

UNIVERSIDAD DEL BÍO-BÍO  
FACULTAD DE INGENIERÍA  
DPTO. DE INGENIERÍA INDUSTRIAL

Profesor Guía:  
Luís Ceballos A

# “REASIGNACION DE CAMIONES PARA EL TRANSPORTE DE PRODUCTOS FORESTALES MEDIANTE ALGORITMOS GENETICOS”

---

“PROYECTO DE TÍTULO PRESENTADO EN CONFORMIDAD A LOS  
REQUISITOS PARA OBTENER EL TÍTULO DE INGENIERO CIVIL  
INDUSTRIAL MENCIÓN GESTIÓN”

Autor: Maichel Miguel Aguayo Bustos

Concepción, 2009

## RESUMEN

En Chile, la industria forestal es completamente privada, con una alta concentración en dos grandes firmas; Arauco y Mininco, las que poseen aproximadamente la mitad de las plantaciones del país y que verticalmente integran plantas de celulosa, aserraderos y papeleras. Diariamente en las faenas forestales se deben transportar diferentes productos desde los distintos orígenes en los predios hasta diferentes destinos determinados. Las empresas forestales subcontratan el servicio de transporte a diferentes empresas de servicios llamadas EMSEFOR. Las EMSEFOR perciben sus ingresos por cada kilómetro recorrido con carga desde un origen determinado a un destino cualquiera. El costo del recorrido siguiente a realizar, es decir el retorno por una nueva carga ya sea al mismo origen u otro es asumido íntegramente por la empresa que presta el servicio.

A nivel país, las empresas mandantes utilizan un Sistema de Asignación de Camiones (Asicam) para la programación del transporté forestal. Esta programación tiene diferentes problemas asociados tales como: sobre carga de algunos camiones, jornada extensas de trabajo y tramos largo de recorrido sin carga. Asicam define una ventana de tiempo a todos sus despachos (Viajes). La idea en este estudio, es reasignar los viajes a los distintos camiones con el objetivo de minimizar los kilómetros recorridos sin carga. El problema a resolver (NP-hard) se puede interpretar como un problema de programación de la producción de “n” trabajos sobre “m” máquinas paralelas idénticas con tiempos de preparación o setup dependientes.

Se proponen dos metaheurísticas basadas en algoritmos genéticos para optimizar la programación de camiones para el transporte forestal. El primer algoritmo genético (GA) utiliza los principios de la evolución genética, mientras el segundo algoritmo (GALS) combina la evolución genética con búsqueda local. Los algoritmos reducen los kilómetros recorridos sin carga en un 31 por ciento y aumentando la productividad por camión en un 25 por ciento.

## DEDICATORIA

### Para

Mi mejor Amigo que me ha apoyado y guiado durante toda mi vida, **Jesús**.  
“Si tengo tu amistad lo tengo todo, pues estás dentro de mí”.

### Para

Mis padres, Irma y Froilan. Por su sacrificio, apoyo incondicional y por todos los valores inculcados.

## ÍNDICE DE CONTENIDO

<b>1</b>	<b>CAPITULO I: INTRODUCCION</b>	<b>8</b>
<b>2</b>	<b>CAPITULO II: "SITUACION DEL SECTOR Y TRANSPORTE FORESTAL".</b>	<b>11</b>
2.1	Sector forestal	11
2.1.1	Disponibilidad Recursos	11
2.1.2	Mercado Forestal	12
2.1.3	Ciclo Forestal	14
2.1.4	Software aplicados en la Industria Forestal	15
2.2	Transporte Forestal	16
2.2.1	Asicam	17
2.2.2	Resultados Asicam	18
2.2.3	Desventajas Asicam	20
2.3	Descripción del problema	20
2.4	Objetivos del estudio	23
2.4.1	Objetivo General	23
2.4.2	Objetivo específico	23
<b>3</b>	<b>CAPITULO III: MARCO TEORICO</b>	<b>24</b>
<b>3.1</b>	<b>Introducción a la Optimización</b>	<b>24</b>
<b>3.2</b>	<b>Metaheurísticas de Optimización</b>	<b>25</b>
3.2.1	Concepto de Metaheurística	26
3.2.2	Clasificación Técnicas Metaheurísticas	28
3.2.3	Técnicas Metaheurísticas basadas en trayectoria	30
3.2.4	Técnicas Metaheurísticas basadas en población	32
<b>3.3</b>	<b>Metaheurísticas Usadas</b>	<b>34</b>
3.3.1	Algoritmos Evolutivos	34
3.3.2	Tipos de Algoritmos Evolutivos	40
3.3.3	Algoritmos Genéticos	40
<b>4</b>	<b>CAPITULO IV: REASIGNACION CAMIONES USANDO ALGORITMOS GENETICOS</b>	<b>47</b>
<b>4.1</b>	<b>Introducción</b>	<b>47</b>
<b>4.2</b>	<b>Problema Reasignación Camiones</b>	<b>47</b>
4.2.1	Modelo Matemático	47
<b>4.3</b>	<b>Origen de los Datos</b>	<b>50</b>
4.3.1	Matriz distancia entre orígenes y destinos	50
4.3.2	Programa entregado por ASICAM	50
<b>4.4</b>	<b>Información de entrada a los algoritmos</b>	<b>51</b>
4.4.1	Matriz de Distancia	51
4.4.2	Matriz Tiempos Viajes en vacío	51

<b>4.5</b>	<b>Diseño Algoritmo Genético (GA)</b>	52
4.5.1	Descripción Algoritmo Genético (GA)	52
4.5.2	Representación soluciones	53
4.5.3	Inicio Población	54
4.5.4	Evaluación y Asignación Fitnes	54
4.5.5	Reproducción	54
4.5.6	Cruzamiento	56
4.5.7	Mutación	59
<b>4.6</b>	<b>Diseño Algoritmo Genético con Búsqueda Local (GALS)</b>	60
4.6.1	Descripción Algoritmo Genético con Búsqueda Local (GALS)	60
4.6.2	Búsqueda Local	61
<b>5</b>	<b>CAPITULO V: ANALISIS EXPERIMENTAL Y RESULTADOS</b>	63
<b>5.1</b>	<b>Análisis Experimental</b>	63
5.1.1	Ejecución Algoritmos	63
5.1.2	Parámetros	63
5.1.3	Diseño de experimentos	64
<b>5.2</b>	<b>Resultados obtenidos por los algoritmos</b>	69
5.2.1	Algoritmo Genético (GA)	69
5.2.2	Algoritmo Genético Híbrido (GALS)	70
5.2.3	Análisis comparativo	70
5.2.4	Óptimos Generales	71
5.2.5	Tiempo Ejecución	72
<b>5.3</b>	<b>Resultados Economicos</b>	73
<b>6</b>	<b>CAPITULO VI: CONCLUSIONES Y TRABAJO FUTURO</b>	74
<b>7</b>	<b>BIBLIOGRAFIA Y REFERENCIAS</b>	77
<b>8</b>	<b>ANEXOS</b>	

## INDICE DE FIGURAS

<b>Figura 2.1:</b> Uso actual del suelo .....	11
<b>Figura 2.2:</b> Evolución de las Exportaciones chilenas .....	13
<b>Figura 2.3:</b> Ciclo Forestal .....	15
<b>Figura 2.4:</b> Ilustración problema a resolver .....	21
<b>Figura 3.1:</b> Clasificación Técnicas Optimización.....	24
<b>Figura 3.2:</b> Clasificación de las meta heurísticas .....	30
<b>Figura 3.3:</b> Funcionamiento de un EA [18].....	36
<b>Figura 3.4:</b> Pseudo-Código Algoritmo Evolutivo.....	37
<b>Figura 3.5:</b> Estructura Genética de un individuo en EAs .....	39
<b>Figura 3.6:</b> Algoritmos evolutivos.....	40
<b>Figura 3.7:</b> Principio de trabajo de un AG .....	41
<b>Figura 3.8:</b> Representación binaria de un individuo formado por un cromosoma.....	42
<b>Figura 3.9:</b> Operador cruzamiento single-point .....	45
<b>Figura 3.10:</b> Operador cruzamiento Two-point.....	45
<b>Figura 3.11:</b> Operador reproducción bit-wise .....	46
<b>Figura 4.1:</b> Modelo Matemático.....	48
<b>Figura 4.2:</b> Restricción del problema .....	49
<b>Figura 4.3:</b> Diagrama Flujo Algoritmo Genético (GA) .....	52
<b>Figura 4.4:</b> Representación de las soluciones.....	53
<b>Figura 4.5:</b> Representación Grafica Cromosoma .....	53
<b>Figura 4.6:</b> Penalización soluciones que violan la restricción .....	54
<b>Figura 4.7:</b> Selección Por Torneo .....	55
<b>Figura 4.8:</b> Operador Cruzamiento Vector Secuencia.....	57
<b>Figura 4.9:</b> Operador Cruzamiento Vector Asignación .....	57
<b>Figura 4.10:</b> Descendientes obtenido luego de aplicar el operador cruzamiento .....	58
<b>Figura 4.11:</b> Operador mutación vector secuencia.....	59
<b>Figura 4.12:</b> Operador mutación vector Asignación .....	60
<b>Figura 4.13 :</b> Diagrama Flujo GALS.....	61
<b>Figura 4.14:</b> Búsqueda Local.....	62
<b>Figura 5.1:</b> Factores y niveles del diseño experimental(GA).....	64
<b>Figura 5.2:</b> Grafico Convergencia de las soluciones .....	65
<b>Figura 5.3:</b> Ajuste de parámetros GA .....	66
<b>Figura 5.4:</b> Mejor Combinación de parámetros GA .....	66
<b>Figura 5.5:</b> Factores y niveles del diseño experimento (GALS) .....	67

<b>Figura 5.6:</b> Combinación de parámetros GALS que produce la mayor cantidad de soluciones factibles .....	68
<b>Figura 5.7:</b> Mejor Combinación de parámetros GA .....	68
<b>Figura 5.8:</b> Parámetros para el análisis comparativo .....	68
<b>Figura 5.9:</b> Grafico Resultados de Algoritmo Genético (GA) .....	69
<b>Figura 5.10:</b> Grafico Resultados de Algoritmo Genético hibrido (GALS).....	70
<b>Figura 5.11:</b> Grafico Resultados Comparativos.....	71
<b>Figura 5.12:</b> Grafico Óptimos Globales.....	71
<b>Figura 5.13:</b> Reducción Kilómetros.....	72

## 1 **CAPITULO I: INTRODUCCION**

La industria forestal es la segunda generadora de divisas para el país y la primera basada en un recurso forestal renovable. Genera cerca de 130 mil empleos directos y algo más de 300 mil indirectos en las áreas de silvicultura y cosecha, industria primaria y secundaria y servicios. A lo largo de los últimos 15 años, ha generado aproximadamente el 13% del total anual de los retornos por exportaciones que realizó Chile [1]. La industria forestal es completamente privada, constituida principalmente por dos grandes firmas; Arauco y Mininco (CMPC), las que poseen aproximadamente la mitad de las plantaciones del país y que verticalmente integran plantas de celulosa, aserraderos y papeleras.

La logística en la industria forestal comienza con la tala de árboles (eucaliptos y pinos) los cuales se dejan a “borde de camino”. En esta primera etapa el proceso consiste en voltear los troncos y dimensionarlos para luego apilarlos. Luego, se continúa con el trozado y optimizado de la madera. Que depende de los métodos que utiliza cada empresa forestal. Para el caso de Arauco utiliza un sistema de trozado y optimizado en planta, mientras que CMPC trabaja con sistemas de cortes y trozado en el bosque. Posteriormente son llevados a borde de camino con tractores en terrenos planos y torres en terrenos con pendientes, para pronto ser cargados en camiones. Luego, estos camiones se dirigen a sus respectivos destinos que pueden ser: canchas de acopio, planta de celulosa, aserraderos, otros. El valor de los productos forestales se incrementa con las diferentes actividades de la logística forestal. Los costos medios de producción, desde el bosque a planta, se dividen aproximadamente en: el 30 por ciento corresponde a la cosecha, el 42 por ciento para el transporte, el 14 por ciento para la construcción de caminos, el 4 por ciento para la carga, y el 10 por ciento para otros procesos [2].

Parte de los procesos de la logística (transporte, carguío, caminos, cosecha, etc.) son subcontratados a empresas externas especializadas en dichos procesos. Éstas reciben el nombre de Empresas de Servicios Forestales, conocidas por la sigla EMSEFOR. Dichas empresas prestan diferentes tipos de servicios. Uno de ellos es el transporte de productos forestales.

Las EMSEFOR que se dedican al transporte forestal, perciben sus ingresos por cada kilómetro recorrido con carga desde un origen determinado a un destino cualquiera que puede ser un aserradero, planta de celulosa, otros. El costo del recorrido siguiente a realizar, es decir el retorno por una nueva carga ya sea al mismo origen u otro es asumido íntegramente por la empresa que presta el servicio. Es por esta razón, que una adecuada asignación de trabajos para cada patente, que permita minimizar los kilómetros sin carga, provocará un aumento en los márgenes de utilidad de la empresa que presta el servicio de transporte.

En las faenas forestales se deben trasladar grandes volúmenes de madera desde los diferentes orígenes a un conjunto de clientes geográficamente distribuidos. La complejidad está al decidir que viajes realizará cada camión de modo de cumplir con las demandas diarias, al mínimo costo, respetando las restricciones técnicas y políticas. A nivel país, las empresas mandantes utilizan un Sistema de Asignación de Camiones (Asicam) para la programación del transporté forestal. Esta programación tiene diferentes problemas asociados tales como: sobre carga de algunos camiones, jornada extensas de trabajo y tramos largo de recorrido sin carga. Asicam define una ventana de tiempo a todos sus despachos (Viajes). Esta ventana corresponde al intervalo de tiempo que existe entre la hora de presentación a un determinado origen para cargar los productos demandados por un cliente y la hora de llegada a su correspondiente destino para descargarlos.

El problema a tratar es el típico problema de programación de la producción, el cual también puede ser interpretado, dentro del ámbito de la programación de la producción, como un problema de secuenciamiento de “n” trabajos sobre “m”

máquinas paralelas no relacionadas con tiempos de preparación o setup dependientes. Los problemas de programación de la producción han sido estudiados intensamente en la literatura. Su carácter combinatorio hace difícil su resolución mediante métodos exactos debido a su gran complejidad computacional asociada a la categoría NP hard. La alternativa a dicho enfoque es utilizar metaheurísticas que intenten resolver el problema mediante la aplicación de criterios de búsqueda, obteniéndose buenos resultados en tiempo computacional aceptable.

El presente estudio, efectúa una breve descripción del sector forestal y una revisión en la literatura de las metaheurísticas de optimización. Además, propone dos metaheurísticas basadas en el principio de los algoritmos genéticos para optimizar la actual asignación de camiones para el transporte de productos forestales, considerando como criterio de optimización el número de kilómetros recorridos sin carga.

## 2 CAPITULO II: “SITUACION DEL SECTOR Y TRANSPORTE FORESTAL”.

### 2.1 SECTOR FORESTAL

Durante la última década, el sector forestal chileno ha generado aproximadamente el 13 por ciento del total anual de los retornos por exportaciones que realizó el país, constituyéndose en la segunda actividad económica más importante, después de la minería, y la primera basada en un recurso natural renovable.

#### 2.1.1 Disponibilidad Recursos

La superficie total del país es de 75.662.560,5 hectáreas. Las categorías de uso más importantes corresponden a Áreas Desprovistas de Vegetación, que representan un 32,7%, Praderas y Matorrales con un 27,1% y Bosque con 20,7% del total. Le siguen en importancia las Nieves y Glaciares 6,1%, Humedades 5,9% y terrenos Agrícolas 5,0%. El resto de las categorías de uso tales como Áreas Urbanas e Industriales, Aguas Continentales y Áreas no reconocidas cubren en total un 2,3% de la superficie nacional. [3]



**Figura 2.1: Uso actual del suelo**

Fuente: Corporación Nacional Forestal, CONAF.

La superficie nacional de Bosque es de 15.637.232,5 hectáreas, de las cuales 13.430.602,5 hectáreas corresponden a Bosque Nativo, representando un 85,9% del total de Bosques, mientras que la Plantación Forestal abarca 2.119.004,5 hectáreas y el Bosque Mixto, 87.625,4 hectáreas, correspondientes a un 13,5% y 0,5% respectivamente.

### **2.1.2 Mercado Forestal**

La industria está orientada principalmente a la exportación, con cerca de mil empresas chilenas que envían a los mercados externos productos forestales con diversos grados de elaboración pulpa química en primer lugar de importancia, seguida de molduras, madera aserrada, madera elaborada; tableros y chapas; puertas, ventanas y piezas para la construcción; astillas; papel periódico, maderas en trozos, entre otros a un total de casi 100 mercados en los cinco continentes [4], destacando América del Norte como cliente principal en especial Estados Unidos seguido de Asia, con China y Japón como los mercados más importantes, Europa y América del Sur. Con proyectos de producción a gran escala, el rubro de la celulosa es la principal actividad económica del sector forestal. Chile está posicionado en el mundo como un productor relevante de celulosa de fibra larga, con uno de los más bajos costos de producción, y ocupa el quinto puesto entre las naciones proveedoras, debido especialmente a la celulosa blanqueada de fibra larga, que usa como materia prima el Pino radiata.

Por otro lado, la industria de la madera aserrada y sus derivados es el segundo gran rubro más importante de exportación forestal, con 37 % del total de productos, que se envían a una diversidad de mercados, destacando principalmente Estados Unidos, Japón y México. Durante el período enero-julio de 2008, las exportaciones forestales chilenas alcanzaron un valor de US\$ 3.234,9 millones, lo que representa un incremento de 13,5% respecto del mismo período de 2007 y de 43% respecto de enero-julio de 2006, año peak en las exportaciones forestales a EE.UU. [4]

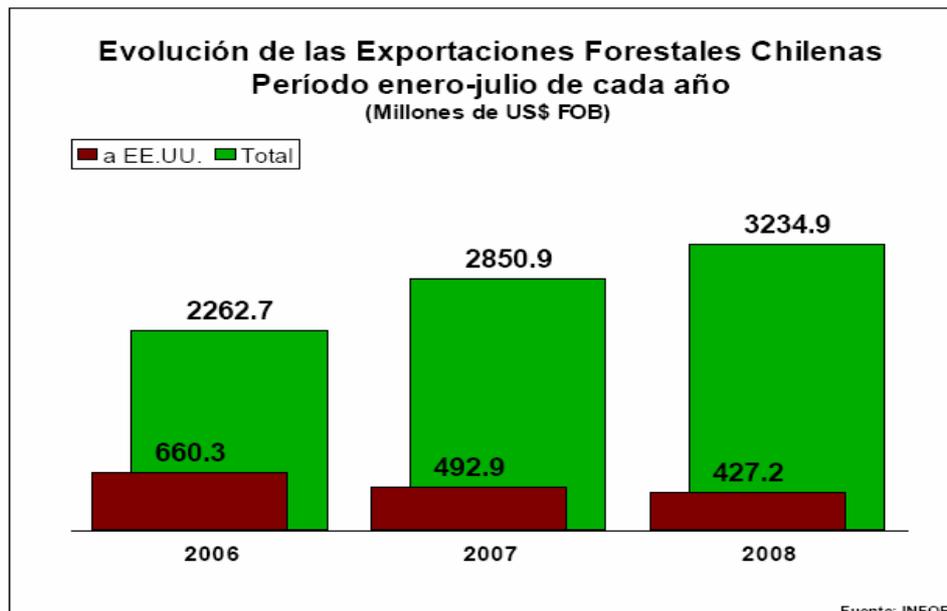


Figura 2.2: Evolución de las Exportaciones chilenas

[4]

El significativo crecimiento alcanzado a pesar de la gran baja experimentada por las exportaciones a EE.UU. (-13,3% en relación a enero-julio de 2007 y de -35,3% respecto de enero-julio de 2006), se fundamenta, principalmente, en el buen comportamiento de las siguientes variables: el precio internacional de la pulpa, el volumen de exportación de la pulpa de eucalipto, el volumen y el precio de exportación de las astillas de eucalipto y de los tableros contrachapados, y el volumen de exportación de los tableros MDF y de los papeles y cartulinas multicapas. Pero también se basa en las mayores oportunidades que se han abierto en casi todos los principales mercados de destino y en los aumentos de volumen exportable derivados de la puesta en marcha de proyectos de inversión liderados por las grandes empresas del sector.

