



UNIVERSIDAD AUTÓNOMA  
DE AGUASCALIENTES

## UNIVERSIDAD AUTÓNOMA DE AGUASCALIENTES

### CENTRO DE CIENCIAS BÁSICAS

“Diseño de un Modelo Basado en Algoritmos Genéticos (AG’s) Utilizando Migración para Mejorar el Proceso de Diversificación en la Resolución del Problema del Viajante de Comercio (TSP)”

### TESIS

QUE PARA OBTENER EL GRADO DE DOCTOR EN CIENCIAS DE LA COMPUTACIÓN  
INTELIGENCIA ARTIFICIAL

### PRESENTA

MC. FRANCISCO JAVIER ORNELAS ZAPATA

### TUTOR

DR. ALEJANDRO PADILLA DÍAZ

### COMITÉ TUTORAL

DRA. EUNICE ESTHER PONCE DE LEÓN SENTÍ

DR. FELIPE PADILLA DÍAZ

DR. ALBERTO OCHOA O. ZEZZATTI

DRA. ELVA DÍAZ DÍAZ

DRA. MA. DE LOURDES YOLANDA MARGAIN FUENTES

**Aguascalientes, Ags. Mayo del 2010.**

## **AGRADECIMIENTOS**

---

Agradezco primeramente a Dios por la vida, por mi familia, mi trabajo y por haber tenido la oportunidad de participar en un proyecto tan importante para mí en lo personal, académico y profesional.

A mis hijos Angela, Jose Antonio, Fransisco y Sophia y a mi esposa Gabriela por su paciencia y apoyo demostrado en cada día transcurrido desde el inicio hasta la culminación de este proyecto.

A mi madre Maria de Lourdes y a mi abuela Angela por estar a mi lado siempre apoyando cada uno de los proyectos que me he propuesto realizar.

A mi director de tesis Alejandro por su amistad y apoyo no solo en este proyecto sino en el ámbito profesional y laboral.

A mis asesores Dr. Felipe Padilla, Dra. Eunice Ponce de Leon, Dr. Alberto Ochoa, Dra. Elva Diaz y a la Dra Lourdes Margain, por la luz que han aportado a mi crecimiento intelectual y crecimiento y fortalecimiento de mi proyecto.

A mis amigos Alejandro, Miguel, Julio, Alberto, Sayuri, Aurora, Dolores, Alfonso y Ricardo por el apoyo incondicional que día a día he recibido de ellos.

Y a todos aquellos que de forma directa o indirecta han participado en este trabajo y que escapan a mi memoria les extiendo mi agradecimiento y reconocimiento.

## DEDICATORIAS

---

Dedico este trabajo a mis hijos que son el más grande e importante de mis proyectos.



## CARTAS DE LIBERACIÓN



UNIVERSIDAD AUTÓNOMA  
DE AGUASCALIENTES



UNIVERSIDAD AUTÓNOMA  
DE AGUASCALIENTES  
Comemoración del Bicentenario del inicio de la Independencia de México  
y del Centenario de la Revolución Mexicana

Centro de Ciencias Básicas

**M. en C. FRANCISCO JAVIER ORNELAS ZAPATA  
PASANTE DEL DOCTORADO EN CIENCIAS  
DE LA COMPUTACIÓN  
P R E S E N T E .**

Estimado (a) Alumno (a) Ornelas:

Por medio de este conducto me permito comunicar a Usted que habiendo recibido los votos aprobatorios de los revisores de su trabajo de tesis titulado: **“Diseño de un Modelo Basado en Algoritmos Genéticos (AG,s) Utilizando Migración para Mejorar el Proceso de Diversificación en la Resolución del Problema del Viajante de Comercio (TSP)”**, hago de su conocimiento que puede imprimir dicho documento y continuar con los trámites para la presentación de su examen de grado.

Sin otro particular me permito saludarle muy afectuosamente.

ATENTAMENTE  
Aguascalientes, Ags., 14 de junio de 2010  
“LUMEN PROFERRE”  
EL DECANO

DR. FRANCISCO JAVIER ÁLVAREZ RODRÍGUEZ



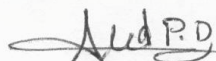
c.c.p.- Archivo

Aguascalientes, Ags. Mayo de 2010

**DR. FRANCISCO JAVIER ÁLVAREZ RODRÍGUEZ  
DECANO DEL CENTRO DE CIENCIAS BÁSICAS  
UNIVERSIDAD AUTÓNOMA DE AGUASCALIENTES  
P R E S E N T E.**

Por este conducto hago de su conocimiento que el **M.C. Francisco Javier Ornelas Zapata** egresado del **Doctorado en Ciencias de la Computación** del Centro de Ciencias Básicas de la Universidad Autónoma de Aguascalientes, ha integrado satisfactoriamente el documento de tesis titulado **“Diseño de un Modelo Basado en Algoritmos Genéticos (AG’s) Utilizando Migración para Mejorar el Proceso de Diversificación en la Resolución del Problema del Viajante de Comercio (TSP)”**, por lo que doy mi voto aprobatorio para que continúe con los trámites para presentar el examen de grado reglamentario.

ATENTAMENTE



---

DR. ALEJANDRO PADILLA DÍAZ

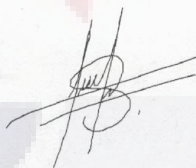


Aguascalientes, Ags. Mayo de 2010

**DR. FRANCISCO JAVIER ÁLVAREZ RODRÍGUEZ  
DECANO DEL CENTRO DE CIENCIAS BÁSICAS  
UNIVERSIDAD AUTÓNOMA DE AGUASCALIENTES  
P R E S E N T E.**

Por este conducto hago de su conocimiento que el **M.C. Francisco Javier Ornelas Zapata** egresado del **Doctorado en Ciencias de la Computación** del Centro de Ciencias Básicas de la Universidad Autónoma de Aguascalientes, ha integrado satisfactoriamente el documento de tesis titulado **“Diseño de un Modelo Basado en Algoritmos Genéticos (AG’s) Utilizando Migración para Mejorar el Proceso de Diversificación en la Resolución del Problema del Viajante de Comercio (TSP)”**, por lo que doy mi voto aprobatorio para que continúe con los trámites para presentar el examen de grado reglamentario.

**ATENTAMENTE**



---

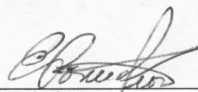
**DR. FELIPE PADILLA DÍAZ**

Aguascalientes, Ags. Mayo de 2010

**DR. FRANCISCO JAVIER ÁLVAREZ RODRÍGUEZ  
DECANO DEL CENTRO DE CIENCIAS BÁSICAS  
UNIVERSIDAD AUTÓNOMA DE AGUASCALIENTES  
P R E S E N T E.**

Por este conducto hago de su conocimiento que el **M.C. Francisco Javier Ornelas Zapata** egresado del **Doctorado en Ciencias de la Computación** del Centro de Ciencias Básicas de la Universidad Autónoma de Aguascalientes, ha integrado satisfactoriamente el documento de tesis titulado **“Diseño de un Modelo Basado en Algoritmos Genéticos (AG’s) Utilizando Migración para Mejorar el Proceso de Diversificación en la Resolución del Problema del Viajante de Comercio (TSP)”**, por lo que doy mi voto aprobatorio para que continúe con los trámites para presentar el examen de grado reglamentario.

ATENTAMENTE



---

**DRA. EUNICE ESTHER PONCE DE LEÓN SENTÍ**

Aguascalientes, Ags. Mayo de 2010

**DR. FRANCISCO JAVIER ÁLVAREZ RODRÍGUEZ  
DECANO DEL CENTRO DE CIENCIAS BÁSICAS  
UNIVERSIDAD AUTÓNOMA DE AGUASCALIENTES  
P R E S E N T E.**

Por este conducto hago de su conocimiento que el **M.C. Francisco Javier Ornelas Zapata** egresado del **Doctorado en Ciencias de la Computación** del Centro de Ciencias Básicas de la Universidad Autónoma de Aguascalientes, ha integrado satisfactoriamente el documento de tesis titulado **“Diseño de un Modelo Basado en Algoritmos Genéticos (AG’s) Utilizando Migración para Mejorar el Proceso de Diversificación en la Resolución del Problema del Viajante de Comercio (TSP)”**, por lo que doy mi voto aprobatorio para que continúe con los trámites para presentar el examen de grado reglamentario.

**ATENTAMENTE**



---

**DR. ALBERTO OCHOA ORTIZ ZEZZATTI  
ASESOR DE TESIS**

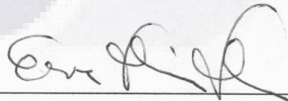


Aguascalientes, Ags. Mayo de 2010

**DR. FRANCISCO JAVIER ÁLVAREZ RODRÍGUEZ  
DECANO DEL CENTRO DE CIENCIAS BÁSICAS  
UNIVERSIDAD AUTÓNOMA DE AGUASCALIENTES  
P R E S E N T E.**

Por este conducto hago de su conocimiento que el **M.C. Francisco Javier Ornelas Zapata** egresado del **Doctorado en Ciencias de la Computación** del Centro de Ciencias Básicas de la Universidad Autónoma de Aguascalientes, ha integrado satisfactoriamente el documento de tesis titulado **“Diseño de un Modelo Basado en Algoritmos Genéticos (AG’s) Utilizando Migración para Mejorar el Proceso de Diversificación en la Resolución del Problema del Viajante de Comercio (TSP)”**, por lo que doy mi voto aprobatorio para que continúe con los trámites para presentar el examen de grado reglamentario.

ATENTAMENTE



---

**DRA. ELVA DÍAZ DÍAZ**

## RESUMEN

---

El presente trabajo es el resultado de una ardua investigación que abarca desde el sustento teórico que envuelve a los Algoritmos Genéticos (AG), las bases de las que surgen, su evolución hasta nuestros días, sus fortalezas y debilidades, el tipo de problemas que mejor resuelven, hasta la generación de nuevos recursos que pueden fortalecerlos y mejorarlos para resolver el problema del viajante de comercio y generar las rutas óptimas para la recolección de basura en San Francisco de los Romo.

Los AG están basados en la Teoría de la Evolución de las Especies propuesta por Charles Darwin (Biólogo Británico) donde se plantea la forma en que distintas especies han evolucionado por medio de un lento proceso de *selección natural*, que es como el medio ambiente influye en su supervivencia y/o reproducción [Darwin1859]. También influyeron los trabajos que realizó con aproximadamente 28,000 plantas de guisantes que aisló y polinizó de forma artificial Gregor Mendel (Monje Austriaco), para luego basándose en los resultados obtenidos de sus experimentos, describir las leyes que rigen la herencia y que actualmente apoyan los estudios que se realizan en genética [Riol2008].

Se generó un modelo que permite mejorar la diversificación de las poblaciones dentro de los AG's y que se refleja en la calidad de los resultados obtenidos; ésto se logra introduciendo el operador de *migración* (que ha sido utilizado en trabajos como los modelos de islas) y que junto con una adecuada elección de los operadores de selección, cruza y mutación permiten alcanzar dicho objetivo.

Para probar el modelo se resolvió el problema del viajante de comercio (TSP por sus siglas en inglés), ya que por sus características, durante muchos años ha sido uno de los problemas que se toman como base para probar distintas heurísticas.

El TSP consiste en encontrar la ruta más corta o menos costosa para recorrer cierto número de ciudades, pasando por todas sin que estas se repitan [Bryant2000]. Por esta razón se hace un estudio concienzudo del mismo.



## ÍNDICE

<i>Agradecimientos</i>	<i>i</i>
<i>Dedicatorias</i>	<i>ii</i>
<i>Cartas de liberación</i>	<i>iii</i>
<i>Resumen</i>	<i>ix</i>
<i>Índice</i>	<i>xi</i>
<b>CAPÍTULO 1</b>	<b>1</b>
<i>Introducción</i>	<b>1</b>
<i>1.1. Descripción General del Problema de Investigación</i>	<b>1</b>
<i>1.1.1. Algoritmos Genéticos (AG)</i>	1
<i>1.1.2. Genética de poblaciones</i>	2
<i>1.1.3. Problema del Viajante de Comercio</i>	3
<i>1.2. Justificación y Relevancia de la Investigación</i>	<b>4</b>
<i>1.3. Descripción General del Enfoque de Investigación, Métodos y Técnicas de Investigación.</i>	<b>6</b>
<b>CAPÍTULO 2</b>	<b>7</b>
<i>Estructuración de la Investigación</i>	<b>7</b>
<i>2.1. Descripción del Problema de Investigación</i>	<b>7</b>
<i>2.2. Objetivo General y Objetivos Específicos de la Investigación</i>	<b>8</b>
<i>2.2.1. Objetivo General</i>	8
<i>2.2.2. Objetivo Principal</i>	8
<i>2.2.3. Objetivos particulares</i>	9
<i>2.3. Pregunta General y Preguntas Específicas</i>	<b>9</b>
<i>2.3.1. Pregunta General</i>	9
<i>2.3.2. Preguntas Específicas</i>	10
<i>2.4. Metas e Hipótesis de la Investigación</i>	<b>10</b>
<i>2.4.1. Metas</i>	10
<i>2.4.2. Hipótesis principal</i>	11
<i>2.5. Descripción y Especificación del Enfoque, Métodos y Técnicas de Investigación</i>	<b>11</b>
<b>CAPÍTULO 3</b>	<b>13</b>
<i>Marco Teórico</i>	<b>13</b>
<i>3.1. Base Teórica</i>	<b>13</b>
<i>3.1.1 Computación Evolutiva (CE)</i>	13
<i>3.1.2. Algoritmos Genéticos</i>	16
<i>3.1.3. Genética de Poblaciones</i>	28

3.1.4. Problema del Viajante de Comercio	29
3.2. Trabajos Relacionados	31
<b>CaPÍTULO 4</b>	<b>38</b>
<b>Desarrollo y Validación</b>	<b>38</b>
4.1 Modelo Propuesto y Trabajo Realizado.	38
4.1.1. Migración o mutación	38
4.1.2. Mejoramiento del modelo propuesto	50
4.1.3. Diseño de experimentos	61
4.1.4. Costo computacional	63
<b>CAPÍTULO 5</b>	<b>71</b>
<b>Reporte y Discusión de Resultados</b>	<b>71</b>
<b>CAPÍTULO 6</b>	<b>73</b>
<b>Conclusiones y Trabajo Futuro</b>	<b>73</b>
6.1. Conclusiones	73
6.2. Trabajo Futuro	74
<b>Anexo A</b>	<b>76</b>
<b>Pseudocodigos</b>	<b>76</b>
A1. Genera individuo	76
A2. Evaluar individuo	77
A3. selecció_ elit	77
A4. selecció_ MUE.	78
A5. Cruzar individuo.	79
A6. Mutar	80
A7. Limpiar	81
<b>Anexo B</b>	<b>82</b>
<b>Artículos</b>	<b>82</b>
B1. MICAI '09	82
B2. COMCEV '08	89
B3. COMCEV '07	97
B4. Primer Congreso Estatal "La Investigación en el Posgrado"	107
B5. Segundo Congreso Estatal "La Investigación en el Posgrado"	109
B6. Tercer Congreso Estatal "La Investigación en el Posgrado"	111



<i>Anexo C</i>	<b>121</b>
<i>Datos utilizados en el diseño de experimentos</i>	<b>121</b>
<i>Glosario</i>	<b>148</b>
<i>Bibliografía</i>	<b>149</b>



## CAPÍTULO 1

### INTODUCCIÓN

---

#### 1.1. DESCRIPCIÓN GENERAL DEL PROBLEMA DE INVESTIGACIÓN

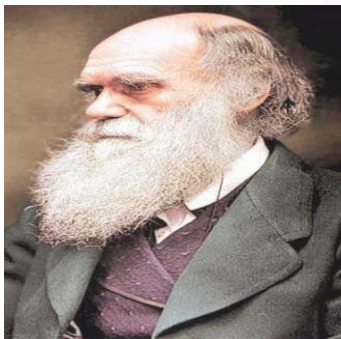
---

Generar un modelo basado en Algoritmos Genéticos que permita mejorar la forma en que trabajan, seleccionando las herramientas adecuadas para cada operador e introduciendo nuevos operadores para mejorar la diversificación de las poblaciones para resolver el TSP.

##### 1.1.1. ALGORITMOS GENÉTICOS (AG)

---

Los AG's son algoritmos bioinspirados que agrupan técnicas o métodos basados en la evolución natural y la genética y toman como base la "Teoría de la Evolución de las Especies" propuesta por *Charles Darwin* y los descubrimientos realizados por *Gregor Mendel* en el campo de la genética.



Figs. 1a y 1b Charles Darwin y Gregor Mendel

La Teoría de la Evolución de las Especies propuesta por Darwin maneja la idea de que dentro de la naturaleza los individuos que tienen algún o algunos factores que los favorecen (por pequeños que sean) sobre los miembros de su misma especie o sobre individuos de otras especies les dan ventajas para que sus características perduren en el proceso evolutivo. Por el contrario cualquier variación perjudicial (igualmente por pequeña que sea) tenderá a ser eliminada [Darwin1859]. De igual forma los AG's intentan que los individuos o posibles soluciones evolucionen hasta alcanzar un alto grado de competitividad lo que resultaría en una buena solución al problema que se desea resolver [Holland1975][Goldberg1989].

Los AG han demostrado ser una buena opción para resolver una gran variedad de problemas de optimización, ya que es relativamente fácil representar casi cualquier problema en ellos. Desafortunadamente con frecuencia convergen rápidamente a óptimos locales ya que el proceso de diversificación se realiza en una pequeña parte por el cruzamiento entre padres, pero el operador que produce la mayor cantidad de diversidad llamado *mutación* normalmente se utiliza en porcentajes muy pequeños debido a que es un proceso aleatorio y aún cuando su funcionamiento es adecuado, no discrimina entre soluciones que ya se tengan con una buena calidad y soluciones malas, lo que podría provocar la pérdida de algunas buenas soluciones y esto traería por consecuencia mayor carga de trabajo sobre el AG y en ocasiones quizás la pérdida de soluciones que resuelven de mejor manera el problema que las que se obtengan al finalizar su ejecución [Goldberg1989].

