

**COMBINACIÓN DE TECNICAS DE MINERIA DE DATOS PARA
RESOLVER UN PROBLEMA DE OPTIMIZACION: UBICACIÓN
DE PRODUCTOS EN UN CENTRO DE DISTRIBUCION**

DIEGO ORLANDO BUITRAGO CARRILLO

**UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERIA
MAESTRÍA EN INGENIERIA DE SISTEMAS Y COMPUTACIÓN
BOGOTÁ, COLOMBIA
2006**

**COMBINACIÓN DE TECNICAS DE MINERIA DE DATOS PARA
RESOLVER UN PROBLEMA DE OPTIMIZACION: UBICACIÓN
DE PRODUCTOS EN UN CENTRO DE DISTRIBUCION**

DIEGO ORLANDO BUITRAGO CARRILLO

**Tesis de grado para optar al título de:
Magíster en Ingeniería de Sistemas y Computación**

**Asesor:
Dr. JOSÉ ABÁSULO PRIETO**

**UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERIA
MAESTRÍA EN INGENIERIA DE SISTEMAS Y COMPUTACIÓN
BOGOTÁ, COLOMBIA
2006**

*Dedicado a mis padres,
a mi esposa y al hermoso
bebé que está en camino*

AGRADECIMIENTOS

Los más sinceros agradecimientos a:

El doctor José Abásolo Prieto, asesor de la tesis, por sus valiosos aportes en todo el desarrollo del proyecto.

Gustavo Adolfo Torres, Juan Carlos Cruz, Ana Marlén López, Sandra Milena Sánchez y Nelson Gutiérrez, todos ellos colaboradores de Carvajal S.A por su apoyo y constante colaboración en la obtención de información.

A todos los profesores y personal administrativo del Departamento de Ingeniería de Sistemas de la Universidad de los Andes por todo el desarrollo académico, profesional y personal que cultivaron en mí a lo largo de estos últimos años.

CONTENIDO

INTRODUCCION	10
OBJETIVOS	11
JUSTIFICACION	12
1 DESCRIPCION DEL PROBLEMA	13
1.1 CONTEXTO	13
1.2 RESTRICCIONES CONSIDERADAS	15
1.3 FORMALIZACIÓN DEL PROBLEMA	16
1.4 COMPLEJIDAD DEL PROBLEMA	18
2 ESTADO DEL ARTE EN EL DISEÑO DE CENTRO DE DISTRIBUCIÓN	20
3 MARCO TEORICO	23
3.1 EL MARKET BASKET ANALYSIS	23
3.2 LOS ALGORITMOS GENETICOS	27
3.2.1 Elementos de un algoritmo genético	27
3.2.2 Pseudocódigo de un algoritmo genético	28
3.2.3 ¿Por qué funcionan los Algoritmo Genéticos?	29
3.2.4 Operadores genéticos de cruce y mutación	30
3.2.5 Algoritmos genéticos en problemas de permutación	31
4 Estrategia de Solución	34
4.1 Análisis de la información histórica	34
4.2 Aplicación de los Algoritmos Genéticos al Problema	35
4.2.1 Modelo de codificación	35
4.2.2 Función de evaluación	38
4.2.3 Mecanismo de selección	39
4.2.4 Operadores genéticos	40
4.2.5 Criterio de parada	42
5 PROTOTIPO DE EVALUACIÓN Y PRUEBAS	43
5.1 Definición del centro de distribución y los grupos de ítems frecuentes	43
5.2 Implementación del Algoritmo Genético al prototipo	46
5.2.1 Modelo de codificación	46
5.2.2 Función de evaluación	47
5.2.3 Operadores genéticos	47
5.2.4 Criterio de parada	48
5.3 Programación del Algoritmo Genético	48
5.4 Experimentación y análisis de resultados	50
5.4.1 Efecto de la mutación	50
5.4.2 Efecto del cruce	51
5.4.3 Efecto del tamaño de la población	53
5.4.4 Efecto de la aleatoriedad	56
5.5 Herramienta de simulación para comparación de resultados	57
5.5.1 Descripción funcional de la herramienta	57
5.5.2 Generación de pedidos entrantes	58
5.5.3 Diseño de la herramienta	58

5.5.4	Ejecución de la herramienta con diferentes distribuciones de productos	60
5.5.5	Evaluación de resultados	62
6	IMPLEMENTACIÓN EN UN CASO REAL	64
6.1	Obtención de información	64
6.2	Limpieza de información	65
6.3	Soporte mínimo, máximo y promedio	65
6.4	Modelo de asociación	65
6.5	Items frecuentes Obtenidos	66
6.6	Ajustes al algoritmo genético	67
6.6.1	Función de distancia	67
6.6.2	Programación del algoritmo	68
6.7	Ajustes al modelo de simulación	68
6.8	Resultados Obtenidos	68
7	PROCEDIMIENTO PROPUESTO	71
7.1	Etapa 1 “Análisis de Información”	71
7.1.1	Requerimientos	71
7.1.2	Actividades	71
7.1.3	Software disponible	72
7.1.4	Resultados esperados	72
7.1.5	Sugerencias	72
7.2	Etapa 2 “Desarrollo del Algoritmo Genético”	72
7.2.1	Requerimientos	73
7.2.2	Actividades	73
7.2.3	Software disponible	73
7.2.4	Resultado	74
7.3	Etapa 3 “Evaluación de resultados”	74
7.3.1	Requerimientos	74
7.3.2	Actividades	74
7.3.3	Software disponible	75
7.3.4	Resultado	75
	CONCLUSIONES	77
	BIBLIOGRAFIA	79

LISTA DE FIGURAS

Figura 1. Costos por actividad dentro de un centro de distribución	14
Figura 2. Interpretación del problema	17
Figura 3. Cruce de Productos	24
Figura 4. Taxonomía	26
Figura 5. División del espacio de soluciones de un problema.	29
Figura 6. Codificación binaria del centro de distribución	36
Figura 7. Codificación en un arreglo 3-dimensional	36
Figura 8. Codificación de arreglo con listas	37
Figura 9. Mecanismo de cruce	41
Figura 10. Mecanismo de mutación	42
Figura 11. Nomenclatura de las localizaciones	43
Figura 12. Representación del Centro de distribución	44
Figura 13. Posibles ubicaciones de productos en el Centro de Distribución	44
Figura 14. Distribución de productos considerada	45
Figura 15. Codificación de la distribución base	46
Figura 16. Cálculo de distancia entre dos localizaciones	47
Figura 17. Mecanismo de cruce aplicado al prototipo	48
Figura 18. Mecanismo de mutación aplicado al prototipo	48
Figura 19. Efecto de la mutación	51
Figura 20. Efecto del Cruce	52
Figura 21. Efecto Tamaño Población	53
Figura 22. Adaptación Media por Población	54
Figura 23. Efecto de la Aleatoriedad	56
Figura 24. Distribución de productos sugerida por el algoritmo genético	60
Figura 25. Distribución de productos determinada por la frecuencia de venta	61
Figura 26. Distancia de pedidos	62
Figura 27. Distancia promedio prototipo	63
Figura 28. Distancia recorrida en los pedidos	68
Figura 29. Distancia promedio caso real	69
Figura 30. Etapas de ejecución del proyecto	76

LISTA DE TABLAS

Tabla 1. Ventajas y desventajas de los modelos de codificación considerados	38
Tabla 2. Tamaño de muestra a diferentes niveles de confianza	55
Tabla 3. Librerías para desarrollo de algoritmos genéticos	74

LISTA DE ANEXOS

ANEXO 1. DIAGRAMA DE CLASES DEL ALGORITMO GENÉTICO	81
ANEXO 2. DIAGRAMA DE SECUENCIA PRINCIPAL	82
ANEXO 3. DIAGRAMA DE CLASES DEL MODELO DE SIMULACIÓN	83
ANEXO 4. RESULTADOS EN INTELLIGENT MINER	84
ANEXO 5. ITEMS FRECUENTES ENCONTRADOS	85
ANEXO 6. DESCRIPCION FUNCIONAL DEL SOFTWARE	86

INTRODUCCIÓN

El problema de encontrar una organización de productos dentro de un centro de distribución, que minimice las distancias que los operarios deben recorrer para obtener los pedidos, es en general un problema de permutación y por tanto evaluar todas las posibles soluciones resulta computacionalmente ineficiente.

Los problemas de permutación han sido objeto de cientos de estudios y han generado una importante cantidad de métodos de solución. Sin embargo, hasta el momento ningún método ha sido capaz de encontrar soluciones óptimas en todos los casos y por tanto ninguno ha sido globalmente aceptado.

El proyecto que se describe en este documento busca evaluar una alternativa de solución al problema, que involucra el utilizar dos técnicas de minería de datos. Por una parte los algoritmos genéticos, como mecanismo para explorar el espacio de posibles soluciones y por otro lado el Market Basket Analysis, como mecanismo para encontrar relaciones entre los productos que en muchas ocasiones no resultan obvias y con estas poder diferenciar una buena solución al problema de otra que no lo sea.

Este documento comienza con los objetivos que se buscaron con el proyecto, luego describe en detalle cada una de las características particulares del problema planteado junto con las restricciones a que hubo lugar con el fin de acotarlo, posteriormente enuncia los criterios que tradicionalmente son utilizados para encontrar soluciones al problema, más adelante presenta una breve revisión bibliográfica tanto de los algoritmos genéticos como del Market Basket Analysis, que pretende ofrecer un contexto y establecer una terminología que permitan entender más fácilmente el resto del documento, luego describe en detalle la estrategia de solución que se plantea y por último muestra los resultados obtenidos al aplicar la estrategia en un caso experimental y un caso real. Como consolidación de las experiencias adquiridas a lo largo del proyecto se presenta por último un procedimiento en el que se incluyen todos los pasos que se sugiere seguir al momento de aplicar la técnica propuesta a un problema de las mismas características.

OBJETIVOS

OBJETIVO GENERAL

Analizar la viabilidad de resolver un problema de optimización consistente en determinar la organización de productos que minimice la distancia que recorren operarios para despachar ordenes de pedido dentro de un centro de distribución, utilizando algoritmos genéticos y conjuntos de ítems frecuentes (Market Basket Analysis).

OBJETIVOS ESPECIFICOS

- Profundizar el estudio de la técnica de algoritmos genéticos como herramienta para resolver problemas de optimización.
- Explorar procedimientos que permitan utilizar los resultados del Market Basket Analysis como insumo de una función de adaptación, requerida por el algoritmo genético.
- Desarrollar una aplicación prototipo, basada en un caso real, para ilustrar los pasos a seguir al tratar de resolver un problema de optimización con algoritmos genéticos, y a la vez evaluar la bondad de los modelos utilizados.

JUSTIFICACION

Con el auge de las ventas por catálogos e Internet, cada vez es más común que las empresas dejen de abrir almacenes a los que vayan sus clientes y en cambio construyan grandes centros de distribución en los que se armen los pedidos y se despachen al lugar que se designe.

La forma como los productos estén organizados dentro de los centros de distribución es clave para el éxito del negocio. Una buena distribución puede lograr que los operarios gasten menos tiempo recogiendo los productos, evitar que se cometan errores en la conformación de los pedidos y optimizar las políticas de reabastecimiento, entre otras.

Buscar distribuciones de productos en centros de distribución, que optimicen uno o varios objetivos es un problema de complejidad exponencial sobre el que hacer una búsqueda exhaustiva resulta poco factible.

Tradicionalmente, para determinar las ubicaciones de los productos en los centros de distribución se calcula la rotación de cada uno y se asignan las ubicaciones más cercanas a los productos con mayor rotación. Sin embargo, esto puede generar congestión en pequeños sectores del centro de distribución.

El trabajo investigativo que se plantea en este documento busca una nueva forma de obtener solución al problema anteriormente descrito mediante la utilización de dos técnicas de minería de datos: los algoritmos genéticos (como mecanismo de búsqueda de solución) y el Market Basket Análisis (como forma de obtener reglas del negocio que permitan evaluar cada una de las soluciones que se planteen).

1 DESCRIPCION DEL PROBLEMA

1.1 CONTEXTO

Los Centros de Distribución son negocios en los que la mayor parte de recursos y esfuerzos se concentran en administrar eficientemente bodegas en las que ingresan mercancías adquiridas en grandes volúmenes y salen bienes con un valor agregado, proveniente de unir productos que originalmente no lo estaban o de crear nuevos productos compuestos (como es el caso de los distribuidores de computadores).

La administración de una bodega requiere de una planeación exhaustiva en la que se tenga en cuenta el flujo del negocio, el cual usualmente comprende, según [BAR05]:

- Entrada de mercancía, que incluye:
 - Recibir la mercancía que llega de los proveedores en empaques usualmente grandes.
 - Ubicar estratégicamente la mercancía para que pueda ser almacenada y posteriormente desglosada en unidades más pequeñas.
- Salida de mercancía, que normalmente comprende:
 - Recibir los pedidos de los clientes, verificar si hay existencias suficientes y producir listados u órdenes que permitan la consecución de los productos.
 - Recoger cada uno de los productos de los listados y armar los pedidos de los clientes.
 - Hacer una verificación de que todos los productos requeridos por los clientes fueron obtenidos.

- Empacar cada uno de los pedidos en cajas o empaques adecuados para las características de los productos.
- Despachar los pedidos para que sean entregados en las ubicaciones que los clientes requieran.

Según [BAR05], en el proceso de recolección y armado de los pedidos se concentra el 55% de los costos operativos de la bodega.

De acuerdo a otra fuente [FRA03], el alistamiento de los productos en el armado de los pedidos, representa el 62% de los costos de un centro de distribución. En la figura 1 se puede ver la distribución de los costos entre las diferentes actividades, de acuerdo con lo planteado por [FRA03].

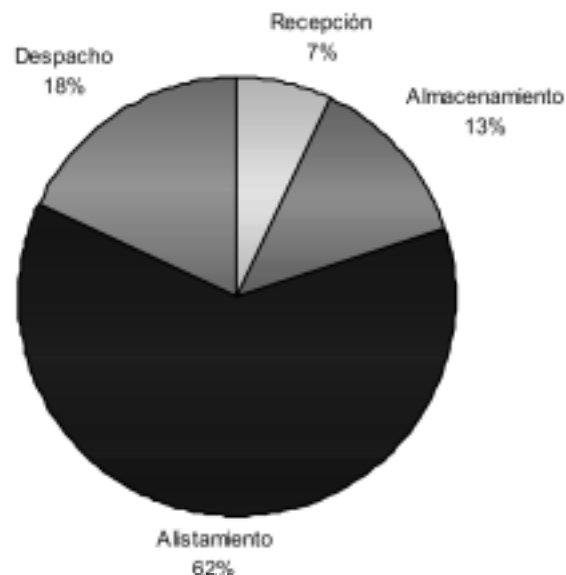


Figura 1. Costos por actividad dentro de un centro de distribución según [FRA03]

Normalmente la recolección y armado de los pedidos se divide en las siguientes subactividades:

- Desplazamiento de los operarios a través de la bodega.

- Búsqueda de los productos.
- Extracción u obtención de los productos.
- Labores administrativas del pedido como por ejemplo registrar la salida de los productos.

De estas, el desplazamiento de los operarios a través de la bodega es, de acuerdo a [BAR05], la subactividad más costosa ya que representa el 55% del costo total de la actividad. Teniendo en cuenta el costo que tiene el proceso de recolección y armado de los pedidos dentro del flujo de trabajo de una bodega, se puede llegar a la conclusión de que el desplazamiento de los operarios concentra el 30.25% de los costos operativos de la bodega.

Este resultado nos lleva a concluir que la ubicación o distribución que tengan los tipos de productos dentro de las bodegas es un punto neurálgico a ser tenido en cuenta en la búsqueda de la reducción de los costos operativos de los centros de distribución.

1.2 RESTRICCIONES CONSIDERADAS

Luego de las primeras secciones de este documento, queda la sensación de que el problema sobre el que se pretende trabajar es: dado un centro de distribución de mercancías, encontrar la ubicación óptima de cada uno de los productos dentro de una bodega para minimizar las distancias que recorren los operarios encargados de recoger cada uno de los pedidos entrantes, y de esta manera reducir los costos operativos del negocio. Sin embargo, este problema es demasiado complejo para ser considerado con los recursos con que se cuenta, razón por la cual se hace necesario establecer un conjunto de restricciones que lo acoten sin volverlo trivial. Las restricciones que se establecieron fueron:

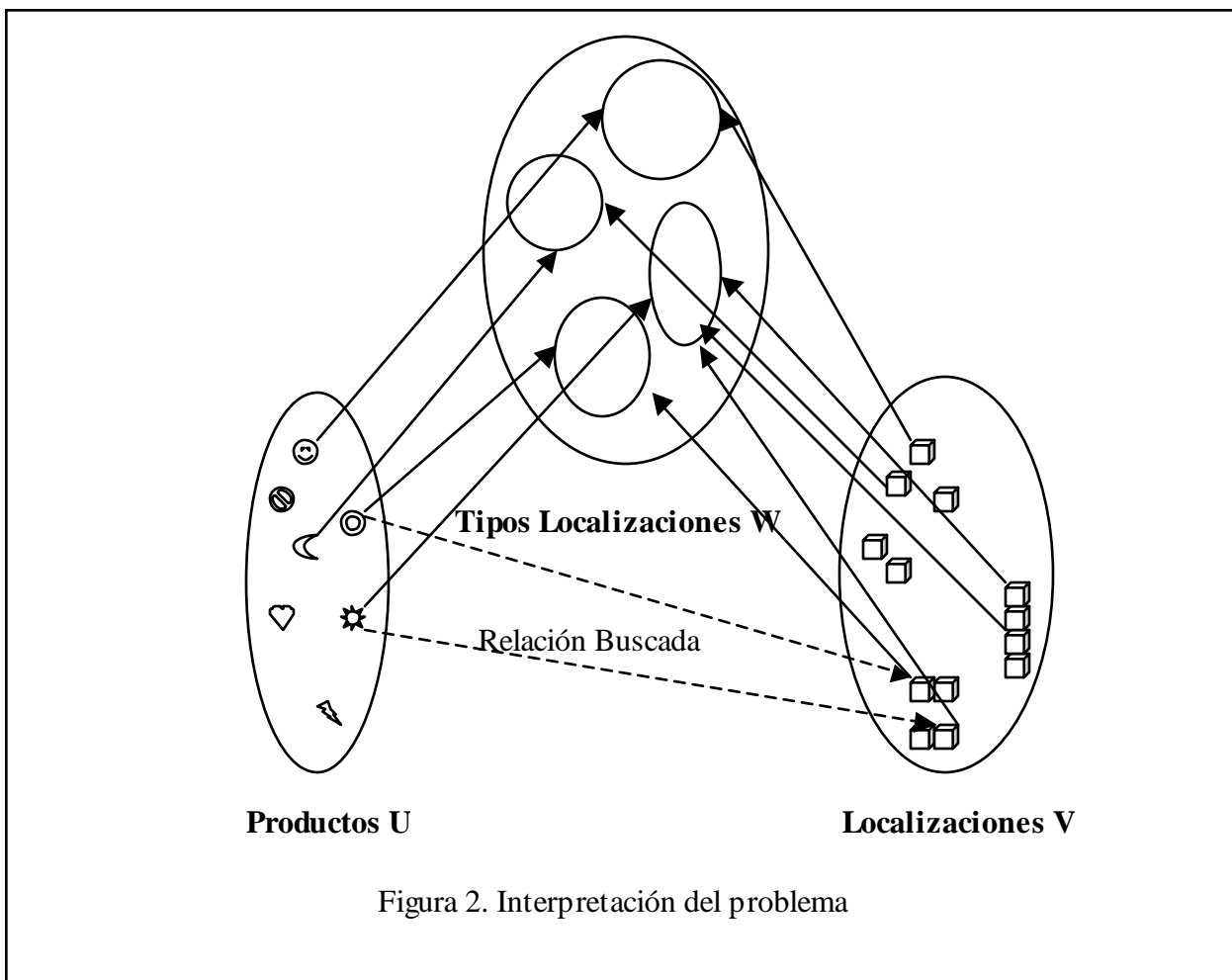
- El método resultante solo podrá ser aplicado en centros de distribución que lleven algún tiempo operando y registrando la información de los diferentes pedidos que se les hayan hecho.
- Un tipo de producto se ubica en una sola localización de la bodega.
- Cada una de las localizaciones de la bodega, debido a sus características, se clasifica en una clase o tipo de localización.
- Para cada uno de los tipos de productos que se comercializan se tiene determinado el tipo de localización que se debe utilizar para almacenarlo.
- Existe una política de reabastecimiento que garantiza que siempre existe producto disponible en la localización asignada a este.

1.3 FORMALIZACIÓN DEL PROBLEMA

Con estas restricciones consideradas, el problema se puede formalizar de la siguiente manera:

Existe un conjunto de tipos de producto U , un conjunto de localizaciones V de una bodega y un conjunto W de tipos de localizaciones. Cada localización es de un tipo y varias localizaciones pueden ser de un mismo tipo. Un tipo de producto solo puede ser ubicado en un tipo de localización y a un mismo tipo de localización pueden corresponder varios tipos de productos.