### 2.1.3 Ciclo Forestal

Para conocer con mayor detalle el entorno en el cual se inserta la etapa de transporte en la industria forestal, se presenta a continuación una breve descripción de las diferentes actividades que se llevan a cabo en la elaboración de los diferentes productos forestales, y los aspectos logísticos más importantes (ver figura 2.3). El ciclo forestal se inicia con la formación y producción de plantas en viveros a partir de semillas o de reproducción vegetativa. Las plantas que se obtengan del proceso son trasladadas a un lugar donde se establecerá el bosque mediante la plantación. Una vez establecida la plantación se inicia lo que se denomina Manejo Forestal, que consiste en determinadas intervenciones silviculturales que posibilitan la obtención de madera de alto valor para diferentes usos industriales. Las intervenciones habituales son las podas y los raleos [5].

- **Podas:** Consisten en la eliminación parcial de las ramas inferiores de los árboles, y con su aplicación se asegura la obtención de madera libre de nudos, generando un producto de alto valor comercial.
- **Raleos:** Existen dos tipos de raleos, el primero consiste en la extracción de los árboles defectuosos con el fin de asegurar que los árboles remanentes, que se extraerán en la cosecha, serán los mejores, constituyéndose en árboles sanos, troncos de mayor diámetro y de mejor forma, el segundo es de tipo comercial permite disminuir la densidad de árboles por hectárea.

Antes de intervenir un bosque es necesario diseñar las redes de caminos y la localización de canchas de acopio en las faenas de extracción, considerando aspectos ambientales, técnicos y de seguridad. Luego, y siguiendo con un orden logístico, viene la fase de Cosecha Forestal, que consiste en la tala de los árboles adultos. Posteriormente esta madera es transportada a los distintos centros de consumo (aserraderos, canchas de acopio, plantas de celulosa, etc.) [5], donde es procesada, para finalmente ser llevada a sus clientes en el mercado nacional y extranjero. Es esta última etapa del ciclo forestal, en donde se inserta el problema de asignación de camiones que se presenta más adelante.



**Figura 2.3** Ciclo Forestal [6]

#### 2.1.4 Software aplicados en la Industria Forestal

La complejidad de los procesos forestales, son responsable de la utilización cada vez más frecuente de herramientas computacionales como auxilio en el proceso de toma de decisiones.

A fines de la década de los 80, las principales firmas forestales chilenas, han comenzado a poner en práctica modelos desarrollados conjuntamente con académicos de la Universidad de Chile. Estos sistemas apoyan decisiones sobre la programación diaria de camiones (ASICAM), la optimización del diseño de caminos de accesos a los bosques y la localización de canchas de acopio en labores de cosecha (PLANEX) y la planificación de cosecha a corto, mediano y largo plazo (OPTICORT, OPTIMED y MEDFOR). Todos estos sistemas se han hecho con el tiempo cada vez más sofisticados, pero siempre manteniendo su génesis.

## 2.2 TRANSPORTE FORESTAL

Un problema de importancia en las faenas forestales es el transporte diario de madera desde los orígenes en los predios en explotación a diferentes destinos intermedios o finales: aserraderos, plantas de celulosa, canchas de trozado, etc. En los predios se trabaja con equipos de extracción (ejemplo de ello son las torres), se cargan los trozos en camiones que los transportaran a los diferentes destinos de acuerdo a requerimientos de éstos y al tipo de madera, la cual se define físicamente por largo y diámetro del trozo. El transporte es realizado por camiones que hoy en día están prácticamente estandarizados en términos de capacidad de carga, factibilidad de trasportar los productos y posibilidades de acceso a distintos orígenes.

Los procesos de carga y descarga se efectúan con grúas. Los camiones son Subcontratados y se tienen establecidos convenios tarifarios con ellos. Cada día existe un cierto volumen a transportar desde cada origen, el que se compone de madera de distintos productos correspondientes a la producción del día más los productos en stock (si es que existen) del día anterior. Para los distintos destinos, se define una demanda diaria por cada tipo de madera. Esta demanda puede ser satisfecha desde cualquier origen que produzca ese tipo de madera. Se permite una fluctuación de hasta un 10% en satisfacer la demanda [2]. El problema de programación de viajes de camiones es combinatoriamente complejo y no es trivial encontrar soluciones eficientes. De ahí que una solución manual difícilmente puede alcanzar un resultado que se acerque al óptimo.

Evitar las esperas de los camiones en carga y descarga, buscar viajes con trayectos lo más corto posible, abastecer en forma regular a lo largo del día cada uno de los destinos, respetar restricciones técnicas, horas de alimentación y la jornada de trabajo del personal, etc. Hace imperioso la necesidad de implementar un programa de transporte lo más eficiente posible.

Es así, que nace como respuesta a estas problemáticas el Sistema de Asignación de Camiones (ASICAM), que fue el primer modelo operativo en el mundo aplicado al transporte forestal [7]

### **2.2.1 ASICAM**

ASICAM fue diseñado para optimizar la eficiencia de la flota de camión responsable del transporte de productos forestales. El sistema es considerado como el más avanzado en su categoría y ha sido usado por empresas forestales en Chile, Sudáfrica, Venezuela y Brasil. El objetivo fundamental del software es poder hacer una programación eficiente del transporte de madera en los diferentes centros (Bosques, Aserraderos, Celulosas, canchas de acopio) reduciendo al mínimo los gastos de transporte y respetando las restricciones técnicas, políticas y operacionales de la empresa.

El sistema está fundamentado en dos nociones básicas: un sistema centralizado administrativo que programa y controla todos los viajes (Central de Transporte), y un modelo de simulación manejado por reglas heurísticas que permiten generar tales decisiones. La determinación de un programa de camiones es un problema complejo, existen muchas combinaciones de viajes posibles, que claramente no puede ser solucionado eficientemente en forma manual utilizando un modelo matemático.

Al fin de cada día, se debe entregar a los chóferes y operadores de grúas en el programa del día siguiente, que indica para cada camión su programa de carga en orígenes, viajes y descargas en destinos con horarios. Un camión típicamente hace unos 4 viajes en una jornada de trabajo. En forma consistente al programa de camiones, a los operadores de grúas se les entrega un horario de atención de camiones. El objetivo central del sistema es mover en forma adecuada la madera. [2]

El problema de programación de camiones es complejo, ya que hay muchas combinaciones de viajes posibles y constituidos no solo por viajes en ciclos. De ahí que una solución manual difícilmente puede alcanzar un resultado que se acerque al óptimo. Por este motivo, se planteó un esquema computarizado basado en un modelo de simulación de los viajes y de los procesos de carga y descarga. Dada la complejidad combinatorial del problema, no es posible resolverlo mediante un modelo matemático optimizante.

### **2.2.2 Resultados Asicam**

El ordenamiento administrativo y el sistema heurístico computacional permiten una significativa mejora del aprovechamiento de los vehículos, maquinarias y personal. Un ejemplo de ellos se ve en el tiempo que tomaban los chóferes en hacer sus tareas, tiempo que llegaba a las 14 horas diarias, mientras que después de la implementación de este sistema este tiempo se redujo a 10 horas diarias. Donde aún se exceden en 2 horas el horario normal de trabajo de cada persona. Pero que sin duda mejoró la calidad de vida de cada conductor. [2]

Para el caso de Arauco, dio resultados tan dramáticos, como la disminución de los costos de transporte entre un 15% y 25%. Por ejemplo, si una forestal que se manejaba con 120 camiones, un mes después de aplicar estos conceptos (centralizar las decisiones y usar los sistemas computacionales) había bajado el número a 80.

A continuación, en el presente cuadro se muestra las diferencias al implementar ASICAM.

- 1) No planificar la Asignación de viajes.
- 2) Planificar manualmente el sistema.
- 3) Planificar o Programar utilizando ASICAM

### EJEMPLO REAL DE TRANSPORTE

#### Antecedentes

Número de orígenes:	6	Tipos de Productos:	5
Numero de destinos	8	Distancia Flete:	46,6 Km.

	Sin planificación	Planificación Manual	Simulación Asicam
<b>1. Camiones</b>			
Volumen transportado (m <sup>3</sup> ssc)	2390	2390	2390
Nº de camiones utilizados	46	41	38
Facturación por camión día (\$)	41041	46046	49681
Horas total de trabajo (hr)	11	9,2	9,6
Horas efectivas (hr)	6,8	8	8,9
Tiempo no productivo	4,2	1,2	0,7
<b>2. Equipos de Carguío</b>			
Horas de trabajo	11	9,7	10,5
Horas efectivas	9,5	6,5	6,7
Tiempo no productivo	1,5	3,2	3,8
<b>3. Equipo de descarga</b>			
Horas de trabajo	10	7,3	8,9
Horas efectivas	5,8	5,6	5,8
Tiempo no productivo	4,2	1,7	3,1
Tiempo espera camiones (hr)	0,5	0,18	0,15

**Fuente:** Weintraub A. 1990. Un Sistema de Asignación de Camiones para el Transporte de Productos Forestales. Revista de Ingeniería de Sistemas U. De Chile.Vol. VII. (1).Pág. 80.

### **2.2.3 Desventajas Asicam**

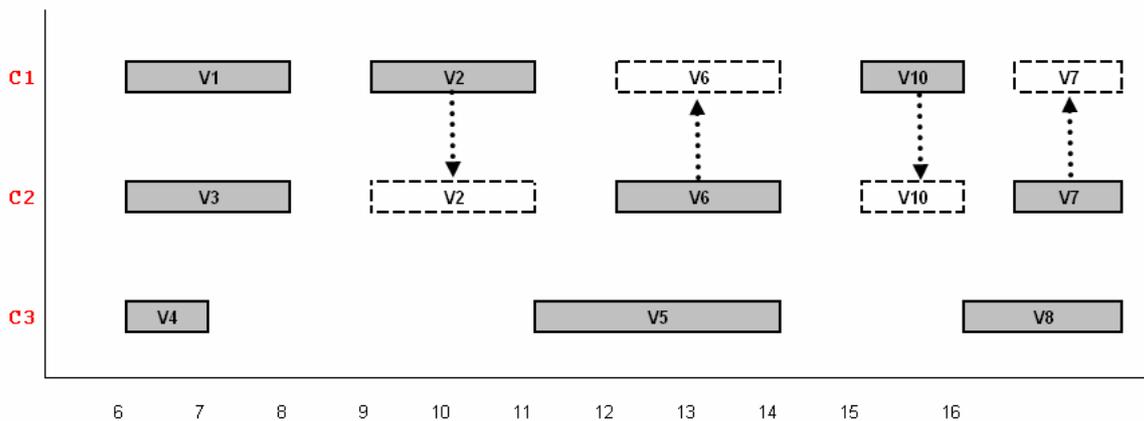
Si bien la programación de camiones entregada por ASICAM es bastante aceptable, no se puede afirmar que sea la óptima (Global). La heurística con la cual trabaja el modelo es de tipo greedy (golosa). Esto significa que al ir asignando camiones elige en cada iteración lo que “le parece mejor” sin mirar más allá (decisiones irreversibles), pudiendo caer en algunos casos en óptimos locales.

## **2.3 DESCRIPCIÓN DEL PROBLEMA**

Diariamente en las faenas forestales se deben transportar diferentes productos desde los distintos orígenes en los predios hasta los destinos determinados, que pueden ser; planta de celulosa, aserraderos, canchas de trozado, canchas de acopio, otros. Para el traslado maderero se utilizan camiones, los cuales son subcontratados.

Independiente de la programación que se utilice las EMSEFOR tienen diferentes problemas asociados tales como: sobre carga de algunos camiones, jornadas de trabajo muy extensas y tramos largos de recorrido sin carga.

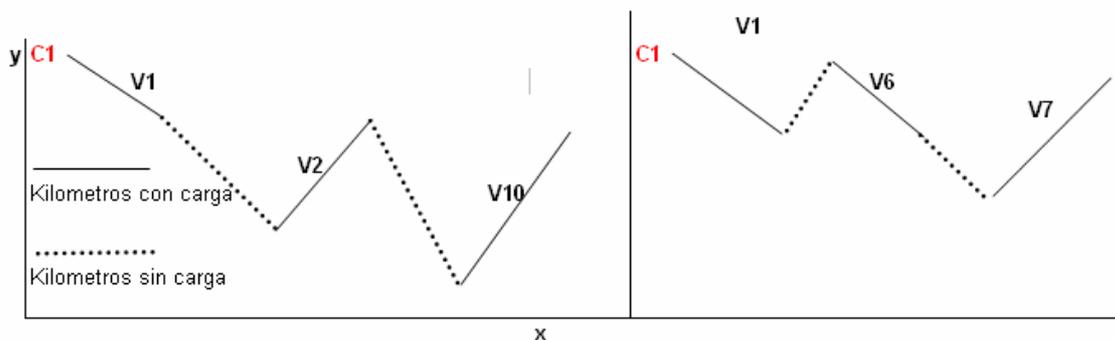
En la actualidad a nivel país, las empresas mandantes utilizan Asicam para la programación del transporté forestal. Asicam define una ventana de tiempo a todos sus despachos (Viajes) esta ventana corresponde al intervalo de tiempo que existe entre la hora de presentación a un determinado origen para cargar los productos demandados por un cliente y la hora de llegada a su correspondiente destino para descargarlos. La idea, es reasignar los viajes programados por Asicam a los distintos camiones con el objetivo de minimizar la cantidad total de kilómetros sin carga. La figura N° 2.4 (a) esquematiza las ventanas de tiempo definidas por Asicam (Viajes) para cada camión.



**Figura 2.4 (a):** Ilustración problema a resolver

Fuente: Elaboración Propia

El camión1 (C1) con la programación entregada por Asicam realiza los viajes 1,2 y 10. Con la reasignación propuesta efectuara los viajes 1,6 y 7. La idea de la reprogramación se ilustra en la figura N° 2.4 (b), donde el camión1 disminuye el recorrido de los kilómetros sin carga al modificar los viajes a realizar. Cabe mencionar, que con la nueva programación se producen externalidades negativas o positivas, las cuales dan origen a nuevos problemas



a) Antes Reasignación

(b) Después Reasignación

**Figura 2.5 (b):** Ilustración problema a resolver

La única restricción que se debe cumplir para reasignar dos viajes entre las patentes, es que el tiempo disponible entre los viajes debe ser mayor al tiempo de traslado en ir desde el destino del primer viaje al origen del segundo. Por ejemplo, vamos a analizar si al camión 1 (C<sub>1</sub>) se le pueda reasignar el viaje 6 (V6) después del viaje 1. El tiempo disponible entre los viajes es de 3 horas. Por otro

lado, el tiempo de traslado desde el destino de V1 al origen del viaje V6 es de 2 horas. Por lo tanto, V6 se le puede asignar a C<sub>1</sub> después de efectuar el viaje1.

**Tiempo disponible entre viajes (3 horas) > tiempo traslado entre los viajes (2 horas)**

## **2.4 OBJETIVOS DEL ESTUDIO**

### **2.4.1 Objetivo General**

Reasignar la programación de camiones para el transporte de productos forestales minimizando la cantidad total de kilómetros recorridos sin carga.

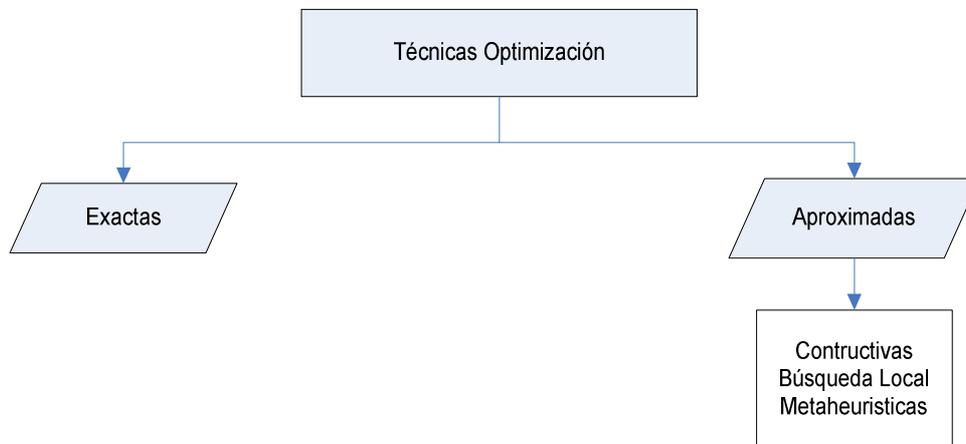
### **2.4.2 Objetivo específico**

- Estudiar las distintas metaheurísticas existentes en la literatura
- Analizar el principio de trabajo de los Algoritmos Genéticos
- Comparar la eficiencia de los algoritmos Propuestos
- Modelar los algoritmos en lenguaje computacional (C y C++)
- Proponer nuevas líneas de investigación

### 3 CAPITULO III: MARCO TEORICO

#### 3.1 INTRODUCCIÓN A LA OPTIMIZACIÓN

Existen áreas del conocimiento y del quehacer humano donde surgen problemas relacionados con la mejora de determinados procesos o la obtención de mejores soluciones (generales o particulares). La disciplina que estudia este tipo de problemas y sus respectivas alternativas es conocida como optimización. Los problemas de optimización pueden clasificarse en función de diferentes factores como son su complejidad, la existencia o no de restricciones, su carácter estático o dinámico, lineal o no lineal, mono-objetivo o multi-objetivo, etc. En cuanto a las técnicas de búsqueda, estas se pueden clasificar en función de si aseguran obtener el resultado óptimo (técnicas exactas o clásicos) o si por el contrario permiten obtener soluciones cercanas al óptimo (técnicas aproximadas o heurísticas) [8]



**Figura 3.1:** Clasificación Técnicas Optimización

**Fuente:** Elaboración Propia

Métodos optimización exactos o clásicos pueden ser clasificados en dos grupos distintos: métodos directos y métodos basados en gradiente. Los métodos de búsqueda directos únicamente la función objetivo  $F(X)$ , y los valores de las restricciones  $(g_j(x), h_k(x))$  son usados para guiar la estrategia de búsqueda,

considerando el método basado en el gradiente usa las derivada de primer y segundo orden de la función objetivo y o las restricciones para guiar el proceso de búsqueda desde la información de las derivadas no es usada, el método de búsqueda directa es usualmente lento, requiriendo muchas evaluaciones de funciones para la convergencia. Por la misma razón ellos pueden ser aplicados a muchos problemas sin un cambio mayor en el algoritmo. Por otro lado el método basado en el gradiente converge rápidamente a una cercana solución óptima, pero no es eficiente en problemas no diferenciables o discontinuos. En suma, hay algunas dificultades comunes con muchas técnicas clásicas directas y basadas en el gradiente.

