



**UNIVERSIDAD AUTÓNOMA DEL ESTADO DE  
HIDALGO**

**INSTITUTO DE CIENCIAS BÁSICAS E INGENIERÍA**

**ÁREA ACADÉMICA DE INGENIERÍA**

**APLICACIÓN DE ALGORITMO GENÉTICO PARA OPTIMIZAR EL  
MODELO DE NILS BOYSEN Y SIMON EMDE CON BASE EN LA  
DETERMINACIÓN DEL NÚMERO DE SUPERMERCADOS EN UN  
SISTEMA DE PRODUCCIÓN PULL**

**T E S I S**

**QUE PARA OBTENER EL GRADO DE MAESTRO  
EN CIENCIAS EN INGENIERÍA INDUSTRIAL**

**P R E S E N T A**

**José Roberto Valencia Vera**

Director: Juan Carlos Seck Tuoh Mora  
Co Director: Joselito Medina Marín

**Cd. del Conocimiento, Mineral de la Reforma, Hidalgo, México. Agosto 2016**



**ACTA DE REVISIÓN Y AUTORIZACIÓN DE IMPRESIÓN**

Documento oficial expedido por el Área Académica de Ingeniería ICBI/AAI/3109/2016

**M. EN A. JULIO CÉSAR LEINES MEDÉCIGO**  
**DIRECTOR DE ADMINISTRACIÓN ESCOLAR**  
**P R E S E N T E**

*Por este conducto le comunico que el jurado asignado al pasante, de la Maestría en Ciencias en Ingeniería Industrial, C. José Roberto Valencia Vera, con número de cuenta Q04065, quien presenta el trabajo de tesis titulado “Aplicación de Algoritmo Genético para Optimizar el Modelo de Nils Boysen y Simon Emde con base en la Determinación del Número de Supermercados en un Sistema de Producción Pull”, después de revisar el trabajo en reunión de sinodales, ha decidido autorizar la impresión del mismo una vez realizadas las correcciones que fueron acordadas.*

*A continuación, se anotan las firmas de conformidad de los integrantes del jurado.*

PRESIDENTE: Dr. Norberto Hernández Romero  
 SECRETARIO: Dr. Joselito Medina Marín  
 VOCAL: Dr. Juan Carlos Seck Tuoh Mora  
 SUPLENTE: Dr. Héctor Rivera Gómez

*Sin otro particular, reitero a Usted la seguridad de mi atenta consideración.*

**A T E N T A M E N T E**  
**“AMOR, ORDEN Y PROGRESO”**  
 MINERAL DE LA REFORMA, HGO., A 18 DE OCTUBRE DE 2016.

**DR. ORLANDO ÁVILA POZOS**  
**DIRECTOR DEL ICBI**



Ciudad del Conocimiento  
 Carretera Pachuca - Tulancingo km. 4.5  
 Colonia Carboneras  
 Mineral de la Reforma, Hidalgo, México, C.P. 42184  
 Tel. +52 771 7172000 exts. 4000 y 4001, Fax 2109  
 aai\_icbi@uaeh.edu.mx

## ÍNDICE GENERAL

RESUMEN .....	vi
ABSTRACT.....	vii
INTRODUCCIÓN .....	viii
CAPÍTULO 1. PROPÓSITO Y ORGANIZACIÓN .....	1
1.1 Planteamiento del problema.....	1
1.2 Propósito de la investigación .....	11
1.3 Objetivo general.....	13
1.4 Objetivos específicos. ....	14
1.5 Justificación de la investigación .....	15
1.6 Viabilidad.....	16
1.7 Delimitación.....	17
1.8 Organización del estudio .....	18
CAPÍTULO 2. MARCO TEÓRICO.....	20
2.1 Los Sistemas de Suministro de Materiales .....	20
2.1.1 Surgimiento de sistemas JIT y Kan-Ban.....	22
2.1.2 Sistema de supermercado.....	23
2.2. Métodos de Optimización de Modelos Matemáticos .....	25
2.2.1 Los Algoritmos Genéticos .....	26
2.2.2 Clasificación general de los Algoritmos Genéticos .....	30
2.2.3 Operadores de los Algoritmos Genéticos .....	31
CAPÍTULO 3. METODOLOGÍA .....	34
3.1 Descripción del objeto de estudio.....	34
3.2 Diseño teórico de la propuesta.....	36
3.2.1 Problema de Ubicación del Supermercado .....	40
3.2.2 Procedimiento de la solución .....	41
3.3 Aplicación de la propuesta.....	42
3.3.1 Resolución en MATLAB de los modelos de los autores.....	44
3.3.2 Propuesta y resolución en MATLAB del algoritmo genético para la búsqueda de la solución óptima del modelo de los autores .....	51
3.4 Obtención de datos para la simulación de resultados .....	59
CAPÍTULO 4. ANÁLISIS Y DISCUSIÓN .....	62
4.1 Resultados.....	62

4.2 Análisis .....	67
4.3 Discusión .....	68
CAPÍTULO 5. CONCLUSIONES .....	69
5.1 Conclusiones relativas a los objetivos específicos y al general.....	69
5.2 Aportaciones originales .....	72
5.3 Límites del modelo planteado.....	73
5.4 Recomendaciones para futuros estudios .....	73
Bibliografía .....	75

### ÍNDICE DE FIGURAS

Figura 1. Impacto de los diferentes métodos de surtimiento de las piezas en la frontera de la línea. ....	22
Figura 2. Un concepto de piso en la fábrica con dos supermercados .....	24
Figura 3. Resultados del ejemplo resuelto por los autores .....	46
Figura 4. Programa para la resolución de la ecuación 3 en MATLAB .....	47
Figura 5. Todos los posibles resultados del programa para la resolución de la ecuación 3 en MATLAB del ejemplo de los autores.....	48
Figura 6. Programa para la resolución de la ecuación 1 en MATLAB .....	48
Figura 7. Ejemplo de introducción de la ruta solución del problema en MATLAB .	49
Figura 8. Programa para la resolución de la ecuación 4 en MATLAB .....	49
Figura 9. Todos los posibles resultados del modelo para la resolución de la ecuación 4 en MATLAB del ejemplo de los autores .....	50
Figura 10. Ejemplo de elaboración de matriz de vectores ruta aleatorios .....	52
Figura 11. Ejemplo de elaboración de matriz de vectores ruta aleatorios .....	53
Figura 12. Ejemplo de elaboración de matriz de cruce para los vectores ruta aleatorios óptimos .....	53
Figura 13. Ejemplo de cruce para los vectores ruta aleatorios óptimos con valores repetidos en el cruce .....	54
Figura 14. Ejemplo de mutación para los vectores ruta aleatorios óptimos .....	55
Figura 15. Ejemplo de la nueva matriz mutada para los vectores ruta aleatorios óptimos.....	56
Figura 16. Programa para la generación del algoritmo genético MATLAB .....	58
Figura 17. Programa para la generación del ciclo del algoritmo genético MATLAB	58
Figura 18. Comportamiento de resolución del problema con Algoritmos Genéticos	59
Figura 19. Programa para resolver ejemplos de los autores en MATLAB.....	61

## ÍNDICE DE GRÁFICAS

Gráfica 1. Resultados con “Sistemas de Supermercado” en la base de datos de Le Servier de la Universidad Autónoma del Estado de Hidalgo .....	5
Gráfica 2. Resultados con “Justo a Tiempo” en la base de datos de Le Servier de la Universidad Autónoma del Estado de Hidalgo.....	5
Gráfica 3. Resultados con “Kan-Ban” en la base de datos de Le Servier de la Universidad Autónoma del Estado de Hidalgo.....	6
Gráfica 4. Resultados con “Justo a Tiempo” en Springer Link de la Universidad Autónoma del Estado de Hidalgo .....	7
Gráfica 5. Resultados con “Kan-Ban” en Springer Link de la Universidad Autónoma del Estado de Hidalgo .....	8
Gráfica 6. Resultados con “Sistema de Supermercado” en Springer Link de la Universidad Autónoma del Estado de Hidalgo.....	8
Gráfica 7. Tendencia de artículos publicados del 2005 al 2014 con temas referentes a “Kan-Ban-Supermercado” y “JIT Supermercado”, su promedio y tendencia.....	10
Gráfica 8. Comparación de Resolución de la Simulación .....	66
Gráfica 9. Diferencia en el área de solución óptima de la solución propuesta contra la de los autores. ....	70

## ÍNDICE DE TABLAS

Tabla 1. Resultados de búsquedas cruzadas en la base de datos de la Universidad Autónoma del Estado de Hidalgo .....	9
Tabla 2. Estructura porcentual de las unidades económicas, personal ocupado total y producción bruta, según la integración en empresas o en establecimientos por estratos de personal ocupado .....	12
Tabla 3. Notación matemática del problema .....	39
Tabla 4. Notación del programa de MATLAB.....	43
Tabla 5. Datos del ejemplo de los autores .....	45
Tabla 6. Resultados de los autores.....	63
Tabla 7. Resultados del trabajo de simulación .....	64
Tabla 8. Resultados del trabajo y comparación al de los autores .....	67

## RESUMEN

La idea de producir las unidades necesarias en las cantidades justas en el momento exacto se vuelve algo ineludible en las empresas en la actualidad para lograr sobresalir ante sus competidores y mantenerse en el mercado. El uso de sistemas de “Supermercado” para el surtimiento de materiales en la planta, es una evolución del sistema Kan Ban como una de las herramientas para lograr el Justo a Tiempo (JIT) de la producción.

Este trabajo presenta una resolución del primero de los modelos matemáticos de los estudios de Nils Boysen y Simon Emde en los temas de aplicación para surtimiento de materiales por medio de Supermercados resolviendo sus funciones matemáticas por medio de la resolución por medio del método de Algoritmos Genéticos inspirados en los binarios con el objetivo de comprobar que se pueden llegar a resultados más óptimos, donde para las simulaciones se utilizan 40 individuos, usando un porcentaje de mutación del 5%, con 10 repeticiones de 100 iteraciones cada una. La parte novedosa del trabajo es que tiene características que lo hacen propio y único en su resolución y simulación, como lo son que la ruta tiene que ir en orden ascendente y no puede repetir números, el valor inicial de la ruta siempre debe de ser 1 y el valor final del vector siempre debe de ser igual al número de estaciones que se tienen en el problema. Estas circunstancias hacen que la simulación se torne interesante y a su vez la complique en cuestión del tiempo de resolución como se vió en el estudio.

Los resultados obtenidos en este trabajo presentan dificultades en el proceso de simulación de dichas resoluciones al ser resueltos en MATLAB debido a que se dificulta la programación y el tiempo de simulación es 600% más, además de que los resultados obtenidos no son favorables por casi \$100,000 a diferencia que los resultados propuestos por los autores y que presentan en sus escritos.

## ABSTRACT

Nowadays the idea of producing the necessary units in the precise quantities at the exact time becomes something inevitable in business to stand before their competitors and stay on the market. The use of systems "Supermarket" for supplying materials on the workshop is an evolution of Kan Ban system as one of the tools to achieve JIT production.

This paper presents a resolution of the first mathematical models of studies Nils Boysen and Simon Emde on implementation issues for the supply of materials through Supermarkets systems, solving their math functions through a resolution by the method of genetic algorithms inspired binaries models in order to verify and reach optimal results, 40 individuals for simulations are used, using a mutation rate of 5%, with 10 repetitions of 100 iterations each. The original part of this paper is that it has features that make it unique in resolution and simulation, for example the route must be in ascending order and cannot repeat any numbers, the initial value of the route must always be 1 and the final value of the vector should always be equal to the number of stations that have the problem. These circumstances make the simulation becomes interesting and complicated in matter of time for the resolution.

The results obtained in this paper have difficulties in the simulation of such resolutions to be solved in MATLAB because the programming is difficult and the simulation time is 600% more, besides the results are not favorable for almost \$100,000 in comparison with the results proposed by the authors presented in their writings.

## INTRODUCCIÓN

El objetivo de cualquier empresa del sector privado es aumentar la productividad en menor costo, para lograr esto, se están buscando nuevas tecnologías y filosofías de aplicación basados en sistemas ya existentes tales como el de Justo a Tiempo (JIT) y Kan-Ban. Estos sistemas tienen como objetivo mantener los flujos de trabajo en movimiento todo el tiempo, desde la recepción de la materia prima en la planta de producción, el almacenaje y flujo de materiales durante su proceso, hasta la entrega final del producto al cliente, eliminando inventarios en proceso, inventarios de producto terminado, reduciendo las distancias de viaje, eliminando defectos y desperdicios, y utilizando al máximo los valores del espacio en la planta.

En las últimas décadas la industria automotriz japonesa ha sido tomada como referencia para sus competidores con el Sistema de Producción Toyota o *Toyota Production System* (TPS), las empresas del occidente han desarrollado la filosofía de producción japonesa y le han llamado Lean Manufacturing (Shah R, 2003), el JIT se encuentra aplicada dentro de estas filosofías y dentro de esta técnica se encuentra el sistema Kan Ban como principal herramienta para una fácil aplicación del surtimiento de los materiales.

La mayoría de los investigadores destacan las ventajas que Lean Manufacturing ofrece, solo Cusumano (Cusumano D, 1994) en su trabajo titulado “Limits of Lean Manufacturing” menciona algunas desventajas derivadas de la aplicación de esta filosofía de producción. Una de las desventajas más importantes según Cusumano, es producir lotes muy pequeños de todos los productos de manera frecuente lo que resulta muy complicado, ya que se requiere de gran coordinación de los proveedores y el sistema de producción, para realizar entregas frecuentes, a los diferentes clientes, en la mezcla y cantidades de todos los productos que se ofrecen, en el momento exacto que el cliente lo solicita.

Si aunado al problema de surtimiento de materiales, cambios rápidos de modelos en la producción y toda la coordinación que esto implica, se agrega el que no se tiene una implementación bien establecida y aplicada, donde se sabe que cada empresa con el objetivo de reducción de faltantes en la línea de producción tiene una manera diferente de aplicar el sistema de surtimiento de materiales por medio de JIT y Kan Ban basándose meramente en la experiencia y conocimientos empíricos de dicha implementación, el problema se amplifica ya que no hay fundamentos bien establecidos y matemáticamente probados que nos digan cual es la mejor fórmula a seguir para poder lograr el mejor resultado sobre lo esperado. Esto sin estar deduciendo que sus implementaciones estén totalmente mal aplicadas, sino que pueden mejorar sus derivaciones al cambiar de enfoque de implementación.

Los sistemas Kan-Ban se tratan de un sistema informativo que armoniosamente controla la producción necesaria de los productos en las cantidades necesarias y en el tiempo necesario, en cada proceso (Monden Y, 1983). El sistema Kan-Ban es una de las formas o elementos para gestionar el método de producción JIT. Sin embargo, a veces el montaje en las líneas tienen un proceso muy largo y la distancia desde el almacén hasta las estaciones es demasiado larga. En este caso, se utiliza otra estrategia en la que hay áreas de inventario dispersas, que se utilizan para alimentar las estaciones cerca de ellas. Por lo general, en la práctica, por cada 20 a 30 estaciones u operaciones se suministran por medio de un almacén descentralizado denominado “Supermercado” (Emde S, Optimally locating in-house logistics areas to facilitate JIT-supply of mixed-model assembly lines, 2012).

El concepto de surtimiento de materiales por medio de supermercados surge en la empresa automotriz de Toyota, modelo inspirador y arquetipo del método Kan-Ban. La idea del supermercado surge gracias a Taiichi Ohno, quien con su famosa descripción de su inspiración para Kan-Ban al regresar de una visita de Estados Unidos durante la década de 1950, en la que él queda más impresionado con el tipo de manejo de materiales que se llevaba en las tiendas de autoservicio de

supermercados, que con los modelos de manufactura de las empresas de Estados Unidos (Towill Denis R, 2010). La idea fue de tener todos los bienes disponibles en todo momento, Ohno (Ohno T, 1988) dice:

“Dentro del supermercado se nos ocurrió la idea de ver el proceso de la tienda como un tipo de línea de producción. El proceso posterior (cliente) va al proceso anterior (supermercado) para adquirir las piezas necesarias (productos) en el momento y en la cantidad necesaria. El proceso anterior inmediatamente produce la cantidad que acaba de tomar (re-almacenamiento de las estanterías)” (p. 26).

El presente trabajo explica la evolución de los sistemas de surtimiento de materiales en las empresas hasta llegar al sistema de surtimiento de supermercados y como es que las empresas aplican estas metodologías de una manera más empírica que bien documentada. Explica además lo referente a la aplicación básica de este concepto de supermercados en las líneas de ensamble de las empresas, pero implementado desde una manera íntegramente diferente: una visión matemática, resolviendo los modelos que proponen Nils Boysen y Simon Emde en su artículo denominado *Optimally locating in-house logistics areas to facilitate JIT-supply of mixed-model assembly lines* en 2012, que es donde inician su investigación con respecto a modelación matemática para el sistema de surtimiento de supermercados, el cual trata el tema del número óptimo de supermercados a implementar, solución que conlleva también a saber su ubicación en la nave y las estaciones que va a suministrar. Este artículo es la fuente principal de donde parte este trabajo y es donde surge la idea de llegar al objetivo de mejorar sus resultados publicados.

Los autores en su artículo resuelven sus funciones por medio de programación dinámica, en este trabajo se pretende comprobar que con la resolución por medio del método de Algoritmos Genéticos se puede llegar a resultados más

óptimos. Se explican los métodos de resolución actuales más importantes y por qué se eligió este método de solución para este modelo matemático.

Se detalla paso a paso como se resuelven con este método de aplicación, como es que fueron traducidos al lenguaje de programación en el software MATLAB y al final se comparan los resultados publicados con los autores con los de este trabajo para determinar qué método de resolución es el más óptimo.

## CAPÍTULO 1. PROPÓSITO Y ORGANIZACIÓN

### 1.1 Planteamiento del problema

Actualmente las filosofías de TPS y Lean Manufacturing proporcionan ventajas competitivas a las empresas que las aplican, sin embargo, no todas las empresas pueden obtener los beneficios potenciales que esta ofrece, por ejemplo en Estados Unidos solo 5% la han implementado correctamente (Ramesh VN, 2010).

Tal cual lo explica José Alfredo Jiménez García en su investigación “Análisis de la Reducción de Pérdidas por falta de Surtimiento Mediante Métodos Heurísticos y De Optimización En Una Planta Manufacturera Caso GKN” (García, 2013) se busca presentar la solución al problema de pérdidas por falta de surtimiento, que atiende una problemática de las empresas que aplican la filosofía de producción Lean Manufacturing y en donde se encuentra que en la literatura especializada, no se encontró un trabajo donde se plantee un problema similar al que aquí se trata, sin embargo, se encontraron algunos temas con cierta relación que fueron clasificados como JIT y Kan-Ban.

En este sentido, lean Manufacturing, se resume en una filosofía que intenta reducir siete tipos básicos de desperdicios como son inventarios, sobreproducción, transportes innecesarios, espera, productos defectuosos, sobre procesamiento y movimientos innecesarios (Padilla, 2010). En esta tesis se pone énfasis en el desperdicio espera que es provocado por la falta de surtimiento de materiales, lo que ocasiona el ocio a las máquinas y operarios de las células de producción, así como una afectación a la eficiencia global del equipo (OEE, por sus siglas en inglés). De acuerdo con Villaseñor y Galindo (Galindo, 2007), el porcentaje de actividades que agregan valor al producto solo es del 5%, el 95% restante es desperdicio y dentro de ese 95%, 15% se debe a las esperas.

En un sistema de manufactura un tema esencial para lograr la eliminación de todo tipo de desperdicio (entre ellos la espera), es el flujo de materiales, por lo que se considera que el objetivo más importante de un sistema de manufactura es el flujo continuo de los materiales. El problema más frecuente para lograr el flujo continuo, es la falta de un sistema de manejo de materiales que soporte el flujo continuo en las células y la fabricación de lotes pequeños (Wilson, 2003). Generalmente, los esfuerzos se enfocan hacia la reducción del trabajo en proceso, como idea de que al eliminarlo, se resolverá un gran porcentaje de los desperdicios. Kim (S Kim, 2008), en su trabajo, definieron políticas de control de la producción con un WIP equilibrado, considerando los cambios de modelo mediante el planteamiento de un modelo de programación entera mixta y una heurística. Un comentario de elevada importancia para el contexto de esta investigación es: “escaso WIP y frecuentes cambios de modelo, puede provocar ocio en los equipos”.

