

ALGORITMOS GENÉTICOS Y PROGRAMACIÓN EVOLUTIVA EN LA SOLUCIÓN DE PROCESOS Y SU APLICACIÓN EN UN CONTEXTO EMPRESARIAL DE MANEJO DE INVENTARIOS Y RUTAS DE DISTRIBUCIÓN

Jaimes Sequeda, Dora Elizabeth y Díaz Silva, José Fabián
Fundación Universitaria de San Gil –UNISANGIL–
Facultad de Ciencias Naturales e Ingeniería
Ingeniera de Sistemas
San Gil, Colombia

dh0ritha@hotmail.com
jdiaz@unisangil.edu.co

Resumen— En la actualidad, el nivel de exigencia del mercado implica mejoras y optimización en los diferentes procesos empresariales e industriales. La demanda, la competencia, la implementación de nuevas tecnologías, los clientes, son variables que evidencian la necesidad de mejorar los procesos conocidos por medio del uso de herramientas que logren un desarrollo optimizado en menor tiempo y que permita evolucionar para permanecer en el mercado. Es aquí donde la aplicación de los algoritmos genéticos (AGs) representa una solución eficiente a dichos requerimientos, ya que estos usan una analogía directa con el comportamiento natural, trabajando con una población de posibilidades; cada una de las cuales representa una solución factible a un problema dado, con un valor que equivaldría al grado de efectividad de un organismo para competir con otros. Cuanto mayor sea la adaptación de un ente a su entorno, mayor será la posibilidad de que trascienda al siguiente nivel, con una población nueva de soluciones mejoradas que reemplaza sus antecesoras y con mayor proporción de buenas características, lo que permite que a lo largo de su desarrollo y evolución, conserve sólo las mejores características para llegar al ente más eficiente y mejor adaptado, que equivale a la solución óptima del problema.

Con el ánimo de evidenciar la utilidad de imitar el comportamiento de evolución biológica en el desarrollo de sistemas artificiales, surge el diseño y la implementación de una herramienta que simula un proceso empresarial de gestión de inventarios y rutas de distribución, por medio de un AG programado en Lenguaje multiplataforma Java y una base de datos portable en Derby, que, cumpliendo los supuestos de la evolución natural, logrará encontrar una población de soluciones óptimas al problema planteado. El desarrollo de este proyecto incluye las fases de análisis, diseño global, implementación del sistema, pruebas y análisis de resultados; etapas que podrán apreciarse de forma sencilla en el desarrollo de este artículo.

Palabras clave— genética, algoritmos, optimización, inventarios, rutas de distribución

Abstract— At present, the level of market requirements and optimization improvements in various business and industrial processes, demand, competition, the implementation of new technologies, customers are variables that show the need to improve known processes through use of tools that achieve better development in less time than allowed to evolve to stay on the market, this is where the application of genetic algorithms represent an efficient solution to these requirements because they use a direct analogy with the natural behavior working with a population of possibilities, each of which represents a feasible solution to a given problem, assigning each a value equivalent to the degree of effectiveness of an agency to compete with others, the greater the adaptation of an entity to environment the greater the possibility of going beyond the next level, with a new population of improved solutions to replace their predecessors and with a higher proportion of good features, which allows over its development and evolution keeps only the best features to get to being more efficient and better adapted than equal to the optimal solution.

In an attempt to demonstrate the usefulness of imitating the behavior of biological evolution in the development of artificial systems, It arise the design and implementation of a tool that simulates a business process, inventory management and distribution routes, through a programmed AG cross-platform Java Language and portable database in Derby, that meet to comply the theories of the natural evolution, it will find of a population of optimal solutions to

the problem. The development of this project includes the phases of analysis, overall design, system implementation, testing and results analysis; stages can easily be seen in the development of this article.

Key words— genetic, algorithms, optimization, inventories, distribution routes

I. INTRODUCCIÓN

Este proyecto inicia al determinar las variables relevantes implicadas en el proceso de gestión de los inventarios y rutas de distribución que pueden ser optimizadas mediante la aplicación de algoritmos genéticos; y, posteriormente, implementar una simulación de este proceso, aplicando la técnica en mención sobre lenguaje multiplataforma JAVA e inspirados en la metodología Extreme Programming, ya que sus características incrementan la productividad y calidad del desarrollo, con el objetivo de establecer la aplicabilidad de los algoritmos genéticos y la computación evolutiva en la optimización del proceso de administración de inventarios y rutas de distribución.