Las técnicas exactas garantizan encontrar la solución óptima para cualquier instancia de cualquier problema en un tiempo acotado. El inconveniente de estos métodos es que el tiempo necesario para llevarlos a cabo, aunque acotado, crece exponencialmente con el tamaño del problema, ya que la mayoría de éstos son NP-HARD. Esto provoca en muchos casos que el tiempo necesario para la resolución del problema sea inabordable (cientos de años). Por lo tanto, los algoritmos aproximados para resolver estos problemas están recibiendo una atención cada vez mayor por parte de la comunidad internacional a lo largo de los últimos treinta años. Estos métodos sacrifican la garantía de encontrar el óptimo a cambio de encontrar una buena solución en un tiempo razonable. Dentro de los algoritmos aproximados se pueden encontrar tres tipos: los heurísticos constructivos (también llamados voraces), los métodos de búsqueda local (o métodos de seguimiento del gradiente) y las metaheurísticas.

### **3.2 METAHEURÍSTICAS DE OPTIMIZACIÓN**

En las últimas décadas se ha desarrollado una nueva clase de algoritmos de aproximación que básicamente tratan de combinar métodos heurísticos básicos en un marco de alto nivel orientado a buscar la eficiencia en el proceso de búsqueda. Durante los años 80 y principios de los 90 estas técnicas se conocieron

como heurísticas modernas. Dichas técnicas consisten en procedimientos sistemáticos de prueba que ofrecen soluciones aceptables, no necesariamente óptimos globales, para problemas donde el espacio de soluciones es indeterminado o lo suficientemente amplio como para que no pueda ser resuelto en un tiempo aceptable. En muchos casos, las técnicas heurísticas se diseñan en función de las características particulares del problema a resolver. En la actualidad existe un gran número de técnicas heurísticas de optimización.

Además del concepto de heurística, es habitual hablar de metaheurística. Una metaheurística se puede describir como un proceso iterativo maestro que guía y modifica las operaciones de heurísticas subordinadas para producir de forma eficiente soluciones de alta calidad. Las heurísticas subordinadas pueden ser procedimientos de alto o bajo nivel, un método de búsqueda local o simplemente un método constructivo. Aunque se suele considerar como metaheurística a toda aquella generalización de una determinada heurística para cualquier tipo de problema, en la práctica es necesario analizar detalladamente el problema a resolver para determinar cual de ellas es la que se presupone puede tener un mayor éxito en la búsqueda de soluciones, aunque no se pueda determinar con total seguridad si se alcanzara dicho éxito.

### **3.2.1 Concepto de Metaheurística**

El termino metaheurísticas se obtiene de anteponer a *heurística* el sufijo *meta* que significa mas allá o a un nivel superior. Las concepciones actuales de lo que es una metaheurística están basados en las diferentes interpretaciones de lo que es una forma inteligente de resolver un problema. Las *metaheurísticas* son estrategias inteligentes para diseñar o mejorar procedimientos heurísticos muy generales con un alto rendimiento. El termino metaheurística apareció por primera vez en el artículo seminal sobre búsqueda *tabú* de Fred Glover en 1986. A partir de entonces han surgido multitud de propuestas de pautas para diseñar buenos procedimientos para resolver ciertos problemas que, al ampliar su campo de aplicación, han adoptado la denominación de metaheurísticas [10]

De las diferentes descripciones de metaheurísticas que se encuentran en la literatura se pueden destacar ciertas propiedades fundamentales que caracterizan a este tipo de métodos: [11]

- Las metaheurísticas son estrategias o plantillas generales que guían el proceso de búsqueda
- El objetivo es una exploración del espacio de búsqueda eficiente para encontrar soluciones (casi) óptimas.
- Las metaheurísticas son algoritmos no exactos y generalmente son no deterministas.
- Pueden incorporar mecanismos para evitar las áreas del espacio de búsqueda no óptimas.
- El esquema básico de cualquier metaheurística es general y no depende del problema a resolver.

Las metaheurísticas hacen uso de conocimiento del problema que se trata resolver en forma de heurísticos específicos que son controlados de manera estructurada por una estrategia de más alto nivel. Las metaheurísticas utilizan funciones de bondad (funciones de fitness) para cuantificar el grado de adecuación de una determinada solución.

Resumiendo esos puntos, se puede acordar que una metaheurística es una estrategia de alto nivel que usa diferentes métodos para explorar el espacio de búsqueda. En otras palabras, una metaheurística es una plantilla general no determinista que debe ser rellenada con datos específicos del problema (representación de las soluciones, operadores para manipularlas, etc.) y que permite abordar problemas con espacios de búsqueda de gran tamaño (por ejemplo,  $2^{1000}$  posibles soluciones). Por lo tanto es de especial interés el correcto equilibrio (generalmente dinámico) que haya entre diversificación e intensificación. [11]

El termino diversificación se refiere a la exploración del espacio de búsqueda, mientras que intensificación se refiere a la explotación de algún área concreta de ese espacio. El equilibrio entre estos dos aspectos contrapuestos es de gran importancia, ya que por un lado deben identificarse rápidamente la regiones prometedoras del espacio de búsqueda global y por el otro lado no se debe malgastar tiempo en las regiones que ya han sido exploradas o que no contienen soluciones de alta calidad.

Entre las principales técnicas metaheurísticas de optimización podemos encontrar el enfriamiento simulado (en ingles Simulated Annealing, SA), la búsqueda tabú (en ingles Tabu Search, TS) , los algoritmos genéticos (en ingles Genetic Algorithms, GA) , la búsqueda dispersa (en ingles Scatter Search, SS) , los algoritmos memeticos (en ingles Memetic Algorithms, MA), la búsqueda en vecindario variable (en ingles Variable Neighborhood Search, VNS), la búsqueda local guiada (en ingles Guided Local Search, GLS), los procedimientos de búsqueda basados en procedimientos adaptativos aleatorizados avaros (en ingles Greedy Randomized Adaptive Search Procedures, GRASP) , la optimización mediante colonias de hormigas (en ingles Ant Colony Optimization, ACO) , la búsqueda local iterada (en ingles Iterated Local Search, ILS), las redes neuronales (en ingles Neural Networks, NN), y otras. Además, se pueden incluir las técnicas híbridas que combinan aspectos de diferentes metaheurísticas. [10]

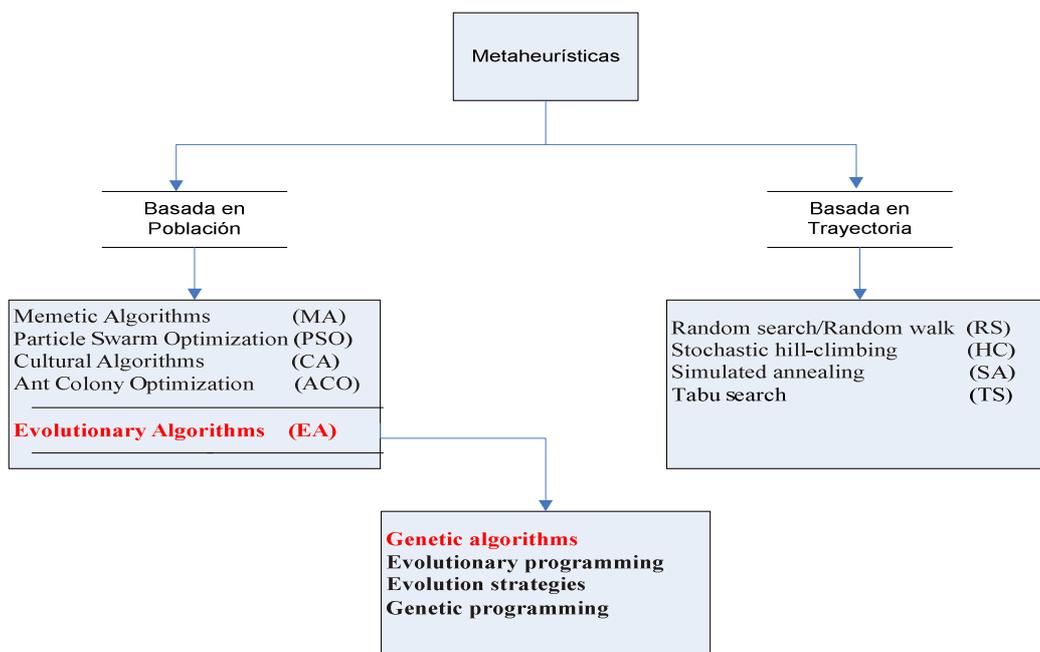
### 3.2.2 Clasificación Técnicas Metaheurísticas

En la literatura existen diferentes formas de clasificar las metaheurísticas en función de sus características propias. Entre las principales clasificaciones encontramos: [8]

- **Técnicas inspiradas vs. no inspiradas en la naturaleza.** Mucha metaheurísticas, como por ejemplo los algoritmos genéticos, enfriamiento simulado, optimización mediante colonias de hormigas, etc. basan su funcionamiento en aspectos inspirados de la naturaleza Otras, como por ejemplo búsqueda tabú o la búsqueda local guiada, no simulan ningún aspecto de la naturaleza.

- **Técnicas basadas en trayectorias vs. Poblacionales.** Un aspecto diferenciador entre metaheurísticas radica en el número de soluciones que se utilizan en el proceso de optimización. Metaheurísticas como enfriamiento simulado utilizan una sola solución durante el proceso de búsqueda, por lo que se suelen denominar métodos de trayectoria ya que la solución describe una trayectoria desde la solución de partida hasta encontrar la solución final. Por otro lado, técnicas como los algoritmos genéticos hacen uso de un conjunto de soluciones (población) que son optimizadas de forma simultánea durante la búsqueda.
- **Técnicas estáticas vs. Dinámicas.** Otro aspecto diferenciador entre metaheurísticas radica en la función de aptitud utilizada. Mientras que en la mayoría de heurísticas utilizan la misma función objetivo durante todo el proceso de búsqueda otras, como la búsqueda local guiada, modifican dicha función en tiempo de ejecución, lo que ayuda a escapar de mínimos locales.
- **Técnicas basadas en estructuras de vecindario único vs. Vecindarios múltiples.** La mayoría de los algoritmos trabajan con una estructura de vecindario simple, es decir, la topología del espacio objetivo no cambia durante la búsqueda. Sin embargo, existen metaheurísticas como la búsqueda en vecindario variable que diversifican la búsqueda mediante el uso de diferentes espacios de soluciones.
- **Meta heurísticas con memoria vs. Sin memoria.** Uno de los criterios más utilizados para clasificar la metaheurísticas es el uso que hacen de su historia de búsqueda, es decir, si utilizan memoria o no. Heurísticas como el enfriamiento simulado utilizan exclusivamente el estado actual del proceso de búsqueda a la hora de determinar la próxima forma de actuación. Otras técnicas como la búsqueda tabú utilizan información previa (memoria) del proceso de búsqueda a la hora de tomar nuevas decisiones.

En presente trabajo, se clasificaran de acuerdo al número de soluciones que utilizan en el proceso de búsqueda. Se utilizara este criterio debido a que es una de la más utilizada en la literatura. A continuación en la figura 3.2 se presenta la clasificación de las metaheurísticas mas empleadas.



**Figura 3. 2:** Clasificación de las meta heurísticas

**Fuente:** Elaboración Propia

Dentro del grupo de las metaheurísticas basadas en población se encuentran la familia de los algoritmos evolutivos. Al cual pertenecen los algoritmos genéticos, programación evolutiva, estrategias evolucionistas y la programación genética. En este estudio se propondrán dos algoritmos basados en algoritmos genéticos

### 3.2.3 Técnicas Metaheurísticas basadas en trayectoria

La principal característica de estos métodos es que parten de un punto y mediante la exploración del vecindario van actualizando la solución actual, formando una trayectoria. La mayoría de estos algoritmos surgen como extensiones de los métodos de búsqueda local simples a los que se les añade alguna característica para escapar de los mínimos locales. Esto implica la necesidad de una condición de parada más compleja que la de encontrar un mínimo local. Normalmente se

termina la búsqueda cuando se alcanza un número máximo predefinido de iteraciones, se encuentra una solución con una calidad aceptable, o se detecta un estancamiento del proceso.

### **Ascenso de Colinas (HC)**

El método básico de búsqueda local es el conocido como ascenso de colinas (en inglés Hill Climbling, HC). Dicho nombre, que para problemas de minimización se denomina descenso de colinas, se debe a que cada para admitir una determinada solución, esta debe conllevar una mejora respecto a la anterior, es decir, no se permiten movimientos que empeoren la calidad de la solución. Por esta razón, el algoritmo de ascenso de colinas se detiene cuando se encuentra un óptimo local, no siendo efectivo en problemas combinatorios complejos. [12]

### **Simulated Annealing (SA)**

El Enfriamiento Simulado o Simulated Annealing (SA) es una de las más antiguas entre las metaheurísticas y seguramente es el primer algoritmo con una estrategia explícita para escapar de los óptimos locales. El SA fue inicialmente presentado en [13]. La idea del SA es simular el proceso de recocido del metal y del cristal. Para evitar quedar atrapado en un óptimo local, el algoritmo permite elegir una solución peor que la solución actual. En cada iteración se elige, a partir de la solución actual  $s$ , una solución  $s_0$  del vecindario  $N(s)$ . Si  $s_0$  es mejor que  $s$  (es decir, tiene un mejor valor en la función de fitness), se sustituye  $s$  por  $s_0$  como solución actual. Si la solución  $s_0$  es peor, entonces es aceptada con una determinada probabilidad que depende de la temperatura actual  $T$  y de la variación en la función de fitness,  $f(s_0) - f(s)$  (caso de minimización). Esta probabilidad generalmente se calcula siguiendo la distribución de Boltzmann:

$$p(s'|T, s) = e^{-\frac{f(s') - f(s)}{T}}.$$

### **Tabú Search (TS)**

La Búsqueda Tabú o Tabu Search (TS) es una de las metaheurísticas que se han aplicado con más éxito a la hora de resolver problemas de optimización combinatoria. Los fundamentos de este método fueron introducidos en [14]. La idea básica de la búsqueda tabú es el uso explícito de un historial de la búsqueda (una memoria de corto plazo), tanto para escapar de los óptimos locales como para implementar su estrategia de exploración y evitar buscar varias veces en la misma región. Esta memoria de corto plazo es implementada en esta técnica como una lista tabú, donde se mantienen las soluciones visitadas más recientemente para excluirlas de los próximos movimientos. En cada iteración se elige la mejor solución entre las permitidas y la solución es añadida a la lista tabú.

### **3.2.4 Técnicas Metaheurísticas basadas en población**

Los métodos basados en población se caracterizan por trabajar con un conjunto de soluciones (población) en cada iteración, a diferencia de las técnicas basadas en trayectoria antes que únicamente utilizan un punto del espacio de búsqueda por iteración. El resultado final proporcionado por este tipo de algoritmos depende fuertemente de la forma en que manipula la población.

### **Scatter Search (SS)**

La Búsqueda Dispersa es una metaheurística cuyos principios fueron presentados en [15] y que actualmente está recibiendo una gran atención por parte de la comunidad científica. El algoritmo se basa en mantener un conjunto relativamente pequeño de soluciones tentativas (llamado conjunto de referencia) que se caracteriza tanto por contener buenas soluciones como soluciones diversas. Este conjunto se divide en subconjuntos de soluciones a las cuales se les aplica una operación de recombinación y mejora. Para realizar la mejora o refinamiento de soluciones se suelen utilizar mecanismos de búsqueda local.

### **Ant Colony Optimization**

Los sistemas basados en Colonias de Hormigas o Ant Colony Optimization (ACO) [16] son unas metaheurísticas inspiradas en el comportamiento en la de las hormigas reales cuando realizan la búsqueda de comida. Este comportamiento es el siguiente: inicialmente, las hormigas exploran el área cercana a su nido de forma aleatoria. Tan pronto como una hormiga encuentra la comida, la lleva al nido. Mientras que realiza este camino, la hormiga va depositando una sustancia química denominada feromona. Esta sustancia ayudará al resto de las hormigas a encontrar la comida. Esta comunicación indirecta entre las hormigas mediante el rastro de feromona las capacita para encontrar el camino más corto entre el nido y la comida. Esta funcionalidad es la que intenta simular este método para resolver problemas de optimización. En esta técnica, el rastro de feromona es simulado mediante un modelo probabilístico.

### **Algoritmos Meméticos (MA)**

Dentro de las técnicas heurísticas mas recientes encontramos los algoritmos meméticos (Memetic Algorithms) [17]. Los MAs presentan aspectos de especial similitud con los algoritmos evolutivos, sobre todo en lo referente a las mejoras individuales de las soluciones junto con procesos cooperativos y competitivos de tipo poblacional. Los componentes de las poblaciones en MAs no se denominan individuos, sino agentes. En cada generación se actualiza la población de agentes, usando para tal una población temporal obtenida mediante la recombinación de algunos agentes seleccionados. La selección se encarga de elegir una muestra de los mejores agentes contenidos en la población actual haciendo uso de una determinada función objetivo. Por otro lado, el proceso de reemplazo incide en el aspecto competitivo, limitando el tamaño de la población y permitiendo de esta forma que otros nuevos agentes puedan entrar y diversificar la búsqueda.

### **3.3 METAHEURÍSTICAS USADAS**

En este trabajo se aplicaran dos algoritmos pertenecientes a la familia de los algoritmos evolutivos. Dentro de los algoritmos evolutivos se detallaran especialmente las particularidades de los algoritmos genéticos, por ser la metaheurística a implementar en código de programación C++ en esta tesis.

#### **3.3.1 Algoritmos Evolutivos**

Algoritmos evolucionistas (EAs) imitan los principios evolucionistas naturales para constituir procedimientos búsqueda y optimización. EAs son diferentes a los procedimientos clásicos de búsqueda y optimización en unas variedades de formas.

