) (1 + c)^g \left[1 - \frac{\delta(H)}{l-1} p_c\right] [p_{sm} = 1 - p_m o(H)]$$

Capítulo 4

Algoritmo Genético aplicado al modelo (Q, r) Multi-Artículo

4.1. Introducción

En el presente capítulo se muestran los resultados obtenidos por cada una de las metodologías expuestas en el capítulo anterior, haciendo distinción entre los resultados obtenidos para el modelo de inventario (3.4) y el modelo de inventario (3.3), ellos se diferencian en que en el primero no se consideran pedidos pendientes en el momento de recibir nuevamente las Q_i unidades solicitadas para surtir el artículo i -ésimo, y el segundo si considera este tipo de pedidos. Los resultados mostrarán la conveniencia de los Algoritmos Genéticos en el caso del problema (3.3).

A continuación se ilustra la implementación y resultados obtenidos en cada una de las metodologías y en cada uno de los dos problemas, los cuales fueron aplicados a un problema ficticio de inventarios con 100 artículos, donde los datos de costos por unidad, demandas anuales y tiempos de reposición fueron generados de manera aleatoria, los cuales son recopilados en la siguiente tabla.

Costos, demandas y tiempos de reposición para 100 artículos

i	c_i	D_i	l_i	θ_i	σ_i
1	50	22	81	4.88219	2.20957
2	37	16	20	0.876712	0.936329
3	10	19	24	1.24932	1.11773
4	66	20	42	2.30137	1.51703
5	56	20	86	4.71233	2.17079
6	71	23	62	3.90685	1.97658
7	51	21	62	3.56712	1.88868
8	74	23	93	5.86027	2.4208
9	52	20	61	3.34247	1.82824
10	85	24	10	0.657534	0.810885
11	79	15	82	3.36986	1.83572
12	46	20	64	3.50685	1.87266
13	71	17	53	2.46849	1.57114
14	43	19	34	1.76986	1.33036
15	75	21	33	1.89863	1.37791
16	30	17	53	2.46849	1.57114
17	88	21	30	1.72603	1.31378
18	26	18	14	0.690411	0.83091
19	7	23	25	1.57534	1.25513
20	68	16	33	1.44658	1.20274
21	75	16	27	1.18356	1.08792
22	84	24	64	4.20822	2.05139
23	30	16	84	3.68219	1.9189
24	73	16	82	3.59452	1.89592
25	49	16	82	3.59452	1.89592
26	42	15	73	3	1.73205
27	74	19	87	4.52877	2.12809
28	86	21	80	4.60274	2.1454
29	7	23	28	1.76438	1.3283
30	64	22	99	5.96712	2.44277

i	c_i	D_i	l_i	θ_i	σ_i
31	22	22	18	1.08493	1.0416
32	36	19	68	3.53973	1.88142
33	25	22	29	1.74795	1.3221
34	75	24	31	2.03836	1.42771
35	90	20	40	2.19178	1.48047
36	59	18	27	1.33151	1.15391
37	64	22	55	3.31507	1.82073
38	84	17	95	4.42466	2.10349
39	62	17	45	2.09589	1.44772
40	52	19	62	3.2274	1.7965
41	35	21	64	3.68219	1.9189
42	58	21	74	4.25753	2.06338
43	33	16	46	2.01644	1.42001
44	37	16	87	3.8137	1.95287
45	34	17	92	4.28493	2.07001
46	43	22	77	4.6411	2.15432
47	46	19	35	1.82192	1.34978
48	82	23	81	5.10411	2.25923
49	19	24	22	1.44658	1.20274
50	31	18	55	2.71233	1.64691
51	52	15	64	2.63014	1.62177
52	44	20	73	4	2
53	24	22	44	2.65205	1.62851
54	53	18	75	3.69863	1.92318
55	80	19	89	4.63288	2.15241
56	6	17	15	0.69863	0.835841
57	40	24	22	1.44658	1.20274
58	8	19	87	4.52877	2.12809
59	51	20	47	2.57534	1.60479
60	49	17	47	2.18904	1.47954
61	70	15	96	3.94521	1.98625
62	21	15	77	3.16438	1.77887
63	53	23	98	6.17534	2.48502
64	42	22	31	1.86849	1.36693
65	78	16	18	0.789041	0.88828

i	c_i	D_i	l_i	θ_i	σ_i
66	60	23	44	2.7726	1.66511
67	32	21	55	3.16438	1.77887
68	32	21	61	3.50959	1.87339
69	23	24	97	6.37808	2.52549
70	86	15	54	2.21918	1.48969
71	27	19	46	2.39452	1.54742
72	15	19	99	5.15342	2.27012
73	33	20	33	1.80822	1.3447
74	67	15	69	2.83562	1.68393
75	64	21	96	5.52329	2.35017
76	64	15	70	2.87671	1.69609
77	1	20	36	1.9726	1.40449
78	34	22	57	3.43562	1.85354
79	82	19	10	0.520548	0.72149
80	29	21	89	5.12055	2.26286
81	54	17	46	2.14247	1.46372
82	27	21	37	2.12877	1.45903
83	12	18	95	4.68493	2.16447
84	35	16	51	2.23562	1.4952
85	74	18	35	1.72603	1.31378
86	89	17	17	0.791781	0.889821
87	78	17	62	2.88767	1.69931
88	8	21	23	1.32329	1.15034
89	31	20	16	0.876712	0.936329
90	22	19	62	3.2274	1.7965
91	29	17	35	1.63014	1.27677
92	89	23	42	2.64658	1.62683
93	50	23	75	4.72603	2.17394
94	68	17	87	4.05205	2.01297
95	76	21	41	2.3589	1.53587
96	16	22	96	5.7863	2.40547
97	82	24	95	6.24658	2.49931
98	10	24	28	1.8411	1.35687
99	68	18	79	3.89589	1.9738
100	66	16	65	2.84932	1.68799

4.2. Resultados para el problema sin pedidos pendientes

El problema (3.4) puede ser tratado por las tres metodologías expuestas en el capítulo anterior. En lo que sigue se ilustra la manera en que fueron implementadas tales metodologías y los resultados obtenidos.

4.2.1. Multiplicadores de lagrange

Como fue discutido en el capítulo anterior, debido a la no convexidad de la función objetivo, el modelo de inventarios (Q, r) , usualmente es restringido al caso en el que no hayan pedidos pendientes en el momento en que llega una nueva orden, para poder garantizar optimalidad en el problema y poder resolverlo a través de multiplicadores de Lagrange. A continuación será descrita la forma que toma el problema para la implementación de tal metodología.

Como en [37], con el fin de resolver el problema a través de multiplicadores de Lagrange, se introducen dos constantes de penalización λ y μ , para las restricciones de frecuencia de pedido y nivel de satisfacción, respectivamente, las cuales se pueden escribir como:

$$\frac{1}{N} \sum_{i=1}^N \frac{D_i}{Q_i} - F \leq 0$$

y

$$S - \frac{1}{D_{tot}} \sum_{i=1}^N D_i G(r_i) \leq 0$$

donde el lagrangiano está dado por:

$$\ell = \sum_{i=1}^N c_i \left(\frac{Q_i + 1}{2} + r_i - \theta_i \right) + \lambda \left(\frac{1}{N} \sum_{i=1}^N \frac{D_i}{Q_i} - F \right) + \mu \left(S - \frac{1}{D_{tot}} \sum_{i=1}^N D_i G(r_i) \right).$$

Para hallar el valor óptimo de Q_i , se deriva con respecto a Q_i y se iguala a cero, es decir

$$\frac{\partial \ell}{\partial Q_i} = \frac{c_i}{2} - \frac{\lambda D_i}{N Q_i^2} = 0$$

$$Q_i(\lambda) = \sqrt{\frac{2\lambda D_i}{N c_i}}$$

ahora derivando con respecto a λ e igualando a cero, se tiene que:

$$\frac{1}{N} \sum_{i=1}^N \frac{D_i}{Q_i} = F$$

Sustituyendo el valor de Q_i

$$\frac{1}{N} \sum_{i=1}^N \frac{D_i}{\sqrt{\frac{2\lambda D_i}{N c_i}}} = F$$

simplificando, se tiene

$$\frac{1}{\sqrt{2N\lambda}} \sum_{i=1}^N \sqrt{D_i c_i} = F$$

de donde

$$\lambda = \left(\frac{1}{F \sqrt{2N}} \sum_{i=1}^N \sqrt{D_i c_i} \right)^2$$

por lo tanto el valor óptimo de Q_i , ver [48], está dado por:

$$Q_i = \sqrt{\frac{2\left(\frac{1}{F\sqrt{2N}} \sum_{i=1}^N \sqrt{D_i c_i}\right)^2 D_i}{N c_i}}.$$

Ahora para hallar el valor óptimo de r_i , se deriva con respecto a r_i y se iguala a 0, es decir

$$\frac{\partial \ell}{\partial r_i} = c_i - \frac{\mu D_i}{D_{tot}} G'(r_i) = 0$$

o equivalentemente,

$$\frac{\partial \ell}{\partial r_i} = c_i - \frac{\mu D_i}{D_{tot}} g_i(r_i) = 0.$$

Suponiendo que la demanda durante el tiempo de reposición es aproximada por una distribución normal con media θ y desviación σ , entonces la función de distribución esta dada por: $g(x) = \frac{\phi(z)}{\sigma}$, con $z = \frac{x-\theta}{\sigma}$, luego

$$c_i - \frac{\mu D_i}{D_{tot}} \frac{\phi\left(\frac{x-\theta_i}{\sigma_i}\right)}{\sigma_i} = 0$$

y como $\phi(z) = \frac{e^{-z^2/2}}{\sqrt{2\pi}}$,

se tiene que

$$c_i - \frac{\mu D_i}{D_{tot} \sigma_i \sqrt{2\pi}} e^{-(\frac{r_i - \theta_i}{\sigma_i})^2/2} = 0$$

de donde

$$e^{-(\frac{r_i - \theta_i}{\sigma_i})^2/2} = \frac{\sqrt{2\pi} c_i D_{tot} \sigma_i}{\mu D_i}$$

$$\left(\frac{r_i - \theta_i}{\sigma_i}\right)^2 = -2 \ln \left(\frac{\sqrt{2\pi} c_i D_{tot} \sigma_i}{\mu D_i} \right)$$

y por lo tanto

$$r_i(\mu) = \theta_i + \sqrt{-2\sigma_i^2 \ln \left(\frac{\sqrt{2\pi} c_i D_{tot} \sigma_i}{\mu D_i} \right)}.$$

Ahora como el tiempo de reposición es aleatorio y la desviación de la demanda durante el tiempo de reposición está dada por

$$\sigma_i = \sqrt{\theta_i + (d\sigma_l)^2}$$

entonces, el valor óptimo para $r_i(\mu)$, como puede verse en [36], [37], está dado por:

$$r_i(\mu) = \theta_i + \sqrt{-2(\theta_i + (d\sigma_l)^2) \ln \left(\frac{\sqrt{2\pi} c_i D_{tot} \sqrt{\theta_i + (d\sigma_l)^2}}{\mu D_i} \right)}$$

y el valor de Q_i , el cual se calculo anteriormente está dado por:

$$Q_i = \sqrt{\frac{2\left(\frac{1}{F\sqrt{2N}} \sum_{i=1}^N \sqrt{D_i c_i}\right)^2 D_i}{N c_i}}$$

Como se afirma en [48], hallar el valor de μ , es muy complejo, pues observemos que si derivamos con respecto a μ e igualando a cero se obtiene

$$S = \frac{1}{D_{tot}} \sum_{i=1}^N D_i \sum_{k=0}^{r_i} \frac{e^{-\lambda} \lambda^k}{k!}$$

y sustituyendo el valor de r_i se tiene que:

$$S = \frac{1}{D_{tot}} \sum_{i=1}^N D_i \sum_{k=0}^{\theta_i + \sqrt{-2\sigma_i^2 \ln\left(\frac{\sqrt{2\pi} c_i D_{tot} \sigma_i}{\mu D_i}\right)}} \frac{e^{-\lambda} \lambda^k}{k!}$$

Es por esto, que con el fin de hallar, por lo menos un valor aproximado de μ , que optimice el valor de r_i , usualmente se recurre a un algoritmo basado en la técnica de bisección, sin embargo, debido a que para poder inicializar el algoritmo es necesario una cota para dicho valor, hemos encontrado que para que el valor de r_i sea real es necesario que

$$\frac{\sqrt{2\pi} c_i D_{tot} \sigma_i}{\mu D_i} \leq 1$$

es decir, que

$$\mu \geq \frac{\sqrt{2\pi} c_i D_{tot} \sigma_i}{D_i}$$

dicha expresión proporciona una cota inferior para inicializar el algoritmo.

A continuación, haciendo algunas adaptaciones sobre lo expuesto en [37] y en [48], presentamos el algoritmo que busca el valor de el multiplicador de Lagrange μ a través de la técnica de bisección.

Algoritmo (Modelo multi-producto(Q,r))

Paso 0 Escoger un valor inicial para μ_1 , y un nivel de tolerancia ϵ . Tomar $t = 1$

Paso 1 Usar μ_t para calcular r_i^t para $i : 1, \dots, N$

Paso 2 Calcular el nivel de servicio

$$S_t = \frac{1}{D_{tot}} \sum_{i=1}^N D_i G_i(r_i^t)$$

Si $|S_t - S| < \epsilon$, Pare; $r_i^* = r_i^t$, para $i : 1, \dots, N$

Sino

Si $S_t < S$, incremente μ_t ; si $S_t > S$, reduzca μ_t

Hacer $t = t + 1$ y regresar al paso 1

4.2.2. Algoritmos Genéticos

La implementación del Algoritmo Genético, en Matlab, para el problema (3.4), en el cual no se consideran pedidos pendientes, tiene la misma estructura utilizada para el problema (3.3), la cual será tratada de manera detallada en la siguiente sección. La diferencia consiste en que en la implementación de este problema no se cobra el costo por unidades faltantes.

4.2.3. Optimización-Simulación

Como se mencionó en el capítulo anterior, la metodología de Optimización-Simulación será implementada a partir del software de simulación Simul8, el cual es un paquete muy utilizado en la actualidad gracias a su amigable presentación, la cual permite recrear visualmente las situaciones reales que van a ser modeladas, sin tener que hacer mayores suposiciones y restricciones sobre el modelo real. Otras de las bondades que presenta dicho software es que permite optimizar el modelo a través del software de optimización OpQuest y programar los eventos en el ambiente del Visual Basic.

En una simulación en Simul8, usualmente existen los siguientes cuatro bloques básicos: 1) El Punto de Entrada de trabajo, el cual es el sitio donde llegan las unidades que van a ser procesadas. 2) El área del almacenamiento, que es lugar donde las unidades esperan para ser procesadas. 3) El centro de trabajo, que es donde se realiza el proceso de trabajo y 4) El punto de salida, que es de donde salen las unidades procesadas.

Para la implementación del modelo fueron utilizadas sólo tres de los bloques anteriormente descritos, los cuales fueron adaptadas de la siguientes forma:

El área del almacenamiento: Para adaptar dicho bloque al modelo de inventarios, por cada artículo se creó una área de almacenamiento, la cual representa la posición del inventario del artículo i . En dicha área se controla la posición del inventario, la cual aumenta o disminuye de acuerdo a si el artículo es demandado o abastecido.

Para abastecer por primera vez el inventario, el visual logic es programado de tal manera que inicialice la simulación con la suma de las cantidades óptimas Q_i y r_i obtenidas del OptQuest. A medida que avanza la simulación la posición del inventario es continuamente monitoreado de tal manera que si ésta alcanza el punto de pedido r_i y después de transcurrido el tiempo de reposición l_i , la posición del inventario es abastecida nuevamente con las Q_i unidades pedidas del artículo i . De acuerdo a los supuestos del modelo el tiempo de reposición sigue una distribución normal, la cual es creada con una varianza de un 10% de la media, dicha distribución es asignada en la programación del visual logic, cada vez que la posición alcanza el punto de pedido r_i .

Cada vez que el inventario se queda sin artículos, pues el momento de abastecer el inventario no se ha cumplido, se hace necesario abastecerlo con una unidad ficticia ya que de lo contrario la simulación no podrá continuar al no suceder ningún evento en ese lapso de tiempo.

El centro de trabajo: Para el modelo de inventarios propuesto, el centro de trabajo fue adaptado como el lugar donde ocurre la demanda del artículo i .

Entre los supuestos del problema de inventarios, la demanda está condicionada a ser considerada por unidad, es decir no se consideran demanda por lotes, por lo tanto dentro de la simulación la demanda es considerada individualmente. Otra de las condiciones impuesta sobre la demanda en el modelo de inventarios propuesto, es que ésta sigue una distribución Poisson con media X , para efectos de la simulación se trabajó con demanda exponencial, ya que con dicha distribución es posible contar el tiempo que transcurre entre dos eventos Poisson, el cual es precisamente el tiempo que cuenta el reloj que trae incorporado el Simul8.

La salida: En la salida del modelo, simplemente se muestra el total de artículos que han sido demandados durante toda la simulación.

El Simul8 tiene la bondad de permitir optimizar el modelo a través del paquete de optimización OptQuest, a continuación será descrita de manera corta, la forma en que fueron manipulados sus elementos básicos:

Para optimizar el modelo simulado a través del OptQuest, lo primero es definir el espacio de búsqueda para las variables de decisión, en este caso Q_i y r_i , es decir predefinir un intervalo de posibles valores que puede tomar cada una de las variables de decisión. Por lo tanto, para definir la cota inferior del intervalo correspondiente a cada variable de decisión, se consideraron las restricciones impuestas en el modelo 3.3, para cada uno de dichos valores, es decir, los valores para los Q_i deben ser valores enteros mayores o iguales a 1 y los valores para los r_i deben ser valores enteros no negativos, finalmente para definir las cotas superiores de los intervalos de búsqueda, se tomaron como referencia los valores obtenidos como óptimos del algoritmo implementado para resolver el problema a través de multiplicadores de Lagrange.

Dado que en el OptQuest sólo es posible definir restricciones lineales ligadas a las variables de decisión, y en vista a que las restricciones del modelo en cuestión no son lineales y además no sólo dependen de las variables de decisión, fue necesario programar dichas restricciones por medio del Visual Logic, lo cual se realizó de la siguiente forma: Para calcular el número de pedidos realizados durante todo el año, inicialmente se contabiliza a lo largo de toda la simulación, el número de veces que la posición de inventario alcanza el punto de pedido r_i (ya que este es el momento en el cual se hace un nuevo pedido). Finalmente, para obtener la frecuencia de pedido anual alcanzada en la simulación completa, se realizó el promedio de frecuencia de pedido de todos los artículos que fueron analizados después de transcurrido el tiempo de simulación indicado. Con respecto a la restricción correspondiente al nivel de satisfacción, primero se calculó la tasa de demanda satisfecha durante los tiempos de reposición, para finalmente calcular el nivel de satisfacción, realizando el promedio de demandas satisfechas, finalizado el tiempo de simulación indicado.

En cuanto al cálculo del costo de inversión del inventario sin considerar pedidos pendientes, este fue estimado de la siguiente manera: El costo por unidad, es calculado a medida que una unidad del artículo i es demandada, de tal manera que a medida que avanza la simulación, la suma de los costos de todas las unidades del artículo i va siendo actualizada. Después de calculado dicho costo y en el momento en que han transcurrido los 365 días a ser simulados, la suma del costo de todos los artículos es calculado e incrementado en el caso en que alguna de las restricciones impuestas en el problema 3.3 sean violadas, de tal manera que si la restricción no satisfecha es la correspondiente a la frecuencia de pedido, se procede sumándole al costo que se tiene hasta ese momento λ veces la diferencia entre la frecuencia de pedido alcanzada por la simulación y la frecuencia exigida como máxima en el problema, donde el factor λ es el parámetro que previamente se ha calculado al utilizar la metodología de multiplicadores de Lagrange, de igual manera si la restricción que se ha violado es la correspondiente al nivel de satisfacción, entonces, el costo que se tiene hasta ese momento es incrementado en μ veces la diferencia entre el nivel de satisfacción, exigido como mínimo, y el nivel de satisfacción alcanzado por la simulación, donde el factor μ al igual que con el factor λ es el mismo que previamente se ha calculado al utilizar la metodología de multiplicadores de Lagrange.

Como criterio de parada en el OptQuest, se eligió finalizar la optimización después de transcurrido cierto tiempo, para determinar dicho tiempo, se tomó como referencia el mismo tiempo empleado por los Algoritmos Genéticos para hallar la solución más adaptada. Dicho criterio de parada permitió además de lograr una comparación razonable entre el OptQuest y los Algoritmos Genéticos, una forma rápida de optimizar el modelo ya que de acuerdo a los experimentos realizados a lo largo del trabajo, se observó que para lograr soluciones de buena calidad en el OptQuest es necesario de un gran número de simulaciones, y por ende una gran cantidad de tiempo, a medida que aumenta el número de artículos.

4.3. Comparación de resultados para el problema sin pedidos pendientes

Como resultado de la implementación de las metodologías de Multiplicadores de Lagrange, Algoritmos Genéticos y Optimización-Simulación para el problema (3.4), a continuación se presentan tres tablas discriminadas a manera de sistemas, las cuales reflejan los resultados obtenidos de cada una de las metodologías en cuestión; en ellas se muestra el número de artículos a ser analizados, el nivel de satisfacción y frecuencia de pedido alcanzados, el costo obtenido como mínimo, así como también el número de iteraciones y el tiempo requerido en las corridas.

En general se puede observar que para el problema 3.4, referido al caso en que no hayan pedidos pendientes, la solución obtenida empleando multiplicadores Lagrange supera tanto en calidad como en eficiencia las soluciones obtenidas con las demás metodologías incluyendo la metodología propuesta basada en Algoritmos Genéticos. Aunque es de resaltar que dicha metodología sólo puede ser aplicada al caso en que no hayan pedidos pendientes, y que ésta únicamente ofrece una solución aproximada al problema, ya que en general dicha metodología es utilizada para funciones continuas y diferenciables y en este caso esta implementándose para un problema cuyas variables de decisión son enteras y discretas. De los resultados obtenidos de implementar Algoritmos Genéticos, se puede concluir que las soluciones alcanzadas como óptimas del problema, al igual que con multiplicadores de Lagrange, están muy cercanas a la frontera de la región factible, lo cual explica que los costos obtenidos como mínimos estén relativamente cerca a los obtenidos en Lagrange. También es importante resaltar que los tiempos empleados para el cálculo de la solución más adaptada en Algoritmos Genéticos fueron relativamente razonables, a pesar que tanto el número de iteraciones como los tiempos de corridas se incrementaron a medida que se fueron aumentando el número de artículos. Finalmente al observar los resultados obtenidos de Simul8 se puede notar, que al tomar como referencia el tiempo empleado para la convergencia en Algoritmos Genéticos, se obtuvieron costos más altos a los obtenidos por esta última metodología, lo cual se ve reflejado en que dichas soluciones están más alejadas de la frontera produciendo con esto costos que aunque son factibles son muy elevados debido a los valores tan altos que deben tomar, en especial los valores de los Q_i , para obtener soluciones que estén en el interior de la región factible.

4.3. COMPARACIÓN DE RESULTADOS PARA EL PROBLEMA SIN PEDIDOS PENDIENTES⁵³

Resultados obtenidos de Multiplicadores de Lagrange						
Sistema	Núm Prod	S_t	F_t	II_i	<i>numiter</i>	<i>TiemCorr</i>
1	10	0.937843	12	1869.94	20000	2.53228236 seg
2	20	0.949996	12	3540.23	6	0.0229334647 seg
3	30	0.948203	12	5661.19	20000	2.7983211 seg
4	40	0.949863	12	7568.47	4	0.0359759991 seg
5	50	0.949858	12	9130.02	5	0.042825075 seg
6	60	0.94942	12	10450.1	3	0.049058495 seg
7	70	0.949871	12	12159.4	3	0.0555892018 seg
8	80	0.950114	12	13466.9	6	0.062383512 seg
9	90	0.950612	12	14650.1	5	0.0720018166 seg
10	100	0.949236	12	16575.6	5	0.0820265462 seg

Resultados obtenidos de Algoritmos Genéticos						
Sistema	Núm Prod	S_t	F_t	II_i	<i>numiter</i>	<i>TiemCorr</i>
1	10	0.94682	12.6583	2056.35	395	36.5988458 seg
2	20	0.944443	13.0342	3720.86	1422	303.868807 seg
3	30	0.949933	13.7061	6101.49	2616	615.125369 seg
4	40	0.949043	13.3556	8305.27	5243	1520.7232 seg
5	50	0.949192	12.1623	10368.9	7290	2777.66017 seg
6	60	0.950075	12.3592	11841.9	8880	4794.00077 seg
7	70	0.948068	12.1108	13916.3	11215	6851.35253 seg
8	80	0.947748	12.048	15122.3	12707	9445.42901 seg
9	90	0.949931	12.0018	16558.9	14378	12611.4256 seg
10	100	0.948678	12.0475	19212.2	15893	15086.4888 seg

Resultados obtenidos de Simul8						
Sistema	Núm Prod	S_t	F_t	II_i	<i>numsimul</i>	<i>TiemCorr</i>
1	10	0.9525485	4.09	3920.1258	18	1 min
2	20	0.9690338	3.025	7343.228	61	6 min
3	30	0.96341120	2.82	13983.6	92	10 min
4	40	0.9684672	2.385	20365.7	191	25 min
5	50	0.9571896	2.812	21211.61488	577	46 min
6	60	0.958853	3.21333	20687.431	699	80 min
7	70	0.9645818	3.51857	22337.762	768	114 min
8	80	0.9733378	3.5025	33616.475	1723	157 min
9	90	0.9070378	1.47111	392301.6708	1239	210 min
10	100	0.9162881	1.566	363676.2539	2564	251 min

4.4. Resultados para el problema con pedidos pendientes

Para el problema (3.3) dos de las metodologías expuestas en el capítulo anterior son implementables, ellas son: Algoritmos Genéticos y Optimización-Simulación. Los multiplicadores de Lagrange no pueden ser aplicados a este problema dado que en la función objetivo, la función correspondiente al término que representa el pedido pendiente no se sabe si es convexa dentro del dominio de definición del problema, como se demostró en el Teorema 3.1.2 y por lo tanto no es posible garantizar optimalidad en el problema. Es de resaltar que bajo la metodología de multiplicadores de Lagrange se ha encontrado que algunas de las soluciones arrojadas como las mejores, no verifican el nivel de satisfacción indicado, ya que el algoritmo que se utiliza para hallar el multiplicador μ , fácilmente puede ocasionar que los valores de los r_i sean imaginarios o definitivamente no logra el nivel de satisfacción indicado, produciendo la no convergencia del método al no alcanzar el nivel de tolerancia exigido.

4.4.1. Algoritmos Genéticos

Para el Algoritmo Genético que dará una solución al problema de inventarios (3.3), en esta sección, se hará la implementación de dicho algoritmo y luego vendrán las tablas de resultados. Existen cuatro parámetros a ajustar en este algoritmo ellos son: Método de cruce, Probabilidad de cruce, Método de mutación y Probabilidad de mutación, para efectos de ajustes, se utilizarán pruebas de hipótesis para promedios y varianzas, cuya teoría fue tratada en capítulos anteriores.

Con el fin de poder aplicar Algoritmos Genéticos en la búsqueda de soluciones para el problema de inventarios, lo primero que se hará es exponer la forma de codificar y decodificar los individuos que harán parte de cada generación. Bajo esta codificación/decodificación lo que se pretende es garantizar una búsqueda diversificada en el espacio de solución.

Codificación

La codificación de cada individuo emplea una cadena binaria (cromosoma) de longitud finita l , donde cada 0 ó 1 representa la presencia o ausencia de una potencia de dos. Cada cromosoma está formado de la siguiente forma: Sean t_Q y t_r los vectores de longitud N , formados por las longitudes de cada cadena de ceros y unos, correspondientes a cada valor de Q_i y r_i respectivamente, así la cadena correspondiente al valor Q_i tiene longitud $t_Q(i)$ (donde $t_Q(i)$ es la i -ésima posición del vector t_Q). Finalmente se unen secuencialmente las cadenas correspondientes de cada Q_i y cada r_i , dando como resultado un cromosoma de longitud l , donde l es igual a la suma de las componentes de los vectores t_Q y t_r . Así en la población inicial se crearán M cromosomas de tamaño l compuestos por ceros y unos generados aleatoriamente con una distribución uniforme.

Por ejemplo, si se tienen tres artículos y los valores iniciales propuestos para Q_1, Q_2, Q_3, r_1, r_2 y r_3 son 40, 120, 15, 40, 180, 100, 30, y 10 respectivamente (obtenidos del algoritmo basado en multiplicadores de Lagrange), entonces $t_Q = [6, 7, 4, 6]$ y $t_r = [8, 7, 5, 4]$, de donde el tamaño del cromosoma está dado por $l = 6 + 7 + 4 + 6 + 8 + 7 + 5 + 4 = 47$ por lo tanto uno de los cromosomas que podría aparecer codificando uno de los individuo de la población sería el siguiente:

[0 0 0 1 1 1 0 1 0 1 1 0 0 1 1 1 0 0 0 1 1 1 0 1 0 1 1 0 0 1 1 1 0 0 0 1 1 1 0 1 0 1 1 0 0 1 1]

Para decodificar cada cromosoma, primero se decodifican las cadenas correspondientes a cada Q_i

y luego las cadenas correspondientes a cada r_i , convirtiendo el número de base dos a base decimal.

Función objetivo

Como fue descrito anteriormente la función objetivo que describe exactamente el promedio de inventario a mano, según [33] está dada por:

$$\sum_{i=1}^N \frac{Q_i + 1}{2} + r_i - \theta_i + B(Q_i, r_i) \quad (4.1)$$

Donde $B(Q_i, r_i)$ es el número esperado de pedido pendiente en cualquier punto en el tiempo, para el artículo i .

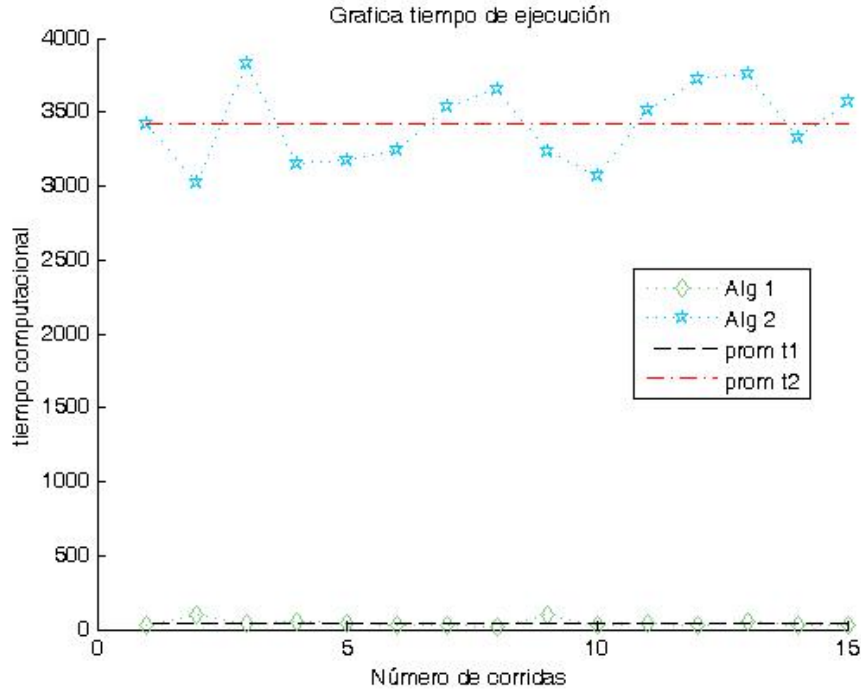
En vista a que uno de los propósitos en el presente trabajo, es comparar la solución obtenida empleando Algoritmos Genéticos con la obtenida mediante Multiplicadores de Lagrange y con el OptQuest, la función objetivo será tratada de dos formas, en la primera, se considerará la función objetivo sin el término que representa el pedido pendiente, por lo tanto, la función objetivo en este caso estaría dada por:

$$\sum_{i=1}^N \frac{Q_i + 1}{2} + r_i - \theta_i$$

Esto con el fin de garantizar que la función objetivo sea convexa y poder utilizar los multiplicadores de Lagrange.

La segunda manera de tratar la función, es cuando se permite que el término que representa el pedido pendiente aparezca en la función. Para este caso nuestro medio de comparación es el OptQuest.

Sin embargo al correr el programa implementado a través de Algoritmos Genéticos, se encontró que cuando las soluciones están muy alejadas del óptimo, puede suceder que la diferencia $r_i - \theta_i$ sea muy grande con $r_i < \theta_i$, lo cual ocasiona que el nivel de inventario a mano tome un valor negativo, generando con esto que la función objetivo tome como óptimo un individuo cuyo costo sea negativo, es por esto que inicialmente se restringió la búsqueda hasta obtener valores de r_i que hicieran que el valor del inventario a mano tomara sólo valores positivos, encontrándose que el tiempo requerido para hallar dichos valores es muy grande, como se puede ver las siguientes figuras:



Como se puede observar de la gráfica de corridas versus tiempo computacional, el promedio de tiempo es muy alto, usando el algoritmo en el cual, se toman los individuos que tienen costos totales mayores que cero (Alg 2), comparado con el promedio de tiempo usando el algoritmo donde se toman todos los individuos calculando su costos de tal manera que si $r - \theta > 0$ no hay pedidos pendientes y si $r - \theta < 0$ no se involucra esta diferencia en el costo (Alg 1).

En tanto que si se observa la gráfica de Corridas versus costos, vemos que los promedios de los costos de ambos algoritmos no están muy alejados. Por lo tanto se tomó la opción del algoritmo 1.

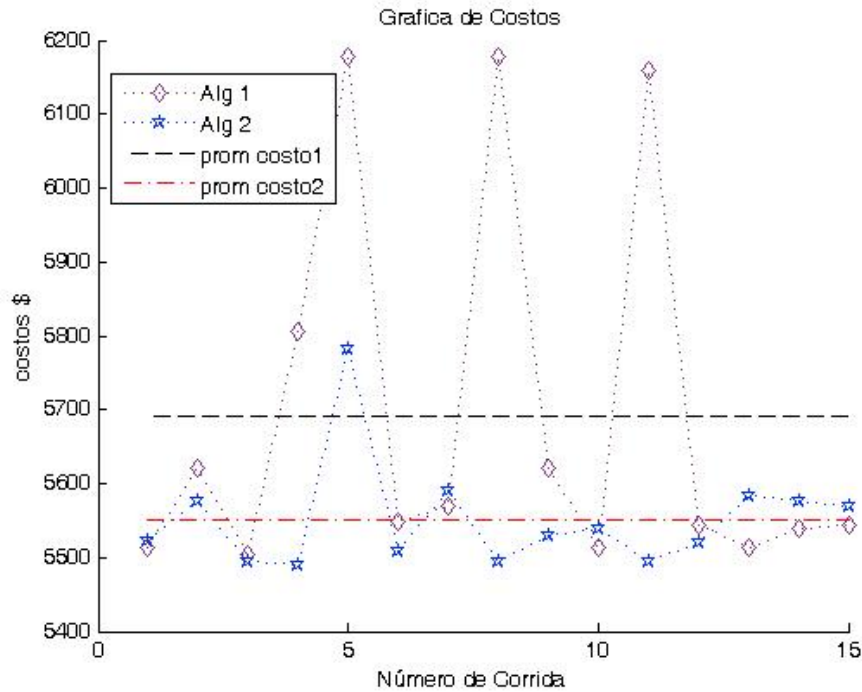
Así, para el caso en el que hayan pedidos pendientes, de acuerdo con [33] es útil ignorar el término $r - \theta$ en la función objetivo, cuando $r_i - \theta_i$ es negativo, ya que sin dicho término se tiene una fórmula simple y además se encontró que el valor de $r - \theta$ es muy pequeño cuando se está muy cerca a la solución óptima y por lo tanto no afecta mucho la calidad de la solución. De lo anterior el valor de adaptación del artículo i utilizando Algoritmos Genéticos esta dada por:

$$\begin{cases} \frac{Q_i+1}{2} + r_i - \theta_i & \text{si } r_i \geq \theta_i \\ \frac{Q_i+1}{2} + B_i(Q_i, r_i) & \text{si } r_i < \theta_i \end{cases}$$

Como se mencionó en la sección anterior, una de las principales dificultades que se puede presentar a la hora de definir la función de Adaptación, es la generación de individuos no factibles durante la evolución. Para penalizar dichos individuos se optó por usar

$$1. \lambda(F_t - F) \quad \text{y} \quad 2. \mu(S - S_t)$$

donde λ y μ son los valores de los multiplicadores de lagrange obtenidos en el algoritmo del método de relajación lagrangiana. Si la restricción de la satisfacción no se cumple, entonces al costo la sumamos $\mu(S - S_t)$ y si la restricción de la frecuencia no se cumple, entonces al costo la



sumamos $\lambda(F_t - F)$.