Por lo anterior se propone generar un nuevo modelo tomando como base el AG simple e introducirle algún o algunos operadores que permitan mejorar el proceso de diversificación basado en el proceso evolutivo descrito en las teorías de Genética de Poblaciones.

---

### ***1.1.2. GENÉTICA DE POBLACIONES***

---

“La genética de poblaciones es la disciplina de la biología que suministra los principios teóricos de la evolución” [Barbadilla2008]

En la genética de poblaciones se parte del supuesto de que los cambios evolutivos que se dan a pequeña escala dentro de una población, dentro de una especie tienen todos los elementos necesarios para explicar la evolución ya que la evolución a gran escala solamente sería una extrapolación de ésta [Barbadilla2008].

En general, cualquier especie está formada por una o más poblaciones en las que los individuos comparten su material genético por medio del cruzamiento (ya que los genes son el único componente que se transmite). Por esta razón, aquellos individuos que participen más activamente o por más tiempo en el proceso de cruzamiento tendrán más descendientes donde se conservaran sus características y estas estarán mayormente representadas en la siguiente generación [Barbadilla2008].

La combinación de distintos genes provocará cambios en el material genético de la población y estos serán normalmente irreversibles ya que es casi imposible que se vuelvan a dar combinaciones genéticas idénticas a alguna previa en distintas etapas del proceso evolutivo.

Dentro de la genética de poblaciones se manejan algunos factores que provocan la evolución y son: la *mutación*, la *deriva genética*, la *migración* y la *selección natural*.

---

### **1.1.3. PROBLEMA DEL VIAJANTE DE COMERCIO**

---

Uno de los problemas más antiguos, famosos y representativos de la optimización combinatoria es el Problema del Viajante de Comercio o TSP (por sus siglas en inglés “Traveling Salesman Problem”) ya que es uno de los más sencillos de plantear, pero detrás de esa sencillez de planteamiento se encuentra un problema muy difícil de resolver debido a que conforme crece el problema, el espacio de búsqueda de soluciones se incrementa exponencialmente [Fogel1988].



El TSP consiste en encontrar la menor distancia para que un viajante recorra N número de ciudades, visitándolas todas, pasando sólo una vez por cada una de ellas y por último regresar a la ciudad en la que inició el recorrido. La distancia de cada recorrido se obtiene sumando la distancia entre cada par de ciudades visitadas.

El TSP se ha utilizado ampliamente y se sigue utilizando como problema de prueba para verificar el funcionamiento de una gran cantidad de algoritmos.

Algunas de las aplicaciones reales donde se utiliza el TSP como base para su resolución son: el diseño de circuitos electrónicos; el ruteo de vehículo; el trazado de redes eléctricas; agua potable y alcantarillado, entre otros.



Fig.2 Muestra un ejemplo del menor costo de viaje de un agente de ventas aplicando el TSP

## 1.2. JUSTIFICACIÓN Y RELEVANCIA DE LA INVESTIGACIÓN

El presente trabajo surge de la necesidad e interés de tener algoritmos sencillos, adaptables, competitivos, que resuelvan los problemas de forma rápida y que los resultados obtenidos sean de muy buena calidad (lo más cercanos al resultado óptimo).



Los AG's cumplen ya con algunos de los elementos anteriormente descritos, por ejemplo; casi cualquier problema de optimización puede ser representado dentro de la estructura de los mismos por lo sencillo de los conceptos sobre los cuales fueron creados (basados en la naturaleza), también se ha comprobado que obtienen muy buenos resultados para casi cualquier tipo de problema de optimización. Desafortunadamente, por algunas de sus características, en ocasiones no son tan competitivos como otras herramientas heurísticas como colonia de hormigas, meméticos, EDAs, entre otros., que resuelven en menos tiempo o dan mejores resultados en determinados tipos de problemas.

El modelo propuesto pretende modificar o mejorar las características de los AG's que representen una debilidad en su funcionamiento.

Se considera que el proceso de diversificación de las poblaciones es deficiente y se propone identificar para cada operador la herramienta más adecuada para apoyar dicho proceso y adicionar algún operador que junto con el operador de *mutación* mejore esta tarea dentro del proceso evolutivo.

Si las modificaciones propuestas mejoran el desempeño de los AG's tendríamos un modelo más competitivo que se podrá utilizar en la industria para resolver los problemas de optimización a los que se enfrentan como son ruteo (rutas de comercialización, mejora de instalaciones eléctricas, de agua potable y alcantarillado, líneas de producción, rondines de patrullas, itinerario de entregas, entre otros) u optimización de recursos.

Se generará una herramienta de software que permita comparar el AG simple y el modelo propuesto. También se verificarán los resultados obtenidos y se compararán contra resultados de otras heurísticas en problemas del TSP.

Utilizando el modelo se resolverá un problema de optimización dentro de la región como caso de uso.

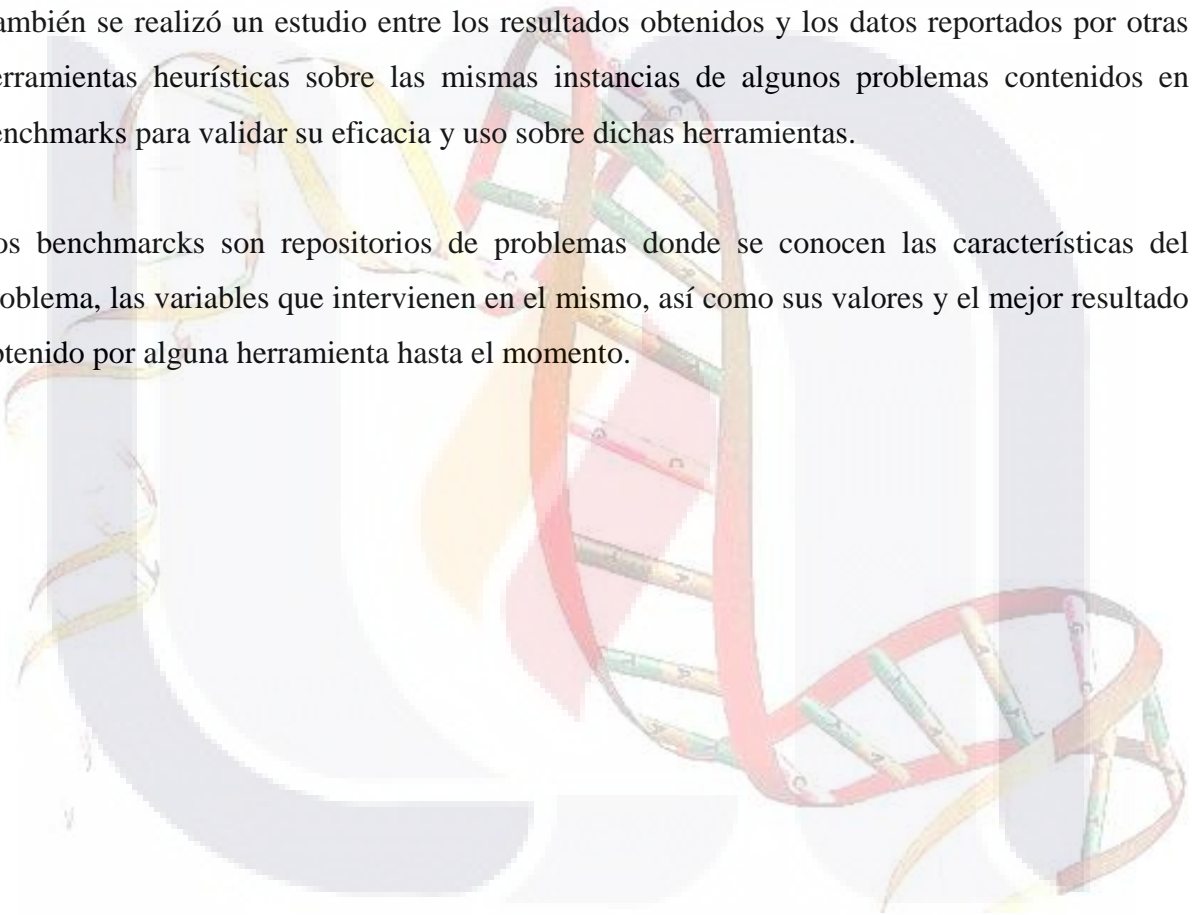
### ***1.3. DESCRIPCIÓN GENERAL DEL ENFOQUE DE INVESTIGACIÓN, MÉTODOS Y TÉCNICAS DE INVESTIGACIÓN.***

---

El enfoque dado a la investigación es el de experimentación y simulación, ya que se desarrolló un nuevo modelo como una herramienta de software y se probó haciendo un estudio comparativo contra el AG simple, sobre algunas instancias del TSPLIB.

También se realizó un estudio entre los resultados obtenidos y los datos reportados por otras herramientas heurísticas sobre las mismas instancias de algunos problemas contenidos en benchmarks para validar su eficacia y uso sobre dichas herramientas.

Los benchmarks son repositorios de problemas donde se conocen las características del problema, las variables que intervienen en el mismo, así como sus valores y el mejor resultado obtenido por alguna herramienta hasta el momento.



## CAPÍTULO 2

### ESTRUCTURACIÓN DE LA INVESTIGACIÓN

---

#### 2.1. DESCRIPCIÓN DEL PROBLEMA DE INVESTIGACIÓN

---

La optimización combinatoria es un área de las matemáticas aplicadas y de las ciencias computacionales y está relacionada con la teoría de la complejidad computacional. La teoría de la complejidad computacional estudia de forma teórica la cantidad de recursos computacionales que utilizará un algoritmo para resolver un problema [Wikipedia1].

Basándose en la complejidad computacional los problemas se clasifican en problemas P, NP y NP-Hard. Los problemas del tipo P son aquellos en los que existe un algoritmo de resolución que es polinomial con el tamaño del problema y en los NP no existe dicho algoritmo de resolución polinomial con el tamaño del problema [Moujahid2005].

Las herramientas heurísticas se utilizan para resolver problemas en los que un método de búsqueda exhaustiva es ineficiente o para aquellos en los que el espacio de búsqueda crece exponencialmente conforme crece el número de variables que intervienen en el problema [Moujahid2005].

El TSP es uno de los problemas más famosos y más estudiados de la optimización combinatoria. Este problema tiene como objetivo que un agente de ventas recorra determinado número de ciudades visitando solamente una vez cada una y por último regresar a la ciudad en que inició el recorrido, con el menor costo posible [Brest2005].

Este problema incrementa su complejidad conforme crece la cantidad de ciudades a visitar, por lo que al llegar a un determinado número, el problema se vuelve intratable por cualquier método de búsqueda exhaustiva, ya que por cada ciudad adicional crece su complejidad exponencialmente.

Los AGs han demostrado ser una buena herramienta para resolver problemas de explosión combinatoria, ya que logran obtener buenas soluciones en tiempos relativamente cortos. Desafortunadamente, su principal problema es que normalmente convergen prematuramente a resultados de buena calidad pero no necesariamente los mejores, esto debido a que el operador que utilizan para diversificar las poblaciones de posibles resultados debe ser utilizado en porcentajes muy pequeños [Goldberg1989], para que afecte lo menos posible los buenos resultados que se hayan obtenido hasta el momento, ya que dicho operador funciona de forma aleatoria y puede afectar tanto a buenos como a malos resultados, por lo que al incrementar su porcentaje se gana en diversificación pero se incrementa el riesgo de pérdida de buenos resultado con que se cuente.

## ***2.2. OBJETIVO GENERAL Y OBJETIVOS ESPECÍFICOS DE LA INVESTIGACIÓN***

---

### ***2.2.1. OBJETIVO GENERAL***

---

Diseñar un modelo que mejore el funcionamiento del AG simple, seleccionando para cada operador las mejores herramientas y adicionando operadores que permitan apoyar el proceso de diversificación de los mismos para resolver instancias del TSP contenidas en el TSPLIB.

### ***2.2.2. OBJETIVO PRINCIPAL***

---

Diseñar un modelo que explote las mejores características del AG simple, así como incrementemente el proceso de diversificación para obtener mejores resultados y a su vez mejorar su rendimiento.

---

### ***2.2.3. OBJETIVOS PARTICULARES***

---

Investigar los fundamentos de los AG's, de donde surgen; la teoría relacionada en su desarrollo; trabajos e investigaciones actuales; los operadores que intervienen; sus bondades y deficiencias; la teoría relacionada con Genética de Poblaciones, así como los fundamentos del Problema del Viajante de Comercio.

Desarrollar la herramienta de software que utilice el modelo de AG generado; hacer un estudio comparativo contra los resultados obtenidos por el AG simple sobre los mismos problemas y en las mismas circunstancias.

Verificar los resultados obtenidos para comprobar la eficiencia del nuevo modelo contra el AG simple.

Publicar los resultados obtenidos en algún congreso o revista, nacional o internacional.

### ***2.3. PREGUNTA GENERAL Y PREGUNTAS ESPECÍFICAS***

---

---

#### ***2.3.1. PREGUNTA GENERAL***

---



¿Es posible generar un modelo que mejore el proceso de diversificación en el AG simple seleccionando las herramientas con mejores características para cada operador y adicionar algún operador que apoye dicho proceso para mejorar la eficacia del mismo y así resolver de mejor forma instancias del TSP?

---

### ***2.3.2. PREGUNTAS ESPECÍFICAS***

---

¿Existe alguna combinación de herramientas (que representen a cada operador) que permitan mejorar el proceso de diversificación de las poblaciones?

¿Existe algún operador que pueda adicionarse o sustituir a los ya existentes en el AG simple para fortalecer el proceso de diversificación de las poblaciones?

¿Estas modificaciones mejoran el rendimiento del AG simple en la resolución del TSP?

## ***2.4. METAS E HIPÓTESIS DE LA INVESTIGACIÓN***

---

---

### ***2.4.1. METAS***

---

- Crear un modelo de AG que mejore el proceso de diversificación de las poblaciones dentro del AG simple.
- Generar el software para aplicar el modelo generado a resolver instancias del TSP.
- Verificar la funcionalidad del modelo en base a los resultados que se obtengan y comparar los resultados obtenidos contra algunas otras herramientas heurísticas que se utilizan para resolver las mismas instancias.

- Generar el material necesario para sustentar los fundamentos sobre los cuales está basada la propuesta de modificación del algoritmo.
- Analizar la complejidad computacional del algoritmo diseñado y tratar de aplicar el mismo a otros problemas de optimización.

---

#### **2.4.2. HIPÓTESIS PRINCIPAL**

---

Crear un modelo que sea capaz de mejorar el funcionamiento del AG simple adicionando o modificando los operadores de diversificación, en la resolución del TSP.

### **2.5. DESCRIPCIÓN Y ESPECIFICACIÓN DEL ENFOQUE, MÉTODOS Y TÉCNICAS DE INVESTIGACIÓN**

---

La creación de un nuevo modelo basado en AG es complicado debido a que existen actualmente muchos desarrollos de este tipo, donde se han realizado mejoras en los procedimientos de cruce, en la forma en que actúa la mutación, agregando algunos otros operadores para apoyar el proceso de diversificación, etc., lo que ha generado algoritmos más competitivos. Gracias a estas modificaciones se ha logrado disminuir las deficiencias de la diversificación de las poblaciones pero aún no se ha logrado corregir significativamente. Además, esto ha permitido generar algunos nuevos modelos que actualmente son líneas de investigación que evolucionan a la par de los AG's.

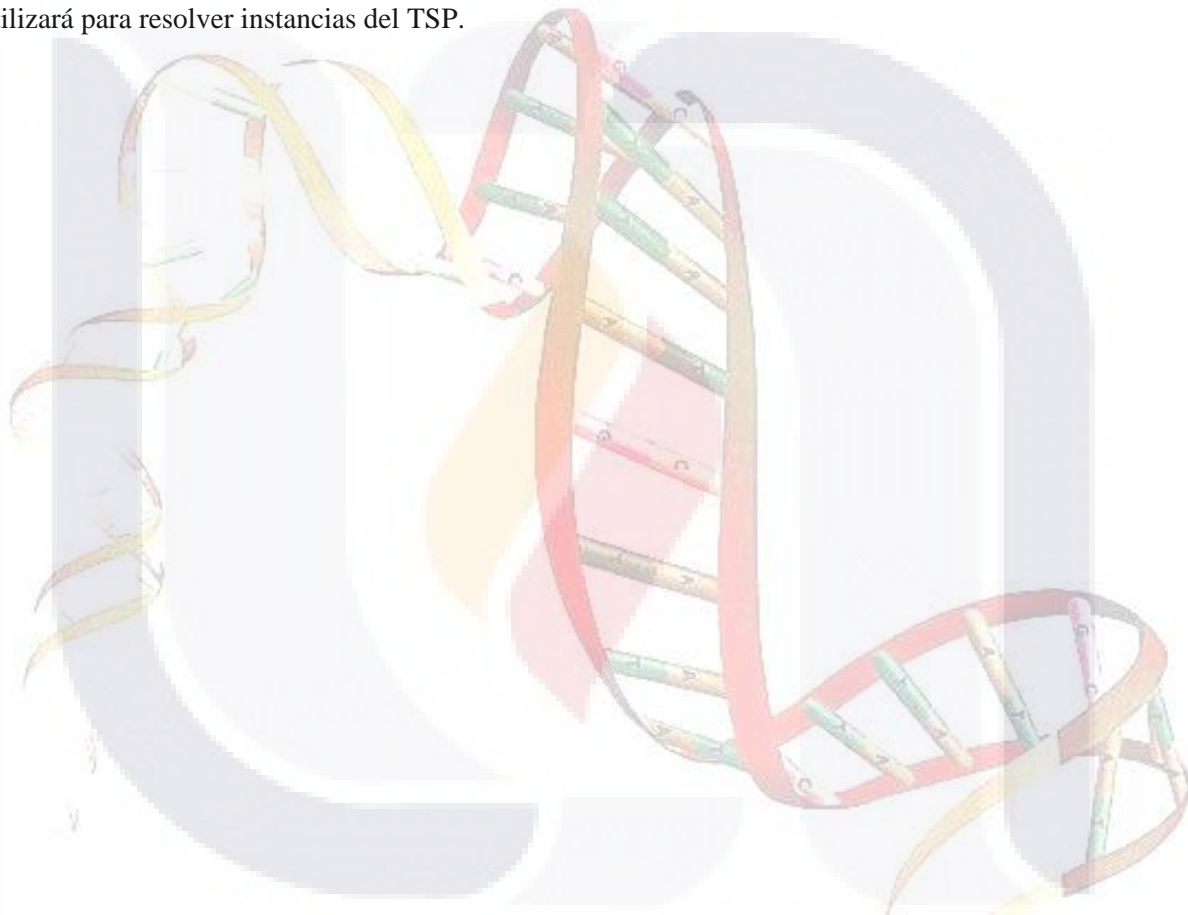
Para lograr obtener el modelo propuesto se debe hacer una identificación de los operadores que participan en la naturaleza para que se genere la evolución de las distintas especies y la forma en que interactúan entre sí, ya que este proceso ha demostrado ser bastante bueno (el mejor ejemplo es el alto grado de adaptación de plantas y animales a sus diferentes ecosistemas) y es la base sobre la que se inspiran los AG's.