Alrededor de los años 60, algunos investigadores visionarios coincidieron (de forma independiente) en la idea de implementar algoritmos basados en el modelo de evolución orgánica como un intento de resolver tareas complejas de optimización en computadores. Hoy en día, debido a su robustez, a su amplia aplicabilidad, y también a la disponibilidad de una cada vez mayor potencia computacional, e incluso programas paralelos, el campo de investigación resultante, el de la computación evolutiva, recibe una atención creciente por parte de los investigadores de un gran número de disciplinas.[18]

El marco de la computación evolutiva establece una aproximación para resolver el problema de buscar valores óptimos mediante el uso de modelos computacionales basados en procesos evolutivos (algoritmos evolutivos). Los EA son técnicas de optimización que trabajan sobre poblaciones de soluciones y que están diseñadas para buscar valores óptimos en espacios complejos. Están basados en procesos biológicos que se pueden apreciar en la naturaleza, como la selección natural o la herencia genética.

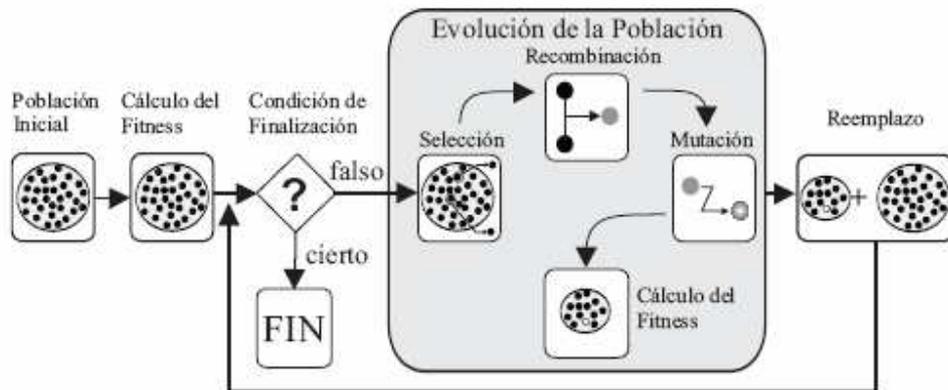
Parte de la evolución esta determinada por la selección natural de individuos diferentes compitiendo por recursos en su entorno. Algunos individuos son mejores que otros y es deseable que aquellos individuos que son mejores sobrevivan y propaguen su material genético

La reproducción sexual permite el intercambio del material genético de los cromosomas, produciendo así descendientes que contienen una combinación de la información genética de sus padres. Este es el operador de recombinación utilizado en los EA, también llamado operador de cruce. La recombinación ocurre en un entorno en el que la selección de los individuos que tienen que emparejarse depende, principalmente, del valor de la función del fitness del individuo, es decir, de que tan bueno es el individuo comparado con los de su entorno.

Como en el caso biológico, los individuos pueden sufrir mutaciones ocasionalmente (operador de mutación). La mutación es una fuente importante de diversidad para los EA. En un EA, se introduce normalmente una gran cantidad de diversidad al comienzo del algoritmo mediante la generación de una población de individuos aleatorios. La importancia de la mutación, que introduce aun mas diversidad mientras el algoritmo se ejecuta, es objeto de debate. Algunos se refieren a la mutación como un operador de segundo plano, que simplemente reemplaza parte de la diversidad original que se haya podido perder a lo largo de la evolución, mientras que otros ven la mutación como el operador que juega el papel principal en el proceso evolutivo.

En la Figura 3.3 se muestra el esquema de funcionamiento de un EA típico. Como puede verse, un EA procede de forma iterativa mediante la evolución de los individuos pertenecientes a la población actual. Esta evolución es normalmente consecuencia de la aplicación de operadores estocásticos de variación sobre la población, como la selección, recombinación y mutación, con el fin de calcular una generación completa de nuevos individuos. El criterio de terminación consiste normalmente en alcanzar un numero máximo de iteraciones (programado

previamente) del algoritmo, o encontrar la solución óptima al problema (o una aproximación a la misma) en caso de que se conozca de antemano.



**Figura 3.3:** Funcionamiento de un EA [18]

En la figura 3.4 se presenta el pseudo-código de los algoritmos evolutivos, el cual se explicara a continuación.

La población inicial esta compuesta usualmente por individuos creados aleatoriamente, aunque también existe cierta tendencia en el uso de técnicas de optimización para crear los individuos que formaran la población inicial, permitiendo así que el EA comience su ejecución v sobre un conjunto de soluciones más prometedoras que las generadas aleatoriamente. Tras la generación de la población inicial, se calcula el valor de adecuación (fitness) de cada uno de los individuos que la forman y el algoritmo entra en el bucle reproductor. Esta etapa consiste en la generación de una nueva población mediante la selección de padres, la recombinación de estos y la mutación de los descendientes obtenidos.

Tras este proceso, los individuos son evaluados. Esta nueva población generada por los operadores de reproducción ( $P_0$ ) se utilizara, junto con la población actual ( $P$ ), para obtener la nueva población de individuos de la siguiente generación. Al final, el algoritmo devolverá la mejor solución encontrada durante la ejecución.

Como puede verse, el algoritmo comprende las tres fases principales: selección, reproducción y reemplazo.

---

**Algoritmo 1** Pseudocódigo de un algoritmo evolutivo

---

```

1:  $P = \text{generarPoblaciónInicial}()$ ;
2:  $\text{evaluar}(P)$ ;
3: while not condiciónParada() do
4:    $P' = \text{seleccionarPadres}(P)$ ;
5:    $P' = \text{aplicarOperadoresDeVariación}(P')$ ;
6:    $\text{evaluar}(P')$ ;
7:    $P = \text{seleccionarNuevaPoblación}(P, P')$ ;
8: end while
9: return la mejor solución encontrada

```

---

**Figura 3.4:** Pseudo-Código Algoritmo Evolutivo

[18]

**Selección:** partiendo de la población inicial  $P$  de individuos, se crea una nueva población temporal ( $P_0$ ) de individuos. Generalmente los individuos más aptos (aquellos correspondientes a las mejores soluciones contenidas en la población) tienen un mayor número de posibilidades que aquellos que tienen menos aptitud (selección natural).

**Reproducción:** en esta fase se aplican los operadores reproductivos a los individuos de la población  $P_0$  para producir una nueva población. Típicamente, esos operadores se corresponden con la recombinación de parejas y con la mutación de los nuevos individuos generados. Estos operadores de variación son, en general, no deterministas, es decir, no siempre se tienen que aplicar a todos los individuos y en todas las generaciones del algoritmo, sino que su comportamiento viene determinado por su probabilidad asociada

**Reemplazo:** finalmente, los individuos de la población original son sustituidos por los individuos recién creados. Este reemplazo usualmente intenta mantener los mejores individuos eliminando los peores.

Los **EAs** son útiles para la solución de problemas NP <sup>1</sup>, ya que trabajan rápidamente en grandes espacios de búsqueda gracias a que son capaces de operar sobre varias soluciones simultáneamente. Para resolver un problema NP las únicas técnicas seguras consisten en ir evaluando exhaustivamente todas las posibilidades hasta encontrar una solución. [19]

Las principales características de los EAs se pueden resumir en que:

- Buscan a partir de una población de distintas soluciones.
- Utilizan una función de adaptación para realizar la búsqueda.
- Utilizan reglas de transición probabilísticas, no reglas deterministas.
- No necesitan grandes requisitos matemáticos para la solución de problemas de interés.
- Se trata de EAs que pueden manejar cualquier tipo de funciones objetivo, y cualquier tipo de restricciones definidas en espacios de búsqueda discretos, continuos o mixtos.
- Tienen mecanismos para disminuir la posibilidad de caer en máximos locales.
- Las soluciones encontradas dependen de unos valores aleatorios que se generan al comenzar.
- Introducen un cierto grado de aleatoriedad en la búsqueda de la solución.
- No siguen ningún camino de búsqueda predeterminado por el espacio de soluciones.

Los principales conceptos que utilizaremos al hablar de EAs se explican a continuación:

- **Cromosoma**: Se trata de una cadena finita de valores definidos en un rango determinado cada uno. Contiene información codificada de una posible solución.
- **Gen**: Es cada uno de los elementos que forman la cadena del cromosoma.

---

<sup>1</sup> Un problema NP es aquel en el que no es posible asegurar que se pueda encontrar una solución en tiempo Polinomial

- Alelo: Es cada uno de los valores (uno o mas) del gen.
- Individuo o genotipo: Cada individuo (ver Figura 3.4) contendría una posible solución al problema. Esta formado por uno o más cromosomas.
- Valor de adecuación o fitness: Valor numérico que se asigna a cada individuo para poder utilizar un criterio capaz de evaluar la proximidad del mismo a una solución óptima.
- Dominancia: Función definida para devolver un único cromosoma a partir de todos los cromosomas que contiene el individuo. Es necesaria en poblaciones en las que los individuos están formados por dos o más cromosomas, cuya aplicación es previa para poder obtener así el valor de adecuación de los individuos.
- Población o fenotipo: Se trata de una expresión (decodificación) del genotipo en el dominio del problema.
- Mutación: Consiste en el cambio del valor de un alelo por otro valor aleatoriamente elegido.
- Vecindario: Conjunto de individuos vecinos a uno dado, es decir, que están situados próximos a el en la población según una topología espacial dada.
- Reproducción: La reproducción consiste en elegir varios individuos para generar descendientes a partir de ellos.
- Convergencia: Este término fue introducido por De Jong en su tesis doctoral. La población converge si el 95% de los individuos que la forman comparten el mismo valor para cada uno de sus genes.

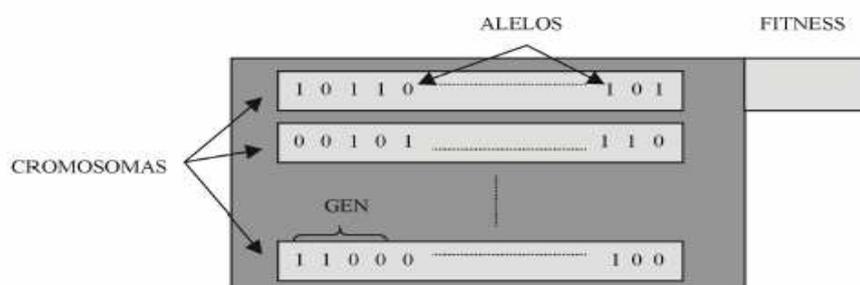
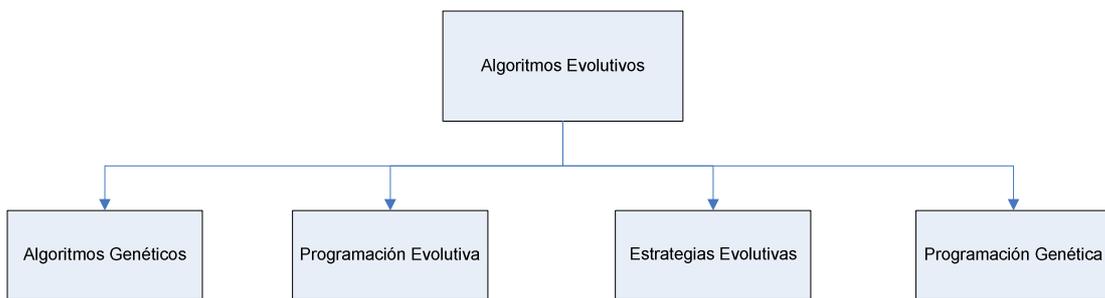


Figura 3.5: Estructura Genética de un individuo en EAs [19]

### 3.3.2 Tipos de Algoritmos Evolutivos

De entre las familias mas importantes que han surgido del estudio de los EAs podemos destacar: Algoritmos Genéticos código binario y parámetros reales, estrategias evolución, programación evolucionista y programación genética. Estos algoritmos se diferencian principalmente en la estructura de los individuos y en el tipo de tratamiento al que se le somete, que ejerce una influencia directa sobre el funcionamiento de los operadores de reproducción y mutación.



**Figura 3.6:** Algoritmos evolutivos

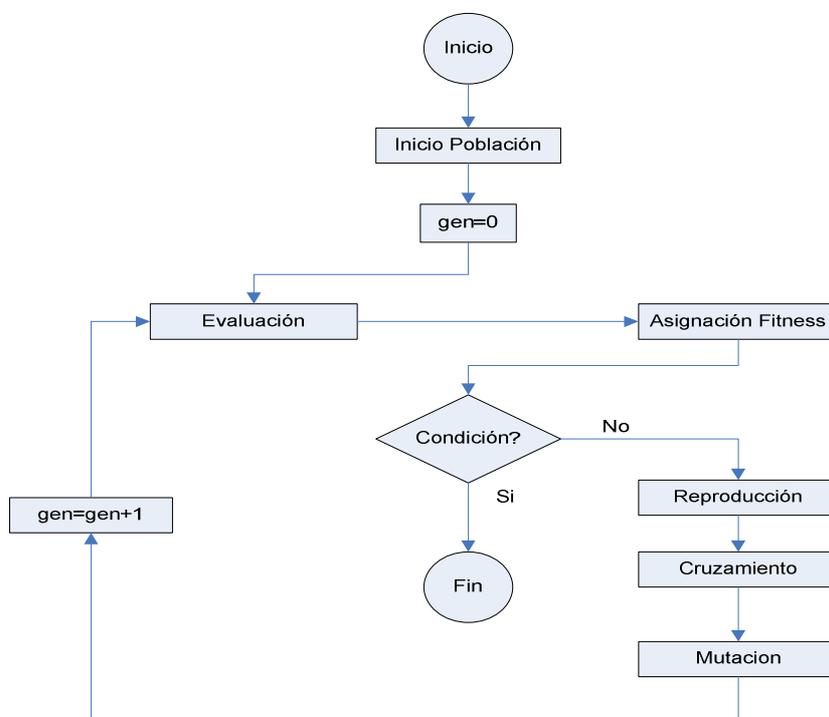
Sin embargo, existen otros algoritmos evolucionistas y inspirados naturaleza, como es optimización ant-colony, evolución simulada, Computación DNA, y algoritmos culturales, descripción de ellas pueden ser encontradas en la literatura.

### 3.3.3 Algoritmos Genéticos

Sobre la última década, algoritmos genéticos (AGs) han sido extensamente usados como herramientas de búsqueda y optimización en varios dominios, incluyendo las ciencias, el comercio e ingeniería. La razón primaria para su éxito es su amplia aplicabilidad, debido a su versatilidad y perspectiva global. El concepto de un algoritmo genético fue primero ideado por John Holland de la universidad de Michigan, Ann Arbor. Después, el y sus estudiantes han contribuido mucho al el desarrollo de este campo. Muchos de los trabajos iniciales de investigación pueden ser encontrados en varias conferencias internacionales. [20]

Los AG(s) son los más conocidos entre los EA con diferencia, y esto ha provocado la aparición en la literatura de gran cantidad de variantes, de forma que es muy difícil dar una regla clara para caracterizarlos. A pesar de esto, la mayoría de los GA se caracterizan por poseer una representación lineal de las soluciones, no suelen incluir parámetros de auto adaptación en los individuos y dotan de mayor importancia al operador de cruzamiento que al de mutación. AG(s) son procedimientos de búsqueda y optimización que están motivados por los principios de genética natural y selección natural. Algunas ideas fundamentales de genética están prestadas y usadas artificialmente para construir algoritmos búsqueda que son robustos y requieren información mínima problema.

El principio trabajo de un AG(s) es muy diferente de las muchas técnicas optimización clásica. En la figura 3.7 se muestra el principio de trabajo de un algoritmo genético, a continuación describiremos en forma detallada un algoritmo genético básico en código binario



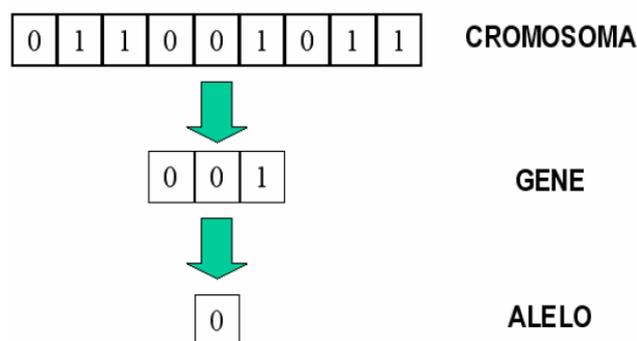
**Figura 3.7:** Principio de trabajo de un AG

Fuente: Elaboración propia

La primera decisión antes de implementar un algoritmo genético, es elegir una representación de las soluciones, que permita recorrer todo el espacio de búsqueda. La codificación variables decisión en cadenas binarias es primordialmente usado para lograr una representación pseudo-chromosomal de la solución.

### Representación de las soluciones

Los cromosomas naturales están hechos de muchos genes, cada uno de los cuales puede tener diferentes valores alelos (tal como, el gen representación para el color de ojos en el cromosomas de una persona puede ser expresado como el color, donde este puede ser azul u otro color). Cuando miramos a una persona, vemos la representación fenotípica de la persona, pero cada característica de la persona está precisamente escrita en su cromosoma, la representación genotípica de la persona. (Ver Figura 3.8)



**Figura 3.8:** Representación binaria de un individuo formado por un cromosoma. [21]

### Inicio Población

Después de la elección esquema representación, hay que realizar la creación de la población. Esta se genera creando cadenas de bit aleatoriamente. El tamaño de la población es un parámetro que se ajusta. Una vez elegida la representación de las soluciones y generada la población inicial, se esta en condiciones de aplicar operaciones genéticas a tales cadena con esperanza de encontrar las mejores

soluciones. Sin embargo, antes de describir las operaciones usadas en AG(s), describiremos un paso intermedio, la asignación de medida bondad de ajuste (fitness) para cada solución en la población

### Asignación de Fitness a la Solución

Es importante reiterar que AG binarios trabajan con representación de las variables decisión, en lugar de las variables decisión.

La evaluación de una solución significa calcular la función objetivo y violación restricciones. Después, una medida debe ser definida para usar el valor de la función objetivo y la violación de las restricciones para asignar un merito relativo a la solución (llamado fitness). En la ausencia de restricciones, el fitness (adaptabilidad) de la cadena es asignado como el valor de la función objetivo. Si existen restricciones en el problema y estas son violadas por las soluciones, la función objetivo debe ser penalizada.

### Operador Selección o Reproducción

El objetivo primario del operador selección es hacer duplicados de buenas soluciones y eliminar las soluciones malas de la población, mientras mantiene un tamaño constante de la población. Esto es logrado por el funcionamiento de las siguientes tareas:

1. Identificar buenas soluciones en una población
2. Hacer múltiples copias de buenas soluciones
3. Eliminar las soluciones malas de la población así que las múltiples copias de las buenas soluciones puedan agregarse en la población.

Existen un número de formas para lograr estas tareas. Algunos métodos comunes son torneo de selección, selección proporcional y selección ranking [22].

➤ Selección por torneo:

Se eligen dos soluciones al azar y el torneo es jugado, la solución con mejor adaptabilidad (fitness) es colocada en el mating pool <sup>2</sup>. Ejecutado sistemáticamente, cada solución puede participar en exactamente dos torneos. La mejor solución en una población ganará las dos veces, por consiguiente harán dos copias de ella en una nueva población. Usando similares argumentos, la peor solución perderá en ambos torneos y será eliminada de la población. De este modo, las peores soluciones no tendrán ninguna copia en la población. Se ha demostrado que la selección por torneo comparada con otros operadores de reproducción presenta mejores resultados en la complejidad del tiempo computacional.