Problema: Se necesita encontrar una relación uno a uno entre U y V que permita a un operario recoger una lista de productos dentro de la bodega, recorriendo la menor distancia posible.



Con la anterior interpretación del problema y suponiendo que existen igual cantidad de tipos de productos que de localizaciones, el número de posibles soluciones del problema se calcula como:

$$\# \text{soluciones} = \prod_{i=1}^k (|V_i|)!$$

donde k es la cantidad de tipos de localizaciones o elementos de W y V_i es el subconjunto de localizaciones que están relacionadas con el i -ésimo elemento de W .

El caso más sencillo ocurriría cuando existan igual número de elementos en W y V y el caso más complejo cuando solo exista un elemento en W .

El aspecto que más dificulta poder llegar a una solución óptima para el problema es que se quiere optimizar la ubicación de los tipos de productos para recoger listas de productos que no se conocen anticipadamente.

1.4 COMPLEJIDAD DEL PROBLEMA

El problema de ubicar tipos de productos en una bodega es altamente complejo por dos razones fundamentales:

- Aunque es posible estimar la forma como van a estar compuestos los pedidos que lleguen al centro de distribución, no se puede determinar con exactitud la composición de estos.
- El número de posibles distribuciones de los productos dentro de la bodega crece exponencialmente con la cantidad de productos y ubicaciones disponibles.

Este último hecho se puede mostrar de la siguiente forma:

Supongamos, para efectos de la demostración, que una bodega cuenta con m localizaciones de igual tamaño, que se requiere ubicar n tipos de productos en la bodega, que cada una de las localizaciones es suficiente para alojar a uno y solo un tipo de producto y que $m \geq n$.

Una buena política para evitar alojar productos en lugares muy apartados es dejar vacías las $m-n$ localizaciones más lejanas al punto de inicio de recolección, con lo cual la localizaciones disponibles m' serían $m'=m-(m-n)=n$.

Utilizando algún criterio podríamos ordenar las m' localizaciones disponibles y analizar la forma como podríamos llenarlas:

- *En la primera localización se podría ubicar cualquiera de los n tipos de productos.*
- *En la segunda localización se podría ubicar cualquiera de los n tipos de productos, excepto el que se ubicó en la primera localización.*

- *En la tercera localización se podría ubicar cualquiera de los n tipos de productos, excepto los que se ubicaron en la primera y segunda localización.*

Esta progresión continúa hasta ocupar la última localización disponible con el tipo de producto que falte. El número de formas distintas de ubicar los n tipos de productos en las m' ubicaciones esta entonces dado por:

$$\text{distribuciones}(m') = n \cdot (n-1) \cdot (n-2) \cdot (n-3) \cdot \dots \cdot (1) = n!$$

2 ESTADO DEL ARTE EN EL DISEÑO DE CENTRO DE DISTRIBUCIÓN

Gracias a la alta complejidad, el problema de distribución de tipos de productos no cuenta con un único método de solución globalmente aceptado. Por el contrario, tiene una amplia gama de métodos de solución, cada uno de los cuales está fundamentado en una o en una combinación de estrategias. En [BAR05] se describen las siguientes:

- Ubicar los productos con mayor rotación en localizaciones físicamente cercanas.
- Hacer que los operarios recojan varios pedidos a la vez.
- Distribuir un mismo pedido entre varios operarios cada uno de los cuales se especializa en un sector de la bodega.

En [FRA03] se encuentran estrategias equivalentes, presentadas a manera de principios de diseño de un centro de distribución:

- Eliminar y combinar actividades cuando sea posible. Por ejemplo:
 - Caminar y extraer productos
 - Caminar y documentar
 - Seleccionar y ordenar
 - Seleccionar, ordenar y empacar
- Asignar los productos más populares a las ubicaciones de más fácil acceso dentro de la bodega, preferiblemente:
 - Al final de las estanterías, cerca de las puertas de despacho
 - En las estanterías, a la altura de la cintura de los operarios
- Balancear el proceso de selección a lo largo de todas las posiciones de estantería para reducir congestión localizada.
 - Evitar pasillos muy visitados a la vez.

- Asignar aquellos productos que se piden juntos en la misma posición de estantería.
 - Identificar la correlación que existe entre los productos de acuerdo al perfil de los pedidos.
- Establecer áreas de selección y de reserva independientes.
 - El área de reserva debe ser de mayor tamaño para almacenar grandes cantidades.
 - El área de selección debe ser condensada para permitir tener variedad de productos en pequeñas cantidades
- Agrupar pedidos similares y de tamaño reducido para minimizar el tiempo total recorrido.
 - Incrementar el número de pedidos que alista un operario durante un tour para disminuir el tiempo de recorrido
- Diseñar los recorridos o rutas dentro de la bodega de tal forma que se reduzca el tiempo de recorrido.
 - Por lo general se deben visitar varias posiciones de estantería para completar un pedido, tener un recorrido diseñado facilita la operación.
- Diseñar listas de alistamiento claras y de fácil comprensión, así como señalización en las estanterías, con el propósito de minimizar el tiempo de búsqueda y los errores.
 - Usar letras de tamaño apropiado, colores y símbolos que eviten confusiones.
- Diseñar vehículos que permitan al operario sentirse cómodo, minimizando el tiempo de extracción, ordenamiento y los errores.
 - La productividad del operario depende de su puesto de trabajo y en este caso es el vehículo de recorrido.
- Seleccionar la mejor combinación equipo-operario que se ajuste a los requerimientos del sistema.
 - Tener en cuenta las características tanto de la estantería como de los equipos de manejo de materiales.

Dependiendo la estrategia o la combinación de estrategias que se utilice para proponer soluciones, se obtienen ventajas y desventajas. Si por ejemplo se utiliza la estrategia de ubicar los productos con mayor rotación en localizaciones físicamente cercanas, se obtiene la gran ventaja de que los pedidos que únicamente involucran productos de alta rotación se pueden alistar con un mínimo recorrido, pero también se genera la desventaja de tener demasiado tráfico de operarios en espacios reducidos. Otro ejemplo es aplicar la estrategia de hacer que los operarios recojan varios pedidos a la vez; esta estrategia permite obtener un mayor rendimiento en la recolección de los pedidos pero genera una demora adicional en el empaque.

3 MARCO TEORICO

Como marco teórico se presenta una breve descripción de las dos técnicas de minería de datos que se utilizan en la estrategia de solución: el Market Basket Analysis y los algoritmos genéticos.

3.1 EL MARKET BASKET ANALYSIS

El Market Basket Analysis es una técnica de minería de datos cuyo objetivo es determinar conjuntos de productos que tienden a venderse juntos [MEG04]. Esta información es usualmente muy importante para los departamentos de mercadeo en las empresas porque permite diseñar nuevas estrategias de venta.

Este tipo de análisis tiene varias características que lo hacen atractivo [MEG04]:

- Es una técnica de minería de datos no dirigida, por lo cual no se concentra en puntos específicos sino en el total de la información.
- El análisis arroja información que en muchos casos no resulta obvia.
- Los resultados pueden conducir rápidamente a la toma de decisiones.

Normalmente el Market Basket Analysis utiliza como materia prima un conjunto de transacciones de clientes en las que se detallan los productos que se despacharon. Con esta información, construye matrices de coocurrencia en la que cruza los productos para determinar que tan comúnmente aparecen los productos juntos. Por ejemplo en la Figura 3 se cruzan productos uno a uno, con lo cual se podrían evaluar reglas del tipo producto x entonces producto y. En esta matriz la diagonal indica la cantidad de veces que los productos aparecieron en las transacciones.

	Producto 1	Producto 2	Producto 3	...	Producto n
Producto 1	4	1	2	...	1
Producto 2	1	3	2	...	2
Producto 3	2	2	4	...	0
...
Producto n	1	2	0	...	1

Figura 3. Cruce de Productos

En la práctica se utilizan cruces de información mas complejos que el de la matriz de coocurrencia simple presentada en la Figura 3, en los que se incluye por ejemplo elementos ficticios que permitan analizar relaciones que involucren varios productos a la vez o que permitan incluir otros aspectos importantes como por ejemplo el tipo de clientes al que se quiere llegar.

Un aspecto que es muy importante en el Market Basket Analysis es determinar las reglas que puedan llegar a ser más útiles para quienes efectúan el análisis; para esto normalmente se calculan las siguientes tres medidas en cada una de las reglas:

- **Soporte:** que es el porcentaje resultante de dividir la cantidad de veces que se presentó una regla entre el número total de transacciones analizadas. Por ejemplo, si luego de analizar 5400 transacciones se determinó que la regla “lápiz entonces papel” apareció en 3200 transacciones, la cobertura de esta regla es $3200/5400 = 60\%$
- **Confianza:** que es el porcentaje resultante de dividir la cantidad de veces que se presentó una regla entre el número de veces que la hipótesis de la regla apareció en las transacciones. Esto en términos estadísticos es la probabilidad condicional de un producto dado otro. Por ejemplo si se retoma la regla “lápiz entonces papel” y se encuentra que el lápiz por si solo apareció en 3600 transacciones, se puede calcular la confianza de la regla como $3200/3600 = 90\%$. Con esta medida se puede por tanto diferenciar la regla “lápiz entonces papel” de la regla “papel entonces lápiz”.

- **Mejoramiento:** esta medida busca detectar reglas que a pesar de tener altos niveles de cobertura y confianza no son útiles porque no predicen el comportamiento de los clientes mejor que una selección aleatoria. El mejoramiento se calcula como:

$$\text{mejoramiento} = \frac{\text{cobertura}(\text{hipótesis} + \text{resultado})}{\text{cobertura}(\text{hipótesis}) \times \text{cobertura}(\text{resultado})}$$

Las reglas que predicen adecuadamente los comportamientos de los clientes tienen porcentajes de mejoramiento mayores a 1.

Un obstáculo que en algunas ocasiones evita obtener buenos resultados de un análisis de este tipo es [MEG04] la presencia de productos que tienen muy alto o muy bajo soporte. El problema radica en que aquellos productos con alto soporte evitan encontrar reglas de asociación e ítems frecuentes sobre aquellos productos con bajo soporte porque los niveles de confianza que se calculan para estos son muy bajos. Una alternativa a este problema es eliminar los productos que tienen soportes muy altos, con la gran desventaja de que estos no serán tenidos en cuenta para el análisis. La otra alternativa es agrupar varios productos con bajo soporte en una misma categoría la cual contará con un soporte más grande. A esta alternativa se le conoce como creación de Taxonomías. En la figura 4 se ilustra el concepto alrededor de una taxonomía.

Una ventaja importante de las taxonomías es que permiten agrupar los productos con base en diferentes criterios (uno a la vez) y de esa manera orientar los análisis hacia aspectos que sean de interés para el negocio (por ejemplo agrupar productos por el tipo, por el tamaño, por el peso, etc)

Los programas que permiten realizar Market Basket Analysis, son capaces de asociar taxonomías con los productos y encontrar relaciones entre categorías de diferente nivel.

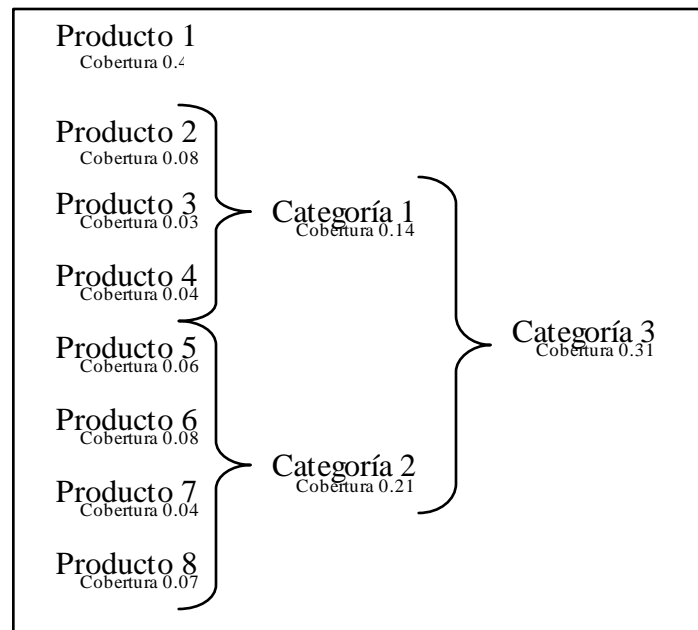


Figura 4. Taxonomía

Como parte de este rápido vistazo sobre el Market Basket Analysis también resulta importante mencionar las limitaciones e inconvenientes que pueden ocurrir cuando se efectúan análisis de este tipo [MEG04]:

- Se requieren grandes volúmenes de información para detectar reglas útiles.
- Pueden surgir reglas demasiado obvias que no agreguen ninguna nueva información.
- Por otro lado también se pueden producir reglas que no tengan ningún sentido y que no sirvan para ser aplicadas en ningún contexto.
- Cuando uno o unos pocos productos se venden en un porcentaje mucho mayor que los demás, puede llegar a ser difícil encontrar reglas útiles.
- En las reglas que se descubran se pueden reflejar los resultados de promociones o campañas publicitarias previas.

3.2 LOS ALGORITMOS GENETICOS

Los algoritmos genéticos son un método de solución de problemas de optimización, inspirado en el funcionamiento de la reproducción y la selección natural. El método fue desarrollado por John H. Holland y sus colaboradores en 1975 y su idea principal es generar una gran cantidad de individuos a los que se les denomina “generación” (que en general es un conjunto de soluciones codificadas de un problema) y evaluarla para determinar cuales de ellos se adaptan mejor (solucionan mejor el problema) para permitirles multiplicarse y cruzarse con el objetivo de producir una generación que estadísticamente debe estar mejor “adaptada”. Esto se repite cientos de veces hasta se acepte alguna de las soluciones.

3.2.1 Elementos de un algoritmo genético

En los algoritmos genéticos comúnmente interactúan los siguientes elementos [KOO04]:

- Cromosoma: es la codificación de una solución al problema.
- Gen: es cada una de las partes constituyentes de un cromosoma.
- Población: grupo de cromosomas en una iteración particular del algoritmo.
- Función de evaluación: es una función que determina que tan bien un cromosoma resuelve el problema considerado.
- Función de adaptación: es una función que determina la posibilidad de reproducción que tiene un cromosoma.
- Mecanismo de selección: estrategia que se utiliza para decidir que cromosomas se seleccionan para pasar a la siguiente generación.
- Operadores genéticos: funciones que modifican los cromosomas para intentar producir nuevos individuos. Los operadores genéticos tradicionales son el cruce y la mutación.

- Criterio de parada: condición que se evalúa para determinar si el algoritmo continúa.

3.2.2 Pseudocódigo de un algoritmo genético

A continuación se presenta el pseudocódigo del algoritmo genético desarrollado por Holland en su versión resumida [WHI93] página 7:

1. *Calcular la población inicial*
2. *Aplicar la función de evaluación a cada cromosoma de la Población*
3. *Seleccionar el cromosoma con el mayor valor en la función de adaptación*
4. *Ejecutar cruce con probabilidad p_c . Si no se realiza el cruce, colocar el cromosoma en Nueva_Población e ir al paso 5. En caso contrario:*
 - a. *Seleccionar un cromosoma de la población con probabilidad uniforme*
 - b. *Seleccionar el punto de cruce entre 1 y L-1 con probabilidad uniforme*
 - c. *Recombinar los cromosomas y colocarlos en Nueva_Población*
5. *Si Nueva_Población no está llena, vaya al paso 3*
6. *Si Nueva_Población está llena, haga Población = Nueva_Población y vaya al paso 2*

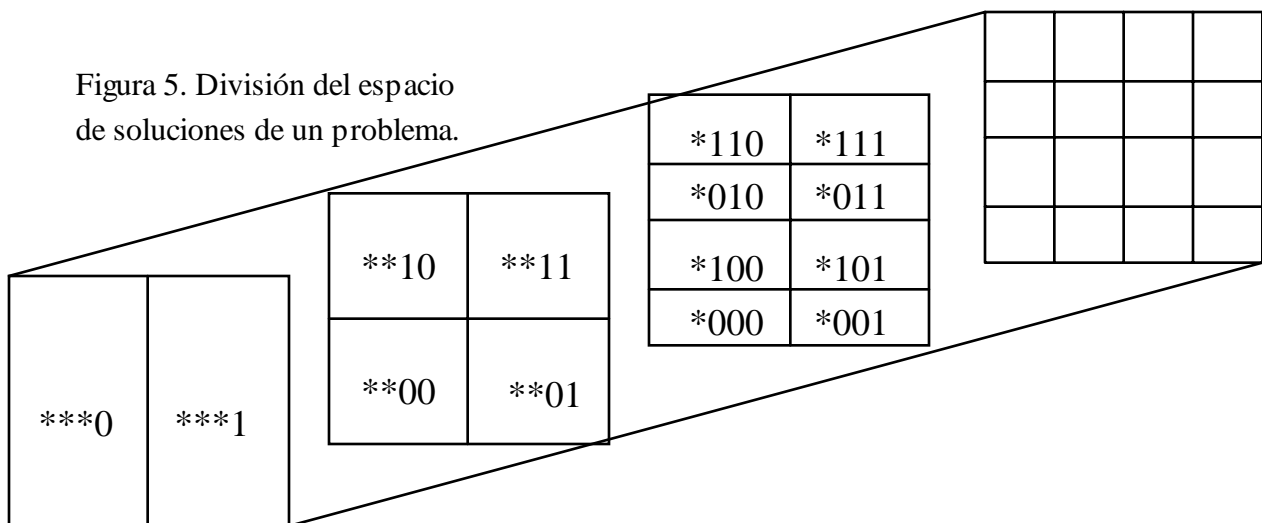
En el paso 1 se crean aleatoriamente un grupo de cromosomas y se conforma la población inicial. En el siguiente paso se aplica la función de evaluación a cada uno de los cromosomas de la población para luego obtener la adaptación de cada uno. En el paso 3 se selecciona el cromosoma con la mejor adaptación y se remueve de la población. En el paso 4 se genera un número aleatorio para determinar si sobre el cromosoma seleccionado se aplica o no el operador de cruce. Si no se aplica el operador de cruce, el cromosoma se incluye en la siguiente población, si se aplica el operador de cruce se debe seleccionar aleatoriamente otro cromosoma, generar un número aleatorio entre 1 y la longitud de los cromosomas y generar dos cromosomas nuevos con la información cruzada de los dos cromosomas originales. En los pasos 5 y 6 se evalúa si la nueva

población ya tiene la cantidad acordada de cromosomas, si no es así el algoritmo se repite desde el paso 3 en caso contrario el algoritmo arranca de nuevo desde el paso 2.

3.2.3 ¿Por qué funcionan los Algoritmo Genéticos?

Tal vez, valor del trabajo desarrollado por Holland no está concentrado mayormente en su algoritmo genético sino en la hipótesis que planteó para justificar por qué funciona esta técnica y en la demostración matemática que logró para sustentarla. A continuación se presentan algunos puntos claves de la hipótesis.

- Todo problema tiene un espacio de soluciones posible. Si una solución del problema se puede representar por una cadena binaria, el tamaño del espacio de búsqueda es 2^L donde L es el la longitud de la cadena [WHI93].
- El espacio de soluciones se puede dividir en grupos de familias de soluciones dependiendo de los elementos fijos y variables que se establezcan. Por ejemplo, si las soluciones de un problema se pueden codificar por una cadena de cuatro bits y si se deja fijo un bit, el espacio de soluciones se divide en dos partes. Si se dejan fijos dos bits, el espacio de búsqueda se divide en cuatro partes. Ver Figura 5



- Una solución particular es representante de varias familias de soluciones simultáneamente. Volviendo al ejemplo anterior, la solución particular 1011 es

representante de las familias 1^{***} , $*0^{**}$, $**1^*$, $***1$, 10^{**} , $1*1^*$, $1^{**}1$, $*01^*$, $*0*1$, $**11$, 101^* , $10*1$ y $*011$.

- Una solución particular evalúa implícitamente a todas las familias de soluciones que representa.
- Al seleccionar aleatoriamente soluciones particulares, se distribuye el muestreo de las familias de soluciones de manera uniforme.
- A medida que se descartan las soluciones particulares con menor adaptación, y se permite la reproducción de aquellas con buena adaptación, se tiende a favorecer ciertas familias de soluciones.
- Al aplicar operadores genéticos sobre soluciones particulares se puede llegar a soluciones que antes no habían sido consideradas.
- Luego de seleccionar, descartar, y modificar conjuntos de soluciones muchas veces, estas tienden a volverse homogéneas.

La demostración completa que Holland desarrolló se encuentra en [HOL75], sin embargo, dado que este trabajo es difícil de conseguir, se pueden encontrar buenos acercamientos en [WHI93] y [GOL89].