Algunos de los operadores que participan en la Genética de Poblaciones referentes a la evolución no se han aplicado a los AG o por alguna razón no han dado los resultados esperados. Por esta razón, el

objetivo de esta propuesta es identificar cuales operadores se encargan de la diversificación y tratar de adaptarlos al AG simple.

Una vez identificados todos los operadores, se generará un modelo y una herramienta de software que nos permita experimentar y calibrar los parámetros para cada operador y evaluar si esto mejora el funcionamiento del AG simple.

Si los resultados que se obtengan mejoran las características del AG simple esta herramienta se utilizará para resolver instancias del TSP.



## CAPÍTULO 3

### MARCO TEÓRICO

---

#### 3.1. BASE TEÓRICA

---

##### 3.1.1 COMPUTACIÓN EVOLUTIVA (CE)

---

La CE es el área de la Inteligencia Artificial que agrupa técnicas inspiradas en los trabajos realizados por Charles Darwin (Teoría de la Evolución de las Especies), August Weistman (el Seleccionismo) y Gregor Mendel (trabajos sobre Genética) que simulan el proceso evolutivo natural [Hoffmann1989][Hinterding1997][Michelawicz1994].

Desde los años 30 varios investigadores se dieron cuenta que el proceso evolutivo natural le proporcionaba a algunos individuos mecanismos que los volvían más aptos para sobrevivir en el proceso de selección natural. Ellos veían esto como un proceso de aprendizaje que trataron de simular por medio de algoritmos [Santana2006].

Existen muchos trabajos que se realizaron en esta área, algunos de los cuales sentaron las bases de los conceptos de población; la representación binaria de las soluciones; el concepto de recombinación o cruzamiento; la selección de los individuos más aptos; la mutación en si; las bases de la evolución por medio de algoritmos; las bases de los que actualmente conocemos como redes neuronales; la robótica evolutiva, etc., que se ha utilizado en gran medida para la solución de problemas de optimización con muy buenos resultados.

Algunos de los requerimientos básicos para poder representar la evolución en algoritmos computacionales son:

- Poder codificar de alguna forma todos los parámetros que intervienen en la resolución del problema.
- Tener herramientas (normalmente cruzamiento y/o mutación) que permitan *evolucionar* las posibles soluciones que se tengan previamente codificadas a soluciones de mejor calidad
- Tener alguna función que permita medir la calidad de las soluciones.
- Un mecanismo que permita aplicar el concepto de selección de la teoría de Darwin.

La CE se encuentra dividida en tres principales áreas que son:

- Programación Evolutiva
- Estrategias Evolutivas
- Algoritmos Genéticos

#### 3.1.1.1. PROGRAMACIÓN EVOLUTIVA (PE)

La PE fué propuesta en los 60's por Lawrence Fogel. "La programación evolutiva enfatiza los nexos de comportamiento entre padres e hijos, en vez de buscar emular operadores genéticos específicos" [Fogel1966].

La PE surge como un método para evolucionar autómatas de estado finito para la predicción de series temporales



La PE consiste en una población inicial aleatoria a la que se aplica un proceso de mutación y se determina que soluciones pasaran a la siguiente generación por un proceso de selección (que puede ser por torneo o elitismo).

En la PE se considera el estado inicial, el siguiente estado y por medio de selección se determina que soluciones participaran en la nueva generación.

#### 3.1.1.2. ESTRATEGIAS EVOLUTIVAS (EE)

---

Las EE se desarrollaron a mediados de los 60's en Alemania por el grupo de Ingo Rechenberg para resolver problemas hidráulicos de alto grado de complejidad [Bäck1996].

En la propuesta original de la EE, se tenía un padre del que se generaba un hijo por medio de mutación y si el hijo tenía mejores características que el padre se conservaba, de lo contrario se eliminaba, a este proceso se le llamo selección extintiva.

En trabajos posteriores, se introdujo el concepto de población donde se tenían varios padres de los que se generaba un hijo que podía sustituir al peor padre de la población (selección extintiva) [Santana2006].

Schwefel introdujo el concepto de selección de hijos múltiples donde existían dos modelos, en el primero, los  $n$  mejores individuos obtenidos de la unión de padres e hijos se conservan, mientras que en el segundo solo los  $n$  mejores hijos de la siguiente generación sobreviven [Schwefel1981].

Actualmente se utiliza también el concepto de recombinación en EE que puede ser: sexual o panmíctica.

#### 3.1.1.3 ALGORITMOS GENÉTICOS AG'S

---

Los AG's fueron desarrollados en los 60's por John Holland en su interés por resolver problemas de aprendizaje de máquinas [Holland1975]. Están inspirados en la Teoría de la Evolución de las Especies de Darwin y los trabajos realizados en Genética por Gregor Mendel.

La forma en que trabaja el AG simple es generando una población inicial de soluciones (normalmente aleatoria), después se evalúa la aptitud da cada solución, se selecciona de manera probabilística a los individuos en base a su aptitud y se aplican sobre los individuos seleccionados operadores de cruce y mutación para generar la siguiente población. Este proceso se repite cíclicamente hasta llegar a un criterio de parada especificado.

El funcionamiento del AG se estudiará a fondo en la siguiente sección.

---

### ***3.1.2. ALGORITMOS GENÉTICOS***

---

Los AG son métodos adaptativos que se utilizan para resolver problemas de optimización y búsqueda. Reciben este nombre ya que están basados en los mecanismos de evolución descrita en las teorías de Darwin y la genética de Mendel [Holland1975] [Goldberg1989].

Al igual que en la naturaleza, las poblaciones de individuos evolucionan de una generación a otra por medio de la selección natural (supervivencia de los más aptos), el cruzamiento y la mutación, los AG's son capaces de ir creando soluciones para problemas del mundo real, simulando este proceso. La evolución de las soluciones hacia valores óptimos del problema está condicionada con una buena codificación del problema y la elección adecuada de los operadores que participaran en el proceso evolutivo.

Dentro del proceso natural los individuos compiten por los recursos que el medio provee y por participar en el proceso de reproducción. Normalmente los individuos que tienen ventajas sobre el resto de su población, son los que sobreviven y tienen mayores probabilidades de generar un gran número de descendientes. De forma contraria los individuos poco dotados generalmente son eliminados y su material genético normalmente desaparece.

En ocasiones, puede existir la combinación de buenas características provenientes de varios ancestros lo que generalmente produce individuos que son llamados superdotados ya que tienen una superioridad sobre el resto de la población y es cuando se efectúan los saltos evolutivos.

Los AG's al ser una herramienta inspirada en el proceso evolutivo trabajan con una población inicial de individuos, cada uno de los cuales representa una posible solución al problema planteado. Estas soluciones son evaluadas por medio de una función que indica el grado de *adaptació*; entre mayor sea el grado de adaptación es más fácil ser seleccionado para participar en el proceso de reproducción donde se combinarán las características de individuos bien adaptados lo que producirá individuos que compartirán características de ambos predecesores donde se espera un mayor grado de adaptación.

De esta forma, se cuenta con una nueva población que contiene las mejores características de la población anterior lo que genera que la búsqueda de soluciones se centre en las aéreas más prometedoras del espacio de búsqueda. Si el AG fue diseñado de forma adecuada se espera que este converja después de varias generaciones a una solución óptima.

**Inicio**

*Generar* población inicial

*Evaluar* el grado de adaptación (fitness)

**Mientras no** terminar **hacer**

**Para**  $i=1$  **hasta** (tamaño de población / 2) **hacer**

*Seleccionar* dos individuos de la generación anterior para cruzarlos (se seleccionan proporcionalmente a su grado de adaptación)

*Cruzar* con cierta probabilidad los dos individuos para generar dos descendientes

*Mutar* los dos descendientes con cierta probabilidad

*Evaluar* el grado de adaptación de los descendientes

*Insertar* los descendientes en la nueva población

**Si** la población converge **entonces**

terminar = verdadero

**Fin mientras**

**Fin**

Pcdo1. Pseudocódigo Algoritmo Genético Simple

Los AG se utilizan principalmente para resolver problemas que no pueden ser resueltos por técnicas de búsqueda exhaustivas o por técnicas especializadas, por ejemplo en problemas de búsqueda y optimización donde el espacio de búsqueda crece exponencialmente (explosión combinatoria).

A continuación se describen las principales características del AG simple, así como los operadores que utiliza en el proceso de evolución.

### ***3.1.2.1 CODIFICACIÓN Y FUNCIÓN DE EVALUACIÓN***

---

Dentro del AG se necesita una forma de representar toda la información que interviene en el proceso de solución del problema a resolver.

La codificación de estos parámetros representa posibles soluciones que forman parte del espacio de búsqueda del problema, que serán evaluadas para determinar el grado de adaptación de las mismas (que tan bien resuelven el problema).



La representación de cada cromosoma o individuo (que es el nombre que normalmente se utiliza en AG), puede hacerse por medio de elementos binarios (representación más sencilla), numéricos o alfanuméricos. La elección del tipo de elementos a utilizar normalmente está dada por la complejidad y características del problema a representar.



Fig. 3 Muestran distintas codificaciones de individuos en el AG

En la primera codificación se representa un individuo binario que se utilizará para resolver el problema del OneMax (donde la función objetivo es encontrar el máximo número de unos). La segunda codificación es numérica y se utiliza para representar las distintas ciudades que participan y su posición para resolver el TSP. Por último tenemos una codificación alfabética para representar combinaciones de aminoácidos así como la posición que cada uno ocupa dentro de la cadena.

Así como en la naturaleza los individuos más aptos son los que sobreviven, de la misma forma en el AG cada individuo cuenta con características que pueden ser evaluadas y definir su permanencia dentro de la población y generar descendencia.

La función objetivo es aquella que utilizamos para determinar el grado de “adaptación” de cada individuo. En esta se evalúa cada una de las partes que componen a un individuo (cada parámetro utilizado para llegar a la solución del problema), así como en ocasiones la posición en la que se encuentra cada elemento, su tipo, tamaño, etc. Al resultado de esta evaluación se le llama *fitness*.



Una vez evaluado cada individuo podemos determinar que tan bueno es de acuerdo al problema a solucionar. Por esta razón, una buena función objetivo, además de una adecuada codificación permitirán que el AG converja a las soluciones óptimas. Por el contrario una mala definición de la función o una mala codificación tendrán como consecuencia que el AG converja prematuramente a óptimos locales de mala calidad ya que estos dos elementos son los que dirigen la búsqueda dentro del espacio de soluciones.

El AG requiere de una población de individuos que es creada (normalmente de forma aleatoria) al inicio. Esta población es la que va evolucionando de generación en generación por medio de los operadores de selección, cruce y mutación a soluciones normalmente de mejor calidad.

El tamaño de la población se regula dependiendo del tipo de problema a resolver y el número de variables involucradas en la resolución del mismo. En esta población, en cada generación se conserva a aquellos individuos que mejor resuelven el problema y que serán los progenitores de los nuevos individuos que formaran parte de la misma.

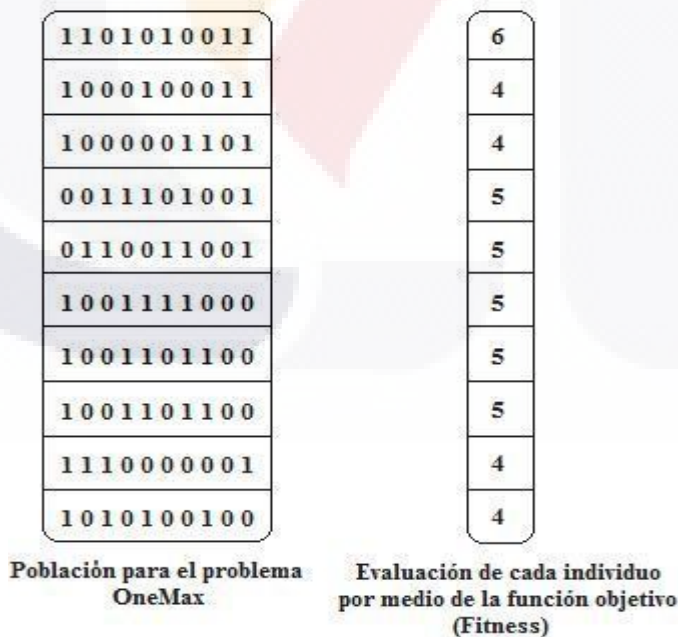


Fig. 4 Muestra de una población cuyos individuos son binarios y del resultado de la evaluación realizada a cada uno con base a la función objetivo.

SELECCIÓN.

Una vez obtenido el fitness para cada individuo de la población, existen varias formas de determinar que individuos formaran parte de la siguiente generación y podrían ser progenitores de los nuevos individuos. El AG simple utiliza el *elitismo* como método de selección, pero existen otros tipos como son el método de la ruleta o el muestreo universal estocástico [Moujahid2007].

El *elitismo* consiste en seleccionar aquellos individuos que mejor evaluación tuvieron basado en la función objetivo. Si se tratara de maximizar el resultado de una función se elegirían aquellos individuos que obtuvieron un fitness mayor, en caso de tratarse de una minimización de la función se elegirían a los de menor valor de fitness.

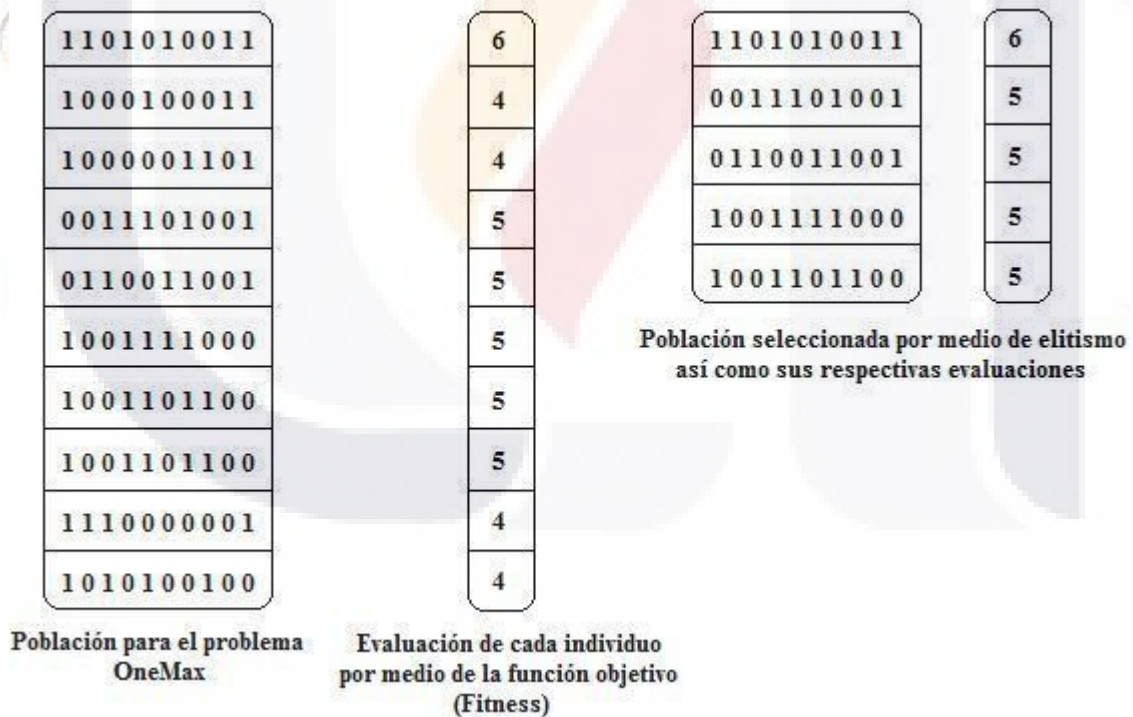


Fig.5 Selección de un 50% de los individuos por medio de elitismo

El método de la ruleta consiste en generar una ruleta imaginaria en la que cada individuo tiene asignada una fracción de la misma que es proporcional a su valor de aptitud. Los valores de esta ruleta van de cero a cien y para cada individuo se tiene un valor de inicio y uno de fin de su área correspondiente dentro de la ruleta. Se toman valores (aleatoriamente), y se selecciona al individuo en cuya área se tenga el valor generado [Santana 2006].