Población inicial

Como se mencionó anteriormente la población inicial de individuos, puede ser generada aleatoriamente o utilizando alguna heurística de construcción y su tamaño puede permanecer constante o variar de población a población. En la implementación propuesta para el algoritmo, la población inicial será generada aleatoriamente, esto con el fin de garantizar diversidad en población.

Para generar la población inicial lo primero es definir el tamaño de la población, es decir un valor adecuado para M que garantice cubrir adecuadamente el espacio de búsqueda sin conducir a excesivos costos computacionales. En la implementación propuesta se trabajó con un valor de 100 individuos, dado que según [41] éste es el valor con el que trabaja internamente el software de optimización optQuest, el cual es nuestro medio de comparación cuando aparece la función de pedido pendiente.

Conocido el valor del tamaño de la población M se realiza un proceso iterativo en el cual se crea de manera aleatoria y de una en una hasta completar M , una cadena de tamaño l formada por unos y ceros (cromosoma), con los cuales y a medida que se van creando se va formando un vector de valores en base decimal correspondientes a los Q_i , y otro vector con los valores correspondientes a los r_i , también en base decimal, luego se analiza si los valores correspondientes a los Q_i son diferentes de cero, si se cumplen las condiciones anteriores, la cadena de ceros y unos es almacenada en la matriz que representa la población inicial, de lo contrario, para el valor o valores respectivos de Q_i iguales a cero se les genera aleatoriamente una cadena de longitud igual al número de bits de ceros y unos que representa dicho Q_i hasta que se obtenga una representación diferente de cero. Luego se almacena en la matriz y se van generando aleatoriamente nuevos

elementos a los cuales se le hace el mismo proceso descrito anteriormente hasta obtener los M elementos de la población inicial. Para la formación de cada una de las generaciones subsiguientes se toman los individuos mejor adaptados, de entre padres e hijos.

Operadores Genéticos

En los siguientes párrafos serán descritos los operadores elegidos para utilizar en la programación propuesta de Algoritmos Genéticos, así como también los valores de probabilidad de cruce y mutación. Se ajustaron cuatro parámetros: Métodos de cruce y mutación, Probabilidades de cruce y mutación.

Operador de Selección:

Con el fin de preseleccionar los individuos que harán parte de la nueva población, se utilizará el procedimiento basado en el rango, dado que es un procedimiento que garantiza una selección muy uniforme de los individuos, permitiendo con esto que todos los individuos tengan probabilidad de ser seleccionados.

Bajo dicho procedimiento, los elementos se ordenan de menor a mayor de acuerdo al valor de la función objetivo f , así, el mejor individuo tiene rango M mientras que el peor individuo tiene rango 1. Dicha selección se hace de manera aleatoria, sin embargo, los individuos mejor adaptados tienen más probabilidad de sobrevivir, aunque gracias a la aleatoriedad y uniformidad en la selección, todos los individuos tienen probabilidad de ser seleccionados.

Operador de cruce:

Método de cruce: Los métodos de Cruce programados fueron: mono punto, dos puntos, tres puntos, cuatro puntos, uniforme, segmentado. Estos algoritmos se ejecutaron haciendo 30 repeticiones cada uno, desde 10 hasta 190 generaciones aumentando de a 10 generaciones; en las siguientes tablas se disponen los promedios y varianzas de los resultados de estos algoritmos para cada una de las generaciones correspondientes.

Método de Cruce monopunto.		
Nº generaciones	Promedio	Varianza
10	6001.1	53118
20	5729.6	45302
30	5713.7	59632
40	5631.7	29538
50	5649.3	39909
60	5732.7	51576
70	5700.7	55512
80	5736.8	80887
90	5652.5	36040
100	5579.8	14762
110	5692.3	65319
120	5634.5	30584
130	5660.6	59895
140	5663.6	59542
150	5698.9	63389
160	5661.8	42680
170	5618.8	43111
180	5599.1	17852
190	5700	76437

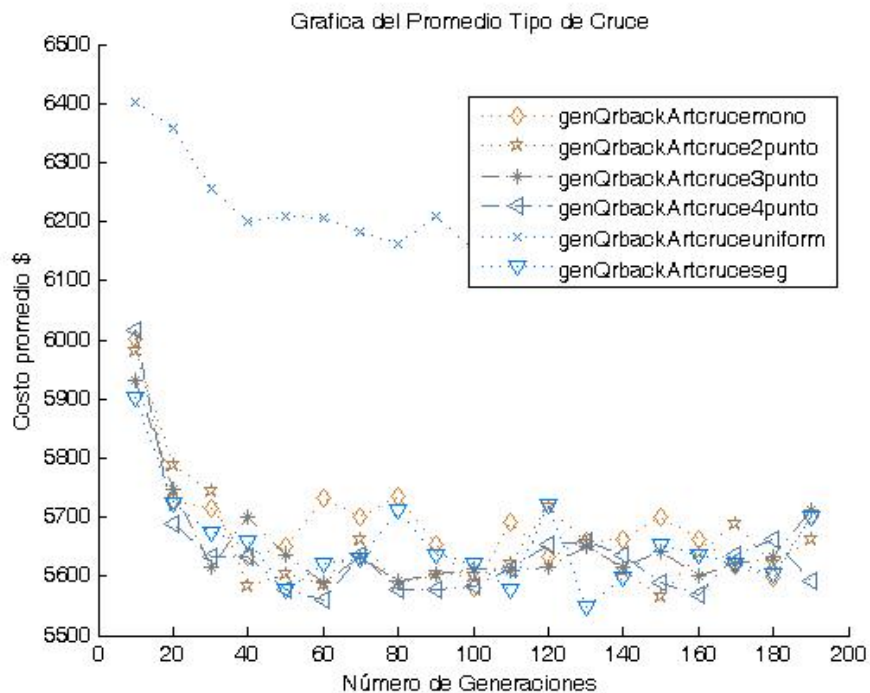
Método de Cruce de dos puntos.		
Nº generaciones	Promedio	Varianza
10	5981.6	56444
20	5787.8	64117
30	5743.1	65028
40	5583.1	17055
50	5604.3	31320
60	5585.6	17172
70	5660.8	47636
80	5585.2	19796
90	5604.5	21179
100	5601.8	24199
110	5621.8	29690
120	5719	78864
130	5656.1	58790
140	5610.2	36513
150	5564.8	13267
160	5639.7	43849
170	5689.3	56531
180	5619.9	46554
190	5661.5	55721

Método de Cruce de tres puntos.		
Nº generaciones	Promedio	Varianza
10	5930.8	56215
20	5748.1	62178
30	5615.4	28048
40	5700	63255
50	5636.7	48958
60	5588.9	28894
70	5632.2	41046
80	5592.7	38290
90	5604.4	38770
100	5611.8	48966
110	608.3	39358
120	5615.9	42175
130	5651.8	48264
140	5615.1	39358
150	5642.1	49745
160	5600.8	41935
170	5619.9	43261
180	5634.2	48164
190	5711.7	80145

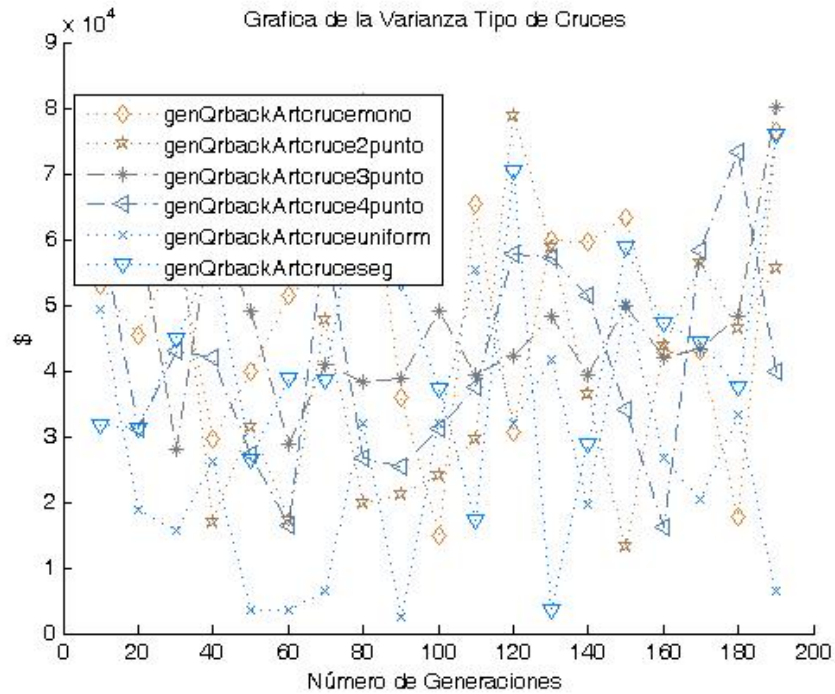
Método de Cruce de cuatro puntos.		
Nº generaciones	Promedio	Varianza
10	6015.1	61850
20	5689	31066
30	5632.8	42996
40	5632.2	41969
50	5578.9	27136
60	5560.2	16564
70	5634.6	60553
80	5578.9	26713
90	5578.7	25445
100	5584.6	31202
110	5613.7	37555
120	5653.4	57880
130	5659.1	57208
140	5635.7	51355
150	5590.4	34041
160	5567.5	16211
170	5637.3	58301
180	5662.1	73198
190	5593.4	39739

Método de Cruce uniforme.			Método de Cruce segmentado.		
Nº generaciones	Promedio	Varianza	Nº generaciones	Promedio	Varianza
10	6402.5	49433	10	5902.7	31809
20	6359.3	18747	20	5725	31197
30	6256.1	15740	30	5673.2	44870
40	6200.2	26259	40	5658.6	58130
50	6210.3	3589.5	50	5577.3	26496
60	6206.4	3481.4	60	5620.5	38902
70	6181.6	6536.4	70	5629.1	38477
80	6162	31853	80	5711.3	71835
90	6208.4	2609.9	90	5635.4	53552
100	6149.4	32063	100	5622.1	37179
110	6107	55394	110	5578.9	17306
120	6166.9	31991	120	5719.5	70266
130	6156.2	41812	130	5547	3564
140	6164.2	19632	140	5597.3	28794
150	6141.1	50029	150	5652.7	58858
160	6151.9	26640	160	5636.9	47337
170	6189.3	20369	170	5622.7	44388
180	6168.5	33281	180	5604.5	37395
190	6186.5	6322.4	190	5701.2	75867

Cuyas gráficas de promedios y varianzas respectivamente son:



De acuerdo a las pruebas de hipótesis realizadas por MATLAB, se tiene la siguiente información,



la cual depende del número de veces que cierto costo resultaba igual en 3 cifras significativas a otro costo.

El numero datos a comparar es: 2

warning: Algún conjunto de datos no es normal con nivel de significancia del 5%.

Los métodos genQrbackArtruceseg y genQrbackArtruce3punto son iguales en 3 cifras significativas para N° Gen= 10 sus varianzas son 31809.04453 y 56215.05905 respectivamente.

El numero datos a comparar es: 1

El método genQrbackArtruce4punto es el mejor numéricamente para N° Gen= 020.

El numero datos a comparar es: 2

warning: Algún conjunto de datos no es normal con nivel de significancia del 5%.

Los métodos genQrbackArtruce3punto y genQrbackArtruce4punto son iguales en 3 cifras significativas para N° Gen= 030 sus varianzas son 28048.13644 y 42996.44176 respectivamente.

El numero datos a comparar es: 1

El método genQrbackArtruce2punto es el mejor numéricamente para N° Gen= 040

El numero datos a comparar es: 3

warning: Algún conjunto de datos no es normal con nivel de significancia del 5%.

Los métodos genQrbackArtruceseg, genQrbackArtruce4punto y genQrbackArtruce2punto son iguales en 3 cifras significativas para N° Gen= 050 sus varianzas son 26495.65815, 27136.20690 y 31319.87874 respectivamente.

El numero datos a comparar es: 2

warning: Algún conjunto de datos no es normal con nivel de significancia del 5 %.

Los métodos genQrbackArtruce4punto y genQrbackArtruce2punto son iguales en 3 cifras significativas para N° Gen= 060 sus varianzas son 16564.34051 y 17172.28099 respectivamente.

El numero datos a comparar es: 3

warning: Algún conjunto de datos no es normal con nivel de significancia del 5 %.

Los métodos genQrbackArtruceseg, genQrbackArtruce3punto y genQrbackArtruce4punto son iguales en 3 cifras significativas para N° Gen= 070 sus varianzas son 38476.51670, 41045.56138 y 60552.85159 respectivamente.

El numero datos a comparar es: 3

warning: Algún conjunto de datos no es normal con nivel de significancia del 5 %.

Los métodos genQrbackArtruce4punto, genQrbackArtruce2punto y genQrbackArtruce3punto son iguales en 3 cifras significativas para N° Gen= 080 sus varianzas son 26713.33761, 19795.67693 y 38289.92912 respectivamente.

El numero datos a comparar es: 3

warning: Algún conjunto de datos no es normal con nivel de significancia del 5 %.

Los métodos genQrbackArtruce4punto, genQrbackArtruce3punto y genQrbackArtruce2punto son iguales en 3 cifras significativas para N° Gen= 090 sus varianzas son 25444.77744 , 38770.23313 y 21178.75638 respectivamente.

El numero datos a comparar es: 3

warning: Algún conjunto de datos no es normal con nivel de significancia del 5 %.

Los métodos genQrbackArtrucemono, genQrbackArtruce4punto y genQrbackArtruce2punto son iguales en 3 cifras significativas para N° Gen= 100 sus varianzas son 14761.78925 , 31201.93601 y 24198.97102 respectivamente.

El numero datos a comparar es: 1

El método genQrbackArtruceseg es el mejor numéricamente para N° Gen= 110

El numero datos a comparar es: 2

warning: Algún conjunto de datos no es normal con nivel de significancia del 5 %.

Los métodos genQrbackArtruce3punto y genQrbackArtrucemono son iguales en 3 cifras significativas para N° Gen= 120 sus varianzas son 42175.08227 y 30583.97133 respectivamente.

El numero datos a comparar es: 1

El método genQrbackArtruceseg es el mejor numéricamente para N° Gen= 130

El numero datos a comparar es: 3

warning: Algún conjunto de datos no es normal con nivel de significancia del 5 %.

Los métodos genQrbackArtruceseg, genQrbackArtruce2punto y genQrbackArtruce3punto son iguales en 3 cifras significativas para N° Gen= 140 sus varianzas son 28793.85639 , 36513.00658 y 39357.68099 respectivamente.

El numero datos a comparar es: 2

warning: Algún conjunto de datos no es normal con nivel de significancia del 5 %.

Los métodos `genQrbackArtcruce2punto` y `genQrbackArtcruce4punto` son iguales en 3 cifras significativas para N° Gen= 150 sus varianzas son 13266.60038 y 34040.93855 respectivamente.

El numero datos a comparar es: 1

El método `genQrbackArtcruce4punto` es el mejor numéricamente para N° Gen= 160

El numero datos a comparar es: 4

warning: Algún conjunto de datos no es normal con nivel de significancia del 5 %.

Los métodos `genQrbackArtcrucemono`, `genQrbackArtcruce3punto`, `genQrbackArtcruceseg` y `genQrbackArtcruce4punto` son iguales en 3 cifras significativas para N° Gen= 170 sus varianzas son 43110.71138 , 43260.73917 , 44388.16453 y 58300.50662 respectivamente.

El numero datos a comparar es: 3

warning: Algún conjunto de datos no es normal con nivel de significancia del 5 %.

Los métodos `genQrbackArtcrucemono`, `genQrbackArtcruceseg` y `genQrbackArtcruce2punto` son iguales en 3 cifras significativas para N° Gen= 180 sus varianzas son 17852.46474 , 37394.78266 y 46554.04691 respectivamente.

El numero datos a comparar es: 1

El método `genQrbackArtcruce4punto` es el mejor numéricamente para N° Gen= 190.

De acuerdo a la información anterior, en los casos que hubo elementos a comparar ocurrió que algún conjunto de datos no es normal con nivel de significancia del 5%, por consiguiente, se analizan las varianzas y se toma el de menor varianza, esto se hizo analizando a partir de 100 generaciones (dado que el ajuste se fue en base a un problema de cuatro productos, el cual converge, alrededor de 100 generaciones), de lo anterior se tiene que:

- cruce mono punto fue el mejor 4 veces
- cruce segmentado punto fue el mejor 3 veces
- cruce 4 puntos fue el mejor 2 veces
- cruce 2 puntos fue el mejor 1 vez

Por lo tanto, se eligió el método cruce mono punto.

Probabilidad de Cruce. Los valores de Probabilidad de cruce programados fueron: 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9 y 0.95. Estos algoritmos se ejecutaron haciendo 30 repeticiones cada uno, desde 10 hasta 190 generaciones aumentando de a 10 generaciones; en las siguientes tablas se disponen los promedios y varianzas de los resultados de estos algoritmos para cada una de las generaciones correspondientes.

Probabilidad 0.6		
N° generaciones	Promedio	Varianza
10	6044.6	93522
20	5751.9	40537
30	5665.6	37485
40	5673.8	51953
50	5621.7	17574
60	5661.4	35283
70	5654.1	44407
80	5616.3	27412
90	5608.1	27267
100	5616.4	20002
110	5701.3	67265
120	5644.3	38678
130	5637.4	26092
140	5603.6	29245
150	5622.7	20644
160	5646.2	37383
170	5607.2	16908
180	5668	53024
190	5644.4	47901

Probabilidad 0.65		
N° generaciones	Promedio	Varianza
10	5973.2	67247
20	5774.4	48358
30	5666	45704
40	5659.2	53870
50	5615.3	28162
60	5653.7	48231
70	5696.9	64906
80	5575.8	18966
90	5619.4	30683
100	5655.2	47906
110	5627.8	40056
120	5662.4	48716
130	5632.1	30032
140	5647	50167
150	5702.4	67804
160	5678.6	52644
170	5693.7	68241
180	5627.6	31288
190	5614.8	20625

Probabilidad 0.7		
N° generaciones	Promedio	Varianza
10	5915.2	35534
20	5700.8	15347
30	5682.7	42057
40	5699.4	61604
50	5633.5	41993
60	5643.9	47872
70	5706.7	62094
80	5595	18201
90	5616	28562
100	5739.5	62785
110	5701.1	48044
120	5632.6	26719
130	5666.1	52095
140	5630	27742
150	5691.5	58144
160	5600.9	28298
170	5645.8	38022
180	5585.6	18870
190	5691.1	56173

Probabilidad 0.75		
N° generaciones	Promedio	Varianza
10	5973.3	56505
20	5743	33685
30	5695.5	46774
40	5670	59595
50	5613.5	19342
60	5654.8	42595
70	5643.2	39364
80	5680.8	66922
90	5644.9	38039
100	5637.7	41706
110	5694.5	69048
120	5635.9	40655
130	5667.3	49729
140	5657.8	49136
150	5675.9	50664
160	5591	27369
170	5634.2	41315
180	5703.2	75557
190	5673.4	53302

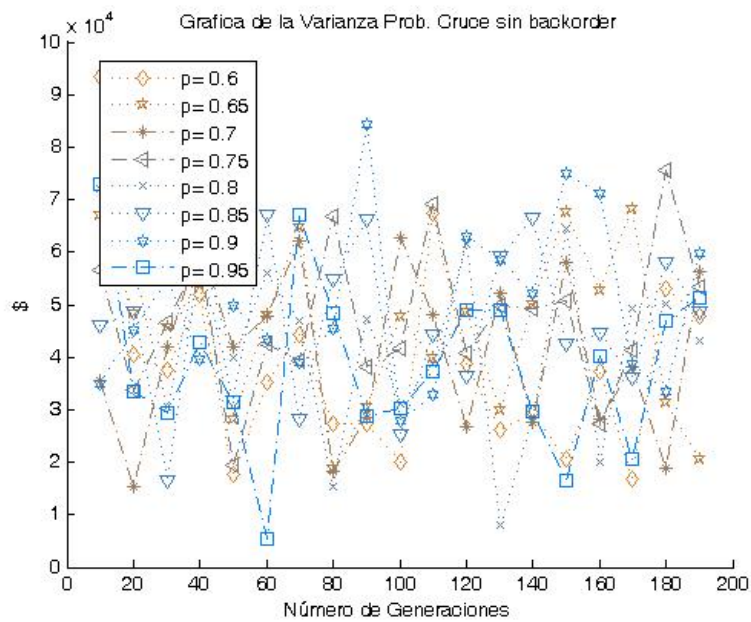
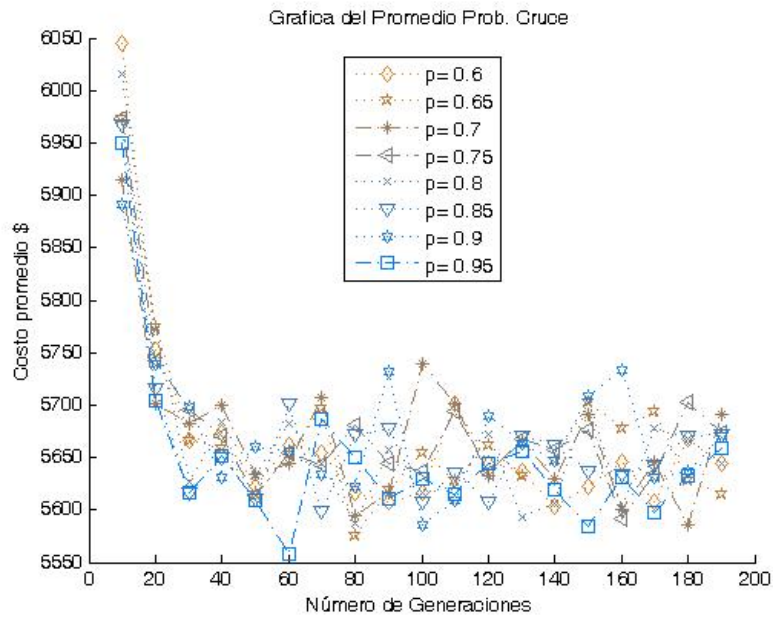
Probabilidad 0.8		
N° generaciones	Promedio	Varianza
10	6016.1	72121
20	5771	57273
30	5627.3	30600
40	5683.6	61399
50	5632.8	39992
60	5682	56035
70	5644.6	46867
80	5587.1	15433
90	5656.9	47126
100	5616.5	31268
110	5622.6	38709
120	5671.5	61454
130	5593.1	8081.1
140	5606.3	28214
150	5706.4	64343
160	5601.2	19929
170	5677.9	49292
180	5667.3	50225
190	5644.2	43135

Probabilidad 0.85		
N° generaciones	Promedio	Varianza
10	5967.4	45909
20	5715.3	48824
30	5615.9	16635
40	5649.7	59336
50	5611.4	31752
60	5701.2	67159
70	5598.6	28357
80	5671.4	54742
90	5678.1	66312
100	5608	25409
110	5635.5	44261
120	5607.1	36518
130	5670.4	59272
140	5661.6	66661
150	5637.6	42594
160	5633.9	44612
170	5643	35969
180	5670.1	58005
190	5672.1	48769

Probabilidad 0.9		
N° generaciones	Promedio	Varianza
10	5891.9	35066
20	5738.9	45128
30	5698.5	54840
40	5630.9	39649
50	5659.7	49938
60	5655.8	43556
70	5634	39102
80	5621.9	45426
90	5731.3	84234
100	5585.8	27934
110	5609.2	33019
120	5689.7	63154
130	5662.8	58616
140	5647.3	52317
150	5708.9	75036
160	5733.1	71187
170	5630.9	38810
180	5637.1	33573
190	5670.6	59696

Probabilidad 0.95		
N° generaciones	Promedio	Varianza
10	5949.8	72866
20	5704.2	33373
30	5615.8	29497
40	5652.1	42919
50	5608.7	31431
60	5558.7	5456.9
70	5687.4	67183
80	5649.9	48422
90	5610.1	28873
100	5630.2	30260
110	5614.8	37222
120	5644.1	48922
130	5656	49026
140	5619.9	29808
150	5584	16540
160	5631	40087
170	5596.8	20669
180	5633.2	46945
190	5658.7	51402

Cuyas gráficas de promedios y varianzas respectivamente son:



De acuerdo a las pruebas de hipótesis realizadas por MATLAB, se tiene la siguiente información:

El numero datos a comparar es: 2

warning: Algún conjunto de datos no es normal con nivel de significancia del 5 %.

Las probabilidades $p= 0.9$ y $p= 0.7$ son iguales en 3 cifras significativas para N° Gen= 010 sus varianzas son 35065.74296 y 35533.62959 respectivamente.

El numero datos a comparar es: 3

warning: Algún conjunto de datos no es normal con nivel de significancia del 5 %.

Las probabilidades $p= 0.7$, $p= 0.95$ y $p= 0.85$ son iguales en 3 cifras significativas para N° Gen= 020 sus varianzas son 15347.14118 , 33373.06505 y 48824.48608 respectivamente.

El numero datos a comparar es: 3

warning: Algún conjunto de datos no es normal con nivel de significancia del 5 %.

Las probabilidades $p= 0.95$, $p= 0.85$ y $p= 0.8$ son iguales en 3 cifras significativas para N° Gen= 030 sus varianzas son 29496.79905 , 16635.37778 y 30600.13244 respectivamente.

El numero datos a comparar es: 3

warning: Algún conjunto de datos no es normal con nivel de significancia del 5 %.

Las probabilidades $p= 0.9$, $p= 0.85$ y $p= 0.95$ son iguales en 3 cifras significativas para N° Gen= 040 sus varianzas son 39648.82870 , 59336.38225 y 42918.61664 respectivamente.

El numero datos a comparar es: 7

warning: Algún conjunto de datos no es normal con nivel de significancia del 5 %.

Las probabilidades $p= 0.95$, $p= 0.85$, $p= 0.75$, $p= 0.65$, $p= 0.6$, $p= 0.8$ y $p= 0.7$ son iguales en 3 cifras significativas para N° Gen= 050 sus varianzas son 31431.31706 , 31751.53842 , 19341.59029 , 28162.41142 , 17573.92554 , 39991.63586 y 41993.12140 respectivamente.

El numero datos a comparar es: 1

La probabilidad $p= 0.95$ es el mejor numéricamente para N° Gen= 060

El numero datos a comparar es: 1

La probabilidad $p= 0.85$ es el mejor numéricamente para N° Gen= 070

El numero datos a comparar es: 3

warning: Algún conjunto de datos no es normal con nivel de significancia del 5 %.

Las probabilidades $p= 0.65$, $p= 0.8$ y $p= 0.7$ son iguales en 3 cifras significativas para N° Gen= 080 sus varianzas son 18966.16364 , 15433.40308 y 18200.65202 respectivamente.

El numero datos a comparar es: 4

warning: Algún conjunto de datos no es normal con nivel de significancia del 5 %.

Las probabilidades $p= 0.6$, $p= 0.95$, $p= 0.7$ y $p= 0.65$ son iguales en 3 cifras significativas para N° Gen= 090 sus varianzas son 27266.88344, 28872.51587, 28562.48659 y 30682.96620 respectivamente.

El numero datos a comparar es: 2

warning: Algún conjunto de datos no es normal con nivel de significancia del 5 %.

Las probabilidades $p= 0.9$ y $p= 0.85$ son iguales en 3 cifras significativas para N° Gen= 100 sus varianzas son 27934.32826 y 25409.05909 respectivamente.

El numero datos a comparar es: 5

warning: Algún conjunto de datos no es normal con nivel de significancia del 5 %.

Las probabilidades $p= 0.9$, $p= 0.95$, $p= 0.8$, $p= 0.65$ y $p= 0.85$ son iguales en 3 cifras significativas para N° Gen= 110 sus varianzas son 33018.55237, 37221.78827, 38708.61095, 40056.32315 y 44261.43494 respectivamente.

El numero datos a comparar es: 2

warning: Algún conjunto de datos no es normal con nivel de significancia del 5 %.

Las probabilidades $p= 0.85$ y $p= 0.7$ son iguales en 3 cifras significativas para N° Gen= 120 sus varianzas son 36518.02082 y 26719.37954 respectivamente.

El numero datos a comparar es: 1

La probabilidad $p= 0.8$ es el mejor numéricamente para N° Gen= 130.

El numero datos a comparar es: 4

warning: Algún conjunto de datos no es normal con nivel de significancia del 5 %.

Las probabilidades $p= 0.6$, $p= 0.8$, $p= 0.95$ y $p= 0.7$ son iguales en 3 cifras significativas para N° Gen= 140 sus varianzas son 29245.31754, 28214.11043, 29807.54222 y 27741.56316 respectivamente.

El numero datos a comparar es: 1

La probabilidad $p= 0.95$ es el mejor numéricamente para N° Gen= 150.

El numero datos a comparar es: 3

warning: Algún conjunto de datos no es normal con nivel de significancia del 5 %.

Las probabilidades $p= 0.75$, $p= 0.7$ y $p= 0.8$ son iguales en 3 cifras significativas para N° Gen= 160 sus varianzas son 27369.19401, 28297.97495 y 19928.52666 respectivamente.

El numero datos a comparar es: 2

warning: Algún conjunto de datos no es normal con nivel de significancia del 5 %.

Las probabilidades $p= 0.95$ y $p= 0.6$ son iguales en 3 cifras significativas para N° Gen= 170 sus varianzas son 20669.31754 y 16907.70167 respectivamente.

El numero datos a comparar es: 1

La probabilidad $p= 0.7$ es el mejor numéricamente para N° Gen= 180.

El numero datos a comparar es: 1

La probabilidad $p= 0.65$ es el mejor numéricamente para N° Gen= 190.

De acuerdo a la información anterior, en los casos que hubo elementos a comparar ocurrió que

algún conjunto de datos no es normal con nivel de significancia del 5%, por consiguiente, se analizan las varianzas y se toma el de menor varianza, esto se hizo analizando a partir de 100 generaciones (dado que el ajuste se fue en base a un problema de cuatro productos, el cual converge, alrededor de 100 generaciones), de lo anterior se tiene que:

- Probabilidad de cruce 0.7 fue la mejor 3 veces
- Probabilidad de cruce 0.8 fue la mejor 2 veces
- Probabilidad de cruce 0.6 fue la mejor 1 vez
- Probabilidad de cruce 0.65 fue la mejor 1 vez
- Probabilidad de cruce 0.85 fue la mejor 1 vez
- Probabilidad de cruce 0.9 fue la mejor 1 vez
- Probabilidad de cruce 0.95 fue la mejor 1 vez

Por lo tanto se eligió Probabilidad de cruce 0.7.

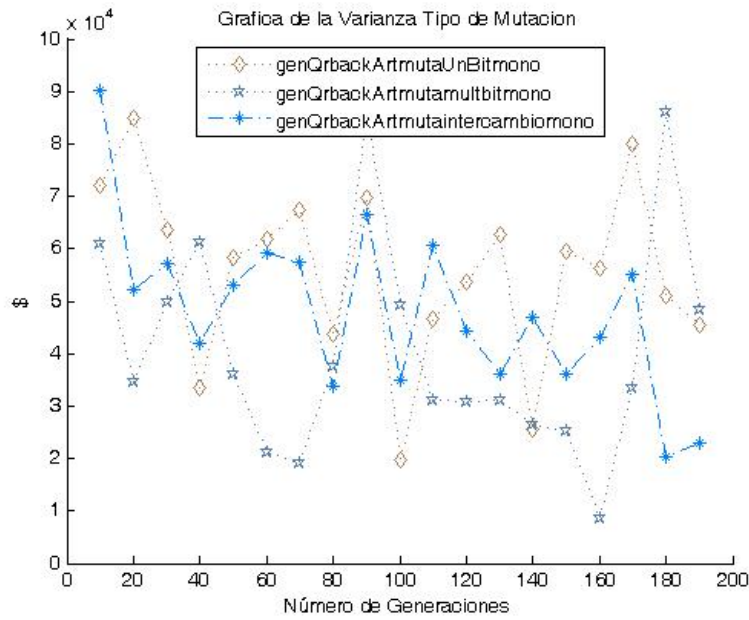
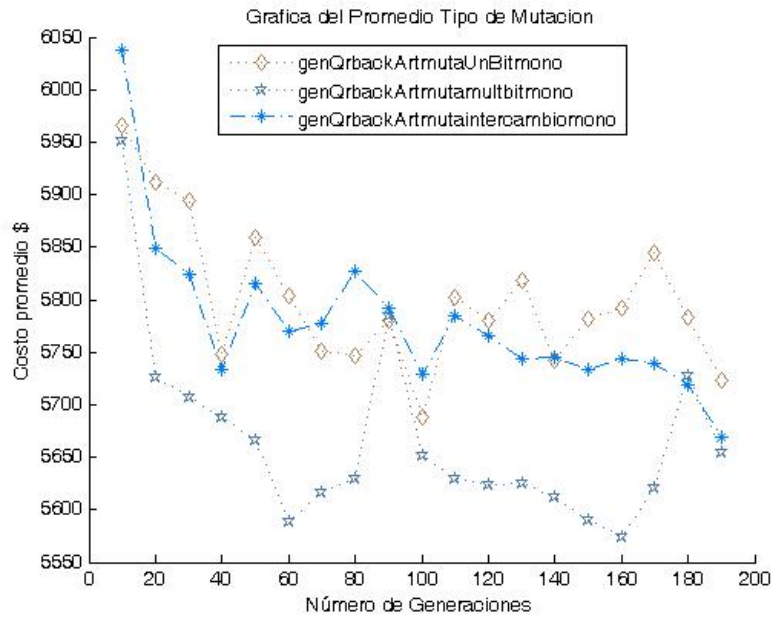
Operador de Mutación:

Método de mutación: Los métodos de mutación programados fueron: un bit, multibit e intercambio. Estos algoritmos se ejecutaron haciendo 30 repeticiones cada uno, desde 10 hasta 190 generaciones aumentando de a 10 generaciones; en las siguientes tablas se disponen los promedios y varianzas de los resultados de estos algoritmos para cada una de las generaciones correspondientes.

Método de Mutación de un bit.			Método de Mutación multibit.		
N° generaciones	Promedio	Varianza	N° generaciones	Promedio	Varianza
10	5965.3	72133	10	5950.8	60942
20	5911.9	85044	20	5725.7	34689
30	5894.9	63633	30	5707.8	49888
40	5748.7	33364	40	5687.6	61353
50	5859.3	58193	50	5666.3	36181
60	5803	61891	60	5589.2	21239
70	5751.6	67416	70	5616.7	19212
80	5746.9	43706	80	5630	37524
90	5780.9	69805	90	5784.7	85202
100	5687.6	19882	100	5650.9	49195
110	5802.2	46621	110	5629	31111
120	5779.9	53765	120	5623.9	30884
130	5818.6	62665	130	5625.1	31098
140	5741.8	25442	140	5612.3	26545
150	5782.3	59570	150	5590.2	25228
160	5791.6	56377	160	5573.8	8588.4
170	5844.3	80100	170	5620.6	33424
180	5783.4	51169	180	5727.7	86121
190	5723.2	45371	190	5654.3	48376

Método de Mutación por intercambio.		
N° generaciones	Promedio	Varianza
10	6037.4	90070
20	5849.1	52065
30	5823.8	57182
40	5733.1	41892
50	5815.2	53113
60	5770.5	59130
70	5776.6	57519
80	5827.1	33869
90	5791.5	66456
100	5729.2	34931
110	5784.7	60605
120	5765.5	44353
130	5744.1	36194
140	5745.1	46830
150	5733.5	36241
160	5743.1	43178
170	5739.3	55212
180	5719	20281
190	5669.6	22847

Cuyas gráficas de promedios y varianzas respectivamente son:



De acuerdo a las pruebas de hipótesis realizadas por MATLAB, se tiene la siguiente información:

El numero datos a comparar es: 2

warning: Algún conjunto de datos no es normal con nivel de significancia del 5 %.