Ali y su equipo propusieron la estandarización del método de surtimiento mediante tarjetas Kan Ban, con la finalidad de reducir el trabajo en proceso. Sugiere la aplicación de la herramienta simulación, para facilitar su implementación. Villanueva (Villanueva, 2008), destaca la importancia de la simulación y sugiere algunos pasos en la construcción de modelos usados en la solución de problemas del mundo real. Comenta las ventajas del uso de la simulación para optimizar el flujo de materiales dentro de un sistema de eventos discretos, así como las desventajas de la herramienta al requerir un tiempo prolongado para construir los modelos de simulación. Standridge y Marvel (Marvel, 2006), realizaron varios proyectos relacionados con Lean Manufacturing y mencionan lo importante que fue la utilización de la herramienta de simulación para el éxito de los proyectos. Abdulmalek y Rajgopal (Rajgopal, 2006) demostraron los beneficios que se pueden lograr implementando la filosofía Lean Manufacturing usando herramientas tales como simulación y el mapeo de la cadena de valor.

Por otra parte, otros autores intentan resolver problemas relacionados con los inventarios, tal es el caso del problema para la localización de depósitos de ruteo, determinación de las rutas de distribución y de las políticas de inventario que plantea Liu y Lin (SC Liu, 2005).

En su planteamiento se tiene un problema difícil de resolver, por lo que recurren a una heurística propuesta por ellos, para su solución. Las heurísticas y meta heurísticas se han utilizado ampliamente en problemas del mundo real, los cuales por su magnitud, resultan muy difíciles de resolver. Las heurísticas se han aplicado a problemas muy específicos y las meta heurísticas se aplican a problemas más generalizados (A Duarte-Muñoz, 2007).

Nauss (Nauss, 2003), aplicó un método de ramificación y acotamiento para la solución del problema de asignación generalizada, que entre sus aplicaciones se encuentra la planificación de rutas de entrega. El método de ramificación es menos flexible que las heurísticas y meta heurísticas, sin embargo es capaz de encontrar soluciones óptimas, contrario a las heurísticas y meta heurísticas que solo encuentran soluciones de muy buena calidad, en un tiempo relativamente corto. En este sentido se convierte en un buen tema de decisión el aplicar un método exacto o una heurística, todo depende de la calidad de los resultados que se desean tener y que el tiempo sea razonable para encontrar la solución.

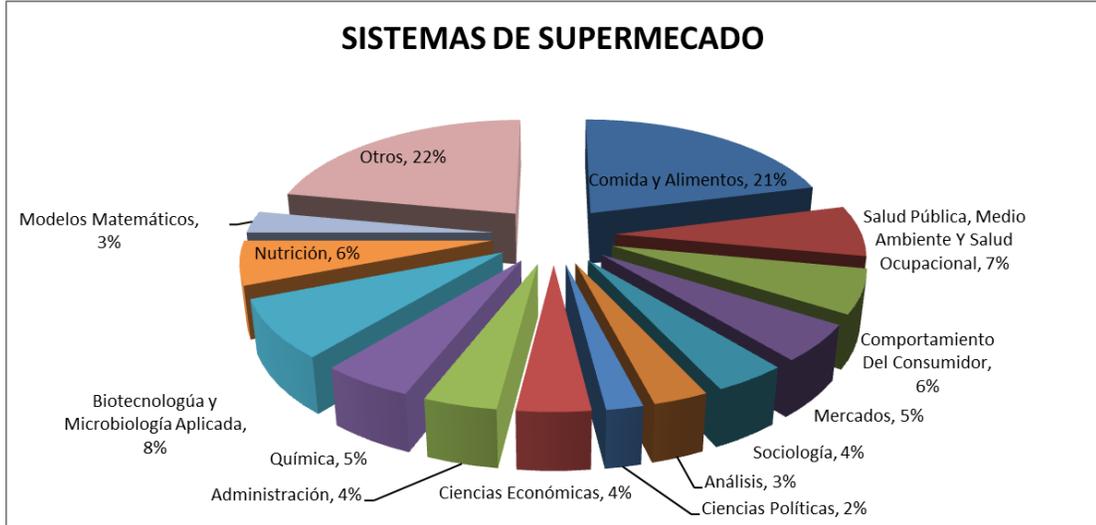
No se encontró un trabajo donde se plantee el problema que se trata en esta tesis, sin embargo, por sus características, tiene cierta relación con el problema de implementación de sistemas de Kan Ban y Justo a Tiempo JIT. A continuación se da una breve reseña de la búsqueda de información con la que se concluye que existe muy poca información validada académicamente que avale a los sistemas de implementación de supermercado:

Se inicia una búsqueda de manera generalizada en toda la base de datos de la Universidad Autónoma del Estado de Hidalgo, donde se tienen varias de las revistas y páginas web más importantes de investigación para áreas de ingeniería, para trabajar en este proyecto se toma a *Springer Link*. Se continua con una exploración de los mismos temas de información pero ahora en la página del Consorcio Nacional de Recursos de Información Científica y Tecnológica (CONRyCIT) perteneciente al Consejo Nacional de Ciencia y Tecnología (CONACyT) de México.

Para analizar la búsqueda de información respecto a los temas claves, se inicia la investigación para el término de “Sistemas de Supermercado” en el buscador del CONRyCIT, donde encontramos que en total existen 41,475 artículos, de los cuales la mayoría da referencias a temas de comidas y alimentos, ciencias económicas, temas de salud y medio ambiente (Gráfica 1), todo se relaciona con las tiendas de Supermercado y no con los sistemas de surtimiento de Supermercado que es lo que se está tratando de encontrar.

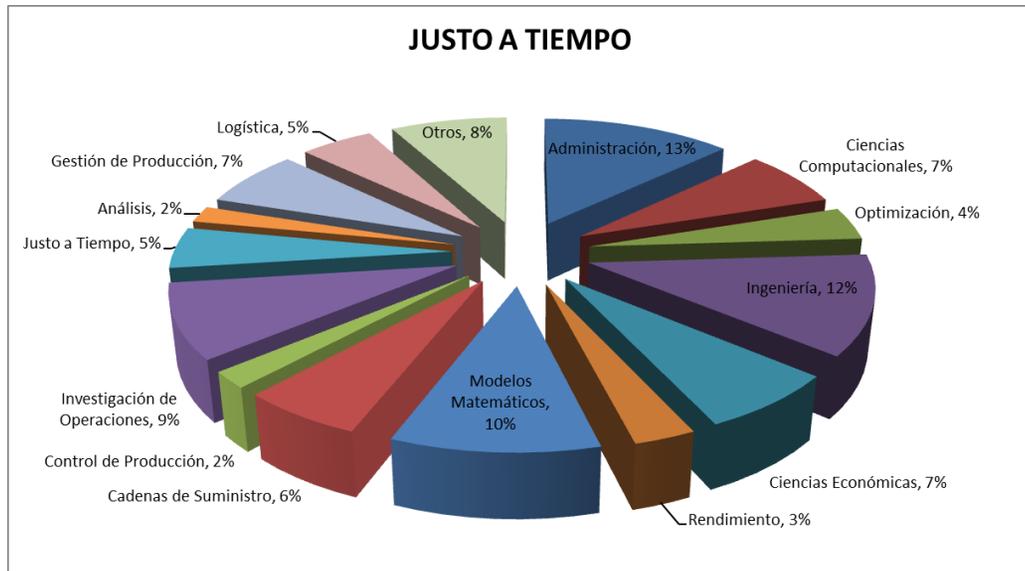
Existen en total 125,963 artículos referentes al término “Justo a Tiempo”, de los cuales principalmente estudian temas referentes a administración, ingeniería, y modelos matemáticos (Gráfica 2) se encuentran la mayoría dentro de la clasificación de resolución por métodos computaciones pero no hay referencias con sistemas de supermercados.

Para el término de “Kan-Ban”, en total ajustan 25,738 resultados, donde dominan principalmente temas de Ingeniería y fabricación, control de producción y administración (Gráfica 3) enfocados mayormente en sistemas matemáticos de implementación.



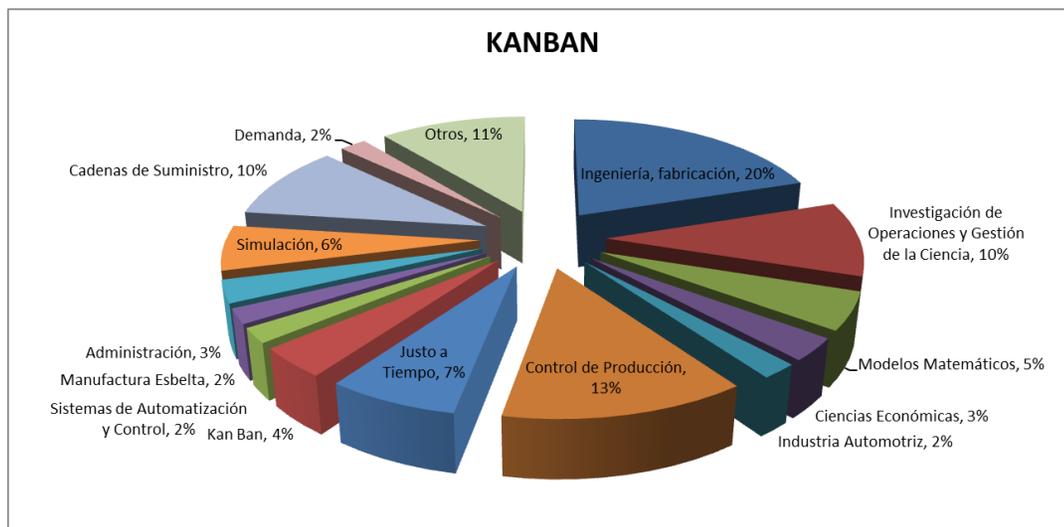
**Gráfica 1. Resultados con “Sistemas de Supermercado” en la base de datos de Le Servier de la Universidad Autónoma del Estado de Hidalgo**

*Fuente: Elaboración propia*



**Gráfica 2. Resultados con “Justo a Tiempo” en la base de datos de Le Servier de la Universidad Autónoma del Estado de Hidalgo**

*Fuente: Elaboración propia*



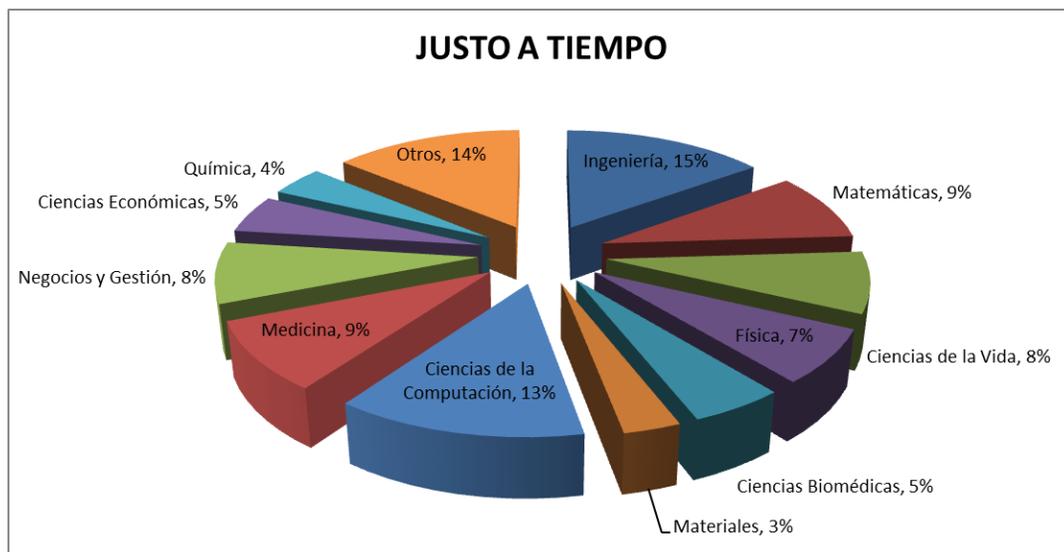
**Gráfica 3. Resultados con “Kan-Ban” en la base de datos de Le Servier de la Universidad Autónoma del Estado de Hidalgo**

*Fuente: Elaboración propia*

Se repite este proceso de búsqueda y se combinan los términos para hacer una búsqueda cruzada más específica, poniendo en común la palabra de supermercado, con la finalidad de afinar los resultados. La búsqueda arroja que para los términos de “Justo a Tiempo + Supermercado” existen 706 artículos y para “Kan-Ban + Supermercado” son arrojados 88 resultados, la mayoría relacionados con el tema de implementación de sistemas de surtimiento de materiales en general y una mínima parte con el tema que se desea investigar. Comparando estos resultados con los totales anteriores, concluimos de manera fehaciente que sólo el 0.5% en promedio tocan temas referentes a supermercados involucrados con sistemas de Kan-Ban y JIT, siendo una minoría considerable a todos los artículos en la red.

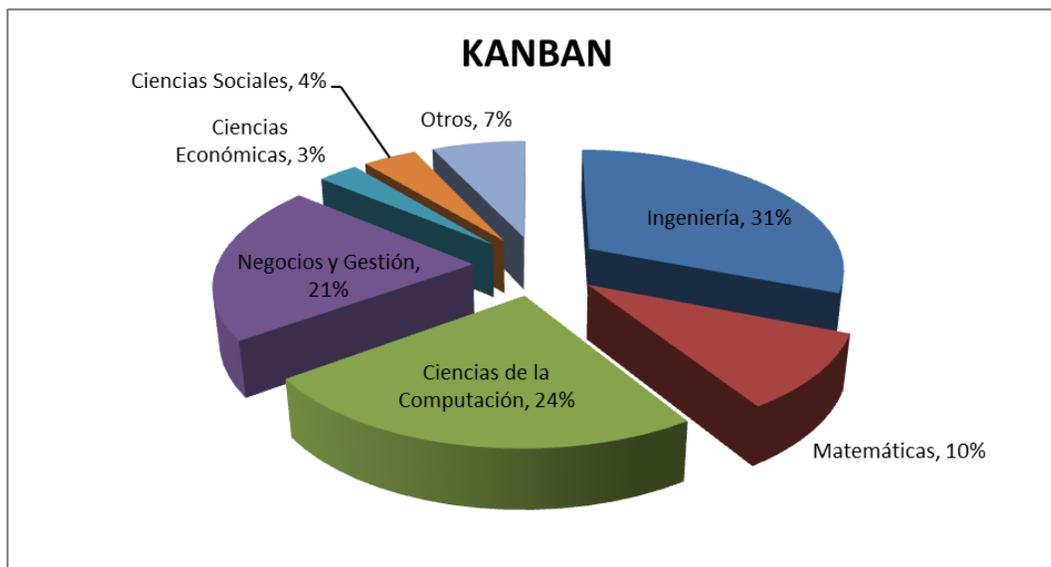
Además de los resultados obtenidos anteriormente, se hace una nueva búsqueda con el mismo procedimiento pero en otro buscador, con la finalidad de comparar los artículos de una manera más clara, específica y concreta. Se cita el ejemplo de la base de datos de la página web de *Springer Link* (Springer Link, 2013), donde al buscar artículos (todos en inglés) con temas referentes a la filosofía de

“Justo a Tiempo” se encuentran 2923 resultados, donde cerca del 40% son temas referentes a ingeniería, matemáticas y computación referente a simulación (Gráfica 4). Con “Kan-Ban” se encuentran cerca de 388 resultados donde el 55% tratan temas referentes a matemáticas y computación referente a simulación (Gráfica 5). Con “Sistemas de supermercado” 5785 resultados, de los cuales se encuentran principalmente los referentes a los temas de ciencias sociales con más de un 50% (Gráfica 6).



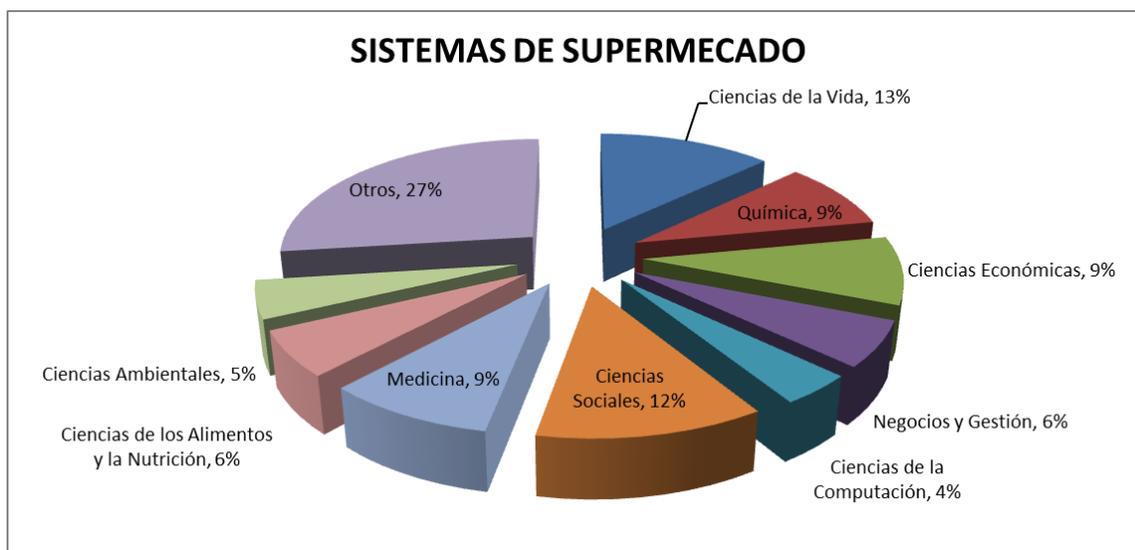
**Gráfica 4. Resultados con “Justo a Tiempo” en Springer Link de la Universidad Autónoma del Estado de Hidalgo**

*Fuente: Elaboración propia*



**Gráfica 5. Resultados con “Kan-Ban” en Springer Link de la Universidad Autónoma del Estado de Hidalgo**

*Fuente: Elaboración propia*



**Gráfica 6. Resultados con “Sistema de Supermercado” en Springer Link de la Universidad Autónoma del Estado de Hidalgo**

*Fuente: Elaboración propia*

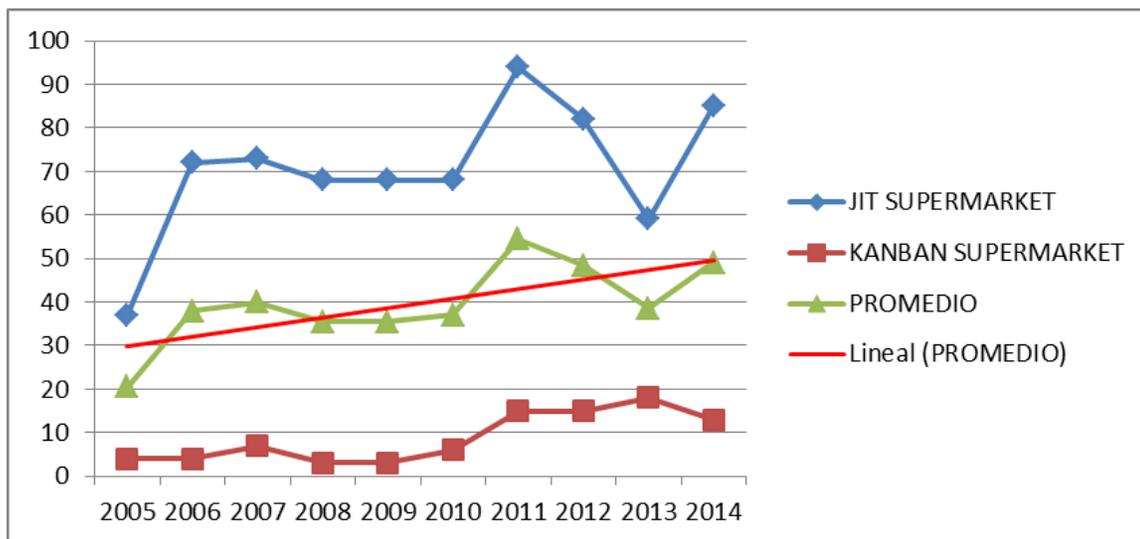
Igual que en la búsqueda pasada, se hace una búsqueda cruzada de dichos términos poniendo en común la palabra de supermercado, ahora la búsqueda muestra que para “Justo a Tiempo + Supermercado” se tienen 34 resultados, de los cuales 22 son descartados ya que toman temas sociales y menos aplicativos en la industria, dejando un total de 12 artículos para revisar. Para “Kan-Ban + Supermercado” son arrojados 19 resultados de los cuales cinco son descartados al tocar temas que no profundizan con el tema central de búsqueda, quedándonos sólo con 14 artículos para revisar.

Analizando las fechas de publicación de dicho grupo de artículos del 2005 hasta 2014 (Tabla 1), encontramos que se publican en promedio 40 artículos por año, donde se muestra una tendencia positiva pero poco evolutiva, siendo de menos de dos artículos por año (Gráfica 7).