➤ Selección proporcional

Las soluciones son copias asignadas, el número de copias es proporcional a sus valores del fitness. Si el fitness promedio de todos los miembros de la población es  $f_{avg}$ , una solución con un fitness  $f_i$  llega a una expectativa  $f_i/f_{avg}$  número de copias. La implementación de este operador de selección puede ser pensado como un mecanismo de rueda de ruleta, donde la rueda es dividida en  $N$  (tamaño población) divisiones, donde el tamaño de cada es marcada por la proporción al fitness de cada miembro de la población. Después, la rueda es girada  $N$  veces, cada vez eligiendo la solución indicada por la flecha.

### Operador Cruzamiento

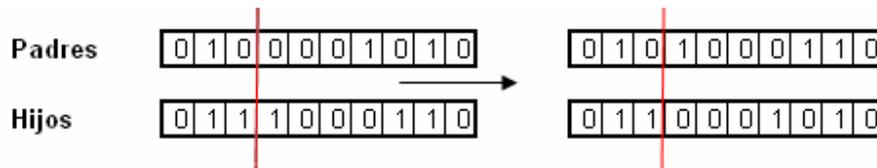
El operador cruzamiento es aplicado después a las cadenas del mating pool. La creación de nuevas soluciones es realizado por el operador de cruzamiento y mutación. Como el operador selección, existen un número de operadores de cruzamiento en la literatura.

---

<sup>2</sup> Mating pool es el lugar donde se colocan las soluciones que sobreviven al primer y segundo torneo

Todos los operadores de cruzamiento existentes tienen en común, que seleccionan dos cadenas que son elegidas del mating pool aleatoriamente. Luego, se selecciona un sitio aleatoriamente a lo largo de la cadena y los sitios a la derecha de este son cambiados entre las dos soluciones creando dos nuevas cadenas. [22]

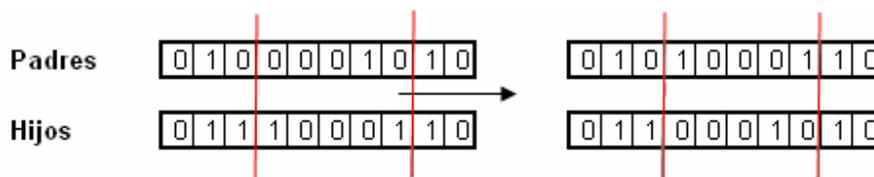
El operador de cruzamiento, single-point (punto simple) se selecciona un sitio de cruce a lo largo de la cadena y se intercambian todos los bits sobre la derecha del sitio de cruce. El operador simple-point se ilustra en la figura 3.9



**Figura 3.9:** Operador cruzamiento single-point

**Fuente:** Elaboración Propia

El concepto anterior de intercambio de información parcial entre dos cadenas puede también mejorar con más lugares de cruzamiento. En un operador cruzamiento dos puntos, dos sitios diferentes de cruzamiento son elegidos aleatoriamente. Esto divide la cadena en tres subcadenas. El operador cruzamiento es completado intercambiando los lugares entre los sitios de cruce. El operador two-point se ilustra en la figura 3.10



**Figura 3. 10:** Operador cruzamiento Two-point

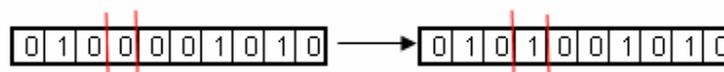
**Fuente:** Elaboración Propia

En orden a preservar algunas buenas cadenas seleccionadas durante el operador de cruzamiento, no todas las cadenas en la población son usadas como cromosomas. Si una probabilidad de cruzamiento  $P_c$  es usada, luego  $100\%P_c$  cadenas en la población serán usadas en la operación cruzamiento y  $(100-P_c)$  % de la población son simplemente copiados a la nueva población. Usualmente,  $P_c=0.5$  es utilizado.

En términos de extender la exploración (o búsqueda), un cruzamiento en un punto preserva la estructura del padre para maximizar la extensión en la descendencia. La extensión de la preservación de la descendencia se reduce con el incremento de sitios de cruce en el operador cruzamiento.

### Operador Mutación

El operador de cruzamiento es principalmente responsable por el aspecto búsqueda de algoritmos genéticos, el operador mutación es también usado para este propósito. El operador de mutación bit-wise cambia un 1 por un 0, y viceversa, con una probabilidad de mutación de  $p_m$ . La necesidad para la mutación es para mantener la diversidad en la población. Figura 3.11 ilustra como una cadena obtenida después del uso del operador de reproducción y cruzamiento ha sido mutado a otra cadena. Se elige un sitio aleatoriamente y se cambia a 1 o 0 según corresponda.



**Figura 3. 11:** Operador reproducción bit-wise

**Fuente:** Elaboración Propia

## **4 CAPITULO IV: REASIGNACION DE CAMIONES USANDO ALGORITMOS GENETICOS**

### **4.1 INTRODUCCIÓN**

La primera parte del capítulo describe el problema de la reasignación de camiones bajo un enfoque de la investigación de operaciones, describiendo el modelo matemático y las dificultades de resolver el problema mediante un algoritmo exacto. Luego, se describe la información de entrada para la implementación de los algoritmos. En la última parte del capítulo, se describe en forma minuciosa los dos algoritmos genéticos implementados y los resultados experimentales obtenidos. El primer algoritmo implementado **GA** (Genetic Algorithm), corresponde a un algoritmo genético con un tamaño de población constante y que conserva las mejores soluciones encontradas en cada generación en un archivo externo. El segundo algoritmo genético implementado **GALS** (Genetic Algorithm Local Search), es un algoritmo híbrido que adicionalmente aplica búsqueda local a cada individuo de la población en cada generación.

### **4.2 PROBLEMA DE REASIGNACIÓN DE CAMIONES**

La programación de ASICAM define una ventana de tiempo a todos sus despachos (Viajes) esta ventana corresponde al intervalo de tiempo que existe entre la hora de presentación a un determinado origen para cargar los productos demandados por un cliente y la hora de llegada a su correspondiente destino para descargarlos.

#### **4.2.1 Modelo Matemático**

La formulación propuesta en este estudio para del problema de reasignación se presenta a continuación: figura N° 4.1

**Datos de entrada**

C= Numero de Camiones

V= Numero de viajes

N= Numero Máximo Viajes Posibles a ser Asignados a un camión

$K_{ij}$  = Kilómetros entre el destino del viaje  $i \in V$  al origen del viaje  $j \in V$

$TD_{ij}$  =Tiempo disponibles entre el viaje  $i \in V$  y el viaje  $j \in V$

$TV_{ij}$  =Tiempo de viaje sin carga entre el destino del *viaje*  $i \in V$  al origen del *viaje*  $j \in V$

**Variables Decisión**

$X_{ij}^c$  =1 si al *camión*  $c$ , se le asigna el *viaje*  $i$  antes del *viaje*  $j$ , 0 en otro caso.

$X_{0j}^c$  =1 si el viaje  $j$  es el primer viaje efectuado por el camión  $c$ , 0 en otro caso.

$X_{i0}^c$  =1 si el viaje  $i$  es el ultimo viaje efectuado por el camión  $c$ , 0 en otro caso.

$$\text{Min} \sum_{c=1}^C \sum_{i=1}^V \sum_{\substack{j=1 \\ j \neq i}}^V X_{ij}^c * K_{ij} \quad (1)$$

s.t

$$X_{ij}^c TD_{ij} \geq X_{ij}^c TV_{ij} \quad \forall i = 1, \dots, V \quad j = 1, \dots, V \quad i \neq j \quad c = 1 \dots C \quad (2)$$

$$\sum_{i=0}^V \sum_{\substack{j=1 \\ j \neq i}}^V X_{ij}^c \leq N \quad \forall c \in C \quad (3)$$

$$\sum_{\substack{i=0 \\ i \neq j}}^V \sum_{c=1}^C X_{ij}^c = 1 \quad \forall j = 1, \dots, V \quad (4)$$

$$\sum_{j=1}^V X_{0j}^c = 1 \quad \forall c = 1, \dots, C \quad (5)$$

$$\sum_{i=1}^V X_{i0}^c = 1 \quad \forall c = 1, \dots, C$$

$$\sum_{\substack{i=0 \\ i \neq h}}^V X_{ih}^c = \sum_{\substack{j=0 \\ j \neq h}}^V X_{hj}^c \quad \forall h = 1, \dots, V \quad i = 1, \dots, C \quad (6)$$

(7)

**Figura 4.1: Modelo Matemático**

La función objetivo (1) de el modelo matemático suma la cantidad total de kilómetros sin carga efectuada por la flota de camiones. La restricción (2) requiere que el Tiempo Disponible (diferencia de tiempo entre hora de presentación en el próximo origen a visitar ( $O_{j+1}$ ) y la hora de llegada al último destino visitado ( $D_j$ ) por un camión cualquiera) debe ser mayor o igual al tiempo de traslado en vacío entre estos dos puntos ( $D_j$  a  $O_{j+1}$ ) (Ver figura N° 4.2). El conjunto de restricciones (3) asegura que a cada camión se le asigne a lo más N viajes. El conjunto de restricciones (4) asegura que a cada viaje sea programado solo una vez y sea efectuado por un camión. El conjunto de restricciones (5) y (6) asegura que cada camión tenga un viaje inicial y uno final respectivamente. El conjunto de restricciones (7) asegura que cada viaje tiene un predecesor y un sucesor en el camión.

$$\text{Tiempo Disponible} \geq \text{Tiempo traslado en vacío}$$

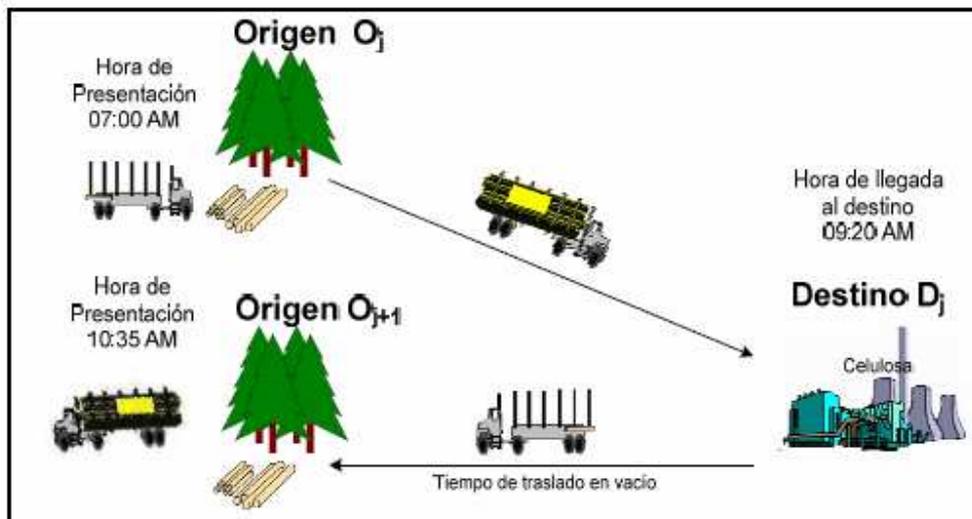


Figura 4.2: Restricción del problema [23]

El la figura N° 4.2 muestra mediante un ejemplo la restricción del problema. Si un camión termina su entrega a las 9:20 AM en el destino  $D_j$  y existe un nuevo trabajo que tiene como hora de presentación las 10:35 AM en el origen  $O_{j+1}$ , significa que hay un tiempo disponible de una hora y quince minutos ( $10:35 - 9:20$ ). Para que sea posible asignar este nuevo trabajo al camión, el tiempo de traslado en vacío

entre ambos puntos (de  $D_j$  a  $O_{j+1}$ ) deberá ser menor o igual al tiempo disponible, de lo contrario no alcanzará a estar a la hora de presentación fijada en el origen  $O_{j+1}$ , siendo infactible de realizar esta secuencia de trabajos.

El modelo matemático anterior posee un carácter combinatorio lo cual hace difícil su resolución mediante un método exacto debido a su gran complejidad computacional asociada a la categoría NP hard . La alternativa para resolver el problema es utilizar una metaheurística que mediante procedimiento de búsqueda y optimización obtenga resultados en tiempo computacional aceptable.

### **4.3 ORIGEN DE LOS DATOS**

Los datos de entrada al problema que se pretende optimizar provienen de dos fuentes específicas: la matriz de distancia y el programa entregado por ASICAM.

#### **4.3.1 Matriz distancia entre orígenes y destinos**

En ésta matriz se encuentran: todos los orígenes y todos los destinos, la distancia en kilómetros que existe entre los orígenes y los destinos (dividida entre kilómetros con pavimento y kilómetros sin pavimento), el valor de los peajes de ida y de vuelta, la tarifa del viaje entre un origen y un destino específico (medida en  $\$/m^3$  y en  $US\$/m^3$ ), el nombre del fundo y el punto de despacho.

#### **4.3.2 Programa entregado por ASICAM**

Es enviado diariamente por correo electrónico a cada transportista. Incluye los viajes asignados a cada patente, detallados con: numero ASICAM, producto, origen, destino, turno, horarios de llegada y salida a los orígenes y destinos.

## **4.4 INFORMACIÓN DE ENTRADA A LOS ALGORITMOS**

Para el diseño de los algoritmos se considero una empresa transportista que posee un total de quince camiones, por tanto se trabajó con quince patentes distintas a las cuales se les reasignaron los viajes. Se considero la información contenida en un programa entregado por ASICAM, correspondientes a un día elegido aleatoriamente. Los viajes realizados corresponden al día 13 Diciembre de 2005. En este, existen 40 viajes realizado por los quince camiones. Con la programación entregada por ASICAM y la matriz de distancia, se crearon tres matrices

### **4.4.1 Matriz de Distancia**

Esta matriz contiene la distancia en kilómetros entre los destinos de cada viaje ( $D_j$ ) y el origen siguiente ( $O_{j+1}$ ) del próximo viaje. Por lo tanto, son lo kilómetros recorridos sin carga que implican ir de un viaje a otro.

### **4.4.2 Matriz Tiempos Viajes en vacío**

La matriz de tiempos en vacío contiene el tiempo en minutos que lleva recorrer la cantidad de kilómetros en vacío. Para los cálculos se considero una velocidad para los camiones de 70 Km./hr en pavimento y 30 Km./hr en no pavimento.

### **4.4.3 Matriz Tiempo Disponible entre viajes**

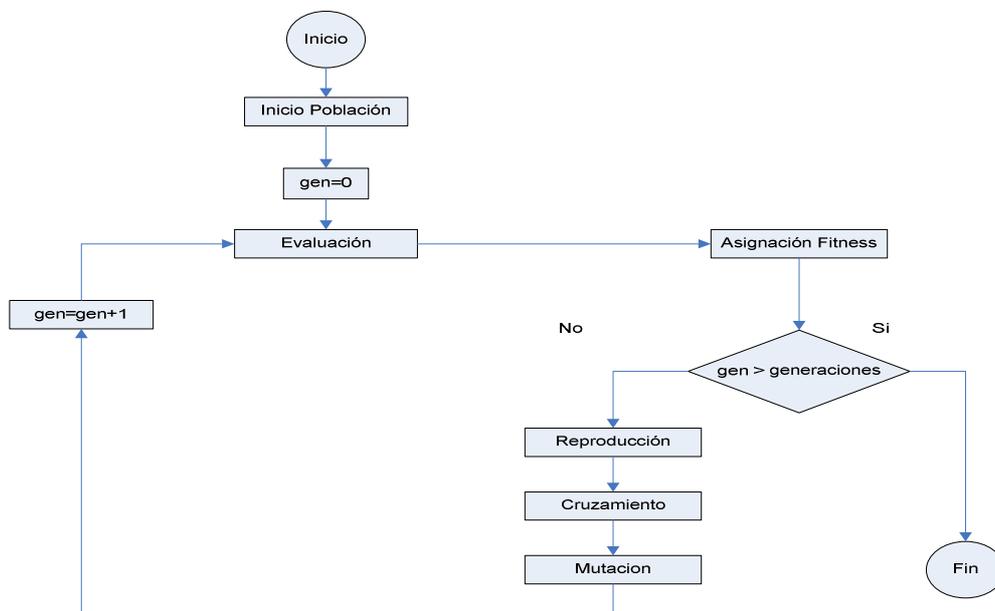
El tiempo disponible entre los viajes se calcula como la diferencia de tiempo entre hora de presentación en el próximo origen a visitar ( $O_{j+1}$ ) y la hora de llegada al último destino visitado ( $D_j$ )

## 4.5 DISEÑO ALGORITMO GENÉTICO (GA)

GA es un algoritmo genético, el cual fue adaptado al problema de tal manera que permitiera recorrer el espacio de búsqueda para obtener mejores soluciones que las entregadas por ASICAM. A continuación se presenta una descripción detallada de cada fase del **GA** implementado en C++. (Ver figura 4.3)

### 4.5.1 Descripción Algoritmo Genético (GA)

El algoritmo genético implementado en C++ se ilustra en la figura 4.3. Las fases del algoritmo son inicio población, la evaluación y asignación fitness, reproducción, cruzamiento, mutación. Además, este algoritmo contiene el principio estilista, guardando en un archivo externo solo las soluciones factibles. Al conservar las soluciones factibles al mismo tiempo guarda los mejores individuos de cada generación, debido a que por cada solución infactible la función objetivo es penalizada. El algoritmo genético conserva el tamaño de la población en toda la ejecución de este.