3.2.4 Operadores genéticos de cruce y mutación

Los operadores genéticos son el mecanismo a través del cual los algoritmos genéticos logran generar soluciones mejor adaptadas de una generación a otra. Existen diferentes implementaciones para los operadores de cruce y mutación. A continuación se describen brevemente los más utilizados [KOO04]:

- Técnicas de cruce: todas las técnicas de cruce buscan crear nuevos cromosomas que sean el resultado de mezclar los elementos de los cromosomas originales.
 - Cruce de un punto: es la implementación más utilizada. Consiste en seleccionar un punto al azar e intercambiar la información genética entre los dos cromosomas.

- Cruce de dos puntos: funciona de manera análoga al cruce de un punto, pero como su nombre lo indica, en lugar de seleccionar un punto de cruce, selecciona dos.
- Cruce uniforme: en esta técnica cada uno de los genes de los cromosomas resultantes se selecciona de manera autónoma.
- Cruce aritmético-lógico: consiste en obtener nuevos individuos al aplicar operaciones aritméticas o lógicas sobre los cromosomas originales.
- Técnicas de mutación: buscan generar variaciones aleatorias sobre elementos particulares.
 - Mutación unibit: es la utilizada tradicionalmente y consiste en modificar aleatoriamente uno de los elementos que constituyen un gen dentro del cromosoma seleccionado.
 - Mutación por genes: funciona de manera análoga a la mutación unibit pero en cambio de modificar un elemento dentro de un gen, modifican el gen completamente.
 - Mutaciones epistáticas: consisten en intercambiar dos elementos dentro de un gen o dos genes dentro de un cromosoma.
 - Mutación multibit: este método tiene selecciona varios elementos dentro de los genes para ser modificados mediante un cálculo de probabilidad.

3.2.5 Algoritmos genéticos en problemas de permutación

En los problemas de permutación no solo son importantes los genes que conformen cada uno de los cromosomas sino es importante la posición de estos dentro del cromosoma. El problema con el que se esta trabajando pertenece a este género de problemas y por tanto es interesante analizar algunas características especiales que usualmente es necesario implementar para resolverlos utilizando algoritmos genéticos [HAU04]:

- Resulta mejor utilizar codificaciones en las que cada uno de los genes sea compacto, es decir que no esté compuesto por piezas mas pequeñas.

- Los mecanismos tradicionales de cruce no funcionan porque muy fácilmente producen individuos que no son viables. Las técnicas de cruce más utilizadas para los problemas de permutación son el cruce ordenado y el cruce cíclico.
- El mecanismo de mutación que se utiliza usualmente es intercambiar dos genes en el cromosoma (mutación epistática).

Para analizar de menor manera la forma como es posible implementar los algoritmos genéticos en problemas de permutación es bueno mencionar un problema tradicional conocido como el problema del agente viajero o TSP (por sus iniciales en inglés).

El TSP es un problema de optimización clásico que consiste en encontrar el recorrido cíclico (tour) más corto para que un agente de viajes pueda visitar N ciudades una sola vez y terminar nuevamente en la ciudad inicial. El problema supone que existen recorridos directos para viajar entre todas las ciudades. El problema, en su forma más simple, utiliza como función de costo la sumatoria de la distancia euclidiana existente entre cada uno de los puntos de la ruta definida:

$$\text{costo} = \sum_{n=0}^N \sqrt{(x_n - x_{n+1})^2 + (y_n - y_{n+1})^2}$$

Donde (x_n, y_n) son las coordenadas de la n -ésima ciudad visitada.

La complejidad de este problema está dada por $(n-1)!/2$ donde n representa el número de ciudades a ser visitadas [LAR98].

En [LAR98] se presenta un resumen de diferentes tipos de implementaciones que se han desarrollado para encontrar soluciones a TSP a través de los algoritmos genéticos. A continuación se describen brevemente unas de las implementaciones mencionadas en [LAR98]:

Representación binaria

En esta representación cada ciudad se codifica como una cadena binaria de $\log_2 n$ bits y un cromosoma o solución del problema como una cadena de $n(\log_2 n)$ bits. Si por ejemplo

se considera un TSP de 6 ciudades, la primera ciudad se codificaría como 000, la segunda como 001 y así sucesivamente. Con esta representación el tour 1-2-3-4-5-6 se representaría por 000 001 010 011 100 101.

Esta representación tiene fuertes problemas para la implementación de los operadores genéticos porque fácilmente se pueden obtener soluciones no factibles.

Para solucionar el problema de los individuos no factibles es posible penalizar estos en la función de evaluación de manera que no pasen de una generación a otra o también se puede implementar una operación de reconstrucción e invocarla luego de aplicar los operadores genéticos.

Representación Ordinal

En esta representación cada ciudad se codifica como un número entero. Para codificar un cromosoma se utilizan dos listas: la primera es una lista ordenada de todas las ciudades existentes y la otra (que es el cromosoma en sí) indica la posición de la ciudad que se debe visitar. Una vez una ciudad se visita, se elimina de la primera lista. Para ejemplificar esto se puede retomar el TSP de 6 ciudades. En este, la lista de ciudades existentes sería $L=(1\ 2\ 3\ 4\ 5\ 6)$. Para obtener la representación del tour 3-5-2-4-6-1 se procedería de la siguiente manera:

Como la primera ciudad es la tres y esta es la tercera en la lista, se incluye un tres en la lista del tour $T=(3)$. Con esto la lista L se reduce a $L=(1\ 2\ 4\ 5\ 6)$.

La segunda ciudad en visitar es la cinco, la cual ocupa el cuarto lugar en la lista, entonces se incluye un 4 en tour $T=(3\ 4)$ y la lista L se reduce a $L=(1\ 2\ 4\ 6)$.

Siguiendo este mismo procedimiento se obtiene que el tour es $T=(3\ 4\ 2\ 2\ 2\ 1)$

Esta representación tiene la gran ventaja de que permite implementar el operador de cruce tradicional descrito por Holland. No obstante el operador de mutación requiere una implementación cuidadosa ya que el i -ésimo elemento del tour solo puede ser cambiado por un número entre 1 y $n-i + 1$.

4 ESTRATEGIA DE SOLUCIÓN

Como método de solución al problema planteado se quiso evaluar una nueva estrategia que consiste en implementar un algoritmo genético que genere y evalúe diferentes distribuciones de productos dentro del centro de distribución. Para evaluar cada una de las distribuciones se utiliza una lista de grupos de productos que usualmente se venden juntos. Esta lista se puede obtener aplicando la técnica de minería de datos conocida como Market Basket Analysis sobre la información histórica de pedidos.

En este capítulo se describirá la forma como se puede implementar la estrategia y se determinarán aquellos aspectos que es mejor definir sobre cada centro de distribución particular. Como primer punto se especifican las tareas relacionadas con el análisis histórico de información y luego se describen los aspectos a considerar en la aplicación de los algoritmos genéticos al problema.

4.1 Análisis de la información histórica

Este análisis pretende determinar conjuntos de productos que usualmente se compran juntos; esto se puede lograr analizando la información histórica de pedidos con la técnica de Market Basket Analysis. Esta técnica usualmente requiere las siguientes tareas:

- Hacer un proceso ETL con la información requerida, que para este caso es simplemente una lista de los pedidos y los productos que los componen.
- Experimentar con diferentes parámetros de entrada, para determinar aquellos con los que se obtienen mejores resultados. Usualmente en el Market Basket Analysis estos parámetros son el soporte mínimo, la confianza mínima y el tamaño máximo del conjunto.

Como resultado de este análisis se produce un conjunto de ítems frecuentes, cada uno con una medida de soporte que indica la cantidad de veces que los productos de un conjunto se vendieron juntos respecto del total de pedidos considerados.

4.2 Aplicación de los Algoritmos Genéticos al Problema

Los algoritmos genéticos son idóneos para ser aplicados al problema porque además de producir diferentes soluciones, permiten premiar aquellas que mantienen juntos los grupos de ítems frecuentes e ir variándolas hasta encontrar la más satisfactoria.

Para aplicar la técnica de algoritmos genéticos al problema particular es necesario, como mínimo, definir los siguientes aspectos:

- El modelo de codificación, que es la forma como se va a representar una posible solución al problema.
- La función que se utiliza para evaluar cada una de las soluciones encontradas (función de evaluación). Que en este caso es la encargada de hacer uso de la información generada por el Market Basket Analysis.
- El mecanismo a través del cual se seleccionan y reproducen las mejores soluciones (mecanismo de selección).
- Los mecanismos que se utilizan para modificar las soluciones para encontrar unas nuevas (operadores genéticos).
- La forma como se determina que ya se llegó a una solución aceptable (criterio de parada).

4.2.1 Modelo de codificación

No hay recomendaciones claras (o por lo menos no se encontraron luego de una revisión bibliográfica exhaustiva del tema) de cómo codificar individuos para problemas particulares. La literatura siempre reconoce que este es un aspecto que depende mucho del problema, que es fundamental diseñar antes de cualquier otro y que debe ser muy bien revisado porque de él dependerán los demás aspectos.

Para el problema con que se está trabajado se plantearon tres tipos diferentes de codificación:

1. Construir una tabla binaria en la que las filas fuesen cada uno de los productos y las columnas cada una de las localizaciones de la bodega. Un valor 1 en el

elemento (a,b) de la tabla representaría la presencia del producto a en la localización b. Este método de codificación se ilustra en la figura 6.

	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	U11	U12	U13	U14	U15	U16	U17	U18	U19	U20
P1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
P3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P4	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
P5	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
P6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
P7	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
P8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
P9	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
P10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
P11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
P12	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
P13	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
P14	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P15	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
P16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
P17	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
P19	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P20	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 6. Codificación binaria del centro de distribución

- Utilizar un arreglo 3-dimensional en el que cada uno de sus elementos represente una localización y en el que se almacene un número entero que indique el tipo de producto almacenado. En la figura 7 se ilustra este método de codificación. Las casillas blancas representan los lugares por los que es posible transitar.

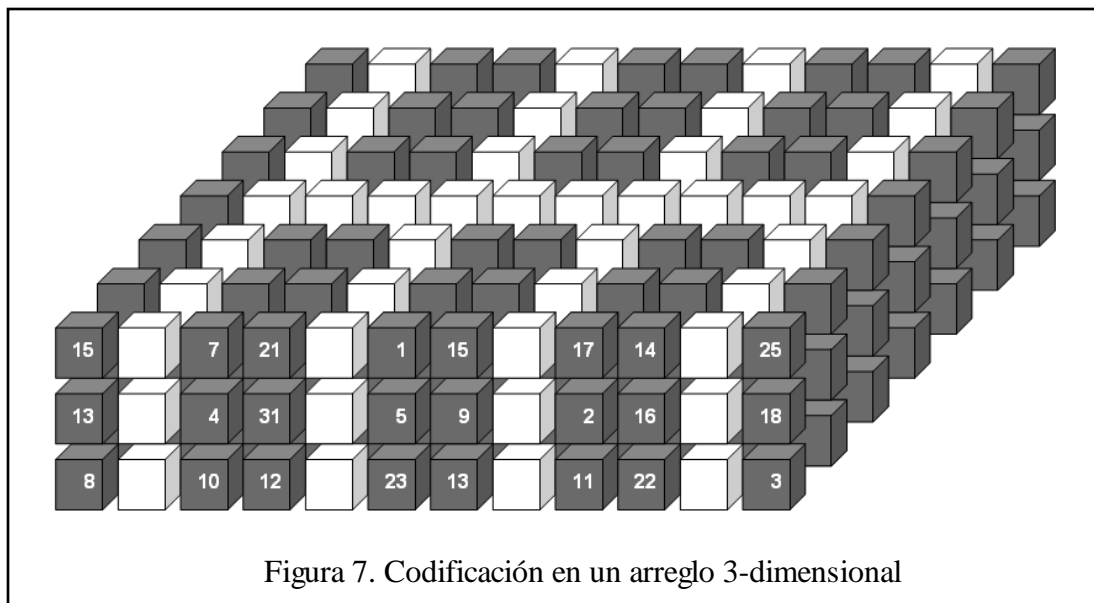
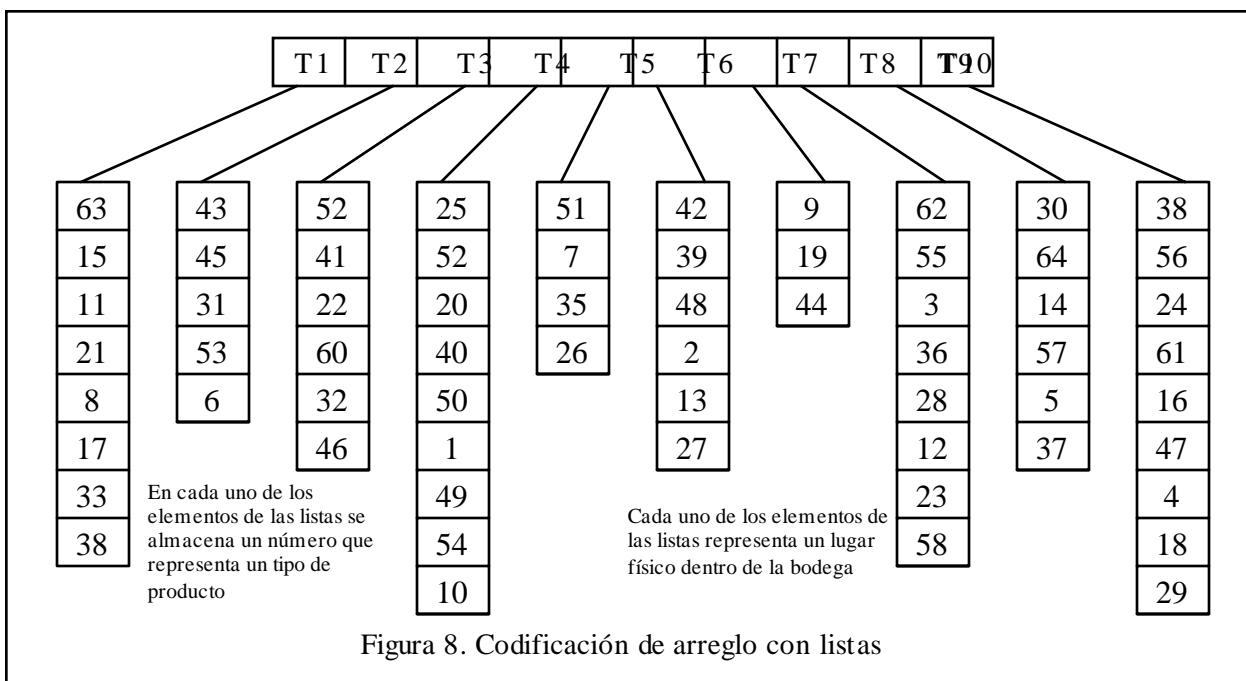


Figura 7. Codificación en un arreglo 3-dimensional

3. Utilizar un arreglo unidimensional en el que cada uno de sus elementos representa un tipo de localización dentro de la bodega y al cual está asociada una lista de tamaño variable en la que cada uno de sus elementos representa una localización particular dentro de la bodega y almacena un número entero único que representa un producto particular. En la figura 8 se ilustra este método de codificación; el arreglo unidimensional (que está representado por el rectángulo ubicado en la parte superior) tiene tantos elementos como tipos de localizaciones. A cada uno de los tipos de localización, está asociada la lista de localizaciones que de este tipo existen en el centro de distribución y dentro de cada una de las listas se almacenan los productos.



A continuación se presenta una tabla en la que se resumen las ventajas y desventajas de los tres modelos de codificación considerados.

	Codificación binaria	Codificación 3-dimensional	Codificación de arreglo con listas
Ventajas	-Por ser binaria, facilita la aplicación de mecanismos tradicionales de	-Es fácil de interpretar y es compacta. -Permite almacenar la información de los espacios	-Es compacta. -Permite distinguir claramente los diferentes tipos de localizaciones.

	algoritmos genéticos.	disponibles para transitar.	
Desventajas	-Por si sola no permite distinguir los diferentes tipos de localizaciones. -Al aplicar un operador tradicional de cruce sobre ella, es posible obtener soluciones no factibles.	- Al aplicar un operador tradicional de cruce sobre ella, es posible obtener soluciones no factibles. -Cuando los tamaños de las localizaciones son muy disímiles, se hace difícil de implementar.	-No permite almacenar la información de los espacios disponibles para transitar. -No es fácil de implementar.

Tabla 1. Ventajas y desventajas de los modelos de codificación considerados. Entre los tres modelos de codificación considerados, se decidió trabajar con el modelo de arreglo con listas de tamaño variable porque se adapta muy bien a las características del problema y en especial porque permite diferenciar claramente los distintos tipos de localizaciones.

4.2.2 Función de evaluación

La función de evaluación tiene la tarea de valorar cada una de las posibles soluciones del problema. Como se mencionó al comienzo de este capítulo, una parte de la estrategia de solución consiste en utilizar la información generada por la aplicación del Market Basket Analysis sobre la información histórica de pedidos del centro de distribución. Por tanto, la tarea de la función es tomar cada uno de los grupos de ítems frecuentes, determinar que tan cercanos físicamente se encuentran los productos que los conforman y dar una ponderación final de “que tan bien” o no una distribución de productos en la bodega “cumple” con las recomendaciones sugeridas por los grupos de ítems frecuentes.

Los parámetros de entrada de la función son: una solución al problema (codificada en la forma que se explicó anteriormente) y el conjunto de ítems frecuentes.

Como parte de la estrategia de solución, se decidió que la función trabaje de la siguiente forma:

- Para cada uno de los grupos de ítems frecuentes, determina las localizaciones que cada uno de los productos ocupa dentro del centro de distribución con base en la solución proporcionada.
- Ordena las localizaciones de la más cercana a la más lejana con el fin de establecer la ruta que se seguiría para la recolección de los productos que conforman cada grupo.
- Calcula la distancia promedio entre los productos. Para esto se calcula la distancia entre cada uno de los puntos de la ruta y se divide entre la cantidad de puntos menos uno. La función que calcula la distancia entre dos puntos del centro de distribución es uno de los aspectos que es mejor definir para de cada centro de distribución particular porque depende de la forma como esté construido el centro y de la tecnología que se utilice en la recolección de pedidos.
- Divide las distancias promedio de cada grupo de ítems frecuentes entre su soporte, de manera que aquellos grupos con un soporte mayor tienen una participación más grande en el cálculo de la función.
- Totaliza las distancias promedio de todos los grupos de ítems frecuentes y la asigna como ponderación final.

4.2.3 Mecanismo de selección

Como mecanismo de selección se decidió utilizar el desarrollado por Holland [WHI93] porque en muchos de los trabajos de investigación de algoritmos genéticos, éste ha demostrado ser eficaz en la creación de generaciones de individuos mejor adaptados. El mecanismo procede de la siguiente forma:

- Para cada una de las soluciones se calcula la función de adaptación $\frac{f_i}{\bar{f}}$ donde f_i es el valor obtenido por la función de evaluación para la i -ésima solución y \bar{f} es el

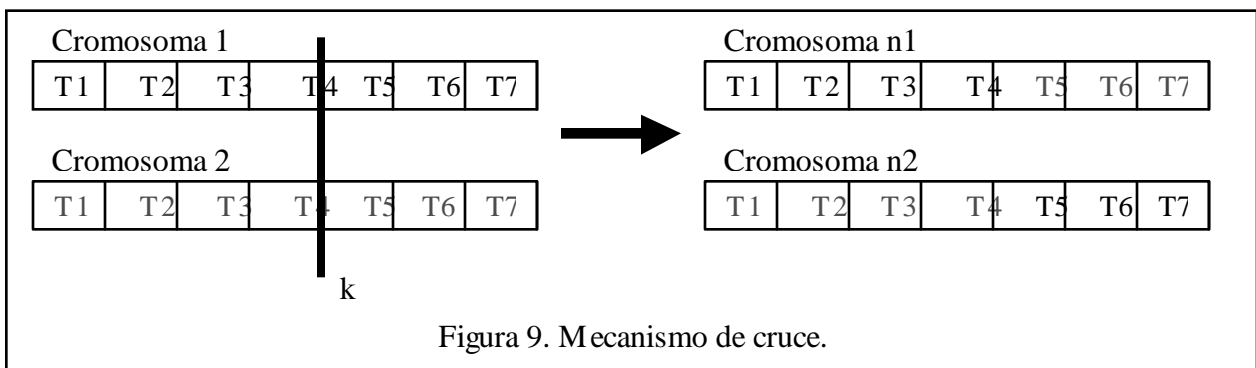
valor promedio $\frac{\sum_{i=1}^n f_i}{n}$.

- Si por ejemplo para una solución específica la función de adaptación fue 2.64, se pasarán 2 copias de la solución a una generación intermedia y generará un número aleatorio entre 0 y 1, si el número es menor que la fracción (0.64 en este caso) se generará una copia adicional de la solución en la generación intermedia.