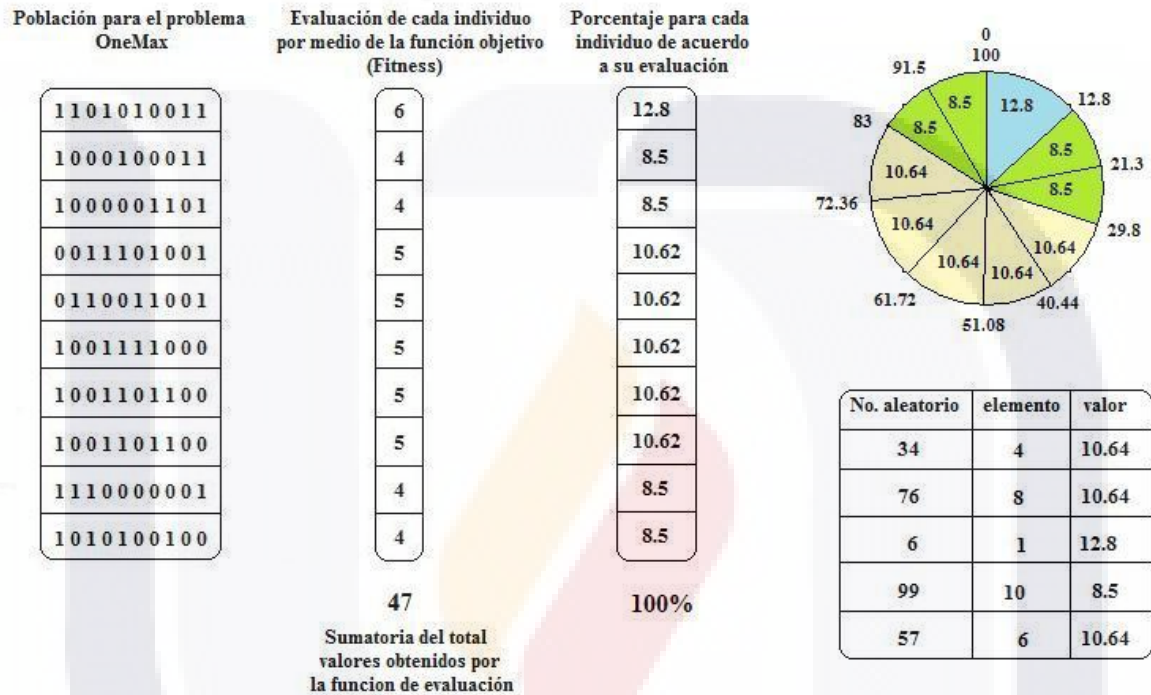


Fig. 6 Selección del 50% de los individuos por ruleta

El proceso que se sigue en el método de selección universal estocástico es similar al de la ruleta. En este también se genera la ruleta imaginaria basada en la evaluación obtenida por cada uno de los individuos de la población, solo que en este, se genera solamente un número al azar que será el primer elemento seleccionado y el resto se obtiene dividiendo la ruleta entre el número de individuos a generar partiendo del elemento previamente seleccionado.

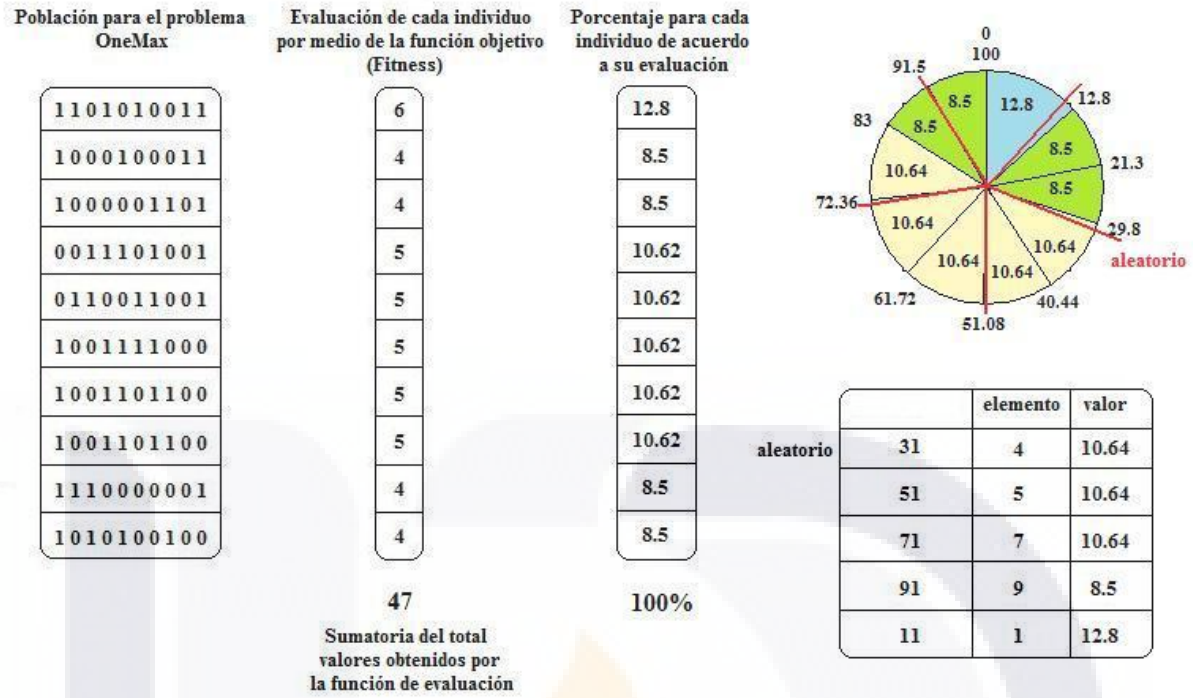


Fig. 7 Selección del 50% de los individuos por muestreo universal estocástico

El número de individuos a seleccionar es un parámetro que normalmente hay que calibrar, pero en el AG simple se selecciona un cincuenta por ciento de individuos para que formen parte de la nueva población [Holland1975].

**CRUZA.**

En la naturaleza no solo importa que un individuo sobreviva al proceso de selección natural gracias a su aptitud, sino que participe en el proceso reproductivo para que su carga genética se preserve en generaciones futuras por medio de sus descendientes.

Dentro del AG solo aquellos individuos que han sido seleccionados podrán cruzarse y generar descendientes que contengan parte del material genético de sus padres lo que dará como resultado una nueva solución al problema.



Dependiendo del tipo de problema y su representación, es el tipo de cruce que se deberá utilizar. Por ejemplo para un problema que tiene una codificación binaria o que no importa que exista repetición de elemento dentro de la codificación se puede utilizar un tipo de cruce de un punto (tipo de cruce más simple), mientras que para un problema más complejo donde la codificación no es binaria y no debe haber repetición dentro de la codificación se debe utilizar un tipo de cruce más complejo como es el GSX.

El tipo de cruce de un punto consiste en dos individuos (previamente seleccionados) dentro de los cuales se determina un mismo punto de cruce de forma aleatoria. Después se procede a tomar del primer padre desde el inicio del mismo hasta el punto de cruce y se adiciona la parte del segundo padre que comprende desde el punto de cruce hasta el final del mismo. Este proceso permite generar dos descendientes ya que si se invierte el uso de los padres el resultado será normalmente un individuo diferente al ya generado.

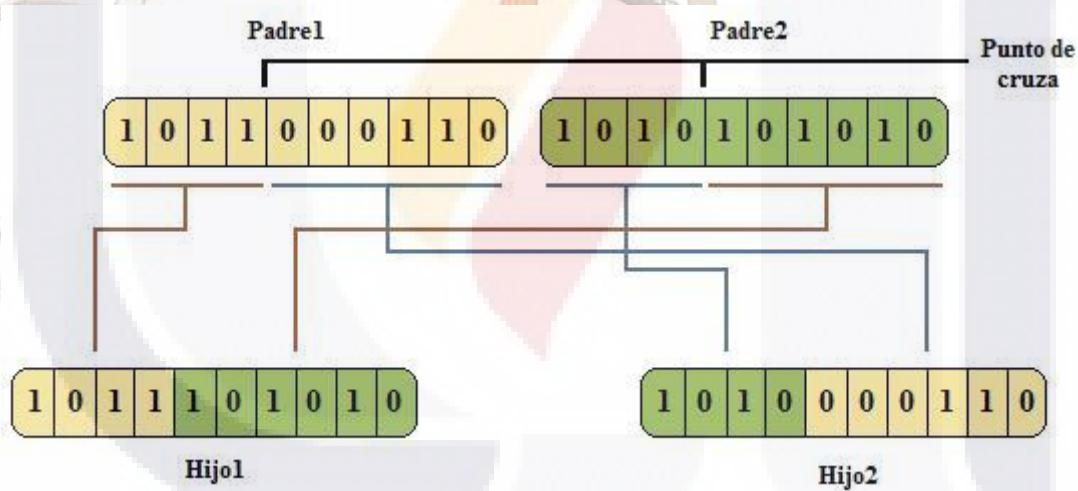


Fig.8 Ejemplo de la cruce de un punto

Existe también el proceso de puntos de cruce múltiple donde se definen varios puntos dentro de ambos padres y se van tomando las partes de cada padre alternadamente. Este tipo de cruce también permite generar dos descendientes.



El Greedy Subtour Crossover (GSX por sus siglas en inglés), es un tipo de cruza que por sus características se puede aplicar con muy buenos resultados a problemas como el TSP, ya que por la forma en que trabaja permite conservar subtours completos, sin repetir elementos a diferencia de otras herramientas (como las mencionadas anteriormente) que al hacer el proceso de cruza no discriminan la existencia de duplicidad en los elementos que conforman al nuevo individuo [Sengoku1998].

La forma en que trabaja el GSX la siguiente

- Se seleccionan dos padres
- Se elige un elemento al azar cuya posición es buscada en ambos padres
- Este elemento es el primero en formar parte del nuevo individuo
- Se hace un corrimiento de posición hacia la izquierda en el primer padre y hacia la derecha del segundo.
- Se verifica si estos elementos no forman parte ya del nuevo individuo. En caso de que no pertenezcan son introducidos a la derecha e izquierda respectivamente de la cadena ya generada.
- Si alguno de los elementos ya forma parte del nuevo individuo se detiene el proceso sobre ese padre.
- Por último, los elementos faltantes se introducen aleatoriamente al final de la cadena.

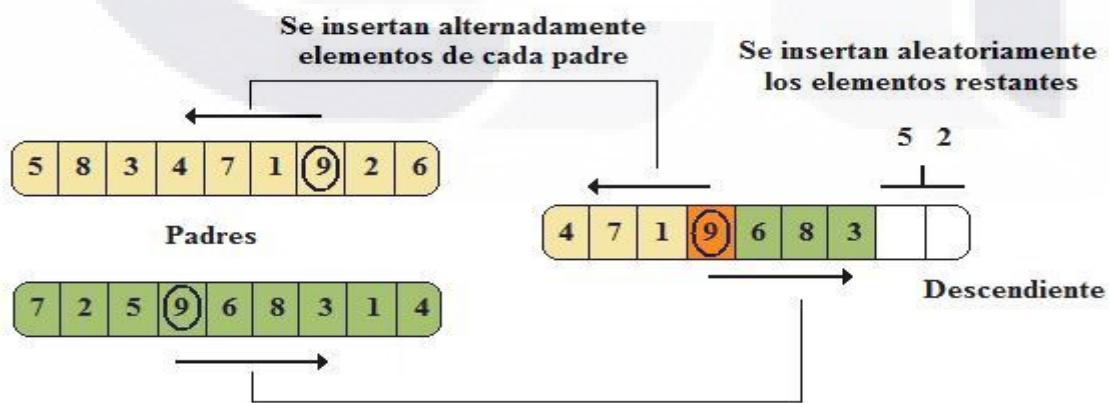


Fig. 9 Greedy Subtour Crossover

```

function GSX(pad1, pad2)
    band1 = true
    band2 = true
    elegir la ciudad  $t$  aleatoriamente
    elegir  $x$ , donde  $pad1_x = t$ 
    elegir  $y$ , donde  $pad2_y = t$ 
    desc  $\leftarrow t$ 
    repite
         $x = x - 1$ 
         $y = y + 1$ 
        si  $band1 = true$  entonces
            si not( $pad1_x \in desc$ ) entonces
                 $desc \leftarrow pad1_x + desc$ 
            else
                 $band1 = false$ 
        si  $band2 = true$  entonces
            si not( $pad2_y \in desc$ ) entonces
                 $desc \leftarrow desc + pad2_y$ 
            else
                 $band2 = false$ 
    mientras  $band1 = true$  or  $band2 = true$ 
    si  $|desc| < |pad1|$  entonces
        adicionar el resto de las ciudades en  $desc$  de forma
        aleatoria
    regresar  $desc$ 
    
```

Pcdo2. Pseudocódigo del Greedy Subtour Crossover

## MUTACIÓN.

En muy pocas ocasiones la naturaleza se equivoca y cuando esto sucede el material genético de algún individuo es modificado produciendo un cambio que puede traerle ventajas o desventajas sobre los demás individuos de su especie.

En los AG la mutación es el principal factor que aporta diversidad a las poblaciones, lo que permite hacer una mayor exploración del espacio de búsqueda, ya que lleva a los individuos mutados, normalmente a saltar de nichos en los que se encuentran la mayor parte de los

individuos realizando una explotación del mismo a zonas inexploradas donde en ocasiones se encuentran resultados de mejor calidad.

En individuos binarios, la mutación normalmente consiste en el cambio de un cero por un uno y viceversa, pero en individuos que tienen algún otro tipo de codificación y en los cuales ningún elemento puede repetirse el proceso es más complejo y por lo regular se utilizan herramientas como puede ser el 2opt.

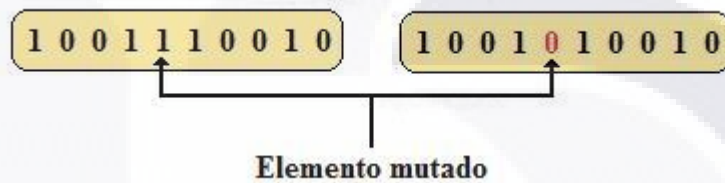


Fig. 10 Mutación de un individuo binario. Se intercambia un uno por un cero.

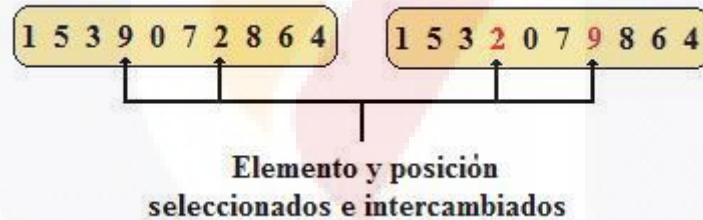


Fig. 11 Mutación en un individuo numérico que representa un tour del TSP donde no se permite repetición de elementos y se utiliza 2opt

En el trabajo realizado por Goldberg, este recomienda utilizar la mutación en porcentajes muy pequeños, debido a que al ser una herramienta basada en el azar su efecto puede ser destructivo y afectar buenas soluciones con las que se cuenta [Goldberg1989].

Estos son los elementos que utilizan el AG simple, pero existen muchas otras variantes, además de algoritmos híbridos que utilizan las mejores cualidades de AG para resolver problemas de optimización.

---

### 3.1.3. GENÉTICA DE POBLACIONES

---

La genética de poblaciones es el área de la genética que se encarga del estudio de la frecuencia de los alelos y los cambios producidos por los cuatro principales factores en el proceso evolutivo: la selección natural, migración, mutación y deriva genética [Barbadilla2008][Wikipedia2]

- *Mutación.* Son variantes en el material genético hereditario, la mayoría de las cuales son eliminadas ya que producen en los individuos mutados desventajas competitivas, pero en ocasiones estos cambios permiten el éxito del individuo y la modificación en el material genético se incorpora al heredado por sus descendientes.

La mutación es considerada como la base de la evolución ya que los cambios que esta produce son los que permiten un aumento en la diversidad genética y permiten la adaptación de los individuos a los cambios que se presentan en el transcurso de su existencia.

Una alta tasa de mutación permitiría una más fácil adaptación del individuo en caso de por ejemplo un cambio ambiental, pero a su vez también provocaría una mayor cantidad de cambios perjudiciales que podría llegar hasta a desencadenar una extinción de la especie.

- *Migración.* Es el intercambio genético entre poblaciones. Es considerado un factor muy importante para la diversidad genética. Gracias a la migración existe un cambio en los alelos lo que produce cambios significativos en las características de los

descendientes de individuos de la población original cruzados con los individuos migrados.

- *Selección natural.* “Es el proceso por el cual una especie se adapta a su medio ambiente” [Futuyma2004].

La selección natural es la responsable de los cambios evolutivos, ya que regula la tasa de reproducción, permitiendo a los individuos más adaptados o con mejores características heredar a sus descendientes dichas características.

La selección natural no solamente consiste en la supervivencia del individuo más apto o adaptado, sino que estos individuos puedan heredar características particulares a sus descendientes.

En la mayoría de los casos no existe un solo individuo que sea superior a los demás en todos aspectos, sino que hay varios individuos cada uno de los cuales puede tener una o varias características que le dan ventajas adaptativas.

- *Deriva genética.* Es un proceso que permite a las poblaciones cambiar la frecuencia genética de forma aleatoria, es decir, es un proceso en el cual si no existe acción de la mutación o la migración la población tiende a perder su variabilidad genética. Ejemplo de ello es la pérdida de características de una generación a otra, donde la codificación de la parte del gen que permitía la existencia de dichas características, difícilmente volverá a encontrarse dentro de la población.

---

#### **3.1.4. PROBLEMA DEL VIAJANTE DE COMERCIO**

---

El Problema del viajante de comercio o TSP (por sus siglas en inglés) es uno de los más antiguos, famoso y representativo de la optimización combinatoria, ya que es muy sencillo de



plantear, pero detrás de esa sencillez se encuentra un problema altamente difícil de resolver debido a que conforme crece el número de variables (ciudades) que involucra el problema, el espacio de búsqueda de soluciones se incrementa exponencialmente [Holstein1998][Moscatto1994].

El TSP consiste en un viajante que debe recorrer  $n$  ciudades, visitarlas todas pasando solamente una vez por cada una de ellas y por último regresar a la ciudad en la que inicio el recorrido. Se conoce la distancia entre cada par de ciudades conectadas y el objetivo es encontrar el recorrido más corto o barato. El costo del recorrido se obtiene sumando las distancias entre ciudades que son parte de dicho recorrido.

La definición matemática del problema es:

Dadas  $n$  ciudades  $\{c_1, c_2, \dots, c_n\}$  y una matriz de distancias entre ellas  $D:n*n$ ;  $D_{ij} \in \mathbb{R}$  (donde  $D_{ij}$  indica la distancia entre las ciudades  $c_i$  y  $c_j$ ), la tarea es encontrar un “tour”, es decir una permutación de las  $n$  ciudades, que minimice la longitud total  $L_{(Tour)}$ , definida como la suma de las distancias:

$$L_{(Tour)} = \sum_{i=0}^{n-1} D_{Tour(i), Tour(i+1)}$$

Fórmula 1. Función objetivo para resolver el TSP

donde  $Tour(n)$  se identifica como  $Tour(0)$  para conseguir un ciclo cerrado [Holstein1998].