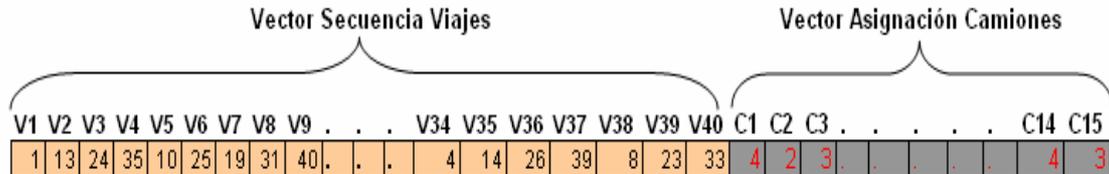


**Figura 4.3:** Diagrama Flujo Algoritmo Genético (GA)

**Fuente:** Elaboración Propia

### 4.5.2 Representación soluciones

Cada solución fue representada mediante un cromosoma, el cromosoma seleccionado consta de dos partes, la primera corresponde al vector secuencia de los viajes y la otra a l vector asignación de los camiones. (Figura 4.4)

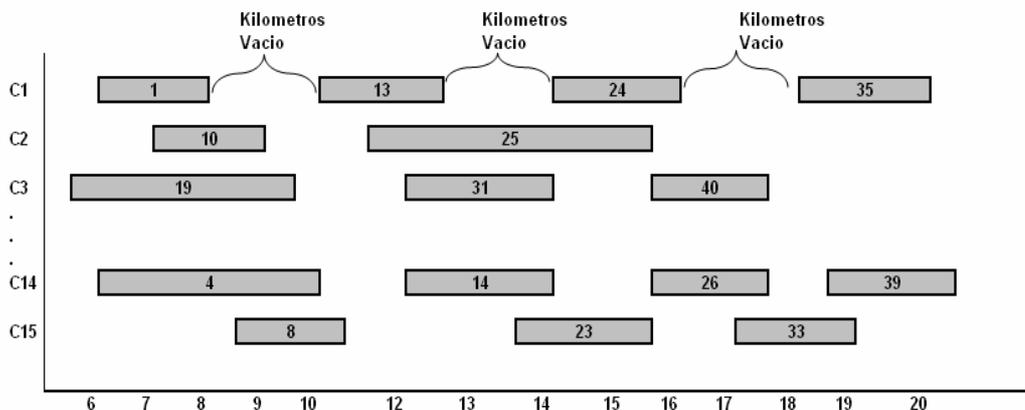


**Figura 4.4:** Representación de las soluciones

**Fuente:** Elaboración Propia

Vector secuencia de los viajes es un cromosoma numérico de genes único, por lo que no puede haber genes repetidos, esto indicaría que un mismo viaje es realizado por dos camiones. Esta parte del cromosoma debe estar enumerado con los números entre 1 y 40. (Total Viajes)

Vector asignación de los camiones indica la cantidad de viajes que realizara cada camión, por ejemplo el camión 1 realizara los viajes 1, 13, 24, 35. La representación grafica del cromosoma se presenta en la figura 4.5



**Figura 4.5:** Representación Grafica Cromosoma

**Fuente:** Elaboración Propia

### 4.5.3 Inicio Población

El inicio de la población se realizó en forma aleatoria, creando  $n$  individuos con idéntica estructura al cromosoma ilustrado en la figura 4.4. El tamaño de la población es un parámetro del algoritmo que se debe ajustar.

### 4.5.4 Evaluación y Asignación Fitness

Antes de la asignación de fitness las soluciones se deben evaluar. La evaluación consiste en calcular la función objetivo (cantidad total de kilómetros sin carga) de cada solución (cromosoma) y comprobar si la restricción mostrada en la figura 4.2 es violada.

Si la restricción no es violada el valor del fitness de la solución, corresponde al valor de la función objetivo. Si la restricción es violada, la solución será penalizada. La penalización corresponde a un castigo porcentual por cada viaje infactible en el cromosoma (solución). La penalización se ilustra en la figura 4.6.

$$Fitness = Km.vacio * \left(1 + \frac{X}{100}\right) * N$$

**Figura 4 6:** Penalización soluciones que violan la restricción

Donde  $X$ , corresponde al parámetro de castigo y  $N$  el número de trabajos infactibles.

### 4.5.5 Reproducción

La reproducción consiste en la selección de los individuos mejores adaptados (fitness), los cuales serán utilizados para crear nuevas soluciones utilizando los operadores de cruzamiento y mutación. Para la reproducción se implementó la selección por torneo. En la selección por torneo, se eligen dos soluciones de la población al azar y se enfrentan en un torneo, la solución con mejor adaptabilidad (fitness) es colocada en el mating pool. Sistemáticamente, cada solución puede participar en exactamente dos torneos. La mejor solución en una población ganará

las dos veces, por consiguiente harán dos copias de ella en una nueva población. Usando similares argumentos, la peor solución perderá en ambos torneos y será eliminada de la población. De este modo, las peores soluciones no tendrán ninguna copia en la población.

Para ilustrar este operador de selección se presenta un ejemplo en la figura 4.7. El tamaño de la población de ocho individuos, cada solución esta representada por  $S(i)$  y con su respectivo kilometraje sin carga. La primera etapa (fuera mating pool) se seleccionan dos soluciones aleatoriamente y juegan el torneo. Por ejemplo el torneo entre  $S(1)$  Y  $S(30)$  gana la solución  $S(1)$ , puesto que tiene un menor valor de fitness. Luego  $S(1)$  se enfrenta en un nuevo torneo a  $S(19)$  donde gana  $S(19)$  debido al menor valor de fitness. Al final la nueva población esta constituida por las soluciones que se encuentran en el mating pool, las peores soluciones son eliminadas de la población y las mejores son copiadas una y dos veces en la nueva población

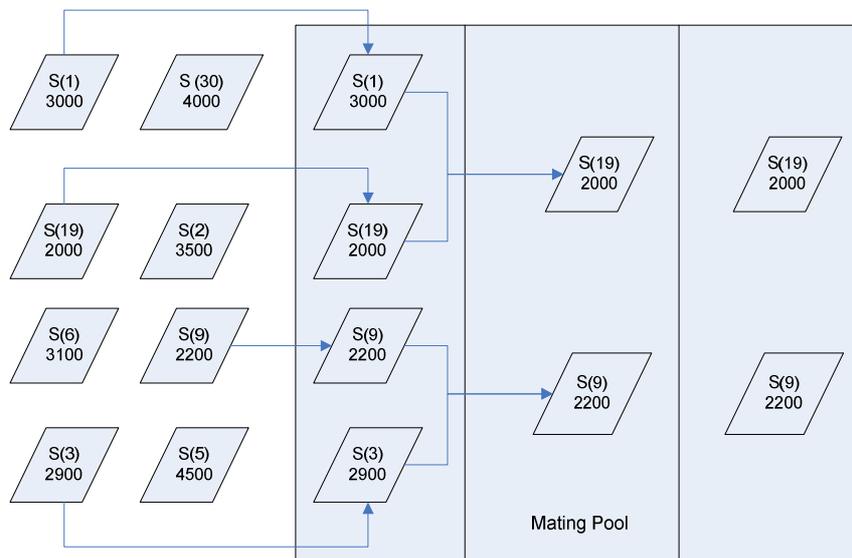


Figura 4 7: Selección Por Torneo

#### 4.5.6 Cruzamiento

Una vez seleccionado los individuos mejores adaptados de la población, el operador de cruzamiento tiene la tarea de crear nuevos individuos (hijos) tomando a las soluciones padres, combinando su material genético del cromosoma (genes). Este operador fue adaptado de [24]

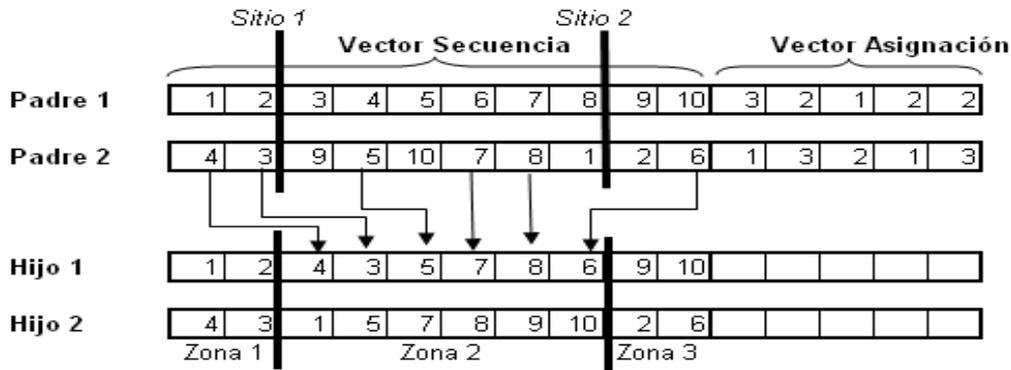
Para el cruzamiento se seleccionan dos individuos (Padres) de la población aleatoriamente, luego se genera un número aleatorio entre 0 y 1. Si el número aleatorio es menor que  $P_c$ <sup>4</sup> se realiza el cruzamiento, sino los individuos se copian en la nueva población. Para ilustrar este operador genético se considera un cromosoma de diez viajes y cinco camiones. El operador de cruzamiento se divide en dos sub-operadores, uno se encarga del cruzamiento de vector secuencia de los viajes y el otro del vector asignación de los camiones.

El operador de cruzamiento del vector de secuencia funciona de la siguiente manera: Primero se seleccionan dos sitios del cromosoma. Los sitios son seleccionados aleatoriamente sobre el vector de secuencia, procediéndose a definir 3 zonas de trabajos. Segundo, los hijos son creados copiando los genes de los cromosomas de los padres de la zona 1 y zona 3.

Finalmente, la zona 2 es creada buscando en el cromosoma del padre contrario los genes que no existen en el cromosoma del hijo de acuerdo a su orden de aparición. En la figura N° 4.8 se ilustra el operador cruzamiento del vector secuencia. El hijo 1 es creado copiando los genes de la zona 1 y zona 3 del cromosoma del padre 1. La zona 2 es creada buscando en el cromosoma del padre 2, los genes que no existen en el hijo 1 de acuerdo a su orden de aparición. Cabe mencionar que los genes de este vector son numéricos y sin repetición.

---

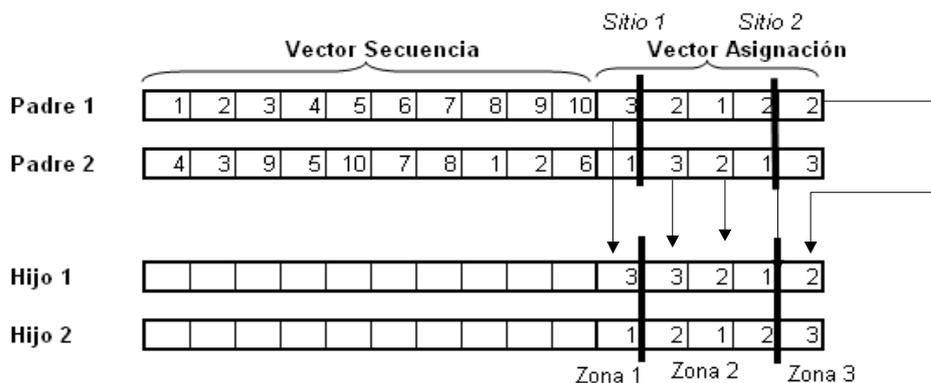
<sup>4</sup>  $P_c$  corresponde al porcentaje de cruzamiento de la población, el cual es un parámetro del algoritmo



**Figura 4 8:** Operador Cruzamiento Vector Secuencia

**Fuente:** Elaboración Propia

El operador cruzamiento del vector de asignación funciona de la siguiente forma: Primero se seleccionan dos sitios del cromosoma. Los sitios son seleccionados aleatoriamente sobre el vector de asignación, procediéndose a definir 3 zonas de trabajos. Segundo, los hijos son creados copiando los genes del cromosoma de los padres de la zona 1 y zona 3. Finalmente, la zona 2 es creada copiando los genes del cromosoma del padre contrario. En la figura N° 4.9 se ilustra el operador cruzamiento del vector de asignación. El hijo 1 es creado copiando los genes de la zona 1 y zona 3 del cromosoma del padre1. La zona 2, es creada copiando los genes del cromosoma de la zona2 del cromosoma del segundo padre. Cabe mencionar que la suma de los genes del vector de asignación debe ser igual al número de viajes, de tal forma que todo viaje sea asignado a un camión.

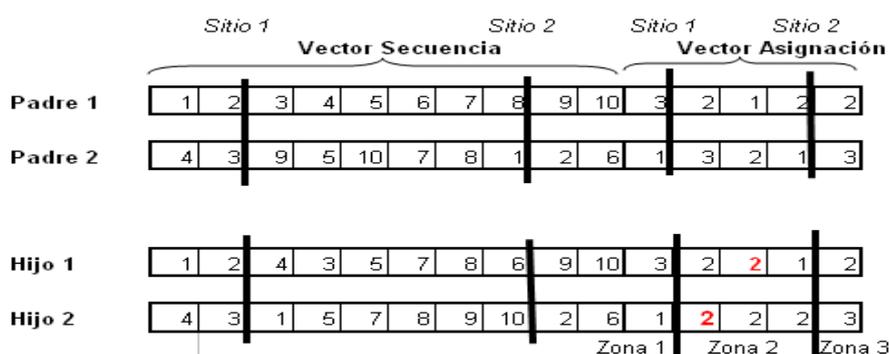


**Figura 4 9:** Operador Cruzamiento Vector Asignación

**Fuente:** Elaboración Propia

Las soluciones creadas (hijos) por el operador cruzamiento del vector asignación no son factibles, puesto que la suma del vector asignación debe ser igual al número de viajes (para el ejemplo 10). Para el hijo 1 la suma del vector de asignación es de 11 y para el hijo 2 la suma 9, ambas soluciones no son viables. Por lo tanto, hay que agregar una condición al operador cruzamiento del vector de asignación de tal forma que solo cree soluciones factibles.

Las condiciones son dos, primero si la suma de los genes del vector de asignación es menor que el total de viajes, se deben seleccionar posiciones aleatorias del vector de secuencia y sumar cantidades aleatorias hasta que la suma de los genes del vector de asignación sea igual al numero total viajes. Segundo, si la suma de los genes es mayor que el total de viajes, se deben seleccionar posiciones aleatorias del vector de asignación y restar cantidades aleatorias hasta que la suma de los genes del vector de secuencia sea igual al total de viajes. Aplicando estas condiciones al operador cruzamiento del vector de secuencia se obtiene los descendientes que se observan en la figura N° 4.10. (Los genes seleccionados a los cuales se les sumaron o restaron están marcados de color rojo)



**Figura 4.10:** Descendientes obtenido luego de aplicar el operador cruzamiento

**Fuente:** Elaboración Propia

Una vez creadas las soluciones hijos, las soluciones padres son eliminadas de la población. Las nuevas soluciones son copiadas en la nueva población, aunque posean un peor valor de la función de fitness.

#### 4.5.7 Mutación

En la mutación se genera un número aleatorio entre 0 y 1. Si el número aleatorio es menor que  $P_m$ <sup>5</sup> se realiza la mutación, sino los individuos se copian en la población nueva. El operador de mutación al igual del operador de cruzamiento se divide en dos sub-operadores, el operador de mutación del vector de secuencia de los viajes y el operador de mutación del vector de asignación de los camiones. Los hijos creados en la mutación reemplazan a los padres en la nueva población. Este operador fue adaptado de [24]

El operador de mutación del vector de secuencia funciona de la siguiente manera. Primero selecciona dos sitios aleatoriamente sobre los genes del vector de secuencia. Segundo, para crear el hijo inserta el gen ubicado en el *sitio 2* en la posición del gen del *sitio 1* desplazando todos genes en una posición incluyendo el gen seleccionado en la posición 1. (Figura N° 4.11)

El operador de mutación del vector de asignación funciona con idéntico principio. Seleccionando dos sitios aleatoriamente e insertando el gen ubicado en la posición 2 en la posición del gen 1 y desplazando a los demás. (Figura N° 4.12)

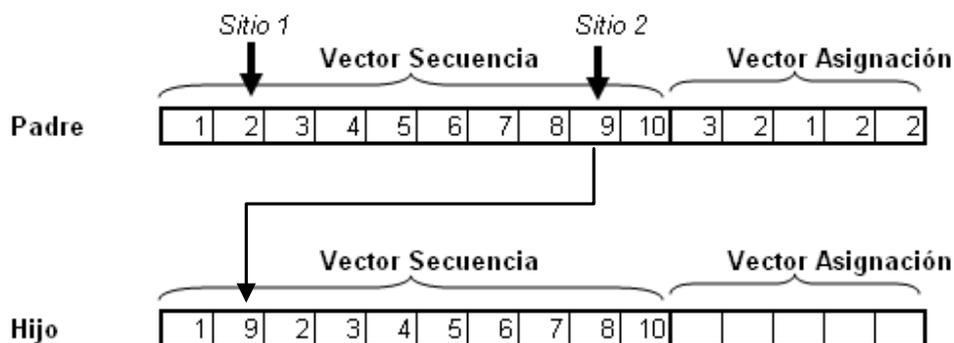
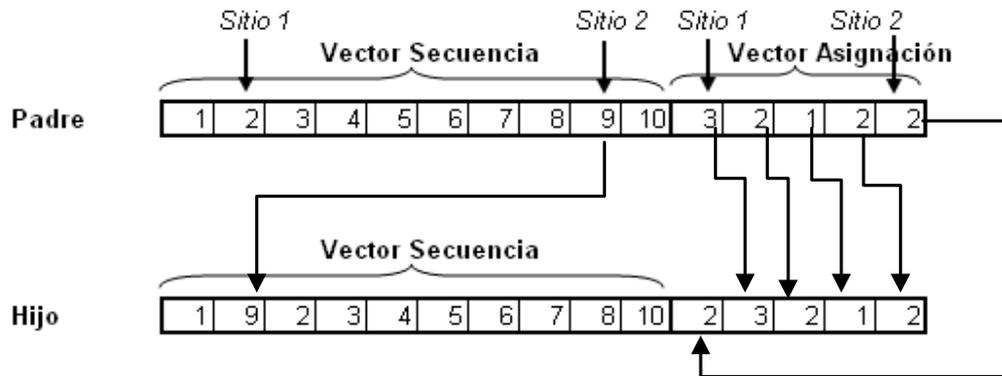


Figura 4.11: Operador mutación vector secuencia

Fuente: Elaboración Propia

<sup>5</sup>  $P_m$  corresponde al porcentaje de cruzamiento de la población, el cual es un parámetro del algoritmo



**Figura 4 12:** Operador mutación vector Asignación

Fuente: Elaboración Propia

#### 4.6 DISEÑO ALGORITMO GENÉTICO CON BÚSQUEDA LOCAL (GALS)

**GALS** es un algoritmo Híbrido propuesto en este trabajo, el cual adicionalmente de aplicar los conceptos de un algoritmo genético clásico utiliza la búsqueda local en cada individuo de la población. La idea de la búsqueda local es ampliar el espacio de búsqueda para mejorar la calidad de las soluciones encontradas. A continuación se presenta una descripción de cada fase del **GALS** implementado en C++ para resolver el problema de asignación de camiones. (Ver figura 4.13)

##### 4.6.1 Descripción Algoritmo Genético con Búsqueda Local (GALS)

La búsqueda se efectúa para cada individuo  $N^6$  veces, en cada iteración se evalúa la nueva solución creada. Si la nueva solución (hijo) tiene un mejor valor de fitness que la solución padre, la solución antigua es eliminada y en su replazo es insertada la nueva solución. Así, la nueva solución es utilizada para completar las iteraciones de la búsqueda local.