**Tabla 1. Resultados de búsquedas cruzadas en la base de datos de la Universidad Autónoma del Estado de Hidalgo**

<b>AÑO</b>	<b>JIT SUPERMARKET</b>	<b>KANBAN SUPERMARKET</b>	<b>PROMEDIO</b>
2005	37	4	20.5
2006	72	4	38
2007	73	7	40
2008	68	3	35.5
2009	68	3	35.5
2010	68	6	37
2011	94	15	54.5
2012	82	15	48.5
2013	59	18	38.5
2014	85	13	49
<b>SUMA</b>	<b>706</b>	<b>88</b>	<b>397</b>
<b>PROMEDIO</b>	<b>70.6</b>	<b>8.8</b>	<b>39.7</b>

*Fuente: Elaboración propia*



**Gráfica 7. Tendencia de artículos publicados del 2005 al 2014 con temas referentes a “Kan-Ban-Supermercado” y “JIT Supermercado”, su promedio y tendencia**

*Fuente: Elaboración propia*

De los últimos resultados se hace un comparativo y se encuentra que en relación a estas dos últimas búsquedas donde existen diez artículos en común, de los cuales tres artículos coinciden y son de los autores que se están estudiando: Nils Boysen, Simon Emde. Con esto podemos concluir que la investigación referente a supermercados aplicados al JIT y Kan-Ban con métodos matemáticos ha sido mayormente realizado por el mismo grupo de autores. Esto demuestra que se ha realizado poca información en materia de investigación en éste tema, siendo ésta una de las principales razones que motivan a este trabajo.

Si bien actualmente no hay alguna regla, norma o procedimiento que rijan como hacer un sistema de supermercado, actualmente las empresas que tienen esta técnica en funcionamiento mencionan que es en base a sus experiencias con la aplicación de JIT y Kan-Ban acorde a las necesidades y espacio disponible de la planta. Cada supermercado es y se aplica de manera diferente y va a depender totalmente de la experiencia y conocimientos de quien lo aplique.

Hasta estos días se han iniciado y se pueden encontrar estudios de investigación respecto a estas técnicas, dando como resultado ya no una metodología totalmente empírica y experimental sino algo más formal mediante modelos cuantitativos resueltos a través de diversos métodos y algoritmos los cuales busquen la optimización de su implementación.

Como se menciona anteriormente, este trabajo propone una nueva metodología de solución para la optimización del modelo propuesto por los autores basada en Algoritmos Genéticos para definir el número de supermercados a implementar en un sistema productivo, todo con base en información totalmente de la ubicación de las estaciones por coordenadas, la demanda de cada una de éstas y el precio de implementación fijo por supermercado.

## **1.2 Propósito de la investigación**

Actualmente no existe información respaldada que diga qué tipo de surtimiento de materiales utilizan las empresas en México, pero se puede deducir por el número de empleados de las empresas, especulando que las de menor tamaño por su número de empleados son las que menos se preocupan por la implementación de sistemas de control para el surtimiento de materiales y las de mayor número de empleados son las empresas que tienen implementada una manufactura de clase mundial, donde entran los sistemas de surtimiento más completos.

En México el mayor porcentaje lo representan las micro, pequeñas y medianas empresas siendo del 99.8%, según los datos que nos arroja el estudio realizado por el Instituto Nacional de Estadística y Geografía (INEGI), en su último censo realizado en el año del 2009 (Tabla 2):

**Tabla 2. Estructura porcentual de las unidades económicas, personal ocupado total y producción bruta, según la integración en empresas o en establecimientos por estratos de personal ocupado**

Estrato de personal ocupado	Unidades económicas		Personal ocupado total		Producción bruta total	
	Integradas por establecimiento	Integradas por empresa	Integrado por establecimiento	Integrado por empresa	Integrada por establecimiento	Integrada por empresa
<b>Total nacional</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>
0 a 10 personas	95.0	95.7	41.8	40.4	8.3	6.9
11 a 50 personas	4.0	3.5	15.3	12.6	9.0	6.4
51 a 250 personas	0.8	0.6	15.9	11.8	17.4	12.8
251 y más personas	0.2	0.2	27.0	35.2	65.3	73.9

*Fuente: Instituto Nacional de Estadística y Geografía (INEGI) 2009*

Veronique Limère en el año de 2013 (Limère V, 2013), da una explicación clara sobre lo que pasa actualmente en las empresas:

Existen demasiados sistemas de suministro de materiales que se encuentran en la práctica, y cada uno de ellos ofrece ciertos beneficios operativos y desventajas. Sin embargo, se ha llevado a cabo poca investigación para profundizar en los diferentes métodos de suministro de materiales y el equilibrio entre ellos. Las decisiones de las empresas se basan principalmente en la intuición y la experiencia pero no existe conocimiento objetivo acerca de las ventajas y desventajas de los diferentes sistemas (p. 35).

Con esta información se concluye que las empresas en México utilizan los sistemas de surtimiento tradicionales más comunes los cuales se basan en sistemas de almacenamiento central, es decir; todo el material que reciben llega a un almacén general donde se acopia y es surtido a la línea de producción mediante la implementación de sistemas básicos de PEPS (Primeras Entradas Primeras Salidas), donde una vez surtidas son almacenados los materiales nuevamente en la línea de ensamble, en cantidades considerablemente grandes y en racks de un tamaño

desmedido. A consecuencia de esto se busca no volver a surtir material en un periodo de tiempo próximo como un día o hasta una semana, dependiendo del tamaño y demanda de las piezas, esto debido a que cada vez más se maneja una creciente variedad de productos, y el tiempo y espacio son cada vez más limitados en la planta.

Este tipo de almacenaje habitual también puede causar estorbo al operador, quita demasiado espacio valioso en el taller y la peor razón es que se pierde el control total del material, ya que tiene muchas más posibilidades de que se pueda dañar o perder injustificadamente. Esto debido a que hay montones de material en la línea causando el descuido de éste con los operadores, provocando algunas veces hasta paros de línea por pérdida de material y mayormente siendo injustificado.

Este trabajo tiene tres principales finalidades: la primera es explicar el surtimiento de materiales de manera generalizada, su evolución hasta llegar al sistema de supermercados y exponer como puede aplicarse en el sector industrial, la segunda es mejorar los resultados de los autores publicados en el artículo arriba mencionado, resolviendo su problema con otro método de simulación y comprobar dichos resultados, la tercera es generalizar ese método y hacer que la aplicación de dicha simulación se haga de una manera sencilla mediante la programación del algoritmo en el software, que sea fácil de entender para poder traducirlo a los problemas reales de las empresas y la última sería aportar información útil en temas donde se tiene poco tiempo e información de estudio en materia de investigación.

### **1.3 Objetivo general**

Diseñar y evaluar el modelo de optimización por medio de Algoritmos Genéticos para resolver las funciones de Nils Boysen y Simon Emde que determinan el número de supermercados en su artículo *Optimally locating in-house logistics areas to facilitate JIT-supply of mixed-model assembly lines (2012)*, con la finalidad de obtener una solución más óptima.

Seguir los lineamientos de los sistemas de producción *Pull* o Jalar que tienen características únicas a la hora de funcionar (Sohal, 1992), pues las máquinas no producen nada hasta que reciban una orden que les solicite que lo hagan, originarias de la demanda del consumidor, de manera que no se generan inventarios innecesarios que quizás al final queden varados y no se vendan, ya que serían solo excedentes de producción sin ningún cliente final asignado todavía, la producción debe de ir hacia un objetivo: satisfacer la demanda.

#### **1.4 Objetivos específicos.**

1. Entender el modelo de Nils Boysen y Simon Emde de su artículo para definir el número de supermercados por medio de su método de solución con programación dinámica.
2. Simular y demostrar el modelo de Nils Boysen y Simon Emde para definir el número de supermercados por medio de una nueva propuesta con Algoritmos Genéticos.
3. Comparar los dos métodos de solución del modelo y validar cual es el que mejor optimiza los resultados, en cuanto a tiempo de resolución, amplitud de variables y resultados.
4. Hacer el modelo más flexible, facilitar y ampliar el método de solución y que se pueda aplicar de una manera fácil y sencilla para generalizar la aplicación de sistemas de supermercado en la industria.

## 1.5 Justificación de la investigación

Con base en los antecedentes anteriores, se identifica un área de oportunidad dentro del área de surtimiento de materiales por medio de supermercados, exponiendo que existe poca información de investigación con fundamento matemático respecto al tema.

Los autores Nils Boysen y Simon Emde profundizan sus investigaciones respecto a esta cuestión, los cuales son elegidos para estudiar este proyecto; teniendo en la actualidad cinco artículos referentes solamente a ello hasta octubre del 2015:

1. *Optimally locating in-house logistics areas to facilitate JIT-supply of mixed-model assembly lines (Simon Emde, Nils Boysen, 2010)*
2. *Optimally routing and scheduling tow trains for JIT-supply of mixed-model assembly lines (Simon Emde, Nils Boysen, 2011)*
3. *Just-in-Time supermarkets for part supplying the automobile industry (Daria Battini, Nils Boysen, Simon Emde, 2012)*
4. *Optimally loading tow trains for just-in-time supply of mixed-model assembly lines (Simon Emde, Malte Fliedner, Nils Boysen, 2013)*
5. *Scheduling the part supply of mixed-model assembly lines in line-integrated supermarkets (Nils Boysen, Simon Emde, 2013)*

Se elegirá por cuestiones de entendimiento para el inicio de estudios de este trabajo, estudiar el tema número uno de los cinco artículos arriba citados, al ser el inicio de la investigación de los autores y principalmente ser el tema para implementar desde cero un supermercado, ya que los demás temas profundizan el

contenido ya sobre temas que relacionan a supermercados ya puestos en marcha en la industria.

A pesar de ser un método de surtimiento conocido y muy sencillo de aplicar, que además se relaciona con sistemas de Kan-Ban y JIT que son temas muy estudiados, con base en la indagación que se hizo anteriormente se concluye que existe poca información en el desarrollo de modelos matemáticos para éste tema, siendo también esta una de las principales razones del por qué estudiar y buscar aportar un método de solución diferente matemáticamente hablando, esperando aportar información que pueda ser utilizada y de apoyo para investigaciones posteriores.

Esto brinda la oportunidad de estudiar un tema poco abordado y transformado por los investigadores, facilitando el poder proponer información y métodos de solución diferentes respecto al tema, con la finalidad de preparar un modelo flexible, fácil de aplicar y resolver. No existe actualmente en la literatura un artículo referente a temas que tomen de base estudios en implementación de sistemas de surtimiento por supermercados y sean resueltos mediante modelos de optimización por Algoritmos Genéticos.

## **1.6 Viabilidad**

En general la investigación tuvo las condiciones necesarias para poderse llevar a cabo. No requirió de procesos especiales o un financiamiento alto, se necesitó material científico del cual basar, apoyar y sostener la investigación, además de soporte humano intelectual y empírico.

Todos los requerimientos en cuanto a material y artículos científicos para sostener la información fueron facilitados como anteriormente se menciona por la base de datos de búsqueda de la Universidad Autónoma del Estado de Hidalgo,

donde se busca información en las páginas prestigiadas para un carácter de ingeniería, como ejemplo: *Springer Link*. Además de la página del Consorcio Nacional de Recursos de Información Científica y Tecnológica (CONRyCIT) perteneciente al Consejo Nacional de Ciencia y Tecnología (CONACyT) de México.

El proyecto no tiene alguna demanda especial para poder en primera instancia simular y demostrar el modelo con el método propuesto por los autores de programación dinámica, para después buscar una posible solución por el método de Algoritmos Genéticos y al final hacer una comparación general de las variables para generar los resultados, los alcances y límites de las variables de entrada.

Se ocupará la programación y simulación por computadora al ser el método más utilizado, rápido y sencillo de resolver en cuanto a problemas de resolución de modelaje y lenguaje matemático. Para comprobar la teoría de este proyecto se utilizará la programación por computadora en el software de MatLab R2012a corriendo a 64 bits, en un equipo de cómputo portátil con un sistema operativo Windows 8 Single Language, con procesador Intel ®, Core™ i7-3630QM CPU @ 2.40 GHz con 8 Gigabytes de memoria RAM y un Terabyte de disco duro.

No se requirió de algún trabajo en campo, alguna inversión extra o de alguna actividad de carácter especial o técnico para la resolución de este trabajo.

## **1.7 Delimitación**

No se pretende ir más allá del primer artículo publicado por los autores que explica de manera muy general la aplicación de los supermercados, si nos ponemos a estudiar más a fondo la implementación de este sistema se requiere de mucho más análisis y estudio, con base en la ubicación de las estaciones por coordenadas, la demanda de cada una de éstas y el precio de implementación fijo por supermercado.

Todos estos conceptos son de básico conocimiento dentro la industria y se pueden calcular fácilmente.

Con este trabajo se busca una aplicación y resolución meramente práctica del modelo de los autores. No se busca una aplicación práctica en una empresa o un caso de estudio en especial, solo la resolución del problema de una manera diferente y generar el algoritmo para que sin importar el número de estaciones, coordenadas y demandas de este algoritmo se pueda aplicar de manera simple con la seguridad de obtener un mejor resultado de la función objetivo.

En general se pretende profundizar dicho modelo y comprobar nuestra hipótesis de verificar que la resolución por Algoritmos Genéticos es mejor que la propuesta por los autores con programación dinámica. Más adelante se muestran las comparaciones de los resultados en forma, no se hará ningún estudio de campo ni se pretenderá poner a prueba de manera física este experimento.

## **1.8 Organización del estudio**

El presente trabajo de investigación está conformado de la siguiente manera:

La primera parte trata de manera general en qué consisten y cuáles son los sistemas de suministro de materiales más comunes en las empresas hoy en día. Explicará brevemente el inicio del sistema JIT y Kan Ban que son la pauta para la aplicación el sistema de Supermercado.

La segunda parte habla de manera generalizada respecto a la solución de problemas por medio de optimización matemática, en este punto se toma el interés e importancia sobre la resolución por medio de Algoritmos Genéticos y por qué fue seleccionada este tipo de resolución para este problema.

La tercera parte da referencia al desarrollo y la problemática y la descripción del objeto de estudio, explicando punto por punto como es la relación de resolución de la implementación de Algoritmos Genéticos con respecto al número óptimo de supermercados a implementar basados en las funciones descritas por Nils Boysen y Simon Emde.

La última parte explica de manera breve la solución a dicho problema con el método propuesto y la comparación que se tiene con los resultados de los autores, las aportaciones y los puntos de estudio que se quedan abiertos para futuras investigaciones.

## **CAPÍTULO 2. MARCO TEÓRICO**

### **2.1 Los Sistemas de Suministro de Materiales**

La principal tarea de un buen método de suministro de materiales es, surtir los materiales adecuados, en el momento adecuado, en el lugar adecuado y en la cantidad exacta a la línea de producción. Con el fin de ser competitivo en la industria, es importante que este proceso de suministro se ejecute de una manera rentable. Sin embargo, no sólo el costo de suministro de las partes debe ser considerado, también el impacto en las operaciones de montaje deben ser consideradas.

Los diferentes sistemas de alimentación de la línea de producción tendrán una influencia en cómo se usarán las partes en la frontera de las estaciones. Por otra parte, es necesario considerar que el espacio disponible en la frontera de la línea que es muy limitado. (Limère V, 2013).

En la industria, se pueden encontrar diferentes sistemas de suministro de materiales, Veronique Limère (2013), explica de manera general los diferentes tipos de surtimiento más utilizados en las industrias:

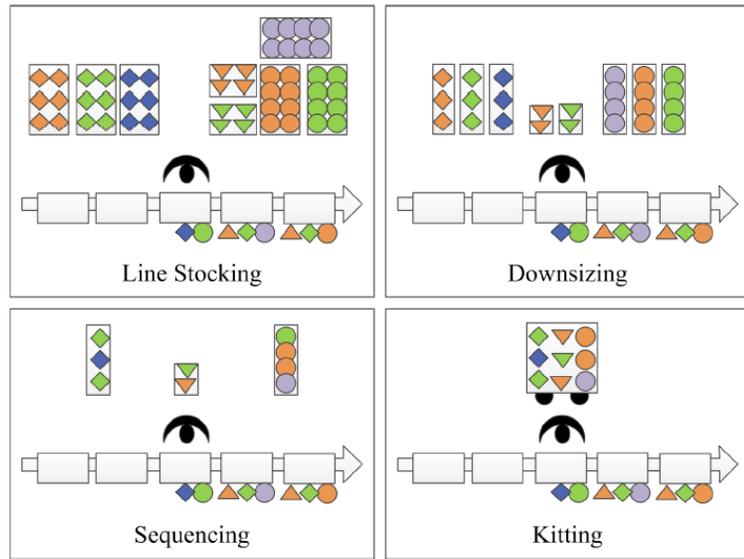
El método más sencillo y común es el de suministro de materiales a granel, almacenamiento general, de reposición continua o de almacenamiento en la línea. Bajo este sistema de almacenamiento, las partes se suministran a la planta de fabricación en grandes cantidades de un solo viaje dentro de un contenedor específico. Los contenedores se almacenan cerca de las estaciones de trabajo en la frontera de la línea, y un sistema de dos cajones o punto de reorden se utiliza para controlar la reposición. Este sistema es el más básico pero el peor para los fines de ahorrar desperdicios de espacio en la línea de producción.

El almacenamiento directo en la línea, causa que las piezas tengan primero que ser re-empacadas en los grandes contenedores que llegan en contenedores más pequeños antes de ser suministrada a la línea. Este sistema de suministro de materiales se denomina “Reducción”.

Un tercer método de alimentación de línea es la “Secuenciación”. La secuenciación de las partes significa que las partes no se almacenan a granel en la frontera de la línea, pero sólo se suministran a la línea en el momento y en la cantidad que se necesitan según el esquema de montaje.

Por último, las piezas también se pueden agrupar en kits antes de que se suministren a la línea. El “Kiteo” suministra una o más operaciones de montaje de un producto en la línea. Sobre todo en un ambiente de modelo mixto de alta varianza cada kit será diferente y los kits serán secuenciados según el programa de producción futuro. La cantidad exacta de los componentes necesarios se almacena en recipientes kit cerca de las estaciones de trabajo de montaje en la frontera de la línea y reposiciones se llevan a cabo de acuerdo con el horario de montaje que se basa en el ciclo de montaje o tiempo de procesamiento. Aquí es donde se da inicio a los sistemas JIT.

Los cuatro modelos arriba mencionados se pueden observar en la figura 1 y son los sistemas de surtimiento más conocidos y usados en las industrias. Como se puede observar el sistema Kitting habla por sí mismo demostrando que es el que menos espacio ocupa, el que más lógica de suministro tiene y es el más flexible a los cambios que puedan surgir en la línea.



**Figura 1. Impacto de los diferentes métodos de surtimiento de las piezas en la frontera de la línea.**

*Fuente: Veronique Limère, 2013. To kit or not to kit: optimizing part feeding in the automotive assembly industry.*

### 2.1.1 Surgimiento de sistemas JIT y Kan-Ban.

JIT se desarrolló por Taiichi Ohno en el TPS, su objetivo principal es reducir continuamente todas las formas de desperdicios en la planta (Brown KA, 1991). JIT señala "el concepto cero", que significa el logro de las metas con cero defectos, cero colas, cero inventarios, cero averías, etc. Garantiza el suministro de piezas adecuadas, en la cantidad adecuada, en el lugar correcto y en el momento adecuado. Por lo tanto, el viejo sistema de adquisición de material y almacenaje central, se cambian a los nuevos conceptos revolucionarios (Womack JP, 1991).

Debido al avance tecnológico, el método convencional del sistema de producción de empuje relacionado con la Planificación de Requerimientos Materiales o *Material Requirement Planning* (MRP) fue cambiado a sistema de producción jalar tipo JIT para satisfacer a la competencia global, donde el trabajo en proceso o

*Work In Process* (WIP) se puede gestionar y controlar con mayor precisión que el sistema de producción de empuje o Kan-Ban (Mason PA, 1999).

El Sistema Kan-Ban es una nueva filosofía, que desempeña un papel importante en el sistema de producción JIT. Kan-Ban es básicamente una tarjeta de plástico que contiene toda la información necesaria para la producción y/o montaje de un producto en cada etapa y los detalles durante su proceso de finalización. El sistema Kan-Ban es una programación de la producción de varias etapas y el sistema de control de inventario. Estas tarjetas se utilizan para controlar el flujo de producción y el inventario. Este sistema facilita el alto volumen de producción y la utilización de alta capacidad con un menor tiempo de producción y el trabajo en proceso (Sendil Kumar C, 2002).