4.2.4 Operadores genéticos

Con base en el método de codificación seleccionado, los operadores genéticos de cruce y mutación se plantearon de la siguiente manera:

- **Cruce:** se planteó utilizar el operador de cruce de 1 punto desarrollado por Holland [WHI93], que funciona de la siguiente manera: se seleccionan dos cromosomas y se genera un número aleatorio k entre 1 y $L-1$, luego se crean dos cromosomas nuevos que son el resultado de intercambiar los elementos de los cromosomas originales en el punto k . Ver Figura 9. Uno de los parámetros de entrada para el algoritmo genético es un porcentaje de cruce que indica sobre qué proporción de la población se aplica el operador de cruce. En el algoritmo genético planteado, cada vez que un individuo va a pasar de una generación a otra se genera un número aleatorio entre 0 y 1, si tal número es menor que el porcentaje de cruce establecido, se aplica el operador sobre tal individuo. El segundo individuo se selecciona aleatoriamente de entre la población.



El operador de cruce de 1 punto se seleccionó de entre un grupo de otras implementaciones en las que se encuentra el cruce de dos puntos, el cruce uniforme y el cruce multipunto, porque en el estudio empírico descrito en [BAI03], se demostró que entre más complejas sean las operaciones de cruce,

peores son los resultados obtenidos. La razón de esto parece estar justificada en que si se aumenta la cantidad de información que se intercambia entre los cromosomas, también se aumenta la probabilidad de que las características positivas de estos no prevalezcan.

- Mutación: en el método de codificación planteado, este operador puede trabajar a nivel de las listas e implementarse como una mutación epistática de intercambio [KOO04], la cual toma dos elementos del cromosoma elegidos en forma aleatoria y los intercambia. Ver figura 10. El porcentaje de cruce es otro de los parámetros de entrada del algoritmo genético y actúa de manera similar al porcentaje de cruce, antes de que un individuo pase de una generación a otra se genera un número aleatorio que indica si se aplica o no el operador de mutación sobre él.

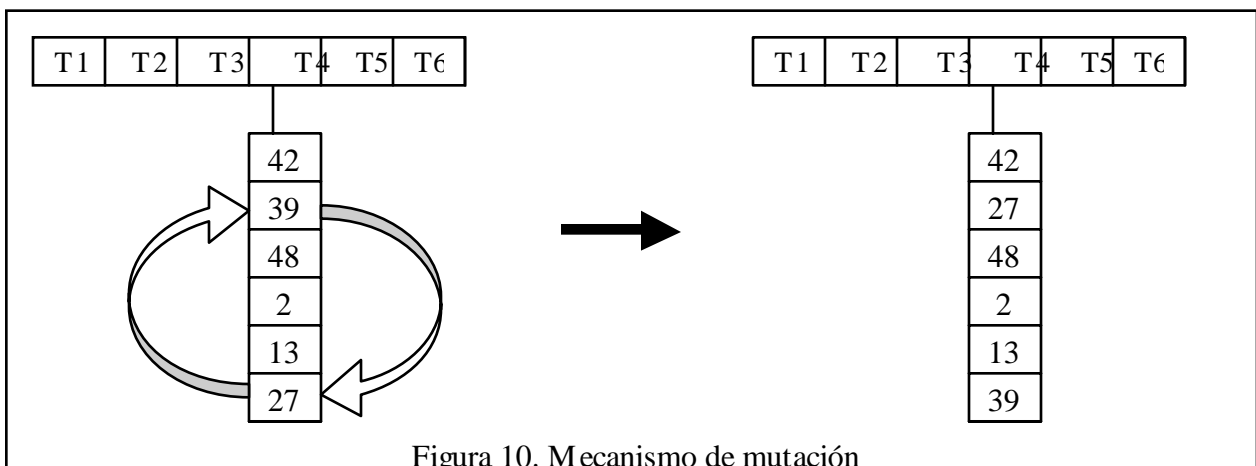


Figura 10. Mecanismo de mutación

Este tipo de mecanismo de mutación, se seleccionó en lugar de otros como la mutación unibit, la mutación por genes o la mutación multibit, porque permite que fácilmente se mantenga la viabilidad de las soluciones. Si se hubiese seleccionado por ejemplo la mutación unibit, que es la que tradicionalmente se utiliza, hubiese sido necesario que luego de modificar aleatoriamente el elemento seleccionado, se tuviesen que modificar algunos de los demás elementos para lograr que el cromosoma resultante siguiese siendo una solución viable al problema.

4.2.5 Criterio de parada

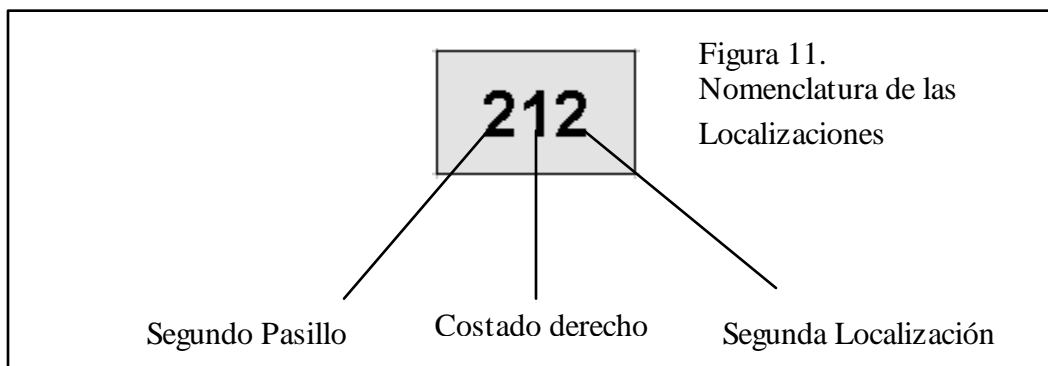
Como criterio de parada se decidió utilizar la convergencia por aptitud [KOO04], porque es un mecanismo que garantiza la estabilización del algoritmo. En este criterio se evalúa la variación de la aptitud promedio de una generación a otra y cuando esta es menor a un umbral establecido como parámetro, se da por terminada la ejecución del algoritmo y se selecciona la solución que tenga el valor de evaluación más bajo.

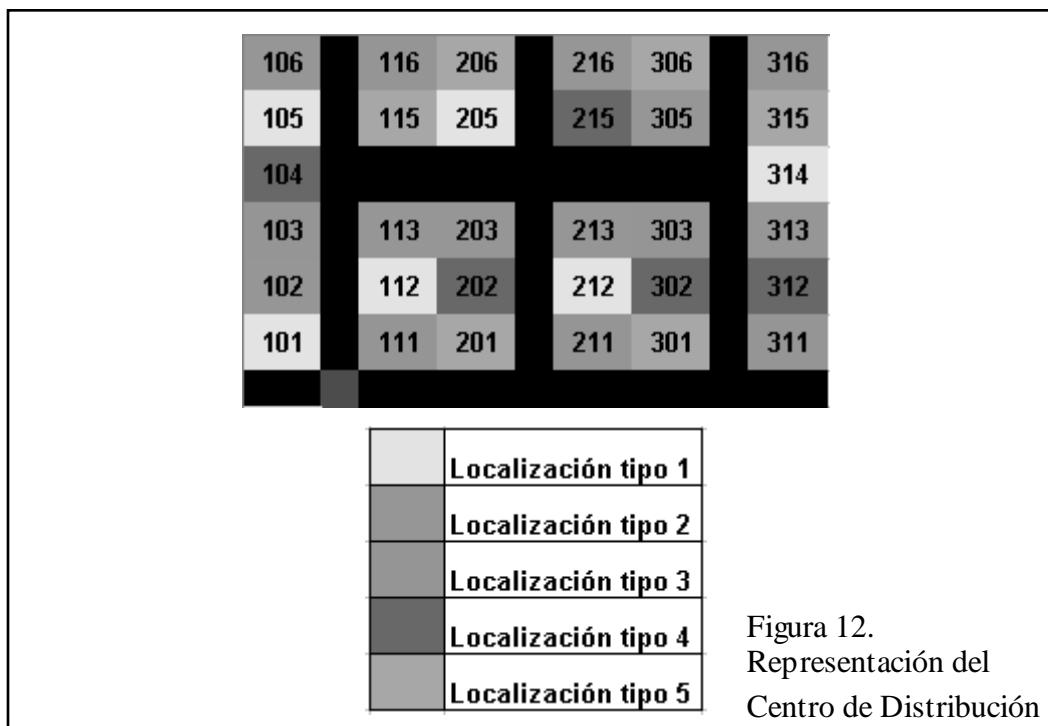
5 PROTOTIPO DE EVALUACIÓN Y PRUEBAS

Para aclarar la forma como la estrategia propuesta se utiliza en la búsqueda de una solución al problema planteado, se decidió hacer un prototipo en el que se apliquen cada uno de los aspectos a tener en cuenta y se puedan evaluar los resultados obtenidos. El capítulo comienza con la definición del centro de distribución y los grupos de productos frecuentes con que se trabajó, luego se describen los elementos del algoritmo genético aplicados al centro de distribución planteado, más adelante se describen una serie de experimentos realizados con el fin de determinar el efecto de los diferentes parámetros sobre el comportamiento del algoritmo y por último se describe la forma como se implementó una herramienta de simulación que permitió comparar el resultado logrado, contra otro obtenido a través de una estrategia tradicional.

5.1 Definición del centro de distribución y los grupos de ítems frecuentes

Para el prototipo se decidió representar un pequeño centro de distribución con 5 tipos diferentes de localizaciones, 32 localizaciones físicas, 32 grupos de productos, 3 pasillos horizontales y dos pasillos verticales. Cada una de las localizaciones se etiquetó con un número de tres dígitos (el primero indicando el pasillo, el segundo para diferenciar si se trata del costado derecho o izquierdo y el tercero como un consecutivo de localización por pasillo y costado. Ver figura 11). En la figura 12 se puede observar gráficamente el centro de distribución planteado; el recuadro en la parte inferior izquierda representa el punto desde el cual los operarios inician la recolección de los pedidos.





Con un centro de distribución tan pequeño parecería fácil analizar todas las posibles configuraciones de productos, pero esto no es tan cierto ya que para este caso el número de posibilidades es 16.253.308.108.800.000 (1.6×10^{16}). En la figura 13 se muestran los detalles de cómo se efectuó el cálculo del número de posibles configuraciones.

	Tipo	No. Localizaciones	No. Distribuciones x Tipo
	1	6	720
	2	9	362880
	3	6	720
	4	5	120
	5	6	720

No. Total de Distribuciones = 720 x 362880 x 720 x 120 x 720 = 16.253.308.108.800.000

Figura 13. Número de posibles ubicaciones de productos en el Centro de Distribución

Dado que para este prototipo no se contaba con información histórica de pedidos, se tomó como punto de partida una configuración factible de productos dentro del centro de distribución Ver figura 14.

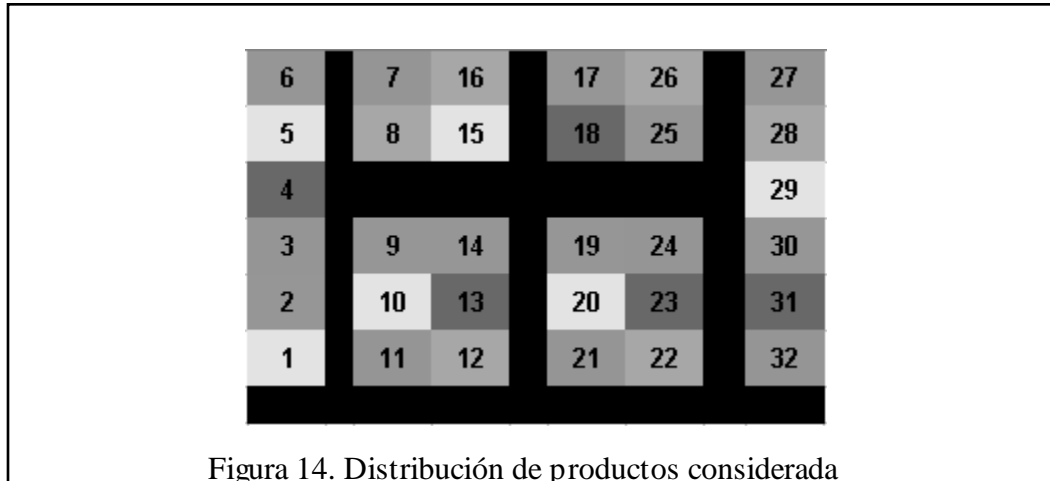


Figura 14. Distribución de productos considerada

Con base en esta distribución, se formaron grupos de productos que estuviesen físicamente cerca, y se les asignó valores de soporte aleatoriamente. Todo esto con el fin de simular los resultados que se obtienen al aplicar el Market Basket Analysis sobre información histórica de pedidos. En la siguiente tabla se observan los conjuntos de productos que se utilizaron:

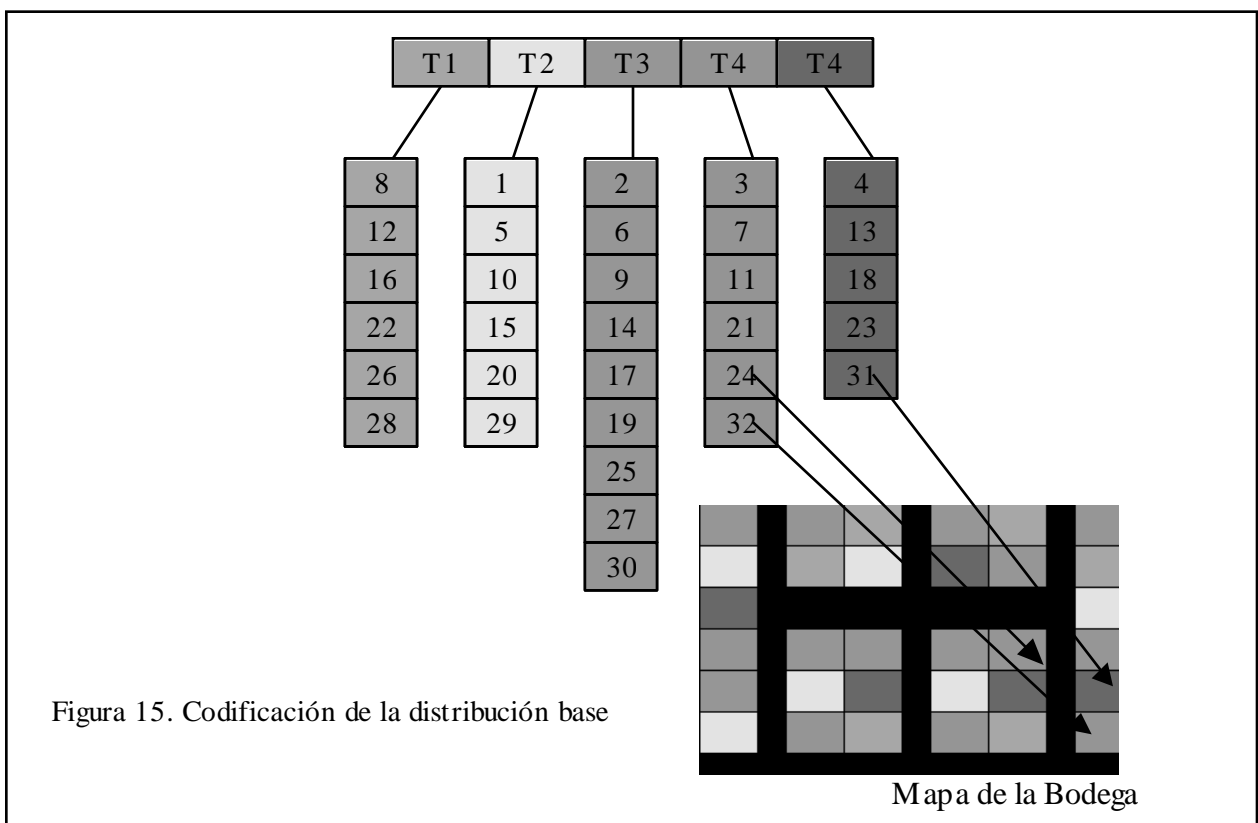
Conjunto	Productos	Soporte
1	15,18	0.8
2	1,11,10	0.6
3	16,18	0.3
4	25,27,29	0.9
5	22,31	0.7
6	6,8,4	0.6
7	2,3,9	0.7
8	4,5,8	0.5
9	6,7,8	0.4
10	12,13,20	0.6
11	14,19,20,21	0.7
12	15,17,18	0.4
13	22,23,24	0.5
14	22,32,31	0.7
15	24,28,29	0.8
16	25,26,27	0.6
17	8,9,14	0.6
18	19,24,25	0.7

5.2 Implementación del Algoritmo Genético al prototipo

En esta sección se hacen algunas ilustraciones de cómo cada uno de los aspectos planteados en el algoritmo genético de la estrategia se aplicó en el prototipo y se especifica la función de distancia entre dos puntos que se utilizó para este caso particular.

5.2.1 Modelo de codificación

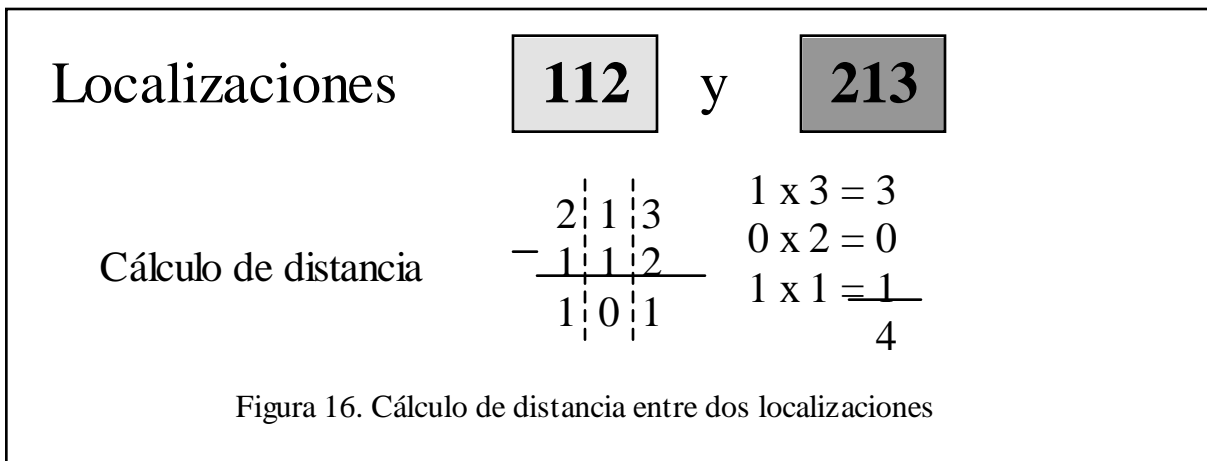
En la figura 15 se ilustra como el modelo de codificación que se planteó en la estrategia de solución se aplica a la distribución de productos de la figura 14. Cada una de las listas se ordenó de la localización más cercana a la más lejana.



5.2.2 Función de evaluación

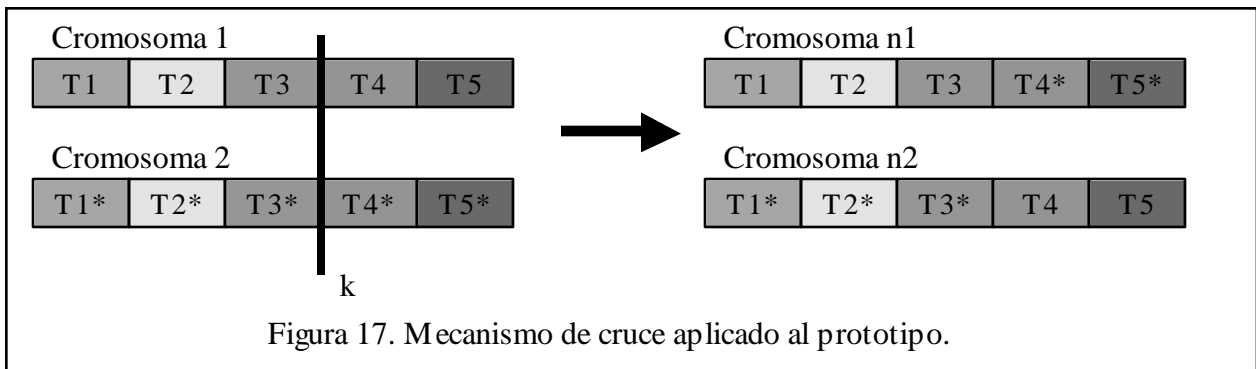
La función de evaluación planteada en la descripción de la estrategia requiere de una función de distancia que determine que tan cerca se encuentran dos localizaciones dentro del centro de distribución. Para el centro planteado en el prototipo la función de distancia se implementó de la siguiente manera:

- Se toman las identificaciones de las dos localizaciones y se determina si para ir de una localización a la otra es necesario cambiar de pasillo, de costado o simplemente desplazarse a través del mismo costado.
- Los cambios de pasillo se penalizan multiplicando por 3 y los cambios de costado multiplicando por 2, siempre y cuando no se produzca un cambio de pasillo. En la figura 16 se muestra el cálculo de la distancia promedio entre dos localizaciones.