Existe una tendencia creciente a la implementación de sistemas de producción y gestión comercial más autónomos e inteligentes, debido a las exigencias del mercado por obtener productos de alta calidad, minimizando costos y tiempos de solución; lo cual se hace complicado con operaciones manuales. Por esta razón, para los países subdesarrollados ha sido difícil alcanzar niveles competitivos a nivel mundial. Es aquí donde empieza la búsqueda de nuevas tecnologías y herramientas que resuelvan y optimicen constantemente los diferentes procesos.

El objetivo principal de este trabajo con Algoritmos Genéticos y computación evolutiva es evidenciar la eficiencia de la técnica en la optimización de sistemas por medio de la simulación de un proceso industrial de manejo de inventarios y rutas de distribución, determinando las variables relevantes implicadas en dichos procesos que pueden ser optimizadas mediante los AGs.

II. ALGORITMOS GENÉTICOS Y PROGRAMACIÓN EVOLUTIVA EN LA OPTIMIZACIÓN DE PROCESOS

Los *algoritmos genéticos* usan una analogía directa con el comportamiento natural. Trabajan con una población de individuos que representan una solución factible a un problema dado. A cada individuo se le asigna un valor o puntuación, relacionado con la eficiencia de dicha solución. En la naturaleza, esto equivaldría al grado de efectividad de un organismo para competir por unos determinados recursos. Cuanto mayor sea la adaptación de un individuo al problema, mayor será la probabilidad de que el mismo sea seleccionado para reproducirse, cruzando su material genético con otro individuo seleccionado de igual forma. Este cruce producirá nuevos individuos descendientes de los anteriores, los cuales comparten algunas de las características de sus padres. Cuanto menor sea la adaptación de un individuo, menor será la probabilidad de que dicho individuo sea seleccionado para la reproducción y, por tanto, de que su material genético se propague en sucesivas generaciones.

De esta manera se produce una nueva población de posibles soluciones, la cual reemplaza a la anterior y verifica la interesante propiedad de que contiene una mayor proporción de buenas características en comparación con la población anterior. Así, a lo largo de las generaciones, las buenas características se propagan a través de la población. Favoreciendo el cruce de los individuos mejor adaptados, van siendo exploradas las áreas más prometedoras del espacio de búsqueda. Si el algoritmo genético ha sido bien diseñado, la población convergerá hacia una solución óptima del problema.

A. Esquema de funcionamiento de un algoritmo genético

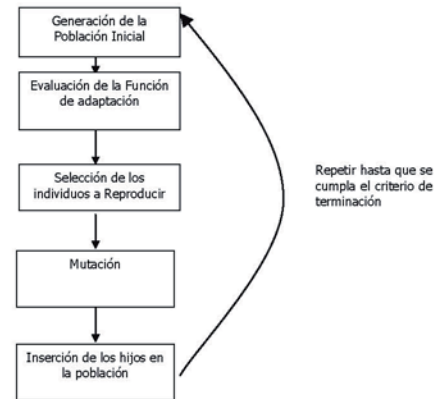


Figura 1. Esquema de funcionamiento de un AG [1]

III. DISEÑO Y ARQUITECTURA DEL SIMULADOR

Durante el desarrollo de esta fase se estudia la interacción, la operación y el comportamiento de las variables que intervienen en el proceso de administración de inventarios y rutas de distribución, según los requisitos preliminares obtenidos en la fase de análisis del sistema, con el fin de determinar los segmentos del proceso que requieren optimización o pueden ser objeto de la aplicación de AGs.

La metodología de desarrollo adoptada toma como referencia las fases y reglas de Extreme Programming (XP), ya que es considerado el más destacado de los procesos ágiles de desarrollo de software y adopta las mejores características de desarrollo de acuerdo a lo que pretende el proyecto, lo cual facilita su aplicación de manera dinámica durante el ciclo de vida del mismo, teniendo en cuenta los recursos humanos y económicos con que se cuenta. Por lo anterior, se elige XP como metodología de desarrollo del presente proyecto ya que, en conclusión, es "un proceso ligero, de bajo riesgo, flexible, predecible, científico y divertido de desarrollar software"[2].