Los métodos genQrbackArmutamultbitmono y genQrbackArmutaUnBitmono son iguales en 3 cifras significativas para N° Gen= 010 sus varianzas son 60942.13799 y 72133.11322 respectivamente.

El numero datos a comparar es: 1

El método genQrbackArmutamultbitmono es el mejor numéricamente para N° Gen= 020.

El numero datos a comparar es: 1

El método genQrbackArmutamultbitmono es el mejor numéricamente para N° Gen= 030.

El numero datos a comparar es: 1

El método genQrbackArmutamultbitmono es el mejor numéricamente para N° Gen= 040.

El numero datos a comparar es: 1

El método genQrbackArmutamultbitmono es el mejor numéricamente para N° Gen= 050.

El numero datos a comparar es: 1

El método genQrbackArmutamultbitmono es el mejor numéricamente para N° Gen= 060.

El numero datos a comparar es: 1

El método genQrbackArmutamultbitmono es el mejor numéricamente para N° Gen= 070.

El numero datos a comparar es: 1

El método genQrbackArmutamultbitmono es el mejor numéricamente para N° Gen= 080.

El numero datos a comparar es: 3

warning: Algún conjunto de datos no es normal con nivel de significancia del 5 %.

Los métodos genQrbackArmutaUnBitmono, genQrbackArmutamultbitmono y genQrbackArmutaintercambiomonos son iguales en 3 cifras significativas para N° Gen= 090 sus varianzas son 69805.19278 , 85201.88654 y 66455.54596 respectivamente.

El numero datos a comparar es: 1

El método genQrbackArmutamultbitmono es el mejor numéricamente para N° Gen= 100.

El numero datos a comparar es: 1

El método genQrbackArmutamultbitmono es el mejor numéricamente para N° Gen= 110.

El numero datos a comparar es: 1

El método genQrbackArmutamultbitmono es el mejor numéricamente para N° Gen= 120.

El numero datos a comparar es: 1

El método genQrbackArtmutamultbitmono es el mejor numéricamente para N° Gen= 130.

El numero datos a comparar es: 1

El método genQrbackArtmutamultbitmono es el mejor numéricamente para N° Gen= 140.

El numero datos a comparar es: 1

El método genQrbackArtmutamultbitmono es el mejor numéricamente para N° Gen= 150.

El numero datos a comparar es: 1

El método genQrbackArtmutamultbitmono es el mejor numéricamente para N° Gen= 160.

El numero datos a comparar es: 1

El método genQrbackArtmutamultbitmono es el mejor numéricamente para N° Gen= 170.

El numero datos a comparar es: 2

warning: Algún conjunto de datos no es normal con nivel de significancia del 5%.

Los métodos genQrbackArtmutaintercambiomonos y genQrbackArtmutamultbitmonos son iguales en 3 cifras significativas para N° Gen= 180 sus varianzas son 20280.90849 y 86120.63144 respectivamente.

El numero datos a comparar es: 2

warning: Algún conjunto de datos no es normal con nivel de significancia del 5%.

Los métodos genQrbackArtmutamultbitmonos y genQrbackArtmutaintercambiomonos son iguales en 3 cifras significativas para N° Gen= 190 sus varianzas son 48376.11155 y 22847.21243 respectivamente.

De acuerdo a la información anterior, en los casos que hubo elementos a comparar ocurrió que algún conjunto de datos no es normal con nivel de significancia del 5%, por consiguiente, se analizan las varianzas y se toma el de menor varianza, esto se hizo analizando a partir de 100 generaciones (dado que el ajuste se fue en base a un problema de cuatro productos, el cual converge, alrededor de 100 generaciones), de lo anterior se tiene que:

- Mutación multibit fue la mejor 8 veces
- Mutación intercambio fue la mejor 2 veces

Por lo tanto se eligió el método mutación multibit.

Probabilidad de Mutación. Para el parámetro Probabilidad de mutación se se hizo el ajuste tomando ciertos valores fijos de probabilidades que no dependen de la longitud del cromosoma. Los valores programados fueron: 0.02, 0.04, 0.06, 0.08 y 0.1. Estos algoritmos se ejecutaron haciendo 30 repeticiones cada uno, desde 10 hasta 190 generaciones aumentando de a 10 generaciones; en las siguientes tablas se disponen los promedios y varianzas de los resultados de estos algoritmos para cada una de las generaciones correspondientes.

Probabilidad 0.02		
N° generaciones	Promedio	Varianza
10	6010.1	76972
20	5774.6	80562
30	5649.2	35476
40	5595.2	15926
50	5682	60824
60	5631.7	37687
70	5570.9	4973.2
80	5612.4	17951
90	5672	60273
100	5676.7	67569
110	5639.1	39366
120	5719.3	77698
130	5619.7	35369
140	5733.1	78158
150	5621.8	31702
160	5641	40758
170	5627.1	26530
180	5650.9	41912
190	5679.4	60660

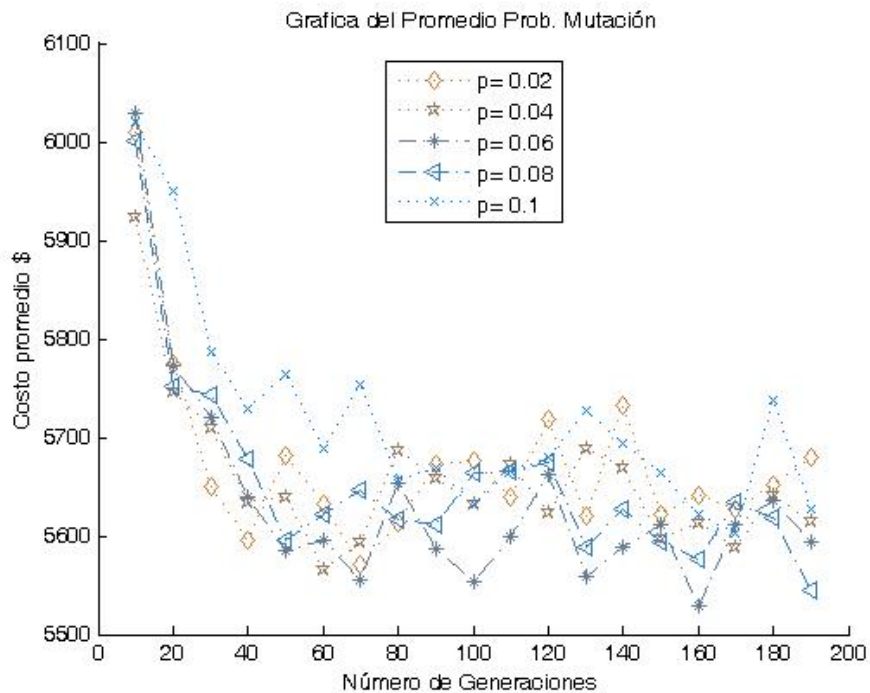
Probabilidad 0.04		
N° generaciones	Promedio	Varianza
10	5923.4	34809
20	5746.5	49303
30	5709.4	71365
40	5633.7	49496
50	5639.8	60854
60	5566.6	28545
70	5594.7	43813
80	5686.5	77201
90	5659.3	69073
100	5632.3	53098
110	5673.1	75579
120	5624.2	49940
130	5688.5	79575
140	5669.8	66649
150	5598.2	50018
160	5612.4	38773
170	5588	39612
180	5640.5	60243
190	5614.4	49337

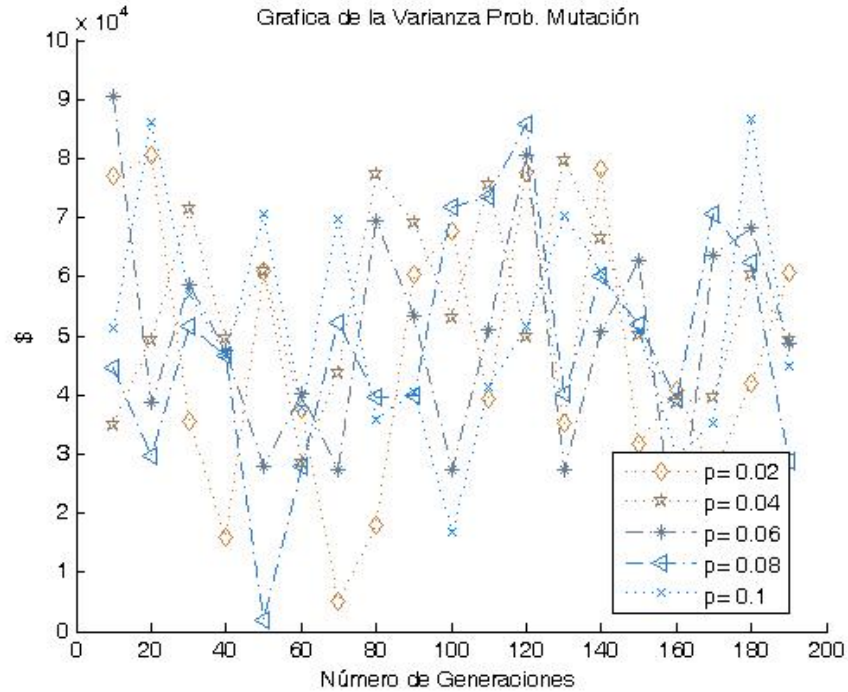
Probabilidad 0.06		
N° generaciones	Promedio	Varianza
10	6029.8	90604
20	5773.3	38790
30	5719.7	58696
40	5639.8	47623
50	5584.7	27825
60	5594.7	40228
70	5555.6	27333
80	5652.9	69339
90	5586.8	53394
100	5554.3	27257
110	5598.4	50951
120	5663.1	80526
130	5559.2	27396
140	5589.5	50863
150	5611.4	62696
160	5529.7	13678
170	5611.8	63727
180	5635.9	68315
190	5594.2	48581

Probabilidad 0.08		
N° generaciones	Promedio	Varianza
10	6001.2	44488
20	5751.4	29772
30	5743.4	51519
40	5678.6	46779
50	5595.4	1866.3
60	5622.8	27998
70	5646.3	52085
80	5616.3	39560
90	5610.7	39803
100	5664.1	71766
110	5666.3	73529
120	5674.3	85819
130	5587.8	40080
140	5627.2	60201
150	5594.1	51774
160	5576.2	39448
170	5633.6	70592
180	5619.2	62455
190	5545.4	28878

Probabilidad 0.1		
Nº generaciones	Promedio	Varianza
10	6019.6	51254
20	5949.3	86203
30	5787.4	56907
40	5728.9	46881
50	5764	70684
60	5688.6	37894
70	5754	69826
80	5656.9	35704
90	5668.7	40516
100	5632.2	16945
110	5666.5	41405
120	5679.4	51532
130	5726.5	70201
140	5694.2	61098
150	5663.8	50590
160	5621.4	29757
170	5603.2	35126
180	5737.7	86720
190	5627.7	44765

Cuyas gráficas de promedios y varianzas respectivamente son:





y de acuerdo a las pruebas de hipótesis realizadas por MATLAB, se tiene la siguiente información:

El numero datos a comparar es: 1

La probabilidad $p = 0.04$ es el mejor numéricamente para N° Gen= 010

El numero datos a comparar es: 4

warning: Algún conjunto de datos no es normal con nivel de significancia del 5%.

Las probabilidades $p = 0.04$, $p = 0.08$, $p = 0.06$ y $p = 0.02$ son iguales en 3 cifras significativas para N° Gen= 020 sus varianzas son 49302.56438, 29771.50536, 38790.43537 y 80561.99335 respectivamente.

El numero datos a comparar es: 1

La probabilidad $p = 0.02$ es el mejor numéricamente para N° Gen= 030

El numero datos a comparar es: 1

La probabilidad $p = 0.02$ es el mejor numéricamente para N° Gen= 040

El numero datos a comparar es: 2

warning: Algún conjunto de datos no es normal con nivel de significancia del 5%.

Las probabilidades $p = 0.06$ y $p = 0.08$ son iguales en 3 cifras significativas para N° Gen= 050 sus varianzas son 27824.97183 y 1866.25053 respectivamente.

El numero datos a comparar es: 1

La probabilidad $p = 0.04$ es el mejor numéricamente para N° Gen= 060

El numero datos a comparar es: 2

warning: Algún conjunto de datos no es normal con nivel de significancia del 5 %.

Las probabilidades $p= 0.06$ y $p= 0.02$ son iguales en 3 cifras significativas para N° Gen= 070 sus varianzas son 27333.08354 y 4973.15862 respectivamente.

El numero datos a comparar es: 2

warning: Algún conjunto de datos no es normal con nivel de significancia del 5 %.

Las probabilidades $p= 0.02$ y $p= 0.08$ son iguales en 3 cifras significativas para N° Gen= 080 sus varianzas son 17951.28203 y 39560.08881 respectivamente.

El numero datos a comparar es: 2

warning: Algún conjunto de datos no es normal con nivel de significancia del 5 %.

Las probabilidades $p= 0.06$ y $p= 0.08$ son iguales en 3 cifras significativas para N° Gen= 090 sus varianzas son 53393.59697 y 39802.91786 respectivamente.

El numero datos a comparar es: 1

La probabilidad $p= 0.06$ es el mejor numéricamente para N° Gen= 100

El numero datos a comparar es: 1

La probabilidad $p= 0.06$ es el mejor numéricamente para N° Gen= 110

El numero datos a comparar es: 1

La probabilidad $p= 0.04$ es el mejor numéricamente para N° Gen= 120

El numero datos a comparar es: 1

La probabilidad $p= 0.06$ es el mejor numéricamente para N° Gen= 130

El numero datos a comparar es: 1

La probabilidad $p= 0.06$ es el mejor numéricamente para N° Gen= 140

El numero datos a comparar es: 4

warning: Algún conjunto de datos no es normal con nivel de significancia del 5 %.

Las probabilidades $p= 0.08$, $p= 0.04$, $p= 0.06$ y $p= 0.02$ son iguales en 3 cifras significativas para N° Gen= 150 sus varianzas son 51773.92989, 50017.93257, 62695.98585 y 31701.66594 respectivamente.

El numero datos a comparar es: 1

La probabilidad $p= 0.06$ es el mejor numéricamente para N° Gen= 160

El numero datos a comparar es: 3

warning: Algún conjunto de datos no es normal con nivel de significancia del 5 %.

Las probabilidades $p= 0.04$, $p= 0.1$ y $p= 0.06$ son iguales en 3 cifras significativas para N° Gen= 170 sus varianzas son 39612.37194, 35126.44611 y 63726.94460 respectivamente.

El numero datos a comparar es: 3

warning: Algún conjunto de datos no es normal con nivel de significancia del 5 %.

Las probabilidades $p= 0.08$, $p= 0.06$ y $p= 0.04$ son iguales en 3 cifras significativas para N^o $Gen= 180$ sus varianzas son 62455.26085, 68314.99149 y 60243.03126 respectivamente.

El numero datos a comparar es: 1

La probabilidad $p= 0.08$ es el mejor numéricamente para N^o $Gen= 190$

De acuerdo a la información anterior, en los casos que hubo elementos a comparar ocurrió que algún conjunto de datos no es normal con nivel de significancia del 5 %, por consiguiente, se analizan las varianzas y se toma el de menor varianza. luego analizando a partir de 100 generaciones se tiene que

- Probabilidad de mutación 0.06 fue la mejor 5 veces
- Probabilidad de mutación 0.04 la mejor 2 veces
- Probabilidad de mutación 0.02 fue la mejor 1 vez
- Probabilidad de mutación 0.08 fue la mejor 1 vez
- Probabilidad de mutación 0.1 fue la mejor 1 vez

Por lo tanto se eligió Probabilidad de mutación 0.06.

4.4.2. Optimización-Simulación

En cuanto a la implementación para el problema (3.3), en que se consideran pedidos pendientes, tanto en el Simul8 como en el OptQuest la estructura del modelo fue en esencia la misma, la única diferencia radica en como es calculado el costo de inversión total, con este fin lo que se hizo fue cobrar al finalizar la simulación el costo por las unidades demandadas durante el tiempo de reposición y que no se lograron satisfacer del inventario existente en ese momento.

4.5. Comparación de resultados para el problema con pedidos pendientes

Al igual que en la sección correspondiente a la comparación de resultados para el problema 3.4, se presentan dos tablas discriminadas a manera de sistemas, las cuales reflejan los resultados obtenidos de las metodologías de Optimización-Simulación y Algoritmos Genéticos, en ellas se muestran los mismos items correspondientes a las tablas del problema (3.4).

En estas tablas se puede observar que el comportamiento de Algoritmos Genéticos Vs Optimización-Simulación, es similar en calidad al comportamiento del problema (3.4)

4.5. COMPARACIÓN DE RESULTADOS PARA EL PROBLEMA CON PEDIDOS PENDIENTES79

Resultados obtenidos de Algoritmos Genéticos						
Sistema	Núm Prod	S_t	F_t	II_i	$numiter$	$TiemCorr$
1	10	0.939029	13.8583	2067.72	329	22.7675915 seg
2	20	0.94996	14.0633	3796.5	1649	213.490743 seg
3	30	0.949738	13.5122	6093.64	2593	536.013405 seg
4	40	0.949372	13.0967	8507.92	4786	1432.90367 seg
5	50	0.949543	12.8323	10056.8	6689	2371.29648 seg
6	60	0.949533	12.0003	11722.5	9108	4073.88314 seg
7	70	0.944931	12.006	14465.7	10586	6311.83563 seg
8	80	0.947888	11.6125	16555.2	12721	9218.42154 seg
9	90	0.949926	11.5968	18286.3	14448	10650.5558 seg
10	100	0.949961	10.6535	20583.9	14978	14726.0587 seg

Resultados obtenidas de Optimización-Simulación						
Sistema	Núm Prod	S_t	F_t	II_i	$numsimul$	$TiemCorr$
1	10	0.9633635	4.17	3366.00209	16	1 min
2	20	0.9592869	7.28	3499.94515	45	4 min
3	30	0.9664139	2.63	12860.75286	97	9 min
4	40	0.9695722	2.41	20388.097	286	24 min
5	50	0.9593969	2.76	18753.121	506	40 min
6	60	0.9585519	3.03833	21110.502	809	68 min
7	70	0.9558886	3.01429	23304.935	1218	105 min
8	80	0.9568722	3.8925	23389.55	1678	154 min
9	90	0.9074966	1.46667	391475.5207	1868	178 min
10	100	0.9156793	1.447	367852.5868	2550	246 min

Conclusiones

En el presente trabajo de grado, fue desarrollado un algoritmo de Optimización basado en Algoritmos Genéticos con el fin de optimizar el problema asociado al sistema de inventario (Q, r) Multi-Artículos, cuya política de decisión fue minimizar la inversión total de inventario anual, sujeta a un número máximo esperado de pedidos por año y a un nivel mínimo esperado de satisfacción. Las pruebas realizadas y que son reportadas a manera de tablas en el capítulo anterior permiten llegar a las siguientes conclusiones:

Los resultados obtenidos de la metodología de multiplicadores de Lagrange para el caso descrito en (3.4), muestran que ésta, tiene menor tiempo de cómputo y mejor calidad en la solución, en comparación con las demás metodologías, sin embargo, como se mencionó anteriormente, dicha metodología sólo puede ser aplicada al caso en que no hayan pedidos pendientes, y que además únicamente ofrece una solución aproximada al problema, ya que en general dicha metodología es válida para funciones continuas y diferenciables y en este caso está siendo implementada para un problema cuyas variables de decisión son enteras y discretas.

El Algoritmo Genético fue implementado para resolver un problema de inventarios de 100 artículos empleando un tiempo de cómputo de 14726.0587 seg en un computador Turion 64, con un procesador 120 GB y 1.24 GB de memoria RAM, para el caso en el que hayan pedidos pendientes, y un tiempo de 15086.4888 seg en un computador Pentium 4 con un procesador 100 GB y 1.24 GB de memoria RAM para el caso sin pedidos pendientes. Los resultados muestran que las soluciones obtenidas con dicha metodología superan la calidad de las soluciones obtenidas por que el OptQuest en el mismo tiempo de cómputo, en especial cuando aumenta el número de artículos a ser analizados.

El algoritmo propuesto, basado en Algoritmos Genéticos ofrece una solución aceptable en un tiempo de cómputo razonable, y una solución alternativa para el caso en que hayan pedidos pendientes ante la carencia de una metodología con herramientas analíticas para una solución con más precisión, cualquiera sean las condiciones sobre Q y r , en dicho caso.

Problemas abiertos

Algunos de los problemas abiertos son los siguientes:

Realizar un estudio sobre el tipo de penalización más conveniente cuando se utilizan Algoritmos Genéticos en problemas de inventarios.

Comparar los resultados obtenidos por Algoritmos Genéticos con las soluciones analíticas obtenidas en [55], para el caso en que los valores del tamaño del pedido tienden a cero o tienden a infinito.

Estudiar algunos teoremas relacionados con la convergencia de los Algoritmos Genéticos.

Anexos

Tablas de los resultados obtenidos de Multiplicadores de Lagrange, sin considerar pedidos pendientes en el momento en que llega una nueva orden:

Resultados del método de Lagrange para 10 productos.										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
1	50	22	81	4.88219	2.20957	1.8401	8.2069	0.933797	11.9559	212.238
2	37	16	20	0.876712	0.936329	1.82421	2.73682	0.976516	8.77093	102.572
3	10	19	24	1.24932	1.11773	3.82377	4.11051	0.994764	4.96891	47.7308
4	66	20	42	2.30137	1.51703	1.52707	4.58714	0.934062	13.097	201.254
5	56	20	86	4.71233	2.17079	1.65782	7.69052	0.914958	12.0641	213.197
6	71	23	62	3.90685	1.97658	1.57888	6.61305	0.914521	14.5673	248.19
7	51	21	62	3.56712	1.88868	1.78008	6.52052	0.94106	11.7972	196.015
8	74	23	93	5.86027	2.4208	1.54655	8.71055	0.880484	14.8718	268.143
9	52	20	61	3.34247	1.82824	1.7204	6.15932	0.938311	11.6252	191.207
10	85	24	10	0.657534	0.810885	1.47405	2.1487	0.967038	16.2817	189.396

Resultados del método de Lagrange para 20 productos.										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
1	50	22	81	4.88219	2.20957	1.75099	8.3993	0.944281	12.5643	219.63
2	37	16	20	0.876712	0.936329	1.73587	2.79931	0.97998	9.2173	103.25
3	10	19	24	1.24932	1.11773	3.6386	4.16879	0.995499	5.22179	47.3877
4	66	20	42	2.30137	1.51703	1.45312	4.71907	0.9445	13.7635	207.521
5	56	20	86	4.71233	2.17079	1.57753	7.8967	0.928801	12.678	222.496
6	71	23	62	3.90685	1.97658	1.50242	6.80114	0.928444	15.3086	258.831
7	51	21	62	3.56712	1.88868	1.69388	6.6791	0.950293	12.3976	201.905
8	74	23	93	5.86027	2.4208	1.47165	8.97543	0.900923	15.6287	284.973
9	52	20	61	3.34247	1.82824	1.63708	6.315	0.948014	12.2169	197.136
10	85	24	10	0.657534	0.810885	1.40266	2.20701	0.971988	17.1103	191.319
11	79	15	82	3.36986	1.83572	1.15024	5.40047	0.865673	13.0407	205.853
12	46	20	64	3.50685	1.87266	1.74058	6.66312	0.954049	11.4904	185.222
13	71	17	53	2.46849	1.57114	1.29167	4.68972	0.921284	13.1612	203.562
14	43	19	34	1.76986	1.33036	1.75469	4.27887	0.97035	10.8281	145.613
15	75	21	33	1.89863	1.37791	1.39681	4.10939	0.945691	15.0343	218.187
16	30	17	53	2.46849	1.57114	1.98711	5.4995	0.973145	8.55516	120.737
17	88	21	30	1.72603	1.31378	1.28951	3.73994	0.937351	16.2852	233.963
18	26	18	14	0.690411	0.83091	2.19638	2.62061	0.98991	8.19532	78.738
19	7	23	25	1.57534	1.25513	4.78489	5.055	0.997217	4.8068	41.1047
20	68	16	33	1.44658	1.20274	1.28045	3.3476	0.943013	12.4956	172.805

Resultados del método de Lagrange para 30 productos.										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
1	50	22	81	4.88219	2.20957	1.75498	8.51047	0.949713	12.5358	225.288
2	37	16	20	0.876712	0.936329	1.73982	2.83605	0.981807	9.19634	104.682
3	10	19	24	1.24932	1.11773	3.64689	4.20339	0.99589	5.20992	47.7752
4	66	20	42	2.30137	1.51703	1.45643	4.7953	0.949908	13.7322	212.662
5	56	20	86	4.71233	2.17079	1.58113	8.01488	0.935915	12.6492	229.215
6	71	23	62	3.90685	1.97658	1.50585	6.90894	0.935598	15.2738	266.606
7	51	21	62	3.56712	1.88868	1.69774	6.77099	0.95509	12.3694	206.689
8	74	23	93	5.86027	2.4208	1.47501	9.12488	0.911261	15.5932	296.156
9	52	20	61	3.34247	1.82824	1.64081	6.4051	0.953051	12.1891	201.918
10	85	24	10	0.657534	0.810885	1.40586	2.24115	0.974587	17.0714	194.357
11	79	15	82	3.36986	1.83572	1.15286	5.53126	0.880484	13.0111	216.288
12	46	20	64	3.50685	1.87266	1.74454	6.75224	0.958456	11.4643	189.412
13	71	17	53	2.46849	1.57114	1.29462	4.77835	0.929243	13.1313	209.959
14	43	19	34	1.76986	1.33036	1.75869	4.33561	0.97311	10.8035	148.139
15	75	21	33	1.89863	1.37791	1.39999	4.17818	0.950973	15.0001	223.466
16	30	17	53	2.46849	1.57114	1.99163	5.56504	0.975631	8.53571	122.771
17	88	21	30	1.72603	1.31378	1.29245	3.80849	0.943527	16.2482	240.125
18	26	18	14	0.690411	0.83091	2.20138	2.64949	0.990807	8.17669	79.5539
19	7	23	25	1.57534	1.25513	4.79579	5.09164	0.997457	4.79587	41.3993
20	68	16	33	1.44658	1.20274	1.28337	3.40852	0.94858	12.4672	177.047
21	75	16	27	1.18356	1.08792	1.22201	2.95978	0.948732	13.0932	179.042
22	84	24	64	4.20822	2.05139	1.4142	7.0958	0.920378	16.9707	301.954
23	30	16	84	3.68219	1.9189	1.93217	7.20129	0.966667	8.28085	134.556
24	73	16	82	3.59452	1.89592	1.23864	5.99929	0.89767	12.9174	220.758
25	49	16	82	3.59452	1.89592	1.51185	6.53539	0.939568	10.5831	181.143
26	42	15	73	3	1.73205	1.58113	5.88068	0.95186	9.48691	154.192
27	74	19	87	4.52877	2.12809	1.34062	7.29877	0.90348	14.1725	254.583
28	86	21	80	4.60274	2.1454	1.30739	7.29747	0.895451	16.0625	287.965
29	7	23	28	1.76438	1.3283	4.79579	5.45872	0.997292	4.79587	42.6456
30	64	22	99	5.96712	2.44277	1.5512	9.42346	0.921455	14.1826	270.844

Resultados del método de Lagrange para 40 productos.										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
1	50	22	81	4.88219	2.20957	1.76986	8.52098	0.950204	12.4304	226.186
2	37	16	20	0.876712	0.936329	1.75458	2.83955	0.981973	9.11902	105.085
3	10	19	24	1.24932	1.11773	3.67781	4.2067	0.995926	5.16611	47.9629
4	66	20	42	2.30137	1.51703	1.46878	4.80252	0.950397	13.6168	213.545
5	56	20	86	4.71233	2.17079	1.59453	8.02604	0.936557	12.5428	230.215
6	71	23	62	3.90685	1.97658	1.51861	6.91911	0.936243	15.1454	267.781
7	51	21	62	3.56712	1.88868	1.71213	6.77969	0.955524	12.2654	207.5
8	74	23	93	5.86027	2.4208	1.48751	9.1389	0.912189	15.462	297.656
9	52	20	61	3.34247	1.82824	1.65473	6.41363	0.953506	12.0866	202.723
10	85	24	10	0.657534	0.810885	1.41778	2.2444	0.974824	16.9279	195.139
11	79	15	82	3.36986	1.83572	1.16264	5.54343	0.881801	12.9017	217.636
12	46	20	64	3.50685	1.87266	1.75934	6.76069	0.958855	11.3679	190.141
13	71	17	53	2.46849	1.57114	1.30559	4.7867	0.92996	13.0209	210.941
14	43	19	34	1.76986	1.33036	1.7736	4.34101	0.973361	10.7127	148.692
15	75	21	33	1.89863	1.37791	1.41186	4.18469	0.95145	14.874	224.399
16	30	17	53	2.46849	1.57114	2.00852	5.57127	0.975858	8.46393	123.211
17	88	21	30	1.72603	1.31378	1.30341	3.81497	0.944085	16.1116	241.177
18	26	18	14	0.690411	0.83091	2.22005	2.65224	0.990889	8.10793	79.8683
19	7	23	25	1.57534	1.25513	4.83646	5.09514	0.997479	4.75554	41.5662
20	68	16	33	1.44658	1.20274	1.29425	3.41428	0.949083	12.3624	177.809
21	75	16	27	1.18356	1.08792	1.23237	2.96499	0.949233	12.9831	179.821
22	84	24	64	4.20822	2.05139	1.42619	7.10719	0.921197	16.828	303.414
23	30	16	84	3.68219	1.9189	1.94855	7.20948	0.966982	8.21122	135.047
24	73	16	82	3.59452	1.89592	1.24914	6.01096	0.898765	12.8088	221.994
25	49	16	82	3.59452	1.89592	1.52467	6.54495	0.940169	10.4941	181.925
26	42	15	73	3	1.73205	1.59453	5.88882	0.952329	9.40714	154.816
27	74	19	87	4.52877	2.12809	1.35199	7.31154	0.904502	14.0533	255.949
28	86	21	80	4.60274	2.1454	1.31848	7.3108	0.896573	15.9274	289.588
29	7	23	28	1.76438	1.3283	4.83646	5.46245	0.997316	4.75554	42.8141
30	64	22	99	5.96712	2.44277	1.56435	9.43695	0.922261	14.0633	272.128
31	22	22	18	1.08493	1.0416	2.66817	3.60597	0.992247	8.24536	84.8127
32	36	19	68	3.53973	1.88142	1.93838	7.00756	0.967351	9.80201	159.733
33	25	22	29	1.74795	1.3221	2.50296	4.74112	0.988212	8.78959	106.116
34	75	24	31	2.03836	1.42771	1.50934	4.48995	0.957024	15.901	240.47
35	90	20	40	2.19178	1.48047	1.25779	4.36093	0.928564	15.901	251.824
36	59	18	27	1.33151	1.15391	1.47375	3.421	0.964914	12.2138	166.756
37	64	22	55	3.31507	1.82073	1.56435	6.25399	0.946752	14.0633	238.15
38	84	17	95	4.42466	2.10349	1.20032	6.78321	0.868911	14.1629	248.532
39	62	17	45	2.09589	1.44772	1.39714	4.43556	0.946964	12.1677	188.371
40	52	19	62	3.2274	1.7965	1.61283	6.20889	0.951504	11.7806	196.971

Resultados del método de Lagrange para 50 productos.										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
1	50	22	81	4.88219	2.20957	1.72775	8.47521	0.948038	12.7333	222.845
2	37	16	20	0.876712	0.936329	1.71283	2.82435	0.981241	9.34129	103.75
3	10	19	24	1.24932	1.11773	3.5903	4.19234	0.995769	5.29204	47.3818
4	66	20	42	2.30137	1.51703	1.43383	4.77113	0.94824	13.9487	210.32
5	56	20	86	4.71233	2.17079	1.55659	7.97747	0.933725	12.8486	226.432
6	71	23	62	3.90685	1.97658	1.48248	6.87481	0.933396	15.5146	263.354
7	51	21	62	3.56712	1.88868	1.67139	6.74182	0.95361	12.5644	204.53
8	74	23	93	5.86027	2.4208	1.45212	9.07773	0.908091	15.8389	291.82
9	52	20	61	3.34247	1.82824	1.61535	6.37651	0.951497	12.3812	199.769
10	85	24	10	0.657534	0.810885	1.38404	2.23029	0.973783	17.3405	192.506
11	79	15	82	3.36986	1.83572	1.13497	5.49021	0.875966	13.2162	212.339
12	46	20	64	3.50685	1.87266	1.71747	6.72394	0.957095	11.645	187.488
13	71	17	53	2.46849	1.57114	1.27453	4.75032	0.926796	13.3383	207.255
14	43	19	34	1.76986	1.33036	1.7314	4.31756	0.972256	10.9738	146.776
15	75	21	33	1.89863	1.37791	1.37827	4.15636	0.949343	15.2365	221.015
16	30	17	53	2.46849	1.57114	1.96073	5.54418	0.974862	8.67024	121.682
17	88	21	30	1.72603	1.31378	1.2724	3.78677	0.941624	16.5043	237.331
18	26	18	14	0.690411	0.83091	2.16722	2.64028	0.990529	8.30556	78.8704
19	7	23	25	1.57534	1.25513	4.72138	5.07994	0.997383	4.87146	41.057
20	68	16	33	1.44658	1.20274	1.26345	3.3892	0.946863	12.6637	175.056
21	75	16	27	1.18356	1.08792	1.20305	2.94232	0.947021	13.2995	177.021
22	84	24	64	4.20822	2.05139	1.39226	7.05755	0.91758	17.2382	297.819
23	30	16	84	3.68219	1.9189	1.90219	7.17391	0.965593	8.41137	133.284
24	73	16	82	3.59452	1.89592	1.21942	5.95999	0.893923	13.121	217.188
25	49	16	82	3.59452	1.89592	1.48839	6.50335	0.937517	10.7499	178.998
26	42	15	73	3	1.73205	1.55659	5.8534	0.950263	9.63643	152.531
27	74	19	87	4.52877	2.12809	1.31982	7.25581	0.899982	14.3959	250.634
28	86	21	80	4.60274	2.1454	1.28711	7.25256	0.891606	16.3157	283.23
29	7	23	28	1.76438	1.3283	4.72138	5.44624	0.997213	4.87146	42.2978
30	64	22	99	5.96712	2.44277	1.52713	9.37815	0.918699	14.4061	267.174
31	22	22	18	1.08493	1.0416	2.60468	3.59134	0.991943	8.44634	83.7924
32	36	19	68	3.53973	1.88142	1.89225	6.97278	0.965978	10.0409	157.651
33	25	22	29	1.74795	1.3221	2.4434	4.72126	0.987742	9.00383	104.875
34	75	24	31	2.03836	1.42771	1.47343	4.4616	0.955179	16.2885	236.997
35	90	20	40	2.19178	1.48047	1.22786	4.3264	0.925328	16.2885	247.369
36	59	18	27	1.33151	1.15391	1.43868	3.39928	0.963431	12.5115	164.44
37	64	22	55	3.31507	1.82073	1.52713	6.2155	0.94442	14.4061	234.496
38	84	17	95	4.42466	2.10349	1.17176	6.71873	0.862276	14.5081	241.916
39	62	17	45	2.09589	1.44772	1.3639	4.40499	0.944643	12.4643	185.445
40	52	19	62	3.2274	1.7965	1.57445	6.17197	0.9494	12.0677	194.053
41	35	21	64	3.68219	1.9189	2.01757	7.29598	0.970167	10.4085	161.79
42	58	21	74	4.25753	2.06338	1.56729	7.4483	0.938993	13.3989	230.516
43	33	16	46	2.01644	1.42001	1.81366	4.75623	0.97316	8.82192	120.339
44	37	16	87	3.8137	1.95287	1.71283	7.11431	0.954499	9.34129	153.81
45	34	17	92	4.28493	2.07001	1.84178	7.88842	0.959141	9.23018	153.829
46	43	22	77	4.6411	2.15432	1.86308	8.37034	0.958279	11.8084	200.414
47	46	19	35	1.82192	1.34978	1.67399	4.3484	0.969381	11.3502	154.72
48	82	23	81	5.10411	2.25923	1.37947	8.04919	0.903811	16.6731	298.055
49	19	24	22	1.44658	1.20274	2.9274	4.38525	0.992724	8.19839	83.6452
50	31	18	55	2.71233	1.64691	1.98477	5.91717	0.97417	9.06908	130.114