Los supermercados son parte del famoso TPS y tienen una larga tradición en muchas industrias. Sin embargo, la logística interna dentro de este tipo fue antes organizada con sistemas Kan-Ban puros (Vatalaro J, 2005).

### **2.1.2 Sistema de supermercado**

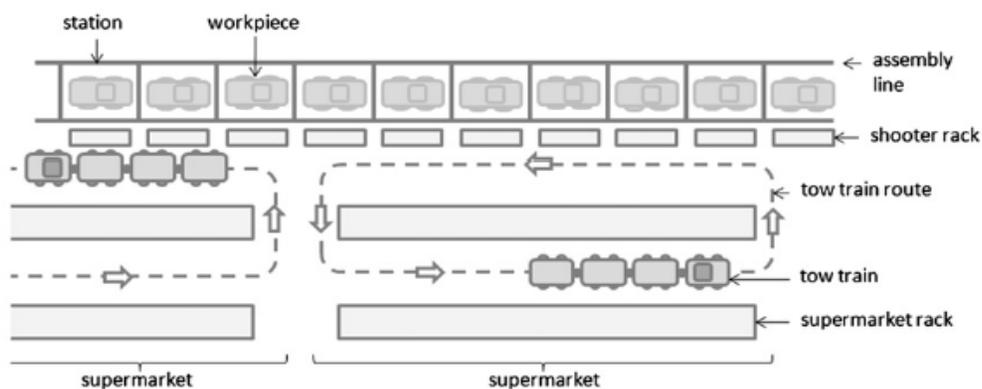
Los supermercados son considerados como áreas de almacenamiento descentralizadas dispersas que actúan como almacén intermedio para alimentar estaciones cercanas. Algunas de las ventajas de estos sistemas es la entrega rápida y frecuente de las piezas y de consolidación de carga al ser suministrada por carritos (Tow Trains). Sin embargo, los supermercados consumen espacio en el piso de la fábrica, que es escaso y caro (Emde S, *Optimally locating in-house logistics areas to facilitate JIT-supply of mixed-model assembly lines*, 2012).

El tipo de almacenaje habitual también puede causar estorbo al operador, quita demasiado espacio valioso en el taller y la peor razón es que se pierde el

control total del material, una de las principales ventajas es evitar los grandes y estorbosos racks de la línea y el almacenamiento excesivo de partes.

Al ver que las secuencias de producción generalmente se conocen con suficiente antelación (entre tres y cuatro días en la industria automotriz), la lista de materiales determina exactamente cuántas partes se necesitan dentro de un determinado intervalo de tiempo en cada estación. Los sistemas clásicos de Kan-Ban según el TPS (Monden Y, 1983) no aprovechan esta información, ya que sólo surten el material que ya ha alcanzado un nivel de crítico stock ya definido. Sin embargo, a la espera de ser "sorprendido" por las bajas existencias, lo que podría requerir de entregas de emergencia si el próximo paro de línea programado en la estación no está demasiado lejos.

Esto es innecesario al conocer la información acerca del programa de la producción y la demanda de partes por estación, así como los plazos de entrega de material. Como consecuencia, conociendo estas condiciones, se puede ejecutar sin problemas un sistema moderno y bien planificado de surtimiento basado en los supermercados (Golz J, 2012).



**Figura 2. Un concepto de piso en la fábrica con dos supermercados**

*Fuente: Simon Emde & Nils Boysen, 2012. Optimally locating in-house logistics areas to facilitate JIT-supply of mixed-model assembly lines*

La Figura 2 tomada de Nils Boysen, Simon Emde (Emde S, *Optimally locating in-house logistics areas to facilitate JIT-supply of mixed-model assembly lines*, 2012), muestra la distribución de una fábrica con dos supermercados. Donde explican que el almacén central o de recibo de materiales distribuye las partes en lotes considerablemente grandes a los supermercados en la línea, donde estos preparan el material para el surtimiento en la línea por medio de carritos o *Tow Trains*, los cuales rodean las estaciones correspondientes al área de surtimiento del supermercado. Estos autores comentan que el proceso delimitado de la distribución de las partes en la estaciones requiere de una ruta definida, una programación de entregas y la carga de materiales a remolcar en los trenes ya asignados al supermercado.

Si bien existen muchas maneras de estudiar y poner en marcha un supermercado en la industria, se tiene que empezar por lo básico: determinar el número de supermercados a implementar en la nave, lo que nos lleva también a su localización y saber que estaciones se van a surtir, siendo éste el primer aporte de Nils Boysen y Simon Emde en sus estudios en cuanto a modelos matemáticos y el tema de estudio de este trabajo. Cada vez más estos autores van evolucionando sus estudios aportando hasta el momento modelos matemáticos para la distribución de las partes en la estaciones, definición de una ruta, una programación de entregas y la carga de materiales a remolcar en los trenes.

## **2.2. Métodos de Optimización de Modelos Matemáticos**

Tradicionalmente, hay dos clases principales de algoritmos de optimización utilizados, y estos se clasifican en: la técnica basada en el cálculo y la técnica enumerativa. Técnicas de optimización basadas en cálculo, que emplean el mecanismo de búsqueda de gradiente para resolver la superficie de error o la superficie diferenciable de una función objetivo (Gill PE, 1981). Sin embargo, para

una función objetivo mal definida o multimodal, las soluciones óptimas locales (limitadas) se obtienen con frecuencia. En el procesamiento de la señal, las funciones objetivo de esta categoría son comunes ya que la señal puede ser ruidosa, confusa, vaga, y discontinua.

La programación dinámica (PD) es capaz de manejar el problema de la solución óptima local y se considera como una de las principales técnicas enumerativas en la búsqueda de operaciones (Bellman RE, 1952). Su simplicidad, robustez, y la popularidad deben enfrentarse al alto consumo computacional. Además, la PD puede descomponer los problemas complejos en un tamaño moderado, una situación que es ampliamente conocido como la "maldición de la dimensionalidad" (Tang KS, 1994). Un nuevo método de solución que es más óptimo a los mencionados anteriormente son los Algoritmos Genéticos (AG).

### **2.2.1 Los Algoritmos Genéticos**

Los AG fueron desarrollados por John Holland, en la universidad de Michigan en la década de 1970 (Holland, 1975), junto a su equipo de investigación. Éstos combinan las nociones de supervivencia del más apto con un intercambio estructurado y aleatorio de características entre individuos de una población de posibles soluciones, conformando un algoritmo de búsqueda que puede aplicarse para resolver problemas de optimización en diversos campos (Goldberg D, 1989).

Los AG son esquemas de representación que aplican una técnica de búsqueda basada en la teoría de la evolución de Charles Darwin. Están basados en los procedimientos naturales de selección para la evolución, en los que los individuos más aptos de una población son los que sobreviven al adaptarse más fácilmente a las características del entorno en el cual se encuentran. Este proceso biológicamente se controla por medio de los genes de un individuo, en los cuales se encuentra la

codificación de cada uno de los atributos o características de un ser vivo y que pueden ser transferidos a sus descendientes cuando se reproducen.

La combinación de los genes de un individuo forman los cromosomas, que en conjunto proyectan las cualidades de cada ser vivo. Esta estructura biológica refleja el grado de supervivencia indirectamente, en adaptación y el nivel de reproducción; así, los individuos que más se adaptan a su medio ambiente son los que más sobreviven a las adversidades y más se reproducen, transmitiendo esas buenas cualidades a las siguientes generaciones originando seres cada vez con mayor capacidad y facilidad de adaptarse a su entorno.

Los AG son algoritmos de optimización utilizados para maximizar o minimizar una función determinada, son algoritmos heurísticos adaptativos de búsqueda basada en las ideas evolutivas de selección y genética natural. Representan una explotación inteligente de una búsqueda al azar utilizado para resolver problemas de optimización (Sivaraj R, 2011). Cada generación se compone de una población de cadenas de caracteres que son análogas al cromosoma que se ven en el ADN. Cada individuo representa un punto en un espacio de búsqueda y una posible solución. Los individuos de la población entonces se hacen pasar por un proceso de evolución (Sivaraj R, 2011).

Los AG se basan en una analogía con la estructura genética con las siguientes bases: Los individuos en una población compiten por los recursos y parejas. Las personas más exitosas en cada "competencia" producen más descendencia que aquellos individuos que funcionan mal. Los genes de los individuos "buenos" se propagan por toda la población para que dos buenos padres puedan producir descendientes que son mejores que cualquiera de la combinación de los padres. Así, cada generación será más adecuada para su entorno (Goldberg D, 1989).

Un parámetro crítico de los AG es el forzar el proceso de selección de los mejores individuos entre toda la población para la próxima generación. Si se escogen individuos con genes “malos”, entonces la tasa de convergencia hacia la solución óptima es muy baja. Si se escogen individuos con genes “buenos”, entonces la tasa de convergencia hacia la solución probablemente sea atrapada en un rango óptimo local debido a la diversidad de población.

Por lo tanto los métodos de selección en la población controlan la precisión, que a su vez determinan la rapidez con la cobertura de algoritmos. La velocidad de convergencia de los diferentes esquemas de selección se estudió primero por Goldberg y Deb (Goldberg D, 1989). El mecanismo de selección debe ser elegido de tal manera que se aproxime a la solución óptima global sin ser atrapado en el óptimo local. También debe abarcar el conocimiento de los datos existentes (Sivaraj R, 2011).

Los AG difieren en algunos aspectos respecto a los métodos tradicionales (Goldberg D, 1989):

- a) Los AG trabajan con la codificación del conjunto de parámetros, no con los parámetros en sí.
- b) Los AG buscan en una población de puntos, no un punto en particular, es decir, evalúan un grupo de soluciones buscando el óptimo en lugar de un punto a la vez.
- c) Los AG evalúan las posibles soluciones (función de aptitud), sin aplicar ningún proceso de inferencia.
- d) Los AG usan reglas de transiciones probabilísticas en lugar de reglas determinísticas.

La codificación más común es por medio de cadenas binarias, aunque se han utilizado también números reales y letras. Esta forma de codificación originalmente

fue propuesta por John Holland y actualmente es muy popular debido a que resulta muy sencilla de implementar.

Para aplicar un AG, primero se codifican los interruptores como una cadena de longitud finita. Una manera simple de hacerlo es asignado el valor de 1 a aquellos interruptores que se encuentren encendidos y el valor de 0 en caso contrario. De este modo la cadena 11110, significa que los primeros cuatro interruptores se encuentran encendidos, mientras que el último está apagado.

Después se aplican operadores genéticos a las cadenas de la población y se evalúan para encontrar la solución. Esta forma de codificación ha gozado de mucha popularidad debido a que es la que se propuso originalmente por Holland y es muy sencilla de implementar; sin embargo, no es la única forma de codificación en un AG. El esquema de representación determina la estructura del problema en el AG así como la forma de aplicar los operadores genéticos.

El alfabeto que forma los genes de los cromosomas puede consistir de los dígitos binarios (0 y 1), números de punto flotante, enteros, patrones de símbolos con un determinado orden y significado, matrices, etc. (Michalewicz, 1992) ha hecho diversos experimentos comparando AG con representación numérica y binaria y recomienda utilizar representación de punto flotante para problemas de optimización numérica debido a que ofrece mayor consistencia, precisión y velocidad durante su ejecución.

Existen aplicaciones particulares que trabajan con patrones determinados de símbolos para representar los genes de un cromosoma, para los cuales se aplican los operadores genéticos a bloques de símbolos y no a símbolos individuales. En este tipo de problemas se presentan combinaciones de datos de diferentes tipos, motivo que origina la representación genética mediante estructuras un poco más complejas.

### 2.2.2 Clasificación general de los Algoritmos Genéticos

Según R.Sivaraj et al. (2011), la clasificación de los Algoritmos Genéticos es la siguiente:

#### 1. Los mecanismos tradicionales:

- Los métodos de selección proporcionales
  - Selección por vuelta de ruleta
  - Muestreo determinístico
  - El remanente de muestreo estocástico
  - Selección del remanente estocástico con reemplazo
  - Selección del remanente estocástico sin reemplazo
  - Selección universal estocástica
- Selección de clasificación
  - Selección de clasificación lineal
  - Selección truncada
- Selección de torneo
  - Selección de torneo binaria
  - La selección más grande del torneo
  - Torneo de selección de iones Boltzmann
  - Selección Torneo correlativa

#### 2. Mecanismos de selección de alternativas

- Selección de rango
- Selección de género-específica
  - Algoritmo genético con la diferenciación cromosoma.
  - Apareamiento restringido
  - Selección basada en la familia Coorelative
- Búsqueda genética selección basada en
  - Gimnasio esquema de selección uniforme (FUSS)
  - Selección Reserva

### 2.2.3 Operadores de los Algoritmos Genéticos

La mecánica de los AG es muy simple, ya que solamente involucra copiar cadenas de caracteres e intercambiar subcadenas aplicando algunos operadores. Los operadores básicos utilizados en un AG son los siguientes:

- a) Selección.
- b) Cruzamiento.
- c) Mutación.

La selección es un proceso en el cual cada cadena individual es copiada de acuerdo a los valores de su función de aptitud  $f$ . Intuitivamente se puede pensar que la función  $f$  es una medida de rendimiento, utilidad o bienestar que se desea maximizar o minimizar. Al copiar cadenas de acuerdo a esa función significa que aquellas cadenas con un valor más alto o más bajo dependiendo el caso y van a tener una probabilidad más alta de contribuir o aportar a la siguiente generación.

Este operador es una versión artificial de la selección natural según la Teoría de Darwin de la supervivencia de los individuos más capaces, aptos y superiores sobre los demás. El cruzamiento es un operador que básicamente consiste en intercambiar subcadenas de una población por medio de un punto de corte y se puede aplicar a un par de cadenas de dos formas:

- a) Seleccionar un punto de cruce fijo en cada una de las cadenas.
- b) Seleccionar dos números enteros  $k$  entre 1 y la longitud de la cadena menos uno  $[1, \text{long} - 1]$  en forma aleatoria que determinen los puntos de cruce.

Normalmente el cruzamiento se maneja dentro de la implementación del AG como un porcentaje que indica con qué frecuencia se efectuará. Esto significa que no todas las parejas de cromosomas se cruzarán, sino que habrán algunas que pasarán

intactas a la siguiente generación. De hecho existe una estrategia en la que el individuo más apto a lo largo de las distintas generaciones no se cruza con nadie y se mantiene intacto hasta que surge otro individuo mejor que él, que lo desplazará.

La mutación realiza un cambio a uno de los genes de un cromosoma elegido aleatoriamente. Cuando se usa una representación binaria, un bit se sustituye por su complemento (un cero se cambia por un uno y viceversa). Este operador permite la introducción de nuevo material cromosómico en la población, tal y como sucede con sus equivalentes biológicos.

Al igual que la cruce, la mutación se puede manejar con una probabilidad que indica con qué frecuencia se efectuará, aunque a diferencia del cruzamiento, esta ocurre más esporádicamente (la probabilidad de cruce normalmente se encuentra de 0.6 a 0.95 mientras que el de mutación normalmente oscila entre 0.001 y 0.01) (Fogel, 1995).

La mutación es necesaria debido a que, aunque la selección y la cruce son operadores efectivos de búsqueda en un AG, ocasionalmente dejan de analizar material genético útil. El operador de mutación protege al AG de pérdidas prematuras de oportunidades de análisis de secciones del espacio de estados cuando se utiliza conjuntamente con los otros operadores genéticos. Sin embargo, debido a que la probabilidad de aplicarse es muy baja, es considerado un operador secundario (Goldberg D, 1989).

Existen variantes de los operadores antes descritos que permiten alterar el comportamiento de un AG. Por ejemplo, resultados teóricos han comprobado que la convergencia de un AG hacia la solución de un problema es difícil de obtener, sin embargo, algunos estudios han arrojado resultados que permiten comprobar que ciertas modificaciones al operador de cruce puede generar cualquier punto del espacio de estados mientras que otros operadores no lo logran.

También debe considerarse que generar diversos puntos no asegura una búsqueda eficiente. Debe haber un balance adecuado entre la generación y la exploración de poblaciones además de las características particulares de cada población como su representación y su tamaño, que permita obtener un resultado apropiado durante la búsqueda (Fogel, 1995). A la fecha no existe un procedimiento exacto y consistente para determinar el tamaño ideal de la población y el número de generaciones que deben analizarse en un problema particular, por lo que aún son temas de investigación.

## CAPÍTULO 3. METODOLOGÍA

La metodología de desarrollo para este trabajo está basada en los autores Pacheco, A. y Cruz, M.C. (2006) y consta de los siguientes pasos:

- Descripción del objeto de estudio
- Diseño teórico de la propuesta
- Aplicación de la propuesta
- Resultados

### 3.1 Descripción del objeto de estudio

En los sistemas de producción de hoy en día, la siempre creciente variedad de productos posee un gran reto para los sistemas de logística internos usados para alimentar las líneas de ensamble con las partes necesarias. Como respuesta a este reto muchas empresas, especialmente de la industria automotriz han definido el concepto de Supermercado como parte de una promesa para una estrategia para permitir entregas flexibles de pequeños lotes de material a un costo bajo (Emde S, *Optimally locating in-house logistics areas to facilitate JIT-supply of mixed-model assembly lines*, 2012).

Un reto importante dentro de este contexto es el de alimentar las partes a las unidades productivas (estaciones) en la línea de producción. De un lado, los materiales y las partes deben siempre buscar a las estaciones de trabajo a tiempo para evitar costos excesivos a causa de los paros de línea. Del otro lado, almacenamientos excesivos en las estaciones y/o el tránsito de la línea en el taller permite un costo elevado en el manejo y manutención de este material. De acuerdo al principio del JIT, una cantidad considerable de fabricantes están adoptando el tan llamado concepto de Supermercado (Emde S, *Optimally locating in-house logistics areas to facilitate JIT-supply of mixed-model assembly lines*, 2012).

Un problema importante de optimización a considerar es la determinación del número y el lugar de instalación de las áreas de los supermercados. La implementación de supermercados conlleva un costo para poderlo poner en marcha, si se piensa en crear un gran número de supermercados, implicara muchos más costos que beneficios. Tener muy pocos, implica una mala ubicación, por otro lado, disminuirá en gran medida sus efectos positivos, por lo que, en el peor de los casos, es mejor mantener los sistemas tradicionales centralizados de almacenamiento.

Utilizando el espacio disponible lo mejor posible al seleccionar el número óptimo de supermercados y su estratégica localización son de importancia crítica (Emde S, *Optimally locating in-house logistics areas to facilitate JIT-supply of mixed-model assembly lines*, 2012). En este trabajo se presenta un esquema de para resolver este problema de manera más óptima que los propios autores con los métodos de solución de Algoritmos Genéticos en vez de programación polinomial dinámica en el tiempo utilizada por ellos.

Del lado negativo, los supermercados consumen espacio en el piso de la fábrica, que es escaso y caro. Las piezas son guardadas en estantes diseñados para facilitar el acceso, de tal manera que los trabajadores puedan tomar las piezas de una manera cómoda, de manera análoga a los clientes de un supermercado tradicional, y por lo tanto suelen ser menos eficiente con el espacio de los almacenes tradicionales. Una implementación efectiva del concepto de supermercado también requiere de una inversión en equipos, personal y mantenimiento. Encontrar el mejor trabajo respecto a esta disyuntiva y la investigación de los beneficios operativos de los supermercados son cuestiones importantes que este trabajo abordará.

Acorde a Simon Emde y Nils Boysen, la planificación y el control de este concepto de logística interna, asciende al concepto de una tarea compleja que involucra varios problemas de decisión relacionados entre sí:

1. Decidir sobre el número y la ubicación de los supermercados descentralizados.
2. Determinar el número de trenes de remolque al supermercado y asignar segmentos de línea para ellos.
3. Determinar horario fijo de entrega de cada tren de remolque.
4. Decidir sobre los contenedores para ser cargados por tour de un tren de remolque

En este modelo se propone el inicio de la implementación de los sistemas de supermercados tomando el primer punto de los arriba mencionados y será estudiado con el primer modelo matemático propuesto por Simon Emde y Nils Boysen.

### **3.2 Diseño teórico de la propuesta**

Refiriéndonos al objetivo de la optimización, minimizar la suma de distancias ponderadas de cada supermercado a las estaciones que surte, regularmente es usado el problema de localización en el plano o *Lay-Out*, este es ciertamente aplicable, si se conocen las peculiaridades de transporte de los carritos en el surtimiento (Emde S, Optimally locating in-house logistics areas to facilitate JIT-supply of mixed-model assembly lines, 2012). Los carritos *Tugger* usualmente no visitan a todas las estaciones de manera individual, sino que viajan en rutas predeterminadas con múltiples paradas.