El TSP es una herramienta para test de algoritmos heurísticos. Gracias a su naturaleza NP-Completo hace posible que se pueda resolver una gran familia de problemas equivalentes mediante transformaciones particulares para distintos casos [Moscatto1996].

3.2. TRABAJOS RELACIONADOS

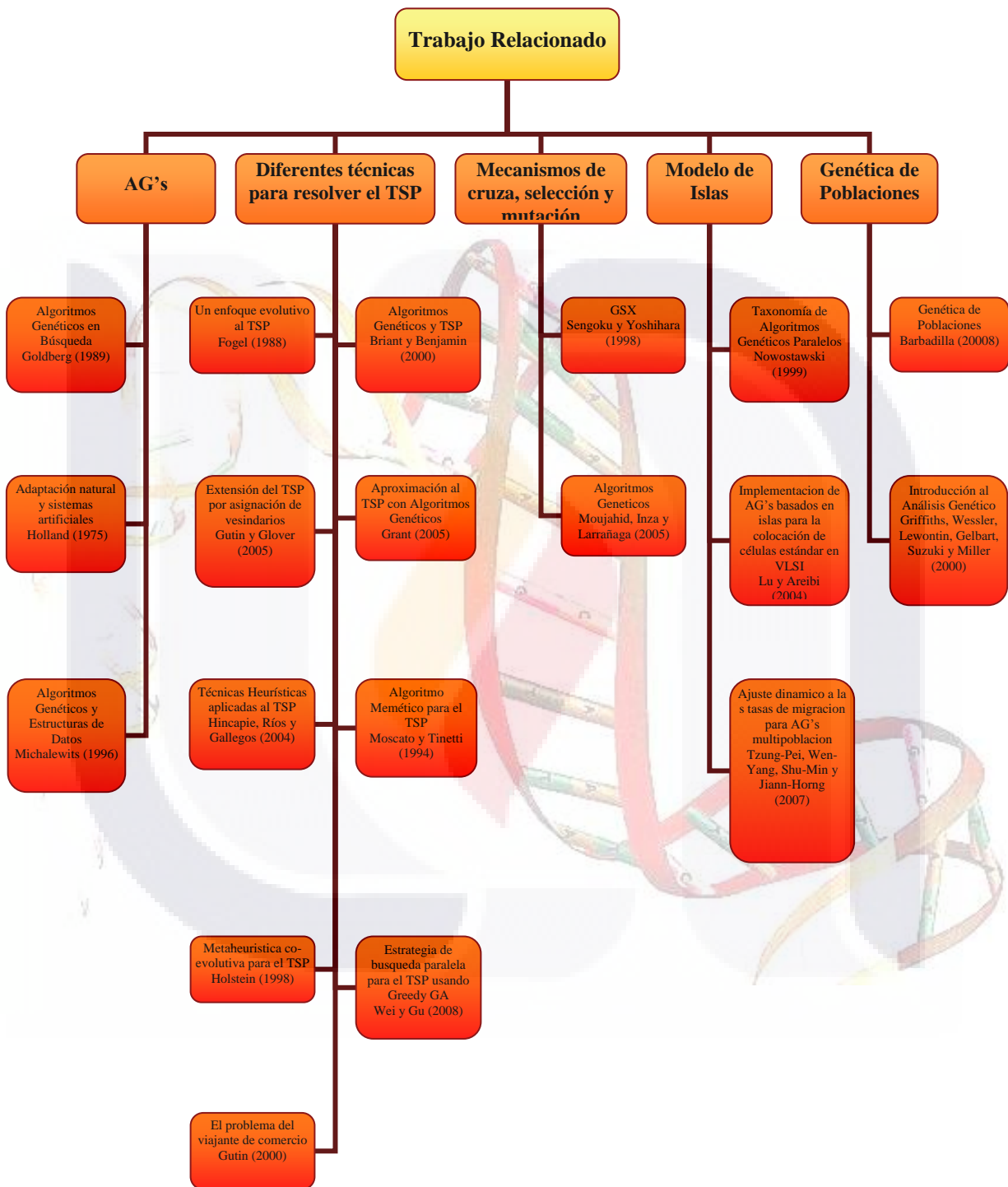


Fig. 12 Diagrama de trabajo relacionado

Primeramente tenemos los trabajos realizados por Holland y Goldberg [Holland1975][Goldberg1986] que sientan las bases de los AG's. Muestran la forma de codificar soluciones a problemas de optimización, la forma en que actúa la selección natural, la cruce y la mutación así como la forma en que se lleva a posibles soluciones a evolucionar a soluciones de muy buena calidad.

En el trabajo de Michalewicz "Genetic Algorithms + Data Structures = Evolution Programs" [Michalewicz1996] se exponen varias formas de aplicar los distintos operadores, como generar las poblaciones, las características que debe tener una función objetivo, tamaño de la población, entre otros, además de algunos problemas en los que se ha utilizado a los AG's para solucionarlos.

Se estudiaron también algunos trabajos que muestran distintas herramientas basadas en AG's, así como otras heurísticas que se han utilizado para solucionar el TSP, dentro de las cuales encontramos:

Tenemos por ejemplo el trabajo de Fogel "An Evolutionary Approach to the Traveling Salesman Problem" [Fogel1988], donde nos muestra algunos de los conceptos propuestos por Holland y enfatiza algunas características tanto del cruzamiento como de la mutación que son dos operadores que sirven para mantener la diversidad de la población y generar nuevos individuos basándose en el material genético seleccionado de una determinada población.

El trabajo de Bryant & Benjamin "Genetic Algorithms and the Traveling Salesman Problem" [Bryant2000], de Grant "Approximating the Traveling Salesman Problem with Genetic Algorithms" [Grant2005] y de Gutin "The Travelling Salesman Problem" [Gutin2000] proporcionan información sobre la base teórica del TSP, sus características, objetivo, entre otros. Además muestran información sobre la forma en que trabajan los AG's, los operadores que participan, algunos parámetros utilizados y algunos resultados obtenidos al resolver este problema.

En su trabajo “Further Extension of the TSP Assign Neighborhood” Gutin y Glover [Gutin2005], proponen el resolver el TSP dividiendo el mismo en vecindarios que serian aquellas ciudades con las que se tiene conexion mas cercana e ir construyendo la solución basándose en dicha conexion.

En el trabajo de Hincapié, Ríos y Gallegos “Técnicas Heurísticas Aplicadas al Problema del Cartero Viajante (TSP)” [Hincapié2004], se muestra la forma en que funcionan algunas herramientas heurísticas como los AG’s, el Recocido Simulado (SA), la Optimizacion por Colonia de Hormigas (ACO) y la Búsqueda Tabu (TS), su base teorica y la forma en que resuelven el TSP.

Moscato y Tinetti en su trabajo “Blending Heuristics with a Population-Based Approach: A ‘Memetic’ Algorithm for the Traveling Salesman Problem” [Moscato1994] manejan las bases del TSP y de los Algoritmos Memeticos (MA) y un concepto interesante que es la convergencia prematura al utilizar poblaciones pequeñas y se hace un estudio sobre los tamaños de poblacion que se debería utilizar al aplicar esta heuristica.

Holstein en su trabajo de grado utiliza el calculo de la complejidad computacional, así como muestra conceptos del TSP, utilize los AG y los MA para solucionar dicho problema [Holstein1998].

Por ultimo en el trabajo de Wei y Gu “Parallel Search Strategies for TSPs using a Greedy Genetic Algorithm” [Wei2008] utilizan la generacion de un banco genetico basado en bloques constructores y sobre la información de este banco se genera la población inicial y se realiza la mutacion. Tambien aplican el concepto de inmigración el cual basan en la introducción de individuos nuevos generados de forma similar a los creados en la población inicial.

En cuanto a trabajos específicos de los operadores que actúan en el AG tenemos el trabajo de Sengoku y Yoshihara “A Fast TSP Solver Using GA on JAVA” nos muestra una aportación hecha sobre el operador de cruza que es especifico para el TSP ya que permite el conservar subtours completos, lo que para este problema es muy importante.

En el trabajo de Moujahid, Inza y Larrañaga “Algoritmos Geneticos” [Moujahid2005], se muestra un estudio donde se muestran lo que es la codificación de las posibles soluciones, la creación de la función objetivo así como varias herramientas que han sido utilizadas durante varios años para realizar el trabajo de los distintos operadores. Por ejemplo tenemos que para el proceso de selección se cuenta con herramientas como lo son elitismo, ruleta, torneo, muestreo universal estocástico, entre otros.

Para el proceso de cruce se maneja el más común que es la cruce de un punto, multipunto y el operador de cruce uniforme.

En cuanto al operador de mutación se maneja que aun cuando en la mayoría de los trabajos se maneja un porcentaje de mutación y cruce fijo, existen otros trabajos interesantes que manejan probabilidades de mutación variable [Moujahid2005].

Se maneja el proceso de reducción que no es muy común en AG's pero si en otras heurísticas pertenecientes a computo evolutivo y que consiste en eliminar a los peores individuos de la población de la generación anterior y los individuos generados por cruce.

Un proceso muy importante del trabajo de Moujahid, Inza y Larrañaga para el presente es la utilización de AG's paralelos en los que se utiliza el modelo de islas y muestra varias representaciones.

También se muestran los criterios de parada más utilizados, además de que se muestran algunas de las representaciones por medio de las cuales se puede resolver el TSP por un AG.

Existen múltiples trabajos relacionados con los AG's paralelos, modelo de islas y migración, entre los que se encuentran:

Primero tenemos el trabajo de Nowostawski y Poli “Parallel Genetic Algorithm Taxonomy” [Nowolsky1999], en el que se comenta que los AG's secuenciales han dado muy buenos



resultados para algunos problemas en particular pero hay otros en los que el poder utilizar paralelismo permite encontrar soluciones mas rápidamente o soluciones de mejor calidad.

En este trabajo se manejan distintos tipos de paralelismo de AG's dentro de los que se encuentran: la paralelizacion maestro-esclavo síncrona y asíncrona, subpoblaciones estaticas con migración, solapamiento estatico de subpoblaciones, AG's con paralelismo masivo, multi Demes, métodos hibridos, entre otros. Tambien se muestran las principales características de cada uno y el enfoque que normalmente se les da.

El paralelismo en AG's normalmente pretende distribuir la carga computacional del mismo en multiples procesadores, ya sea por medio de distribución de poblaciones que se trabajen de forma paralela (normalmente llamadas subpoblaciones o demes) o distribuyendo algunas funciones especificas como puede ser la evaluación, el cruzamiento o la mutacion.

Se introduce también el concepto de migracion que consiste en el intercambio de individuos entre poblaciones ya sea por vecindarios o sobre toda las poblaciones. Los principales criterios a tonar en cuenta para la migración son: la topología que se utiliza, el criterio para seleccionar los migrantes, cuantos individuos se migraran y cada cuando se migrara.

Existen otros trabajos como el de Lu y Areibi "*An Island-Based GA Implementation for VLSI Standard-Cell Placement*" [Lu2004] donde se hace un estudio sobre el modelo de islas (específicamente del modelo de grano grueso), donde se pretende distribuir la carga computacional de los AG's en varios equipos (normalmente Workstations) para mejorar el tiempo de respuesta.

Hacen un estudio sobre el AG síncrono basado en islas, en el que se divide la población total en subpoblaciones (cuyo numero es igual al numero de equipos en los que se va a ejecutar el modelo), que evolucionan por medio de los operadores de selección, cruza y mutacion de manera independiente. El proceso de migración se efectua al llegar a un determinado criterio que pude ser una detección de cercania a la convergencia o al llegar a converger. Este proceso

se debe realizar al mismo tiempo en todas las subpoblaciones por lo que uno de sus principales problemas es que si una subpoblacion se dilata mucho en llegar al criterio determinado las demás se tienen que esperar hasta que esta termine por lo que se pierde una gran cantidad de tiempo de computador.

Tambien se maneja que la topología mas utilizada en AG's paralelos en la topología de anillo ya que normalmente permite llegar a mejores soluciones que en el AG serial, pero una de sus principales diferencias es que si el numero de individuos en una subpoblacion es muy pequeña (esto sucede cuando se tienen muchos equipos entre los cuales distribuir la población), normalmente los resultados peores que en AG serial.

Ellos proponen una topología basada en un grafo completo a la que llaman "to-all" en la que están interconectados todos los demes lo que les permite compartir información entre todos ellos. La migración no la realizan directamente de una población a las demás sino que primero se migran a un bloque llamado pool de migrantes y de este se eligen los individuos a introducir a cada deme. Los individuos que proporciona cada deme para el pool son sus mejores individuos, adicionalmente no permiten la migración de los mismos individuos en distintas generaciones.

Otro de los estudios realizados por Lu y Areibi es sobre los AG asíncronos en los que se maneja una población maestra que esta interconectada de forma dual con todas y cada una de las subpoblaciones. El principal beneficio que esto tiene es el que no hay perdida de tiempo computacional ya que cada subpoblacion migra a sus mejores individuos cuando llega al criterio determinado, desafortunadamente se complica el proceso de evaluación, de selección y el tiempo en que la población maestra debe recibir o enviar a los individuos a migrar.

Una propuesta para mejorar los AG's síncronos es el Balanceo Dimanico que consiste en determinar cuales son las poblaciones que mas se dilata y mover algunos de los individuos de estas a alguna o algunas de las poblaciones que tienen mejor desempeño.

Dentro de los resultados que reportan se maneja que al aumentar el numero de procesadores en los modelos síncronos (dividir la población en mayor numero de subpoblaciones) se pierde la velocidad ya que aun cuando el costo computacional esta mas distribuido se incrementa el costo por las comunicaciones.

Por otro lado los modelos asíncronos mostraron una mejor eficiencia pero su costo de implementación es mayor.

Tzung-Pei, Wen-Yang, Shu-Min y Jiann-Horng investigan en su trabajo *“Dynamically Adjusting Migration Rates for Multi-Population Genetic Algorithms”* la configuración de parámetros que mejores resultados permiten obtener. Ellos manejan que un ajuste adaptativo de los distintos parámetros permite mejorar los resultados obtenidos por los AG's multi-poblacion.

Su diseño adaptativo evalua en cada generación si los resultados que se obtendrían al migrar los individuos de una subpoblacion paralela, mejoran significativamente los resultados, el porcentaje de individuos a migrar se incrementa, en caso contrario se disminulle.

Sus experimentos los realizaron con 16 subpoblaciones de 30 individuos. La poblacion inicial se genera de manera aleatoria, utilizan una topologia de anillo (como se comento la mas utilizada), se toma un 65 porciento de cruza y una mutacion del 5 porciento y la seleccion se realice por torneo, se utilize un proceso de remplazo de los peores individuos por los migrados (mejores de cada subpoblacion) y se tiene un criterio de parade de 2000 generaciones.

Dentro de sus experimentos evaluan el modelos con migracion adaptativa contra un proceso de migracion fijado cada determinado numero de generaciones.

Dentro de los resultados se demostro que la migracion adaptativa permite obtener resultados de mejor calidad en un menor numero de generaciones.

## CAPÍTULO 4

### DESARROLLO Y VALIDACIÓN

---

#### 4.1 MODELO PROPUESTO Y TRABAJO REALIZADO.

---

##### 4.1.1. MIGRACIÓN O MUTACIÓN

---

Como parte de la generación del modelo se propuso originalmente, tomar como base el AG simple y sustituir el operador de mutación por el de migración (este operador está inspirado en uno de los cuatro factores que participan en la evolución según se muestra en genética de poblaciones), para mejorar el proceso de diversificación de las poblaciones.

Una segunda propuesta fue agregar el operador de migración para apoyar el trabajo realizado por la mutación en la diversificación de las poblaciones.

El proceso de migración biológica, sirve para introducir material genético de una población a otra, lo que genera una mayor variabilidad y permite evitar que debido a la endogamia se llegue a la extinción de la especie, ya que al no haber diversidad los individuos pierden la capacidad para adaptarse a los cambios existentes en el medio [Barbadilla2008].

El hombre ha utilizado la migración entre razas de animales para generar individuos con características deseables, por ejemplo, para la creación de las razas de perros que actualmente se tienen, se utilizó la migración. Cada raza fue obtenida cruzando animales que presentaban características deseables (carácter, fuerza, inteligencia, etc.) hasta obtener individuos



totalmente adaptados al uso que el hombre deseaba darles. Este mismo proceso se ha utilizado para mejorar razas de bovinos, equinos, porcinos, etc. [Magofke2003][Caraviello2004]

De manera general este proceso consiste en seleccionar individuos de razas distintas que contengan características deseables y realizar una cruce entre ellos. Estos deben tener en una alta medida la o las características deseadas, para que genéticamente sean transmitidas a sus descendientes.

En algunas ocasiones se realiza primero un proceso de purificación de las características deseables, esto se logra cruzando padres con hijos, lo que provoca que las características más arraigadas en estos individuos se vayan intensificando debido a que se va perdiendo la variabilidad genética. Desafortunadamente esto también provoca que los animales comiencen a tener problemas genéticos.

Dentro del AG se va produciendo un efecto similar ya que conforme se avanza en el número de generaciones, los individuos tienden a presentar cada vez mayor parecido entre sí y se converge a óptimos locales. Este efecto es producido ya que el factor de mutación que se utiliza para diversificar el material genético se utiliza en porcentajes muy pequeños pues al ser un proceso aleatorio no discrimina entre buenas y malas soluciones [Goldberg1998]. Por lo anterior se necesita un mecanismo de diversificación que permita su aplicación a mayor escala.

En el modelo inicial propuesto se pretendía estudiar si las características de la migración eran suficiente buenas para sustituir a la mutación o si era necesario conservar el proceso de mutación e introducir nuevos individuos por medio de migración (que se espera contengan características frescas) y antes de que participen en la creación de nuevos individuos, evaluarlos al igual que a las de los individuos conservados de la generación anterior y que compiten por su supervivencia.