Resultados del método de Lagrange para 60 productos.										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
1	50	22	81	4.88219	2.20957	1.68361	8.42721	0.945686	13.0671	219.341
2	37	16	20	0.876712	0.936329	1.66907	2.80849	0.98045	9.58616	102.354
3	10	19	24	1.24932	1.11773	3.49859	4.17741	0.995599	5.43076	46.7739
4	66	20	42	2.30137	1.51703	1.3972	4.73821	0.945899	14.3143	206.939
5	56	20	86	4.71233	2.17079	1.51683	7.92643	0.930645	13.1854	222.461
6	71	23	62	3.90685	1.97658	1.44461	6.82826	0.930298	15.9212	258.704
7	51	21	62	3.56712	1.88868	1.6287	6.70215	0.951533	12.8937	201.418
8	74	23	93	5.86027	2.4208	1.41503	9.01317	0.903613	16.2541	285.67
9	52	20	61	3.34247	1.82824	1.57409	6.3376	0.949316	12.7058	196.673
10	85	24	10	0.657534	0.810885	1.34869	2.21555	0.972658	17.795	189.751
11	79	15	82	3.36986	1.83572	1.10598	5.43369	0.869549	13.5626	206.729
12	46	20	64	3.50685	1.87266	1.6736	6.68546	0.955187	11.9503	184.709
13	71	17	53	2.46849	1.57114	1.24197	4.71204	0.923349	13.6879	203.381
14	43	19	34	1.76986	1.33036	1.68717	4.29307	0.971062	11.2615	144.772
15	75	21	33	1.89863	1.37791	1.34306	4.12666	0.947057	15.6359	217.467
16	30	17	53	2.46849	1.57114	1.91064	5.51589	0.973786	8.89752	120.081
17	88	21	30	1.72603	1.31378	1.23989	3.75716	0.93895	16.9369	233.295
18	26	18	14	0.690411	0.83091	2.11186	2.62781	0.990141	8.52328	77.8266
19	7	23	25	1.57534	1.25513	4.60078	5.06413	0.997279	4.99916	40.5242
20	68	16	33	1.44658	1.20274	1.23118	3.36289	0.944453	12.9957	172.17
21	75	16	27	1.18356	1.08792	1.17232	2.91854	0.944619	13.6482	174.085
22	84	24	64	4.20822	2.05139	1.35669	7.00525	0.913634	17.6901	291.931
23	30	16	84	3.68219	1.9189	1.8536	7.1367	0.96409	8.63186	131.439
24	73	16	82	3.59452	1.89592	1.18827	5.90605	0.888618	13.465	212.114
25	49	16	82	3.59452	1.89592	1.45037	6.45966	0.934633	11.0317	175.926
26	42	15	73	3	1.73205	1.51683	5.81626	0.948022	9.88904	150.137
27	74	19	87	4.52877	2.12809	1.28611	7.19689	0.895036	14.7733	245.027
28	86	21	80	4.60274	2.1454	1.25423	7.19089	0.886163	16.7434	276.512
29	7	23	28	1.76438	1.3283	4.60078	5.42939	0.997103	4.99916	41.7577
30	64	22	99	5.96712	2.44277	1.48812	9.3162	0.914815	14.7838	261.961
31	22	22	18	1.08493	1.0416	2.53814	3.5761	0.991614	8.66775	82.7253
32	36	19	68	3.53973	1.88142	1.84392	6.93641	0.964493	10.3041	155.471
33	25	22	29	1.74795	1.3221	2.38099	4.70056	0.987235	9.23986	103.578
34	75	24	31	2.03836	1.42771	1.43579	4.4319	0.953179	16.7155	233.358
35	90	20	40	2.19178	1.48047	1.19649	4.29006	0.921804	16.7155	242.687
36	59	18	27	1.33151	1.15391	1.40193	3.37656	0.961826	12.8394	162.015
37	64	22	55	3.31507	1.82073	1.48812	6.17511	0.941888	14.7838	230.662
38	84	17	95	4.42466	2.10349	1.14183	6.65002	0.854958	14.8884	234.887
39	62	17	45	2.09589	1.44772	1.32906	4.37291	0.942122	12.791	182.376
40	52	19	62	3.2274	1.7965	1.53423	6.13325	0.947116	12.384	190.994

Resultados del método de Lagrange para 60 productos.										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
41	35	21	64	3.68219	1.9189	1.96604	7.26005	0.968876	10.6814	159.631
42	58	21	74	4.25753	2.06338	1.52726	7.40113	0.936185	13.7502	226.619
43	33	16	46	2.01644	1.42001	1.76734	4.73028	0.972007	9.05317	118.718
44	37	16	87	3.8137	1.95287	1.66907	7.07351	0.952465	9.58616	151.491
45	34	17	92	4.28493	2.07001	1.79474	7.84645	0.957332	9.47213	151.602
46	43	22	77	4.6411	2.15432	1.81549	8.32641	0.956428	12.118	197.502
47	46	19	35	1.82192	1.34978	1.63123	4.32297	0.968054	11.6477	152.567
48	82	23	81	5.10411	2.25923	1.34423	7.98773	0.899089	17.1102	291.571
49	19	24	22	1.44658	1.20274	2.85263	4.36793	0.992428	8.4133	82.6057
50	31	18	55	2.71233	1.64691	1.93407	5.88733	0.973063	9.30681	128.403
51	52	15	64	2.63014	1.62177	1.3632	5.11532	0.937286	11.0035	164.673
52	44	20	73	4	2	1.71122	7.36952	0.953983	11.6876	185.906
53	24	22	44	2.65205	1.62851	2.43009	6.16462	0.984494	9.05317	113.463
54	53	18	75	3.69863	1.92318	1.47916	6.63665	0.936705	12.1691	194.913
55	80	19	89	4.63288	2.15241	1.23694	7.17352	0.881074	15.3605	252.729
56	6	17	15	0.69863	0.835841	4.27233	3.09841	0.997955	3.97909	27.2157
57	40	24	22	1.44658	1.20274	1.96604	3.97255	0.982144	12.2073	140.36
58	8	19	87	4.52877	2.12809	3.91154	9.75069	0.992932	4.85742	57.4216
59	51	20	47	2.57534	1.60479	1.58945	5.34724	0.95794	12.583	181.898
60	49	17	47	2.18904	1.47954	1.495	4.70902	0.955736	11.3712	160.107

Resultados del método de Lagrange para 70 productos.										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
1	50	22	81	4.88219	2.20957	1.67936	8.42721	0.945686	13.1002	219.235
2	37	16	20	0.876712	0.936329	1.66486	2.80849	0.98045	9.61042	102.276
3	10	19	24	1.24932	1.11773	3.48976	4.17741	0.995599	5.4445	46.7297
4	66	20	42	2.30137	1.51703	1.39368	4.73821	0.945899	14.3505	206.823
5	56	20	86	4.71233	2.17079	1.513	7.92643	0.930645	13.2187	222.353
6	71	23	62	3.90685	1.97658	1.44096	6.82826	0.930298	15.9615	258.574
7	51	21	62	3.56712	1.88868	1.62459	6.70215	0.951533	12.9264	201.313
8	74	23	93	5.86027	2.4208	1.41145	9.01317	0.903613	16.2953	285.538
9	52	20	61	3.34247	1.82824	1.57012	6.3376	0.949316	12.7379	196.57
10	85	24	10	0.657534	0.810885	1.34529	2.21555	0.972658	17.8401	189.606
11	79	15	82	3.36986	1.83572	1.10319	5.43369	0.869549	13.5969	206.618
12	46	20	64	3.50685	1.87266	1.66938	6.68546	0.955187	11.9805	184.612
13	71	17	53	2.46849	1.57114	1.23884	4.71204	0.923349	13.7226	203.27
14	43	19	34	1.76986	1.33036	1.68291	4.29307	0.971062	11.29	144.68
15	75	21	33	1.89863	1.37791	1.33967	4.12666	0.947057	15.6755	217.339
16	30	17	53	2.46849	1.57114	1.90582	5.51589	0.973786	8.92003	120.009
17	88	21	30	1.72603	1.31378	1.23676	3.75716	0.93895	16.9798	233.158
18	26	18	14	0.690411	0.83091	2.10653	2.62781	0.990141	8.54485	77.7574
19	7	23	25	1.57534	1.25513	4.58916	5.06413	0.997279	5.01181	40.4836
20	68	16	33	1.44658	1.20274	1.22807	3.36289	0.944453	13.0285	172.064
21	75	16	27	1.18356	1.08792	1.16936	2.91854	0.944619	13.6827	173.975
22	84	24	64	4.20822	2.05139	1.35327	7.00525	0.913634	17.7348	291.788
23	30	16	84	3.68219	1.9189	1.84892	7.1367	0.96409	8.6537	131.369
24	73	16	82	3.59452	1.89592	1.18527	5.90605	0.888618	13.499	212.004
25	49	16	82	3.59452	1.89592	1.44671	6.45966	0.934633	11.0596	175.836
26	42	15	73	3	1.73205	1.513	5.81626	0.948022	9.91406	150.056
27	74	19	87	4.52877	2.12809	1.28286	7.19689	0.895036	14.8106	244.907
28	86	21	80	4.60274	2.1454	1.25106	7.19089	0.886163	16.7857	276.376
29	7	23	28	1.76438	1.3283	4.58916	5.42939	0.997103	5.01181	41.7171
30	64	22	99	5.96712	2.44277	1.48436	9.3162	0.914815	14.8212	261.84
31	22	22	18	1.08493	1.0416	2.53174	3.5761	0.991614	8.68969	82.6549
32	36	19	68	3.53973	1.88142	1.83926	6.93641	0.964493	10.3302	155.387
33	25	22	29	1.74795	1.3221	2.37498	4.70056	0.987235	9.26324	103.503
34	75	24	31	2.03836	1.42771	1.43217	4.4319	0.953179	16.7578	233.222
35	90	20	40	2.19178	1.48047	1.19347	4.29006	0.921804	16.7578	242.551
36	59	18	27	1.33151	1.15391	1.39839	3.37656	0.961826	12.8719	161.911
37	64	22	55	3.31507	1.82073	1.48436	6.17511	0.941888	14.8212	230.542
38	84	17	95	4.42466	2.10349	1.13895	6.65002	0.854958	14.9261	234.766
39	62	17	45	2.09589	1.44772	1.32571	4.37291	0.942122	12.8234	182.272
40	52	19	62	3.2274	1.7965	1.53036	6.13325	0.947116	12.4154	190.894

Resultados del método de Lagrange para 70 productos.										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
41	35	21	64	3.68219	1.9189	1.96108	7.26005	0.968876	10.7084	159.544
42	58	21	74	4.25753	2.06338	1.5234	7.40113	0.936185	13.7849	226.507
43	33	16	46	2.01644	1.42001	1.76288	4.73028	0.972007	9.07608	118.644
44	37	16	87	3.8137	1.95287	1.66486	7.07351	0.952465	9.61042	151.413
45	34	17	92	4.28493	2.07001	1.79021	7.84645	0.957332	9.4961	151.525
46	43	22	77	4.6411	2.15432	1.81091	8.32641	0.956428	12.1486	197.403
47	46	19	35	1.82192	1.34978	1.62711	4.32297	0.968054	11.6772	152.472
48	82	23	81	5.10411	2.25923	1.34084	7.98773	0.899089	17.1535	291.431
49	19	24	22	1.44658	1.20274	2.84543	4.36793	0.992428	8.43459	82.5373
50	31	18	55	2.71233	1.64691	1.92919	5.88733	0.973063	9.33036	128.327
51	52	15	64	2.63014	1.62177	1.35976	5.11532	0.937286	11.0314	164.583
52	44	20	73	4	2	1.7069	7.36952	0.953983	11.7172	185.811
53	24	22	44	2.65205	1.62851	2.42395	6.16462	0.984494	9.07608	113.389
54	53	18	75	3.69863	1.92318	1.47542	6.63665	0.936705	12.1999	194.814
55	80	19	89	4.63288	2.15241	1.23382	7.17352	0.881074	15.3994	252.604
56	6	17	15	0.69863	0.835841	4.26155	3.09841	0.997955	3.98916	27.1833
57	40	24	22	1.44658	1.20274	1.96108	3.97255	0.982144	12.2382	140.26
58	8	19	87	4.52877	2.12809	3.90167	9.75069	0.992932	4.86971	57.3821
59	51	20	47	2.57534	1.60479	1.58544	5.34724	0.95794	12.6148	181.796
60	49	17	47	2.18904	1.47954	1.49123	4.70902	0.955736	11.4	160.014
61	70	15	96	3.94521	1.98625	1.17197	6.25152	0.877207	12.799	202.461
62	21	15	77	3.16438	1.77887	2.13971	6.71176	0.976933	7.0103	96.9619
63	53	23	98	6.17534	2.48502	1.6678	9.95345	0.935789	13.7906	244.437
64	42	22	31	1.86849	1.36693	1.83234	4.56216	0.975615	12.0065	151.613
65	78	16	18	0.789041	0.88828	1.14665	2.29397	0.954887	13.9537	162.104
66	60	23	44	2.7726	1.66511	1.5675	5.5906	0.954713	14.6731	216.105
67	32	21	55	3.16438	1.77887	2.05094	6.63535	0.974484	10.2392	143.886
68	32	21	61	3.50959	1.87339	2.05094	7.11493	0.972854	10.2392	148.186
69	23	24	97	6.37808	2.52549	2.58619	11.4504	0.977702	9.28006	146.405
70	86	15	54	2.21918	1.48969	1.05734	4.05089	0.890575	14.1865	202.993

Resultados del método de Lagrange para 80 productos.										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
1	50	22	81	4.88219	2.20957	1.64947	8.4023	0.944434	13.3376	217.242
2	37	16	20	0.876712	0.936329	1.63522	2.8003	0.980031	9.78461	101.424
3	10	19	24	1.24932	1.11773	3.42763	4.16971	0.99551	5.54319	46.3421
4	66	20	42	2.30137	1.51703	1.36886	4.72113	0.944652	14.6107	204.877
5	56	20	86	4.71233	2.17079	1.48607	7.8999	0.929001	13.4583	220.114
6	71	23	62	3.90685	1.97658	1.41531	6.80406	0.928645	16.2508	255.946
7	51	21	62	3.56712	1.88868	1.59567	6.68158	0.950427	13.1607	199.527
8	74	23	93	5.86027	2.4208	1.38633	8.9795	0.901216	16.5906	282.117
9	52	20	61	3.34247	1.82824	1.54216	6.31743	0.948155	12.9688	194.794
10	85	24	10	0.657534	0.810885	1.32134	2.20793	0.97206	18.1634	187.94
11	79	15	82	3.36986	1.83572	1.08355	5.40406	0.866095	13.8434	203.502
12	46	20	64	3.50685	1.87266	1.63966	6.66552	0.954172	12.1977	183.011
13	71	17	53	2.46849	1.57114	1.21678	4.69212	0.921509	13.9713	201.073
14	43	19	34	1.76986	1.33036	1.65295	4.2804	0.970427	11.4946	143.491
15	75	21	33	1.89863	1.37791	1.31582	4.11125	0.945839	15.9596	215.29
16	30	17	53	2.46849	1.57114	1.87189	5.50126	0.973215	9.08171	119.061
17	88	21	30	1.72603	1.31378	1.21475	3.74179	0.937525	17.2876	230.836
18	26	18	14	0.690411	0.83091	2.06903	2.62138	0.989935	8.69973	77.1026
19	7	23	25	1.57534	1.25513	4.50746	5.05598	0.997224	5.10265	40.1406
20	68	16	33	1.44658	1.20274	1.20621	3.34924	0.943169	13.2647	170.392
21	75	16	27	1.18356	1.08792	1.14854	2.90621	0.943339	13.9307	172.269
22	84	24	64	4.20822	2.05139	1.32918	6.978	0.911523	18.0563	288.487
23	30	16	84	3.68219	1.9189	1.816	7.11744	0.96329	8.81056	130.298
24	73	16	82	3.59452	1.89592	1.16417	5.87786	0.885772	13.7437	209.176
25	49	16	82	3.59452	1.89592	1.42095	6.43696	0.933094	11.2601	174.093
26	42	15	73	3	1.73205	1.48607	5.797	0.946828	10.0938	148.682
27	74	19	87	4.52877	2.12809	1.26002	7.16613	0.892385	15.0791	241.786
28	86	21	80	4.60274	2.1454	1.22879	7.15864	0.88324	17.09	272.646
29	7	23	28	1.76438	1.3283	4.50746	5.4207	0.997044	5.10265	41.3704
30	64	22	99	5.96712	2.44277	1.45794	9.28394	0.912738	15.0898	258.93
31	22	22	18	1.08493	1.0416	2.48667	3.56824	0.99144	8.84719	81.9862
32	36	19	68	3.53973	1.88142	1.80652	6.91758	0.963703	10.5175	154.12
33	25	22	29	1.74795	1.3221	2.3327	4.68987	0.986966	9.43114	102.707
34	75	24	31	2.03836	1.42771	1.40667	4.4165	0.952114	17.0616	231.111
35	90	20	40	2.19178	1.48047	1.17223	4.27115	0.919921	17.0616	239.893
36	59	18	27	1.33151	1.15391	1.3735	3.3648	0.960973	13.1052	160.482
37	64	22	55	3.31507	1.82073	1.45794	6.15415	0.940538	15.0898	228.355
38	84	17	95	4.42466	2.10349	1.11867	6.6139	0.851008	15.1966	230.88
39	62	17	45	2.09589	1.44772	1.3021	4.35627	0.940778	13.0558	180.509
40	52	19	62	3.2274	1.7965	1.50312	6.11317	0.9459	12.6404	189.141

Resultados del método de Lagrange para 80 productos.										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
41	35	21	64	3.68219	1.9189	1.92616	7.24145	0.968191	10.9025	158.282
42	58	21	74	4.25753	2.06338	1.49628	7.37663	0.934688	14.0348	224.299
43	33	16	46	2.01644	1.42001	1.73149	4.71686	0.971394	9.24059	117.684
44	37	16	87	3.8137	1.95287	1.63522	7.05236	0.951383	9.78461	150.082
45	34	17	92	4.28493	2.07001	1.75834	7.8247	0.95637	9.66822	150.244
46	43	22	77	4.6411	2.15432	1.77867	8.30365	0.955444	12.3688	195.731
47	46	19	35	1.82192	1.34978	1.59814	4.30981	0.967348	11.8888	151.2
48	82	23	81	5.10411	2.25923	1.31697	7.95566	0.896558	17.4644	287.823
49	19	24	22	1.44658	1.20274	2.79477	4.35899	0.992272	8.58747	81.8863
50	31	18	55	2.71233	1.64691	1.89484	5.8719	0.972475	9.49948	127.317
51	52	15	64	2.63014	1.62177	1.33555	5.09617	0.935818	11.2313	162.958
52	44	20	73	4	2	1.67651	7.34807	0.952938	11.9295	184.198
53	24	22	44	2.65205	1.62851	2.3808	6.15099	0.984165	9.24059	112.544
54	53	18	75	3.69863	1.92318	1.44916	6.61388	0.935221	12.421	192.911
55	80	19	89	4.63288	2.15241	1.21185	7.14045	0.877992	15.6785	249.08
56	6	17	15	0.69863	0.835841	4.18568	3.09316	0.997914	4.06147	26.9242
57	40	24	22	1.44658	1.20274	1.92616	3.96221	0.981762	12.46	139.148
58	8	19	87	4.52877	2.12809	3.83221	9.73504	0.992787	4.95798	56.979
59	51	20	47	2.57534	1.60479	1.55721	5.33045	0.956993	12.8435	180.219
60	49	17	47	2.18904	1.47954	1.46468	4.69332	0.954735	11.6066	158.594
61	70	15	96	3.94521	1.98625	1.1511	6.22049	0.874002	13.031	199.558
62	21	15	77	3.16438	1.77887	2.10162	6.69565	0.976434	7.13737	96.2236
63	53	23	98	6.17534	2.48502	1.63811	9.92389	0.934281	14.0406	242.083
64	42	22	31	1.86849	1.36693	1.79972	4.54963	0.975086	12.2242	150.402
65	78	16	18	0.789041	0.88828	1.12624	2.28449	0.953864	14.2066	160.568
66	60	23	44	2.7726	1.66511	1.53959	5.57282	0.953686	14.939	214.201
67	32	21	55	3.16438	1.77887	2.01443	6.61888	0.973929	10.4248	142.775
68	32	21	61	3.50959	1.87339	2.01443	7.09735	0.972261	10.4248	147.039
69	23	24	97	6.37808	2.52549	2.54015	11.4277	0.977221	9.44827	145.353
70	86	15	54	2.21918	1.48969	1.03852	4.02893	0.887789	14.4437	200.295
71	27	19	46	2.39452	1.54742	2.08599	5.56187	0.979664	9.10838	113.679
72	15	19	99	5.15342	2.27012	2.79865	10.0216	0.984002	6.78899	94.0125
73	33	20	33	1.80822	1.3447	1.93587	4.55474	0.979448	10.3313	122.577
74	67	15	69	2.83562	1.68393	1.17659	5.05051	0.905798	12.7487	187.814
75	64	21	96	5.52329	2.35017	1.42442	8.70071	0.911812	14.7429	248.936
76	64	15	70	2.87671	1.69609	1.20385	5.15685	0.910583	12.46	184.452
77	1	20	36	1.9726	1.40449	11.1207	6.64721	0.999563	1.79845	10.235
78	34	22	57	3.43562	1.85354	2.00027	6.98202	0.972146	10.9985	154.582
79	82	19	10	0.520548	0.72149	1.19698	1.86916	0.969202	15.8733	159.662
80	29	21	89	5.12055	2.26286	2.11606	9.34598	0.96907	9.9241	153.22

Resultados del método de Lagrange para 90 productos.										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
1	50	22	81	4.88219	2.20957	1.62747	8.37676	0.943125	13.5179	215.415
2	37	16	20	0.876712	0.936329	1.61341	2.79192	0.979594	9.91686	100.711
3	10	19	24	1.24932	1.11773	3.38192	4.16185	0.995417	5.61811	46.035
4	66	20	42	2.30137	1.51703	1.35061	4.70362	0.943349	14.8081	203.118
5	56	20	86	4.71233	2.17079	1.46625	7.87266	0.927282	13.6402	218.033
6	71	23	62	3.90685	1.97658	1.39644	6.77921	0.926916	16.4705	253.511
7	51	21	62	3.56712	1.88868	1.57439	6.66049	0.949273	13.3385	197.909
8	74	23	93	5.86027	2.4208	1.36784	8.94483	0.898702	16.8148	278.867
9	52	20	61	3.34247	1.82824	1.5216	6.29674	0.946943	13.1441	193.184
10	85	24	10	0.657534	0.810885	1.30372	2.20013	0.971438	18.4089	186.528
11	79	15	82	3.36986	1.83572	1.0691	5.37343	0.862459	14.0305	200.511
12	46	20	64	3.50685	1.87266	1.61779	6.64508	0.953113	12.3625	181.568
13	71	17	53	2.46849	1.57114	1.20056	4.67166	0.919582	14.1601	199.044
14	43	19	34	1.76986	1.33036	1.63091	4.26743	0.969765	11.65	142.46
15	75	21	33	1.89863	1.37791	1.29827	4.09545	0.944567	16.1753	213.446
16	30	17	53	2.46849	1.57114	1.84693	5.48628	0.972619	9.20446	118.238
17	88	21	30	1.72603	1.31378	1.19855	3.72602	0.936035	17.5212	228.735
18	26	18	14	0.690411	0.83091	2.04144	2.61481	0.989721	8.81731	76.573
19	7	23	25	1.57534	1.25513	4.44735	5.04767	0.997167	5.17161	39.872
20	68	16	33	1.44658	1.20274	1.19012	3.33524	0.941828	13.444	168.893
21	75	16	27	1.18356	1.08792	1.13323	2.89355	0.942002	14.119	170.745
22	84	24	64	4.20822	2.05139	1.31145	6.94998	0.909313	18.3003	285.389
23	30	16	84	3.68219	1.9189	1.79179	7.09772	0.962457	8.92964	129.343
24	73	16	82	3.59452	1.89592	1.14864	5.84879	0.882782	13.9295	206.487
25	49	16	82	3.59452	1.89592	1.402	6.41366	0.931486	11.4122	172.487
26	42	15	73	3	1.73205	1.46625	5.77725	0.945582	10.2302	147.436
27	74	19	87	4.52877	2.12809	1.24322	7.13443	0.889602	15.2829	238.818
28	86	21	80	4.60274	2.1454	1.2124	7.12538	0.88017	17.321	269.081
29	7	23	28	1.76438	1.3283	4.44735	5.41184	0.996983	5.17161	41.0979
30	64	22	99	5.96712	2.44277	1.43849	9.25076	0.910563	15.2938	256.185
31	22	22	18	1.08493	1.0416	2.4535	3.56021	0.991259	8.96677	81.4447
32	36	19	68	3.53973	1.88142	1.78243	6.8983	0.96288	10.6596	152.992
33	25	22	29	1.74795	1.3221	2.30159	4.67895	0.986686	9.55861	102.045
34	75	24	31	2.03836	1.42771	1.38791	4.40072	0.951003	17.2922	229.224
35	90	20	40	2.19178	1.48047	1.15659	4.25172	0.91795	17.2922	237.441
36	59	18	27	1.33151	1.15391	1.35518	3.35274	0.960082	13.2824	159.231
37	64	22	55	3.31507	1.82073	1.43849	6.13264	0.939128	15.2938	226.356
38	84	17	95	4.42466	2.10349	1.10375	6.57649	0.846842	15.402	227.112
39	62	17	45	2.09589	1.44772	1.28474	4.33919	0.939374	13.2322	178.912
40	52	19	62	3.2274	1.7965	1.48307	6.09257	0.944629	12.8113	187.549
41	35	21	64	3.68219	1.9189	1.90048	7.22242	0.967476	11.0499	157.166
42	58	21	74	4.25753	2.06338	1.47633	7.35148	0.933122	14.2245	222.262
43	33	16	46	2.01644	1.42001	1.7084	4.70313	0.970756	9.36548	116.849
44	37	16	87	3.8137	1.95287	1.61341	7.03068	0.950253	9.91686	148.877
45	34	17	92	4.28493	2.07001	1.73489	7.80242	0.955365	9.7989	149.088

Resultados del método de Lagrange para 90 productos.										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
46	43	22	77	4.6411	2.15432	1.75495	8.28032	0.954416	12.536	194.218
47	46	19	35	1.82192	1.34978	1.57683	4.29633	0.966613	12.0495	150.09
48	82	23	81	5.10411	2.25923	1.2994	7.92263	0.893903	17.7004	284.394
49	19	24	22	1.44658	1.20274	2.7575	4.34986	0.992109	8.70353	81.3588
50	31	18	55	2.71233	1.64691	1.86957	5.85611	0.971862	9.62787	126.435
51	52	15	64	2.63014	1.62177	1.31774	5.07652	0.934282	11.3831	161.473
52	44	20	73	4	2	1.65415	7.32607	0.951847	12.0908	182.738
53	24	22	44	2.65205	1.62851	2.34905	6.13705	0.983822	9.36548	111.828
54	53	18	75	3.69863	1.92318	1.42983	6.59051	0.93367	12.5889	191.16
55	80	19	89	4.63288	2.15241	1.19569	7.10632	0.874753	15.8904	245.703
56	6	17	15	0.69863	0.835841	4.12986	3.0878	0.997871	4.11636	26.7246
57	40	24	22	1.44658	1.20274	1.90048	3.95163	0.981365	12.6284	138.212
58	8	19	87	4.52877	2.12809	3.7811	9.71906	0.992635	5.02499	56.6467
59	51	20	47	2.57534	1.60479	1.53644	5.31325	0.956004	13.0171	178.812
60	49	17	47	2.18904	1.47954	1.44515	4.67723	0.953689	11.7635	157.327
61	70	15	96	3.94521	1.98625	1.13575	6.18845	0.870632	13.2071	196.778
62	21	15	77	3.16438	1.77887	2.07359	6.67917	0.975914	7.23383	95.5832
63	53	23	98	6.17534	2.48502	1.61627	9.89354	0.932705	14.2303	239.895
64	42	22	31	1.86849	1.36693	1.77572	4.53681	0.974534	12.3894	149.359
65	78	16	18	0.789041	0.88828	1.11122	2.27478	0.952797	14.3986	159.225
66	60	23	44	2.7726	1.66511	1.51906	5.55459	0.952615	15.1409	212.491
67	32	21	55	3.16438	1.77887	1.98757	6.60203	0.97335	10.5657	141.806
68	32	21	61	3.50959	1.87339	1.98757	7.07935	0.971643	10.5657	146.033
69	23	24	97	6.37808	2.52549	2.50627	11.4045	0.976719	9.57597	144.429
70	86	15	54	2.21918	1.48969	1.02467	4.00629	0.884863	14.6389	197.752
71	27	19	46	2.39452	1.54742	2.05817	5.54797	0.979219	9.23149	112.928
72	15	19	99	5.15342	2.27012	2.76133	10.0021	0.983656	6.88075	93.4406
73	33	20	33	1.80822	1.3447	1.91005	4.54263	0.978998	10.4709	121.751
74	67	15	69	2.83562	1.68393	1.1609	5.02689	0.90342	12.921	185.706
75	64	21	96	5.52329	2.35017	1.40542	8.66865	0.90961	14.9421	246.277
76	64	15	70	2.87671	1.69609	1.1878	5.13358	0.908345	12.6284	182.449
77	1	20	36	1.9726	1.40449	10.9724	6.63946	0.999554	1.82275	10.1531
78	34	22	57	3.43562	1.85354	1.9736	6.9642	0.971525	11.1472	153.523
79	82	19	10	0.520548	0.72149	1.18102	1.86206	0.968512	16.0878	158.425
80	29	21	89	5.12055	2.26286	2.08784	9.32368	0.968376	10.0582	152.165
81	54	17	46	2.14247	1.46372	1.37662	4.52765	0.948401	12.3491	165.969
82	27	21	37	2.12877	1.45903	2.16379	5.21375	0.98276	9.70521	112.506
83	12	18	95	4.68493	2.16447	3.00492	9.52264	0.987293	5.99018	76.082
84	35	16	51	2.23562	1.4952	1.65887	4.97589	0.966578	9.64511	124.94
85	74	18	35	1.72603	1.31378	1.21006	3.74245	0.937587	14.8753	193.988
86	89	17	17	0.791781	0.889821	1.0723	2.24072	0.948275	15.8538	176.673
87	78	17	62	2.88767	1.69931	1.14542	5.05152	0.898555	14.8417	213.451
88	8	21	23	1.32329	1.15034	3.97513	4.4481	0.9967	5.28284	40.899
89	31	20	16	0.876712	0.936329	1.9707	2.96705	0.987208	10.1487	95.3463
90	22	19	62	3.2274	1.7965	2.28009	6.93707	0.980536	8.333	106.694

Tablas de los resultados obtenidos de Algoritmos Genéticos, sin considerar pedidos pendientes en el momento en que llega una nueva orden:

Resultados del Algoritmo Genético para 10 productos.										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
1	50	22	81	4.88219	2.20957	2	8	0.999881	11	530.89
2	37	16	20	0.876712	0.936329	3	3	1	5.33333	467.062
3	10	19	24	1.24932	1.11773	4	4	1	4.75	167.507
4	66	20	42	2.30137	1.51703	2	5	0.962371	10	310.11
5	56	20	86	4.71233	2.17079	2	8	0.105747	10	56
6	71	23	62	3.90685	1.97658	1	7	0.070693	23	390.5
7	51	21	62	3.56712	1.88868	3	7	1	7	787.077
8	74	23	93	5.86027	2.4208	2	9	0.361157	11.5	296
9	52	20	61	3.34247	1.82824	1	6	0.977281	20	216.192
10	85	24	10	0.657534	0.810885	1	2	0.66361	24	454.11

Resultados del Algoritmo Genético para 20 productos.										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
1	50	22	81	4.88219	2.20957	2	8	0.968813	11	430.89
2	37	16	20	0.876712	0.936329	2	3	0.999995	8	393.062
3	10	19	24	1.24932	1.11773	4	4	1	4.75	127.507
4	66	20	42	2.30137	1.51703	1	5	0.962371	20	673.11
5	56	20	86	4.71233	2.17079	2	8	0.371402	10	252
6	71	23	62	3.90685	1.97658	1	7	0.0240448	23	106.5
7	51	21	62	3.56712	1.88868	2	7	1	10.5	838.077
8	74	23	93	5.86027	2.4208	1	9	0.523014	23	565.34
9	52	20	61	3.34247	1.82824	1	7	0.977281	20	216.192
10	85	24	10	0.657534	0.810885	1	2	1	24	964.11
11	79	15	82	3.36986	1.83572	1	5	1	15	1155.78
12	46	20	64	3.50685	1.87266	3	7	0.908462	6.66667	275.685
13	71	17	53	2.46849	1.57114	2	5	0.382781	8.5	319.5
14	43	19	34	1.76986	1.33036	2	4	0.0916995	9.5	172
15	75	21	33	1.89863	1.37791	1	4	0.787943	21	157.603
16	30	17	53	2.46849	1.57114	3	6	0.382781	5.66667	105
17	88	21	30	1.72603	1.31378	2	3	1	10.5	860.11
18	26	18	14	0.690411	0.83091	2	3	1	9	307.049
19	7	23	25	1.57534	1.25513	5	6	0.632446	4.6	13.4726
20	68	16	33	1.44658	1.20274	1	3	0.677291	16	445.633

Resultados del Algoritmo Genético para 30 productos.										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
1	50	22	81	4.88219	2.20957	3	9	0.52126	7.33333	55.8904
2	37	16	20	0.876712	0.936329	3	3	1	5.33333	467.062
3	10	19	24	1.24932	1.11773	3	8	1	6.33333	82.5068
4	66	20	42	2.30137	1.51703	1	5	1	20	772.11
5	56	20	86	4.71233	2.17079	2	8	0.999606	10	716.11
6	71	23	62	3.90685	1.97658	2	7	1	11.5	1178.11
7	51	21	62	3.56712	1.88868	1	8	0.203342	21	280.5
8	74	23	93	5.86027	2.4208	1	9	0.998408	23	787.34
9	52	20	61	3.34247	1.82824	1	6	0.425705	20	390
10	85	24	10	0.657534	0.810885	1	2	1	24	921.61
11	79	15	82	3.36986	1.83572	1	5	0.999848	15	918.781
12	46	20	64	3.50685	1.87266	3	8	0.998323	6.66667	321.685
13	71	17	53	2.46849	1.57114	1	5	1	17	1102.74
14	43	19	34	1.76986	1.33036	1	4	1	19	461.396
15	75	21	33	1.89863	1.37791	2	4	1	10.5	1132.6
16	30	17	53	2.46849	1.57114	4	5	0.987703	4.25	165.945
17	88	21	30	1.72603	1.31378	1	4	0.833901	21	508.11
18	26	18	14	0.690411	0.83091	2	3	1	9	320.049
19	7	23	25	1.57534	1.25513	5	8	0.999788	4.6	37.9726
20	68	16	33	1.44658	1.20274	1	3	1	16	989.633
21	75	16	27	1.18356	1.08792	1	3	0.995185	16	436.233
22	84	24	64	4.20822	2.05139	1	8	0.91323	24	570.51
23	30	16	84	3.68219	1.9189	3	8	0.0810906	5.33333	210
24	73	16	82	3.59452	1.89592	1	6	0.0855818	16	438
25	49	16	82	3.59452	1.89592	1	6	0.897738	16	485.368
26	42	15	73	3	1.73205	2	6	1	7.5	777
27	74	19	87	4.52877	2.12809	1	8	0.999777	19	774.871
28	86	21	80	4.60274	2.1454	1	8	0.0159605	21	43
29	7	23	28	1.76438	1.3283	6	5	1	3.83333	99.6493
30	64	22	99	5.96712	2.44277	2	9	0.993238	11	738.104