La arquitectura del sistema puede entenderse por medio del análisis de los diferentes diagramas UML elaborados, que constituyen diseños simples y sencillos, fácilmente entendibles e implementables, representando las interacciones organizadas entre objetos y sus vinculaciones, pretendiendo mostrar la manera en que ellos colaboran entre sí. Entre los más representativos tenemos:

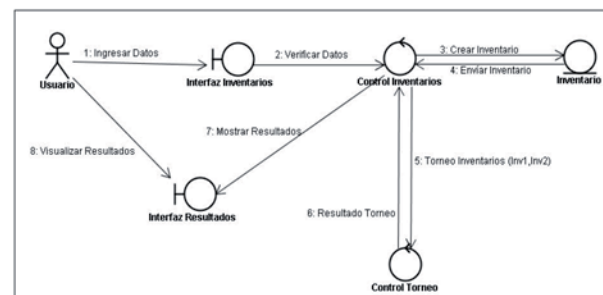


Figura 2. Diagrama de colaboración caso de uso calcular inventarios
Fuente: Autor del Proyecto

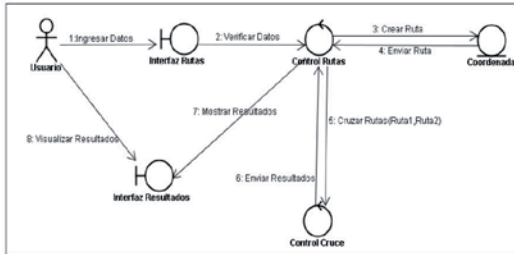


Figura 3. Diagrama de colaboración caso de uso hallar ruta
Fuente: Autor del Proyecto

IV. IMPLEMENTACIÓN DEL SISTEMA, ESCENARIO Y PLANTEAMIENTO DEL MARCO DE GESTIÓN

Para el desarrollo del sistema, se empleó el modelo vista controlador, que se ve representado en el uso de tres capas de programación diferenciando interfaz, control y entidades del sistema, planteadas previamente en el diseño y la arquitectura.

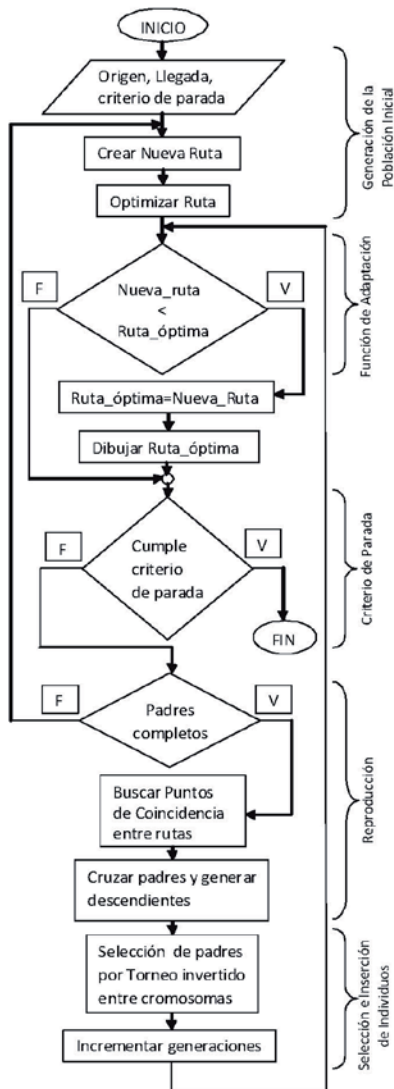


Figura 4. Diagrama de Procedimientos - Hallar Ruta
Fuente: Autor del Proyecto

La Figura 4 representa el diagrama de flujo de procesos empleado para el módulo Hallar Ruta, donde se evidencia el esquema de funcionamiento de un AG.

A. Generación de la población inicial

La población inicial de un AG puede ser creada de muy diversas formas: desde generar aleatoriamente el valor de cada gen para cada individuo, utilizar una función o generar alguna parte de cada individuo y luego aplicar una búsqueda local.

En el caso del uso *calcular inventarios*, la población inicial se crea a partir de un valor de asignado como lote de pedido generado aleatoriamente, que se convertirá en valor binario para dar lugar al primer cromosoma madre, el cual empezará a ser optimizado durante el proceso de acuerdo a la función de adaptación. Posteriormente y de la misma forma, se genera el cromosoma padre, con los cuales iniciará la reproducción por medio del de cruce mono-punto entre padres y torneo entre descendientes, siendo evaluado por la función de adaptación.

Para el caso de uso *hallar ruta*, la población inicial surge creando dos vectores de coordenadas, que se convertirán en padres de la población. El llenado de cada vector inicia con el punto (x, y) de origen seleccionado por el usuario, y continúa con la adición de nuevas posiciones mediante la selección aleatoria de puntos o esquinas relacionadas a esa coordenada según la base de datos, hasta llegar a la coordenada destino de la distribución (Fig. 5). Esta ruta es optimizada suprimiendo los puntos repetidos dentro de su recorrido.