El problema utilizado para probar el modelo propuesto es el TSP.

La forma en que trabaja el primer modelo propuesto es:

- Se genera una población inicial, de forma aleatoria.
- Se evalúa cada uno de los individuos de la población (posibles soluciones) por medio de la función de evaluación.
- Se seleccionan por medio de elitismo a los individuos cuyo costo del viaje sea más pequeño, hasta tener, en caso de que el número de individuos lo permita, un cincuenta por ciento, que formaran parte de la nueva población; en caso de que esto no sea posible (por el número de individuos) se tomarán el cincuenta por ciento menos un individuo.
- Se generan nuevos individuos utilizando el cruzamiento de dos individuos elegidos al azar (de los que fueron previamente seleccionados), hasta cubrir un cuarenta por ciento del tamaño total de la población inicial, esto en caso de que en el paso anterior se cubra el cincuenta por ciento, sino se generará por cruce un individuo adicional.
- Se generan aleatoriamente los individuos necesarios para completar el tamaño total de la población.
- Se hace un ciclo que regresa al segundo paso hasta que se llegue a cumplir un criterio de parada.

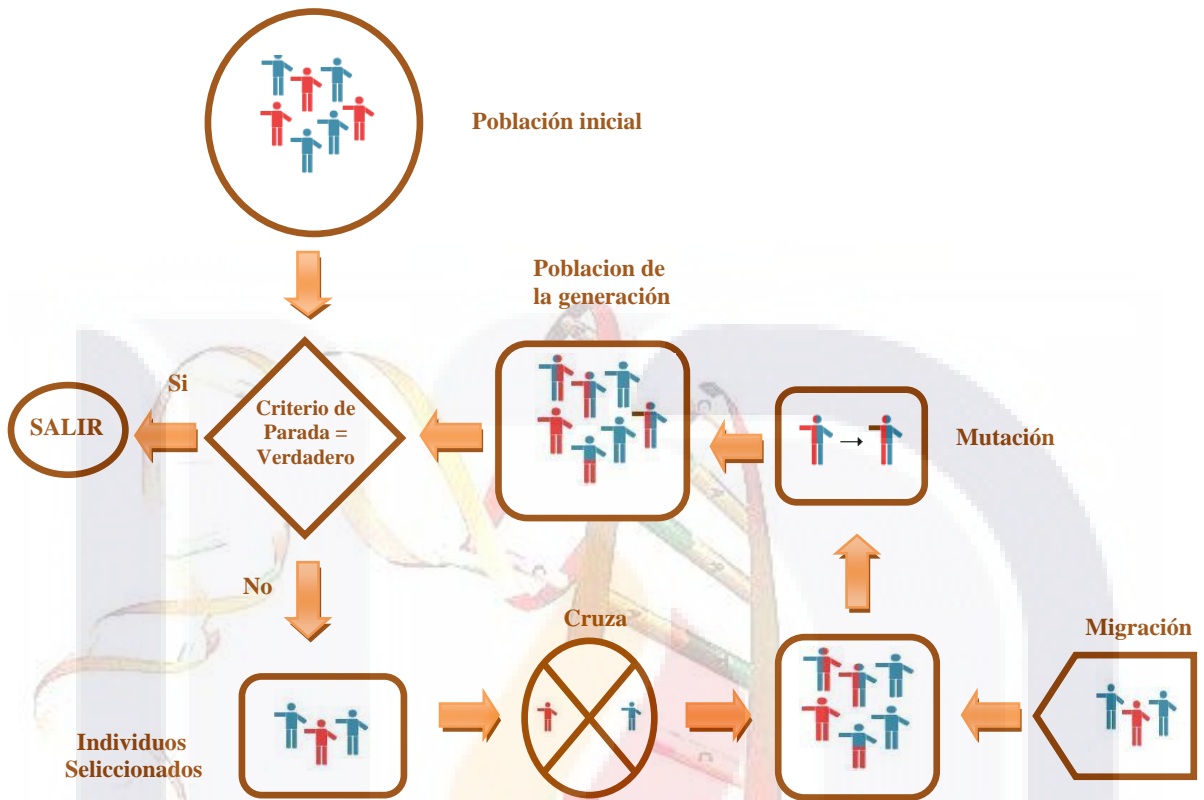


Fig. 13 Diagrama del modelo propuesto en su primera version

En la segunda vertiente se introduce el proceso de mutación después de completar el tamaño inicial de la población como se muestra en la figura 13.

Se desarrolló una herramienta de software para verificar el comportamiento del modelo propuesto en comparación con el AG simple.

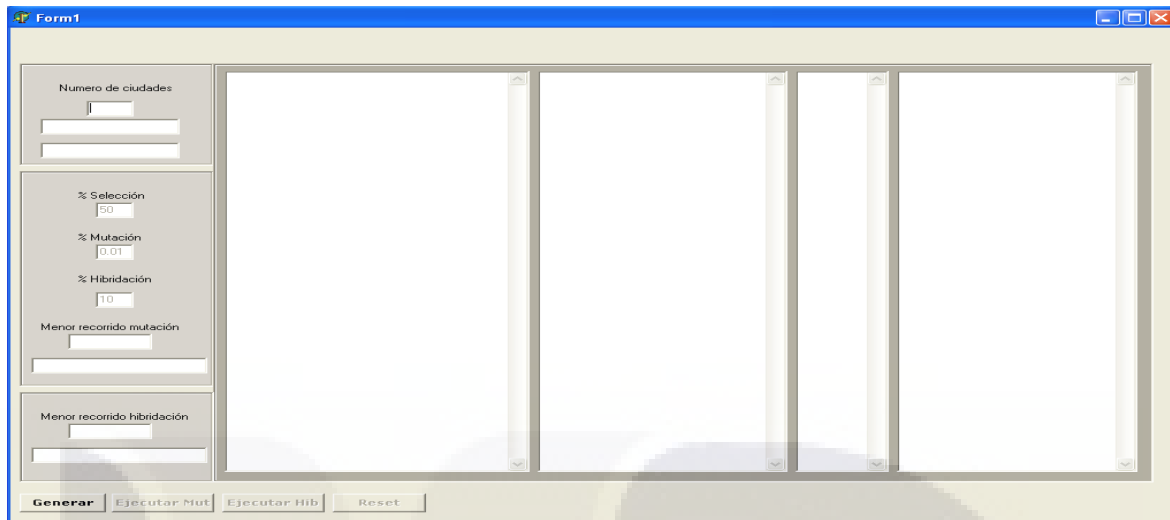


Fig.14 Herramienta de Software para resolver el TSP utilizando un AG simple y el modelo propuesto de AG utilizando migración como apoyo en el proceso de diversificación.

La forma de representar cada posible solución se realizó por medio de una cadena de valores, donde cada valor representa una ciudad. En esta cadena no existe repetición de ciudades. El número de ciudades es introducido por el usuario.

Para representar la población sobre la que trabajaba el AG se utilizaban matrices. A consecuencia de la utilización de estas estructuras, la herramienta funcionaba para un máximo de 30 ciudades ya que se saturaba la memoria.

La herramienta trabajaba sobre un grafo plano, completo, no dirigido, para el cual se generan también aleatoriamente las distancias entre ciudades y se guardan en una matriz de costos.

0	12	11	15	5
12	0	5	6	20
11	5	0	5	28
15	6	5	0	16
5	20	28	16	0

**Matriz de costos**

Fig. 15 Matriz de costos (distancias entre ciudades).

Primero se genera la población inicial con un tamaño igual al número de ciudades. De la misma forma el tamaño del cromosoma es igual al número de ciudades. Los individuos serán generados de forma aleatoria y sin repetición de ciudades.

2	5	3	1	4
1	4	2	3	5
2	3	1	4	5
5	3	4	2	1
4	1	5	2	3

**Población inicial**

Fig. 16 Población inicial aleatoriamente generada. Individuos sin repetición de componentes.

Para poder evaluar la calidad de los resultados obtenidos por el modelo propuesto se efectúa una copia de la población inicial y se ejecutan ambos algoritmos (AG simple y el modelo propuesto cada uno sobre una copia de la población inicial).

La función objetivo sobre la que se evalúan los individuos es la de minimizar el costo del viaje. El costo del viaje se obtiene sumando el valor de la matriz de costos para cada par de ciudades invocadas en el recorrido de acuerdo al orden en que aparecen en el individuo y agregando la distancia entre la última y primera ciudades de dicho recorrido.

2	5	3	1	4	80
1	4	2	3	5	59
2	3	1	4	5	67
5	3	4	2	1	56
4	1	5	2	3	50

**Población**                      **Fitness**

Fig. 17 Para cada individuo (tour) se calcula el fitness basado en los valores que se tienen en la matriz de costos para cada par de ciudades.

Después de la evaluación, se utiliza el método de selección elitista para elegir a los mejores individuos dentro de la población (menor costo de viaje), que formaran parte de la nueva generación y podrán participar en el proceso de cruza. El número de individuos que se conserva (si el tamaño de la población lo permite) es el cincuenta por ciento de la población inicial, de lo contrario, el cincuenta por ciento menos un individuo.

4	1	5	2	3
5	3	4	2	1

**Elementos seleccionados  
por elitismo**

Fig. 18 Individuos seleccionados por medio de elitismo.

Para el AG simple, se obtiene por medio de cruzamiento el cincuenta por ciento restante, en caso de que en el paso anterior se seleccione el cincuenta por ciento, en caso contrario se genera por cruzamiento el cincuenta por ciento más un individuo y para el nuevo modelo al igual se genera un cuarenta por ciento si se eligió el cincuenta por ciento, de lo contrario se genera el cuarenta por ciento más un individuo.

Para el cruzamiento se utiliza el algoritmo propuesto por Sengoku y Yoshihara llamado Greedy Subtour Crossover (GSX) [Sengoku1998].

El funcionamiento del GSX es el siguiente:



- Se seleccionan dos padres de forma aleatoria.
- Se busca la posición de un mismo elemento (que forme parte del recorrido) en ambos padres y se define como la posición inicial.
- El elemento elegido es el primer elemento del nuevo individuo.
- Se hace un recorrido en el primer padre hacia el inicio del mismo y en el segundo hacia el fin a partir de la posición inicial.
- Se verifica si cada uno de los elementos de la nueva posición en cada padre son válidos para formar parte del nuevo individuo. Un elemento es válido siempre que no se encuentre contenido ya en el nuevo individuo.
- El proceso de recorrido de los padres es cíclico y se detiene hasta que los padres no tengan hacia donde ser recorridos.
- Por último se adicionan al final los elementos que no estén todavía contenidos en el nuevo individuo.

Este tipo de cruce permite conservar subtours que estén contenidos en los padres seleccionados lo que para el TSP es un factor importante.

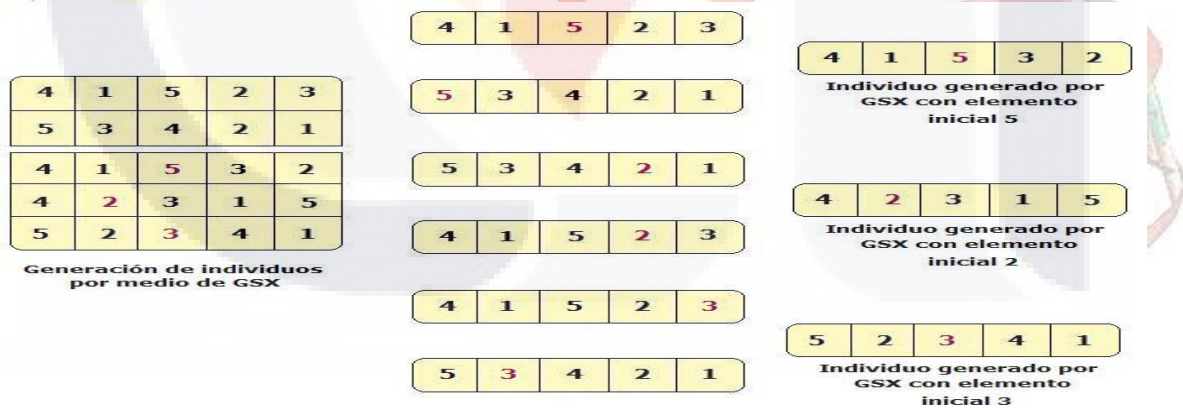


Fig.19 Descendientes obtenidos por medio de GSX para completar el tamaño de la población original.

Para el modelo propuesto como ya se comentó, solamente se obtiene un cuarenta por ciento (del cincuenta faltante) por medio de cruce y se completa la población generando nuevos

individuos de forma aleatoria (se utiliza el mismo proceso que para generar la población inicial). El número de individuos a generar es el diez por ciento de la población total, ya que en trabajos realizados en el campo de la genética se ha detectado que al tener una cantidad mayor al diez por ciento de mutación se provoca que el material genético pierda la esencia de la población principal [Blasco2008]

En caso de que por el número de individuos no se pueda generar exactamente el cuarenta por ciento de los individuos, se generará el cuarenta por ciento más un individuo y la migración se haría de un diez por ciento menos un individuo.

El último paso dentro del AG simple es la mutación. El método que utilizamos fue el 2opt. Este proceso se describe a continuación:

- Se selecciona un individuo.
- Se selecciona un elemento del individuo, así como una posición dentro del mismo.
- Se hace el intercambio entre el elemento seleccionado y el elemento que se encuentra en la posición también seleccionada.



Fig.20 Muestra el proceso de mutación en el que se elige el elemento con valor 4 y se intercambia a la posición

2.

Este proceso se realiza en un uno por ciento del número de elementos contenidos en la población, es decir, el número de individuos multiplicado por el número de ciudades.

Estos procesos se realizan de forma cíclica hasta llegar a un criterio de parada que nosotros definimos en doscientas cincuenta generaciones o cuando durante veinte generaciones no exista ningún individuo que mejore al mejor subtour obtenido en las generaciones anteriores.

En la figura siguiente se muestra una ejecución con 15 ciudades donde se muestra la matriz de costos, la población inicial, la población final, su fitness, así como el mejor recorrido obtenido por el AG simple y por el modelo propuesto.

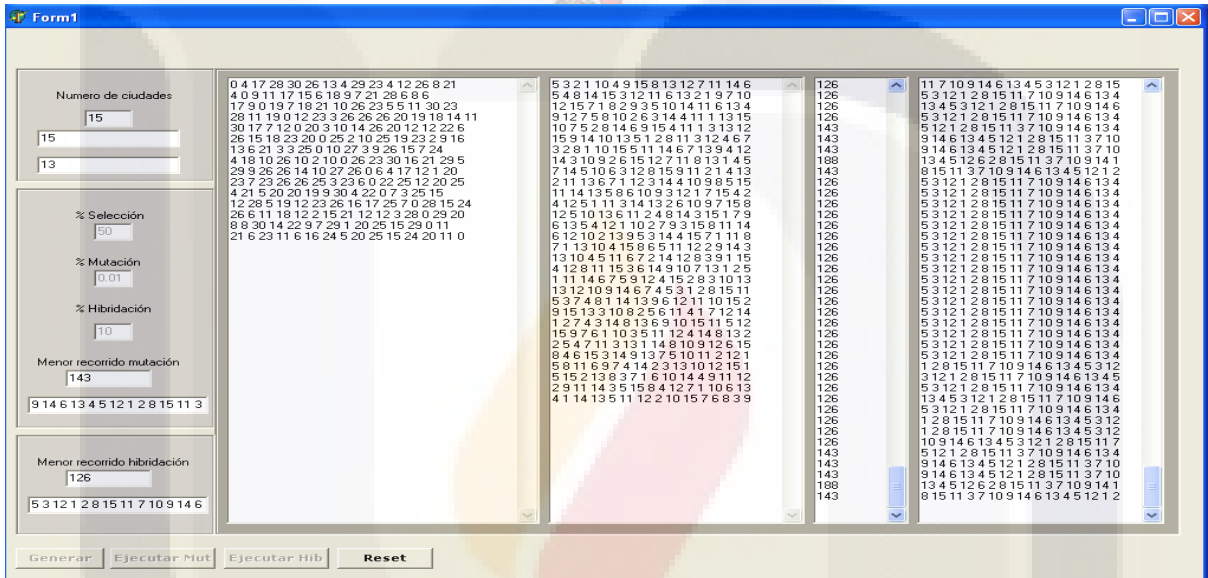


Fig.21 Ejemplo de una ejecución de la herramienta de software con parámetro de entrada igual a 15 ciudades

#### 4.1.1.1. ALGORITMO DEL MODELO PROPUESTO

Captura número de ciudades

Número de generaciones = 0

Converja = 0

Genera matriz de costos

**Para** i desde 1 **hasta** número de ciudades **hacer**

Población inicial [i] = Genera individuo

**Para** i desde 1 **hasta** número de ciudades **hacer**

Población respaldo [i] = Población inicial[i]

**Mientras** no se alcance *número de generaciones* = 250 y no *converja* = 20 **hacer** {AG simple}

**Para** *i* desde 1 **hasta** número de ciudades **hacer** {fitness}

*Evaluar individuo* de población inicial

**Para** *i* desde 1 **hasta** 50 % del número de ciudades **hacer** {selección}

*Seleccionar individuo* para nueva población

**Para** *i* desde 1 **hasta** 50% del número de ciudades **hacer** {cruza}

*Cruzar individuo* e insertar en nueva población

**Para** *i* desde 1 **hasta** 1% del número de ciudades **hacer** {mutación}

*Mutar individuo* de la nueva población

Población inicial = nueva población

**Fin mientras**

Usar población respaldo

**Mientras** no se alcance *número de generaciones* = 250 y no *converja* = 20 **hacer** {modelo propuesto}

**Para** *i* desde 1 **hasta** número de ciudades **hacer** {fitness}

*Evaluar individuo* de población respaldo

**Para** *i* desde 1 **hasta** 50 % del número de ciudades **hacer** {selección}

*Seleccionar individuo* para nueva población 2

**Para** *i* desde 1 **hasta** 40% del número de ciudades **hacer** {cruza}

*Cruzar individuo* e insertarlos en nueva población 2

**Para** *i* desde 1 **hasta** 10% del número de ciudades **hacer** {migración}

*Genera individuo* para nueva población

Población respaldo = nueva población 2

**Fin mientras**

**Mostrar** mejor individuo nueva población y nueva población 2

Pcdo 3. Pseudocódigo del modelo propuesto

Para el caso de la segunda vertiente, se introduce el proceso de mutación que se utiliza en el AG simple en el modelo propuesto después del proceso de migración.