Resultados del Algoritmo Genético para 40 productos.										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
1	50	22	81	4.88219	2.20957	3	8	0.920885	7.33333	430.89
2	37	16	20	0.876712	0.936329	2	4	1	8	485.562
3	10	19	24	1.24932	1.11773	4	9	1	4.75	142.507
4	66	20	42	2.30137	1.51703	2	4	1	10	1036.11
5	56	20	86	4.71233	2.17079	1	8	0.371402	20	308
6	71	23	62	3.90685	1.97658	1	8	0.998974	23	823.114
7	51	21	62	3.56712	1.88868	4	7	0.999958	5.25	532.077
8	74	23	93	5.86027	2.4208	4	9	0.994397	5.75	750.34
9	52	20	61	3.34247	1.82824	2	8	0.10005	10	52
10	85	24	10	0.657534	0.810885	1	2	1	24	751.61
11	79	15	82	3.36986	1.83572	2	5	0.0331996	7.5	513.5
12	46	20	64	3.50685	1.87266	3	7	0.603857	6.66667	114.685
13	71	17	53	2.46849	1.57114	1	5	1	17	925.237
14	43	19	34	1.76986	1.33036	8	4	0.822429	2.375	375.396
15	75	21	33	1.89863	1.37791	2	4	1	10.5	945.103
16	30	17	53	2.46849	1.57114	3	6	1	5.66667	525.945
17	88	21	30	1.72603	1.31378	1	4	0.993649	21	860.11
18	26	18	14	0.690411	0.83091	3	3	0.645273	6	60.0493
19	7	23	25	1.57534	1.25513	5	5	1	4.6	72.9726
20	68	16	33	1.44658	1.20274	1	4	1	16	1091.63
21	75	16	27	1.18356	1.08792	1	3	1	16	848.733
22	84	24	64	4.20822	2.05139	1	8	0.277939	24	294
23	30	16	84	3.68219	1.9189	2	8	0.999503	8	309.534
24	73	16	82	3.59452	1.89592	1	6	1	16	1270.6
25	49	16	82	3.59452	1.89592	1	7	0.584675	16	289.368
26	42	15	73	3	1.73205	1	6	0.5	15	147
27	74	19	87	4.52877	2.12809	1	7	0.948571	19	663.871
28	86	21	80	4.60274	2.1454	1	7	0.0159605	21	516
29	7	23	28	1.76438	1.3283	6	5	0.992573	3.83333	61.1493
30	64	22	99	5.96712	2.44277	1	9	0.980316	22	386.104
31	22	22	18	1.08493	1.0416	3	4	1	7.33333	295.132
32	36	19	68	3.53973	1.88142	3	8	0.781172	6.33333	268.57
33	25	22	29	1.74795	1.3221	3	6	1	7.33333	343.801
34	75	24	31	2.03836	1.42771	1	5	0.999985	24	672.123
35	90	20	40	2.19178	1.48047	1	4	1	20	1242.74
36	59	18	27	1.33151	1.15391	1	4	1	18	570.441
37	64	22	55	3.31507	1.82073	1	6	0.235063	22	480
38	84	17	95	4.42466	2.10349	1	7	0.999114	17	636.329
39	62	17	45	2.09589	1.44772	1	4	0.473595	17	93
40	52	19	62	3.2274	1.7965	1	6	0.999999	19	846.175

Resultados del Algoritmo Genético para 50 productos.										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
1	50	22	81	4.88219	2.20957	1	8	0.999982	22	680.89
2	37	16	20	0.876712	0.936329	4	4	1	4	411.562
3	10	19	24	1.24932	1.11773	3	4	0.993072	6.33333	92.5068
4	66	20	42	2.30137	1.51703	1	5	0.999023	20	640.11
5	56	20	86	4.71233	2.17079	2	8	0.992571	10	688.11
6	71	23	62	3.90685	1.97658	1	7	0.995013	23	823.114
7	51	21	62	3.56712	1.88868	1	8	0.0294672	21	255
8	74	23	93	5.86027	2.4208	1	9	0.221109	23	111
9	52	20	61	3.34247	1.82824	2	8	0.231385	10	52
10	85	24	10	0.657534	0.810885	4	3	1	6	1431.61
11	79	15	82	3.36986	1.83572	1	5	1	15	1234.78
12	46	20	64	3.50685	1.87266	4	8	0.998323	5	551.685
13	71	17	53	2.46849	1.57114	4	5	0.17498	4.25	284
14	43	19	34	1.76986	1.33036	1	4	1	19	676.396
15	75	21	33	1.89863	1.37791	1	4	1	21	1282.6
16	30	17	53	2.46849	1.57114	2	6	1	8.5	390.945
17	88	21	30	1.72603	1.31378	1	4	1	21	1212.11
18	26	18	14	0.690411	0.83091	3	3	0.997279	6	216.049
19	7	23	25	1.57534	1.25513	5	5	1	4.6	100.973
20	68	16	33	1.44658	1.20274	2	4	1	8	819.633
21	75	16	27	1.18356	1.08792	1	3	0.138316	16	375
22	84	24	64	4.20822	2.05139	2	8	0.997624	12	738.51
23	30	16	84	3.68219	1.9189	4	8	0.190341	4	60
24	73	16	82	3.59452	1.89592	2	6	0.584675	8	504.1
25	49	16	82	3.59452	1.89592	2	6	0.0289852	8	147
26	42	15	73	3	1.73205	4	6	0.998054	3.75	462
27	74	19	87	4.52877	2.12809	1	7	0.948571	19	737.871
28	86	21	80	4.60274	2.1454	1	7	0.742568	21	464.164
29	7	23	28	1.76438	1.3283	5	6	1	4.6	96.1493
30	64	22	99	5.96712	2.44277	1	10	0.112249	22	320
31	22	22	18	1.08493	1.0416	3	3	1	7.33333	284.132
32	36	19	68	3.53973	1.88142	2	7	0.967056	9.5	286.57
33	25	22	29	1.74795	1.3221	4	5	1	5.5	418.801
34	75	24	31	2.03836	1.42771	1	4	0.980979	24	784.623
35	90	20	40	2.19178	1.48047	1	4	0.448465	20	675
36	59	18	27	1.33151	1.15391	9	4	0.999262	2	599.941
37	64	22	55	3.31507	1.82073	1	6	0.64661	22	363.836
38	84	17	95	4.42466	2.10349	4	7	0.607773	4.25	342.329
39	62	17	45	2.09589	1.44772	1	4	0.224532	17	372
40	52	19	62	3.2274	1.7965	1	6	0.666424	19	144.175
41	35	21	64	3.68219	1.9189	3	8	0.999503	7	448.623
42	58	21	74	4.25753	2.06338	1	7	1	21	652.063
43	33	16	46	2.01644	1.42001	10	4	1	1.6	593.458
44	37	16	87	3.8137	1.95287	2	7	0.0254175	8	222
45	34	17	92	4.28493	2.07001	1	8	0.999903	17	330.312
46	43	22	77	4.6411	2.15432	5	8	0.735908	4.4	230.433
47	46	19	35	1.82192	1.34978	1	4	1	19	629.192
48	82	23	81	5.10411	2.25923	2	8	0.984885	11.5	975.463
49	19	24	22	1.44658	1.20274	4	4	0.355207	6	76
50	31	18	55	2.71233	1.64691	3	6	0.999995	6	303.418

Resultados del Algoritmo Genético para 60 productos.										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
1	50	22	81	4.88219	2.20957	1	8	0.999982	22	755.89
2	37	16	20	0.876712	0.936329	1	3	1	16	559.562
3	10	19	24	1.24932	1.11773	3	8	1	6.33333	127.507
4	66	20	42	2.30137	1.51703	2	5	0.962371	10	475.11
5	56	20	86	4.71233	2.17079	1	7	0.0436211	20	392
6	71	23	62	3.90685	1.97658	1	7	0.070693	23	35.5
7	51	21	62	3.56712	1.88868	1	7	0.381984	21	204
8	74	23	93	5.86027	2.4208	4	9	0.994397	5.75	713.34
9	52	20	61	3.34247	1.82824	2	6	0.231385	10	286
10	85	24	10	0.657534	0.810885	1	2	0.66361	24	114.11
11	79	15	82	3.36986	1.83572	1	6	1	15	879.281
12	46	20	64	3.50685	1.87266	1	7	1	20	666.685
13	71	17	53	2.46849	1.57114	1	5	0.632429	17	357.237
14	43	19	34	1.76986	1.33036	1	4	1	19	547.396
15	75	21	33	1.89863	1.37791	2	5	1	10.5	1095.1
16	30	17	53	2.46849	1.57114	2	5	0.998038	8.5	225.945
17	88	21	30	1.72603	1.31378	1	4	0.999999	21	596.11
18	26	18	14	0.690411	0.83091	4	3	1	4.5	320.049
19	7	23	25	1.57534	1.25513	6	8	1	3.83333	100.973
20	68	16	33	1.44658	1.20274	1	4	0.999923	16	445.633
21	75	16	27	1.18356	1.08792	1	3	1	16	998.733
22	84	24	64	4.20822	2.05139	1	8	0.999927	24	780.51
23	30	16	84	3.68219	1.9189	4	8	0.98778	4	234.534
24	73	16	82	3.59452	1.89592	1	5	0.0855818	16	474.5
25	49	16	82	3.59452	1.89592	1	6	0.0289852	16	269.5
26	42	15	73	3	1.73205	1	6	0.958368	15	357
27	74	19	87	4.52877	2.12809	1	6	0.236263	19	555
28	86	21	80	4.60274	2.1454	1	7	0.868087	21	851.164
29	7	23	28	1.76438	1.3283	3	8	0.99996	7.66667	89.1493
30	64	22	99	5.96712	2.44277	2	9	0.980316	11	770.104
31	22	22	18	1.08493	1.0416	5	8	1	4.4	240.132
32	36	19	68	3.53973	1.88142	1	8	0.967056	19	286.57
33	25	22	29	1.74795	1.3221	3	5	1	7.33333	306.301
34	75	24	31	2.03836	1.42771	2	6	1	12	859.623
35	90	20	40	2.19178	1.48047	2	4	0.210409	10	405
36	59	18	27	1.33151	1.15391	2	4	1	9	717.941
37	64	22	55	3.31507	1.82073	3	6	0.0343237	7.33333	64
38	84	17	95	4.42466	2.10349	1	7	0.955408	17	552.329
39	62	17	45	2.09589	1.44772	4	4	1	4.25	1017.05
40	52	19	62	3.2274	1.7965	8	7	0.938626	2.375	456.175

Resultados del Algoritmo Genético para 60 productos.										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
41	35	21	64	3.68219	1.9189	3	7	0.886454	7	203.623
42	58	21	74	4.25753	2.06338	1	8	1	21	855.063
43	33	16	46	2.01644	1.42001	2	5	1	8	461.458
44	37	16	87	3.8137	1.95287	2	8	1	8	413.893
45	34	17	92	4.28493	2.07001	3	9	0.905176	5.66667	143.312
46	43	22	77	4.6411	2.15432	2	8	0.863234	11	316.433
47	46	19	35	1.82192	1.34978	2	4	0.999938	9.5	445.192
48	82	23	81	5.10411	2.25923	1	7	0.995469	23	893.463
49	19	24	22	1.44658	1.20274	4	4	1	6	257.515
50	31	18	55	2.71233	1.64691	4	8	0.917593	4.5	117.418
51	52	15	64	2.63014	1.62177	2	5	1	7.5	695.233
52	44	20	73	4	2	1	8	0.99379	20	308
53	24	22	44	2.65205	1.62851	3	6	0.584593	7.33333	116.351
54	53	18	75	3.69863	1.92318	1	7	0.987343	18	386.973
55	80	19	89	4.63288	2.15241	1	8	0.999999	19	1309.37
56	6	17	15	0.69863	0.835841	5	4	1	3.4	79.8082
57	40	24	22	1.44658	1.20274	4	4	0.355207	6	160
58	8	19	87	4.52877	2.12809	8	10	0.0486401	2.375	28
59	51	20	47	2.57534	1.60479	2	6	0.999638	10	659.158
60	49	17	47	2.18904	1.47954	2	8	0.889524	8.5	309.237

Resultados del Algoritmo Genético para 70 productos.										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
1	50	22	81	4.88219	2.20957	1	7	0.999362	22	530.89
2	37	16	20	0.876712	0.936329	2	3	1	8	337.562
3	10	19	24	1.24932	1.11773	3	4	1	6.33333	167.507
4	66	20	42	2.30137	1.51703	1	6	1	20	772.11
5	56	20	86	4.71233	2.17079	4	8	0.0436211	5	280
6	71	23	62	3.90685	1.97658	1	7	0.941197	23	397.114
7	51	21	62	3.56712	1.88868	1	7	0.590641	21	277.077
8	74	23	93	5.86027	2.4208	1	10	0.99992	23	713.34
9	52	20	61	3.34247	1.82824	2	6	0.817698	10	424.192
10	85	24	10	0.657534	0.810885	3	4	0.208716	8	127.5
11	79	15	82	3.36986	1.83572	1	5	0.420161	15	316
12	46	20	64	3.50685	1.87266	2	7	0.999969	10	597.685
13	71	17	53	2.46849	1.57114	4	5	0.999785	4.25	499.237
14	43	19	34	1.76986	1.33036	1	4	1	19	396.896
15	75	21	33	1.89863	1.37791	1	4	0.998542	21	757.603
16	30	17	53	2.46849	1.57114	1	5	0.999984	17	210.945
17	88	21	30	1.72603	1.31378	4	4	0.99997	5.25	1036.11
18	26	18	14	0.690411	0.83091	3	4	0.997279	6	177.049
19	7	23	25	1.57534	1.25513	5	5	1	4.6	90.4726
20	68	16	33	1.44658	1.20274	1	3	1	16	1397.63
21	75	16	27	1.18356	1.08792	1	4	1	16	961.233
22	84	24	64	4.20822	2.05139	1	7	0.997624	24	528.51
23	30	16	84	3.68219	1.9189	8	8	0.361103	2	150
24	73	16	82	3.59452	1.89592	2	5	0.0855818	8	365
25	49	16	82	3.59452	1.89592	1	6	0.989928	16	313.868
26	42	15	73	3	1.73205	1	5	0.875893	15	210
27	74	19	87	4.52877	2.12809	1	8	0.0486401	19	37
28	86	21	80	4.60274	2.1454	1	7	0.0465482	21	516
29	7	23	28	1.76438	1.3283	4	8	1	5.75	75.1493
30	64	22	99	5.96712	2.44277	1	12	0.210328	22	480
31	22	22	18	1.08493	1.0416	3	4	1	7.33333	383.132
32	36	19	68	3.53973	1.88142	4	8	0.967056	4.75	250.57
33	25	22	29	1.74795	1.3221	3	8	1	7.33333	493.801
34	75	24	31	2.03836	1.42771	2	4	0.999999	12	1084.62
35	90	20	40	2.19178	1.48047	2	4	0.210409	10	495
36	59	18	27	1.33151	1.15391	2	4	1	9	1071.94
37	64	22	55	3.31507	1.82073	1	6	0.0343237	22	480
38	84	17	95	4.42466	2.10349	1	7	0.98519	17	846.329
39	62	17	45	2.09589	1.44772	1	4	0.473595	17	186
40	52	19	62	3.2274	1.7965	1	6	0.999992	19	482.175

Resultados del Algoritmo Genético para 70 productos.										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
41	35	21	64	3.68219	1.9189	2	7	1	10.5	571.123
42	58	21	74	4.25753	2.06338	2	8	0.999912	10.5	797.063
43	33	16	46	2.01644	1.42001	2	5	0.495382	8	49.5
44	37	16	87	3.8137	1.95287	4	8	0.0254175	4	74
45	34	17	92	4.28493	2.07001	1	8	1	17	449.312
46	43	22	77	4.6411	2.15432	3	9	0.978481	7.33333	208.933
47	46	19	35	1.82192	1.34978	2	4	0.0885426	9.5	92
48	82	23	81	5.10411	2.25923	1	7	0.0346393	23	492
49	19	24	22	1.44658	1.20274	4	4	1	6	219.515
50	31	18	55	2.71233	1.64691	4	6	0.995386	4.5	334.418
51	52	15	64	2.63014	1.62177	8	5	0.590201	1.875	305.233
52	44	20	73	4	2	1	7	0.308538	20	22
53	24	22	44	2.65205	1.62851	3	8	0.996206	7.33333	248.351
54	53	18	75	3.69863	1.92318	1	7	0.562261	18	227.973
55	80	19	89	4.63288	2.15241	1	7	0.998452	19	629.37
56	6	17	15	0.69863	0.835841	4	3	1	4.25	73.8082
57	40	24	22	1.44658	1.20274	3	4	1	8	502.137
58	8	19	87	4.52877	2.12809	9	12	0.236263	2.11111	24
59	51	20	47	2.57534	1.60479	8	5	0.0542703	2.5	382.5
60	49	17	47	2.18904	1.47954	2	5	1	8.5	799.237
61	70	15	96	3.94521	1.98625	1	6	0.999997	15	1088.84
62	21	15	77	3.16438	1.77887	2	7	1	7.5	238.048
63	53	23	98	6.17534	2.48502	1	10	0.999808	23	573.707
64	42	22	31	1.86849	1.36693	4	5	1	5.5	656.523
65	78	16	18	0.789041	0.88828	2	2	0.993595	8	250.455
66	60	23	44	2.7726	1.66511	1	6	0.769477	23	283.644
67	32	21	55	3.16438	1.77887	3	8	1	7	426.74
68	32	21	61	3.50959	1.87339	4	7	0.999734	5.25	255.693
69	23	24	97	6.37808	2.52549	3	12	0.924235	8	232.804
70	86	15	54	2.21918	1.48969	1	4	0.441515	15	387

Resultados del Algoritmo Genético para 80 productos.										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
1	50	22	81	4.88219	2.20957	1	8	0.997187	22	580.89
2	37	16	20	0.876712	0.936329	2	4	1	8	707.562
3	10	19	24	1.24932	1.11773	3	8	0.131842	6.33333	15
4	66	20	42	2.30137	1.51703	2	5	0.992617	10	343.11
5	56	20	86	4.71233	2.17079	1	8	0.105747	20	420
6	71	23	62	3.90685	1.97658	1	7	0.32319	23	355
7	51	21	62	3.56712	1.88868	1	7	0.990539	21	302.577
8	74	23	93	5.86027	2.4208	1	10	0.994397	23	750.34
9	52	20	61	3.34247	1.82824	1	6	1	20	710.192
10	85	24	10	0.657534	0.810885	1	2	1	24	581.61
11	79	15	82	3.36986	1.83572	4	5	0.999848	3.75	800.281
12	46	20	64	3.50685	1.87266	1	7	0.999737	20	344.685
13	71	17	53	2.46849	1.57114	1	5	0.946437	17	712.237
14	43	19	34	1.76986	1.33036	1	4	1	19	482.896
15	75	21	33	1.89863	1.37791	2	4	1	10.5	757.603
16	30	17	53	2.46849	1.57114	3	5	0.999999	5.66667	240.945
17	88	21	30	1.72603	1.31378	1	3	0.958261	21	596.11
18	26	18	14	0.690411	0.83091	3	3	1	6	281.049
19	7	23	25	1.57534	1.25513	8	5	0.996819	2.875	30.9726
20	68	16	33	1.44658	1.20274	4	3	0.983124	4	309.633
21	75	16	27	1.18356	1.08792	2	3	0.138316	8	262.5
22	84	24	64	4.20822	2.05139	1	8	0.058918	24	336
23	30	16	84	3.68219	1.9189	2	7	0.565772	8	54.5342
24	73	16	82	3.59452	1.89592	2	6	0.0855818	8	109.5
25	49	16	82	3.59452	1.89592	2	6	0.96377	8	215.868
26	42	15	73	3	1.73205	1	8	0.281851	15	315
27	74	19	87	4.52877	2.12809	1	7	0.948571	19	589.871
28	86	21	80	4.60274	2.1454	2	7	0.999994	10.5	1066.16
29	7	23	28	1.76438	1.3283	5	6	0.999999	4.6	92.6493
30	64	22	99	5.96712	2.44277	1	9	0.346085	22	352
31	22	22	18	1.08493	1.0416	5	3	0.997434	4.4	141.132
32	36	19	68	3.53973	1.88142	3	9	1	6.33333	556.57
33	25	22	29	1.74795	1.3221	1	5	0.99935	22	193.801
34	75	24	31	2.03836	1.42771	2	4	0.915276	12	634.623
35	90	20	40	2.19178	1.48047	1	4	1	20	1647.74
36	59	18	27	1.33151	1.15391	1	3	0.925904	18	422.941
37	64	22	55	3.31507	1.82073	1	6	0.235063	22	128
38	84	17	95	4.42466	2.10349	6	7	0.0517533	2.83333	336
39	62	17	45	2.09589	1.44772	1	4	1	17	831.055
40	52	19	62	3.2274	1.7965	2	6	0.938626	9.5	352.175

Resultados del Algoritmo Genético para 80 productos.										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
41	35	21	64	3.68219	1.9189	2	7	0.997208	10.5	238.623
42	58	21	74	4.25753	2.06338	2	8	0.450336	10.5	319
43	33	16	46	2.01644	1.42001	2	5	0.999987	8	312.958
44	37	16	87	3.8137	1.95287	2	7	0.996043	8	395.393
45	34	17	92	4.28493	2.07001	1	8	0.635119	17	143.312
46	43	22	77	4.6411	2.15432	1	8	0.99842	22	337.933
47	46	19	35	1.82192	1.34978	1	4	1	19	514.192
48	82	23	81	5.10411	2.25923	2	8	0.481622	11.5	205
49	19	24	22	1.44658	1.20274	3	4	1	8	248.015
50	31	18	55	2.71233	1.64691	2	7	0.995386	9	303.418
51	52	15	64	2.63014	1.62177	1	5	1	15	591.233
52	44	20	73	4	2	2	7	1	10	660
53	24	22	44	2.65205	1.62851	3	6	0.999997	7.33333	296.351
54	53	18	75	3.69863	1.92318	2	7	1	9	863.973
55	80	19	89	4.63288	2.15241	2	8	0.86428	9.5	309.37
56	6	17	15	0.69863	0.835841	5	4	0.640785	3.4	37.8082
57	40	24	22	1.44658	1.20274	1	4	0.677291	24	62.137
58	8	19	87	4.52877	2.12809	8	10	0.236263	2.375	20
59	51	20	47	2.57534	1.60479	1	5	0.997085	20	353.158
60	49	17	47	2.18904	1.47954	2	5	1	8.5	578.737
61	70	15	96	3.94521	1.98625	2	6	0.702307	7.5	318.836
62	21	15	77	3.16438	1.77887	2	7	0.944538	7.5	154.048
63	53	23	98	6.17534	2.48502	4	9	0.190683	5.75	26.5
64	42	22	31	1.86849	1.36693	2	8	0.998746	11	425.523
65	78	16	18	0.789041	0.88828	1	2	0.99985	16	406.455
66	60	23	44	2.7726	1.66511	3	6	1	7.66667	1003.64
67	32	21	55	3.16438	1.77887	4	8	0.680731	5.25	106.74
68	32	21	61	3.50959	1.87339	3	8	0.090188	7	64
69	23	24	97	6.37808	2.52549	3	12	0.986995	8	140.804
70	86	15	54	2.21918	1.48969	4	4	1	3.75	841.151
71	27	19	46	2.39452	1.54742	2	5	1	9.5	434.848
72	15	19	99	5.15342	2.27012	2	10	0.0824007	9.5	45
73	33	20	33	1.80822	1.3447	3	4	0.991192	6.66667	138.329
74	67	15	69	2.83562	1.68393	1	5	1	15	1016.01
75	64	21	96	5.52329	2.35017	2	9	0.990106	10.5	766.51
76	64	15	70	2.87671	1.69609	2	8	0.967223	7.5	519.89
77	1	20	36	1.9726	1.40449	7	7	0.767765	2.85714	2.0274
78	34	22	57	3.43562	1.85354	1	7	0.916745	22	104.189
79	82	19	10	0.520548	0.72149	2	2	1	9.5	1310.32
80	29	21	89	5.12055	2.26286	2	8	0.0118221	10.5	159.5

Resultados del Algoritmo Genético para 90 productos.										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
1	50	22	81	4.88219	2.20957	1	8	0.52126	22	305.89
2	37	16	20	0.876712	0.936329	2	3	1	8	633.562
3	10	19	24	1.24932	1.11773	3	4	1	6.33333	172.507
4	66	20	42	2.30137	1.51703	1	5	1	20	706.11
5	56	20	86	4.71233	2.17079	1	8	0.999999	20	968.11
6	71	23	62	3.90685	1.97658	1	7	0.070693	23	390.5
7	51	21	62	3.56712	1.88868	4	6	0.99799	5.25	481.077
8	74	23	93	5.86027	2.4208	2	9	0.00774319	11.5	407
9	52	20	61	3.34247	1.82824	1	6	0.926972	20	320.192
10	85	24	10	0.657534	0.810885	4	2	1	6	1219.11
11	79	15	82	3.36986	1.83572	1	6	0.227765	15	118.5
12	46	20	64	3.50685	1.87266	1	7	0.210509	20	138
13	71	17	53	2.46849	1.57114	2	4	1	8.5	1244.74
14	43	19	34	1.76986	1.33036	1	4	1	19	633.396
15	75	21	33	1.89863	1.37791	1	4	0.998542	21	795.103
16	30	17	53	2.46849	1.57114	2	6	0.998038	8.5	255.945
17	88	21	30	1.72603	1.31378	2	3	0.999429	10.5	728.11
18	26	18	14	0.690411	0.83091	2	3	0.942498	9	138.049
19	7	23	25	1.57534	1.25513	5	10	1	4.6	111.473
20	68	16	33	1.44658	1.20274	2	3	0.983124	8	683.633
21	75	16	27	1.18356	1.08792	1	3	0.433005	16	525
22	84	24	64	4.20822	2.05139	1	7	0.058918	24	42
23	30	16	84	3.68219	1.9189	1	7	1	16	549.534
24	73	16	82	3.59452	1.89592	2	6	0.0855818	8	255.5
25	49	16	82	3.59452	1.89592	2	8	0.999636	8	681.368
26	42	15	73	3	1.73205	4	6	0.999973	3.75	420
27	74	19	87	4.52877	2.12809	1	8	0.982182	19	478.871
28	86	21	80	4.60274	2.1454	1	7	0.999718	21	1109.16
29	7	23	28	1.76438	1.3283	3	6	1	7.66667	110.149
30	64	22	99	5.96712	2.44277	1	9	0.112249	22	160
31	22	22	18	1.08493	1.0416	3	4	1	7.33333	317.132
32	36	19	68	3.53973	1.88142	1	7	0.904508	19	178.57
33	25	22	29	1.74795	1.3221	3	5	1	7.33333	418.801
34	75	24	31	2.03836	1.42771	1	4	0.999745	24	747.123
35	90	20	40	2.19178	1.48047	2	4	0.994949	10	702.74
36	59	18	27	1.33151	1.15391	1	3	0.925904	18	186.941
37	64	22	55	3.31507	1.82073	1	6	0.235063	22	384
38	84	17	95	4.42466	2.10349	1	7	1	17	1098.33
39	62	17	45	2.09589	1.44772	2	4	0.473595	8.5	124
40	52	19	62	3.2274	1.7965	4	6	0.982135	4.75	248.175
41	35	21	64	3.68219	1.9189	6	7	0.958097	3.5	326.123
42	58	21	74	4.25753	2.06338	1	8	0.997307	21	478.063
43	33	16	46	2.01644	1.42001	1	5	1	16	510.958
44	37	16	87	3.8137	1.95287	1	7	0.0254175	16	74
45	34	17	92	4.28493	2.07001	1	7	0.997118	17	279.312

Resultados del Algoritmo Genético para 90 productos.											
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i	
46	43	22	77	4.6411	2.15432	1	8	0.99842	22	380.933	
47	46	19	35	1.82192	1.34978	8	6	0.946699	2.375	376.192	
48	82	23	81	5.10411	2.25923	1	8	0.999959	23	1016.46	
49	19	24	22	1.44658	1.20274	4	4	1	6	238.515	
50	31	18	55	2.71233	1.64691	4	8	1	4.5	582.418	
51	52	15	64	2.63014	1.62177	4	10	0.999957	3.75	539.233	
52	44	20	73	4	2	2	8	0.99865	10	462	
53	24	22	44	2.65205	1.62851	1	6	1	22	368.351	
54	53	18	75	3.69863	1.92318	2	6	0.999927	9	492.973	
55	80	19	89	4.63288	2.15241	4	8	0.110623	4.75	480	
56	6	17	15	0.69863	0.835841	4	3	1	4.25	61.8082	
57	40	24	22	1.44658	1.20274	1	4	0.677291	24	162.137	
58	8	19	87	4.52877	2.12809	3	9	0.99882	6.33333	55.7699	
59	51	20	47	2.57534	1.60479	2	5	0.359978	10	51	
60	49	17	47	2.18904	1.47954	2	5	0.999426	8.5	382.737	
61	70	15	96	3.94521	1.98625	2	6	0.0690646	7.5	105	
62	21	15	77	3.16438	1.77887	4	7	0.0376304	3.75	94.5	
63	53	23	98	6.17534	2.48502	2	9	0.00647714	11.5	344.5	
64	42	22	31	1.86849	1.36693	2	5	0.262596	11	231	
65	78	16	18	0.789041	0.88828	2	3	1	8	991.455	
66	60	23	44	2.7726	1.66511	1	6	0.321326	23	360	
67	32	21	55	3.16438	1.77887	8	8	0.984466	2.625	314.74	
68	32	21	61	3.50959	1.87339	1	8	0.0305073	21	80	
69	23	24	97	6.37808	2.52549	5	12	0.986995	4.8	255.804	
70	86	15	54	2.21918	1.48969	1	4	1	15	1013.15	
71	27	19	46	2.39452	1.54742	2	5	0.999854	9.5	164.848	
72	15	19	99	5.15342	2.27012	3	10	0.473058	6.33333	60	
73	33	20	33	1.80822	1.3447	6	5	0.999087	3.33333	187.829	
74	67	15	69	2.83562	1.68393	2	5	0.969889	7.5	413.014	
75	64	21	96	5.52329	2.35017	1	8	0.997073	21	478.51	
76	64	15	70	2.87671	1.69609	1	6	0.528973	15	167.89	
77	1	20	36	1.9726	1.40449	9	8	0.244314	2.22222	7.5	
78	34	22	57	3.43562	1.85354	3	7	0.993102	7.33333	223.189	
79	82	19	10	0.520548	0.72149	1	2	1	19	1105.32	
80	29	21	89	5.12055	2.26286	4	9	0.174351	5.25	87	
81	54	17	46	2.14247	1.46372	1	5	1	17	883.307	
82	27	21	37	2.12877	1.45903	4	5	1	5.25	253.023	
83	12	18	95	4.68493	2.16447	2	10	0.557867	9	15.7808	
84	35	16	51	2.23562	1.4952	2	5	1	8	359.253	
85	74	18	35	1.72603	1.31378	4	4	1	4.5	1167.27	
86	89	17	17	0.791781	0.889821	1	2	0.999844	17	686.032	
87	78	17	62	2.88767	1.69931	1	5	0.99224	17	359.762	
88	8	21	23	1.32329	1.15034	4	8	1	5.25	149.414	
89	31	20	16	0.876712	0.936329	3	3	1	6.66667	453.322	
90	22	19	62	3.2274	1.7965	8	8	1	2.375	302.997	

Resultados del Algoritmo Genético para 100 productos.										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
1	50	22	81	4.88219	2.20957	1	10	0.997187	22	455.89
2	37	16	20	0.876712	0.936329	2	3	1	8	504.062
3	10	19	24	1.24932	1.11773	3	10	0.999989	6.33333	97.5068
4	66	20	42	2.30137	1.51703	2	4	0.677431	10	409.11
5	56	20	86	4.71233	2.17079	2	8	0.975875	10	660.11
6	71	23	62	3.90685	1.97658	1	7	0.999834	23	858.614
7	51	21	62	3.56712	1.88868	1	7	0.381984	21	76.5
8	74	23	93	5.86027	2.4208	1	9	0.221109	23	407
9	52	20	61	3.34247	1.82824	3	6	0.10005	6.66667	312
10	85	24	10	0.657534	0.810885	2	2	1	12	624.11
11	79	15	82	3.36986	1.83572	2	5	0.999999	7.5	1155.78
12	46	20	64	3.50685	1.87266	2	7	0.991788	10	252.685
13	71	17	53	2.46849	1.57114	8	5	1	2.125	854.237
14	43	19	34	1.76986	1.33036	4	4	0.992409	4.75	461.396
15	75	21	33	1.89863	1.37791	1	3	0.0841159	21	37.5
16	30	17	53	2.46849	1.57114	1	5	0.998038	17	315.945
17	88	21	30	1.72603	1.31378	1	4	1	21	1388.11
18	26	18	14	0.690411	0.83091	2	3	1	9	385.049
19	7	23	25	1.57534	1.25513	8	5	1	2.875	93.9726
20	68	16	33	1.44658	1.20274	2	4	0.355207	8	442
21	75	16	27	1.18356	1.08792	4	3	1	4	1148.73
22	84	24	64	4.20822	2.05139	1	7	0.990251	24	738.51
23	30	16	84	3.68219	1.9189	4	8	0.886454	4	294.534
24	73	16	82	3.59452	1.89592	1	7	0.376921	16	438
25	49	16	82	3.59452	1.89592	2	6	1	8	779.368
26	42	15	73	3	1.73205	2	6	1	7.5	462
27	74	19	87	4.52877	2.12809	2	8	0.994929	9.5	552.871
28	86	21	80	4.60274	2.1454	2	8	0.0159605	10.5	645
29	7	23	28	1.76438	1.3283	5	8	0.999999	4.6	92.6493
30	64	22	99	5.96712	2.44277	1	9	0.112249	22	96
31	22	22	18	1.08493	1.0416	5	3	0.81017	4.4	163.132
32	36	19	68	3.53973	1.88142	2	7	0.387105	9.5	54
33	25	22	29	1.74795	1.3221	2	6	1	11	431.301
34	75	24	31	2.03836	1.42771	2	4	1	12	1047.12
35	90	20	40	2.19178	1.48047	1	4	0.70744	20	702.74
36	59	18	27	1.33151	1.15391	1	3	1	18	865.441
37	64	22	55	3.31507	1.82073	1	8	0.999879	22	811.836
38	84	17	95	4.42466	2.10349	2	8	0.773047	8.5	552.329
39	62	17	45	2.09589	1.44772	1	4	0.996499	17	490.055
40	52	19	62	3.2274	1.7965	4	7	0.247235	4.75	286
41	35	21	64	3.68219	1.9189	4	8	0.0810906	5.25	122.5
42	58	21	74	4.25753	2.06338	2	7	0.136957	10.5	377
43	33	16	46	2.01644	1.42001	1	6	1	16	345.958
44	37	16	87	3.8137	1.95287	2	7	0.0254175	8	259
45	34	17	92	4.28493	2.07001	6	9	0.999987	2.83333	432.312
46	43	22	77	4.6411	2.15432	3	8	0.99842	7.33333	531.433
47	46	19	35	1.82192	1.34978	1	6	1	19	399.192
48	82	23	81	5.10411	2.25923	1	7	0.0119346	23	533
49	19	24	22	1.44658	1.20274	3	4	0.998434	8	162.515
50	31	18	55	2.71233	1.64691	1	6	0.149235	18	46.5