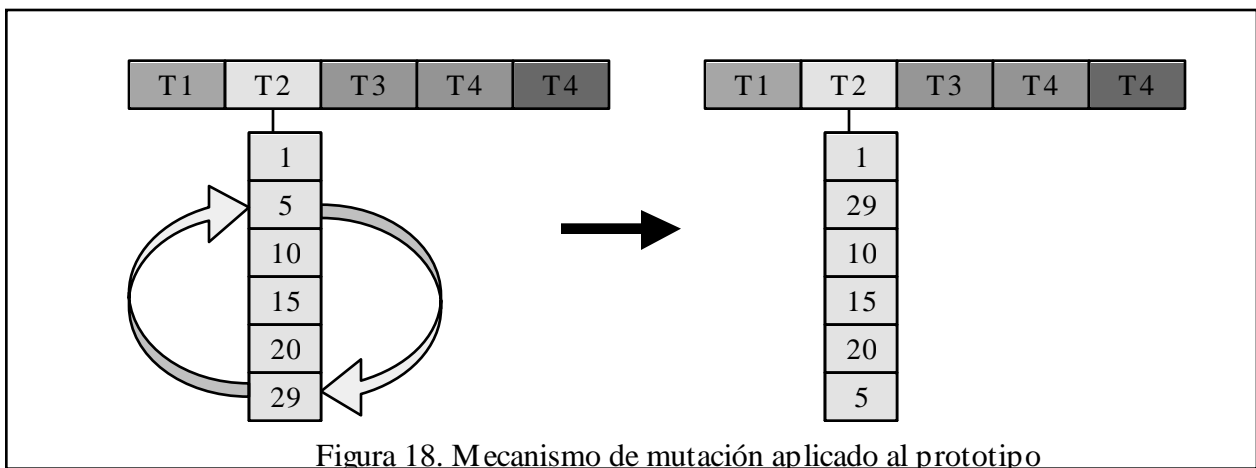


5.2.3 Operadores genéticos

En la figura 17 se ilustra la forma como el operador de cruce se aplica sobre dos soluciones codificadas (cromosomas).



En la figura 18 se observa como se aplica el operador de mutación sobre una solución codificada del problema.



5.2.4 Criterio de parada

Aunque el criterio de parada se seleccionó para el algoritmo genético es la convergencia por aptitud en algunos de los experimentos que se realizaron con el prototipo se utilizó como criterio de parada un número de iteraciones establecido como parámetro.

5.3 Programación del Algoritmo Genético

A pesar de que existen múltiples librerías y paquetes de recursos que buscan incrementar la velocidad de programación de los algoritmos genéticos, se decidió hacer un desarrollo

propio principalmente porque la implementación del modelamiento de los individuos que se diseñó resultaba difícil de lograr y generaba demasiados cambios.

Para el desarrollo del algoritmo genético se utilizó el enfoque de desarrollo orientado por objetos.

Luego de analizar las implicaciones de los aspectos seleccionados en el algoritmo genético de la estrategia, se identificaron las siguientes clases:

- **AlgoritmoGenetico:** es la clase encargada de invocar y/o implementar las funcionalidades propias del algoritmo genético, tal como la generación de la población inicial, la selección de los individuos la generación de las variables aleatorias y la finalización de la ejecución entre otras.
- **Configuracion:** está encargada de conocer los diferentes tipos de localizaciones, las localizaciones que pertenecen a cada uno de los tipos y los productos que se pueden alojar en cada uno de estos.
- **Poblacion:** es el grupo total de individuos que existe en una generación.
- **Cromosoma:** es un individuo y por tanto representa una solución al problema. Está compuesto por un conjunto de genes.
- **Gen:** representa cada una de las listas de tipos de productos. Está compuesto por un conjunto de alelos.
- **Alelo:** representa la ubicación de un producto particular en una localización específica.
- **ItemFrecuente:** representa un grupo de elementos que fueron determinados como frecuentes a través del análisis histórico de la información de pedidos del centro de distribución. Cada ItemFrecuente tiene asociado un valor de soporte.
- **GrupoItems:** es el conjunto de todos los ítems frecuentes encontrados.
- **FuncionEvaluacion:** agrupa las funcionalidades necesarias para determinar que tan buena o no es una solución particular.
- **OperadorGenetico:** agrupa las funcionalidades que permiten efectuar las operaciones de cruce y mutación sobre los cromosomas.

- **CriterioParada:** es la clase que determina cuando se detiene o continúa el algoritmo genético.

En el Anexo 1 se encuentra un diagrama de clases en el que se puede determinar de mejor manera la forma en que las clases fueron diseñadas e interactúan para lograr el algoritmo genético.

En el Anexo 2 se presenta el diagrama de secuencia del principal procedimiento del algoritmo genético.

5.4 Experimentación y análisis de resultados

Los resultados que se obtienen de un algoritmo genético dependen en gran manera de las decisiones que se tomen para su desarrollo, pero también dependen de los parámetros de entrada que se utilicen para su ejecución. Los parámetros que comúnmente se modifican en un algoritmo genético son el porcentaje de mutación, el porcentaje de cruce y el tamaño de la población. A continuación se muestran algunos experimentos que se hicieron para determinar el efecto de cada uno de los parámetros.

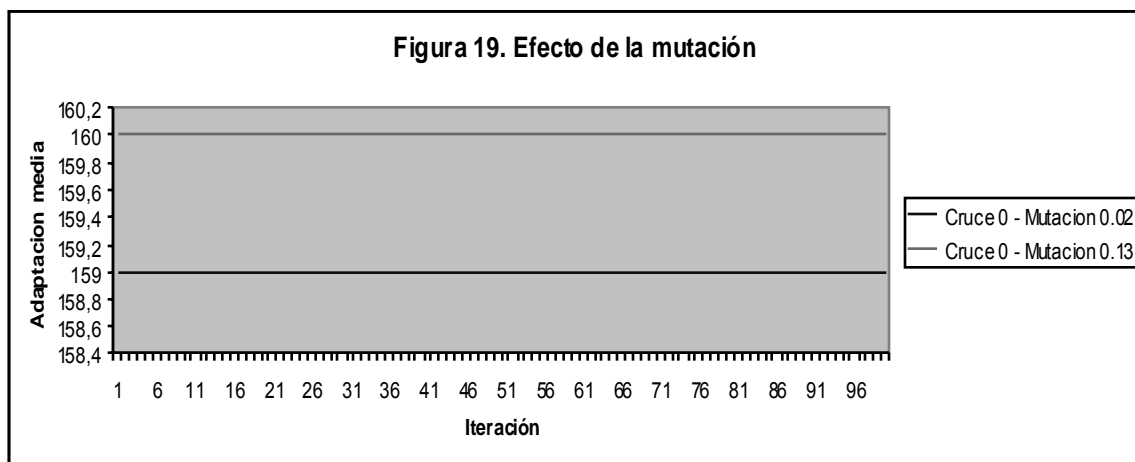
5.4.1 Efecto de la mutación

De acuerdo con la teoría y los diferentes trabajos desarrollados en torno a los algoritmos genéticos, la operación de mutación tienen la misma razón de ser en los algoritmos que en la naturaleza: generar nuevos individuos cada cierto tiempo, los cuales podrán ser mejor o peor adaptados que los demás.

De acuerdo con [WHI93] un porcentaje pequeño de mutación (del orden de 1%) logra evitar que el algoritmo genético caiga en mínimos o máximos locales y un porcentaje elevado de mutación (del orden del 10% o superior) genera inestabilidad en el algoritmo genético.

En este primer experimento se quiso evaluar el efecto que tiene el operador de mutación por si solo. Para esto se programó el algoritmo para ejecutarse durante 100 generaciones con 100 individuos por generación, un porcentaje de cruce de 0% y dos distintos porcentajes de mutación (2% y 13%).

La figura 19 se puede observar el comportamiento de la adaptación media de la población durante cada una de las generaciones, la cual en resumen, se mantuvo inmodificable a través del experimento. Estos resultados indican que el operador de mutación por si solo no genera ninguna diferencia para lograr una mejor adaptación de la población. Esto en otras palabras quiere decir que la mutación por si sola no permite encontrar mejores soluciones para el problema.

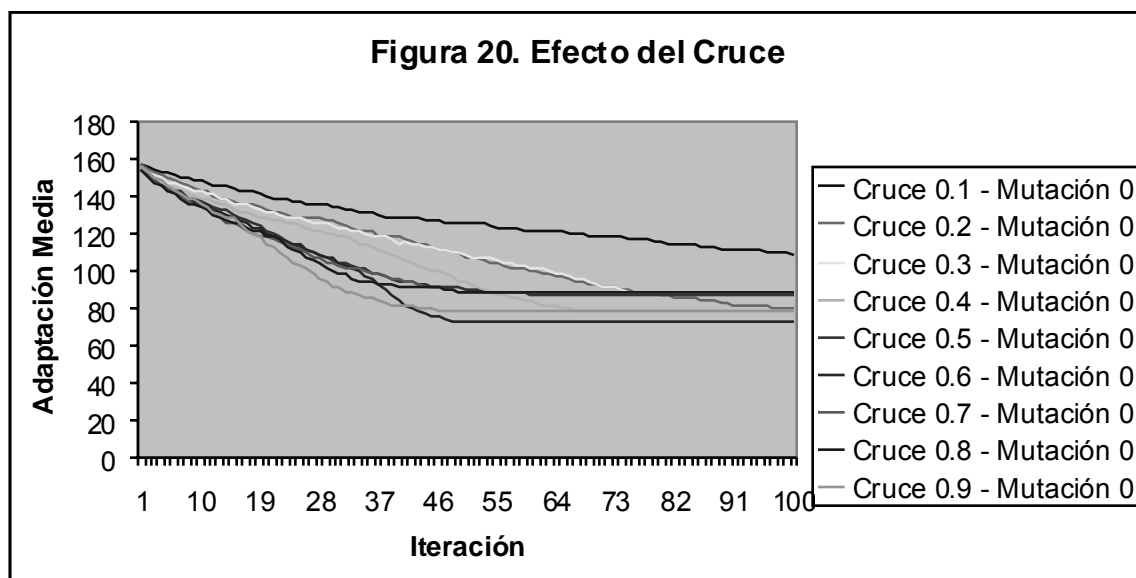


5.4.2 Efecto del cruce

El operador de cruce es el encargado de generar nuevas soluciones o individuos a partir de dos elementos ya existentes en la población anterior. Según [WHI93] el porcentaje de cruce que se establezca sobre un algoritmo genético afectará directamente la velocidad de convergencia de este.

Para este experimento se quiso evaluar el efecto del operador de cruce por si solo. Para ello se ejecutó el algoritmo genético varias veces con porcentajes de cruce entre 10% y 90%, porcentaje de mutación de 0% y un tamaño de población de 100 individuos.

En la figura 20 se presenta una gráfica del comportamiento de la adaptación media de la población a través de cada una de las 100 generaciones para los diferentes porcentajes de cruce considerados. Los resultados obtenidos ratifican que el porcentaje de cruce es directamente proporcional a la velocidad de convergencia del algoritmo, por ejemplo para el porcentaje de cruce de 10%, las 100 iteraciones no fueron suficientes para que el algoritmo genético se estabilizara, en cambio con el porcentaje de cruce de 90% tan solo fueron necesarias 49 iteraciones para lograr la estabilización. Aunque en el experimento no resultó evidente que para porcentajes de cruce muy altos se encontrasen mínimos locales, si se nota una diferencia entre los valores de adaptación media a los que se estabilizó el algoritmo genético con porcentajes de cruce de 70 y 80% y los que se lograron con las ejecuciones a 20, 30 y 40%.



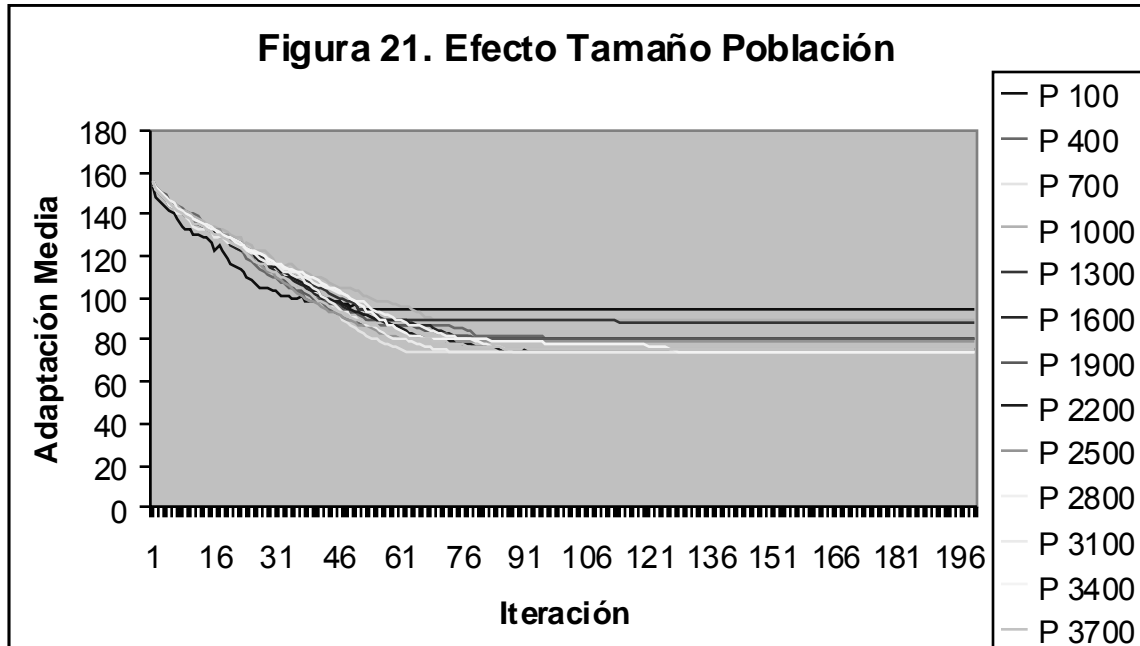
La mejor adaptación media se logró sin embargo con un porcentaje de cruce de 60%, pero si se observa el comportamiento de la adaptación media con este porcentaje a lo largo de las 100 generaciones se nota que alrededor de la generación 30 ocurrió un cambio en la tendencia de esta. Un análisis profundo de esa ejecución particular del algoritmo demostró que en la generación número 35 se produjo una operación de

mutación que generó un individuo mejor adaptado el cual influyó sobre la convergencia del algoritmo.

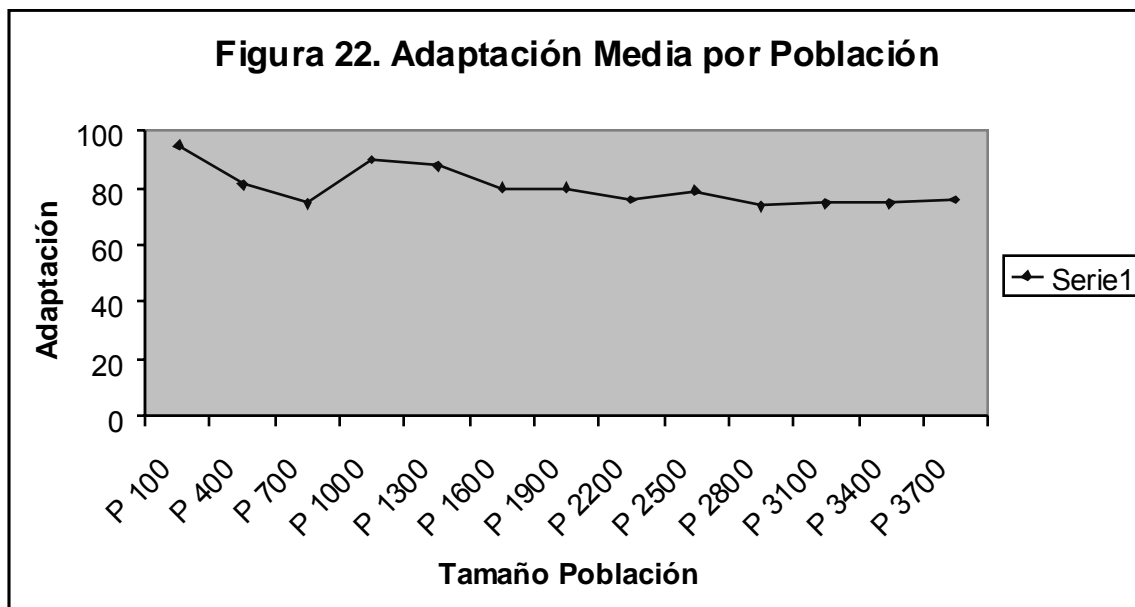
5.4.3 Efecto del tamaño de la población

Otro de los parámetros de entrada del algoritmo genético es el tamaño de la población o número de individuos en cada una de las generaciones. Desafortunadamente, este no es un parámetro sobre el cual se encuentren muchas recomendaciones en la literatura de algoritmos genéticos. En este experimento se fijaron los parámetros de cruce y mutación en 40% y 5% respectivamente y se varió el tamaño de la población entre 100 y 3700 individuos por generación. Con este experimento se pretendió evaluar dos aspectos: el efecto del tamaño de la población respecto de la velocidad de convergencia del algoritmo y el valor de la adaptación media respecto del tamaño de la población.

En la figura 21 se presenta una gráfica del comportamiento de la adaptación media de la población en las 200 iteraciones que se efectuaron en el algoritmo, para cada uno de los tamaños de población considerados. En esta se puede observar una notable diferencia en la velocidad de convergencia que se obtuvo con la población de 100 individuos; sin embargo, no se percibe un cambio muy notable entre las velocidades de convergencia de los demás tamaños de población.



En el Figura 22 se muestra una gráfica de la adaptación final obtenida en cada una de las ejecuciones del algoritmo genético con los diferentes tamaños de población. En la gráfica se puede observar una tendencia a mejorar la adaptación media de la población con el aumento del tamaño de población hasta cierto punto (en este caso 2800) en el que se logra una estabilización de los resultados. Sin embargo se presentó un caso excepcional de adaptación con el tamaño de población de 700, el cual fue generado gracias a una operación de mutación que logró un individuo mucho mejor adaptado que los demás.



La estabilización de los resultados que se observó luego del experimento realizado con el tamaño de población de 2800 se puede explicar por medio de la teoría de muestreo. Esta teoría establece los tamaños de muestra que se deben tomar sobre una población para poder extender los resultados al conjunto total con un cierto nivel de confianza. Lo particular de la teoría es que luego de un alto nivel de confianza, así se aumente el tamaño de la muestra que se escoge, los resultados que se obtienen son muy similares.

Cuando se conoce el tamaño total de la población, la teoría de muestreo establece que el tamaño de la muestra está dado por [LAR99]:

$$n = \frac{Z^2 pqN}{NE^2 + Z^2 pq}$$

Donde:

- n es el tamaño de la muestra
- Z es el nivel de confianza como una variable aleatoria con distribución normal
- p es la variabilidad positiva (para experimentos que no tienen antecedentes se calcula como 0.5)
- q es la variabilidad negativa (es complementaria a la variabilidad positiva)
- N es el tamaño de la población

- E es la precisión o error (no es complementario con el nivel de confianza)

Si se toma en cuenta la forma como funcionan los algoritmos genéticos, tiene sentido que el tamaño de la población que se utilice sea considerado como una muestra del espacio total de soluciones ya que cada uno de los individuos representa implícitamente a una familia de soluciones. En la siguiente tabla se observan los tamaños de muestra sugeridos por la teoría de muestreo para el experimento considerado a diferentes niveles confianza:

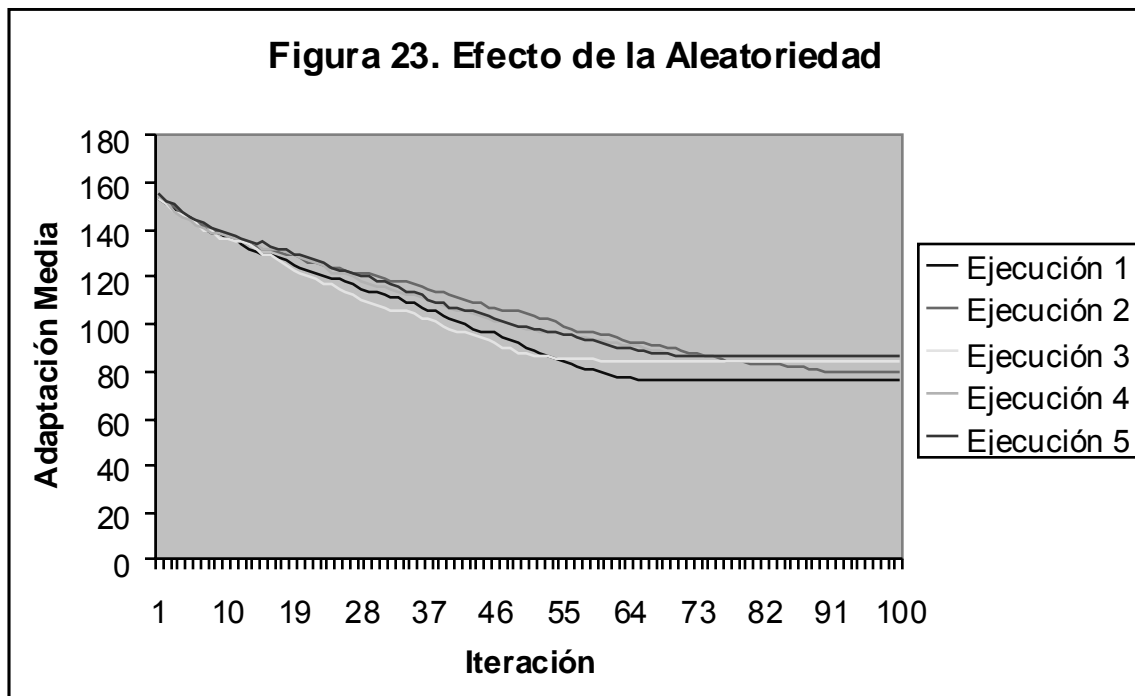
Confianza	Muestra
95	1681
97	2209
99	3393
99.9	5968
99.99	8649
99.999	11342

Tabla 2. Tamaño de muestra a diferentes niveles de confianza

5.4.4 Efecto de la aleatoriedad

Luego de haber detectado en los experimentos previos casos en los que gracias a cambios afortunados en alguno de los individuos se lograron muy buenos resultados, se decidió hacer un experimento en el que se ejecutase varias veces el algoritmo con los mismos parámetros de entrada y así poder determinar que tanto influye la “suerte” en la obtención de resultados de un algoritmo genético.