<sup>6</sup> **N** es el número de iteraciones que se ejecutara en la búsqueda local, lo cual es un parámetro del algoritmo

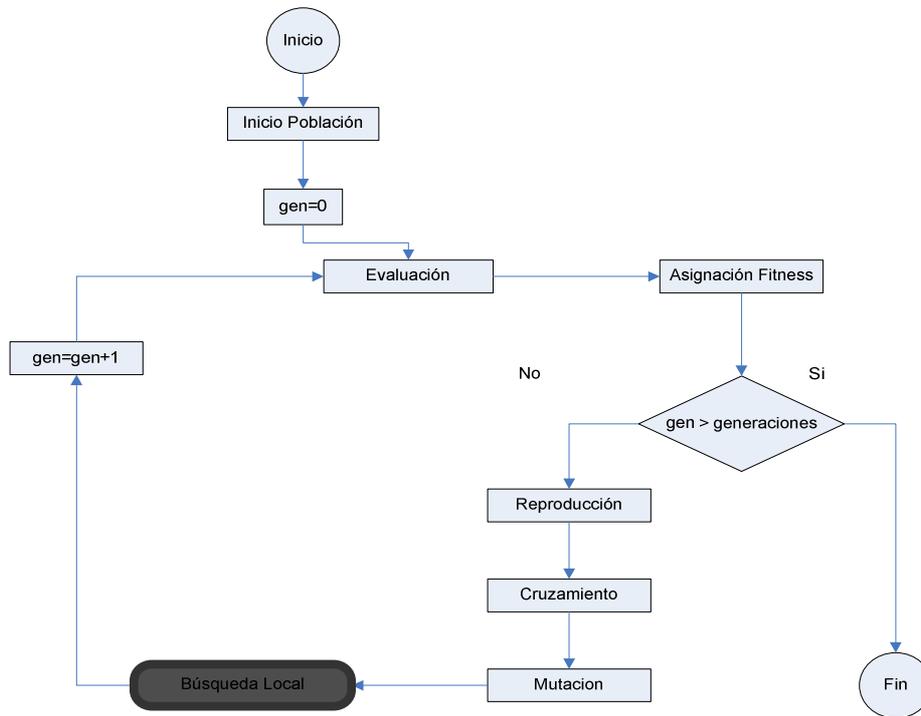
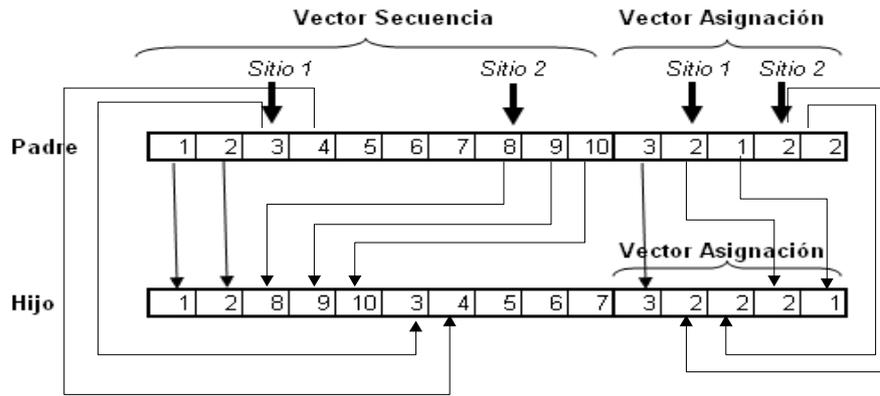


Figura 4.13 : Diagrama Flujo GALS

Fuente: Elaboración Propia

#### 4.6.2 Búsqueda Local

El procedimiento de búsqueda local que describe la estructura del vecindario se ilustra en la figura N° 4.14 La búsqueda local modifica la estructura del cromosoma del vector secuencia y del vector asignación. Para el vector de secuencia, genera dos posiciones aleatorias del cromosoma (sitio 1 y sitio 2), luego copia en el hijo los genes que se encuentran entre la posición cero y la posición anterior al sitio1 (*viaje 1 y 2*). Luego, copia en el hijo los genes que se encuentran entre la posición del sitio 2 y el término del vector de secuencia (*viajes 8, 9,10*). Para finalizar, copia en el hijo los genes que se encuentran entre la posición del sitio 1 y la anterior al sitio2 (*viajes 3, 4, 5, 6,7*). El cromosoma del vector de secuencia es de genes numéricos y sin repetición, la búsqueda local debe crear soluciones reales y factibles que permita recorrer el espacio de búsqueda. La búsqueda local del vector de asignación funciona de idéntica forma.



**Figura 4.14:** Búsqueda Local

**Fuente:** Elaboración Propia

## 5 CAPITULO V: ANALISIS EXPERIMENTAL Y RESULTADOS

### 5.1 ANÁLISIS EXPERIMENTAL

El análisis experimental se realiza con el objetivo de ajustar los parámetros de los algoritmos de tal manera que obtengan buenas soluciones. Para el ajuste de los parámetros, se utilizó la técnica estadística diseño de experimentos [25]. Específicamente un diseño experimental factorial de cinco factores para el algoritmo genético (**GA**) y un diseño factorial de tres factores para **GALS**.

#### 5.1.1 Ejecución Algoritmos

Para la ejecución de los algoritmos, se utilizó un procesador Intel Pentium dual-core de 1.83 GHz con 1.20 Gb de RAM y 120 Gb de disco duro.

#### 5.1.2 Parámetros

Los algoritmos poseen un conjunto de parámetros que pueden ser ajustados, dependiendo de estos la calidad de las soluciones encontradas. A continuación se define los parámetros utilizados.

- **Número de Generaciones:** Establece la cantidad de iteraciones que se realiza en cada corrida, es el criterio de detención del algoritmo.
- **Tamaño de la Población:** Es la cantidad de soluciones que hay dentro de una generación.
- **Porcentaje de Cruzamiento:** indica la probabilidad que tienes dos soluciones de seleccionadas para efectuar un cruzamiento de genes para crear descendientes
- **Porcentaje de Mutación:** Indica la probabilidad que tiene un gen de un cromosoma de ser cambiado aleatoriamente

- **Porcentaje de Penalización:** Corresponde al porcentaje de penalización de la función objetivo cuando una solución genere secuencias infactibles dentro del cromosoma
- **Numero iteraciones:** Establece la cantidad de iteraciones que se realizara para la búsqueda local

El algoritmo genético (**GA**) utiliza los cinco primero parámetros y el algoritmo genético híbrido (**GALS**) emplea los seis parámetros.

### 5.1.3 Diseño de experimentos

El diseño de experimento del algoritmo (**GA**) se presenta en la tabla N° 5.1. Los factores corresponden a los parámetros y los niveles los distintos valores que se le asignan a cada parámetro. Estos valores fueron elegidos efectuando replicas pilotos. *La variable de respuesta Y* corresponde al número de soluciones factibles encontradas en cada ejecución del algoritmo

$$Y = T.poblacion * Generaciones * P.Cruzamiento * P.Mitacion * Penalizacion$$

Factores	Niveles		
	1	2	3
Tamaño Población	500	1000	-
Generaciones	1000	2000	-
Porcentaje Cruzamiento	50	70	80
Porcentaje Mutación	40	60	-
Penalización (%)	10	20	-

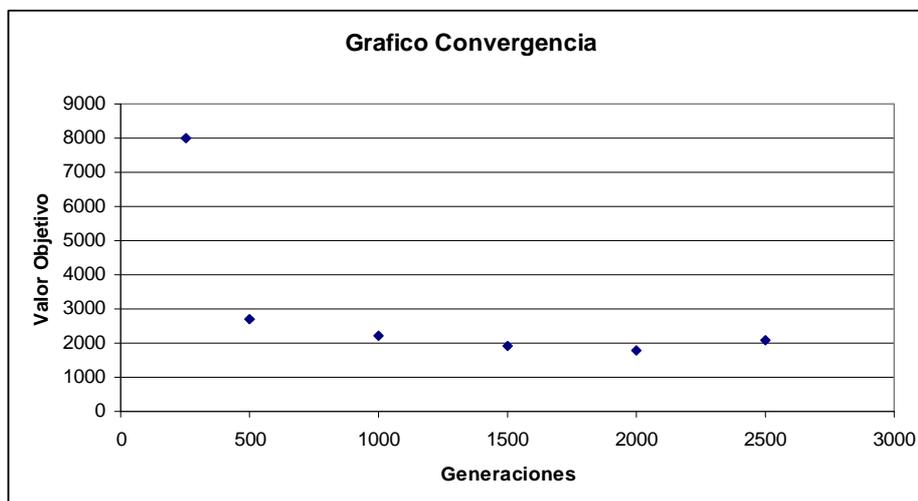
**Figura 5.1:** Factores y niveles del diseño experimental (GA)

**Fuente:** Elaboración Propia

Los tratamientos del diseño corresponden a las distintas combinaciones de los factores para cada nivel, existen un total de cuarenta y ocho combinaciones. Para cada combinación se ejecuto diez replicas, por lo tanto se ejecuto el algoritmo cuatrocienta ochenta veces registrando el numero de soluciones factibles obtenidas. El detalle del experimento para cada tratamiento y replica se encuentra en el ANEXO A

De los resultados obtenidos por el experimento factorial diseñado, se puede inferir:

- Al aumentar al doble el tamaño de la población, la cantidad de soluciones factibles aumenta en igual proporción en la mayoría de los casos.
- El aumento de la población va de la mano con el aumento en el tiempo de ejecución del algoritmo, debido a la mayor carga computacional
- El aumento del número de generaciones provoca un aumento el tiempo de ejecución, debido a la mayor carga computacional
- La interacción más significativa se produce entre el porcentaje de cruzamiento y el de mutación, a tal extremo que una combinación no adecuada no genera solución factible alguna. Por lo tanto, una combinación acertada de los parámetros en la eficiencia y eficacia del algoritmo
- . La convergencia de las soluciones se alcanza para un número de generaciones de dos mil. (Grafico 5.1)
- El porcentaje de penalización no afecta el tiempo de ejecución del algoritmo, pero afecta la cantidad de soluciones encontradas.



**Figura 5. 2:** Grafico Convergencia de las soluciones

**Fuente:** Elaboración Propia

- La interacción de los parámetros que produce la mayor cantidad de soluciones se presenta en la tabla 5.2

Parámetros	GA
Tamaño Población	1000
Generaciones	2000
Porcentaje Cruzamiento	70
Porcentaje Mutación	40
Penalización	20

**Figura 5. 3:** Ajuste de parámetros GA

**Fuente:** Elaboración Propia

La variable de respuesta analizada en el experimento analiza la cantidad total de soluciones factibles encontradas en cada iteración, no analiza la calidad de soluciones. La combinación de los factores que genera la mejor calidad de soluciones se presenta en la figura 5.3

Parámetros	GA
Tamaño Población	500
Generaciones	2000
Porcentaje Cruzamiento	80
Porcentaje Mutación	40
Penalización	10

**Figura 5.4:** Mejor Combinación de parámetros GA

**Fuente:** Elaboración Propia

El diseño de experimento del algoritmo genético con búsqueda local propuesto **(GALS)** se presenta en la tabla N° 5.4. Los factores corresponden a los parámetros y los niveles los distintos valores que se le asignan a cada parámetro. Estos valores fueron elegidos efectuando replicas pilotos. *La variable de respuesta* Y corresponde al número de soluciones factibles encontradas en cada ejecución del algoritmo

El tamaño de la población, el número de generaciones y la penalización de la función objetivo se considero como un bloque y fue seleccionado de los buenos resultados obtenidos por el experimento factorial anterior, para el cual se obtuvieron las mejores soluciones. El tamaño de población fue de 500 individuos,

un número de dos mil generaciones y un diez por ciento de penalización. El modelo del diseño experimental se presenta a continuación:

$$Y = P.Cruzamiento * P.Mitacion * Iteraciones$$

Factores	Niveles		
	1	2	3
Porcentaje Cruzamiento	80	50	30
Porcentaje Mutación	50	30	15
Iteraciones	10	30	50

**Figura 5. 5:** Factores y niveles del diseño experimento (GALS)

**Fuente:** Elaboración Propia

Los tratamientos del diseño corresponden a las distintas combinaciones de los factores para cada nivel, existen un total de veinte y siete combinaciones. Para cada combinación se ejecuto diez replicas, por lo tanto se ejecuto el algoritmo doscientas veinte veces registrando el numero de soluciones factibles obtenidas. El detalle del experimento para cada tratamiento y replica se encuentra en el ANEXO B

De los resultados obtenidos por el experimento factorial diseñado, se puede inferir:

- La interacción mas importante es la producida por el porcentaje de cruzamiento y mutación
- Un aumento en el número de iteraciones produce un aumento del tiempo de ejecución del algoritmo.
- La interacción de los parámetros que genero la mayor cantidad de soluciones se presenta en la tabla 5.6
- La interacción de parámetros que genero la mejor calidad de las soluciones se presenta en la tabla 5.7

Parámetros	GALS
Tamaño Población	500
Generaciones	2000
Porcentaje Cruzamiento	30
Porcentaje Mutación	15
Penalización	10
Iteraciones	50

**Figura 5.6:** Combinación de parámetros GALS que produce la mayor cantidad de soluciones factibles

**Fuente:** Elaboración Propia

Parámetros	GALS
Tamaño Población	500
Generaciones	2000
Porcentaje Cruzamiento	50
Porcentaje Mutación	15
Penalización	10
Iteraciones	50

**Figura 5.7:** Mejor Combinación de parámetros GA

**Fuente:** Elaboración Propia

Para el análisis comparativo de los algoritmos se utilizó la combinación que genere las mejores soluciones (valor fitness) y se efectuaron un total de veinte replicas para cada algoritmo. Los parámetros utilizados para el análisis comparativo se muestra en la figura 5.8.

Parámetros	GA	GALS
Tamaño Población	500	500
Generaciones	2000	2000
Porcentaje Cruzamiento	80	50
Porcentaje Mutación	40	15
Penalización	10	10
Iteraciones	-	50

**Figura 5.8:** Parámetros para el análisis comparativo

**Fuente:** Elaboración Propia

No se puede efectuar el análisis con idénticos porcentajes de cruzamiento y mutación, ya que cada algoritmo se comporta en forma diferente. Si se utilizara idénticos porcentajes, un algoritmo generaría un buen número de soluciones mientras el otro no generaría solución alguna.

## 5.2 RESULTADOS OBTENIDOS POR LOS ALGORITMOS

La asignación entregada por Asicam genera un total de 2504 kilómetros sin carga y una jornada promedio de 7 horas por camión. En la jornada de trabajo no está considerado el viaje desde la empresa al primer destino ni el viaje de retorno a la empresa transportista. Los resultados obtenidos por los algoritmos se describen a continuación:

### 5.2.1 Algoritmo Genético (GA)

El algoritmo genético propuesto (**GA**) obtuvo una solución que reduce los kilómetros en vacío en un 31,7%. La solución generó un total de 1745 kilómetros sin carga para quince camiones y 2133 kilómetros para catorce camiones. El gráfico 5.9 se ilustra el resultado obtenido comparado con la solución generada por Asicam.

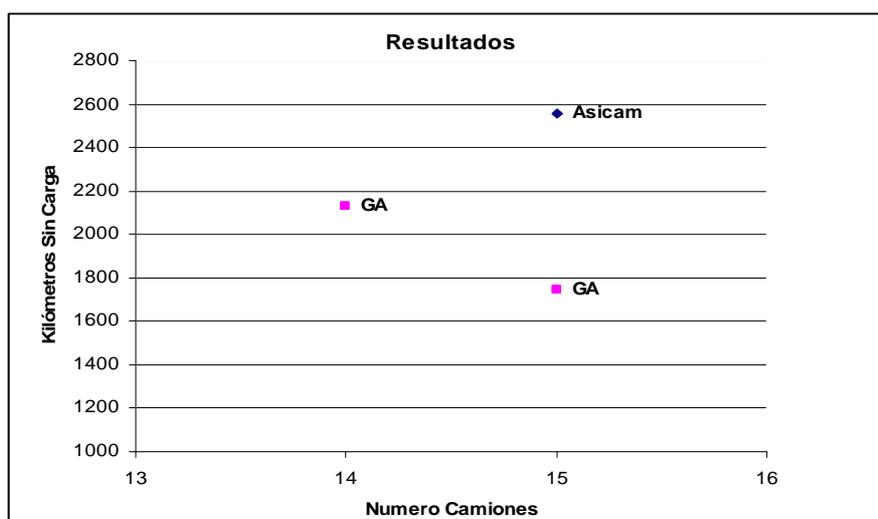


Figura 5.9: Gráfico Resultados de Algoritmo Genético (GA)

### 5.2.2 Algoritmo Genético Híbrido (GALS)

El algoritmo genético con búsqueda local propuesto (**GALS**) logro una solución que reduce los kilómetros en vacío en un 31,9%. La solución genero un total de 1740 kilómetros sin carga para quince camiones. El grafico 5.10 se ilustra el resultado obtenido comparado con la solución generada por Asicam.

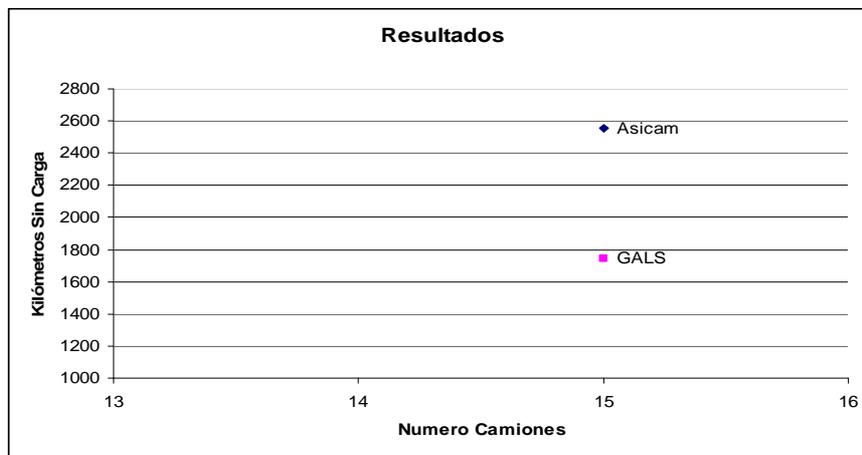


Figura 5.10: Grafico Resultados de Algoritmo Genético híbrido (GALS)

### 5.2.3 Análisis comparativo

Los algoritmos genéticos propuestos (**GA**) y (**GALS**) generaron una mejor solución comparada con la obtenida por Asicam considerando como objetivo de optimización la cantidad total de kilómetros sin carga.