Después de probar las dos vertientes del modelo inicial se detectó que al adicionar la migración para apoyar a la mutación en la diversificación de las poblaciones, los resultados se mejoraron.

Los pseudocódigos para cada función llamada en el algoritmo se encuentran en el Anexo A.

Se realizó un estudio comparativo para determinar si el modelo propuesto en su vertiente dos era mejor que el AG simple; para ello; se ejecutó la aplicación con número de ciudades igual a 5, 10, 15, 20, 25 y 30 teniendo 20 ejecuciones para cada valor de ciudad.

Los resultados obtenidos mostraron que en casi un once por ciento de las ocasiones, los resultados eran mejores, mientras que sólo en aproximadamente un tres por ciento de las ocasiones se empeoraban los resultados. A continuación se muestra una tabla con los resultados obtenidos.

	5				10				15				20				25				30			
1	11	36	11	36	14	60	12	55	17	129	12	129	16	186	12	186	32	195	12	195	20	275	12	275
2	11	68	11	68	14	90	12	90	14	139	12	139	16	162	12	162	18	234	12	234	40	189	12	189
3	11	40	11	40	16	65	12	65	16	118	12	118	19	168	12	168	22	196	12	196	27	229	12	220
4	11	52	11	52	13	100	12	100	17	135	12	135	36	169	12	169	20	211	12	211	20	218	12	218
5	11	52	11	52	14	59	12	59	16	137	12	137	24	136	12	136	23	196	12	196	21	236	12	247
6	11	62	11	62	64	85	12	85	20	105	12	105	20	143	12	137	15	238	12	238	24	210	12	210
7	11	53	11	53	12	117	13	96	25	117	12	117	18	177	12	177	20	175	12	175	18	247	12	247
8	11	64	11	64	16	60	12	60	14	100	12	100	17	148	12	150	18	197	12	202	27	212	12	212
9	11	51	11	51	14	87	12	87	15	112	12	112	15	161	12	161	33	173	12	173	20	214	12	124
10	11	69	11	69	14	94	12	94	15	143	13	126	16	156	12	156	20	180	12	180	29	208	12	208
11	11	59	11	59	13	88	12	88	18	125	12	125	19	169	12	169	14	220	12	220	21	201	12	201
12	11	66	11	66	16	91	12	91	18	112	13	105	21	180	12	180	19	250	12	250	18	262	12	262
13	11	58	11	58	13	84	13	78	13	110	12	110	18	146	12	146	17	172	12	172	21	226	12	226
14	11	68	11	68	13	80	12	80	15	99	12	99	14	127	12	127	26	180	12	180	23	236	12	236
15	11	53	11	53	16	89	12	89	16	141	12	141	23	132	12	132	34	183	12	183	18	232	12	232
16	11	54	11	54	13	70	12	70	13	108	12	108	19	148	12	148	18	231	12	231	20	212	13	210
17	11	56	11	56	14	89	12	89	17	146	12	146	14	133	12	133	17	222	13	203	21	253	12	253
18	11	53	11	53	13	101	12	101	16	127	13	116	13	197	12	197	22	192	12	172	19	199	12	199
19	11	51	11	51	13	45	12	45	18	94	12	94	15	126	13	123	17	206	12	206	21	212	12	212
20	11	75	11	75	14	62	12	62	17	95	12	95	16	189	12	189	26	178	12	178	19	312	12	308

Porcentaje de mejoramiento 10,83%  
 Porcentaje de empeoramiento 3,33%

Tabla1. Muestra un cuadro comparativo de los resultados obtenidos por el AG simple y por el modelo propuesto.



---

#### 4.1.2. MEJORAMIENTO DEL MODELO PROPUESTO

---

Basado en los resultados obtenidos por el modelo propuesto se prosiguió a un mejoramiento del mismo.

El primer paso fué el cambiar la estructura de datos que se utilizaba por una estructura dinámica, con lo que el número de ciudades con las que se podía trabajar se incrementó considerablemente. Por las características de la programación utilizada, se fijó en un máximo de 3500 ciudades.

Revisando la bibliografía sobre AG's encontramos algunos trabajos realizados sobre AG's paralelos que utilizaban ya el operador de migración y basaban su funcionamiento en el modelo de islas.

Como se comento en un capitulo anterior el modelo de islas tiene dos principales vertientes: los algoritmos de grano fino y algoritmos de grano grueso [Whitley1998] [Londoño2006].

Los algoritmos de grano fino están más enfocados a distribuir en varios procesadores la carga computacional de los operadores del AG, de forma que un procesador se puede encargar de la selección, otro del cruzamiento y otro de la mutación y se comuniquen entre ellos [Londoño2006].

En cuanto a los algoritmos de grano grueso, estos dividen la población total en subpoblaciones que evolucionan de forma independiente y migran individuos cada determinado tiempo. Normalmente migran bajo dos principales criterios el primero es al detectan una cercanía considerable a la convergencia de la poblacion desde la que se deca migrar, la otra es cuando dicha poblacion llego a la convergencia.

Uno de los principales objetivos del modelo de islas según la literatura [Tzung-Pei2007][Lu2004][ Skolicki2005], pretenden conseguir no solamente una mejor

diversificación de las poblaciones, sino que, al distribuir la carga computacional de los algoritmos entre varios procesadores, se mejore su rendimiento.

Desafortunadamente varios de los trabajos reportan que al incrementar el número de procesadores entre los que se dividen las poblaciones se incrementa el costo debido a la comunicación entre los mismos.

Otro problema es que si se manejan algoritmos sincronicos los procesadores de mayor potencia deben detener su ejecución hasta que el procesador mas lento libere sus procesos y en el caso de los asincronicos son mucho mas complejos de implementar con lo que se ve afectada una de las principales características del AG que es su facilidad para implementar casi cualquier problema por medio de ellos.

Dentro del modelo de islas se manejan varias arquitecturas que son comparables con algunas topologías de red. Por ejemplo está la de estrella, en la que se divide la población total en varias subpoblaciones y se selecciona la que tiene mejores individuos como central o maestra. Todas las poblaciones evolucionan de forma independiente y al llegar a algún criterio especificado migran a sus mejores individuos de la central hacia las demás subpoblaciones y de éstas a la central. Los criterios que se utilizan para la migración son un determinado número de generaciones o una convergencia a óptimos locales.

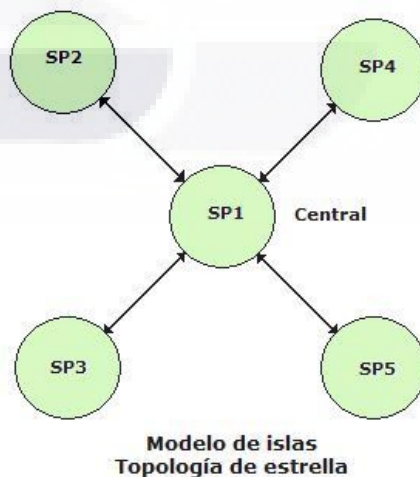


Fig. 22 Topología de estrella utilizada en los algoritmos de grano grueso

En este tipo de topología el flujo de datos se hace de la población central a las demás subpoblaciones y de estas a la central.

Otro tipo es la de anillo (que es de las mas utilizadas por los resultados que a demostrado tener en contra del AG secuencial [Tzung-Pei2007]) donde cada subpoblación está conectada con dos poblaciones vecinas hasta formar un anillo cerrado. Para cada subpoblación existe un flujo de salida hacia una población vecina y un flujo de entrada desde el otro vecino. La forma en que se migra es similar a la del algoritmo de estrella.

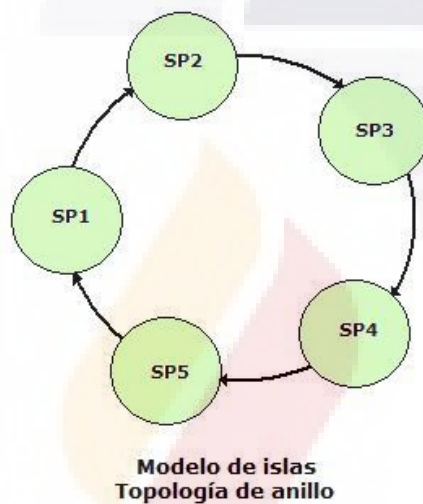


Fig. 23 Topología de anillo utilizada en el algoritmo de grano grueso.

Un algoritmo más es el de malla, en el que todas las subpoblaciones están interconectadas y el flujo es de cada una con todas las demás. La migración se hace siguiendo los criterios de los dos algoritmos anteriores.

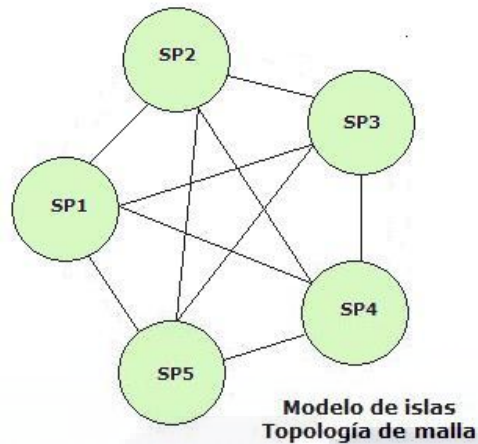


Fig. 24 Topología de Malla utilizada en el algoritmo de grano grueso.

Los parámetros que se utilizan en el modelo de islas son la topología a utilizar (anillo, cliente servidos o estrella, malla, entre otros), la evaluación de que individuos son los que se van a migrar, determinar cada cuantas generaciones se va a migrar o al llegar a que criterio y por ultimo el numero de individuos a migrar.

Actualmente los trabajos que se realizan sobre el modelo de islas van enfocados principalmente a encontrar la combinación de parámetros que permitan obtener mejores resultados.

Otro concepto muy importante que surge de los AG's paralelos y que tiene una analogía con la naturaleza es la "Especiación".

En la naturaleza se tienen registrados casos como lo es el del Sengi de Namibia que es un mamífero con un gran parecido a un ratón pero que cuenta con patas similares a las de los antílopes, trompa parecida a la del oso hormiguero y cola de rata.

Resientemente se han hecho estudios genéticos sobre este pequeño animal y se ha encontrado que pertenece a un grupo ancestral común de varios animales africanos cuyos parientes más próximos son los elefantes, los osos hormigueros, entre otros esto es la especiación. Estos



estudios fueron realizados por la “California Academy of Science” y alguna información se encuentra en un artículo titulado “Mystery Sengi”.

Basados en este concepto se puede determinar que el material genético de una población, si se desarrolla en condiciones diferentes permite generar individuos con características muy diversas lo cual es altamente deseable en un AG.

Basándonos en el estudio de los AG's paralelos, se determinó realizar algunas modificaciones más al modelo propuesto. Entre ellas y ya que hasta el momento, el proceso de migración se realizaba por medio de la introducción de individuos generados aleatoriamente, se comenzó por introducir una población adicional que se genera al igual que la población inicial y la de respaldo del mismo tamaño y al inicio del algoritmo. Esta población también se genera de manera aleatoria.

Durante la ejecución del algoritmo, esta población evoluciona de manera paralela a la población de respaldo y cada determinado número de generación se toma un diez por ciento de esta y se introducen en la población de respaldo.

La población paralela evoluciona al igual que la población principal aplicando a ella los operadores de selección, cruzamiento y mutación.

Otra modificación que se realizó, es la selección de las herramientas que detectamos, mejores resultados proporcionan al AG. Por ejemplo se sustituyó la selección elitista pura por una selección en la que se combina elitismo y el muestreo universal estocástico, esto seleccionando un treinta y tres por ciento de individuos a seleccionar por medio de elitismo (para no perder las mejores soluciones obtenidas hasta el momento) y el resto de la selección se obtiene por muestreo universal estocástico.

Para el cruzamiento, se sigue utilizando el GSX con una pequeña modificación, la selección del elemento inicial no se realiza de forma aleatoria, sino que se hace un recorrido en los



padres desde el final del primero e inicio del segundo para intentar detectar el elemento que permite generar un recorrido mayor para conservar subtours de mayor tamaño.

Actualmente la herramienta funciona de la siguiente forma:

Primero se genera la población inicial y se hace una copia de la misma, también se genera la población paralela. La población inicial y la copia nos sirven para sobre la primera ejecutar el AG simple y sobre la segunda el modelo propuesto y hacer sobre los resultados obtenidos de cada una un diseño de experimentos que nos permita determinar si el modelo propuesto tiene una mejora significativa. El tamaño de la población es un parámetro definido por el usuario.

Las tres poblaciones evolucionan bajo una simulación de paralelismo, aplicando primeramente la evaluación por medio de la función objetivo que en este caso consiste en sumar el costo entre los pares de ciudades que conforman el cromosoma tomando en cuenta su orden de aparición.

Una vez evaluados se selecciona un cincuenta por ciento de los individuos que pasan a formar parte de la nueva generación. Para ello se elige un treinta y tres por ciento de los individuos necesarios por elitismo y el sesenta y cuatro por ciento restante por muestreo universal estocástico.

Sobre los individuos seleccionados se realiza el proceso de cruce (por medio de GSXM) para obtener el cincuenta por ciento restante para conformar el total de la población de la nueva generación. Este proceso se sigue en la población inicial y en la paralela así como para la población de respaldo exceptuando en esta cada cierto número de generaciones (definido por el usuario) donde solo se genera por cruzamiento un cuarenta por ciento de los individuos y el diez restante se migra por medio la selección de los mejores individuos de la población paralela.

El operador de mutación se aplica a las tres poblaciones en un uno por ciento y se discrimina al mejor individuo obtenido de dicha generación.

Este proceso se efectua por un numero de generaciones que es definido por el usuario.

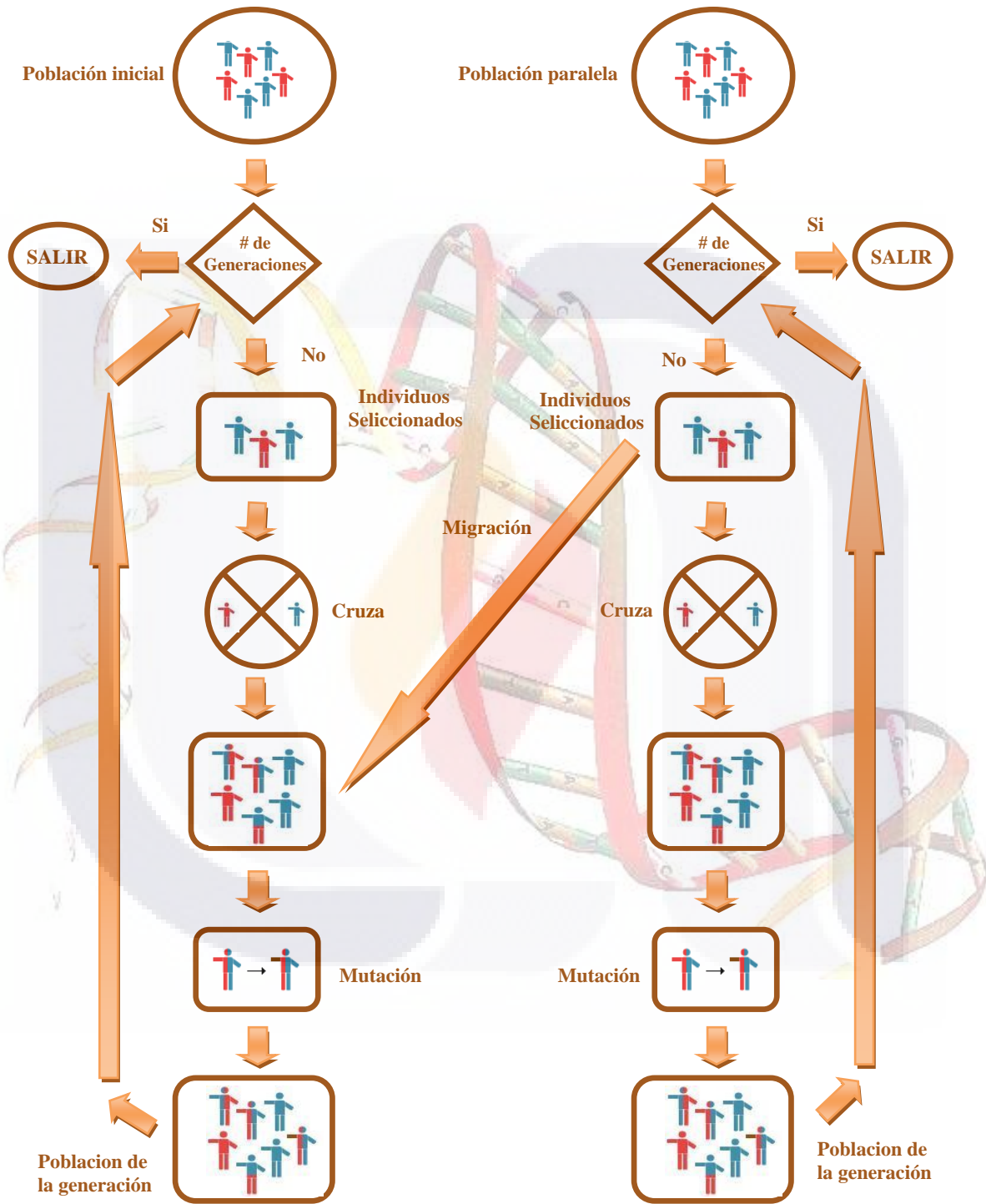


Fig. 25 Diagrama del modelo de migración utilizando población paralela.