Como la ruta del supermercado hacia las estaciones no es una línea recta, entonces cuando calculamos las distancias, debemos considerar la manera en que los carros *Tuggers* van a viajar: primero, desde el supermercado a la primera estación de trabajo a la que van a surtir en su ruta, después de estación a estación y finalmente, de regreso al supermercado a volver a llenar los carritos para volverlos a surtir. Aun

así minimizar las distancias no necesariamente garantizan la mejor solución. Esto debido a que los *Tugger* tienen una capacidad limitada de cuantas y que estaciones puede surtir en un recorrido desentendiendo del consumo de partes en esas estaciones.

Las estaciones con menor demanda pueden ser una ruta, ya que los carritos no podrán ser capaces de servir a todas las estaciones por igual. Los supermercados saturados tendrán que usar carritos, rutas y almacenamientos de seguridad adicionales, lo que conlleva un mayor costo de operación. Por lo tanto las distancias deberán de ser ponderadas por la demanda total de las estaciones surtidas por cada supermercado. La cantidad de partes exactas a surtir son determinadas por el consumo de ellas en el turno una vez que se conoce la secuencia.

Independientemente la demanda de los bins por estación o turno puede ser estimada con cierta precisión, como esto no requiere de un conocimiento muy profundo de la sincronización y composición de la secuencia del modelo sino una evaluación del promedio del volumen de producción. Finalmente como tercer componente, se fija un costo de creación y manutención de los supermercados. Tomando todos estos puntos en consideración, se usa la notación siguiente y definimos el problema de localización del supermercado como sigue:

Asumiendo una línea de montaje de modelo mixto clásico, a lo largo de los cuales hay  $s=1, \dots, S$  estaciones de trabajo, que tienen que ser suministradas con  $d_s$  contenedores de partes del supermercado. La posición de cada estación en el *Lay-Out* de la fábrica se identifica por dos dimensiones de coordenadas  $(a_s, b_s)$ , que define la posición de ubicación de un *Tugger* o remolques (los que jalan a los carritos de supermercados o *Tow Trains* para su cambio en la línea) para el suministro de la estación de  $s$ . Con estas coordenadas, las distancias entre la estación  $s$  y  $s+1$ , así como las distancias de  $z_{is}$  entre la estación  $s$  y el supermercado  $i$ , donde  $i = 1, \dots, n$ ,

para ser cubierto por un *Tugger* al visitar estos lugares, se puede calcular fácilmente. Se deben de tomar en cuenta las siguientes consideraciones:

- El surtimiento de estaciones en el supermercado debe de ser siempre consecutivo, no se pueden hacer surtimiento o entregas de material de la estación 1 a la 3 y de ahí a la 5, y otro surtimiento de la estación 2 a la 4.
- Los cuellos de botella son siempre un desperdicio (en costo) en rutas adicionales, vehículos y horarios ajustados, por eso es que no hay restricciones para el número de estaciones que puede surtir un supermercado. Si las distancias y las cargas en un supermercado son correctas, tal programación será ajustada de forma natural en la función objetivo.
- Las partes son empacadas en bines de igual tamaño (estandarizados), lo cual es un requerimiento para sus racks en la línea.
- Las rutas, los horarios y las cargas para cada carrito de supermercado pueden ser determinados solamente por el tiempo más tardado, cuando la secuencia de producción es conocida. Por lo tanto el problema de localización del supermercado puede ser resuelto en base a estimaciones agregadas.
- Considerando que las entregas más rápidas en las rutas más cortas son el objetivo inicial, los supermercados tiene que ser instalados en la línea. Exactamente qué tan cerca de la línea puedan estar van a depender del tamaño de las estaciones a las que van a surtir y la cantidad de espacio reservada para dichos supermercados.
- Como las partes no son entregadas punto a punto del supermercado a cada estación, sino son surtidas mediante un recorrido extenso con múltiples paradas, los supermercados deben estar lo más cerca posible a todas las

estaciones que les corresponde surtir. Por lo tanto deben localizarse justamente en medio de toda el área de surtimiento.

**Tabla 3. Notación matemática del problema**

Notación	
$S$	Estaciones de trabajo ( $s = 1, \dots, S$ )
$N$	Variable para el número de supermercados a implementar ( $i=1, \dots, n$ )
$T$	Precio fijo por supermercado a implementar
$D_s$	Número de bins o contenedores del supermercado
$A_s$	Coordenadas en X de la posición de las estaciones de trabajo
$b_s$	Coordenadas en Y de la posición de las estaciones de trabajo
$e_s$	Distancia entre estaciones $s$ y $s+1$
$z_{is}$	Distancia entre estación $s$ y el supermercado $i$ ( $i = 1, \dots, n$ )
$x_i$	Variable asignada para la primera estación a surtir en el supermercado $i$ a la estación $s$ .

*Fuente: Simon Emde & Nils Boysen, 2012. Optimally locating in-house logistics areas to facilitate JIT-supply of mixed-model assembly lines*

Dada  $s = 1, \dots, S$  estaciones de trabajo consecutivas a un suministro de partes del supermercado, el problema de ubicación del supermercado consiste en dividir las estaciones en un número variable de  $n = 1, \dots, S$  subconjuntos disjuntos, cada uno atendida por un supermercado diferente. Una solución se codifica como un vector  $X(n) = \{1, x_2, x_3, \dots, x_n, S+1\} \rightarrow \{2, \dots, S\}$ , donde  $x_i$  representa a la estación más a la izquierda del suministro del supermercado, en consecuencia, la estación más a la derecha servida por el supermercado  $i$  se determina por  $x_{i+1}-1$ . Puesto que todas las estaciones se deben incluir en el área de surtimiento de algún supermercado, la estación más a la izquierda del primer supermercado siempre es 1, y la última estación más a la derecha es siempre  $S$ .

El vector es de longitud  $n+1$  y sus miembros son configurados hacia el objetivo (1) de reducirse al mínimo y las limitaciones (2) y (3) se observan:

### 3.2.1 Problema de Ubicación del Supermercado

Minimizar  $F(X(n))$

$$\mathbf{min} F(X(n)) = \sum_{i=1}^n \sum_{s=x_i}^{x_{i+1}-1} ds \cdot \left( \sum_{T=x_i}^{x_{i+1}-2} e_T + z_{i,x_i} + z_{i,x_{i+1}-1} \right) + n \cdot T \quad (1)$$

Sujeto a:

$$x_{i+1} \geq x_i + 1 \quad \forall i = 1, \dots, n \quad (2)$$

$$z_{is} = |a_{x_i} + (a_{x_{i+1}-1} - a_{x_i})/2 - a_s| + |b_{x_i} + (b_{x_{i+1}-1} - b_{x_i})/2 - b_s|$$

$$\forall i = 1, \dots, n; -s = 1, \dots, S \quad (3)$$

La función objetivo (1) tiene como finalidad reducir al mínimo el número de los supermercados  $n$  ponderado por su costo fijo  $T$  y la suma sobre todos los supermercados de la demanda total en las estaciones de los respectivos suministros de estos supermercados multiplicado por la longitud su ruta a través de las estaciones y su regreso. Hay que tener en cuenta el cálculo de las distancias que simula la ruta de un tren de remolque que tendrá que salir del supermercado: En primer lugar, la distancia  $z_{i,x_i}$  del supermercado  $i$  a la primera estación  $x_i$  en su zona de suministro, después la distancia  $\sum_{T=x_i}^{x_{i+1}-2} e_T$  de una estación a otra, y finalmente, la distancia  $z_{i,x_{i+1}-1}$  al volver al supermercado.

Se debe considerar también que el área suministrada de la línea por un supermercado puede seguir cualquier tipo de camino, ya sea lineal o serpentinamente, esto puede ser codificado en la distancia entre las estaciones  $e_T$ . La restricción (2) asegura que no hay zonas de abastecimiento traslapadas y la restricción (3) ayuda a calcular la distancia de cada supermercado para cada estación. Teniendo en cuenta que los pisos de la fábrica se caracterizan por líneas de conducción del lado de línea y curvas cerradas, la métrica *Manhattan* (o rectilínea) es método más adecuado para medir estas distancias. Los supermercados están

colocados en medio de los dos puntos extremos de su área de surtimiento, ya que los *Tuggers* tendrán que comenzar su recorrido al salir del supermercado y finalmente acabarlo mediante su retorno nuevamente a él, probablemente desde el otro extremo de la zona de alimentación del supermercado. Además se debe considerar que, en muchos casos, los supermercados pueden, por supuesto, no encontrarse exactamente en el medio de las dos estaciones debido a que el ensamble en la línea de producción puede presentar un desplazamiento constante que no es idéntico en todas partes, esto no tiene que ser modelado explícitamente.

Un problema práctico que pueda surgir cuando se utiliza la función objetivo anterior es causada por la inclusión de dos factores de costo muy diferentes en una función objetivo común: Por un lado, el término que estima el costo futuro de operación y trata de minimizar las distancias y demandas en las áreas de suministro de todos los supermercados, los cuales tienden a ser menores mientras más supermercados existan. Por otra parte, el costo de implementación y de mantenimiento inferido mediante la creación de nuevos supermercados. Si bien el compromiso entre estos dos factores sin duda existe, estableciendo un costo exacto coeficiente  $T$  puede ser difícil en la práctica. Por lo tanto, propondremos un algoritmo en la siguiente sección que calcula todos los pares no dominados de estimación de costos de operación y el recuento de supermercado  $n$ . Todas estas soluciones son óptimas para su respectiva  $n$ .

### **3.2.2 Procedimiento de la solución**

Para cada supermercado  $i$ , la estación  $x_i$  más a la izquierda en su zona de suministro debe ser determinada. Esto ajusta automáticamente la estación más derecha del área del supermercado precedente como  $x_{i-1}$ . Dado que los conjuntos de estaciones ordenadas suministrados por cada supermercado no se pueden traslapar y los valores de distancia y demanda dependen sólo de la zona del supermercado al que

pertenecen y no en los que vienen antes o después, las soluciones óptimas pueden ser eficientemente construidas por un enfoque de programación.

Sea  $k$  la primera estación en el área de un supermercado,  $G(k)$  el costo mínimo para el intervalo de la estación desde 1 hasta  $k-1$  con  $G(1):= 0$ ,  $f(j, k)$  el valor objetivo para el supermercado que sirve la estación  $j$  hasta  $k-1$ , determinado por:

$$f(j, k) = \sum_{s=j}^{k-1} d_s * (\sum_{T=j}^{k-2} e_T + \hat{z}_j + \overline{z}_{k-1}) \quad (4)$$

Donde  $\hat{z}_s$  es calculado con la ecuación (3) (con  $x_i=j$  y  $x_{i+1}=k$ ). La programación es definida como:

$$G(k) = \min_{1 \leq j \leq k-1} \{G(j) + f(j, k)\} \quad (5)$$

Ahora el objetivo es hallar el camino hacia  $S + 1$  con el  $G(S+1)$  para un número dado  $n$  de supermercados. Una descripción formal del procedimiento es el cálculo de la frontera de todos los valores pares no dominados  $(n, p_n, G_n(S+1))$ , (donde  $n$  es el número total de supermercados,  $p_n$  codifica la correspondiente solución óptima, que puede ser decodificada por una sencilla recuperación hacia atrás, y  $G_n(S+1)$ , es el valor de la función objetivo asociada con la solución).

Para saber interpretar la solución, se tiene que entender que se está creando una estación fantasma en las notaciones de la formula (S+1). Esto con el objetivo de poder interpretar a que estaciones va a poder surtir material cada supermercado, teniendo en cuenta la condición que se explica en este punto.

### 3.3 Aplicación de la propuesta

Ahora que se tiene toda la notación matemática para programar se explicará brevemente como es que fue resuelto este problema mediante Algoritmos Genéticos,

pero antes vamos a entender la notación para nuestros valores de entrada para la programación en la tabla 4:

**Tabla 4. Notación del programa de MATLAB**

Valores de entrada para la programación	
$d$	Vector que determina los valores de la demanda de la estación $n$
$c$	Vector que determina los valores de las coordenadas de la estación $n$
$i$	Valor para la estación de inicio de la ruta de cálculo
$f$	Valor para la estación final de la ruta de cálculo
$ts$	Valor para determinar el número de supermercados que se quieren implementar
$CS$	Valor para determinar el costo fijo de implementación de los supermercados
$NI$	Valor para el número de individuos que se harán para ir acumulando resultados
$Iter$	Valor para el número de iteraciones que se harán para ir acumulando resultados
$ProbM$	Valor para determinar la probabilidad de mutación para los Algoritmos Genéticos
$N$	Valor para el número de supermercados implementados en el problema

*Fuente: Elaboración propia*

Para los valores del vector  $d$  ocupamos un vector con dimensiones  $1 \times n$  y se deben de introducir de manera sucesiva y ordenada conforme a las estaciones, por ejemplo:  $d = (3, 7, 1, 8, 10)$ , esto significa que para la estación 1 se presenta una demanda de 3, para la estación 2 una demanda de 7 y así sucesivamente.

Para los valores del vector  $c$  ocupamos un vector con dimensiones  $2 \times n$  y se deben de introducir de manera sucesiva y ordenada conforme a las estaciones, por ejemplo:  $c = \begin{pmatrix} 1, 5, 12, 12, 12 \\ 1, 1, 1, 10, 17 \end{pmatrix}$ , esto significa que la estación 1 tiene coordenadas de  $x$  en 1 y de  $y$  de 1, para la estación 4 tiene coordenadas de  $x$  en 12 y en  $y$  de 10 y así sucesivamente.

Con los valores  $i$  y  $f$  se determina qué estaciones se deben de cubrir para el costo, por ejemplo:  $i = 2$  y  $f = 5$ , se está determinando que se quiere calcular la ruta de la estación dos hasta la estación 5.

Con el valor  $ts$  se define cuantos supermercados se quieren implementar en el problema, si el número de supermercados que se introdujo supera al número de estaciones del problema, aparecer un mensaje de error debido a que es una incongruencia poder poner más supermercados que estaciones disponibles.

Con el valor  $CS$  estamos determinado el costo fijo de implementación por cada supermercado, este valor es de suma importancia ya que en el radica el objetivo del problema que es reducir los costos de implementación y se debe de calcular o aproximar de manera particular dependiendo de cada empresa.

El valor de  $NI$ , es básico para la resolución del algoritmo genético, la condicionante que tiene es que debe de ser múltiplo de 4, esto debido a que no se pueden combinar parejas que no se puedan dividir en partes iguales, mientras más individuos se introduzcan más exacta y óptima es la solución pero es más tardado el tiempo de solución en el programa.

El valor de  $Iter$  es básico para la resolución del algoritmo genético y no tiene ninguna restricción, mientras más iteraciones se introduzcan más exacta y óptima es la solución pero es más tardado el tiempo de solución en el programa.

El valor de  $ProbM$  es básico para la resolución del algoritmo genético y la única restricción que se tiene es que no debe de ser mayor a uno, mientras más grande sea este número, más cambios existen en los vectores propuestos por el programa y puede retrasar la optimización del programa de simulación.

### **3.3.1 Resolución en MATLAB de los modelos de los autores**

Primeramente para saber que la propuesta de este trabajo está funcionando de manera adecuada, se resuelven los algoritmos del artículo de los autores uno a uno y se compararán con los resultados de los autores en el problema que manejan de

ejemplo para la comprobación de su teoría en su artículo. En la Tabla 5 se muestran los valores de dicho problema y son los valores que se dan de ejemplo en este trabajo para ir paso a paso entendiendo la resolución a lo largo del desarrollo de esta propuesta.

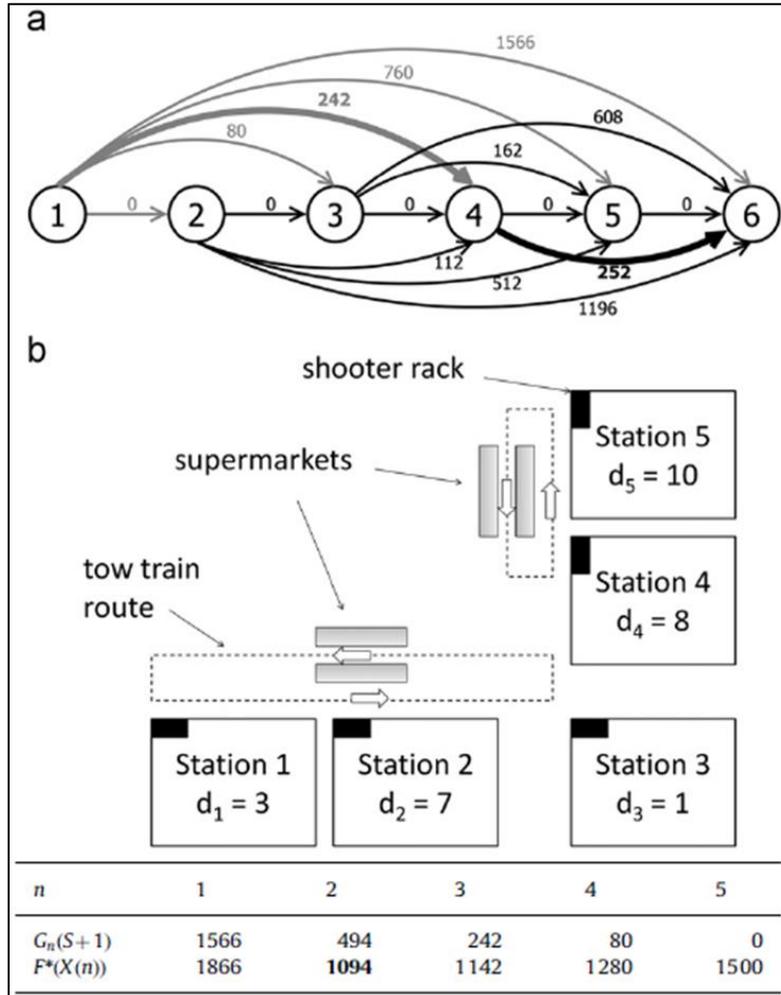
**Tabla 5. Datos del ejemplo de los autores**

Station $s$	1	2	3	4	5
$d_s$	3	7	1	8	10
$a_s$	1	5	12	12	12
$b_s$	1	1	1	10	17
$e_s$	4	7	9	7	

$\Gamma = 300$

*Fuente: Simon Emde & Nils Boysen, 2012. Optimally locating in-house logistics areas to facilitate JIT-supply of mixed-model assembly lines*

En la figura 3 muestran los resultados generales y el resultado óptimo de dicho problema por medio de su solución con programación dinámica. A continuación se explica paso a paso como está desarrollado el algoritmo y los resultados que nos arroja en el programa de MATLAB y se compara con los resultados de los autores, esto para comprobar el funcionamiento de la propuesta de este trabajo desde el inicio:



**Figura 3. Resultados del ejemplo resuelto por los autores**

*Fuente: Simon Emde & Nils Boysen, 2012. Optimally locating in-house logistics areas to facilitate JIT-supply of mixed-model assembly lines*

1. Primero se realizó el programa para calcular los valores relacionados de las distancias y demandas entre las estaciones determinadas por el valor de sus coordenadas y la demanda con la implementación por medio de la ecuación (3) (ver Figura 4). Todo esto referido a una ruta ya predeterminada de surtimiento de supermercados. Los resultados se muestran en la figura 5 y se pueden comparar con los resultados de los autores en la parte a de la figura 3.

```

function s=Super(d,c,i,f)
zi=abs(c(1,i)+((c(1,f-1)-c(1,i))/2)-c(1,i))+abs(c(2,i)+((c(2,f-1)-
c(2,i))/2)-c(2,i));
zf=abs(c(1,i)+((c(1,f-1)-c(1,i))/2)-c(1,f-1))+abs(c(2,i)+((c(2,f-
1)-c(2,i))/2)-c(2,f-1));
x=d(i:f-1);
dt=sum(x);
for j=1:size(c,2)-1
xc=abs(c(1,j)-c(1,j+1));
yc=abs(c(2,j)-c(2,j+1));
e(j)=xc+yc;
end
e;
et=sum(e(i:f-2));
s=dt*(et+zi+zf);

```

**Figura 4. Programa para la resolución de la ecuación 3 en MATLAB**

*Fuente: Elaboración propia*

2. Ya que se tiene el valor del supermercado predeterminado que calcula su relación entre la demanda y su distancia, se le tiene que subir un costo fijo de implementación denotado por  $T$  por cada supermercado que se pretenda implementar. Esto nos da el valor total de implementación con la ecuación (1) (Ver Figura 6). El valor de la ecuación que se pretende minimizar.