Para el experimento se establecieron los parámetros de entrada en: porcentaje de cruce 40%, porcentaje de mutación 5%, tamaño de población 2200 y número de iteraciones 100. La ejecución del algoritmo se repitió 5 veces. En la figura 23 se observa una gráfica del comportamiento de la adaptación media en cada de las ejecuciones del algoritmo. En esta se puede apreciar que si existe una variación significativa en el comportamiento y los resultados obtenidos por el algoritmo genético. Por ejemplo la segunda ejecución tardó mucho mas tiempo en estabilizarse que la tercera y la adaptación media final que se encontró en la quinta ejecución es significativamente mayor que la encontrada en la primera ejecución.



Los resultados de este experimento llevan a concluir que la aleatoriedad o “suerte” si juega un papel importante en la obtención de resultados de un algoritmo genético y que por tanto para producir una solución satisfactoria se debería antes ejecutar el algoritmo genético en varias ocasiones.

5.5 Herramienta de simulación para comparación de resultados

Con el objetivo de comparar los resultados obtenidos por el algoritmo genético desarrollado contra una solución generada a través de una estrategia tradicional, se tomó la decisión de implementar una herramienta de simulación. En esta sección se describen algunos de los aspectos de diseño de la herramienta y se presentan los resultados obtenidos.

5.5.1 Descripción funcional de la herramienta

La simulación toma dos parámetros de entrada:

- Un conjunto de pedidos junto con los productos que componen cada uno.

- Una distribución factible de la bodega en la que se detalla la ubicación de cada uno de los productos.

Con estos parámetros, la simulación programa un evento de llegada por cada uno de los pedidos, traza una ruta para recoger los productos a ser despachados y calcula la distancia que se debe recorrer para la consecución del pedido.

Desafortunadamente, para el prototipo no se contaba con información de pedidos y por esto se tuvo que adicionar una funcionalidad en la herramienta que generase esta información de manera que respetase el soporte de los grupos de ítems frecuentes descritos en la definición del prototipo.

5.5.2 Generación de pedidos entrantes

Para generar pedidos que presentaran el comportamiento descrito por los ítems frecuentes se implementó una funcionalidad en la simulación con los siguientes pasos:

1. Se determina aleatoriamente el tamaño de un pedido.
2. Se selecciona un producto aleatoriamente y se adiciona al pedido.
3. Se determinan los grupos de ítems frecuentes en los que se encuentra el producto seleccionado.
4. Para cada grupo de ítems se genera un número aleatorio, si este es menor que el valor de soporte del grupo, todos los productos de este se adicionan al pedido siempre y cuando no existan previamente en el pedido y no sobrepasen el número de productos permitidos para el pedido.
5. Se repiten los pasos 2 a 4 hasta que se alcanza el tamaño máximo de productos para el pedido.
6. Se repiten los pasos 1 a 5 hasta producir la cantidad de pedidos requerida.

5.5.3 Diseño de la herramienta

Luego de analizar las funcionalidades que se requerían para la herramienta de simulación y evaluar algunas de las herramientas genéricas de simulación existentes se tomó la decisión de hacer la simulación con un desarrollo a la medida.

Para lograr el desarrollo de la herramienta de simulación se siguió un enfoque orientado por objetos. Bajo este enfoque se determinó que era necesario desarrollar las siguientes clases:

- **Simulacion:** es la clase principal del programa. Está encargada de solicitar la generación de los pedidos entrantes y utilizarlos como insumo, junto con la distribución de productos que se haya establecido, para determinar la distancia que se debe recorrer para la consecución de cada uno de los pedidos.
- **muestraPedidos:** su función es garantizar que se produzcan pedidos que tengan el comportamiento descrito por la lista de ítems frecuentes. Para lograr esto, implementan la lista de pasos que se enunciaron anteriormente.
- **pedido:** representa un conjunto de productos junto con un número que los identifique de manera única.
- **grupoItems:** agrupa todo el conjunto de ítems frecuentes que se determinen.
- **itemsFrecuentes:** contiene un conjunto de productos que se despachan usualmente juntos en los pedidos. Almacena adicionalmente el valor de soporte que se determinó para el conjunto.
- **configuracion:** es la clase encargada de conocer las localizaciones asignadas a cada uno de los productos dentro de la distribución de productos que se determine.
- **funcionEvaluacion:** esta clase se encarga de determinar que tan distantes se encuentran los productos pertenecientes a uno de los pedidos.
- **cromosoma:** esta clase se utiliza para representar la solución planteada y permitir que otras clases tengan acceso a la información de la ubicación de los productos. Está compuesta por varios conjuntos de genes.
- **gen:** representa cada una de las listas de tipos de productos. Está compuesto por un conjunto de alelos.
- **alelo:** representa la ubicación de un producto particular en una localización específica.

En el Anexo 3 se puede observar un diagrama de clases del software en el que gráficamente se determinan las relaciones entre las clases y su dependencia.

5.5.4 Ejecución de la herramienta con diferentes distribuciones de productos

Con la herramienta de simulación se quisieron comparar tres distribuciones de productos: una es la que se tomó como base para producir los grupos de ítems frecuentes (ver figura 15) y que por tanto debería ser la óptima, otra es aquella que sugiera el algoritmo genético como la mejor y la otra es una en la que se organicen los productos con base en la frecuencia de ventas, como es sugerido en [FRA03].

Para seleccionar la segunda distribución de productos se ejecutó el algoritmo genético en varias ocasiones variando los parámetros de cruce entre 0.3 y 0.4, los parámetros de mutación entre 0.01 y 0.05 y dejando fijo el tamaño de la población en 2800. Finalmente el mejor resultado se obtuvo con el porcentaje de cruce de 0.4 y el porcentaje de mutación de 0.05. En la figura 24 se puede observar la distribución de productos sugerida por el algoritmo genético.

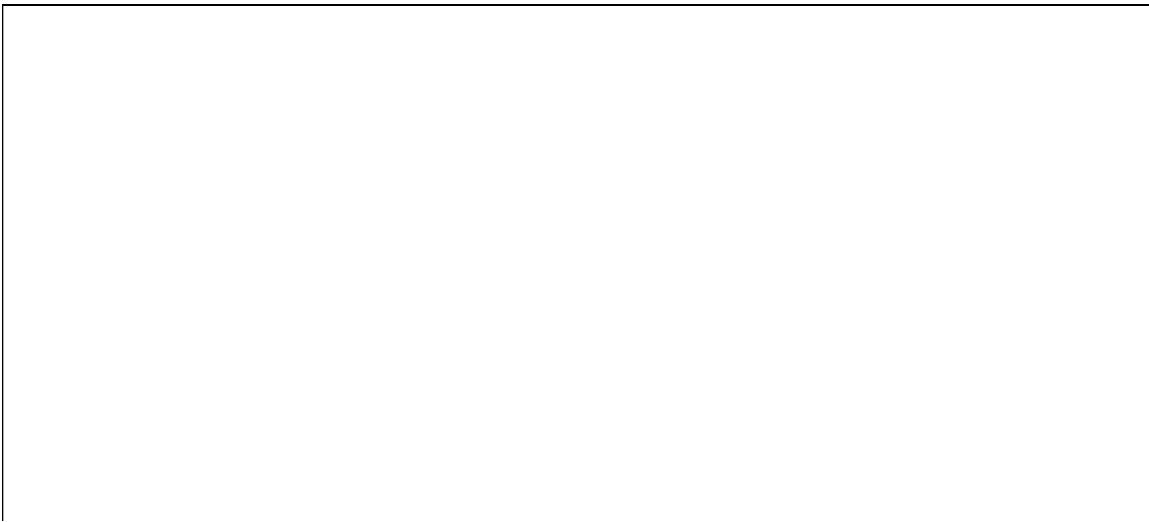
14	7	12	6	26	25
20	8	5	4	27	28
13					29
3	9	19	17	24	30
2	10	18	15	23	31
1	11	16	21	22	32

Figura 24. Distribución de productos sugerida por el algoritmo genético

Como se mencionó, la otra distribución de productos que se quiere comparar es una en la que se ordenen los productos de acuerdo con su frecuencia de venta. Como claramente en este experimento no se cuenta con información histórica que permita determinar la frecuencia de venta de cada uno de los productos, estos datos se obtuvieron de los pedidos generados por la herramienta de simulación. Resulta válido extraer la frecuencia de venta de los productos con base en esta información porque aquellos productos que participan en varios conjuntos de ítems frecuentes muy probablemente aparecerán en mayor proporción e incluso se podría llegar a esperar que la distribución de productos que se sugiera, tiene ventaja frente a las otras porque organiza los productos conociendo por adelantado los pedidos que se van a solicitar. El mecanismo que se siguió para obtener esta distribución de productos fue el siguiente:

- Se generaron 100 pedidos mediante la funcionalidad de la herramienta de simulación.
- Se determinó el soporte de cada uno de los productos.
- Se categorizaron los productos de acuerdo con el tipo de localización en la que deben ser almacenados.
- Dentro de cada una de las categorías se ordenaron descendientemente los productos de acuerdo con su soporte.
- Para cada una de las categorías se tomaron uno a uno los productos y en su orden se asignaron a las localizaciones iniciando en la más cercana al punto de inicio de recolección y terminando en la más lejana.

La distribución obtenida se puede observar en la figura 25.



14	24	12	27	28	30
29	8	10	4	17	16
18					1
21	19	25	2	11	6
9	15	13	5	31	23
20	3	22	32	26	7

Figura 25. Distribución de productos determinada por la frecuencia de venta de cada producto

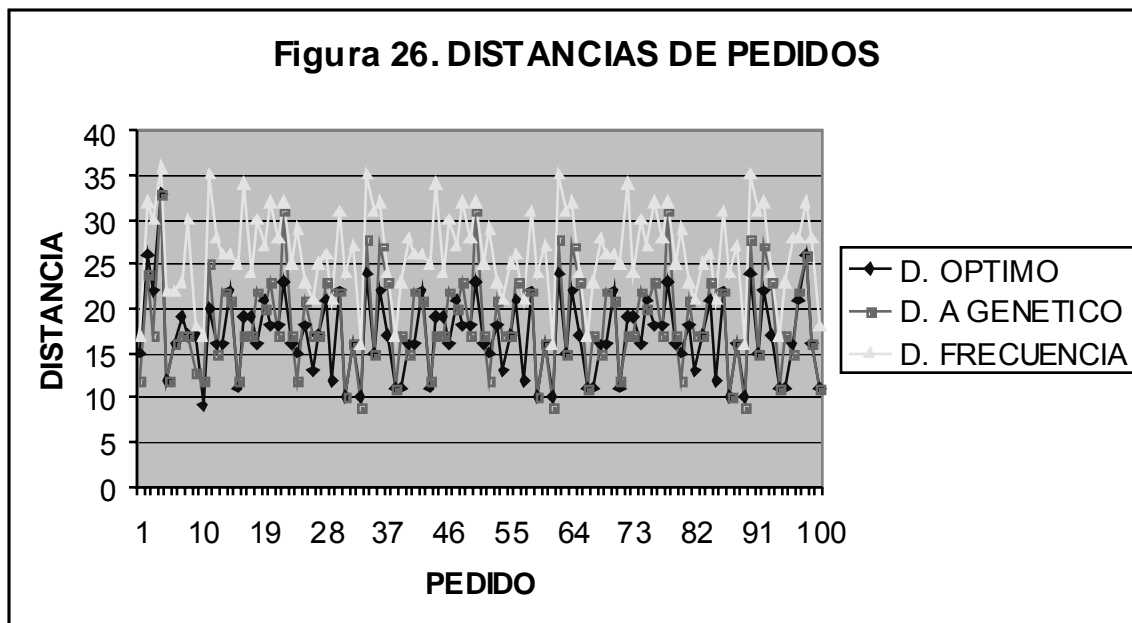
5.5.5 Evaluación de resultados

Luego de ejecutar la herramienta de simulación para cada una de las distribuciones consideradas y con el mismo grupo de 100 pedidos que se utilizó para determinar la frecuencia de venta de los productos se determinó, como era de esperarse, que la distribución en la que se recorre una menor distancia promedio para obtener los pedidos, es aquella que se tomó de base para crear la lista de grupos de ítems frecuentes.

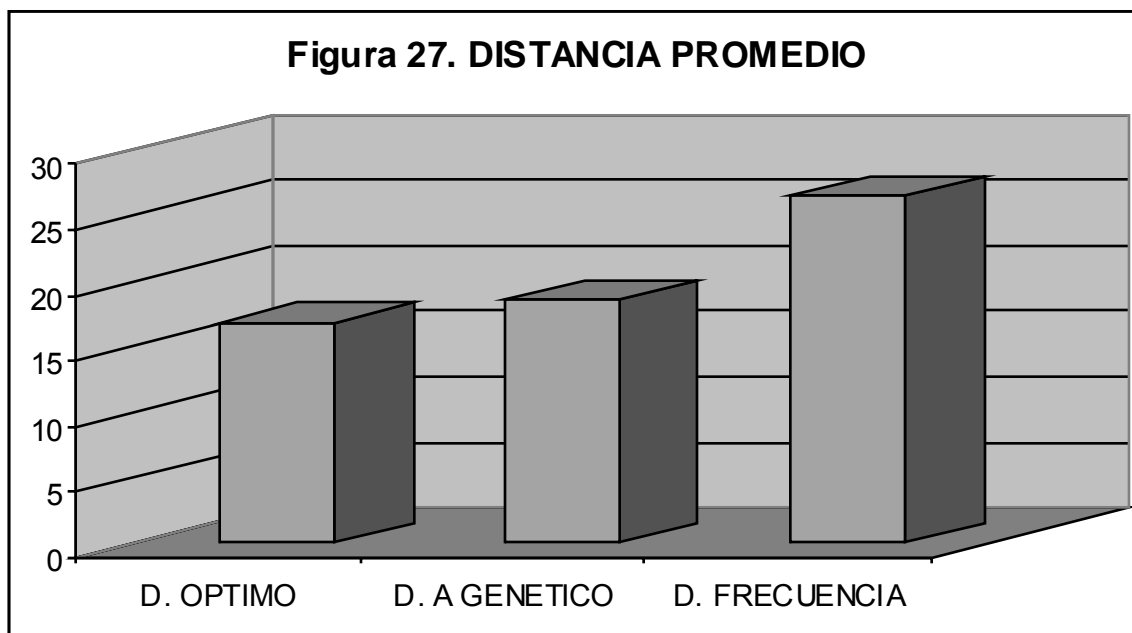
Entre las dos distribuciones restantes, la que menor distancia promedio generó, fue la sugerida por el algoritmo genético.

El bajo rendimiento obtenido por la distribución sugerida a través de la frecuencia de venta de los productos se puede atribuir quizás a que las medidas de soporte que se encontraron en los productos no fueron muy diferentes entre sí y por tanto la asignación de las localizaciones pudo haberse convertido en un proceso pseudo-aleatorio.

En figura 26 se presenta una gráfica de la distancia calculada para cada uno de los pedidos considerados en cada una de las tres distribuciones. Es importante resaltar que hubo incluso algunos puntos en los que la distribución sugerida por el algoritmo genético tuvo mejor rendimiento que la utilizada como base en la definición de los ítems frecuentes.



En figura 27 se encuentra una gráfica en la que se muestran los valores de las distancias promedio encontradas para cada una de las distribuciones que se consideraron.



6 IMPLEMENTACIÓN EN UN CASO REAL

La factibilidad o no de la estrategia de solución propuesta se puede evaluar de mejor manera, aplicándola a un caso real. Gracias a la colaboración de la multinacional Carvajal S.A. se pudo tener acceso a la información e instalaciones del centro de distribución de Ofixpress S.A.

Ofixpress S.A. es una empresa del grupo Carvajal S.A. creada en 1996, que se especializa en la distribución de suministros para oficina. El centro de distribución de Ofixpress S.A. al que se pudo tener acceso se encuentra ubicado en Bogotá en la carrera 50 # 9A-35 y cuenta con 33 tipos diferentes de localización, 4169 localizaciones distribuidas en un área de 7000 metros cuadrados y 4006 tipos de productos en existencia al momento del análisis.

La identificación de cada una de las localizaciones se logra a través de un código compuesto por 4 números:

- El primero identifica el pasillo.
- El segundo indica el módulo que es una estructura metálica estándar diseñada para almacenar estivas en la parte superior. Un valor impar en este número indica que el módulo se encuentra ubicado en el costado derecho y uno par que se encuentra en el costado izquierdo.
- El tercero indica el nivel o altura al que se encuentran los productos.
- El cuarto indica la posición dentro del módulo. Dependiendo del tamaño de las localizaciones, un módulo se puede dividir hasta en 6 posiciones y tres niveles.

6.1 Obtención de información

Para el desarrollo de este estudio se pudo obtener información de pedidos correspondiente al periodo comprendido entre septiembre de 2005 y marzo de 2006. La información fue suministrada en archivos planos (uno por mes) y fue cargada en una base de datos DB2. El total de registros de ventas cargados fue de 1.031.843.

6.2 Limpieza de información

- Se tuvieron que rechazar 203.960 líneas de ventas por incluir productos que fueron creados temporalmente para satisfacer pedidos específicos de ciertos clientes.
- Se tuvieron que rechazar 66.823 líneas de ventas porque correspondían a productos descontinuados.
- Se rechazaron 18.520 líneas adicionales por contener productos que no tienen una localización asignada en la bodega debido a que tienen muy baja rotación.

6.3 Soporte mínimo, máximo y promedio

Los 742.540 registros de ventas que resultaron luego de limpiar la información, corresponden a 59.352 pedidos diferentes.

En los registros de ventas se hace referencia a 2903 productos distintos. El soporte mínimo fue 0,0000168 y correspondió a 87 productos que únicamente aparecieron en 1 pedido. El soporte máximo fue 0,426 que corresponde a un producto que se vendió en 25.302 pedidos. El soporte promedio fue 0,0043.

6.4 Modelo de asociación

En desarrollo del modelo de asociación se utilizó la herramienta Intelligent Miner de IBM. Para construir el modelo en el software fueron necesarias, entre otras, las siguientes tareas:

- Definir la fuente de datos a utilizar dentro de la base de datos DB2.
- Identificar dentro de la fuente de datos el campo que identifica la transacción y el campo de los ítems.
- Definir los parámetros de ejecución del modelo:
 - Soporte mínimo: Establece el umbral de soporte bajo el cual se descartarán los ítems de productos. Se puede permitir que el software establezca el valor de este parámetro. Para calcular el valor, el software identifica el soporte de cada uno de los ítems por separado y luego

determina un valor tal que la cuarta parte de los productos tenga un soporte superior.

- Confianza mínima: Indica el valor mínimo de confianza que puede tener una regla de asociación para ser tomada en cuenta. El valor por defecto de este parámetro es 25%, sin embargo cuando se buscan ítems frecuentes, el valor de este parámetro no tiene relevancia.
- Máximo tamaño de las reglas: para el caso de búsqueda de ítems frecuentes indica el número máximo de ítems que se considerará dentro de cada grupo. Es posible establecer que no exista límite, caso en el cual el software busca grupos de ítems incrementando el tamaño hasta que el límite de memoria del computador lo permita.

6.5 Ítems frecuentes Obtenidos

Para la obtención de ítems frecuentes se ejecutó varias veces el modelo de asociación modificando el parámetro de soporte mínimo. Finalmente se decidió aceptar los resultados obtenidos con un soporte mínimo de 0.25 y un tamaño ilimitado porque con estos fue posible obtener grupos de hasta 10 ítems frecuentes.