La solución obtenida por **GALS** obtuvo una mejora de un 0,2% comparada con la solución obtenida por **GA**. El grafico 5.11 se ilustra el análisis comparativo de las soluciones obtenidas por los algoritmos propuestos con la solución generada por Asicam

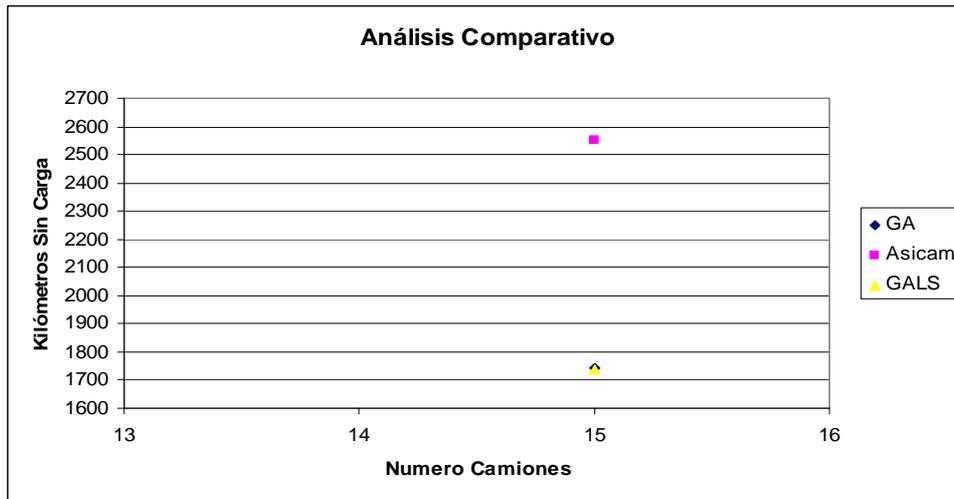


Figura 5. 11: Grafico Resultados Comparativos

### 5.2.4 Óptimos Generales

El óptimo general encontrado fue generado por el algoritmo **GALS** con un total de 1740 kilómetros sin carga. Reduciendo en 814 kilómetros la actual asignación. Los resultados se ilustran en el grafico 5.12

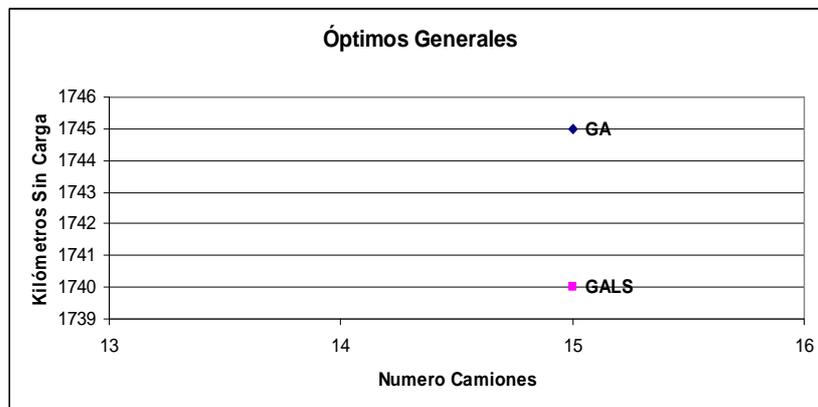


Figura 5.12: Grafico Óptimos Globales

A continuación en la tabla 5.13, se presenta la programación para los distintos camiones obtenida por ASICAM y la obtenida por GA.

Programación de ASICAM					Programación de GA				
Camiones	Viajes				Camiones	Viajes			
C1	6	29	36	-	C1	10	19	32	-
C2	3	17	-	-	C2	3	18	26	39
C3	10	21	32	-	C3	4	14	34	-
C4	11	-	-	-	C4	16	27	35	-
C5	2	18	28	-	C5	6	-	-	-
C6	1	16	27	35	C6	5	17	-	-
C7	12	26	38	-	C7	13	29	38	-
C8	5	15	-	-	C8	12	21	30	-
C9	13	24	33	-	C9	11	25	-	-
C10	8	22	39	-	C10	9	20	37	-
C11	7	37	-	-	C11	24	31	40	-
C12	9	20	30	40	C12	7	28	-	-
C13	34	-	-	-	C13	8	23	36	-
C14	4	14	23	-	C14	2	15	-	-
C15	19	25	31	-	C15	1	22	33	-

Figura 5. 13: Comparación de la programación

La programación de ASICAM genera una jornada laboral promedio de 7 horas por camión, esta jornada no incluye los tiempos de viajes desde la ubicación de la empresa transportista al primer destino y ni el retorno desde el último viaje a la empresa. Mientras, GA encuentra una programación que arroja una jornada laboral promedio de 6 horas y 34 minutos.

### 5.2.5 Tiempo Ejecución

El tiempo promedio de ejecución del algoritmo genético **AG** fue de 20 segundos, mientras que el tiempo promedio del algoritmo **GALS** fue de 100 segundos. El aumento del tiempo de ejecución es efecto del aumento de sentencias realizadas por el algoritmo provocado por el procedimiento de la búsqueda local

### 5.3 RESULTADOS ECONOMICOS

El algoritmo genético con búsqueda local propuesto (**GALS**) genero una solución que reduce los kilómetros sin carga en un 31,9%. Logrando una disminución de 814 kilómetros. Con la información de la figura 5.12, se calculo una reducción en el consumo de combustible de 281 litros.

Reduccion (Km)	814
Rendimiento (Km/L)	2,9
Precio Combustible (\$/L)	400
Litros Ahorro	281

Considerando veinte y cinco días de trabajo, se proyecta un aumento de la productividad mensual por camión de \$504.828 o un aumento en margen de la empresa transportista de \$ 7.572.414.

Bajo el supuesto que un camión facture mensualmente \$2.000.000. Con la reasignación efectuada por los algoritmos propuestos, se lograra un aumento de la productividad por camión de un 25 por ciento.

## 6 CAPITULO VI: CONCLUSIONES Y TRABAJO FUTURO

Los algoritmos propuestos disminuyen en aproximadamente un 31 por ciento la cantidad de kilometrajes sin carga en relación con la actual asignación generada por ASICAM, es decir es posible reducir la cantidad total de kilómetros recorridos en 814 kilómetros. Lo cual se traduce directamente en un aumento de la productividad de un 25 por ciento por cada camión. En relación a la jornada de trabajo, los algoritmos logran una disminución en la jornada promedio de la flota en un 6 por ciento. Cabe mencionar que las dos metaheurísticas propuestas superan significativamente la asignación empleada por la empresa (Asicam).

Los resultados obtenidos benefician directamente a las empresas subcontradas para el transporte de productos forestales debido a que estas reciben solo el pago por kilómetro recorrido con carga, y el costo del recorrido sin carga es asumido íntegramente por la empresa. Por lo que el aumento en la productividad por camión afectara positivamente el margen de utilidad de estas. Si bien el problema de reasignación esta planteado para solo una empresa de transporte y un día de programación, se puede ampliar su alcance al resto de las empresas transportistas y a todos los días de operación.

La gran diferencia de los algoritmos propuestos con los métodos clásicos de optimización es que utilizan una población de soluciones en vez de una única solución, el tiempo de ejecución no es exponencial y requieren una información mínima del problema.

El algoritmo genético híbrido (**GALS**) entrega resultados que son mejores que el algoritmo genético (**AG**) en un 0,2 por ciento. En relación al tiempo de ejecución, **GALS** demora cinco veces el tiempo de **GA**. Por lo tanto, la búsqueda local no tiene ningún efecto positivo en la calidad de las soluciones encontradas, solo empeora el desempeño del algoritmo genético.

Los algoritmos están diseñados de tal forma que emplean tres matrices (matriz tiempo disponible, matriz viajes en vacío, matriz distancias), una recomendación es combinar las dos primeras matrices y crear una matriz (matriz de tiempo factible) la cual se utiliza para evaluar la restricción del problema. Este cambio, disminuirá el tiempo de ejecución de los algoritmos.

La restricción del problema afecta el recorrido del espacio de búsqueda de los algoritmos, por ende la calidad de las soluciones. En este trabajo para evaluar la restricción se utilizó una matriz de tiempos de viajes sin carga calculada con una velocidad de 70 Km/h en pavimento y 30 Km/h no pavimento. Por lo tanto, efectuar un análisis de sensibilidad de estos tiempos afectará positivamente las soluciones generadas por los algoritmos.

Una forma de mejorar la eficiencia de los algoritmos es optimizar el código fuente. Las mejoras pueden ser reduciendo el número de matrices y mejorando los procedimientos de los algoritmos.

La interacción de los operadores de cruzamiento y mutación rigen el desempeño de los algoritmos, por lo que la incorporación de nuevos operadores y comparación con los implementados en este trabajo constituyen futuras líneas de investigación. Las sugerencias son diseñar nuevos procedimientos de mutación y cruzamiento.

Un estudio de la convergencia de los algoritmos propuestos, la incorporación de nuevos procedimientos de búsqueda local, que permitan mejorar la calidad de las soluciones presenta futuras líneas de investigación.

Algunas sugerencias para trabajos futuros son: efectuar diseños de experimentos adicionales a los efectuados para ajustar los parámetros de los algoritmos y realizar un análisis de sensibilidad del tiempo de viaje en vacío, de tal forma que permita reducir el número de soluciones infactibles.

La incorporación de nuevos objetivos que beneficien a las empresas mandantes o el bienestar de los trabajadores, tales como: la reducción del número de camiones o la reducción de la jornada laboral de trabajo abren un campo muy interesante de investigación. La nueva línea de investigación para los autores es incorporar nuevos objetivos al problema, y por lo tanto pasar de un problema mono-objetivo a uno multi-objetivo.

La información necesaria para el funcionamiento de los algoritmos consta de 3 matrices de un tamaño definido por la cantidad de viajes. Para que la integración de los algoritmos con Asicam sea factible, hay que efectuar un análisis intermedio de la programación entregada por este software y la matriz de distancia. El análisis debe generar las matrices con las cuales los algoritmos basan su funcionamiento

## 7 BIBLIOGRAFIA Y REFERENCIAS

[1] CORMA. 2009. Recurso Forestal. [En línea]  
<http://www.corma.cl> [consulta: 17 marzo 2009]

[2] ANDRES WEINTRAUB. 1990. Un Sistema de Asignación de Camiones para el Transporte de Productos Forestales. Revista de Ingeniería de Sistemas, Universidad de Chile. 7(1). Pág. 73-83.

[3] CONAF. 2009. Antecedentes generales del uso actual del suelo. [En línea]  
[www.conaf.cl](http://www.conaf.cl) [Consulta: 5 marzo 2009].

[4] INFOR 2009. Estadísticas Exportaciones. [En línea]  
[http://www.infor.cl/estadisticas\\_mercado/mercado\\_forestal/Mercado%20Forestal%20N34\\_JulioSeptiembre08.pdf](http://www.infor.cl/estadisticas_mercado/mercado_forestal/Mercado%20Forestal%20N34_JulioSeptiembre08.pdf) [consulta: 3 marzo 2009]

[5] WEINTRAUB, A. “Empresas Forestales y su Mirada Logística”. Revista Logistic

[6] CHILE PAIS FORESTAL 2009. Ciclo forestal. [En línea]  
[www.chilepaisforestal.cl](http://www.chilepaisforestal.cl) [consulta: 17 marzo 2009]

[7] ANDRES WEINTRAUB. “Cosecha Forestal y Logística Operacional”. Revista Logistec (LGT), edición nº 24, Capitulo Nº 1, Sección en La Mira.  
<http://www.logistec.cl>.

[8] BAÑADOS R. 2006. “Meta-heurísticas Híbridas para Optimización Mono-objetivo y Multi-objetivo. Paralelización y Aplicaciones”. Tesis para obtener el grado de Doctor En Informática... España, Almería. Universidad De Almería. Departamento De Arquitectura De Computadores y Electrónica. Pág.3.

[9] Blum C., Roli, A., Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison, ACM Computing Surveys 35(3), 2003, pp. 268-308.

[10] Santana J, Rodríguez C, García F, 2004. Metaheurísticas: una revisión actualizada. Departamento de Estadística, Investigación Operativa y Computación. Universidad de La Laguna, España.

[11] García J, .2006. Algoritmos Basados en Cúmulos de Partículas Para la Resolución de Problemas Complejos.

- [12] Jacobson, S.H., Yucesan, E., Analyzing the Performance of Generalized Hill Climbing Algorithms, *Journal of Heuristics* 10(4), 2004, pp. 387-405.
- [13] S. Kirkpatrick, C. Gelatt, and M. Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598):671\_680, 1983.
- [14] F. Glover. Future Paths for Integer Programming and Links to Artificial Intelligence. *Computers & Operations Research*, 13:533\_549, 1986.
- [15] F. Glover 1997. Heuristics for Integer Programming Using Surrogate Constraints. *Decision Sciences*, Pag 156-166
- [16] M. Dorigo. Optimization, Learning and Natural Algorithms. PhD thesis, Dipartimento di Elettronica, Politécnico di Milano, 1992.
- [17] Moscato, P., Memetic Algorithms, In *Handbook of Applied Optimization*, P.M. Pardalos y M.G.C. Resende (eds.), Oxford University Press, 2000.
- [18] Chicano J. 2007. Metaheurísticas e Ingeniería del Software. Tesis Doctoral. España, universidad de Málaga, Departamento Lenguajes y Ciencias de la Computación. Pag 57-61
- [19] Díaz B. 2002. Algoritmos Evolutivos Celulares con Ratio Vecindario-Población Dinámica. Memoria para obtener el título de E.T.S.I. Informática. España, universidad de Málaga, Departamento Lenguajes y Ciencias de la Computación. Pag 5-9
- [20] Golberg D.E 1989. Genetic Algorithms for search, optimization, and machine Learning. Pág. 20
- [21] Castro S. 2005. Creación de Portafolios de Inversión utilizando Algoritmos Evolutivos Multiobjetivo. Tesis de Grado de Maestro en Ciencias en la Especialidad de Ingeniería Eléctrica Opción Computación. México D.F. Depto. Ingeniería Eléctrica Sección Computación. Pág. 13
- [22] Deb, K 2001. Multi-objective Optimization Using Evolutionary Algorithms. Pág. 87-93
- [23] Marinado D, Matamala M. 2007. "Reprogramación de la Asignación de Camiones Para el Transporte de Productos Forestales bajo un Enfoque Multiobjetivo". Memoria de Ingeniero Civil Industrial. Concepción, Universidad del Bío-Bío, Facultad Ingeniería. Pág. 57

[24] Deb, K 2001. Multi-objective Optimization Using Evolutionary Algorithms. Pág. 434-435

[25] Montgomery, D.C. (2002). Diseño y Análisis de Experimentos. Editorial Limusa.

## 8 ANEXOS

### ANEXO A: Diseño de experimento del algoritmo genético (GA)

	Ejecuciones GA															Promedio
	Población	Generaciones	Pc	Pm	% Penalización	1	2	3	4	5	6	7	8	9	10	
1	500	1000	50	40	10	0	0	0	0	8	0	0	0	0	0	1
2	1000	1000	50	40	10	0	0	0	0	0	0	0	0	0	0	0
3	500	2000	50	40	10	0	23819	0	0	0	0	55	0	0	0	2387
4	1000	2000	50	40	10	0	0	91184	0	0	0	4	0	0	0	9119
5	500	1000	70	40	10	71021	0	70099	0	79792	0	61709	67302	0	67152	41708
6	1000	1000	70	40	10	0	117993	137215	139654	112571	147609	131832	126276	0	144099	105725
7	500	2000	70	40	10	0	17	8	0	171258	152248	162884	0	154167	141875	78246
8	1000	2000	70	40	10	27550	34134	294793	319330	318404	1	297090	0	314212	27655	163317
9	500	1000	80	40	10	0	46080	0	0	10381	48609	49795	2	0	45737	20060
10	1000	1000	80	40	10	0	81898	33	0	97117	113792	35463	912	115621	34728	47956
11	500	2000	80	40	10	125069	0	138185	134641	131867	0	141419	168653	0	134222	97406
12	1000	2000	80	40	10	253260	236132	287182	18321	0	296612	255344	0	215642	176621	173911
13	500	1000	50	60	10	0	0	0	0	0	0	0	0	0	0	0
14	1000	1000	50	60	10	0	0	0	0	0	0	0	0	0	0	0
15	500	2000	50	60	10	0	0	0	0	0	0	0	0	0	0	0
16	1000	2000	50	60	10	0	0	0	0	0	0	0	0	0	0	0
17	500	1000	70	60	10	0	0	0	0	0	0	0	0	0	0	0
18	1000	1000	70	60	10	0	0	0	0	0	0	0	0	0	0	0
19	500	2000	70	60	10	0	0	0	0	0	0	0	0	0	0	0
20	1000	2000	70	60	10	0	0	0	0	0	0	0	0	0	0	0
21	500	1000	80	60	10	0	0	0	0	0	0	0	0	0	0	0
22	1000	1000	80	60	10	0	0	0	0	0	0	0	0	0	0	0
23	500	2000	80	60	10	0	0	0	0	0	0	0	0	0	0	0
24	1000	2000	80	60	10	0	0	0	0	0	0	0	0	0	0	0
25	500	1000	50	40	20	6	0	0	0	0	0	0	0	0	0	1
26	1000	1000	50	40	20	0	0	184917	183709	195007	0	0	0	0	0	56363



**ANEXO B: Diseño de experimento del algoritmo genético (GALS)**

	Tamaño Población	Generaciones	Pc	Pm	Iteraciones	Ejecuciones GALS										Promedio	
						1	2	3	4	5	6	7	8	9	10		
1	500	2000	80	50	10	0	0	0	0	0	0	0	0	0	0	0	0
2	500	2000	50	50	10	0	0	0	0	0	0	0	0	0	0	0	0
3	500	2000	30	50	10	0	0	0	84448	0	12219	1	0	71601	0	16827	
4	500	2000	80	30	10	0	0	0	0	0	0	0	0	0	0	0	
5	500	2000	50	30	10	0	261982	1	0	0	0	0	0	47121	0	30910	
6	500	2000	30	30	10	0	0	0	0	0	0	4	0	27166	241980	26915	
7	500	2000	80	15	10	0	0	0	0	0	0	0	0	0	0	0	
8	500	2000	50	15	10	0	0	0	413449	8016	0	0	0	0	0	42147	
9	500	2000	30	15	10	1234	0	0	0	0	370880	0	271447	297795	0	94136	
10	500	2000	80	50	30	0	0	0	0	0	0	0	0	0	0	0	
11	500	2000	50	50	30	0	0	0	0	0	0	0	0	0	0	0	
12	500	2000	30	50	30	0	18685	99359	80732	0	1217	0	91365	0	73079	36444	
13	500	2000	80	30	30	0	0	0	0	0	0	0	0	0	0	0	
14	500	2000	50	30	30	11	13233	0	51750	0	48	225173	47766	111611	244609	69420	
15	500	2000	30	30	30	121771	342344	115197	0	15222	80427	80427	0	0	0	75539	
16	500	2000	80	15	30	0	0	0	0	0	0	0	0	0	0	0	
17	500	2000	50	15	30	488557	278653	2	432845	0	207308	0	186547	465198	4595	206371	
18	500	2000	30	15	30	349751	0	0	0	53	501945	0	0	2	1	85175	
19	500	2000	80	50	50	0	0	0	0	0	0	0	0	0	0	0	
20	500	2000	50	50	50	0	0	0	0	0	0	0	0	0	0	0	
21	500	2000	30	50	50	2318	2992	28093	12390	0	3	1	0	52641	34312	13275	
22	500	2000	80	30	50	0	0	0	0	0	0	0	0	0	0	0	
23	500	2000	50	30	50	37850	148405	119867	264475	42394	85564	0	10	240457	0	93902	
24	500	2000	30	30	50	4	194547	337611	0	0	332577	174107	0	0	0	103885	
25	500	2000	80	15	50	0	0	0	0	0	0	0	0	0	0	0	
26	500	2000	50	15	50	1	125297	84300	0	411788	5	135132	0	279278	468490	150429	
27	500	2000	30	15	50	280077	535115	337565	568485	505647	0	525972	126564	475805	306110	366134	