El vector solución en esta parte es introducido manualmente en el software y es denotado como un vector de  $1 \times n$  a implementar, siempre debe iniciar con 1 y terminar con  $n+1$  (ver figura 7), hasta aquí no se ha generado ningún cálculo automatizado ya que primeramente se debe de resolver el problema y entenderlo, para posteriormente buscar la forma de que se resuelva de manera automática con tan solo meter los datos básicos del problema.

```

>> C
C =
     1     5    12    12    12
     1     1     1    10    17

>> D
D =
     3     7     1     8    10

>> Super(D,C,1,2)
ans =
     0

>> Super(D,C,1,3)
ans =
    80

>> Super(D,C,1,4)
ans =
   242

>> Super(D,C,1,5)
ans =
   760

>> Super(D,C,1,6)
ans =
  1566

>> Super(D,C,2,3)
ans =
     0

>> Super(D,C,2,4)
ans =
   112

>> Super(D,C,2,5)
ans =
   512

>> Super(D,C,2,6)
ans =
  1196

>> Super(D,C,3,4)
ans =
     0

>> Super(D,C,3,5)
ans =
   162

>> Super(D,C,3,6)
ans =
   608

>> Super(D,C,4,5)
ans =
     0

>> Super(D,C,4,6)
ans =
   252

>> Super(D,C,5,6)
ans =
     0

```

**Figura 5. Todos los posibles resultados del programa para la resolución de la ecuación 3 en MATLAB del ejemplo de los autores**

*Fuente: Elaboración propia*

```

function CF=Costo(D,C,Sol,CS)
Sol;
ts=size(Sol,2)-1;
cts=ts*CS;
for n=1:ts
    t(n)=Super(D,C,Sol(n),Sol(n+1));
end
T=sum(t);
CF=sum(T)+cts;
COSTOFINAL=CF;
SOLUCION=Sol;

```

**Figura 6. Programa para la resolución de la ecuación 1 en MATLAB**

*Fuente: Elaboración propia*

```
>> Sol=[1,3,6]

Sol =

     1     3     6
```

**Figura 7. Ejemplo de introducción de la ruta solución del problema en MATLAB**

*Fuente: Elaboración propia*

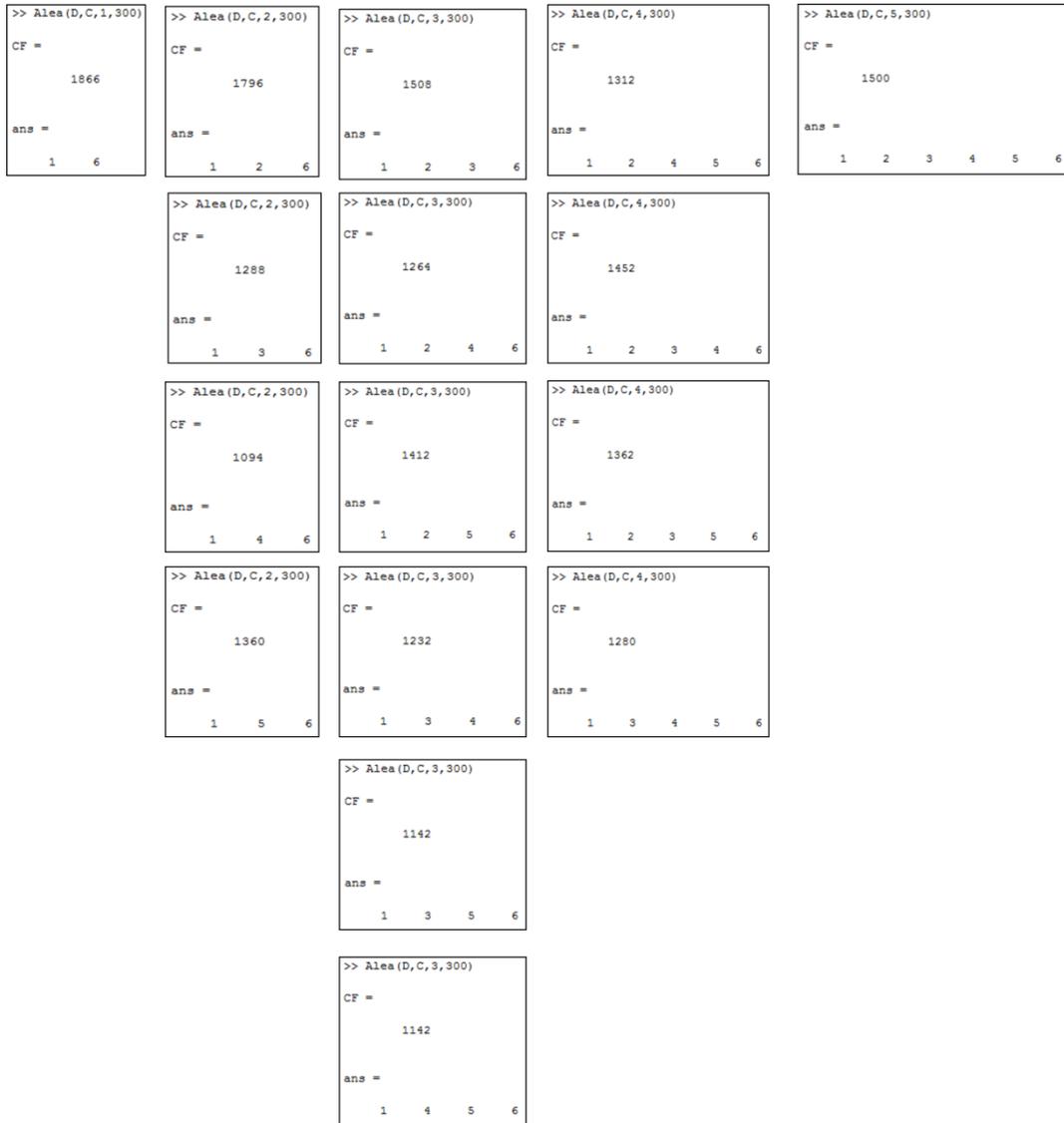
3. Ya que se sabe cómo obtener el valor que se pretende minimizar, se genera una función donde del programa pasado que genera de manera totalmente aleatoria las rutas de supermercados introduciendo solo el número de supermercados deseados a implementar, busque de todas las posibles soluciones de la ruta con el valor de costo de implementación final (Ver figura 8).

```
function [Sol,CF]=Alea(D,C,ts,CS)
B=size(D,1);
E=size(D,2)+1;
if ts<=E-1
    Sol=zeros(1,ts+1);
    Sol(1)=1;
    Sol(ts+1)=E;
    aux=randperm(E-2,ts-1);
    aux=aux+1;
    aux2=sort(aux);
    Sol(B+1:ts)=aux2;
    cts=ts*CS;
    for n=1:ts
        t(n)=Super(D,C,Sol(n),Sol(n+1));
    end
    T=sum(t);
    CF=sum(T)+cts
else
    disp('Error, Más Supermercados que Estaciones de Trabajo');
end
```

**Figura 8. Programa para la resolución de la ecuación 4 en MATLAB**

*Fuente: Elaboración propia*

Se deben de calcular todas las posibles resoluciones, para hacer esto más fácil en el programa se crean las rutas de manera aleatoria para no estar introduciéndolas de manera manual, en la figura 16 se muestran todas las posibles soluciones finales del problema con el costo de implementación de  $T = \$300$  como valor fijo de implementación por supermercado.



**Figura 9. Todos los posibles resultados del modelo para la resolución de la ecuación 4 en MATLAB del ejemplo de los autores**

*Fuente: Elaboración propia*

Los resultados que se muestran en la figura 9 se pueden comparar con los resultados de los autores en la parte a de la figura 3. Con esto logramos resolver y entender la parte de los modelos propuesta por los autores, de aquí en adelante se explica y se habla como es que fue resuelto el problema con Algoritmos Genéticos.

### **3.3.2 Propuesta y resolución en MATLAB del algoritmo genético para la búsqueda de la solución óptima del modelo de los autores**

Para generar la programación del algoritmo genético (ver figura 16) se debe de considerar la siguiente lógica:

- Se sabe que el número de supermercados a implementar no puede ser mayor al número de las demandas y coordenadas X o Y de las estaciones.
- La ruta de implementación de supermercados tiene que ir en orden ascendente y no puede repetir números.
- El valor inicial del vector ruta de implementación siempre debe de ser 1.
- El valor final del vector ruta de implementación siempre debe de ser igual al número de estaciones que se tienen +1 (debido a la estación fantasma que se denota en la solución).
- El número de individuos a cruzar debe de ser múltiplo de 4, debido a las referencias de cruce.
- Para el caso de implementar un supermercado para  $n$  estaciones de trabajo, el resultado siempre es lo mismo.
- Para el caso de que el número de supermercados a implementar sea igual a las estaciones de trabajo, el resultado siempre es lo mismo.

Primeramente se realiza una matriz de vectores ruta aleatorios con el programa anterior, este contiene el mismo número de estaciones (columnas) y un número introducido manualmente de individuos a cruzar (filas). Como ejemplo se utiliza el valor de  $n = 4$  que determina el número de supermercados y el valor de  $NI = 8$  que determina el número de individuos aleatorios creados (Ver Figura 10). Para ejemplo de las imágenes que se presentan a continuación, se siguen tomando los valores del ejemplo de los autores para un mayor entendimiento.

<b>n = 4</b>	<b>NI = 8</b>		
<b>1</b>	<b>3</b>	<b>4</b>	<b>6</b>
<b>1</b>	<b>2</b>	<b>5</b>	<b>6</b>
<b>1</b>	<b>2</b>	<b>3</b>	<b>6</b>
<b>1</b>	<b>3</b>	<b>5</b>	<b>6</b>
<b>1</b>	<b>2</b>	<b>3</b>	<b>6</b>
<b>1</b>	<b>3</b>	<b>5</b>	<b>6</b>
<b>1</b>	<b>4</b>	<b>5</b>	<b>6</b>
<b>1</b>	<b>2</b>	<b>3</b>	<b>6</b>

**Figura 10. Ejemplo de elaboración de matriz de vectores ruta aleatorios**

*Fuente: Elaboración propia*

La matriz de vectores ruta aleatorios se resuelve y se acomoda ascendentemente por medio de sus costos finales. Solo se queda con la mejor mitad de resultados para cruzarlos, lo demás se desecha (ver Figura 11).

1	3	4	6	=	1232
1	2	5	6	=	1412
1	2	3	6	=	1508
1	3	5	6	=	1142
1	2	4	6	=	1264
1	3	5	6	=	1142
1	4	5	6	=	1142
1	2	3	6	=	1508



①	1	3	5	6	=	1142
②	1	4	5	6	=	1142
③	1	3	5	6	=	1142
④	1	3	4	6	=	1232
⑤	1	2	4	6	=	1264
⑥	1	2	5	6	=	1412
⑦	1	2	3	6	=	1508
⑧	1	2	3	6	=	1508

**Figura 11. Ejemplo de elaboración de matriz de vectores ruta aleatorios**

*Fuente: Elaboración propia*

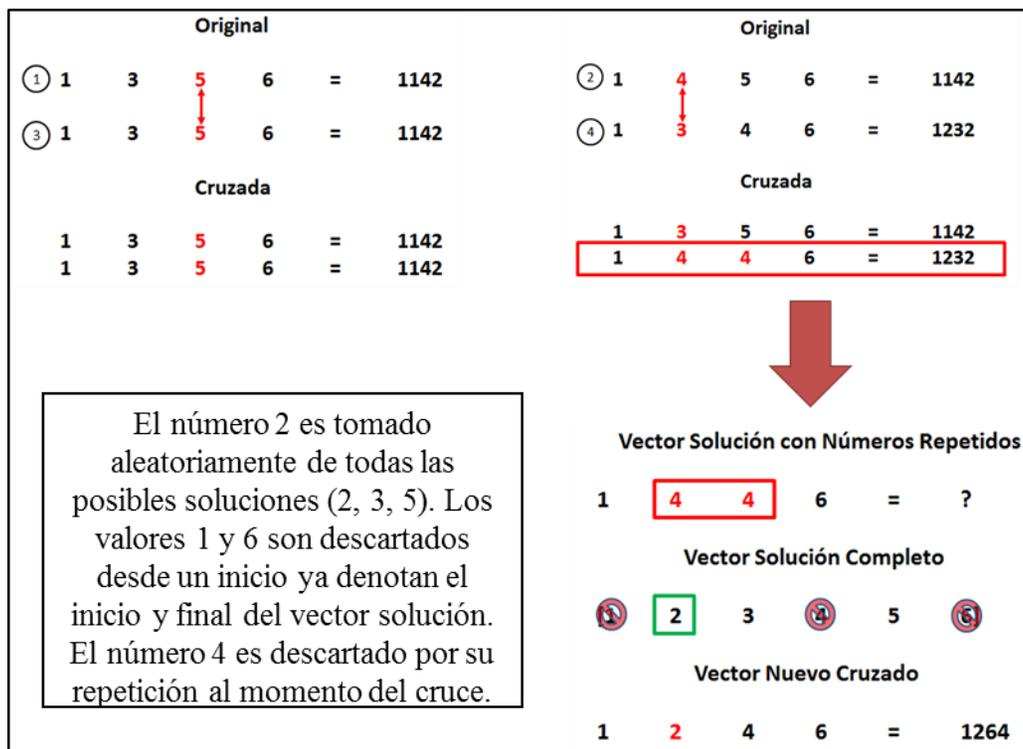
Para el cruce se sigue el procedimiento normal de un algoritmo genético. Se genera una matriz aleatoria de 2 columnas y el número de individuos / 4 filas que nos rige como combinar los vectores. Se determina la posición de los vectores a cruzar de manera aleatoria, esta posición no puede cambiar el primer ni el último valor del vector, ya que es el mismo para todos (ver figura 12).

$\begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$	①	1	3	5	6	=	1142
	③	1	3	5	6	=	1142
	②	1	4	5	6	=	1142
	④	1	3	4	6	=	1232

**Figura 12. Ejemplo de elaboración de matriz de cruce para los vectores ruta aleatorios óptimos**

*Fuente: Elaboración propia*

Al momento de hacer el cruce y se generen los nuevos vectores en la matriz, se puede dar el caso en que se tengan valores repetidos. En estas circunstancias se tienen que desechar los valores repetidos y agregarle valores nuevos a dicho vector, los valores nuevos tienen que ser valores que no pasen del número total de estaciones +1 que se tienen en el problema. El vector tiene que ser llenado con los valores que no aparecen seleccionándolos aleatoriamente de una población de valores que no contiene el vector, acomodado dicho vector ascendentemente nuevamente.

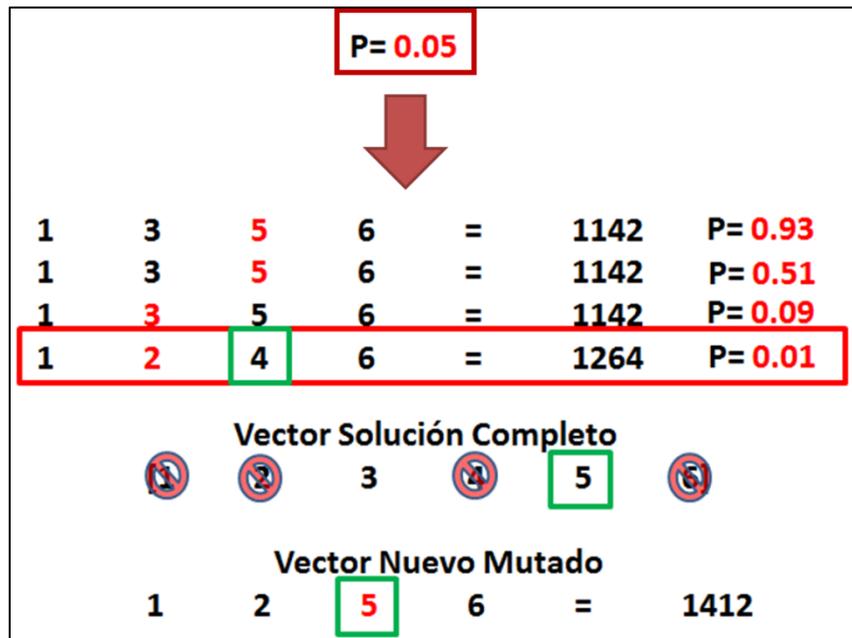


**Figura 13. Ejemplo de cruce para los vectores ruta aleatorios óptimos con valores repetidos en el cruce**

*Fuente: Elaboración propia*

Para la mutación se genera un valor aleatorio entre el 0 y el 1, dicho valor se compara con un índice de mutación que se introduce manualmente en el problema y si este es menor a dicho índice se procede a mutar el valor. Se debe de considerar que el primer y el último valor del vector no se pueden cambiar, solo los valores

intermedios. Para esto antes de generar la mutación se deben de conocer los valores que no tiene vector y así en el caso de que tenga que mutarse se puede elegir uno de esos valores aleatoriamente para agregarlo al vector y volverlo a acomodar ascendentemente (Ver figura 14).



**Figura 14. Ejemplo de mutación para los vectores ruta aleatorios óptimos**

*Fuente: Elaboración propia*

Los nuevos valores de la matriz son resueltos nuevamente y se vuelven a acomodar ascendentemente, quedándonos con el primer vector y su valor solución de la matriz, dejándolo como el mejor de la población (ver figura 15).

<b>1</b>	<b>3</b>	<b>5</b>	<b>6</b>	<b>=</b>	<b>1142</b>
<b>1</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>=</b>	<b>1142</b>
<b>1</b>	<b>3</b>	<b>5</b>	<b>6</b>	<b>=</b>	<b>1142</b>
<b>1</b>	<b>3</b>	<b>5</b>	<b>6</b>	<b>=</b>	<b>1142</b>
<b>1</b>	<b>3</b>	<b>5</b>	<b>6</b>	<b>=</b>	<b>1142</b>
<b>1</b>	<b>3</b>	<b>4</b>	<b>6</b>	<b>=</b>	<b>1232</b>
<b>1</b>	<b>2</b>	<b>5</b>	<b>6</b>	<b>=</b>	<b>1412</b>

**Figura 15. Ejemplo de la nueva matriz mutada para los vectores ruta aleatorios óptimos**

*Fuente: Elaboración propia*

```
function
[Minimos, COSTOFINAL, SOLUCION]=GenSuper (D, C, CS, ts, NI, Iter, ProbM)
B=size (D, 1);
E=size (D, 2)+1;
VC=1:E;
Sol=zeros (NI, ts+1);
Sol (:, 1)=1;
Sol (:, ts+1)=E;
F=size (Sol, 2);
Minimos=zeros (1, Iter);
for N=1:NI
aux=randperm (E-2, ts-1);
aux=aux+1;
aux2=sort (aux);
Sol (N, 2:ts)=aux2;
end
if ts==1||ts>=E-1
SOLUCION=Sol (1, :);
COSTOFINAL=Costo (D, C, SOLUCION, CS);
Minimos (1:Iter)=COSTOFINAL;
else
Sol;
cts=ts*CS;
for I=1:Iter
for N=1:NI
T=0;
for n=1:ts
T=T+Super (D, C, Sol (N, n), Sol (N, n+1));
end
CF (N)=T+cts;
end
[CFS, P] = sort (CF);
Minimos (I)=CFS (1);
I=P (1:NI/2);
```

```

M(1:NI/2,:) = Sol(I(1:NI/2),:);
X=reshape(randperm(NI/2),(NI/4),2);
for A=1:NI/4
cr=randi(F-2)+1;
p1=M(X(A,1),:);
p2=M(X(A,2),:);
h1=zeros(1,F);
h1(1:cr)=p1(1:cr);
h1(cr+1:F)=p2(cr+1:F);
hu1=unique(h1);
if size(hu1,2)<size(h1,2)
VC;
hd1=setdiff(VC,hu1);
g1=size(hd1,2);
f1=randperm(g1,size(h1,2)-size(hu1,2));
H1=[hu1 hd1(f1)];
hu1=sort(H1);
end
h2=zeros(1,F);
h2(cr+1:F)=p1(cr+1:F);
h2(1:cr)=p2(1:cr);
hu2=unique(h2);
if size(hu2,2)<size(h2,2)
VC;
hd2=setdiff(VC,hu2);
g1=size(hd2,2);
f2=randperm(g1,size(h2,2)-size(hu2,2));
H2=[hu2 hd2(f2)];
hu2=sort(H2);
end
aux=NI/2;
CF2=aux+(2*A);
CF=CF2-1;
M(CF,:)=hu1;
M(CF2,:)=hu2;
MMut=M;
CF=zeros(NI,1);
end
M;
for Mut=1:NI
if rand()<ProbM
RAN=setdiff(VC,M(Mut,:));
PMut=randi(F-2)+1;
PRAN=randi(size(RAN));
ARAN=RAN(PRAN);
MMut(Mut,PMut)=ARAN;
MMut(Mut,:)=sort(MMut(Mut,:));
end
T=0;
for n=1:ts
T=T+Super(D,C,MMut(Mut,n),MMut(Mut,n+1));
end
CF(Mut)=T+cts;
end
end