El número total de grupos de ítems frecuentes que se encontró fue de 64929 distribuidos así:

- 755 grupos con 1 producto
- 7028 grupos con 2 productos
- 16745 grupos con 3 productos
- 18607 grupos con 4 productos
- 11954 grupos con 5 productos
- 6246 grupos con 6 productos
- 2703 grupos con 7 productos
- 776 grupos con 8 productos
- 112 grupos con 9 productos
- 3 grupos con 10 productos

Naturalmente los grupos de 1 producto fueron descartados. En el Anexo 4 se puede observar la interfaz de Intelligent miner para visualizar los resultados y en el Anexo 5 una lista de los grupos de ítems con mayor nivel de soporte.

6.6 Ajustes al algoritmo genético

6.6.1 Función de distancia

Para este centro de distribución particular la función de distancia entre dos localizaciones se implemento de la siguiente manera:

- Si es necesario efectuar un cambio de pasillo:
 1. Se encuentra el pasillo horizontal más cercano a ambos puntos.
 2. Se determina la cantidad de pasillos que las localizaciones se encuentran distanciadas y se multiplica por 4 que es la distancia en metros que existe comúnmente entre dos pasillos.
 3. Se determina la cantidad de módulos existente entre la localización 1 y el pasillo encontrado en 1 y se multiplica por 2 que es la distancia promedio en metros que existe entre dos módulos.
 4. Se determina la cantidad de módulos existente entre la localización 2 y el pasillo encontrado en 1 y se multiplica por 2.
 5. Se suman los resultados de 2, 3 y 4.
- Si solo es necesario efectuar un cambio de módulo:
 1. Se determina la cantidad de módulos existente entre la localización 1 y la localización 2 y se multiplica por 2.
- Si solo es necesario hacer un cambio de posición dentro del mismo módulo:
 1. Se determina la cantidad de posiciones existente entre la localización 1 y la localización 2 y se multiplica por 0.5 que es la distancia promedio existente entre dos posiciones.
- Si solo se requiere hacer un cambio de nivel, se calcula una distancia de cero.

6.6.2 Programación del algoritmo

Básicamente la implementación del algoritmo genético que se hizo para el experimento se pudo utilizar en el caso real. Los cambios que fue necesario efectuar fueron:

- Modificar la clase configuración para que leyera la información de tipos de localizaciones, productos y ubicaciones, desde una base de datos.
- Modificar la clase grupoItems para que obtuviera la información de los grupos de ítems frecuentes, desde una base de datos.
- Cambiar la función de evaluación para que se ajustara a la nomenclatura de localizaciones del centro de distribución de Ofixpress.

En el Anexo 6 se hace una descripción funcional del software.

6.7 Ajustes al modelo de simulación

Para poder comparar los resultados del algoritmo genético contra la distribución de productos existente en el centro de distribución, fue necesario realizar unos ajustes:

- Modificar la clase muestraPedidos para que en lugar de generar aleatoriamente los pedidos, los leyera de una base de datos en donde se tenía cargada la información de pedidos del mes de abril.
- Al igual que en el algoritmo genético, modificar la clase grupoItems para que obtuviera la información de los grupos de ítems frecuentes, desde una base de datos.
- De manera similar al algoritmo genético, fue necesario modificar la función de evaluación para que se ajustara a la nomenclatura de localizaciones del centro de distribución de Ofixpress.

En el Anexo 6 se hace una descripción funcional del software.

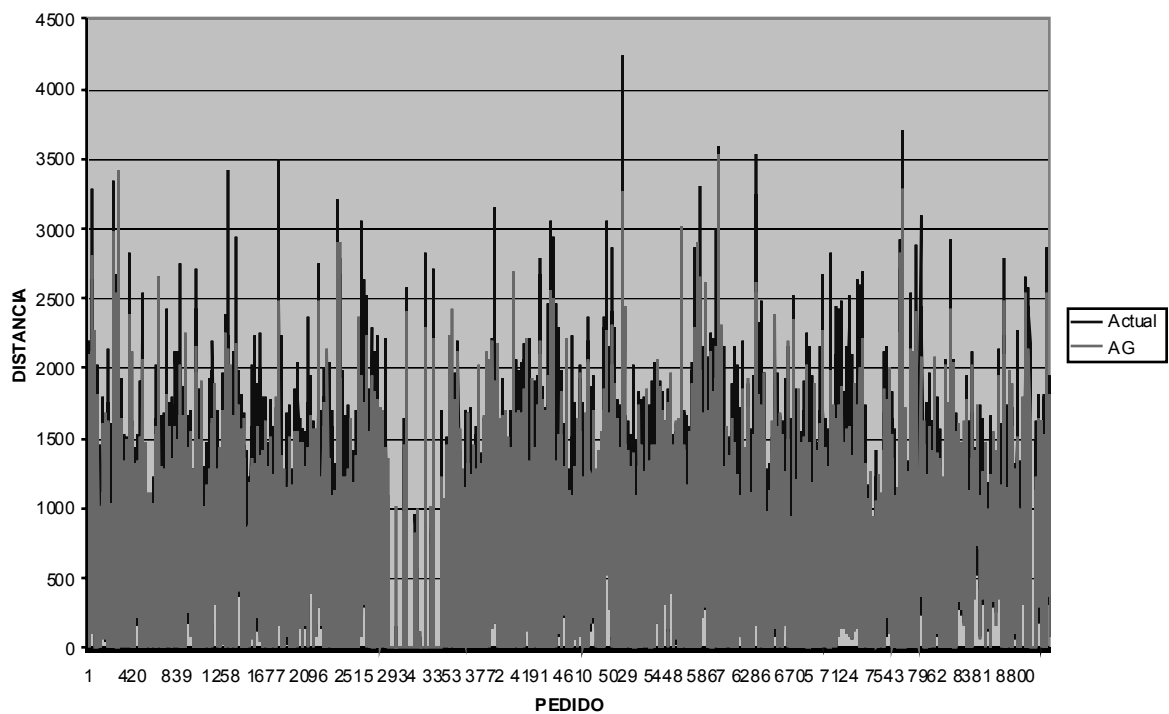
6.8 Resultados Obtenidos

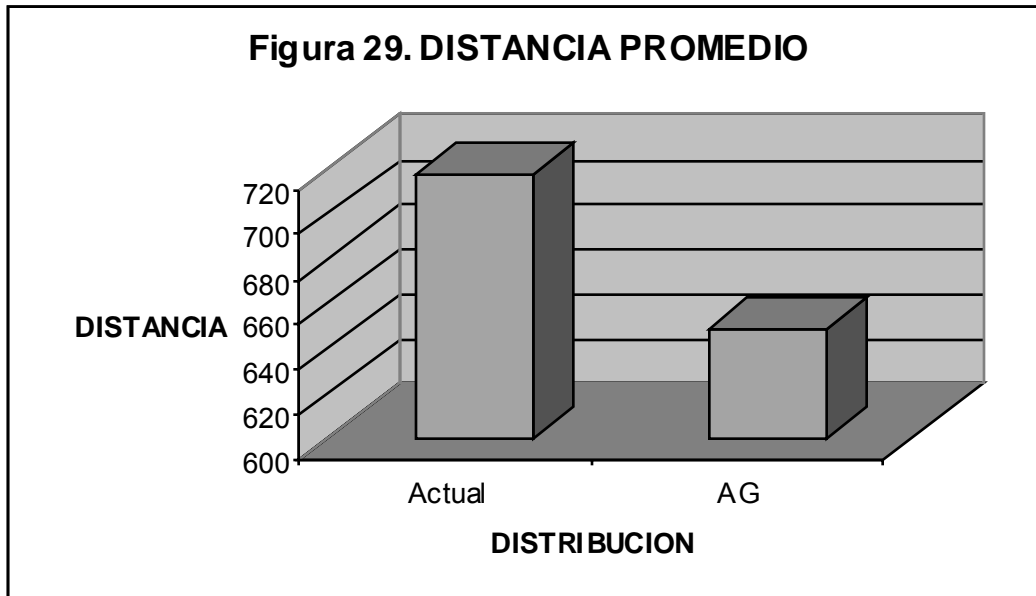
La herramienta de simulación ajustada para el caso de Ofixpress se utilizó para comparar la distancia promedio de recolección de un pedido entre la distribución de productos sugerida por el algoritmo genético y la distribución de productos que actualmente se tiene.

Para la ejecución de la simulación se utilizó la información de pedidos de Ofixpress correspondiente al mes de abril. Esta información está compuesta por 9139 pedidos.

En la figura 28 se puede observar el comportamiento de las dos distribuciones con el total de pedidos. En la figura 29 se presenta el consolidado o distancia promedio que se recorrió en cada una de las distribuciones. Estos resultados mostraron un mejor comportamiento de la distribución sugerida por el algoritmo genético.

Figura 28. DISTANCIA RECORRIDA EN LOS PEDIDOS





7 PROCEDIMIENTO PROPUESTO

Para la obtención y análisis de los resultados se propone ejecutar tres etapas, análisis de información, desarrollo del algoritmo genético y evaluación de resultados. A continuación se describirá cada una de las etapas, teniendo en cuenta la información se requiere como entrada a cada una de ellas, las actividades que se sugiere ejecutar, algunas herramientas de software que pueden ser de utilidad, los resultados que se espera obtener y una lista de sugerencias prácticas resultado de las experiencias adquiridas.

7.1 Etapa 1 “Análisis de Información”

En esta etapa se analiza la información histórica de pedidos para encontrar grupos de productos que usualmente se venden juntos.

7.1.1 Requerimientos

Para este proceso se requiere tener una base de datos con la información histórica relativa a los pedidos. Como mínimo se requiere:

- Código de pedido
- Código de cada uno de los productos

Adicionalmente es importante conocer los productos que actualmente se ofrecen en el centro de distribución.

7.1.2 Actividades

- Recopilar la información histórica de una ventana de tiempo que no esté viciada por cambios drásticos en la forma de operación del negocio, ni estrategias comerciales agresivas.
- Descartar aquella información que hace referencia a productos que fueron creados temporalmente para satisfacer necesidades específicas de los clientes o que involucra productos que ya no se venden más en el negocio.

- Obtener el soporte mínimo, máximo y promedio de los productos para tener una idea de los parámetros a utilizar en la herramienta seleccionada. Esto se puede lograr fácilmente en un motor de base de datos, agrupando la información por los productos, contando la cantidad de veces que apareció cada uno y dividiendo entre la cantidad de pedidos considerados.
- Construir el modelo de asociación en la herramienta de software seleccionada.
- Ejecutar el modelo de asociación varias veces con diferentes parámetros hasta obtener grupos de productos frecuentes con niveles de mejoramiento mayores a 1.
- Descartar los resultados que solo involucren un producto o que no tengan niveles de mejoramiento mayores a 1.

7.1.3 Software disponible

Para efectuar este análisis existen múltiples herramientas, las más conocidas de ellas son las que se distribuyen con los motores de bases de datos Oracle, Microsoft SQL Server (Análisis services) y DB2 (Intelligent miner).

7.1.4 Resultados esperados

Como resultado de esta etapa se obtendrá una lista de grupos de productos frecuentes, cada uno con un porcentaje de soporte.

7.1.5 Sugerencias

Para la ejecución del modelo de asociación se recomienda tomar como parámetros iniciales el soporte promedio y un tamaño ilimitado de grupos.

7.2 Etapa 2 “Desarrollo del Algoritmo Genético”

En esta etapa se desarrolla el algoritmo genético que evalúa miles de posibles distribuciones del centro de distribución y selecciona aquella que mejor cumpla con los

grupos de ítems frecuentes. El cumplimiento de los grupos se determina respecto de que tan lejos (físicamente) se encuentran los productos que componen cada uno de estos.

7.2.1 Requerimientos

Para esta etapa se requieren los grupos de ítems frecuentes obtenidas en la etapa 1 y adicionalmente información de los diferentes contenedores con que cuenta el centro de distribución (tipo de contenedor, nombre, dimensiones, ubicación) y una relación de los tipos de contenedor que se utilizan con cada uno de los productos (código de Producto, tipo de contenedor).

7.2.2 Actividades

- Diseñar un modelo de codificación que permita representar la bodega.
- Definir una función de distancia que permita evaluar que tan cerca o lejos se encuentran dos productos.
- Definir la implementación de los operadores genéticos de cruce y mutación que se utilizarán sobre el modelo de codificación.
- Definir una función de evaluación que involucre los ítems de productos frecuentes y la función de distancia.
- Establecer el criterio que se utilizará para detener la ejecución del algoritmo.
- Implementar el algoritmo genético con las definiciones establecidas previamente.
- Ejecutar el algoritmo genético varias veces ajustando los parámetros de tamaño de población, porcentaje de cruce y porcentaje de mutación hasta obtener los resultados esperados.

7.2.3 Software disponible

Además del software que se desarrolló como parte de este proyecto, es importante mencionar algunas librerías de algoritmos genéticos que facilitan el desarrollo de estos. Las más destacadas se relacionan en la tabla 3.

Software	URL	Plataforma
gajit	http://www.micropraxis.com/gajit/index.html	Java
gaa	http://www.aridolan.com/ga/gaa/gaa.html	Java
galib	http://lancet.mit.edu/ga/	C++
javaEva	http://www-ra.informatik.uni-tuebingen.de/software/JavaEvA/	Java
optiGA	http://www.optiwater.com/optiga.html	VB
Beagle	http://beagle.gel.ulaval.ca/	C++

Tabla 3. Librerías para desarrollo de algoritmos genéticos

7.2.4 Resultado

La distribución de productos que el algoritmo se determine como la mejor.

7.3 Etapa 3 “Evaluación de resultados”

En esta etapa se evalúa el desempeño de la distribución sugerida por el algoritmo contra otras distribuciones incluida la distribución actual de productos. Para esto se debe desarrollar un modelo de simulación que determine la distancia promedio que toma la obtención de los pedidos en cada una de las distribuciones.

7.3.1 Requerimientos

La distribución de productos sugerida por el algoritmo genético, las demás distribuciones a considerar y un conjunto de pedidos que no haya sido utilizado en la primera etapa.

7.3.2 Actividades

- Obtener la información de pedidos a utilizar.
- Crear un modelo de simulación por eventos en el que la variable objetivo sea la distancia promedio recorrida.
- Implementar, en el modelo de simulación, la misma función de distancia definida en la etapa dos.

- Programar los eventos de llegada de los pedidos.
- Ejecutar el modelo de simulación con la distribución de productos sugerida por el algoritmo genético.
- Ejecutar el modelo de simulación con las demás distribuciones de productos consideradas.
- Analizar los resultados obtenidos.

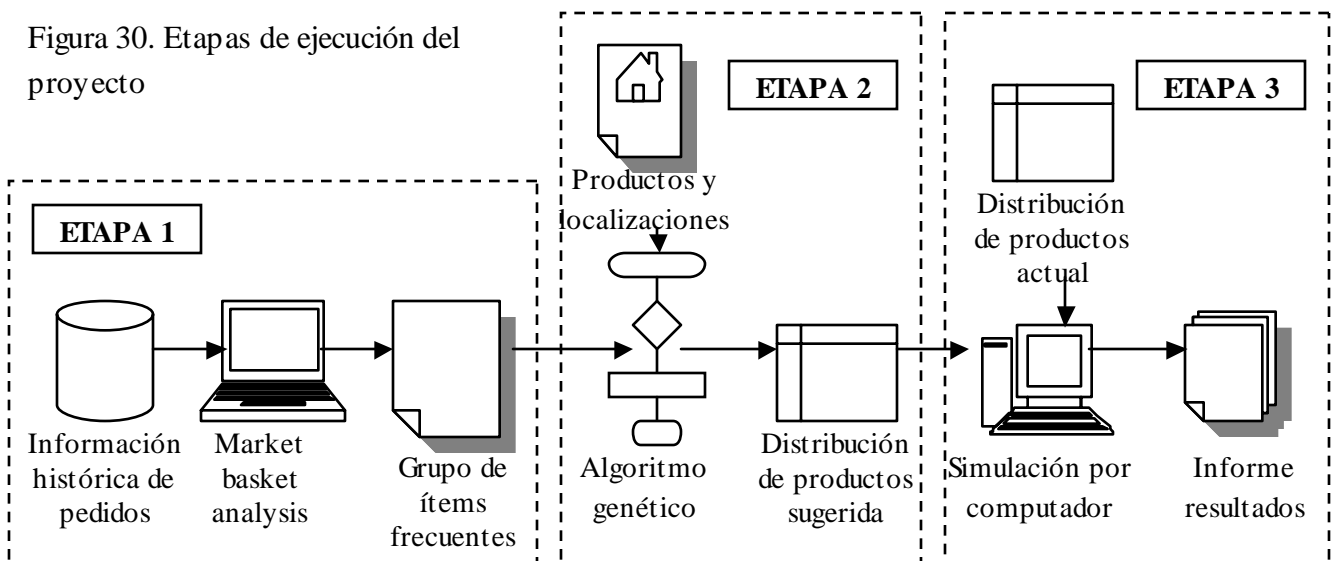
7.3.3 Software disponible

Para desarrollar simulaciones existe una amplia variedad de herramientas, entre ellas está: Ithink Analyst, Arena professional, Aspen Suite y GAMS Development.

7.3.4 Resultado

Cálculo de la distancia promedio recorrida en cada una de las distribuciones de productos.

En la figura 30 se puede ver un diagrama de cada una de las etapas, con los recursos que utilizan, los productos que se deben obtener y la forma en que estas interactúan.



CONCLUSIONES

- Con los resultados obtenidos en la implementación de los algoritmos genéticos dentro de la estrategia y el análisis bibliográfico de otros tipos de implementaciones que se han desarrollado utilizando esta técnica, se puede concluir que los algoritmos genéticos resultan una buena alternativa para encontrar soluciones aproximadas en problemas de permutación en los que resulta imposible evaluar todas las soluciones posibles.
- En los dos casos analizados, la estrategia de solución que se planteó demostró ser capaz de obtener resultados en los que, de acuerdo a la herramienta de simulación, se tienen mejores desempeños en la recolección de pedidos que con otras soluciones tanto empíricas como tradicionales.
- El método de codificación que se seleccione para representar una solución dentro de los algoritmos genéticos es determinante en la forma como se implementan los operadores genéticos y la función de evaluación.
- El operador de cruce en los algoritmos genéticos, es el que tiene la mayor influencia en la generación de las soluciones y el que adicionalmente permite que los algoritmos genéticos converjan hacia alguna solución.
- El operador de mutación por sí solo no genera ninguna diferencia para lograr una mejor adaptación de la población, pero combinado con el operador de cruce puede marcar la diferencia entre obtener un resultado promedio y una buena solución.
- En los algoritmos genéticos, la aleatoriedad juega un papel importante y por tanto así se conozcan los parámetros óptimos con los que se debe ejecutar el algoritmo, es recomendable ejecutarlo varias veces y aceptar el mejor resultado.
- El tamaño de la población que se utilice para la ejecución de los algoritmos genéticos debe ser suficiente para permitir que todas las posibles soluciones se encuentren representadas, pero no debe ser muy amplio porque incrementa el tiempo de ejecución del algoritmo sin ofrecer mejoras substanciales en los resultados obtenidos.

- El Market Basket Analysis es una técnica en la que los resultados obtenidos dependen, en gran porcentaje, de los parámetros de entrada que se establezcan y por tanto siempre que se trabaje con ella es recomendable experimentar con diferentes valores antes de aceptar los resultados.
- La función de distancia entre dos localizaciones de un centro de distribución se debe definir para cada caso particular porque depende completamente de la forma como se encuentre diseñado cada centro.

BIBLIOGRAFIA

[BAI03] Baier Jorge (2003) “Algoritmos genéticos” descargado el 03 de septiembre de 2005 del sitio web <http://www2.ing.puc.cl/~jabaier/iic2622/gp.pdf>

[BAR05] Bartholdi John, Hackman Steven (2005) “Warehouse & Distribution Science” descargado el 5 de enero de 2006 del sitio web <http://www.warehouse-science.com>

[FRA03] Frazelle Ed (1992) “The principles of Order Picking” descargado el 10 de octubre del sitio web <http://www.jvdbconsulting.com/pdf/principles-pricking.pdf>

[GOL89] Goldberg David (1989) “Genetic Algorithms in Search, Optimization, and Machine Learning”, Kluwer Academic Publishers, Boston, MA

[HAU04] Haupt Randy (2004) “Practical Genetic Algorithms”, Wiley Interscience

[HOL75] Holland John (1975) “Adaptation In Natural and Artificial Systems” University of Michigan Press.