Resultados del Algoritmo Genético para 100 productos.										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
51	52	15	64	2.63014	1.62177	4	5	0.800853	3.75	357.233
52	44	20	73	4	2	1	7	0.0668072	20	22
53	24	22	44	2.65205	1.62851	2	8	1	11	308.351
54	53	18	75	3.69863	1.92318	2	7	0.562261	9	360.473
55	80	19	89	4.63288	2.15241	1	5	0.999999	19	1189.37
56	6	17	15	0.69863	0.835841	8	8	1	2.125	70.8082
57	40	24	22	1.44658	1.20274	1	4	0.999923	24	402.137
58	8	19	87	4.52877	2.12809	1	10	0.587623	19	15.7699
59	51	20	47	2.57534	1.60479	1	8	1	20	633.658
60	49	17	47	2.18904	1.47954	4	5	1	4.25	554.237
61	70	15	96	3.94521	1.98625	3	6	1	5	808.836
62	21	15	77	3.16438	1.77887	3	7	1	5	259.048
63	53	23	98	6.17534	2.48502	1	10	0.00647714	23	132.5
64	42	22	31	1.86849	1.36693	1	4	0.998746	22	257.523
65	78	16	18	0.789041	0.88828	4	3	1	4	991.455
66	60	23	44	2.7726	1.66511	4	6	0.321326	5.75	300
67	32	21	55	3.16438	1.77887	2	8	0.999939	10.5	282.74
68	32	21	61	3.50959	1.87339	8	7	0.0305073	2.625	16
69	23	24	97	6.37808	2.52549	4	12	0.850408	6	152.304
70	86	15	54	2.21918	1.48969	1	3	0.999997	15	1185.15
71	27	19	46	2.39452	1.54742	1	6	0.652206	19	56.8479
72	15	19	99	5.15342	2.27012	3	10	0.171413	6.33333	37.5
73	33	20	33	1.80822	1.3447	1	5	1	20	418.829
74	67	15	69	2.83562	1.68393	1	5	1	15	1183.51
75	64	21	96	5.52329	2.35017	2	9	0.141487	10.5	384
76	64	15	70	2.87671	1.69609	1	6	0.998739	15	455.89
77	1	20	36	1.9726	1.40449	12	9	1	1.66667	16.5274
78	34	22	57	3.43562	1.85354	1	7	1	22	410.189
79	82	19	10	0.520548	0.72149	2	2	0.999705	9.5	490.315
80	29	21	89	5.12055	2.26286	7	10	0.79689	3	83.5041
81	54	17	46	2.14247	1.46372	2	6	0.999969	8.5	397.307
82	27	21	37	2.12877	1.45903	1	6	1	21	455.523
83	12	18	95	4.68493	2.16447	4	10	0.937187	4.5	111.781
84	35	16	51	2.23562	1.4952	4	5	0.99928	4	271.753
85	74	18	35	1.72603	1.31378	2	4	1	9	1352.27
86	89	17	17	0.791781	0.889821	2	2	1	8.5	1353.53
87	78	17	62	2.88767	1.69931	1	5	1	17	1022.76
88	8	21	23	1.32329	1.15034	3	5	1	7	113.414
89	31	20	16	0.876712	0.936329	2	4	0.884867	10	65.8219
90	22	19	62	3.2274	1.7965	2	8	1	9.5	269.997
91	29	17	35	1.63014	1.27677	1	5	1	17	474.726
92	89	23	42	2.64658	1.62683	1	5	0.155736	23	267
93	50	23	75	4.72603	2.17394	1	8	0.999999	23	713.699
94	68	17	87	4.05205	2.01297	1	7	0.489685	17	34
95	76	21	41	2.3589	1.53587	3	5	0.407616	7	152
96	16	22	96	5.7863	2.40547	3	11	0.821286	7.33333	91.4192
97	82	24	95	6.24658	2.49931	1	9	0.999769	24	799.781
98	10	24	28	1.8411	1.35687	3	5	0.999928	8	91.589
99	68	18	79	3.89589	1.9738	1	6	0.521033	18	381.079
100	66	16	65	2.84932	1.68799	2	6	0.136633	8	264

Tablas de los resultados obtenidos de Simul8, sin considerar pedidos pendientes en el momento en que llega una nueva orden:

Resultados de Optimización-Simulación para 10 productos.					
i	Q_i	r_i	S_i	F_i	II_i
1	2	8	99.73684	11.7	335.733
2	2	8	100	8.9	276.276
3	7	8	100	2	87.516
4	2	7	98.71212	9.7	296.34
5	4	2	75.36428	4.7	131.32
6	2	8	96.69286	11.5	313.852
7	4	7	97.94045	5	283.624
8	2	7	92.70656	11.3	243.452
9	4	7	99.19872	4.9	289.9
10	7	7	99.18831	3.1	606.756

Resultados de Optimización-Simulación para 20 productos.					
i	Q_i	r_i	S_i	F_i	II_i
1	5	6	98.10924	4.4	327.334
2	5	5	99.375	3.3	224.763
3	7	6	100	2	69.316
4	6	7	100	2.9	457.05
5	8	6	92.24651	2.1	352.8
6	8	5	88.64071	2.5	426.58
7	6	8	98.92857	3.1	391.556
8	7	8	94.82107	2.8	388.168
9	5	8	100	3.9	372.84
10	5	5	90.71956	4.6	375.24
11	5	5	89.26515	3.3	454.464
12	6	7	97.66667	3.4	308.43
13	5	6	95.59594	3.1	425.34
14	7	6	98.46154	2.2	311.88
15	5	5	97.66926	4.2	410.193
16	6	8	100	2.7	537.152
17	7	5	98.5641	2.7	558.527
18	7	7	100	2.4	254.88
19	6	8	100	3.3	59.64
20	8	7	100	1.6	637.075

Resultados de Optimización-Simulación para 30 productos.					
i	Q_i	r_i	S_i	F_i	II_i
1	7	7	100	3.1	456,043
2	6	7	100	2.8	345,345
3	5	8	100	3	77,428
4	6	5	96.28507	2.9	341.22
5	6	6	92.70619	3	281.12
6	6	7	97.28571	3.6	439,824
7	6	6	96.11989	3.1	297,902
8	7	8	94.82107	2.8	386,474
9	6	7	99.23077	3.1	358.54
10	8	6	97.16236	2.7	611,004
11	7	5	88.59159	2.2	540,202
12	6	7	97.4127	3.4	311.65
13	7	5	71.86029	2.3	302,742
14	7	7	100	2.2	353,238
15	6	6	98.41667	3.4	505,155
16	5	8	100	3.2	243,232
17	7	5	98.5641	2.7	1,029,483
18	8	7	100	2	281.52
19	5	7	100	4.1	48.03
20	7	7	100	2	599.25
21	5	8	100	3.4	723.06
22	7	6	95.50422	3.3	516.6
23	6	8	98.85621	2.3	238,799
24	7	6	98.66667	1.9	482,448
25	6	5	92.0237	2.5	268,056
26	6	6	97.99499	2.3	272.16
27	6	7	94.77516	2.8	465,244
28	7	8	95.22609	2.8	682,212
29	7	6	99.23077	2.8	47.46
30	7	6	87.30278	2.9	344.23

Resultados de Optimización-Simulación para 40 productos.					
i	Q_i	r_i	S_i	F_i	II_i
1	5	8	99.375	4.4	430,846
2	8	8	100	1.8	428,274
3	5	9	100	3	87,048
4	7	6	97.26974	2.5	453.42
5	9	8	96.22543	1.9	479.08
6	8	8	97.16667	2.5	593,824
7	8	9	99.56522	2.3	492,954
8	7	7	91.09867	2.8	346,544
9	8	10	100	2.3	594.36
10	6	7	98.20775	3.8	544,452
11	6	8	98.66667	2.6	706,418
12	5	9	99.2711	4.2	377.43
13	7	7	75.28571	2.3	396,984
14	9	8	100	1.8	458.78
15	8	8	100	2.6	776,118
16	9	8	100	1.6	339,856
17	8	7	100	2.2	1,373,482
18	11	6	100	1.4	291,312
19	8	5	98.16667	2.2	46.26
20	9	11	100	1.4	957.95
21	5	8	100	3.4	720.72
22	11	11	99.58333	1.8	1101.45
23	8	4	80.28442	1.7	173,162
24	8	10	100	1.5	834,936
25	9	11	100	1.5	652,698
26	8	7	99.41176	1.4	378.56
27	7	10	100	2.5	700,589
28	9	2	68.91149	2.1	422,497
29	5	9	100	4.1	60.12
30	9	8	93.37502	2	504.6
31	7	6	100	2.8	174.6
32	9	6	90.84993	1.7	275,373
33	10	9	100	1.8	299,224
34	7	8	100	2.7	624,312
35	7	8	100	2.5	831.15
36	5	5	98.57843	3.2	303,456
37	8	8	100	2.3	593.34
38	9	9	99.16667	1.4	823,498
39	6	6	98.66667	2.7	438.62
40	6	5	92.78463	2.7	277,368

Resultados de Optimización-Simulación para 50 productos.					
i	Q_i	r_i	S_i	F_i	II_i
1	6	7	100	3.5	412.913
2	8	4	98.46154	1.8	259.413
3	7	4	100	2	50.752
4	8	5	94.66667	2	441.87
5	5	4	82.32709	3.5	188.44
6	5	5	90.18383	4.2	304.92
7	7	7	97.09565	2.7	380.908
8	7	5	83.92192	2.8	274.186
9	6	8	99.6	3.1	412.36
10	6	7	98.20775	3.8	544.452
11	6	6	91.54781	2.6	545.988
12	8	6	97.7573	2.4	348.91
13	5	5	67.861	3.1	216.84
14	5	6	100	3.5	251.99
15	7	4	92.31209	2.9	443.037
16	7	7	100	2.2	255.728
17	5	5	97.32353	3.7	831.296
18	7	4	100	2.4	169.488
19	7	7	100	2.8	55.32
20	6	7	100	2.3	581.825
21	6	7	100	2.8	691.236
22	6	8	98.28571	3.8	581
23	7	5	87.75021	2.3	175.406
24	7	8	100	1.9	625.632
25	7	7	97.32601	1.9	376.686
26	7	6	98.26797	1.9	302.4
27	7	8	98.94444	2.5	554.325
28	5	5	85.66084	4.2	372.599
29	7	6	99.23077	2.8	47.04
30	7	6	87.30278	2.9	344.23
31	8	5	100	2.6	164.4
32	8	7	95.94853	2.1	292.383
33	6	5	99.375	3.3	144.979
34	5	6	99.375	4.1	396.552
35	5	6	98.12932	3.7	510.3
36	6	5	100	2.5	343.677
37	7	7	99.28571	3	474.73
38	6	7	97.61029	2.3	495.482
39	7	4	91.32403	2.2	347.984
40	8	7	100	1.9	414.414
41	6	8	98.31798	3.5	264.438
42	6	4	83.27084	3	236.256
43	6	7	99.23077	2.4	248.848
44	6	8	96.88259	2.2	244.934
45	5	7	95.84307	3.8	209.2
46	6	5	91.7567	2.8	198.51
47	6	5	98.875	3	1000.296
48	8	7	96.77591	2.7	615.88
49	7	7	100	2.8	156.025
50	6	6	100	2.4	186.964

Resultados de Optimización-Simulación para 60 productos.

i	Q_i	r_i	S_i	F_i	II_i	i	Q_i	r_i	S_i	F_i	II_i
1	6	4	95.47103	3.5	268.541	31	5	8	100	4	189.4
2	5	6	100	3.3	267.036	32	6	6	92.95541	2.9	209.979
3	5	5	100	3	50.388	33	7	6	100	2.5	180.461
4	8	6	95.8022	2	501.27	34	5	6	99.375	4.1	393.432
5	6	8	96.93559	3	378.28	35	7	6	99.56522	2.5	648.45
6	5	6	93.5208	4.2	345.268	36	8	6	99.09091	1.8	453.549
7	6	7	98.47403	3.1	342.672	37	7	5	93.42955	3	368.3
8	8	8	95.7807	2.4	428.582	38	6	6	95.30964	2.3	426.816
9	5	7	98.32751	3.9	321.88	39	6	6	99.23077	2.7	440.44
10	5	7	97.82601	4.6	480.024	40	5	6	94.10487	3.6	274.092
11	1	8	98.63043	18.1	407.124	41	5	5	88.7033	4.4	168.656
12	6	6	95.45004	3.4	270.94	42	5	8	97.19565	3.6	368.736
13	7	7	75.28571	2.3	396.984	43	7	7	100	1.7	273.568
14	8	7	100	1.9	378.776	44	5	6	91.08866	3	159.856
15	6	7	100	3.4	577.269	45	8	7	96.33333	2.1	292
16	5	6	99.54545	3.2	180.048	46	7	7	99.44444	2.4	292.11
17	7	5	98.5641	2.7	1029.483	47	6	8	100	3	1503.812
18	6	4	96.88294	2.9	148.032	48	6	8	94.86916	3.9	569.244
19	6	7	100	3.3	52.95	49	7	5	99.16667	2.8	119.448
20	5	5	100	3	384.625	50	7	8	100	2	261.44
21	6	6	100	2.8	598.104	51	6	7	98.63636	2.4	399.744
22	6	6	93.17882	3.8	450.1	52	8	5	86.16345	2.2	273.02
23	5	6	87.88578	3.1	158.576	53	7	5	93.93662	2.5	132.98
24	5	6	96.38095	2.9	391.704	54	5	7	95.54884	3.2	310.17
25	8	8	98.88889	1.6	451.044	55	6	4	83.29129	3	341.431
26	7	5	97.2807	1.9	263.76	56	5	7	100	3.5	50.12
27	6	7	94.30148	2.8	466.411	57	7	5	98.20833	3.1	53.235
28	5	6	89.49117	4.2	419.634	58	5	5	85.54733	4	36.96
29	7	5	96.28041	2.8	41.28	59	7	7	99.16667	2.3	383.265
30	7	5	83.7031	2.9	307.98	60	6	7	100	2.3	353.952

Resultados de Optimización-Simulación para 70 productos.

i	Q_i	r_i	S_i	F_i	II_i	i	Q_i	r_i	S_i	F_i	II_i
1	6	6	100	3.5	355.482	36	7	6	99.16667	2.2	424.773
2	5	7	100	3.3	310.464	37	5	7	98.94444	4.2	386.57
3	5	7	100	3	68.328	38	6	4	85.02411	2.3	312.702
4	6	8	100	2.9	522.39	39	5	6	98.16667	3.3	386.568
5	6	5	87.53478	3	243.88	40	5	5	91.7717	3.6	235.599
6	5	7	96.65161	4.2	391.468	41	6	8	98.31798	3.5	264.438
7	7	7	97.09565	2.7	380.908	42	6	6	92.74053	3	309.12
8	6	7	90.30558	3.5	294.514	43	6	7	99.23077	2.4	248.848
9	4	6	97.12448	4.9	246.48	44	5	6	91.08866	3	159.856
10	7	7	99.18831	3.1	606.756	45	5	8	97.12542	3.8	246.6
11	6	5	87.12763	2.6	481.816	46	8	6	99.41176	2	281.19
12	5	6	95.43135	4.2	244.72	47	5	5	97.67174	3.9	820.108
13	7	7	75.28571	2.3	396.984	48	7	5	85.62248	3.3	443.22
14	7	6	98.46154	2.2	310.298	49	6	6	99.375	3.4	122.213
15	7	5	96.60131	2.9	490.875	50	5	5	97.375	3	134.332
16	7	8	100	2.2	288.64	51	5	6	96.19315	3.2	308.483
17	1	6	98.04259	21.5	385.061	52	4	7	93.94472	5.1	228.8
18	5	6	100	3.5	191.808	53	6	6	96.2381	3	131.236
19	6	6	100	3.3	45.9	54	5	8	98.52844	3.2	356.622
20	6	8	100	2.3	653.225	55	7	4	84.33704	2.4	379.742
21	7	7	100	2.3	778.752	56	6	6	99.33333	2.7	47.25
22	4	7	96.7028	6	423.5	57	5	7	100	4.4	60.27
23	7	6	92.15368	2.3	201.212	58	2	7	90.16206	10.8	31.626
24	6	6	97.56863	2.2	437.304	59	5	8	99.66667	3.4	351.39
25	5	6	95.80742	2.9	268.056	60	6	7	100	2.3	352.224
26	5	7	99.54545	2.8	281.12	61	7	7	97.33333	2	549.659
27	6	5	84.85617	2.8	372.273	62	5	5	94.31002	2.7	91.98
28	7	5	87.16837	2.8	489.982	63	7	6	86.2088	2.9	280.37
29	7	5	96.80672	2.8	40.71	64	6	7	100	3.8	335.778
30	4	5	82.96174	5.2	216.05	65	5	6	100	2.9	528.016
31	5	5	99.5	4	121.2	66	7	7	98.92045	3.1	462.753
32	5	7	97.03674	3.8	216.783	67	6	6	95.10615	3.7	202.768
33	7	5	100	2.5	156.392	68	6	6	90.72093	3.7	191.216
34	7	6	98.83333	2.7	487.032	69	7	6	84.90836	3	133.056
35	6	5	99.01176	3.2	507.15	70	7	6	100	1.7	630.873

Resultados de Optimización-Simulación para 80 productos.

i	Q_i	r_i	S_i	F_i	II_i	i	Q_i	r_i	S_i	F_i	II_i
1	9	10	100	2.2	664.202	41	8	8	97.60349	2.5	313.076
2	7	10	100	2.1	485.1	42	1	7	92.92042	20.5	124.476
3	9	8	100	1.5	102.804	43	7	10	100	1.7	369.152
4	7	8	98.82353	2.5	574.86	44	15	10	100	0.7	550.432
5	8	7	95.36324	2.1	400.4	45	9	11	100	1.8	469
6	10	5	90.58815	2	505.12	46	7	8	100	2.4	329.55
7	10	10	100	1.5	625.328	47	1	4	93.12879	21	222.288
8	5	8	93.14157	4.3	313.148	48	9	9	98.93333	2.5	815.596
9	6	5	95.65934	3.1	266.76	49	15	6	100	1.1	233.603
10	5	8	98.8711	4.6	546.576	50	8	8	100	1.6	278.124
11	8	7	97.20833	1.9	748.498	51	8	5	89.63816	1.5	359.145
12	7	8	99.375	2.8	405.49	52	7	7	95.04723	2.9	312.62
13	8	5	70.88095	1.9	339.021	53	10	9	100	1.7	260.401
14	1	8	100	19.2	187.354	54	11	8	100	1.4	595.938
15	9	0	65.70088	2.1	350.931	55	8	8	100	2	693.808
16	9	10	100	1.6	405.68	56	3	11	100	5.9	68.915
17	6	6	98.5641	3.1	1091.914	57	7	10	100	3.1	94.465
18	8	10	100	2	369.072	58	1	8	92.68955	22.1	23.016
19	9	13	100	2.1	106.68	59	8	4	92.3208	2	273.36
20	7	10	100	2	813.45	60	10	6	100	1.2	440.064
21	10	5	97.75	1.4	738.972	61	9	0	56.20303	1.3	240.038
22	7	7	97.9057	3.3	587.3	62	8	7	100	1.3	180.32
23	10	7	95.23016	1.3	290.037	63	10	11	100	2	609.96
24	8	8	100	1.5	685.368	64	15	5	100	1	557.72
25	10	5	95.14706	1.2	401.472	65	9	6	100	1.4	745.176
26	8	8	100	1.4	420.56	66	7	9	100	3.1	589.86
27	6	7	94.06771	2.8	465.244	67	11	9	100	1.8	425.448
28	8	7	91.60804	2.4	629.86	68	10	8	97.15354	2.1	338.352
29	15	7	100	1	90.03	69	10	8	92.82153	1.9	197.568
30	5	8	95.18074	4.1	351.77	70	7	5	98.16667	1.7	547.215
31	6	8	100	3.4	203.6	71	9	9	100	1.8	308.14
32	9	9	100	1.7	370.062	72	11	7	90.80455	1.8	153.655
33	1	6	99.65517	22	60.229	73	15	7	100	1.5	621.72
34	10	7	100	1.7	703.56	74	10	8	100	0.9	735.585
35	7	10	100	2.5	1022.4	75	7	5	84.34324	3	349.16
36	8	8	100	1.8	565.056	76	5	7	99.13043	2.8	415.044
37	7	7	99.28571	3	472.7	77	8	7	100	2.2	253.754
38	7	8	97.64706	1.6	629.356	78	9	9	100	2	341.775
39	8	6	100	2	500.864	79	1	4	99.12879	18	144.816
40	6	5	92.78871	2.7	275.73	80	10	7	97.10227	1.6	267.582

Resultados de Optimización-Simulación para 90 productos.

i	Q_i	r_i	S_i	F_i	II_i	i	Q_i	r_i	S_i	F_i	II_i
1	11	11	100	0.5	306	46	11	10	100	1.2	507.585
2	10	12	100	1.4	659.274	47	10	10	100	1.8	2246.456
3	12	11	100	1.1	148.2	48	11	11	99.62963	1.9	1105.38
4	11	10	100	1.3	844.14	49	10	10	100	1.9	240.16
5	11	11	100	1.4	729.68	50	12	10	100	1	426.732
6	11	11	100	1.6	961.884	51	10	10	100	1.4	725.577
7	11	11	100	1.4	711.722	52	11	10	100	1.5	594.66
8	10	11	98.875	1.9	639.364	53	12	10	100	1.2	314.901
9	10	10	100	1.5	658.84	54	11	10	100	1.4	704.424
10	10	11	100	2.1	1068.726	55	10	11	100	1.5	1036.081
11	12	11	100	1.1	1388.64	56	10	12	100	1.3	103.495
12	10	10	100	1.8	583.28	57	10	12	100	2	127.995
13	12	11	98.03571	1.1	1023.318	58	11	11	100	1.7	114.66
14	11	4	100	1.2	356.854	59	11	10	100	1.2	660.96
15	10	10	100	1.8	1066.002	60	11	10	100	1	657.504
16	10	11	100	1.3	454.608	61	11	11	100	1	1061.024
17	10	10	100	1.6	2152.822	62	11	10	100	0.9	287.14
18	10	10	100	1.5	388.368	63	12	12	100	1.5	729.33
19	11	10	100	1.8	93.63	64	11	10	100	1.9	630.3
20	11	9	100	1.1	954.55	65	11	11	100	1.1	1261.48
21	11	10	100	1.2	1257.516	66	10	10	100	2.1	803.358
22	12	11	100	1.6	1153.25	67	11	10	100	1.8	462.232
23	8	11	100	1.7	374.187	68	11	10	98.82353	1.8	428.792
24	11	10	100	1	990.888	69	11	11	99.56522	1.7	279.36
25	11	10	100	1.1	688.806	70	11	11	100	0.9	1344.258
26	10	10	100	1.3	569.52	71	10	11	100	1.5	366.928
27	10	12	100	1.6	1070.528	72	10	12	100	1.9	219.778
28	11	11	98.94737	1.6	1144.791	73	11	11	100	2.4	696.96
29	10	11	100	2	95.61	74	11	11	100	0.9	972.77
30	11	11	100	1.7	813.74	75	10	12	98.975	1.9	851.83
31	10	10	100	1.7	303	76	11	10	100	1	909.937
32	10	10	100	1.6	439.236	77	11	11	100	1.5	451.56
33	7	10	100	2.5	281.483	78	10	10	100	1.8	390.91
34	11	10	100	1.6	970.32	79	10	10	100	1.3	1093.447
35	11	11	100	1.4	1350.9	80	11	12	100	1.4	416.07
36	11	11	100	1.1	872.436	81	10	11	100	1.3	774.012
37	10	10	100	2	783	82	10	10	100	1.7	1138.881
38	11	11	100	1	1070.992	83	11	11	100	1.2	167.232
39	11	10	100	1.2	935.116	84	11	11	100	1.2	525.381
40	11	10	100	1.4	722.631	85	11	10	100	1.2	1013.937
41	10	10	99.375	1.8	433.26	86	10	10	100	1.5	1358.732
42	10	11	100	1.6	694.692	87	10	11	100	1.4	1140.156
43	11	10	100	1	445.166	88	11	12	100	1.6	126.008
44	10	10	100	1.3	399.022	89	10	11	100	1.4	452.135
45	10	10	100	1.8	473.8	90	10	11	100	1.8	317.84

Resultados de Optimización-Simulación para 100 productos.

i	Q_i	r_i	S_i	F_i	II_i	i	Q_i	r_i	S_i	F_i	II_i
1	9	11	100	2.2	718.909	51	13	8	99.09091	0.9	768.258
2	11	11	100	1	656.04	52	12	11	100	1.3	658.9
3	11	12	100	1.1	149.968	53	6	9	100	3	196.091
4	13	11	100	1	1039.83	54	10	10	100	1.4	663.264
5	13	11	99.44444	1.1	782.32	55	9	10	100	1.7	941.356
6	14	11	100	1.2	1072.764	56	10	8	100	1.3	76.615
7	10	12	100	1.5	729.388	57	12	7	100	1.4	94.78
8	12	11	100	1.4	721.886	58	12	11	100	1.4	122.976
9	5	8	100	3.9	385.84	59	13	11	100	1.1	767.295
10	12	9	100	1.7	1066.602	60	12	12	100	0.9	781.056
11	7	11	100	2.2	1045.162	61	9	12	100	1.3	1011.055
12	11	9	100	1.6	572.24	62	11	12	100	0.9	327.6
13	12	13	100	1.1	1201.377	63	10	11	100	2	609.96
14	10	10	100	1.6	575.848	64	11	13	100	1.9	777.37
15	12	12	100	1.6	1308.762	65	12	8	100	1	1081.896
16	11	11	100	1.3	484	66	12	10	100	1.7	841.986
17	12	12	100	1.2	2756.182	67	10	9	100	1.9	400.52
18	10	10	100	1.5	388.368	68	12	11	100	1.5	477.736
19	4	7	100	5.2	43.53	69	11	10	98.48335	1.7	256.896
20	7	10	100	2	813.45	70	7	12	100	1.7	1160.325
21	12	12	100	1.1	1477.008	71	10	11	100	1.5	368.348
22	10	13	100	2.2	1243.55	72	11	12	100	1.8	235.499
23	12	11	100	1	472.175	73	12	9	100	2.2	625.515
24	10	12	100	1.1	1092.12	74	12	10	100	0.9	966.095
25	12	12	100	1	830.79	75	11	12	99.64286	1.7	876.512
26	9	7	100	1.3	408.8	76	10	11	100	1.1	927.444
27	10	11	100	1.6	998.174	77	13	12	100	1.1	509.354
28	14	11	100	1.1	1354.608	78	14	12	100	1	585.59
29	13	11	100	1.2	109.56	79	12	10	100	1	1214.127
30	13	12	100	1.1	925.68	80	12	10	99.375	1.4	372.324
31	14	12	100	1.1	409.4	81	11	13	100	1.2	907.572
32	11	12	100	1.4	549.045	82	9	10	100	2.1	1083.807
33	10	13	100	1.8	398.664	83	14	11	100	0.9	193.898
34	12	8	100	1.4	863.928	84	13	10	100	0.7	545.091
35	11	13	100	1.4	1534.05	85	12	13	100	1.1	1286.43
36	11	13	100	1.1	988.194	86	11	13	100	1.1	1748.588
37	4	13	100	5.3	748.49	87	13	11	100	0.9	1260.414
38	12	12	100	0.9	1198.444	88	11	12	100	1.6	126.008
39	13	10	100	1.1	1021.02	89	12	9	100	1.2	442.835
40	8	13	100	1.9	742.56	90	11	13	100	1.4	385.584
41	10	12	100	1.8	507.462	91	12	10	100	1	475.038
42	7	10	100	2.5	556.14	92	10	10	100	2	1121.139
43	8	11	100	1.4	420.24	93	12	11	100	1.6	711.109
44	12	10	100	0.9	459.586	94	7	12	100	2.1	852
45	4	11	100	4.9	334.6	95	14	12	100	1	1289.806
46	13	10	100	1.1	546.78	96	12	12	100	1.3	227.468
47	12	9	100	1.2	2377.808	97	11	13	100	1.9	1213.758
48	7	10	99.41176	3.3	767.536	98	10	10	100	1.9	127.26
49	12	11	100	1.3	290.878	99	10	11	100	1.1	837.27
50	13	12	100	1	501.208	100	8	13	100	1.9	1081.234

Tablas de los resultados obtenidos de genéticos, considerando pedidos pendientes en el momento en que llega una nueva orden:

Resultados de Algoritmo Genético para 10 productos										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
1	50	22	81	4.88219	2.20957	3	8	0.999881	7.33333	530.89
2	37	16	20	0.876712	0.936329	2	3	1	8	467.062
3	10	19	24	1.24932	1.11773	4	4	1	4.75	167.507
4	66	20	42	2.30137	1.51703	2	5	0.962371	10	310.11
5	56	20	86	4.71233	2.17079	1	7	0.105747	20	1608.67
6	71	23	62	3.90685	1.97658	1	7	0.070693	23	825.019
7	51	21	62	3.56712	1.88868	2	7	1	10.5	787.077
8	74	23	93	5.86027	2.4208	1	9	0.361157	23	597.996
9	52	20	61	3.34247	1.82824	1	6	0.977281	20	216.192
10	85	24	10	0.657534	0.810885	2	2	0.66361	12	454.11

Resultados de Algoritmo Genético para 20 productos										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
1	50	22	81	4.88219	2.20957	3	8	0.52126	7.33333	155.89
2	37	16	20	0.876712	0.936329	1	3	0.988325	16	97.0616
3	10	19	24	1.24932	1.11773	4	4	0.94136	4.75	52.5068
4	66	20	42	2.30137	1.51703	1	5	0.19549	20	495.246
5	56	20	86	4.71233	2.17079	1	8	0.0436211	20	697.858
6	71	23	62	3.90685	1.97658	1	7	0.167342	23	1209.1
7	51	21	62	3.56712	1.88868	2	7	0.99967	10.5	455.577
8	74	23	93	5.86027	2.4208	1	9	0.99992	23	1194.34
9	52	20	61	3.34247	1.82824	1	7	1	20	632.192
10	85	24	10	0.657534	0.810885	1	2	1	24	751.61
11	79	15	82	3.36986	1.83572	1	5	0.812732	15	326.281
12	46	20	64	3.50685	1.87266	2	6	0.210509	10	404.187
13	71	17	53	2.46849	1.57114	2	5	1	8.5	641.237
14	43	19	34	1.76986	1.33036	3	4	1	6.33333	676.396
15	75	21	33	1.89863	1.37791	1	4	1	21	1057.6
16	30	17	53	2.46849	1.57114	4	5	0.998038	4.25	180.945
17	88	21	30	1.72603	1.31378	1	4	0.958261	21	332.11
18	26	18	14	0.690411	0.83091	3	3	1	6	463.049
19	7	23	25	1.57534	1.25513	5	5	0.973308	4.6	69.4726
20	68	16	33	1.44658	1.20274	1	5	0.901748	16	241.633

Resultados de Algoritmo Genético para 30 productos										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
1	50	22	81	4.88219	2.20957	4	8	0.999362	5.5	655.89
2	37	16	20	0.876712	0.936329	2	3	1	8	430.062
3	10	19	24	1.24932	1.11773	4	4	1	4.75	132.507
4	66	20	42	2.30137	1.51703	1	5	1	20	1069.11
5	56	20	86	4.71233	2.17079	1	8	0.371402	20	397.87
6	71	23	62	3.90685	1.97658	1	8	0.999979	23	1000.61
7	51	21	62	3.56712	1.88868	2	7	0.999958	10.5	710.577
8	74	23	93	5.86027	2.4208	1	9	0.983129	23	676.34
9	52	20	61	3.34247	1.82824	1	6	0.425705	20	322.698
10	85	24	10	0.657534	0.810885	1	2	1	24	1134.11
11	79	15	82	3.36986	1.83572	1	5	1	15	958.281
12	46	20	64	3.50685	1.87266	3	7	0.787374	6.66667	367.685
13	71	17	53	2.46849	1.57114	4	5	0.987703	4.25	463.737
14	43	19	34	1.76986	1.33036	2	4	0.0916995	9.5	354.164
15	75	21	33	1.89863	1.37791	1	4	0.998542	21	870.103
16	30	17	53	2.46849	1.57114	2	5	0.0580746	8.5	299.822
17	88	21	30	1.72603	1.31378	2	4	0.999429	10.5	728.11
18	26	18	14	0.690411	0.83091	2	3	1	9	359.049
19	7	23	25	1.57534	1.25513	5	5	1	4.6	93.9726
20	68	16	33	1.44658	1.20274	1	4	0.901748	16	615.633
21	75	16	27	1.18356	1.08792	2	3	1	8	923.733
22	84	24	64	4.20822	2.05139	2	8	0.999999	12	1452.51
23	30	16	84	3.68219	1.9189	2	8	0.75388	8	264.534
24	73	16	82	3.59452	1.89592	1	6	0.897738	16	650.1
25	49	16	82	3.59452	1.89592	1	6	0.999995	16	534.368
26	42	15	73	3	1.73205	1	6	1	15	693
27	74	19	87	4.52877	2.12809	1	7	0.994929	19	774.871
28	86	21	80	4.60274	2.1454	1	8	0.979799	21	636.164
29	7	23	28	1.76438	1.3283	5	5	0.28249	4.6	47.6288
30	64	22	99	5.96712	2.44277	1	10	0.892803	22	642.104

Resultados de Algoritmo Genético para 40 productos										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
1	50	22	81	4.88219	2.20957	1	8	0.0960462	22	3252.8
2	37	16	20	0.876712	0.936329	1	3	0.988325	16	263.562
3	10	19	24	1.24932	1.11773	3	4	1	6.33333	132.507
4	66	20	42	2.30137	1.51703	1	5	0.677431	20	508.11
5	56	20	86	4.71233	2.17079	1	8	0.72347	20	380.11
6	71	23	62	3.90685	1.97658	1	7	1	23	823.114
7	51	21	62	3.56712	1.88868	1	8	0.99799	21	659.577
8	74	23	93	5.86027	2.4208	1	9	0.118694	23	1403.73
9	52	20	61	3.34247	1.82824	2	6	0.425705	10	383.86
10	85	24	10	0.657534	0.810885	1	2	0.208716	24	505.342
11	79	15	82	3.36986	1.83572	3	5	0.6343	5	523.781
12	46	20	64	3.50685	1.87266	2	8	1	10	689.685
13	71	17	53	2.46849	1.57114	2	5	1	8.5	1173.74
14	43	19	34	1.76986	1.33036	2	4	0.2814	9.5	268.02
15	75	21	33	1.89863	1.37791	1	4	0.787943	21	157.603
16	30	17	53	2.46849	1.57114	2	5	0.998038	8.5	210.945
17	88	21	30	1.72603	1.31378	1	4	0.99997	21	1036.11
18	26	18	14	0.690411	0.83091	3	3	1	6	320.049
19	7	23	25	1.57534	1.25513	5	8	0.871827	4.6	20.4726
20	68	16	33	1.44658	1.20274	1	3	1	16	1057.63
21	75	16	27	1.18356	1.08792	2	4	0.999995	8	923.733
22	84	24	64	4.20822	2.05139	4	7	0.997624	6	654.51
23	30	16	84	3.68219	1.9189	8	7	0.0274983	2	428.49
24	73	16	82	3.59452	1.89592	4	5	0.0289852	4	816.8
25	49	16	82	3.59452	1.89592	1	7	0.997822	16	509.868
26	42	15	73	3	1.73205	2	6	0.5	7.5	147
27	74	19	87	4.52877	2.12809	2	8	0.877229	9.5	256.871
28	86	21	80	4.60274	2.1454	1	8	0.573452	21	292.164
29	7	23	28	1.76438	1.3283	5	8	0.999286	4.6	78.6493
30	64	22	99	5.96712	2.44277	1	9	0.999496	22	770.104
31	22	22	18	1.08493	1.0416	4	4	1	5.5	273.132
32	36	19	68	3.53973	1.88142	2	7	0.999963	9.5	286.57
33	25	22	29	1.74795	1.3221	3	5	0.828186	7.33333	181.301
34	75	24	31	2.03836	1.42771	4	5	0.749703	6	297.123
35	90	20	40	2.19178	1.48047	1	4	0.70744	20	747.74
36	59	18	27	1.33151	1.15391	4	4	1	4.5	894.941
37	64	22	55	3.31507	1.82073	1	8	0.929846	22	395.836
38	84	17	95	4.42466	2.10349	1	7	1	17	1392.33
39	62	17	45	2.09589	1.44772	1	4	1	17	769.055
40	52	19	62	3.2274	1.7965	1	8	0.938626	19	170.175