```

```

Sol=MMut;
end
MMut;
[CFS, P] = sort(CF);
SOLUCION=MMut(P(1),:);
COSTOFINAL=CFS(1);
End

```

**Figura 16. Programa para la generación del algoritmo genético MATLAB**

*Fuente: Elaboración propia*

Se genera un ciclo para todo este procedimiento, con un número de iteraciones para los valores desde 1 hasta el total de estaciones. Los mejores resultados para cada caso se guardan en una matriz alterna y al final del ciclo esta matriz es ordenada ascendentemente eligiendo al primer vector y su valor final como solución del problema (ver figura 17).

```

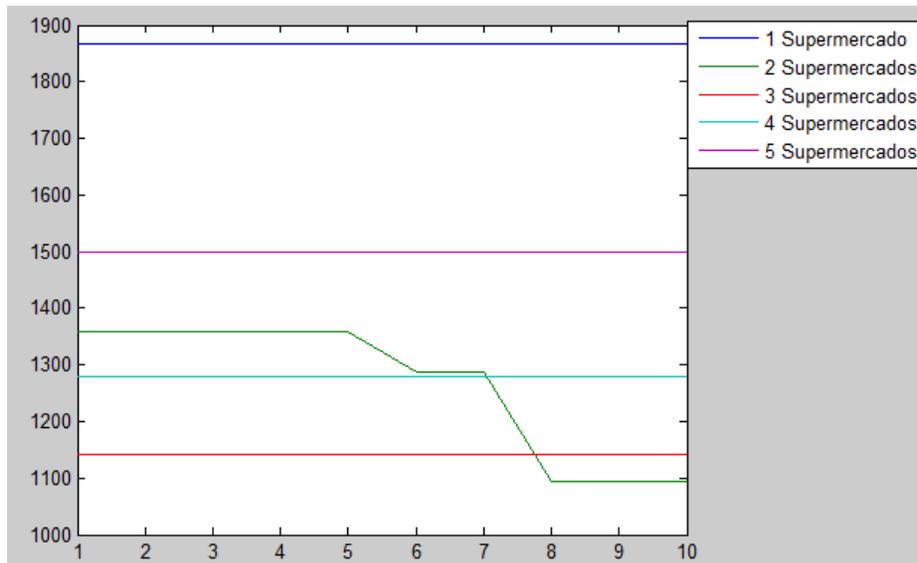
function
[COSTOMINIMO, VECTORSOLUCION]=AutSuper(D,C,CS,NI,Iter,ProbM)
B=size(D,1);
E=size(D,2)+1;
Rts=zeros(E-1,1);
RSol=zeros(E-1,E);
SolV=zeros(E-1,Iter);
for ts=1:E-1
ts;
[Aux,COSTOFINAL,SOLUCION]=GenSuper(D,C,CS,ts,NI,Iter,ProbM);
Rts(ts)=COSTOFINAL;
RSize=size(SOLUCION,2);
for i=1:RSize
RSol(ts,i)=SOLUCION(i);
end
SolV(ts,:)=Aux;
end
[CFS, NS] = sort(Rts);
VECTORSOLUCION=RSol(NS(1),:);
COSTOMINIMO=CFS(1);

```

**Figura 17. Programa para la generación del ciclo del algoritmo genético MATLAB**

*Fuente: Elaboración propia*

En la Figura 18 se muestra el comportamiento de la resolución del algoritmo genético al momento de optimizar en un ciclo de 10 iteraciones, mostrando así que los resultados van disminuyendo (mejorando) hasta llegar al nivel más óptimo dentro del ciclo de iteraciones del problema.



**Figura 18. Comportamiento de resolución del problema con Algoritmos Genéticos**

*Fuente: Elaboración propia*

### 3.4 Obtención de datos para la simulación de resultados

Como no hay datos estadísticos de las pruebas establecidas y no se realizó un trabajo de campo para obtenerlos, se generaron los casos utilizados en nuestro estudio computacional, tal como las manejan los autores. Los datos que se deben de tomar para la simulación son solo tres:

- La demanda de cada estación denotada por  $d_s$
- Las coordenadas en X denotadas por  $a_s$
- Las coordenadas en Y denotadas por  $b_s$

Para la demanda  $d_s$  se define un número al azar de manera uniforme en el intervalo  $[1; 100]$  para cada estación.

Para las coordenadas en la posición  $a_s$  se maneja que  $a_s = a_{s-1} + \text{rnd}_u(1,6)$ ,  $\forall s=2, \dots, S$ , donde  $\text{rnd}_u(1,6)$  significa un número entero dibujado aleatoriamente del intervalo  $[1; 6]$  iniciando con  $a_1 = 1$ .

Para las coordenadas en la posición  $b_s$  se manejan todas como 1.  $b_s$  queda fijo en 1  $\forall s=1, \dots, S$ , por simplicidad de las resoluciones.

Para efectos de simulación el autor maneja diferentes valores del total de supermercado  $S$  y diferentes valores del costo de implementación  $T$ :

$$S = [10, 30, 60, 100, 150, 200, 300]$$

$$T = [500, 1000, 5000, 10000, 25000, 50000]$$

Con estos datos se puede correr una simulación completa y comparar los 2 métodos de solución respecto al tiempo de resolución, al área óptima de resolución y a los límites de variables de entrada posibles del problema (ver Figura 19).

```
function [promedios,promedioT,promedioSM] = ciclo (NI,IT)
S = [10 30 60 100 150 200 300];
T = [500 1000 5000 10000 25000 50000];
totS = size(S,2);
totT = size(T,2);
promedios = zeros(totS, totT);
promedioT = zeros(totS, totT);
promedioSM = zeros(totS, totT);
for i=1:totS
    for j=1:totT
        VCM = zeros(1,40);
        TI= zeros (1,40);
        SFINAL= zeros (1,40);
        for k=1:40
            tic;
```

```

disp([S(i) T(j)]);
[C,D]=Generador(S(i));
[COSTOMINIMO,VECTORSOLUCION]=AutSuper(D,C,T(j),NI,IT,.05);
VCM(k) = COSTOMINIMO;
TI(k)= toc;
ZISE= size(VECTORSOLUCION,2);
CEROS= numel( VECTORSOLUCION(VECTORSOLUCION==0) );
SFINAL(k)=ZISE-CEROS-1;
end
promedios(i,j) = mean(VCM);
promedioT(i,j) = mean(TI);
promedioSM(i,j) = mean (SFINAL);
end
end
end

```

**Figura 19. Programa para resolver ejemplos de los autores en MATLAB**

*Fuente: Elaboración propia*

## CAPÍTULO 4. ANÁLISIS Y DISCUSIÓN

### 4.1 Resultados

Para poder realizar una comparación totalmente fidedigna de este trabajo con los resultados de los autores, se deben desarrollar todos los valores de entrada de la misma manera en que los autores los realizan, resolver el problema por el método propuesto en este trabajo y comparar los resultados. Los datos que se deben de tomar para la simulación anteriormente se explica cómo es que se desarrollan aleatoriamente.

Los autores muestran sus resultados en la tabla 6 y abarcan la solución de todos los valores que proponen inicialmente para el costo fijo de supermercados ( $T$ ) y para el número de supermercados ( $S$ ) que se desean implementar, mostrando el número promedio de supermercados implementados (*Avg.  $n$* ), del costo final de la función para la implementación ( $F^*$ ) y el tiempo computacional de la resolución del programa en segundos (*Avg. CPU Time*). Estos son los tres valores se comparan con el resultado de este proyecto.

En el caso de este proyecto, se realiza la programación en el software MATLAB, donde la programación es compleja, las simulaciones se vuelven lentas y existe el problema de no poder resolver todas las variables propuestas por los autores ya que si se realizan podría llevar demasiado tiempo computacional para obtener los resultados, esto debido a la plataforma del software.

En base a esta problemática se resuelve y se toma de referencia solo a los primeros tres de los siete grupos totales de resultados de los autores para realizar la comparación y poder tomar conclusiones de este trabajo.

**Tabla 6. Resultados de los autores**

Results for all tested fixed cost values  $\Gamma$ .

$S$	$\Gamma$	Avg. n	Avg. number of vehicles	$P^*$	Avg. cost as per simulation	Avg. CPU time (s)
10	500	3.73	3.73	2771.22	4803.33	< 0.1
10	1000	2.8	2.8	4357.61	4188.67	< 0.1
10	5000	1.17	1.17	11,173.61	4665.73	< 0.1
10	10,000	1	1	16,526.5	5,117.8	< 0.1
10	25,000	1	1	31,526.5	5117.8	< 0.1
10	50,000	1	1	56,526.5	5117.8	< 0.1
30	500	10.7	10.7	7982.84	14,378.23	< 0.1
30	1000	7.8	7.8	12,499.6	12,411.83	< 0.1
30	5000	3.8	3.8	32,922.77	12,761.7	< 0.1
30	10,000	2.67	2.73	48,691.26	16,042.57	< 0.1
30	25,000	1.83	2.87	80,382.58	22,756.8	< 0.1
30	50,000	1.13	3.63	116,005.9	34,434.13	< 0.1
60	500	21.1	21.1	16,058.38	28,645.87	< 0.1
60	1000	15.73	15.73	25,057.19	25,259.97	< 0.1
60	5000	7.47	7.47	65,088.24	25,993.63	< 0.1
60	10,000	5.33	5.7	95,837.45	31,137	< 0.1
60	25,000	3.37	6.43	157,750.9	46,514.83	< 0.1
60	50,000	2.4	7	228,267.6	63,660.4	< 0.1
100	500	34.53	34.53	26,309.7	46,992.9	0.1
100	1000	25.53	25.53	41,032.99	41,238.03	0.1
100	5000	12.17	12.17	106,582	43,076.97	0.1
100	10,000	8.63	9.3	156,962.7	52,176.57	0.1
100	25,000	5.37	10.87	258,005.1	76,691.13	0.1
100	50,000	3.8	12.07	372,099.6	102,165.1	0.1
150	500	53.03	53.03	40,109.06	72,638.27	0.4
150	1000	39.63	39.63	62,656.12	63,997.8	0.4
150	5000	18.9	18.9	163,796.6	66,756.9	0.4
150	10,000	13.33	14.33	241,474.1	81,570.47	0.4
150	25,000	8.3	16.53	396,726.5	118,903.6	0.4
150	50,000	6	18	573,081.4	154,177.7	0.4
200	500	69.7	69.7	53,379.96	95,643.57	1.3
200	1000	51.8	51.8	83,182.58	84,201.93	1.3
200	5000	24.67	24.67	217,021.4	88,279.53	1.3
200	10,000	17.77	19.23	320,260.3	107,645.8	1.3
200	25,000	11.13	21.8	525,494.6	157,081.3	1.3
200	50,000	7.9	23.83	758,126.4	203,389	1.3
300	500	105.03	105.03	80,066.35	143,652.8	6.3
300	1000	78.47	78.47	125,042.5	126,822.8	6.3
300	5000	36.93	36.93	324,783.1	132,538.3	6.3
300	10,000	26.3	28.37	478,524.3	161,769.6	6.3
300	25,000	16.87	32.9	784,418.7	230,787.4	6.3
300	50,000	11.93	35.43	1,131,835	301,119.6	6.3

*Fuente: Simon Emde & Nils Boysen, 2012. Optimally locating in-house logistics areas to facilitate JIT-supply of mixed-model assembly lines*

Para la simulación de la propuesta en este trabajo se manejan tres valores adicionales de entrada para la resolución por algoritmos genéticos que son:

-*NI*: números de individuos a introducir en la simulación

-*Iter*: número de iteraciones o ciclos del programa de simulación

-*ProbM*: que es la probabilidad de mutación para cada vector del resultado óptimo cruzado

Para los valores constantes de entrada de esta simulación se manejan *NI* = 100, *Iter* = 40 y *ProbM* = 0.05, se realizan 5 repeticiones o corridas de simulación. Los resultados de cada corrida se muestran en la tabla 7 y su comportamiento en la gráfica 8.

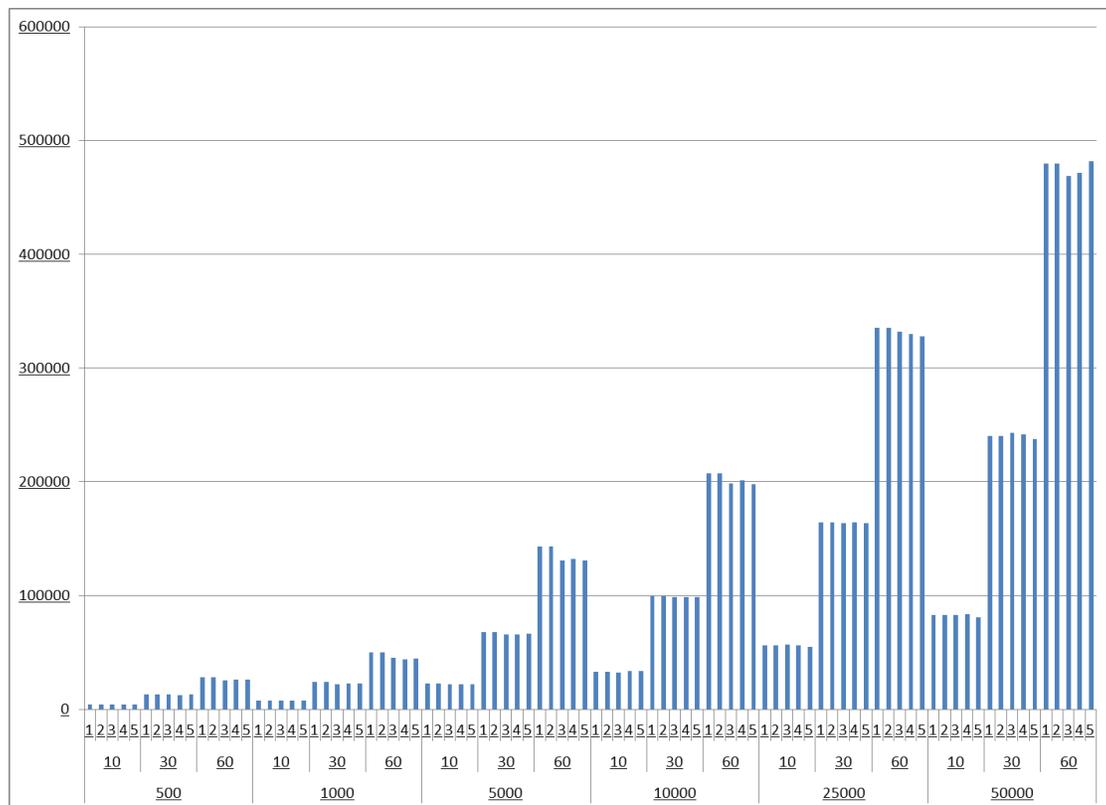
**Tabla 7. Resultados del trabajo de simulación**

S	T	NI	Iter	Prob. Mut.	Corridas	Simulación MATLAB		
						Avg. Cost per simulation	Avg. CPU Time (s)	Avg. n
10	500	40	100	5%	1	4,500.00	7.67	7.08
10	1,000	40	100	5%	1	7,430.00	8.20	5.58
10	5,000	40	100	5%	1	22,370.00	8.13	2.73
10	10,000	40	100	5%	1	32,720.00	7.85	2.00
10	25,000	40	100	5%	1	55,930.00	7.79	1.03
10	50,000	40	100	5%	1	82,820.00	7.77	1.00
30	500	40	100	5%	1	13,440.00	104.02	21.23
30	1,000	40	100	5%	1	23,780.00	101.76	16.90
30	5,000	40	100	5%	1	67,680.00	101.81	7.75
30	10,000	40	100	5%	1	100,130.00	101.66	5.73
30	25,000	40	100	5%	1	164,010.00	101.80	3.70
30	50,000	40	100	5%	1	240,190.00	101.80	2.68
60	500	40	100	5%	1	27,990.00	629.57	43.78
60	1,000	40	100	5%	1	50,090.00	627.68	33.98
60	5,000	40	100	5%	1	143,210.00	627.80	16.05
60	10,000	40	100	5%	1	207,510.00	627.76	11.03
60	25,000	40	100	5%	1	335,520.00	627.93	6.98
60	50,000	40	100	5%	1	479,720.00	627.66	5.13
10	500	40	100	5%	2	4,500.00	0.77	7.65
10	1,000	40	100	5%	2	7,430.00	0.77	5.60
10	5,000	40	100	5%	2	22,370.00	0.75	2.78
10	10,000	40	100	5%	2	32,720.00	0.75	2.00
10	25,000	40	100	5%	2	55,930.00	0.75	1.03

10	50,000	40	100	5%	2	82,820.00	0.75	1.00
30	500	40	100	5%	2	13,440.00	9.67	23.03
30	1,000	40	100	5%	2	23,780.00	9.66	18.15
30	5,000	40	100	5%	2	67,680.00	9.66	7.88
30	10,000	40	100	5%	2	100,130.00	9.65	5.43
30	25,000	40	100	5%	2	164,010.00	9.66	3.50
30	50,000	40	100	5%	2	240,190.00	9.65	2.65
60	500	40	100	5%	2	27,990.00	59.64	50.45
60	1,000	40	100	5%	2	50,090.00	59.60	37.70
60	5,000	40	100	5%	2	143,210.00	59.61	16.58
60	10,000	40	100	5%	2	207,510.00	59.61	11.03
60	25,000	40	100	5%	2	335,520.00	59.61	7.03
60	50,000	40	100	5%	2	479,720.00	59.59	4.95
10	500	40	100	5%	3	4,380.00	7.66	7.10
10	1,000	40	100	5%	3	7,650.00	7.65	5.78
10	5,000	40	100	5%	3	21,990.00	7.64	2.80
10	10,000	40	100	5%	3	32,630.00	7.63	1.95
10	25,000	40	100	5%	3	56,910.00	7.61	1.08
10	50,000	40	100	5%	3	82,620.00	7.62	1.00
30	500	40	100	5%	3	12,820.00	101.69	21.13
30	1,000	40	100	5%	3	22,240.00	101.61	16.60
30	5,000	40	100	5%	3	65,680.00	101.60	8.08
30	10,000	40	100	5%	3	98,360.00	101.74	5.68
30	25,000	40	100	5%	3	163,600.00	101.57	3.60
30	50,000	40	100	5%	3	242,650.00	101.68	2.68
60	500	40	100	5%	3	25,770.00	626.49	43.05
60	1,000	40	100	5%	3	45,590.00	626.61	33.93
60	5,000	40	100	5%	3	130,540.00	627.98	16.00
60	10,000	40	100	5%	3	198,460.00	627.47	11.25
60	25,000	40	100	5%	3	331,990.00	627.43	7.18
60	50,000	40	100	5%	3	468,980.00	628.01	4.90
10	500	40	100	5%	4	4,320.00	7.67	7.23
10	1,000	40	100	5%	4	7,630.00	7.66	5.48
10	5,000	40	100	5%	4	21,940.00	7.61	2.70
10	10,000	40	100	5%	4	33,700.00	7.60	2.00
10	25,000	40	100	5%	4	56,430.00	7.62	1.03
10	50,000	40	100	5%	4	83,890.00	7.61	1.00
30	500	40	100	5%	4	12,770.00	101.99	21.15
30	1,000	40	100	5%	4	22,660.00	101.95	17.03
30	5,000	40	100	5%	4	65,990.00	101.85	8.08
30	10,000	40	100	5%	4	98,880.00	101.96	5.75
30	25,000	40	100	5%	4	163,990.00	101.92	3.65
30	50,000	40	100	5%	4	241,670.00	102.13	2.63
60	500	40	100	5%	4	26,050.00	625.80	43.50
60	1,000	40	100	5%	4	44,180.00	625.56	33.38
60	5,000	40	100	5%	4	132,260.00	625.71	16.15
60	10,000	40	100	5%	4	201,400.00	625.84	11.50
60	25,000	40	100	5%	4	330,090.00	625.70	7.23

60	50,000	40	100	5%	4	471,780.00	625.65	4.98
10	500	40	100	5%	5	4,310.00	7.77	7.15
10	1,000	40	100	5%	5	7,490.00	8.17	5.53
10	5,000	40	100	5%	5	22,230.00	7.80	2.78
10	10,000	40	100	5%	5	33,910.00	7.64	1.98
10	25,000	40	100	5%	5	55,190.00	7.63	1.05
10	50,000	40	100	5%	5	81,020.00	7.60	1.00
30	500	40	100	5%	5	12,920.00	101.73	21.20
30	1,000	40	100	5%	5	22,500.00	101.86	16.63
30	5,000	40	100	5%	5	66,820.00	101.78	8.10
30	10,000	40	100	5%	5	98,550.00	101.67	5.70
30	25,000	40	100	5%	5	163,810.00	101.79	3.65
30	50,000	40	100	5%	5	237,590.00	102.03	2.63
60	500	40	100	5%	5	25,980.00	625.48	43.70
60	1,000	40	100	5%	5	44,760.00	625.19	32.98
60	5,000	40	100	5%	5	130,960.00	626.80	16.13
60	10,000	40	100	5%	5	197,630.00	626.62	10.90
60	25,000	40	100	5%	5	327,950.00	627.30	7.20
60	50,000	40	100	5%	5	481,620.00	627.61	5.15

Fuente: Elaboración propia



**Gráfica 8. Comparación de Resolución de la Simulación**

Fuente: Elaboración propia

## 4.2 Análisis

El promedio de las corridas que se muestra en los resultados del trabajo respecto al costo y la comparación de los resultados de simulación de los autores se puede ver en la última columna de la tabla 8, se puede ver la diferencia del costo final de implementación comparando los resultados del trabajo con el de los autores, este es el valor principal que se desea minimizar, los demás valores sirven de referencia para comparar solo variables adicionales en la implementación. En esta columna se puede observar que en ninguno de los casos se encuentra una solución más óptima a comparación de la de los autores, teniendo en total una diferencia a la de los autores con un promedio de \$92,786.33 en los resultados, con los datos obtenidos anteriormente se puede determinar que la resolución por este método con los algoritmos aplicados no es totalmente viable.