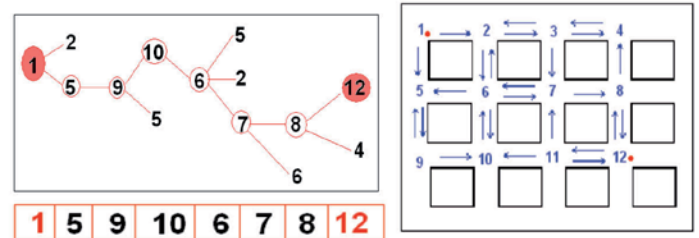


Figura 5. Generación de la población inicial caso de uso Hallar Ruta
Fuente: Autor del Proyecto

B. Función de adaptación

El objetivo de esta función es la optimización de los resultados del proceso. Una característica que debe tener esta función es que debe ser capaz de "castigar" o eliminar las malas soluciones, y de "premiar" o conservar las buenas, de forma que sean estas últimas las que se propaguen con mayor rapidez.

Para el caso del uso *calcular inventarios*, la función de adaptación mide la eficiencia del cromosoma por medio de una resta entre el valor de lote óptimo calculado por el modelo EOQ y el lote aleatorio asignado por el algoritmo, siendo elegido el cromosoma cuyo diferencial obtenga un resultado menor o tienda a cero.

En el caso de uso *hallar ruta*, la adaptación se evalúa según la cantidad de coordenadas que contenga el cromosoma evaluado, siendo mejor adaptado el de menor longitud de ruta, es decir, el que menos coordenadas posea, lo que en la realidad representaría un camino más cercano para llegar al destino.

C. Reproducción

El siguiente paso es generar una nueva población de soluciones a partir de aquellas seleccionadas. A través del operador genético de cruce mono-punto, se crean unas soluciones "hijo" nuevas, que típicamente comparten muchas de las características de sus "padres" y el proceso continúa hasta que una población nueva de soluciones apropiadas se genere [3].

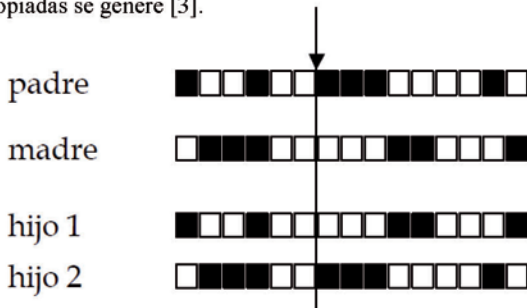


Figura 6. Representación gráfica de un cruce mono-punto [3]

En el módulo de inventarios, se toma aleatoriamente el punto de cruce de los cromosomas *padre* y *madre*, generando un *hijo* que inicia con los genes de la madre hasta el punto de corte y continúa a partir del corte con los genes del padre hasta el punto de llegada.

Para el módulo *hallar ruta de distribución*, se realizará un cruce por cada punto de coincidencia (coordenadas en común) entre los padres, generando dos descendientes por cada punto de coincidencia, iniciando con genes de la madre y terminando con los del padre y viceversa (Fig. 6).

D. Selección e inserción de individuos

El operador de selección a utilizar para el caso de uso *calcular inventarios* es el torneo donde, después de generar los vectores binarios de lote, compiten entre sí de acuerdo a los valores de lote que se someten a la función de adaptación, siendo seleccionado como ganador aquel inventario cuyo diferencial entre el lote óptimo hallado por el método tradicional EOQ y el lote aleatorio del algoritmo se acerque más a cero, el cual efectuará un nuevo cruce y se enviará al próximo torneo para ser enfrentado con el retador descendiente.

Para el caso de uso *hallar ruta*, la selección de individuos se realiza en un "torneo invertido" entre hijos, descartando en cada torneo las dos soluciones menos óptimas, que serán clasificadas por medio de la función de adaptación, que evaluará cuáles tienen el menor número de coordenadas en la ruta. Los dos cromosomas con menor número de coordenadas se convertirán en padres de la nueva generación.

E. Criterio de parada

El proceso del algoritmo se repite hasta que se llega a una condición de finalización. Las condiciones de finalización comunes utilizadas en el simulador son:

- 1) Se ha encontrado una solución que satisface un criterio mínimo. En el caso de cálculo de inventarios, este criterio se cumple cuando el lote aleatorio generado por el algoritmo llega al valor de lote calculado por el método tradicional EOQ básico y representa una reducción en el costo total esperado.