Resultados de Algoritmo Genético para 50 productos										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
1	50	22	81	4.88219	2.20957	1	8	0.999362	22	480.89
2	37	16	20	0.876712	0.936329	1	3	1	16	559.562
3	10	19	24	1.24932	1.11773	3	4	0.94136	6.33333	62.5068
4	66	20	42	2.30137	1.51703	1	5	1	20	1135.11
5	56	20	86	4.71233	2.17079	1	8	0.998113	20	660.11
6	71	23	62	3.90685	1.97658	2	7	0.855195	11.5	468.114
7	51	21	62	3.56712	1.88868	1	8	0.203342	21	439.675
8	74	23	93	5.86027	2.4208	1	9	0.361157	23	751.993
9	52	20	61	3.34247	1.82824	1	6	0.926972	20	502.192
10	85	24	10	0.657534	0.810885	1	2	1	24	1091.61
11	79	15	82	3.36986	1.83572	1	6	0.999999	15	1116.28
12	46	20	64	3.50685	1.87266	1	7	0.999997	20	712.685
13	71	17	53	2.46849	1.57114	2	5	0.835163	8.5	250.737
14	43	19	34	1.76986	1.33036	1	4	1	19	654.896
15	75	21	33	1.89863	1.37791	1	4	1	21	982.603
16	30	17	53	2.46849	1.57114	3	5	0.17498	5.66667	255.933
17	88	21	30	1.72603	1.31378	1	4	0.0944595	21	758.97
18	26	18	14	0.690411	0.83091	3	3	1	6	424.049
19	7	23	25	1.57534	1.25513	5	5	1	4.6	139.473
20	68	16	33	1.44658	1.20274	2	3	1	8	887.633
21	75	16	27	1.18356	1.08792	1	3	0.999995	16	473.733
22	84	24	64	4.20822	2.05139	2	7	0.650241	12	192.51
23	30	16	84	3.68219	1.9189	4	7	1	4	369.534
24	73	16	82	3.59452	1.89592	1	6	1	16	1124.6
25	49	16	82	3.59452	1.89592	1	6	0.999995	16	534.368
26	42	15	73	3	1.73205	4	6	1	3.75	630
27	74	19	87	4.52877	2.12809	1	9	0.0166648	19	979.496
28	86	21	80	4.60274	2.1454	1	8	0.0465482	21	1079.98
29	7	23	28	1.76438	1.3283	5	8	0.28249	4.6	48.2382
30	64	22	99	5.96712	2.44277	1	9	0.112249	22	765.309
31	22	22	18	1.08493	1.0416	3	4	0.999999	7.33333	207.132
32	36	19	68	3.53973	1.88142	2	7	0.967056	9.5	304.57
33	25	22	29	1.74795	1.3221	3	8	1	7.33333	343.801
34	75	24	31	2.03836	1.42771	4	4	0.749703	6	522.123
35	90	20	40	2.19178	1.48047	4	4	0.70744	5	297.74
36	59	18	27	1.33151	1.15391	1	4	1	18	835.941
37	64	22	55	3.31507	1.82073	2	6	1	11	1003.84
38	84	17	95	4.42466	2.10349	2	7	0.12452	8.5	777.23
39	62	17	45	2.09589	1.44772	1	4	0.733852	17	521.055
40	52	19	62	3.2274	1.7965	4	6	1	4.75	846.175
41	35	21	64	3.68219	1.9189	2	7	1	10.5	448.623
42	58	21	74	4.25753	2.06338	1	8	0.908095	21	478.063
43	33	16	46	2.01644	1.42001	2	7	1	8	642.958
44	37	16	87	3.8137	1.95287	2	8	0.999232	8	247.393
45	34	17	92	4.28493	2.07001	2	8	0.0562657	8.5	466.33
46	43	22	77	4.6411	2.15432	4	8	0.99842	5.5	574.433
47	46	19	35	1.82192	1.34978	1	4	0.999998	19	399.192
48	82	23	81	5.10411	2.25923	4	8	0.0346393	5.75	7792.8
49	19	24	22	1.44658	1.20274	3	4	1	8	295.515
50	31	18	55	2.71233	1.64691	3	6	0.999995	6	334.418

Resultados de Algoritmo Genético para 60 productos										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
1	50	22	81	4.88219	2.20957	1	8	0.999982	22	555.89
2	37	16	20	0.876712	0.936329	2	3	1	8	541.062
3	10	19	24	1.24932	1.11773	4	4	1	4.75	182.507
4	66	20	42	2.30137	1.51703	1	5	0.06463	20	3070.81
5	56	20	86	4.71233	2.17079	1	9	0.999933	20	492.11
6	71	23	62	3.90685	1.97658	1	8	0.999998	23	1071.61
7	51	21	62	3.56712	1.88868	4	8	0.999958	5.25	404.577
8	74	23	93	5.86027	2.4208	2	9	0.68111	11.5	602.34
9	52	20	61	3.34247	1.82824	2	6	1	10	892.192
10	85	24	10	0.657534	0.810885	1	3	1	24	1176.61
11	79	15	82	3.36986	1.83572	2	5	0.999848	7.5	721.281
12	46	20	64	3.50685	1.87266	4	8	1	5	528.685
13	71	17	53	2.46849	1.57114	1	4	0.382781	17	1057.87
14	43	19	34	1.76986	1.33036	2	4	1	9.5	590.396
15	75	21	33	1.89863	1.37791	2	4	0.529323	10.5	532.603
16	30	17	53	2.46849	1.57114	2	8	0.946437	8.5	210.945
17	88	21	30	1.72603	1.31378	2	4	0.582595	10.5	684.11
18	26	18	14	0.690411	0.83091	4	3	0.997279	4.5	216.049
19	7	23	25	1.57534	1.25513	5	5	0.999788	4.6	44.9726
20	68	16	33	1.44658	1.20274	2	3	1	8	989.633
21	75	16	27	1.18356	1.08792	1	3	1	16	886.233
22	84	24	64	4.20822	2.05139	2	8	0.650241	12	150.51
23	30	16	84	3.68219	1.9189	2	8	0.999932	8	294.534
24	73	16	82	3.59452	1.89592	1	6	0.997822	16	869.1
25	49	16	82	3.59452	1.89592	2	6	0.770749	8	117.868
26	42	15	73	3	1.73205	2	6	0.281851	7.5	333.525
27	74	19	87	4.52877	2.12809	1	8	0.948571	19	626.871
28	86	21	80	4.60274	2.1454	1	8	0.999718	21	1152.16
29	7	23	28	1.76438	1.3283	4	5	1	5.75	82.1493
30	64	22	99	5.96712	2.44277	2	9	0.505369	11	66.1041
31	22	22	18	1.08493	1.0416	1	4	0.999915	22	97.1315
32	36	19	68	3.53973	1.88142	4	7	1	4.75	628.57
33	25	22	29	1.74795	1.3221	2	5	0.955753	11	218.801
34	75	24	31	2.03836	1.42771	1	4	1	24	1009.62
35	90	20	40	2.19178	1.48047	2	4	0.999998	10	702.74
36	59	18	27	1.33151	1.15391	1	4	0.386945	18	299.337
37	64	22	55	3.31507	1.82073	2	6	0.999103	11	747.836
38	84	17	95	4.42466	2.10349	1	8	0.955408	17	468.329
39	62	17	45	2.09589	1.44772	1	4	0.999977	17	645.055
40	52	19	62	3.2274	1.7965	2	6	0.999344	9.5	352.175

Resultados de Algoritmo Genético para 60 productos										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
41	35	21	64	3.68219	1.9189	1	8	0.190341	21	1359.02
42	58	21	74	4.25753	2.06338	1	8	0.640513	21	130.063
43	33	16	46	2.01644	1.42001	1	5	1	16	642.958
44	37	16	87	3.8137	1.95287	4	8	0.868544	4	210.393
45	34	17	92	4.28493	2.07001	2	8	0.999999	8.5	568.312
46	43	22	77	4.6411	2.15432	1	8	0.38301	22	602.339
47	46	19	35	1.82192	1.34978	1	4	1	19	629.192
48	82	23	81	5.10411	2.25923	2	8	0.957685	11.5	688.463
49	19	24	22	1.44658	1.20274	4	4	1	6	362.015
50	31	18	55	2.71233	1.64691	2	5	0.977047	9	117.418
51	52	15	64	2.63014	1.62177	2	5	1	7.5	539.233
52	44	20	73	4	2	8	8	0.999968	2.5	616
53	24	22	44	2.65205	1.62851	3	8	1	7.33333	224.351
54	53	18	75	3.69863	1.92318	4	6	0.562261	4.5	333.973
55	80	19	89	4.63288	2.15241	1	8	0.99969	19	1109.37
56	6	17	15	0.69863	0.835841	4	6	1	4.25	121.808
57	40	24	22	1.44658	1.20274	4	4	0.999998	6	282.137
58	8	19	87	4.52877	2.12809	3	10	0.999777	6.33333	103.77
59	51	20	47	2.57534	1.60479	4	5	0.0542703	5	2652.35
60	49	17	47	2.18904	1.47954	1	3	0.999998	17	627.737

Resultados de Algoritmo Genético para 70 productos										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
1	50	22	81	4.88219	2.20957	1	8	0.999982	22	530.89
2	37	16	20	0.876712	0.936329	1	4	1	16	652.062
3	10	19	24	1.24932	1.11773	3	8	1	6.33333	142.507
4	66	20	42	2.30137	1.51703	4	5	1	5	937.11
5	56	20	86	4.71233	2.17079	1	8	0.935051	20	492.11
6	71	23	62	3.90685	1.97658	2	7	0.855195	11.5	326.114
7	51	21	62	3.56712	1.88868	2	7	1	10.5	914.577
8	74	23	93	5.86027	2.4208	3	9	0.902681	7.66667	417.34
9	52	20	61	3.34247	1.82824	2	6	1	10	866.192
10	85	24	10	0.657534	0.810885	2	2	1	12	581.61
11	79	15	82	3.36986	1.83572	5	5	1	3	1392.78
12	46	20	64	3.50685	1.87266	1	8	0.998323	20	321.685
13	71	17	53	2.46849	1.57114	1	5	0.987703	17	286.237
14	43	19	34	1.76986	1.33036	1	4	0.999263	19	246.396
15	75	21	33	1.89863	1.37791	1	3	0.529323	21	457.603
16	30	17	53	2.46849	1.57114	8	6	0.835163	2.125	105.945
17	88	21	30	1.72603	1.31378	4	4	1	5.25	1476.11
18	26	18	14	0.690411	0.83091	1	4	1	18	307.049
19	7	23	25	1.57534	1.25513	3	5	1	7.66667	125.473
20	68	16	33	1.44658	1.20274	1	3	1	16	887.633
21	75	16	27	1.18356	1.08792	1	3	1	16	1373.73
22	84	24	64	4.20822	2.05139	1	8	0.997624	24	906.51
23	30	16	84	3.68219	1.9189	1	7	0.999932	16	339.534
24	73	16	82	3.59452	1.89592	2	6	0.0289852	8	794.989
25	49	16	82	3.59452	1.89592	2	6	0.999636	8	534.368
26	42	15	73	3	1.73205	4	6	0.124107	3.75	341.228
27	74	19	87	4.52877	2.12809	1	7	0.948571	19	552.871
28	86	21	80	4.60274	2.1454	8	8	0.112532	2.625	922.565
29	7	23	28	1.76438	1.3283	3	5	0.0920393	7.66667	71.4377
30	64	22	99	5.96712	2.44277	1	9	0.505369	22	194.104
31	22	22	18	1.08493	1.0416	8	4	1	2.75	295.132
32	36	19	68	3.53973	1.88142	2	7	1	9.5	556.57
33	25	22	29	1.74795	1.3221	3	4	0.999964	7.33333	206.301
34	75	24	31	2.03836	1.42771	2	3	0.0766877	12	786.649
35	90	20	40	2.19178	1.48047	2	4	0.999998	10	972.74
36	59	18	27	1.33151	1.15391	1	4	1	18	776.941
37	64	22	55	3.31507	1.82073	1	6	0.101775	22	684.484
38	84	17	95	4.42466	2.10349	2	7	0.12452	8.5	868.154
39	62	17	45	2.09589	1.44772	1	6	1	17	1048.05
40	52	19	62	3.2274	1.7965	4	6	0.938626	4.75	170.175

Resultados de Algoritmo Genético para 70 productos										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
41	35	21	64	3.68219	1.9189	4	8	0.190341	5.25	355.804
42	58	21	74	4.25753	2.06338	2	8	0.999912	10.5	623.063
43	33	16	46	2.01644	1.42001	1	5	1	16	510.958
44	37	16	87	3.8137	1.95287	1	8	0.999999	16	432.393
45	34	17	92	4.28493	2.07001	1	7	0.999999	17	364.312
46	43	22	77	4.6411	2.15432	1	8	0.735908	22	144.433
47	46	19	35	1.82192	1.34978	4	4	1	4.75	537.192
48	82	23	81	5.10411	2.25923	1	8	0.984885	23	975.463
49	19	24	22	1.44658	1.20274	3	6	0.355207	8	110.924
50	31	18	55	2.71233	1.64691	1	6	0.999338	18	241.418
51	52	15	64	2.63014	1.62177	2	4	0.0524263	7.5	465.151
52	44	20	73	4	2	1	7	1	20	682
53	24	22	44	2.65205	1.62851	8	8	0.996206	2.75	224.351
54	53	18	75	3.69863	1.92318	2	7	0.884278	9	466.473
55	80	19	89	4.63288	2.15241	1	8	0.567717	19	149.37
56	6	17	15	0.69863	0.835841	8	4	1	2.125	88.8082
57	40	24	22	1.44658	1.20274	1	4	0.998434	24	222.137
58	8	19	87	4.52877	2.12809	4	10	1	4.75	123.77
59	51	20	47	2.57534	1.60479	1	5	0.163136	20	423.45
60	49	17	47	2.18904	1.47954	4	6	0.889524	4.25	456.237
61	70	15	96	3.94521	1.98625	4	8	0.317082	3.75	562.265
62	21	15	77	3.16438	1.77887	8	8	0.111856	1.875	178.69
63	53	23	98	6.17534	2.48502	1	10	0.629999	23	123.207
64	42	22	31	1.86849	1.36693	1	6	0.989016	22	215.523
65	78	16	18	0.789041	0.88828	4	2	0.993595	4	679.455
66	60	23	44	2.7726	1.66511	2	6	0.999153	11.5	373.644
67	32	21	55	3.16438	1.77887	4	8	0.111856	5.25	313.356
68	32	21	61	3.50959	1.87339	4	8	0.786859	5.25	287.693
69	23	24	97	6.37808	2.52549	4	11	0.292646	6	1003.74
70	86	15	54	2.21918	1.48969	1	3	0.999335	15	841.151

Resultados de Algoritmo Genético para 80 productos										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
1	50	22	81	4.88219	2.20957	5	8	0.693535	4.4	280.89
2	37	16	20	0.876712	0.936329	4	2	0.174552	4	203.753
3	10	19	24	1.24932	1.11773	8	4	1	2.375	102.507
4	66	20	42	2.30137	1.51703	1	4	0.677431	20	442.11
5	56	20	86	4.71233	2.17079	2	8	0.999999	10	800.11
6	71	23	62	3.90685	1.97658	8	7	0.999834	2.875	716.614
7	51	21	62	3.56712	1.88868	2	7	1	10.5	736.077
8	74	23	93	5.86027	2.4208	1	8	0.221109	23	730.023
9	52	20	61	3.34247	1.82824	2	6	0.425705	10	408.583
10	85	24	10	0.657534	0.810885	4	2	1	6	666.61
11	79	15	82	3.36986	1.83572	1	8	1	15	1195.28
12	46	20	64	3.50685	1.87266	4	7	0.0305578	5	621.9
13	71	17	53	2.46849	1.57114	1	5	0.0580746	17	650.552
14	43	19	34	1.76986	1.33036	1	3	0.56867	19	138.896
15	75	21	33	1.89863	1.37791	2	4	0.529323	10.5	420.103
16	30	17	53	2.46849	1.57114	2	5	0.998038	8.5	255.945
17	88	21	30	1.72603	1.31378	1	4	1	21	904.11
18	26	18	14	0.690411	0.83091	1	3	1	18	307.049
19	7	23	25	1.57534	1.25513	1	10	0.323335	23	52.4823
20	68	16	33	1.44658	1.20274	8	4	0.983124	2	683.633
21	75	16	27	1.18356	1.08792	2	4	0.138316	8	477.534
22	84	24	64	4.20822	2.05139	1	8	0.80879	24	360.51
23	30	16	84	3.68219	1.9189	1	7	0.997208	16	219.534
24	73	16	82	3.59452	1.89592	1	6	0.999995	16	978.6
25	49	16	82	3.59452	1.89592	1	6	0.200167	16	382.061
26	42	15	73	3	1.73205	2	6	0.989539	7.5	231
27	74	19	87	4.52877	2.12809	2	7	0.117362	9.5	831.088
28	86	21	80	4.60274	2.1454	1	8	0.742568	21	378.164
29	7	23	28	1.76438	1.3283	10	8	0.953819	2.3	19.1493
30	64	22	99	5.96712	2.44277	2	9	0.999891	11	642.104
31	22	22	18	1.08493	1.0416	2	4	0.999915	11	207.132
32	36	19	68	3.53973	1.88142	1	7	0.206568	19	312.928
33	25	22	29	1.74795	1.3221	1	6	1	22	393.801
34	75	24	31	2.03836	1.42771	1	4	0.915276	24	522.123
35	90	20	40	2.19178	1.48047	2	4	0.999956	10	1152.74
36	59	18	27	1.33151	1.15391	1	3	0.718816	18	216.441
37	64	22	55	3.31507	1.82073	2	6	0.235063	11	1125.75
38	84	17	95	4.42466	2.10349	2	7	0.999977	8.5	1224.33
39	62	17	45	2.09589	1.44772	2	4	0.999647	8.5	583.055
40	52	19	62	3.2274	1.7965	1	6	0.999999	19	586.175

Resultados de Algoritmo Genético para 80 productos										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
41	35	21	64	3.68219	1.9189	4	8	0.999503	5.25	256.123
42	58	21	74	4.25753	2.06338	4	8	0.999912	5.25	768.063
43	33	16	46	2.01644	1.42001	4	5	1	4	411.958
44	37	16	87	3.8137	1.95287	4	6	0.999999	4	580.393
45	34	17	92	4.28493	2.07001	5	7	0.999987	3.4	347.312
46	43	22	77	4.6411	2.15432	4	8	0.978481	5.5	402.433
47	46	19	35	1.82192	1.34978	2	8	0.946699	9.5	422.192
48	82	23	81	5.10411	2.25923	1	8	0.312523	23	805.374
49	19	24	22	1.44658	1.20274	1	4	1	24	295.515
50	31	18	55	2.71233	1.64691	8	6	1	2.25	427.418
51	52	15	64	2.63014	1.62177	8	5	0.0524263	1.875	526.767
52	44	20	73	4	2	1	8	0.0668072	20	533.445
53	24	22	44	2.65205	1.62851	2	12	1	11	260.351
54	53	18	75	3.69863	1.92318	1	5	0.956976	18	307.473
55	80	19	89	4.63288	2.15241	2	7	0.941132	9.5	389.37
56	6	17	15	0.69863	0.835841	2	8	1	8.5	100.808
57	40	24	22	1.44658	1.20274	8	5	1	3	582.137
58	8	19	87	4.52877	2.12809	5	12	1	3.8	99.7699
59	51	20	47	2.57534	1.60479	2	5	1	10	863.158
60	49	17	47	2.18904	1.47954	1	7	1	17	872.737
61	70	15	96	3.94521	1.98625	2	8	0.998849	7.5	458.836
62	21	15	77	3.16438	1.77887	1	7	0.999995	15	238.048
63	53	23	98	6.17534	2.48502	1	10	0.471874	23	1322.76
64	42	22	31	1.86849	1.36693	8	5	0.999996	2.75	278.523
65	78	16	18	0.789041	0.88828	2	2	1	8	835.455
66	60	23	44	2.7726	1.66511	2	6	1	11.5	853.644
67	32	21	55	3.16438	1.77887	1	9	0.256375	21	296.265
68	32	21	61	3.50959	1.87339	1	8	0.999968	21	303.693
69	23	24	97	6.37808	2.52549	2	11	0.99968	12	313.304
70	86	15	54	2.21918	1.48969	1	4	1	15	1185.15
71	27	19	46	2.39452	1.54742	4	6	0.99999	4.75	340.348
72	15	19	99	5.15342	2.27012	8	10	0.998719	2.375	155.199
73	33	20	33	1.80822	1.3447	4	5	1	5	517.829
74	67	15	69	2.83562	1.68393	1	5	0.969889	15	245.514
75	64	21	96	5.52329	2.35017	2	9	0.999972	10.5	894.51
76	64	15	70	2.87671	1.69609	1	5	0.992473	15	519.89
77	1	20	36	1.9726	1.40449	9	8	0.984439	2.22222	9.0274
78	34	22	57	3.43562	1.85354	1	9	0.407097	22	251.164
79	82	19	10	0.520548	0.72149	2	2	1	9.5	1228.32
80	29	21	89	5.12055	2.26286	8	10	0.999994	2.625	388.004

Resultados de Algoritmo Genético para 90 productos										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
1	50	22	81	4.88219	2.20957	1	8	0.52126	22	30.8904
2	37	16	20	0.876712	0.936329	4	8	0.552378	4	226.562
3	10	19	24	1.24932	1.11773	1	7	0.411746	19	64.2585
4	66	20	42	2.30137	1.51703	2	4	0.19549	10	495.281
5	56	20	86	4.71233	2.17079	2	8	0.992571	10	492.11
6	71	23	62	3.90685	1.97658	2	7	0.999834	11.5	645.614
7	51	21	62	3.56712	1.88868	4	8	0.0294672	5.25	1349.96
8	74	23	93	5.86027	2.4208	1	9	0.902681	23	454.34
9	52	20	61	3.34247	1.82824	1	14	0.425705	20	451.172
10	85	24	10	0.657534	0.810885	1	2	1	24	1261.61
11	79	15	82	3.36986	1.83572	1	5	0.976008	15	721.281
12	46	20	64	3.50685	1.87266	1	8	0.999969	20	436.685
13	71	17	53	2.46849	1.57114	2	4	0.835163	8.5	215.237
14	43	19	34	1.76986	1.33036	3	4	1	6.33333	633.396
15	75	21	33	1.89863	1.37791	1	3	0.529323	21	232.603
16	30	17	53	2.46849	1.57114	2	6	0.835163	8.5	225.945
17	88	21	30	1.72603	1.31378	1	4	0.0944595	21	750.712
18	26	18	14	0.690411	0.83091	4	3	0.942498	4.5	99.0493
19	7	23	25	1.57534	1.25513	2	6	0.323335	11.5	45.4941
20	68	16	33	1.44658	1.20274	1	6	1	16	1397.63
21	75	16	27	1.18356	1.08792	1	3	1	16	1223.73
22	84	24	64	4.20822	2.05139	1	7	0.999535	24	990.51
23	30	16	84	3.68219	1.9189	4	7	0.190341	4	279.024
24	73	16	82	3.59452	1.89592	2	8	1	8	1270.6
25	49	16	82	3.59452	1.89592	8	6	0.770749	2	362.868
26	42	15	73	3	1.73205	1	6	0.998054	15	315
27	74	19	87	4.52877	2.12809	2	8	0.587623	9.5	404.871
28	86	21	80	4.60274	2.1454	1	8	0.999955	21	1367.16
29	7	23	28	1.76438	1.3283	8	5	0.0920393	2.875	63.5082
30	64	22	99	5.96712	2.44277	4	9	0.892803	5.5	546.104
31	22	22	18	1.08493	1.0416	4	3	1	5.5	229.132
32	36	19	68	3.53973	1.88142	1	8	0.991123	19	250.57
33	25	22	29	1.74795	1.3221	10	8	1	2.2	331.301
34	75	24	31	2.03836	1.42771	4	4	0.489284	6	450.364
35	90	20	40	2.19178	1.48047	2	4	0.448465	10	603.013
36	59	18	27	1.33151	1.15391	1	3	0.999262	18	275.441
37	64	22	55	3.31507	1.82073	2	8	0.999988	11	875.836
38	84	17	95	4.42466	2.10349	1	7	0.995982	17	972.329
39	62	17	45	2.09589	1.44772	1	3	0.733852	17	87.0548
40	52	19	62	3.2274	1.7965	1	6	0.666424	19	144.175
41	35	21	64	3.68219	1.9189	4	8	0.0274983	5.25	445.641
42	58	21	74	4.25753	2.06338	1	8	0.271112	21	470.613
43	33	16	46	2.01644	1.42001	2	8	0.237059	8	287.965
44	37	16	87	3.8137	1.95287	1	7	0.999999	16	506.393
45	34	17	92	4.28493	2.07001	1	7	0.905176	17	126.312

Resultados de Algoritmo Genético para 90 productos										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
46	43	22	77	4.6411	2.15432	1	8	0.735908	22	273.433
47	46	19	35	1.82192	1.34978	2	3	1	9.5	652.192
48	82	23	81	5.10411	2.25923	2	8	0.957685	11.5	483.463
49	19	24	22	1.44658	1.20274	2	4	0.901748	12	134.015
50	31	18	55	2.71233	1.64691	8	6	0.332681	2.25	245.502
51	52	15	64	2.63014	1.62177	4	8	0.999535	3.75	565.233
52	44	20	73	4	2	1	8	1	20	616
53	24	22	44	2.65205	1.62851	1	8	1	22	368.351
54	53	18	75	3.69863	1.92318	1	8	0.562261	18	68.9726
55	80	19	89	4.63288	2.15241	1	8	0.0457232	19	5576.76
56	6	17	15	0.69863	0.835841	1	4	1	17	58.8082
57	40	24	22	1.44658	1.20274	8	4	1	3	422.137
58	8	19	87	4.52877	2.12809	8	10	0.999966	2.375	111.77
59	51	20	47	2.57534	1.60479	1	8	1	20	786.658
60	49	17	47	2.18904	1.47954	1	5	1	17	652.237
61	70	15	96	3.94521	1.98625	2	6	0.994534	7.5	633.836
62	21	15	77	3.16438	1.77887	4	8	1	3.75	311.548
63	53	23	98	6.17534	2.48502	1	10	0.046459	23	806.33
64	42	22	31	1.86849	1.36693	1	5	0.538322	22	26.5233
65	78	16	18	0.789041	0.88828	4	2	1	4	601.455
66	60	23	44	2.7726	1.66511	1	6	0.769477	23	133.644
67	32	21	55	3.16438	1.77887	8	7	0.999482	2.625	202.74
68	32	21	61	3.50959	1.87339	4	8	0.210177	5.25	315.012
69	23	24	97	6.37808	2.52549	2	12	0.986995	12	232.804
70	86	15	54	2.21918	1.48969	1	4	1	15	1099.15
71	27	19	46	2.39452	1.54742	2	5	0.999854	9.5	232.348
72	15	19	99	5.15342	2.27012	8	10	0.895067	2.375	117.699
73	33	20	33	1.80822	1.3447	8	5	1	2.5	484.829
74	67	15	69	2.83562	1.68393	2	5	0.99999	7.5	647.514
75	64	21	96	5.52329	2.35017	1	8	0.930476	21	670.51
76	64	15	70	2.87671	1.69609	2	5	1	7.5	967.89
77	1	20	36	1.9726	1.40449	10	8	0.507782	2	7.5274
78	34	22	57	3.43562	1.85354	4	8	1	5.5	546.189
79	82	19	10	0.520548	0.72149	2	2	1	9.5	1187.32
80	29	21	89	5.12055	2.26286	4	9	0.0118221	5.25	442.992
81	54	17	46	2.14247	1.46372	1	4	0.0716357	17	1037.39
82	27	21	37	2.12877	1.45903	2	6	1	10.5	320.523
83	12	18	95	4.68493	2.16447	10	10	0.999939	1.8	135.781
84	35	16	51	2.23562	1.4952	4	5	0.0674314	4	306.313
85	74	18	35	1.72603	1.31378	2	3	1	9	834.274
86	89	17	17	0.791781	0.889821	1	3	1	17	1309.03
87	78	17	62	2.88767	1.69931	4	5	1	4.25	1100.76
88	8	21	23	1.32329	1.15034	12	5	1	1.75	137.414
89	31	20	16	0.876712	0.936329	4	4	1	5	298.322
90	22	19	62	3.2274	1.7965	8	7	0.838105	2.375	82.9973

Resultados de Algoritmo Genético para 100 productos										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
1	50	22	81	4.88219	2.20957	1	8	0.0135676	22	1865.14
2	37	16	20	0.876712	0.936329	8	4	1	2	559.562
3	10	19	24	1.24932	1.11773	4	8	0.749086	4.75	12.5068
4	66	20	42	2.30137	1.51703	2	5	0.868581	10	343.11
5	56	20	86	4.71233	2.17079	1	6	0.0149735	20	1047.63
6	71	23	62	3.90685	1.97658	1	8	0.999834	23	752.114
7	51	21	62	3.56712	1.88868	1	8	0.90115	21	277.077
8	74	23	93	5.86027	2.4208	2	9	0.994397	11.5	1009.34
9	52	20	61	3.34247	1.82824	2	8	1	10	632.192
10	85	24	10	0.657534	0.810885	4	2	0.208716	6	589.443
11	79	15	82	3.36986	1.83572	1	8	0.99417	15	760.781
12	46	20	64	3.50685	1.87266	1	8	0.787374	20	160.685
13	71	17	53	2.46849	1.57114	1	5	1	17	1244.74
14	43	19	34	1.76986	1.33036	2	4	1	9.5	676.396
15	75	21	33	1.89863	1.37791	2	5	0.936376	10.5	345.103
16	30	17	53	2.46849	1.57114	4	5	0.987703	4.25	210.945
17	88	21	30	1.72603	1.31378	1	4	1	21	992.11
18	26	18	14	0.690411	0.83091	3	3	1	6	268.049
19	7	23	25	1.57534	1.25513	4	5	0.104716	5.75	59.7038
20	68	16	33	1.44658	1.20274	4	3	1	4	581.633
21	75	16	27	1.18356	1.08792	2	3	1	8	1298.73
22	84	24	64	4.20822	2.05139	1	8	0.650241	24	234.51
23	30	16	84	3.68219	1.9189	2	7	0.958097	8	234.534
24	73	16	82	3.59452	1.89592	2	6	1	8	1197.6
25	49	16	82	3.59452	1.89592	1	6	0.96377	16	215.868
26	42	15	73	3	1.73205	2	6	0.999734	7.5	441
27	74	19	87	4.52877	2.12809	2	8	0.236263	9.5	609.452
28	86	21	80	4.60274	2.1454	1	7	0.999999	21	1281.16
29	7	23	28	1.76438	1.3283	4	9	1	5.75	82.1493
30	64	22	99	5.96712	2.44277	1	9	0.999891	22	994.104
31	22	22	18	1.08493	1.0416	4	4	1	5.5	240.132
32	36	19	68	3.53973	1.88142	4	9	0.991123	4.75	286.57
33	25	22	29	1.74795	1.3221	7	4	1	3.14286	306.301
34	75	24	31	2.03836	1.42771	4	3	1	6	934.623
35	90	20	40	2.19178	1.48047	1	4	1	20	1557.74
36	59	18	27	1.33151	1.15391	2	3	0.925904	9	393.441
37	64	22	55	3.31507	1.82073	2	7	0.978508	11	395.836
38	84	17	95	4.42466	2.10349	1	8	0.98519	17	720.329
39	62	17	45	2.09589	1.44772	4	4	0.996499	4.25	583.055
40	52	19	62	3.2274	1.7965	1	6	0.982135	19	586.175
41	35	21	64	3.68219	1.9189	8	9	0.361103	2.625	251.208
42	58	21	74	4.25753	2.06338	2	8	0.271112	10.5	1001.26
43	33	16	46	2.01644	1.42001	4	4	1	4	510.958
44	37	16	87	3.8137	1.95287	2	7	0.999232	8	376.893
45	34	17	92	4.28493	2.07001	2	7	0.988631	8.5	415.312
46	43	22	77	4.6411	2.15432	2	8	0.999999	11	617.433
47	46	19	35	1.82192	1.34978	4	4	1	4.75	606.192
48	82	23	81	5.10411	2.25923	2	8	0.0847256	11.5	1069.96
49	19	24	22	1.44658	1.20274	2	6	0.901748	12	153.015
50	31	18	55	2.71233	1.64691	4	6	0.999338	4.5	194.918

Resultados de Algoritmo Genético para 100 productos										
i	c_i	D_i	l_i	θ_i	σ_i	Q_i	r_i	S_i	F_i	II_i
51	52	15	64	2.63014	1.62177	1	5	0.999997	15	487.233
52	44	20	73	4	2	4	7	0.841345	5	154
53	24	22	44	2.65205	1.62851	1	8	0.925317	22	104.351
54	53	18	75	3.69863	1.92318	4	7	1	4.5	810.973
55	80	19	89	4.63288	2.15241	1	7	0.993676	19	509.37
56	6	17	15	0.69863	0.835841	2	4	1	8.5	46.8082
57	40	24	22	1.44658	1.20274	4	4	0.999923	6	242.137
58	8	19	87	4.52877	2.12809	1	11	0.0486401	19	542.013
59	51	20	47	2.57534	1.60479	4	6	0.604349	5	174.658
60	49	17	47	2.18904	1.47954	2	6	0.449165	8.5	281.431
61	70	15	96	3.94521	1.98625	1	6	0.0690646	15	679.398
62	21	15	77	3.16438	1.77887	8	7	0.84894	1.875	143.548
63	53	23	98	6.17534	2.48502	1	10	0.046459	23	743.873
64	42	22	31	1.86849	1.36693	4	5	0.538322	5.5	68.5233
65	78	16	18	0.789041	0.88828	3	2	1	5.33333	1381.45
66	60	23	44	2.7726	1.66511	2	8	0.999908	11.5	583.644
67	32	21	55	3.16438	1.77887	2	8	0.944538	10.5	314.74
68	32	21	61	3.50959	1.87339	1	6	0.991734	21	175.693
69	23	24	97	6.37808	2.52549	2	12	0.99968	12	267.304
70	86	15	54	2.21918	1.48969	2	4	1	7.5	1400.15
71	27	19	46	2.39452	1.54742	3	8	0.998541	6.33333	205.348
72	15	19	99	5.15342	2.27012	4	10	0.0824007	4.75	918.009
73	33	20	33	1.80822	1.3447	3	5	0.273907	6.66667	263.376
74	67	15	69	2.83562	1.68393	1	5	0.999999	15	949.014
75	64	21	96	5.52329	2.35017	2	12	0.999972	10.5	638.51
76	64	15	70	2.87671	1.69609	8	8	0.992473	1.875	711.89
77	1	20	36	1.9726	1.40449	4	8	0.997931	5	4.5274
78	34	22	57	3.43562	1.85354	2	8	0.916745	11	155.189
79	82	19	10	0.520548	0.72149	1	2	1	19	1105.32
80	29	21	89	5.12055	2.26286	4	9	0.0839433	5.25	344.535
81	54	17	46	2.14247	1.46372	1	3	0.999548	17	640.307
82	27	21	37	2.12877	1.45903	2	6	1	10.5	496.023
83	12	18	95	4.68493	2.16447	8	12	0.999637	2.25	99.7808
84	35	16	51	2.23562	1.4952	2	5	1	8	341.753
85	74	18	35	1.72603	1.31378	1	3	1	18	1130.27
86	89	17	17	0.791781	0.889821	2	3	1	8.5	1576.03
87	78	17	62	2.88767	1.69931	1	6	0.893075	17	515.762
88	8	21	23	1.32329	1.15034	4	8	0.125001	5.25	71.8795
89	31	20	16	0.876712	0.936329	2	2	0.999575	10	205.322
90	22	19	62	3.2274	1.7965	3	6	0.999992	6.33333	192.997
91	29	17	35	1.63014	1.27677	4	4	0.310815	4.25	154.873
92	89	23	42	2.64658	1.62683	1	5	0.0518866	23	1939.35
93	50	23	75	4.72603	2.17394	2	9	0.992367	11.5	363.699
94	68	17	87	4.05205	2.01297	3	8	0.154003	5.66667	759.454
95	76	21	41	2.3589	1.53587	2	5	1	10.5	1226.72
96	16	22	96	5.7863	2.40547	4	12	0.999936	5.5	227.419
97	82	24	95	6.24658	2.49931	2	8	0.971407	12	922.781
98	10	24	28	1.8411	1.35687	2	5	1	12	106.589
99	68	18	79	3.89589	1.9738	2	10	0.981205	9	755.079
100	66	16	65	2.84932	1.68799	2	6	0.898688	8	537.945