**Tabla 8. Resultados del trabajo y comparación al de los autores**

Valores de Simulación		Simulación MATLAB			Resultados Artículo			Diferencia
S	T	Avg. n	Avg. Cost per simulation	Avg. CPU Time (s)	Avg. n	Avg. Cost per simulation	Avg. CPU Time (s)	
10	500	7.24	\$4,402.00	6.31	3.73	\$2,771.22	0.10	<b>-\$1,630.78</b>
10	1,000	5.59	\$7,526.00	6.49	2.80	\$4,357.61	0.10	<b>-\$3,168.39</b>
10	5,000	2.76	\$22,180.00	6.39	1.17	\$11,173.61	0.10	<b>-\$11,006.39</b>
10	10,000	1.99	\$33,136.00	6.29	1.00	\$16,526.50	0.10	<b>-\$16,609.50</b>
10	25,000	1.04	\$56,078.00	6.28	1.00	\$31,526.50	0.10	<b>-\$24,551.50</b>
10	50,000	1.00	\$82,634.00	6.27	1.00	\$56,526.50	0.10	<b>-\$26,107.50</b>
30	500	21.55	\$13,078.00	83.82	10.70	\$7,982.83	0.10	<b>-\$5,095.17</b>
30	1,000	17.06	\$22,992.00	83.37	7.80	\$12,499.60	0.10	<b>-\$10,492.40</b>
30	5,000	7.98	\$66,770.00	83.34	3.80	\$32,922.77	0.10	<b>-\$33,847.23</b>
30	10,000	5.66	\$99,210.00	83.34	2.67	\$48,691.26	0.10	<b>-\$50,518.74</b>
30	25,000	3.62	\$163,884.00	83.35	1.83	\$80,382.58	0.10	<b>-\$83,501.42</b>
30	50,000	2.65	\$240,458.00	83.46	1.13	\$116,005.90	0.10	<b>-\$124,452.10</b>
60	500	44.90	\$26,756.00	513.39	21.10	\$16,058.38	0.10	<b>-\$10,697.62</b>
60	1,000	34.39	\$46,942.00	512.93	15.73	\$25,057.19	0.10	<b>-\$21,884.81</b>
60	5,000	16.18	\$136,036.00	513.58	7.47	\$65,088.24	0.10	<b>-\$70,947.76</b>
60	10,000	11.14	\$202,502.00	513.46	5.33	\$95,837.45	0.10	<b>-\$106,664.55</b>
60	25,000	7.12	\$332,214.00	513.59	3.37	\$157,750.90	0.10	<b>-\$174,463.10</b>
60	50,000	5.02	\$476,364.00	513.71	2.40	\$228,267.60	0.10	<b>-\$248,096.40</b>

Fuente: Elaboración propia

### 4.3 Discusión

Antes de empezar con la conclusión de la propuesta del nuevo método de resolución, se debe de ir por partes y de manera ordenada, lo primero es ir resolviendo y entendiendo cada uno de los modelos matemáticos de los autores en el programa de MATLAB, para poder entender cómo es que funcionan y como es que determinan sus valores de solución. Todo esto se logró simular en el software y se comprobó su funcionamiento con el ejemplo que los autores manejan. Una vez que se entendió como es que funcionan y como se resuelven las funciones propuestas por los autores, se busca la manera de transferir esas funciones hacia el programa conforme al nuevo método de resolución propuesto de los Algoritmos Genéticos. Se generan ciclos automatizados de esta simulación y se pone a prueba, otra vez con el ejemplo que manejan los autores para comprobar su funcionamiento.

Ya que se comprueba el funcionamiento y la resolución de las funciones matemáticas de los autores y el funcionamiento del método de resolución propuesto, se debe de generar una comparación con una base de datos más robusta. Esta base de datos es generada conforme exactamente las condiciones que proponen los autores para generarse aleatoriamente y es resuelta por el método propuesto en este trabajo de Algoritmos Genéticos. Los resultados se mostraron anteriormente en la Tabla 7 donde, se puede concluir con evidencia suficiente que el método propuesto sustancialmente no mejora de forma general el problema de implementación de sistemas de supermercado. No se excluye totalmente que el método no funcione, se puede concluir que puede resolverse y ponerse a prueba con otro tipo de clasificación de los Algoritmos Genéticos. Se necesita hacer una re evaluación y retroalimentación del programa del software y ver si puede hacerse de una manera más esbelta para mejorar los resultados.

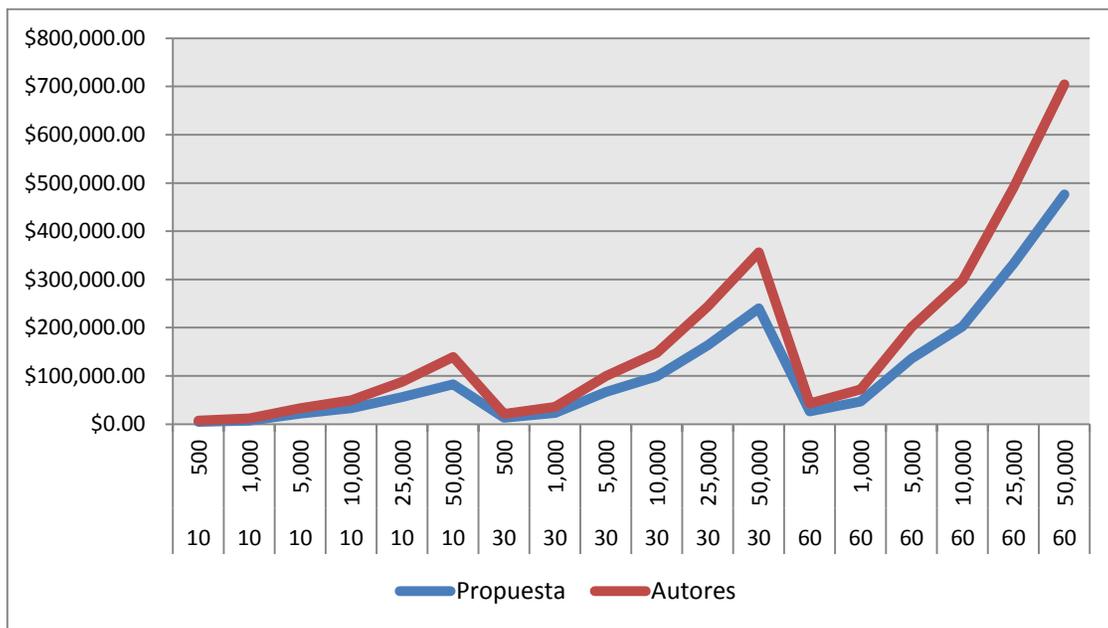
## CAPÍTULO 5. CONCLUSIONES

### 5.1 Conclusiones relativas a los objetivos específicos y al general

Este trabajo tiene principalmente dos objetivos a demostrar, el primero es comprobar que por medio del método de simulación y resolución de los Algoritmos Genéticos se pueden resolver los modelos matemáticos propuestos de los autores Simon Emde y Nils Boysen en su trabajo de implementación de supermercados en una línea de producción y el segundo y más importante es el de demostrar que por este método se pueden obtener áreas de solución más óptimas en comparación con la resolución por medio de programación lineal realizada por los autores. A continuación se presentan la explicación del resultado final y las conclusiones de este trabajo.

En la gráfica 9 fácilmente se puede llegar a la conclusión de los resultados obtenidos en este trabajo están en un área de solución óptima más alejada a la resolución de los autores. Se puede observar que la respuesta es similar en cuanto al comportamiento de la gráfica, comprobando aquí nuestro primero objetivo de poder resolver el problema de los autores por un método alternativo, pero la resolución propuesta tiene un área de solución óptima más alta que la de los autores, donde el objetivo de mejorar los resultados no se cumple ya que se pretendía demostrar lo contrario en este proyecto.

Con esto no se da por hecho que el método de solución de los Algoritmos Genéticos no es funcional, sino que se está manifestando que el algoritmo resuelto por la programación propuesta no es totalmente práctica y funcional para este modelo.



**Gráfica 9. Diferencia en el área de solución óptima de la solución propuesta contra la de los autores.**

*Fuente: Elaboración propia*

En algunas partes de la gráfica se observa que las líneas casi coinciden pero mientras el problema va teniendo valores de entrada más grandes la diferencia en los resultados es cada vez mayor y su comportamiento es similar y cíclico.

Existen diferentes versiones dentro de la clasificación de los Algoritmos Genéticos y no porque con un método de resolución dentro de todas las clasificaciones salió mal, podemos deducir que en general este método de solución es totalmente erróneo para este modelo.

Se puede poner a prueba de estudio el algoritmo de simulación desarrollado en este trabajo y verificar si tiene errores y también comprobar si cumple con todos los requisitos que se requieren para que se pueda considerar como un algoritmo genético.

Se concluye que el método de resolución propuesto en este trabajo a comparación del de los autores es funcional al seguir la misma tendencia y comportamiento en las gráficas al utilizar los mismos datos de entrada del problema pero también es evidente que el área de solución óptima de la propuesta es más alta conforme los datos de entrada van creciendo y se comparan con la solución de los autores.

Se puede demostrar así que la resolución del problema por Algoritmos Genéticos resuelta con la programación propuesta no es totalmente factible y funcional para la puesta en práctica de un proyecto de instalación de sistemas de supermercado debido a la diferencia de los costos poniendo así en marcha una propuesta que no es totalmente factible para la empresa.

Se debe de tomar en cuenta de que este problema se resuelve con la metodología de los Algoritmos Genéticos pero las restricciones tienen ciertas características únicas que lo hacen único en su desarrollo, éstas se explican a detalle en el apartado de aportaciones originales y estas características hacen que la resolución en general del problema presente varias dificultades como lo son la programación para la simulación y la cuestión del tiempo de resolución como se percibió en los resultados del estudio.

Se puede especular que al tener estas características únicas en las restricciones del problema que las diferencian de los métodos comunes de resolución de los Algoritmos Genéticos lleguen a afectar los resultados y no se alcancen mejores áreas óptimas de solución como generalmente lo logran este tipo de implementaciones en comparación de otros métodos de solución en la literatura ya que el problema es resuelto con una metodología ya establecida pero con restricciones totalmente únicas.

## 5.2 Aportaciones originales

El problema es similar al de una ruta crítica o un agente viajero, tiene características que lo hacen propio y único de la resolución común de Algoritmos Genéticos resueltos en la literatura como lo son:

1. La ruta tiene que ir en orden ascendente y no puede repetir números.
2. El valor inicial de la ruta siempre debe de ser 1 y el valor final del vector siempre debe de ser igual al número de estaciones que se tienen.
3. Para la combinación de valores el primer y el último valor del vector no se deben de tomar en cuenta.
4. Para la probabilidad de mutación el primer y el último valor del vector no se deben de tomar en cuenta.

Con el tipo de resolución de Algoritmos Genéticos con las características únicas arriba mencionadas podrían ser una nueva variante o clasificación de los métodos para alcanzar la óptima área de solución de un problema que se asemeje al presente en este trabajo y donde hay muy poca información de investigación al respecto como se explica en la parte del planteamiento del problema.

Al expandir la visión de la implementación practica este trabajo y dejar las plantas de producción, este podría aplicar para la localización de centros de distribución o plantas de producción, siempre y cuando las características arriba mencionadas se ajusten a las del problema.

En conclusión este estudio abre más el panorama sobre la situación actual de la implementación de los sistemas de supermercados pero principalmente de los

Algoritmos Genéticos, ya que existe muy poco sustento matemático o de información para su resolución. Desde un inicio se pretendió llegar a un nuevo método de implementación y resolución ya que existe poca información respecto al tema.

### **5.3 Límites del modelo planteado**

En este trabajo no se determinan o se toman datos reales de algún ejemplo, el objetivo de este trabajo es comparar los resultados de los autores con los obtenidos por nuestro método propuesto, por lo que en ningún caso de la generación de las variables de entrada aleatoria o valores fijos fueron modificados con el fin de tener un resultado totalmente basado en las mismas condiciones y digno de comparar.

Todos los valores son creados aleatoriamente siguiendo las condiciones que los autores proponen sin ninguna modificación. Se plantea solo realizar la solución del problema mediante un método alternativo al de los autores que es el de los Algoritmos Genéticos. No se genera ningún otro tipo de programa para una tercera comparación.

### **5.4 Recomendaciones para futuros estudios**

Como futuras recomendaciones se pueden tomar los siguientes puntos:

- Se puede seguir estudiando y mejorar el algoritmo propuesto para confirmar que realmente puede funcionar o no como objetivo propuesto de este trabajo.
- Se puede resolver el modelo propuesto por los autores por otro tipo de método que este dentro de la clasificación de los Algoritmos Genéticos.

- Se puede proponer mejorar el modelo de los autores por cualquier otro de los nuevos métodos de solución que se están estudiando en la simulación como ejemplo: algoritmo gravitacional, algoritmo de abejas, algoritmo de hormigas etc.
- Se puede continuar resolviendo las demás funciones de los autores propuestas en sus otros 4 artículos publicados donde van mostrando evolución en la implementación de los supermercados en la industria.
- Se pueden modificar los modelos matemáticos propuestos por los autores aumentando más variables para su aplicación, donde se debe de demostrar y resolver esos nuevos modelos modificados.
- Se puede continuar investigando para verificar si existen nuevos modelos propuestos por otros autores en temas para la implementación de supermercados

## Bibliografía

- A Duarte-Muñoz, J. P.-F.-C. (2007). *Metaheurísticas*. Madrid: Dikonsón. ISBN 978-84-9849-016-9.
- Battini D, N. B. (2013). Just-in-Time supermarkets for part supply in the automobile industry. *Springer Verlag J Manag Control* 24, 209–212.
- Bellman RE. (1952). *Dynamic Programming*. Princeton, New Jersey, USA: Princeton University Press.
- Boysen N, B. S. (2011). Scheduling just-in-time part supply for mixed-model assembly lines. *European Journal of Operational Research* 211, 15-25.
- Brown KA, M. T. (1991). A comparison of just in time and batch manufacturing the role of performance obstacles. *Acad Manage* 34, 906–912.
- Cusumano D. (1994). The limits of lean. *Solan Management Review*. Vol. 35, No. 4., 27-32.
- Emde S, B. N. (2012). Optimally locating in-house logistics areas to facilitate JIT-supply of mixed-model assembly lines. *ELSERVIER Int. J. Production Economics* 135, 393–402.
- Emde S, B. N. (2012). Optimally routing and scheduling tow trains for JIT-supply of mixed-model assembly lines. *ELSERVIER European Journal of Operational Research* 212, 282-299.
- Emde S, B. N. (2014). Scheduling the part supply of mixed- model assembly lines in line- integrated supermarkets. *European Journal of Operational Research*, Volume 239, Issue 3, 820–829.
- Fogel, D. B. (1995). *Evolutionary Computation*. IEEE Press.
- Galindo, A. V. (2007). *Manual de Lean Manufacturing*. México: Limusa.
- García, J. A. (01 de 11 de 2013). *Laboratorio Nacional de Informatica Avanzada A.C.* Recuperado el 06 de 07 de 2016, de <http://www.lania.mx/~emezura/util/files/thesis-josealfredo.pdf>

- Gill PE, M. W. (1981). Practical Optimization. Academic Press.
- Goldberg D. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Publishing Company, INC.
- Golz J, G. R.-O. (2012). Part feeding at high-variant mixed-model assembly lines. Flexible Services and Manufacturing Journal Volume 24, Issue 2,, 119-141.
- Hé Hernández Sampieri R, F. C. (2006). Metodología de la investigación. México: Mc Graw-Hill.
- Holland, J. H. (1975). Adaptation in natural and artificial systems. Ann Arbor: The.
- Limère V. (2013). To kit or not to kit: optimizing part feeding in the automotive assembly industry. SPRINGERLINK, Volume 11, Issue 1, 92-98.
- Marvel, C. S. (2006). Why lean needs simulation. IEEE, 1907-1913.
- Mason PA. (1999). MRP II and Kan-Ban formula. Logist Focus 2, 19–23.
- Michalewicz, Z. &. (1992). A Modified Genetic Algorithm for Optimal Controls Problems. Computers & Mathematics with Applications, 83-94.
- Monden Y. (1983). Toyota Production Systems: Practical Approach to Production Management. Atlanta, GA.: Industrial Engineering and Management Press.
- Nauss, R. (2003). Solving the Generalized Assignment Problem: An Optimizing and Heuristic Approach. INFORMS Journal on Computing, 15, 249-266.
- Ohno T. (1988). Toyota Production System: Beyond Large Scale Production. Productivity Press: Cambridge, MA.
- Pacheco A, C. M. (2006). Metodología crítica de la investigación. México, México: CECSA.

Padilla, L. (2010). Lean Manufacturing. Revista Electrónica Ingeniería Primero. Enero No.15., 64-69.

Rajgopal, A. y. (2006). Analyzing the benefits of lean manufacturing and value stream mapping via simulation: a process sector case study. International Journal of production economics. Vol. 107, 223-236.

Ramesh VN. (2010). Balancing WIP and Throughput with Machine Utilization in a Manufacturing Facility. Proc. of. Int. Conf. on Advances in Mechanical Engineering, 10-14.

S Kim, L. H. (2008). Robust production control policies considering WIP balance and setup time in a semiconductor fabrication line. Int J Adv Manuf Technol, 333-343.

SC Liu, C. L. (2005). A heuristic method for the combined location routing and inventory problem. Int J Adv Manuf Technol 26, 372-381.

Sendil Kumar C, P. R. (2002). Literature review of JIT-KANBAN System. Int Adv Manuf Technol 32, 393-408.

Shah R, W. P. (2003). Lean manufacturing: context, practice bundles and performance. Journal of operations management. Vol. 21 No. 2, 129-149.

Sivaraj R, R. T. (2011). A review of selection methods in genetic algorithm. International Journal of Engineering Science and Technology (IJEST), 3292-3298.

Sohal, A. S. (1992). Implementation of JIT in Small Manufacturing Firm . Production and Inventory Management Journal .

Springer Link. (2013). Springer Link. Recuperado el 2013, de Springer Link: <http://link.springer.com/>

Tang KS, M. K. (1994). Fuzzy Control of Water Pressure using Genetic Algorithm. IFAC Workshop on Safe, Reliability and Applications of Emerging Intelligent Control Technologies, 15-20.

Towill Denis R. (2010). Industrial engineering the Toyota Production System. Journal of Management History, Vol. 16 Iss: 3, 327-345.

Vatalaro J, T. R. (2005). Implementing a Mixed Model Kanban System: The Lean Replenishment Technique for Pull Production. Portland, OR.: Productivity Press.

Villanueva, C. (2008). Simulación de procesos clave en la toma de decisiones. DYNA Vol. 83, No.4, 221-227.

Wilson, R. H. (2003). Making Materials Flow. The Lean Enterprise Institute: Massachusetts.

Womack JP. (1991). The machine that change the world the story of lean *production*. New York: Rawson press.