[HOLLA] Holland John (Sin Fecha) “Genetic Algorithms” descargado el 21 de agosto de 2005 del sitio web <http://www.econ.iastate.edu/tesfatsi/holland.GAIntro.htm>

[KOO04] Kook Wolfgang (2004) “Análisis de estado del arte en algoritmos genéticos”, Universidad de los Andes

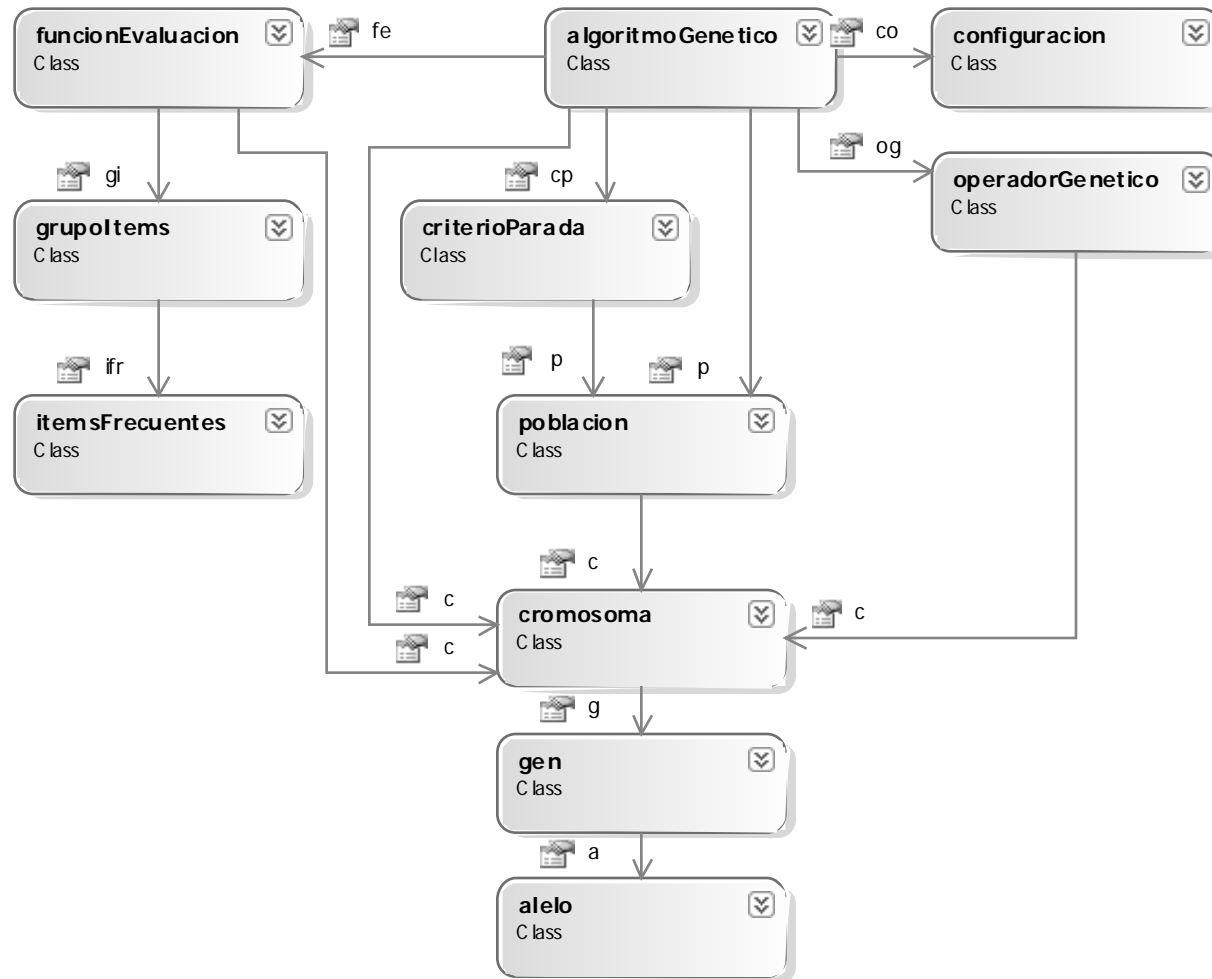
[LAR98] Larrañaga P, Kuijpers C, Murga R (1998) “Tackling the Travelling Salesman Problem with Evolutionary Algorithms: Representations and Operators” descargado el 20 de septiembre de 2005 del sitio web <http://citeseer.ist.psu.edu/210802.html>

[LAR99] Larios Osorio Víctor (1999) “Teoría de Muestreo” Descargado el 17 de abril de 2006 del sitio web <http://www.uaq.mx/matematicas/estadisticas/xu5.html>

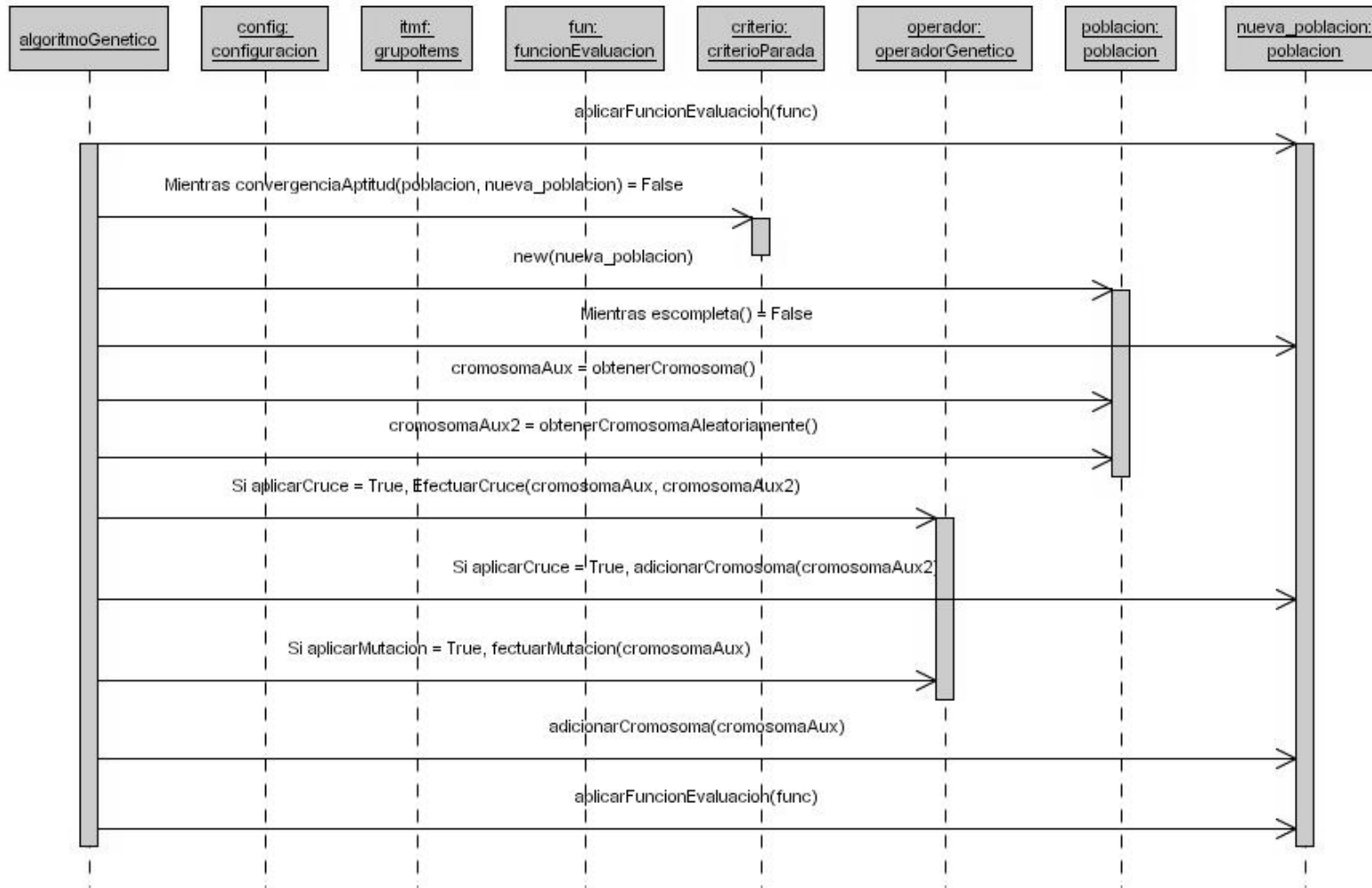
[MEG04] Megaputer Intelligence Inc. (2004) “Market Basket Analysis” Descargado el 18 de octubre de 2005 del sitio web <http://www.megaputer.com/tech/wp/mba.php3>

[WHI93] Whitley Darrell (1993) "A Genetic Algorithm Tutorial" Descargado el 21 de agosto de 2005 del sitio web http://www.cs.uga.edu/~potter/CompIntell/ga_tutorial.pdf

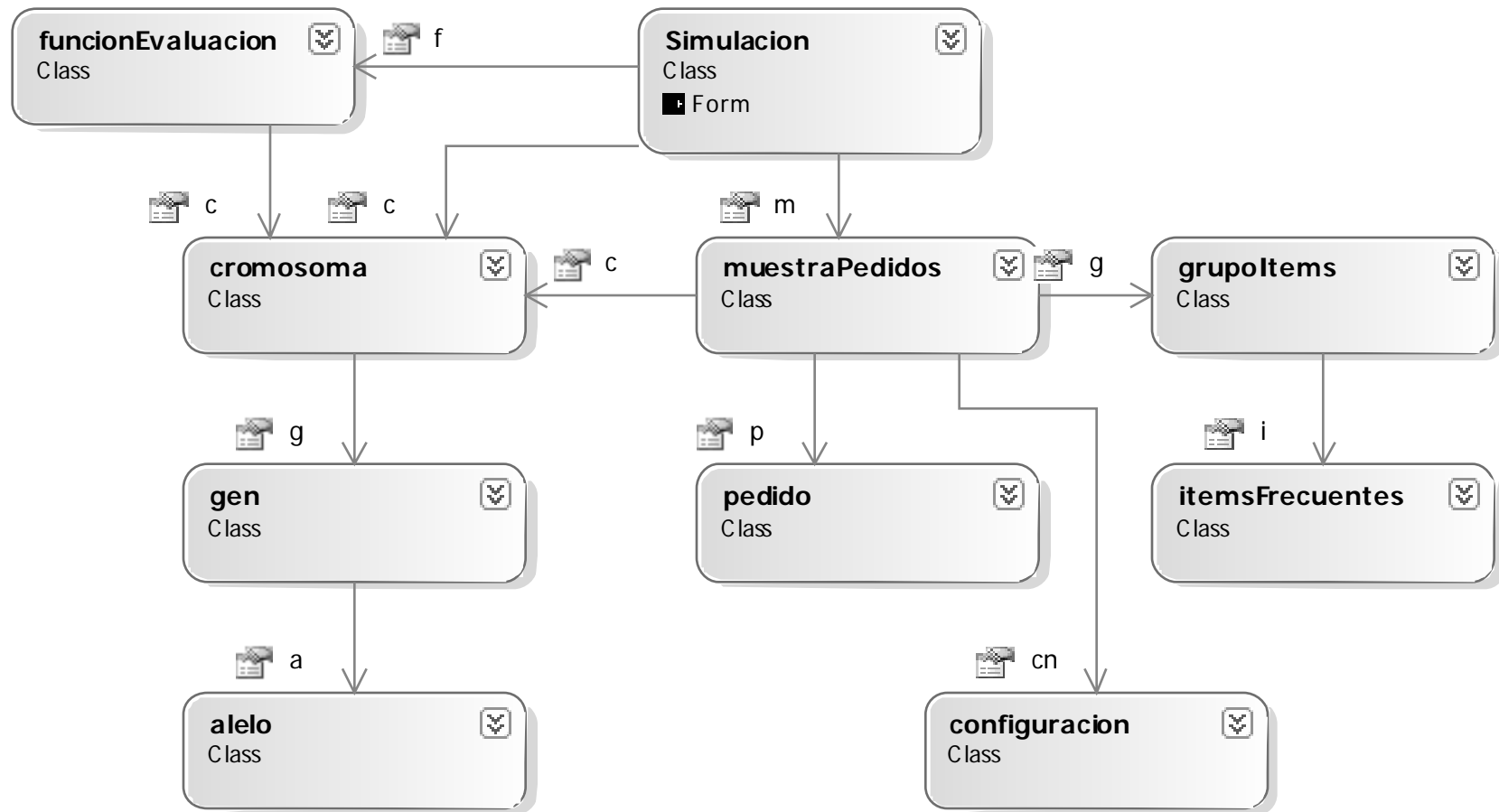
ANEXO 1. DIAGRAMA DE CLASES DEL ALGORITMO GENÉTICO



ANEXO 2. DIAGRAMA DE SECUENCIA PRINCIPAL



ANEXO 3. DIAGRAMA DE CLASES DEL MODELO DE SIMULACIÓN



ANEXO 4. RESULTADOS EN INTELLIGENT MINER

Association Visualizer - C:\DOCUME~1\Diego\CONFIG~1\Temp\VCRI2EGQ.C2T - Local File: IDM22819.xml

File Edit Selected View Help

Rules Item Sets Graph Statistics

Visible item sets:

Item Set	Support	In Rules as Body	In Rules as Head	Items in Set	Number of Rules	Group
[7005125]	42.3288%	0	1,462	1	1,462	3
[7002118]	19.1249%	1	642	1	643	3
[7002130]	15.9809%	2	414	1	416	3
[7020877]	14.1529%	1	459	1	460	3
[7000206]	14.1427%	1	444	1	445	3
[7005126]	12.6988%	1	148	1	149	3
[7005448]	11.6508%	2	290	1	292	3
[7005125]+[7005126]	10.9567%	0	0	2	0	1,177
[7005125]+[7000206]	10.9364%	0	0	2	0	1,178
[7005125]+[7002118]	10.7899%	0	0	2	0	1,187
[7000567]	10.0249%	3	352	1	355	3
[7024399]	9.9828%	1	84	1	85	3
[7002120]	9.5094%	3	112	1	115	3
[7003930]	9.3442%	1	55	1	56	3
[7005125]+[7002130]	9.0393%	1	0	2	1	3
[7024280]	8.7259%	2	85	1	87	3
[7024702]	8.5574%	1	249	1	250	3
[7005125]+[7020877]	8.5102%	1	0	2	1	3
[7000730]	8.4681%	1	155	1	156	3
[7024282]	8.0621%	2	99	1	101	3
[7024101]	7.7824%	2	10	1	12	3
[7004990]	7.7116%	1	597	1	598	3
[7002118]+[7002130]	7.5549%	3	0	2	3	3
[7002698]	6.8962%	2	83	1	85	3
[7000804]	6.8776%	1	15	1	16	3
[7002118]+[7002120]	6.8389%	2	0	2	2	3
[7005125]+[7024399]	6.7917%	0	0	2	0	1,188
[7005125]+[7003930]	6.7849%	0	0	2	0	1,179
[7000678]	6.6266%	2	20	1	22	3
[7004990]	6.2510%	2	25	1	27	3

Item set color: Support

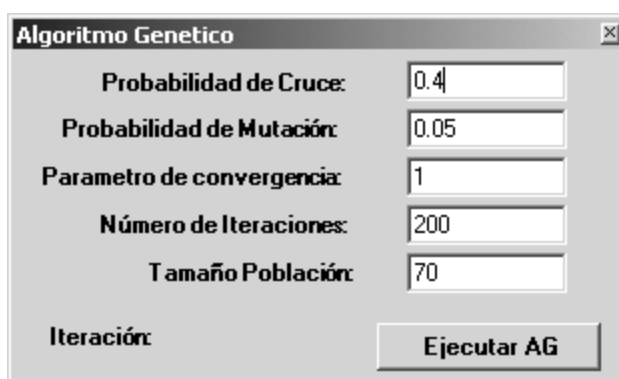
ANEXO 5. ITEMS FRECUENTES ENCONTRADOS

Grupo	Soporte	Lift	Producto 1	Producto 2	Producto 3
63294	10,9567	2,04	7005125	7005126	
64155	10,9364	1,83	7005125	7000206	
63274	10,7899	1,33	7005125	7002118	
64677	9,0393	1,34	7005125	7002130	
63868	8,5102	1,42	7005125	7020877	
64070	7,5549	2,47	7002130	7002118	
64670	6,8389	3,76	7002120	7002118	
63954	6,7917	1,61	7005125	7024399	
63648	6,7849	1,72	7005125	7003930	
63977	6,1834	1,25	7005125	7005448	
64273	5,8313	1,37	7005125	7000567	
64688	5,6477	2,09	7002118	7020877	
63532	5,5735	1,38	7005125	7002120	
64692	5,5449	2,45	7002130	7020877	
64839	5,1877	1,4	7024280	7005125	
63703	5,1725	2,32	7005448	7002118	
63956	5,1371	1,56	7005125	7024101	
63683	4,9636	7,06	7024280	7024282	
64348	4,571	1,34	7024282	7005125	
64808	4,5542	2,03	7005125	7001341	
63874	4,5272	1,56	7005125	7000804	
63955	4,3941	2,2	7020877	7000206	
63312	4,3166	1,2	7005125	7000730	
62117	4,3166	1,19	7005125	7024702	
63511	4,3065	1,32	7005125	7004990	
63721	4,2812	2,23	7000567	7002118	
63704	4,2678	1,58	7002118	7000206	

ANEXO 6. DESCRIPCION FUNCIONAL DEL SOFTWARE DESARROLLADO

Como se mencionó a lo largo del documento, se desarrollaron dos herramientas: el algoritmo genético y la simulación para comparar resultados. Ambas herramientas fueron desarrolladas en el Microsoft Visual Studio 2005. Los archivos de código fuente y ejecutable se incluyeron en un CD adjunto.

El algoritmo genético se desarrolló con una interfaz gráfica similar a la siguiente:



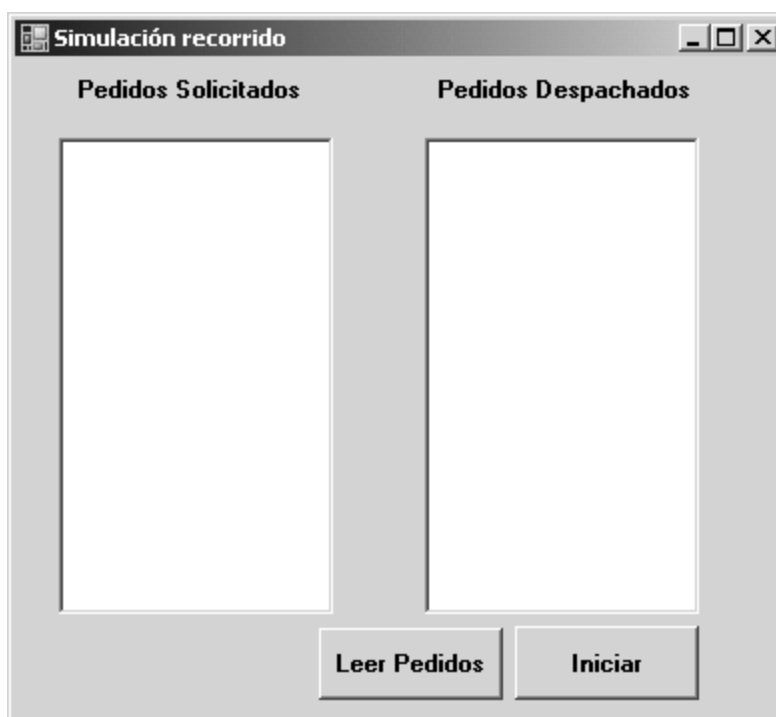
En ella es posible establecer el porcentaje de cruce, el porcentaje de mutación, el umbral de similitud bajo el cual se detendrá la ejecución del algoritmo, un número de iteraciones máximo permitido en la ejecución del algoritmo y el tamaño de la población que se utiliza. El software utiliza el siguiente modelo de datos:



De la tabla Items lee los grupos de ítems frecuentes que se determinaron, de la tabla LocalizadorProducto lee el tipo de localizador en el que se debe almacenar cada uno de los productos y de la tabla localizadoresTotal determina en qué tipo de localizador se clasifica cada una de los localizadores existentes.

El software genera dos archivos de salida: C:\operaciones.sal en el cual se registran todas la operaciones de cruce y mutación que se efectuaron a lo largo de la ejecución del algoritmo y C:\salida.sal en el cual se registra la adaptación media de la población en cada una de las iteraciones y se guarda la mejor distribución de productos que se encontró en la ejecución del algoritmo.

El software de simulación se desarrolló con la siguiente interfaz gráfica:



Para ejecutar el software se debe oprimir primero el botón Leer Pedidos, con lo que la aplicación lee los pedidos registrados en la tabla EvaluationSet e ingresa el número de pedido en la lista de pedidos solicitados (parte izquierda de la pantalla). Una vez finaliza este proceso, es posible oprimir el botón Iniciar con lo que el software toma uno a uno los

pedidos considerados y evalúa la distancia total que se requiere para obtener el pedido. El software lee la tabla `distribucionConsiderada` para determinar la ubicación de cada uno de los productos.

Los resultados de la simulación se registran en el archivo `C:\resPedidos.txt`. En este archivo se almacena la distancia total calculada para cada uno de los pedidos considerados.

La cadena de conexión a la base de datos que se utiliza en cada uno de los desarrollos se puede modificar en los `application configuration file` correspondientes.