Tablas de los resultados obtenidos de Simul8, considerando pedidos pendientes en el momento en que llega una nueva orden:

Resultados de Simul8 para 10 productos					
i	Q_i	r_i	S_i	F_i	II_i
1	4	8	100	5.8	405.876
2	4	8	100	4.2	335.874
3	7	8	100	2	87.516
4	4	7	98.44444	4.6	387.42
5	5	4	82.32709	3.5	188.44
6	4	8	97.49533	5.5	417.648
7	5	7	98.4	3.8	333.718
8	4	7	89.8551	5.3	256.036
9	5	7	98.32751	3.9	321.88
10	7	7	99.18831	3.1	606.756

Resultados de Simul8 para 20 productos					
i	Q_i	r_i	S_i	F_i	II_i
1	6	5	99.28571	3.5	305.769
2	6	5	99.23077	2.8	262.878
3	6	5	100	2.6	57.096
4	6	5	96.8733	2.9	339.9
5	6	5	87.04049	3	245
6	6	5	90.60213	3.6	348.348
7	6	5	93.64785	3.1	260.634
8	6	5	82.60762	3.5	235.466
9	6	5	95.10379	3.1	264.16
10	6	5	93.06307	3.8	424.8
11	6	5	87.12763	2.6	481.816
12	6	5	92.885	3.4	235.75
13	6	5	92.89603	2.6	411.996
14	6	5	98.125	2.8	251.312
15	6	5	94.92368	3.4	448.392
16	6	5	97.19565	2.7	355.52
17	6	5	95.38553	3.1	529.197
18	6	5	100	2.9	174.384
19	6	5	100	3.3	39.03
20	6	5	100	2.3	439.025

Resultados de Simul8 para 30 productos					
i	Q_i	r_i	S_i	F_i	II_i
1	7	7	100	3.1	456.043
2	6	7	100	2.8	345.345
3	6	8	100	2.6	84.708
4	7	5	95.375	2.5	395.67
5	7	6	89.40909	2.4	305.76
6	7	7	94.62354	2.9	484.176
7	7	7	97.09565	2.7	380.908
8	7	8	94.82107	2.8	386.474
9	7	7	98.82353	2.6	382.72
10	8	7	98.92857	2.7	687.822
11	7	5	87.46339	2.2	533.364
12	6	7	97.4127	3.4	311.65
13	7	6	77.97657	2.3	381.972
14	7	7	100	2.2	353.238
15	7	6	98.84868	2.9	557.634
16	6	8	100	2.7	268.576
17	7	6	100	2.7	1158.535
18	8	7	100	2	280.368
19	6	7	100	3.3	52.77
20	7	7	100	2	602.225
21	6	8	100	2.8	778.752
22	7	6	95.68875	3.3	521.5
23	6	8	98.85621	2.3	239.921
24	7	7	99.23077	1.9	555.408
25	6	5	92.0237	2.5	268.056
26	6	6	98.44737	2.3	270.76
27	7	7	96.43756	2.5	491.696
28	7	8	95.22609	2.8	679.349
29	7	6	99.23077	2.8	47.46
30	7	7	91.28406	2.9	385.12

Resultados de Simul8 para 40 productos					
i	Q_i	r_i	S_i	F_i	II_i
1	5	8	99.35484	4.4	431.981
2	8	8	100	1.8	428.274
3	6	9	100	2.6	93.184
4	7	5	94.82353	2.5	396.99
5	8	8	97.98077	2.1	454.16
6	8	8	97.16667	2.5	588.896
7	8	9	99.56522	2.3	521.994
8	7	7	91.09867	2.8	344.85
9	8	10	100	2.3	596.44
10	5	7	97.82601	4.6	478.254
11	5	8	98.91813	3.3	677.488
12	6	9	99.69697	3.4	402.04
13	7	7	82.59244	2.3	464.121
14	8	8	100	1.9	422.168
15	8	8	100	2.6	776.118
16	9	8	100	1.6	339.856
17	8	7	100	2.2	1373.482
18	11	6	100	1.4	291.312
19	8	5	98.16667	2.2	46.26
20	9	11	100	1.4	957.95
21	5	8	100	3.4	723.06
22	11	12	100	1.8	1188.6
23	8	4	80.28442	1.7	173.162
24	8	9	100	1.5	761.976
25	9	11	100	1.5	652.698
26	8	7	99.41176	1.4	380.8
27	7	11	100	2.5	776.833
28	8	3	74.94213	2.4	418.407
29	6	9	100	3.3	65.97
30	9	8	93.37502	2	504.6
31	7	6	100	2.8	173.9
32	9	5	86.71272	1.7	248.157
33	10	9	100	1.8	299.224
34	7	8	100	2.7	626.496
35	7	8	100	2.5	828
36	5	4	95.8531	3.2	254.406
37	8	8	100	2.3	593.34
38	9	9	99.16667	1.4	823.498
39	6	7	99.28571	2.7	503.412
40	5	5	90.36045	3.6	232.869

Resultados de Simul8 para 50 productos					
i	Q_i	r_i	S_i	F_i	II_i
1	6	7	100	3.5	410.189
2	8	4	98.46154	1.8	263.109
3	7	4	100	2	51.116
4	8	5	94.07143	2	444.51
5	5	4	82.32709	3.5	188.44
6	4	5	89.71501	5.5	286.44
7	7	7	97.02899	2.7	382.602
8	7	5	83.97687	2.8	274.186
9	7	8	99.375	2.6	436.28
10	6	7	98.57698	3.8	545.868
11	6	6	91.54781	2.6	545.988
12	8	6	97.7573	2.4	348.91
13	6	5	66.21866	2.6	240.609
14	5	6	100	3.5	250.86
15	7	4	92.60281	2.9	439.11
16	7	7	100	2.2	255.728
17	5	6	98.32237	3.7	976.689
18	7	4	100	2.4	168.48
19	7	7	100	2.8	55.32
20	7	7	100	2	599.25
21	7	7	100	2.3	778.752
22	6	8	98.28571	3.8	581
23	7	6	92.15368	2.3	201.212
24	7	8	100	1.9	625.632
25	8	7	98.42105	1.6	400.86
26	7	6	98.26797	1.9	302.4
27	7	8	98.61111	2.5	551.213
28	4	5	85.37321	5.3	335.789
29	7	6	99.23077	2.8	47.25
30	7	6	87.30278	2.9	344.23
31	8	5	100	2.6	164.4
32	8	7	95.94853	2.1	292.383
33	7	6	100	2.5	181.252
34	5	6	99.375	4.1	398.112
35	6	6	100	3.2	596.25
36	7	5	98.46154	2.2	374.088
37	7	7	99.28571	3	474.73
38	5	8	98.23529	2.7	501.41
39	7	5	94.57463	2.2	405.132
40	8	7	100	1.9	416.598
41	6	8	98.31798	3.5	265.434
42	7	5	88.43113	2.5	311.052
43	6	7	99.23077	2.4	248.848
44	6	8	96.88259	2.2	244.934
45	6	7	91.19353	3	223.6
46	7	5	93.63553	2.4	228.735
47	6	5	98.875	3	1000.296
48	8	7	96.77591	2.7	615.88
49	7	7	100	2.8	155.472
50	6	6	100	2.4	185.932

Resultados de Simul8 para 60 productos

i	Q_i	r_i	S_i	F_i	II_i
1	6	6	100	3.5	359.568
2	5	6	100	3.3	265.881
3	7	5	100	2	60.216
4	6	6	99.47368	2.9	393.03
5	6	7	94.71684	3	325.64
6	6	5	90.53035	3.6	350.196
7	8	4	91.25325	2.3	281.204
8	5	6	86.49375	4.3	245.146
9	7	8	99.375	2.6	438.1
10	6	6	96.91529	3.8	480.378
11	4	5	89.48465	4.2	409.754
12	7	8	99.375	2.8	405.49
13	6	5	67.3645	2.6	251.868
14	6	5	97.5	2.8	250.634
15	5	8	100	4.2	638.673
16	6	8	100	2.7	268.576
17	8	7	100	2.2	1373.482
18	7	7	100	2.4	256.896
19	5	7	100	4.1	48.18
20	8	6	100	1.6	565.675
21	4	7	100	4.4	572.364
22	7	8	99.18919	3.3	668.85
23	5	6	87.34091	3.1	158.015
24	7	7	99.23077	1.9	555.408
25	7	8	98.5	1.9	427.176
26	6	6	97.99499	2.3	272.16
27	4	6	91.29215	4.5	321.703
28	7	7	93.51993	2.8	610.228
29	5	6	100	4.1	39.66
30	6	7	90.46434	3.1	355.54
31	8	7	100	2.6	210.8
32	6	7	96.49247	2.9	235.494
33	5	7	100	4	172.099
34	7	7	100	2.7	553.8
35	8	6	99.16667	2.3	719.55
36	4	5	98.1201	4.1	263.562
37	5	7	98.5733	4.2	390.92
38	7	8	97.64706	1.6	629.356
39	4	6	99.6	4.3	340.704
40	6	6	97.61905	2.7	313.404
41	7	7	95.94575	2.9	263.608
42	6	6	92.74053	3	309.12
43	8	6	100	1.4	260.178
44	5	5	86.67208	3	136.166
45	6	6	88.21347	3	195.4
46	6	6	95.4902	2.8	226.005
47	6	6	100	3	1150.172
48	6	7	90.92043	3.9	510.86
49	5	8	100	4.2	151.759
50	7	8	100	2	260.236
51	5	5	93.61039	3.2	262.679
52	7	5	87.69992	2.9	246.84
53	6	8	100	3	173.419
54	5	4	85.45982	3.2	203.742
55	6	5	87.39625	3	388.583
56	7	5	100	2.2	45.08
57	5	7	100	4.4	59.92
58	6	7	89.5221	3.2	53.214
59	8	6	97.91667	2	351.135
60	6	4	95.12501	2.3	223.776

Resultados de Simul8 para 70 productos

i	Q_i	r_i	S_i	F_i	II_i
1	8	7	100	2.6	493.725
2	6	8	100	2.8	389.004
3	8	8	100	1.8	95.524
4	6	7	100	2.9	455.07
5	7	4	78.81733	2.4	233.52
6	4	5	89.80159	5.5	285.208
7	5	6	97.61787	3.8	269.588
8	7	8	94.82107	2.8	388.168
9	5	7	98.32751	3.9	324.48
10	8	5	94.9934	2.7	542.682
11	6	6	92.024	2.6	550.722
12	7	4	89.86428	2.8	239.89
13	5	8	74.6343	3.1	331.515
14	7	5	96.42857	2.2	270.07
15	6	8	100	3.4	657.594
16	7	5	97.39181	2.2	194.832
17	6	7	100	3.1	1243.592
18	5	4	99.375	3.5	134.784
19	6	4	97.16667	3.3	33.51
20	5	6	100	3	456.025
21	8	7	100	2	810.108
22	8	7	97.02248	2.7	634.2
23	5	6	87.34091	3.1	157.08
24	3	6	95.63636	5	296.4
25	5	6	95.21918	2.9	268.974
26	7	5	97.2807	1.9	263.76
27	5	6	90.46814	3.6	376.941
28	5	7	91.9121	4.2	481.393
29	5	4	94.81635	4.1	29.04
30	8	8	95.13559	2.2	476.47
31	5	5	100	4	122.5
32	6	4	85.93285	2.9	161.784
33	6	8	100	3.3	216.734
34	5	7	100	4.1	464.88
35	6	4	94.00295	3.2	441
36	8	6	100	1.8	454.53
37	5	5	93.32998	4.2	279.56
38	6	4	85.11183	2.3	312.702
39	4	5	95.52211	4.3	287.56
40	7	6	96.70603	2.5	331.695
41	6	4	88.10463	3.5	171.146
42	7	7	97.7591	2.5	395.508
43	8	6	98.75	1.4	257.912
44	6	5	87.59442	2.2	161.71
45	4	4	80.84332	4.9	113.8
46	6	5	91.7567	2.8	198.51
47	5	6	100	3.9	983.456
48	7	6	89.83928	3.3	490.924
49	8	6	100	2.4	150.89
50	7	5	96.25	2	175.784
51	6	8	99.5	2.4	464.286
52	8	6	92.27076	2.2	307.12
53	4	5	93.99447	4.8	95.811
54	6	5	90.01629	2.8	267.54
55	4	6	93.29443	4.5	347.325
56	7	7	100	2.2	57.925
57	7	7	100	3.1	68.915
58	6	6	86.44297	3.2	47.796
59	8	4	91.97358	2	271.32
60	7	6	99	1.9	318.816
61	7	5	93.9604	2	416.564
62	5	5	93.82051	2.7	90.72
63	6	7	90.65179	3.5	281.75
64	8	4	94.76426	2.8	277.141
65	5	4	97.83654	2.9	367.464
66	7	8	100	3.1	527.22
67	7	6	97.5641	3	232.56
68	7	7	95.93877	3	240.464
69	6	5	81.92501	3.8	104.64
70	8	7	99.375	1.5	758.652

Resultados de Simul8 para 80 productos

i	Q_i	r_i	S_i	F_i	II_i
1	5	7	99.0625	4.4	379.544
2	5	8	100	3.3	350.427
3	7	6	100	2	69.68
4	4	6	97.5188	4.6	328.35
5	7	4	78.81733	2.4	233.52
6	5	6	93.5208	4.2	346.808
7	4	6	97.04301	5	259.424
8	3	7	92.9245	7.4	245.146
9	6	7	98.70445	3.1	356.98
10	5	7	97.82601	4.6	480.024
11	7	3	79.21487	2.2	418.696
12	7	6	97.00167	2.8	313.95
13	4	8	73.65521	4.2	295.236
14	6	6	98.66667	2.8	290.862
15	3	7	100	7.4	429.471
16	6	6	97.9798	2.7	208.032
17	6	7	100	3.1	1238.564
18	5	4	99.375	3.5	134.064
19	6	5	100	3.3	38.85
20	5	4	100	3	315.35
21	4	5	98.80702	4.4	397.8
22	7	5	92.54482	3.3	468.65
23	6	4	78.95052	2.3	138.941
24	7	4	87.58601	1.9	364.344
25	4	7	97.1167	3.8	273.564
26	6	4	92.62471	2.3	202.44
27	5	5	88.62621	3.6	329.094
28	5	4	81.29826	4.2	334.971
29	7	5	96.80672	2.8	40.92
30	5	7	90.98002	4.1	313.49
31	2	4	97.13423	11.2	67.8
32	5	7	97.03674	3.8	216.783
33	7	6	100	2.5	181.252
34	3	5	97.94673	7	265.512
35	8	3	90.76364	2.3	475.2
36	3	6	99.47368	5.6	290.049
37	3	7	98.7403	7.4	309.72
38	5	5	88.16769	2.7	317.148
39	7	5	94.57463	2.2	405.132
40	7	4	86.02673	2.5	257.439
41	6	8	98.31798	3.5	265.434
42	4	5	90.21811	4.7	205.344
43	6	7	99.23077	2.4	246.376
44	8	8	98.75	1.6	296.64
45	6	5	84.34632	3	172.6
46	6	6	95.4902	2.8	226.005
47	5	3	89.49842	3.9	622.238
48	1	7	90.51428	25.9	163.048
49	9	6	100	2.2	151.838
50	5	5	96.375	3	133.988
51	4	8	97.61905	3.8	377.883
52	5	6	89.3356	4	223.52
53	7	4	90.55888	2.5	115.758
54	5	4	84.64163	3.2	204.624
55	5	7	97.04844	3.6	450.47
56	5	5	99.33333	3.5	37.205
57	5	7	100	4.4	60.445
58	6	9	95.92217	3.2	65.94
59	6	6	99.00463	2.8	296.31
60	5	5	97.79487	2.8	232.128
61	7	7	97.33333	2	549.659
62	8	7	100	1.3	179.2
63	5	8	92.76005	4.4	298.31
64	5	8	100	4.7	375.124
65	6	4	98.88889	2.4	437.736
66	7	7	98.92045	3.1	462.753
67	8	7	98.77778	2.7	274.208
68	4	8	97.54817	5.7	195.928
69	6	4	78.63163	3.8	94.464
70	5	7	100	2.8	584.46
71	5	7	99.375	3.3	172.53
72	7	8	94.8364	2.8	120.87
73	7	7	98.53333	4.3	363.165
74	7	7	96.59091	1.9	517.98
75	5	5	85.32455	4.2	275.716
76	4	5	93.90847	3.7	281.393
77	3	7	100	6.6	163.158
78	5	8	99.61538	4	229.09
79	6	3	98.33333	2.5	368.936
80	7	9	99.11067	2.5	245.226

Resultados de Simul8 para 90 productos

i	Q_i	r_i	S_i	F_i	II_i
1	11	11	100	1.7	796.089
2	11	12	100	1	698.313
3	11	10	100	1.1	131.768
4	12	11	100	1.1	973.83
5	10	11	100	1.5	698.6
6	10	11	100	2	886.116
7	12	11	100	1.3	750.926
8	11	11	100	1.6	684.376
9	11	11	100	1.4	752.96
10	9	12	100	2.3	1150.5
11	11	11	100	1.1	1299.746
12	11	11	100	1.6	664.7
13	12	12	99.33333	1.1	1124.649
14	12	11	100	1	690.656
15	10	11	100	1.8	1148.469
16	12	11	100	1.1	508.816
17	10	10	100	1.6	2152.822
18	10	10	100	1.5	388.368
19	12	12	100	1.3	113.76
20	12	12	100	0.9	1221.875
21	11	12	100	1.2	1438.164
22	12	11	100	1.6	1153.25
23	9	10	100	1.3	367.081
24	11	11	100	1	1063.848
25	10	11	100	1.2	709.308
26	11	10	100	1.1	606.48
27	8	12	100	1.9	909.482
28	11	11	98.94737	1.6	1149.29
29	10	8	100	2	75.3
30	10	10	100	1.9	697.45
31	9	11	100	2	317.3
32	11	11	100	1.4	508.788
33	10	5	100	1.8	199.784
34	11	11	100	1.6	1037.4
35	12	11	100	1.4	1441.8
36	11	12	100	1.1	930.315
37	10	11	100	2	852.89
38	10	10	100	1.1	933.66
39	11	11	100	1.2	998.816
40	10	9	100	1.6	626.808
41	11	11	100	1.8	502.316
42	12	11	100	1.2	797.916
43	11	11	100	1	477.508
44	10	11	100	1.3	432.6
45	11	11	100	1.4	522.4
46	11	10	100	1.2	507.585
47	12	11	100	1.2	2741.552
48	11	11	99.62963	1.9	1105.38
49	11	10	100	1.7	253.906
50	11	11	100	1.2	429.656
51	11	10	100	1.1	765.829
52	10	11	100	1.7	598.18
53	11	11	100	1.3	324.711
54	11	11	100	1.4	763.518
55	11	10	100	1.3	988.087
56	12	11	100	1.3	104.72
57	10	11	100	2	119.56
58	10	11	100	1.8	108.36
59	12	10	100	1.1	687.735
60	12	11	100	0.9	734.688
61	11	9	100	1	907.848
62	11	10	100	0.9	285.6
63	11	12	100	1.7	694.6
64	9	8	100	2.3	464.321
65	11	10	100	1.1	1174.128
66	11	10	100	1.7	803.88
67	12	10	100	1.6	492.632
68	12	12	100	1.5	513
69	11	11	99.56522	1.7	279.36
70	11	11	100	0.9	1344.258
71	12	12	100	1	431.68
72	12	10	100	1.4	219.62
73	10	11	100	2.5	680.955
74	10	11	100	0.9	943.845
75	9	11	99	2.1	756.413
76	12	12	100	0.9	1110.2
77	10	10	100	1.7	391.778
78	10	11	100	1.8	417.26
79	8	12	100	1.9	1145.167
80	11	11	100	1.4	385.986
81	12	11	100	1	841.428
82	12	9	100	1.3	1164.774
83	8	11	100	1.9	142.308
84	11	11	100	1.2	525.381
85	5	9	100	3.3	644.037
86	10	11	100	1.5	1459.864
87	10	12	100	1.4	1222.317
88	12	12	100	1.3	132.012
89	11	10	100	1.3	449.035
90	9	11	100	2	294.872

Resultados de Simul8 para 100 productos

i	Q_i	r_i	S_i	F_i	II_i
1	12	12	100	1.6	888.024
2	13	11	100	0.9	715.407
3	12	11	100	1.1	148.2
4	11	11	100	1.3	909.48
5	14	12	100	1	878.64
6	14	14	100	1.2	1291.444
7	10	13	100	1.5	781.418
8	13	11	100	1.3	776.094
9	12	7	100	1.4	611.52
10	12	11	100	1.7	1241.478
11	13	12	100	1	1549.07
12	15	10	100	1	753.48
13	12	14	100	1.1	1253.085
14	13	12	100	1	753.71
15	12	13	100	1.6	1391.229
16	11	11	100	1.3	485.936
17	13	5	100	1.1	1699.045
18	12	12	100	1.2	477.936
19	10	11	100	2	97.11
20	11	9	100	1.1	954.55
21	11	12	100	1.2	1438.164
22	12	12	100	1.6	1242.85
23	14	12	100	1	559.878
24	11	7	100	1	772.008
25	14	10	100	0.8	815.49
26	10	12	100	1.3	655.76
27	12	10	100	1.1	1044.854
28	15	14	100	1.1	1703.485
29	14	12	100	1	118.86
30	10	11	100	1.9	757.19
31	13	13	100	1.3	410.3
32	12	12	100	1.3	573.048
33	10	11	100	1.8	348.944
34	12	12	100	1.4	1142.232
35	12	13	100	1.4	1635.3
36	15	10	100	0.9	958.764
37	13	10	100	1.2	948.01
38	13	8	100	0.7	963.794
39	14	13	100	1	1262.352
40	12	14	100	1	982.527
41	12	11	100	1.7	528.378
42	10	14	100	1.6	861.672
43	11	13	100	1	542.192
44	10	13	100	1.3	499.756
45	10	11	100	1.8	515
46	13	13	100	1.1	662.61
47	11	11	100	1.5	2599.254
48	11	12	100	1.9	1192.244
49	11	11	100	1.7	272.234
50	14	11	100	0.8	482.804
51	13	12	100	0.9	1009.076
52	13	14	100	1.2	841.5
53	11	11	100	1.3	323.512
54	12	12	100	1.1	844.956
55	10	12	100	1.5	1122.386
56	13	10	100	1.2	102.97
57	11	12	100	1.7	133.63
58	12	12	100	1.4	131.754
59	13	11	100	1.1	767.295
60	11	13	100	1	796.608
61	11	13	100	1	1214.2
62	11	14	100	0.9	369.6
63	12	14	100	1.5	834.67
64	13	12	100	1.5	810.604
65	10	13	100	1.2	1367.864
66	13	10	100	1.6	894.447
67	12	13	100	1.6	601.16
68	10	13	100	2.1	518.776
69	11	12	100	1.7	301.152
70	9	12	100	1.3	1272.633
71	11	14	100	1.3	471.156
72	13	12	100	1.2	271.839
73	12	14	100	2.2	884.235
74	9	12	100	1.1	959.42
75	11	12	99.64286	1.7	879.823
76	11	14	100	1	1186.633
77	12	12	100	1.3	495.154
78	11	12	100	1.6	479.26
79	11	12	100	1.2	1298.603
80	12	14	100	1.4	491.832
81	13	10	100	1	836.658
82	14	11	100	1.2	1525.632
83	10	12	100	1.4	170.113
84	13	10	100	0.7	545.091
85	10	12	100	1.3	1093.26
86	13	9	100	1	1481.872
87	11	10	100	1.1	1131.435
88	11	10	100	1.6	110.124
89	11	12	100	1.3	511.965
90	13	10	100	1	354.032
91	1	11	100	17.1	321.651
92	12	12	100	1.4	1384.299
93	12	12	100	1.6	768.831
94	13	14	100	1	1314
95	12	13	100	1.3	1280.394
96	12	12	100	1.3	228.344
97	13	12	100	1.4	1208.628
98	11	10	100	1.6	135.366
99	13	13	100	0.9	1120.77
100	13	13	100	0.9	1356.292

Bibliografía

- [1] Ackley, D.H., 1987, A Connectionist Machine for Genetic Hillclimbing, Kluwer Academic, Publishers.
- [2] Alander, J., 1992, On optimal population size of genetic algorithms, Proceedings CompEuro, Computer Systems and Software Engineering, 6th Annual European Computer Conference, págs 65-70.
- [3] Alrefaei, M.H. and S. Andradóttir. 1998, A Simulated Annealing Algorithm with Constant Temperature for Discrete Stochastic Optimization, technical report, Department of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA.
- [4] Bain, L., Engelhardt, M, 1992, Introduction to probability and Mathematical Statistics, 2nd Edition, Duxbury Press, Belmont, California.
- [5] Banks, J., Carson II, and B.L. Nelson, 1995, Dis-crete-Event Systems Simulation, 2nd Edition, Prentice-Hall.
- [6] Benveniste, A., Metivier, M., Priouret, P., 1990, Adaptive Algorithms and Stochastic Approximations, Springer-Verlag, Berlin.
- [7] Biles, W., 1974, A gradient regression search procedure for simulation experimentation, In: Proceedings of the 1974 Winter Simulation Conference, págs 491-497
- [8] Bramlette, M.F., 1991, Initialization, mutation and selection methods in genetic algorithms for function optimization, In: proceedings of the Fourth International Conference on Genetic Algorithms, págs. 100-107.
- [9] Carson, Y. and Maria, A., 1997, Simulation Optimization: Methods and Applications, In: Proceedings of the 1997 Winter Simulation Conference, ed. Sigrun Andradottir, Kevin J. Healy, David H. Withers, Barry L. Nelson: 18-126, Piscataway, New Jersey: Institute of Electrical and Electronic Engineers.
- [10] Chang, Pao-Long , Chou Ying-Chyi, Huang, Ming-Guang, 2007, A $(r; r; Q)$ inventory model for spare parts involving equipment criticality, Operations Research Letters 35, págs. 567-572.
- [11] Daugherty, A., Turnquist, M., 1980, Simulation optimization using response surfaces based on spline approximation, In: Proceedings of the 1980 Winter Simulation Conference, págs. 183-193.
- [12] Davis, L., 1991, Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York.

- [13] De Jong, K., 1975, An analysis of the behaviour of a class of genetic adaptive systems, Tesis doctoral, University of Michigan.
- [14] Deshpande, V., Morris A. Cohen, Karen Donohue, 2003, A Threshold Inventory Rationing Policy for Service-Differentiated Demand Classes, *Management Science*, Vol. 49, No. 6, págs. 683-703.
- [15] Diego-Mas, J., 2006, Optimización de la distribución en planta de instalaciones industriales mediante algoritmos genéticos. Aportación al control de la geometría de las actividades, Tesis Doctoral, Universidad Pontificia de valencia, pág. 444.
- [16] Fogarty, T.C., 1989, Varying the probability of mutation in the genetic algorithm, In: *Proceedings of the Third International Conference on Genetic Algorithms*, 104-109.
- [17] Fu, M., 1994, Optimization via simulation: A review, *Annals of Operations Research* 53, págs. 199-247.
- [18] Fu, M., 2002, Optimization for Simulation: Theory and Practice, *Inform Journal on Computing*, 14 (3), págs. 192-215.
- [19] Fu, M., 2001, Simulation optimization, In: *Encyclopedia of Operations Research and Management Science*, 2nd edition, ed. S. Gass and C. Harris, Boston: Kluwer Academic Publishers, págs. 756-759.
- [20] Fu, M., 2001, Optimization for simulation: Theory vs. Practice, *Inform Journal on Computing* forthcoming.
- [21] Fu, M., Glover, F., April, J., 2005, Simulation Optimization: A review, new development, and application, In: *Proceedings of the 2005 Winter Simulation Conference*, págs. 83-95.
- [22] Gandibleux, X., Ehrgott, M., 2000, A survey and annotated bibliography of multiobjective combinatorial optimization, *OR Spektrum*, 22, págs. 25-460.
- [23] Ghaziri, Hassan; Diaz, Adenso y Glover, Fred, 1996, Optimización heurística y Redes Neuronales, Madrid: Paraninfo, pág 235.
- [24] Goic, M., Caballero, C., 2005, Aplicación de algoritmos genéticos para el mejoramiento del proceso de programación del rodaje en la industria del cine independiente (Revisión: 05 Agosto, 2005).
- [25] Glover, F. y Laguna, M, 1993, Tabu Search. Modern Heuristic Techniques for Combinatorial Problems, C. Reeves, ed., Blackwell Scientific Publishing, págs. 71-140.
- [26] Laguna, M., Kelly, J. P., Gonzalez-Velarde, J. L., y Glover, F., 1995, Tabu Search for the Multilevel Generalized Assignment Problem, *European Journal of Operational Research* 82, págs. 176-189.
- [27] Glover, F., 1996, Ejection Chains, Reference Structures and Alternating Path Methods for Traveling Salesman Problems, *Discrete Applied Mathematics*, 65, págs. 223-253.
- [28] Laguna, M., Kelly, J. P., Gonzalez-Velarde, J. L., y Glover, F., 1998, OptQuest for Crystal Ball User's Guide, Decisioneering Inc. Denver, CO and Optimization Technologies, Inc. Boulder, CO.

- [29] Glover, F., Laguna, M., Martí, R., 2000, Fundamentals of Scatter Search and Path Relinking Control and Cybernetics, 39, págs. 653-684.
- [30] Goldberg, D.E., Lingle. Jr. R. Alleles, 1985, loci and the traveling salesman problem, en Proceedings of the First International Conference on Genetic Algorithms and Their Applications, págs. 154-159.
- [31] Goldberg, D.E., 1989, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, MA.
- [32] Gurkan, G., Ozge, Y., Robinson. S., 1994, Sample-path optimization in simulation, In: Proceedings of the 1994 Winter Simulation Conference, págs. 247-254.
- [33] Hadley, G., Whitin, T.M., 1963, Analysis of Inventory Systems, Prentice-Hall, Inc., Englewood Cliffs, NJ.
- [34] Holland, J., 1975, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor.
- [35] Hopp, W.J., Zhang, R.Q., Spearman, M.L., 1999, Easily implementable inventory control policies, Operations Research; May/Jun 1997; 45, 3; ABI/INFORM Global, págs. 327-340.
- [36] Hopp, W.J., Spearman, M.L., 1997, Factory Physics, first ed. McGraw-Hill, New York.
- [37] Hopp, W.J., Spearman, M.L., 2001, Factory Physics, second ed. McGraw-Hill, New York.
- [38] Kopytov, E., Greenglaz, L., Muravyov, A., Puzinkevich, E., 2007, Modeling of two strategies in inventory control system with random lead time and demand, Computer Modelling and New Technologies, Vol.11, No.1, 21-30
- [39] Kushner, H., Clark, D., 1978, Stochastic Approximation Methods for Constrained and Unconstrained Systems, Springer-Verlag, New York.
- [40] Kushner, H., Yin, G., 1997, Stochastic Approximation Algorithms and Applications, Springer-Verlag, New York.
- [41] Laguna, M., 1997, Optimization of Complex Systems with OptQuest, Latest revision: April 8, 1997, págs. 1-14
- [42] Laguna, M., Martí, R., 2000, Optquest Callable Library User's Manual, Optimization Technologies, Inc., Boulder, CO, (www.opttek.com).
- [43] Larrañaga, P., Kuijpers, C., Murga, R., Inza, I., Dizdarevich, S., 1999, Evolutionary algorithms for the travelling salesman problem: A review of representations and operators, Artificial Intelligence Review, 13, págs. 129-170.
- [44] Larsen, C., Kiesmuller, G.P., 2007, Developing a closed-form cost expression for an (R, s, nQ) policy where the demand process is compound generalized Erlang, Operations Research Letters 35, págs. 567-572.
- [45] Ljung, L., Pflug, G., Walk, H., 1992, Stochastic Approximation and Optimization of Random Systems, Birkhauser-Verlag, Basel.
- [46] Michalewicz, Z., Janikow, C.Z., 1991, Handling constraints in genetic algorithms, In: Proceedings of the Fourth International Conference on Genetic Algorithms, págs. 151-157.

- [47] Mitra, S., Chatterjee, A.K., 2004, Echelon stock based continuous review ($R; Q$) policy or fast moving items, *Omega* 32, India, págs. 161-166.
- [48] Mohammad H. and Rossetti M., 2007, An efficient heuristic optimization algorithm for a two-echelon (R, Q) inventory system, *International journal of Production Economics* 109 págs. 195-213.
- [49] Ólafsson, S. and J. Kim., 2002, Simulation Optimization, In: *Proceedings of the 2002 Winter Simulation Conference*, págs. 79-84.
- [50] Oliver, I.M., Smith, D.J., Holland, J.R.C., 1987, A study of permutation crossover operators on the TSP, en *Genetic Algorithms and Their Applications*, In: *Proceedings of the Second International Conference*, págs. 224-230.
- [51] Osman, I. H. and Kelly J. P., 1996, *Meta-Heuristics: theory and applications*. Boston: Kluwer Academic, págs. 1-21.
- [52] Jones, t., 1995, *Crossover, macromutation, and population-based search*. Edit: Morgan Kaufmann. San Mateo.
- [53] Pasupathy. R., Henderson, S., 2006, A tested of simulation-optimization problems, In: *Proceedings of the 2006 Winter Simulation Conference*, págs. 255-263.
- [54] Plambeck, E., Fu, B., Robinson, S., Suri, R., 1996. Sample-path optimization of convex stochastic performance functions. *Mathematical Programming B*, págs. 137-176.
- [55] Platt, D.E., Robinson, L.W., Freund, R.B., 1997. Tractable (Q, R) heuristic models for constrained service levels. *Management Science* 43, 951-965.
- [56] Robinson, S., 1996, Analysis of sample-path optimization, *Mathematics of Operations Research* 21, págs. 513-528.
- [57] Rosen, S., Harmonosky C., Traband, M.A, 2007, Simulation optimization method that considers uncertainty and multiple performance measures. *European Journal of Operational Research*, 181, págs. 315-330.
- [58] Rubinstein, R., Shapiro, A., 1993, *Discrete Event Systems: Sensitivity Analysis and Stochastic Approximation using the Score Function Method*, John Wiley and Sons, Chichester.
- [59] Ruppert, D., 1991. *Stochastic approximation*. In: *The Handbook of Sequential Analysis*. Marcel Dekker, New York, págs. 503-529.
- [60] Sait, Sadiq and Youssef, Habib, 1999, *Iterative Computer Algorithms with Applications in Engineering : Solving Combinatorial Optimization Problems*. California: Ieee Computer Society 1999, pág 387.
- [61] Shannon, R.E. 1975, Introduction to The Art and Science of simulation, In: *Proceedings of the 1998 Winter Simulation Conference*, págs. 7-14.
- [62] Syswerda, G., 1991, Schedule optimization using genetic algorithms. L. Davis (ed.), *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, págs. 332-349.
- [63] Smith, D., 1976, Automatic optimum-seeking program for digital simulation, *Simulation* 27, págs. 27-32.

- [64] Viteri, K., Salazar, C., Paredes, C., Muñoz, J. Algoritmos Genéticos. <http://www.fiec.espol.edu.ec/investigacion/topico/tabusearch.pdf>
- [65] Wang, I., Spall, J., 1999, A constrained simultaneous perturbation stochastic approximation algorithm based on penalty functions, In: Proceedings of the 1999 American Control Conference, págs. 393-399.
- [66] Wilson, J., 1987, Future direction in response surface methodology for simulation, In: Proceedings of the 1987 Winter Simulation Conference, págs. 378-381.