- 2) Se ha llegado a un número de generaciones fijados. En el caso del módulo de rutas de distribución, el usuario o actor del sistema debe indicar qué cantidad de generaciones desea ejecutar para encontrar su ruta.
- 3) Se acaba el presupuesto (tiempo computacional). En caso que el algoritmo no converja con facilidad, cualquiera de los módulos puede agotar los recursos de máquina y debe iniciarse una nueva ejecución.

V. PRUEBAS Y ANÁLISIS DE RESULTADOS

Durante el desarrollo del proyecto se realizó seguimiento de las actividades, procedimientos, pruebas y comportamientos que mostró el sistema en cada fase. Como evidencia se muestran algunos de los resultados obtenidos en las pruebas de cálculo de lote óptimo de inventario con el mínimo de costo total esperado en situaciones diferentes, además de graficar la búsqueda de rutas de distribución:

Ejercicio nro. 1:

- Costo de ordenar= \$55.000 por orden
- Demanda del producto= 8.000 unidades mensuales
- Costo unitario de almacenamiento= \$1.200 por unidad
- Costo unitario de adquisición \$42.000 por unidad

Cálculo del Lote óptimo por el método tradicional:
 $q^* = \sqrt{(2 \cdot k \cdot D / c)} = \sqrt{(2 \cdot 55000 \cdot 8000 / 1200)} = 856,34 = 856 \text{ u/lote}$

Una vez realizado el análisis y ejecución del algoritmo genético, se obtuvieron los siguientes resultados:

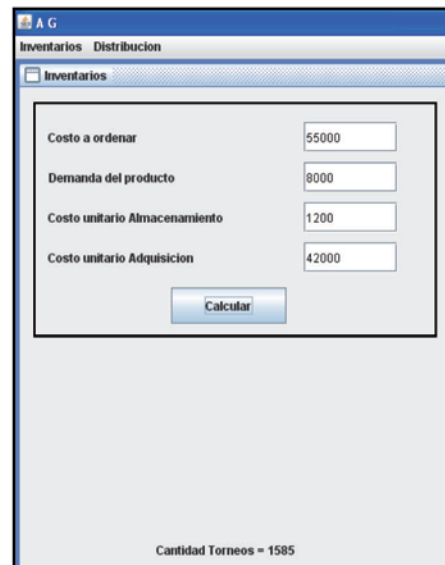


Figura 7. Resultados ejercicio nro. 1 módulo calcular inventarios
 Fuente: Autor del Proyecto

TABLA I
 TABULACIÓN DE RESULTADOS EJERCICIO NRO. 1

Nro. de Iteraciones	Lote de Pedido	Pedidos a Efectuar	Costo Total
1585	856	9	1.008.600
103	857	9	1.009.200
85	862	9	1.012.000
44	884	9	1.025.400

Ejercicio nro. 2:

En esta prueba, el usuario del simulador selecciona para el cálculo de la ruta de distribución los siguientes datos:

Origen: Esquina 12
Destino: Esquina 28



Figura 8. Resultados Ejercicio nro. 2 módulo hallar ruta
Fuente: Autor del Proyecto

Coordenada X1 = 237	Coordenada Y1 = 387
Coordenada Y2 = 160	Coordenada Y2 = 227
Generaciones 17 <input type="text" value="20"/>	
<input type="button" value="Hallar ruta"/>	

Generacion	Cantidad coordenadas
5	8
3	28
1	29

Figura 9. Tabulación de resultados ejercicio nro. 2
Fuente: Autor del Proyecto

VI. CONCLUSIONES

Teniendo en cuenta lo observado durante las fases, además de lo planteado en los objetivos, se puede afirmar, con base en los resultados obtenidos durante el proceso completo de análisis, diseño, implementación y pruebas, que la técnica de AG es una alternativa de optimización valiosa al momento de proponer una solución óptima en un contexto incierto y variable.

La implementación de la técnica de AGs constituye una herramienta útil para la toma de decisiones, ya que por medio del análisis de los resultados permite anticipar el comportamiento de los costos totales en función de los cambios de las variables que inciden en el proceso de inventarios. Además de considerar que el tiempo de respuesta de las soluciones sugeridas es altamente menor al que tendría que usarse en el desempeño netamente humano.

REFERENCIAS

- [1] E. Yolis. *Algoritmos genéticos Aplicado a la Categorización de Documentos*. Trabajo de Investigación. Universidad de Buenos Aires.
- [2] K. Beck. *Extreme Programming Explained*. Addison-Wesley Professional, 2001
- [3] L. Recalde. *Esquemas algorítmicos - Algoritmos genéticos (en línea)*. Disponible en: <http://webdiis.unizar.es/asignaturas/EDA/ea/slides/9- Algoritmos%20geneticos.pdf>