Captura número de ciudades

Número de generaciones = 0

Converja = 0

**Para** i desde 1 **hasta** tamaño de la población **hacer**

**Para** j desde 1 **hasta** tamaño de la población **hacer**

Matriz de costos [i,j] = numero al azar

Matriz de costos [j,i] = Matriz de costos [i,j]

**Fin para**

**Fin para**

**Para** i desde 1 **hasta** tamaño de la población **hacer**

Población inicial [i] = *Generar individuo*

**Para** i desde 1 **hasta** tamaño de la población **hacer**

Población respaldo [i] = Población inicial[i]

**Para** i desde 1 **hasta** tamaño de la población **hacer**

Población respaldo [i] = *Generar individuo*

**Mientras** no se alcance número de generaciones **hacer** {AG simple}

**Para** i desde 1 **hasta** tamaño de la población **hacer** {fitness}

*Evaluar individuo* de población inicial

**Para** i desde 1 **hasta** (33% del 50 % del tamaño de la población) **hacer**

{selección elitista}

*Selección\_elit* para nueva población

**Para** i desde 1 **hasta** (64% del 50 % del tamaño de la población) **hacer**

{selección MUE}

*Selección\_MUE* para nueva población

**Para** i desde 1 **hasta** 50% del tamaño de la población **hacer** {cruza}

*Cruzar e insertar* en nueva población

**Para** i desde 1 **hasta** 1% del tamaño de la población **hacer** {mutación}

*Mutar individuo* de la nueva población menos el mejor

Población inicial = nueva población

**Fin mientras**

Usar población respaldo

**Mientras** no se alcance *número de generaciones* **hacer** {modelo propuesto}

**Para** *i* desde 1 **hasta** *tamaño de la población* **hacer** {fitness}

*Evaluar individuo* de población respaldo

*Evaluar individuo* de población paralela

**Fin para**

**Para** *i* desde 1 **hasta** (33% del 50% del *tamaño de la población*) **hacer**  
{selección elitista}

*Selección\_elit* para nueva población 2

*Selección\_elit* para nueva población paralela

**Para** *i* desde 1 **hasta** (64% del 50 % del *tamaño de la población*) **hacer**  
{selección MUE}

*Selección\_MUE* para nueva población 2

*Selección\_MUE* para nueva población paralela

**Para** *i* desde 1 **hasta** 40% del número de ciudades **hacer** {cruza}

*Cruzar individuo* e insertarlo en nueva población 2

**Fin para**

**Para** *i* desde 1 **hasta** 10% del *tamaño de la población* **hacer** {migración}

*Selección\_elit de población paralela* para nueva población 2

**Para** *i* desde 1 **hasta** 50% del *tamaño de la población* **hacer** {cruza}

*Cruzar individuo* e insertarlo en nueva población paralela

**Fin para**

**Para** *i* desde 1 **hasta** 1% del *tamaño de la población* **hacer** {mutación}

*Mutar individuo* de la nueva población 2 menos el mejor

*Mutar individuo* de la nueva población paralela menos el mejor

**Fin para**

Población respaldo = nueva población 2

Poblacion paralela = nueva población paralela

**Fin mientras**

**Mostrar** mejor individuo nueva población y nueva población 2

También se modificó el diseño de la herramienta para que fuera más sencillo interpretar los datos que se obtenían.

Por último se modificó la herramienta para poder insertar datos a la misma y que no solamente trabaje con datos aleatorios. Actualmente la herramienta permite insertar datos por medio de una matriz de costos en forma de tabla o por medio de una tabla de distancias euclidianas.

A continuación se muestran gráficamente algunas características de la nueva herramienta.

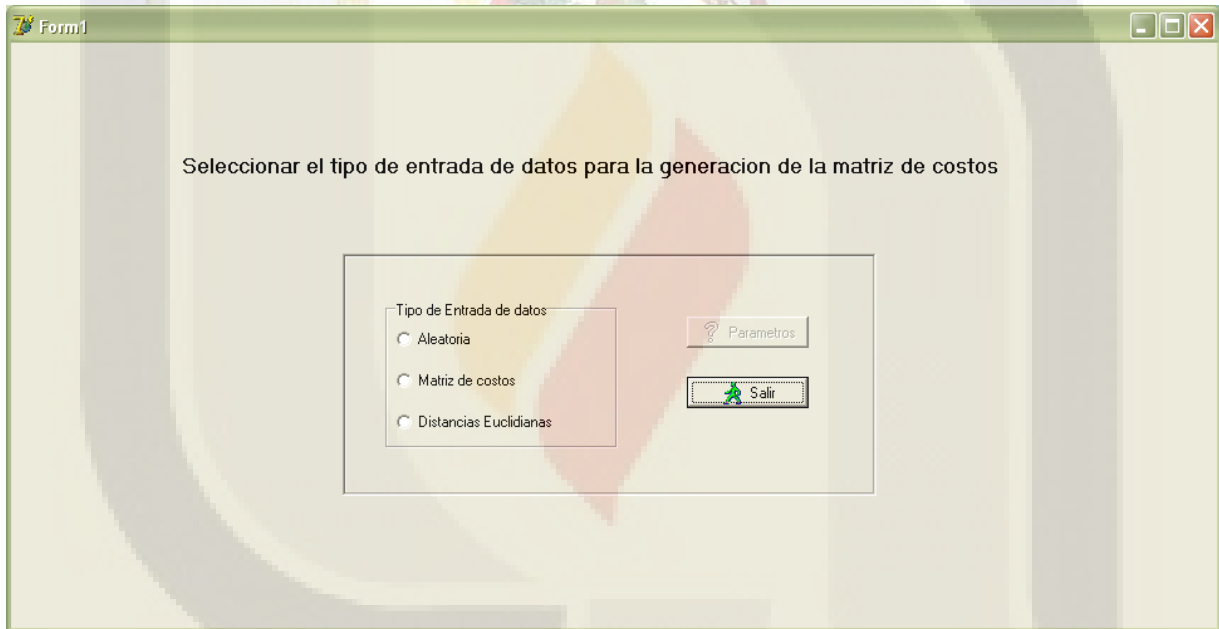


Fig. 26 Segunda versión de la herramienta de software.



**Form1**

**Parametros**

**No. Ciudades**  
100

**No. de Generaciones**  
3000

**Tamaño de la poblacion**  
100

Limpiar  
Generar  
Regresar

**Distancia Euclidiana**

79 378 1048  
80 1178 100  
81 2599 901  
82 3416 143  
83 2961 1605  
84 611 1384  
85 3113 885  
86 2597 1830  
87 2586 1286  
88 161 906  
89 1429 134  
90 742 1025  
91 1625 1651  
92 1187 706  
93 1787 1009  
94 22 987  
95 3640 43  
96 3756 882  
97 776 392  
98 1724 1642  
99 198 1810  
100 3950 1558

Fig. 27 Entrada de los parámetros para un problema con una matriz de distancias Euclidianas.

**Form1**

Poblacion Inicial	Poblacion de Respaldo	Fitness	Poblacion a Migrar	Fitness
4 12 69 7 81 3 20 46 57 47 61	4 12 69 7 81 3 20 46 57 47 61	04	30 62 88 94 92 40 9 23 76 16 4	0359283
96 16 62 78 41 36 10 35 58 96	96 16 62 78 41 36 10 35 58 96	167833	37 32 99 43 62 36 60 71 39 95 4	86
31 49 3 6 78 86 82 65 53 45 5	31 49 3 6 78 86 82 65 53 45 5	7036109	51 13 41 45 56 63 19 11 79 29 3	172339
36 80 27 56 59 81 31 66 58 66	36 80 27 56 59 81 31 66 58 66	91	52 66 95 43 56 9 64 81 98 90 4	4784281
62 2 81 73 63 88 30 29 35 64	62 2 81 73 63 88 30 29 35 64	140718	38 30 13 85 58 23 7 20 95 55 3	72
41 71 37 87 75 83 27 97 15 77	41 71 37 87 75 83 27 97 15 77	0576623	40 74 98 35 97 24 5 64 42 89 4	175977
10 55 4 53 28 29 20 49 60 41	10 55 4 53 28 29 20 49 60 41	21	73 28 81 20 33 55 86 39 15 95 5	4379785
38 94 49 58 33 70 18 24 34 85	38 94 49 58 33 70 18 24 34 85	176485	32 67 26 69 36 65 17 4 66 84 9	18
2 97 92 34 71 35 86 26 7 62 1	2 97 92 34 71 35 86 26 7 62 1	0045804	26 72 79 99 47 56 65 21 49 3 96	154118
13 29 90 39 27 6 59 42 91 76	13 29 90 39 27 6 59 42 91 76	47	87 28 50 38 6 92 62 75 42 94 6	0669321
81 64 71 82 42 32 67 31 14 1	81 64 71 82 42 32 67 31 14 1	166196	46 77 45 64 90 53 66 5 10 98 6	6
75 87 55 51 93 54 84 49 32 1	75 87 55 51 93 54 84 49 32 1	5977370	5 78 30 48 64 83 44 16 98 23 9	186595
78 92 81 6 50 79 49 71 72 31	78 92 81 6 50 79 49 71 72 31	28	67 98 77 5 31 44 14 39 84 17 8	9751841
21 34 23 82 3 39 26 60 51 84	21 34 23 82 3 39 26 60 51 84	177438	44 100 60 78 82 85 73 67 37 36	43
14 23 30 2 4 83 12 50 38 66 1	14 23 30 2 4 83 12 50 38 66 1	9768553	65 75 42 69 58 26 67 31 78 6 1	162031
31 98 25 74 76 87 10 79 100 3	31 98 25 74 76 87 10 79 100 3	04	34 5 99 98 45 79 31 17 49 55 2	6030045
90 20 62 34 43 46 28 93 18 1	90 20 62 34 43 46 28 93 18 1	165663	89 70 94 76 99 41 46 86 14 36 1	11
32 49 76 89 39 72 24 69 27 1	32 49 76 89 39 72 24 69 27 1	8410660	89 55 32 6 50 38 88 47 43 56 8	168101
61 54 90 63 22 45 66 21 69 8	61 54 90 63 22 45 66 21 69 8	04	80 86 3 79 33 42 46 7 19 34 95	1536704
64 4 70 32 8 23 87 55 43 27 8	64 4 70 32 8 23 87 55 43 27 8	158852	86 28 10 93 83 82 29 44 65 61 5	24
85 8 2 55 83 42 96 79 14 81 6	85 8 2 55 83 42 96 79 14 81 6	7267647	97 41 4 74 38 83 23 15 24 57 9	177339
96 37 49 7 35 17 8 81 90 9 86	96 37 49 7 35 17 8 81 90 9 86	91	15 55 69 21 74 93 54 83 34 42 2	0610050
41 55 25 53 91 93 30 64 17 7	41 55 25 53 91 93 30 64 17 7	164928	65 37 39 88 85 15 84 18 62 58 3	47
10 17 64 27 40 45 35 73 30 24	10 17 64 27 40 45 35 73 30 24	8505982	97 88 74 87 85 62 47 58 41 16 5	168329
35 36 49 40 23 45 89 7 22 62	35 36 49 40 23 45 89 7 22 62	47	89 12 45 27 41 57 44 93 6 95 91	1810698
12 74 65 98 68 23 92 10 28 1	12 74 65 98 68 23 92 10 28 1		96 65 14 22 38 1 85 48 50 56 7	

Ejecutar  
Regresar

Fig. 28 Ejecución de la instancia kroA100 del TSPLIB (100 ciudades).

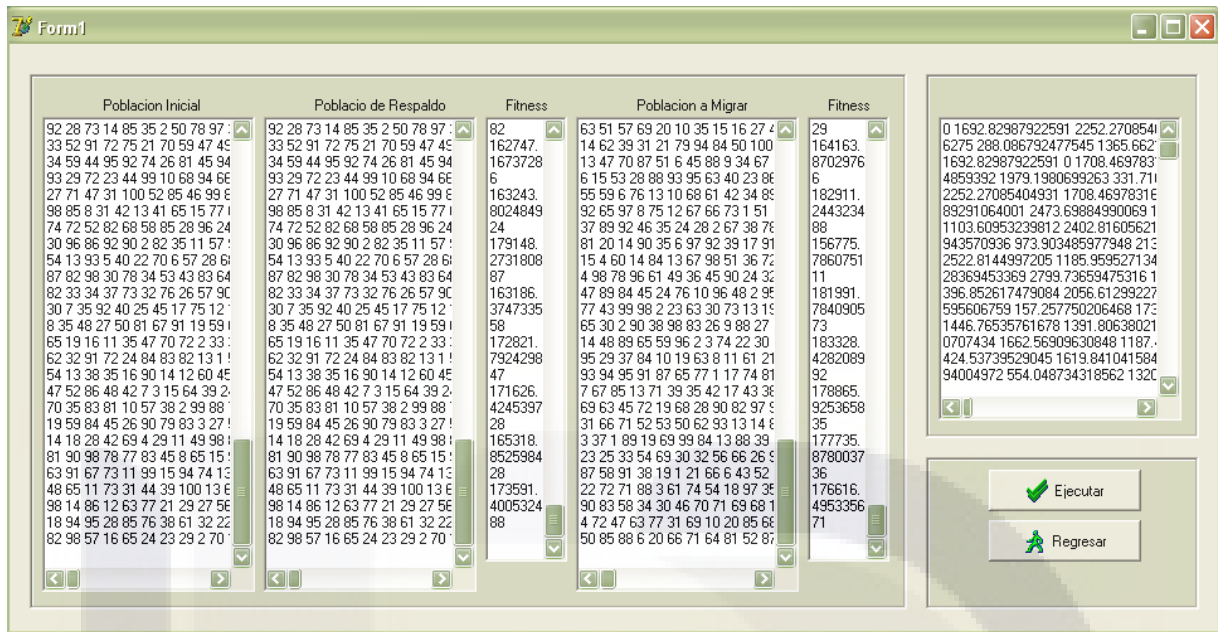


Fig. 29 Generacion de las poblaciones y matriz de costos.

#### 4.1.3. DISEÑO DE EXPERIMENTOS

Para realizar las pruebas sobre el modelo propuesto y comprobar si las mejoras son significativas en comparación a los resultados obtenidos por el AG simple se realiza el siguiente diseño de experimentos.

Se realizaron 1080 ejecuciones utilizando la herramienta desarrollada para resolver la instancia del TSPLIB kroA100. Para ello se utilizaron las siguientes combinaciones de parrametros.

Tamaño de la poblacion en base al numero de ciudades	Numero de Generaciones entre Migraciones	Numero de Generaciones total
50%	5	500
75%	7	1000
100%	9	2000
		3000

Tabla2. Relacion de parámetros para las ejecuciones de la herramienta.

Para cada combinación de parámetros se realizaron 30 ejecuciones y se obtuvieron los resultados que se muestran en el Anexo C.

Después de las 1080 ejecuciones los resultados obtenidos se procesaron en el SPSS que es un programa estadístico informático que permite hacer análisis de datos para la verificación de modelos lineales, no lineales, test de independencia, entre otros.

Primero pasaron los datos a una tabla para que el programa SPSS los pudiera recibir.

Después se cargaron los datos (desde una tabla de Excel) al SPSS y se analizó por medio de una prueba no paramétrica (ya que su distribución no puede ser definida a priori, ya que los datos introducidos son los que la determinan) de dos variables independientes (porque queremos comprobar que al introducir la migración dentro del AG simple los resultados se mejoran por ello tenemos 2 grupos con y sin migración).

La variable a probar fue la diferencia de valor que hubo contra el mejor reportado y los grupos fueron: 1 para el AG simple y 2 para el modelo propuesto.

Los resultados fueron los siguientes:

### NPar Tests

#### Descriptive Statistics

	N	Mean	Std. Deviation	Minimum	Maximum
DIFER	2160	25172.6873	6974.5079	10332.70	48557.99
SGA_IGA	2160	1.5000	.5001	1.00	2.00

## Mann-Whitney Test

### Ranks

	SGA_IGA	N	Mean Rank	Sum of Ranks
DIFER	1.00	1080	1169.38	1262933.00
	2.00	1080	991.62	1070947.00
	Total	2160		

### Test Statistics

	DIFER
Mann-Whitney U	487207.000
Wilcoxon W	1070947.000
Z	-6.623
Asymp. Sig. (2-tailed)	.000

a Grouping Variable: SGA\_IGA

Por medio de los resultados anteriores podemos determinar que el modelo propuesto tiene una mejora significativa al obtener como valor de la media 991.62 contra 1169.38 del AG simple (cuyo objetivo es la minimización de la diferencia).

También se realizó un análisis basado en un modelo general lineal univariado y nos arrojó como resultado que la mejor combinación de parámetros era 3000 generaciones, 100% de población y utilizando el modelo propuesto.

Además se detectó que el número de generaciones entre migraciones no afectaba las soluciones obtenidas.

---

#### **4.1.4. COSTO COMPUTACIONAL**

---

Se realizó un estudio para verificar el costo computacional utilizado por el algoritmo desarrollado. La forma en que se calculó se muestra a continuación.

Se tienen siete funciones a las que llama nuestro algoritmo principal y son:

- Genera individuo
- Evaluar individuo
- Selección\_elit
- Selección\_MUE
- Cruzar individuo
- Mutar
- Limpiar

Tomamos como base para el calculo de la complejidad que si no existe ningún ciclo, la instrucciones son representadas por cualquier variable en este caso utilizaremos n.

Complejidad de Genera individuo

**Para** i desde 1 **hasta** número de ciudades **hacer**

Lista [i] = i

elemento = valor al azar (num ciudades)

Lista1 = Lista

**Para** i desde 1 **hasta** elemento **hacer**

Lista1 = siguiente elemento

Individuo = Lista1

borra Lista1

**Para** i desde número de ciudades-1 **hasta** 2 **hacer**

elemento = valos al azar (i)

Lista1 = Lista

**Para** j desde 1 **hasta** elemento **hacer**

Lista1 = siguiente elemento

Individuo = Individuo + Lista1

borrar Lista1



**Fin para**

En el algoritmo anterior se puede detectar que existen varios ciclos Para los que son representados por  $n$  pero al existir una anidación de dos ciclos Para esto produce una complejidad de  $n*n$  lo que produce una  $n^2$ . Por lo que la complejidad de esta función es  $O(n^2)$ .

Complejidad de Evaluar individuo

```

valor = 0
lista1 = lista
lista2 = siguiente elemento lista
Para i desde 1 hasta numero de ciudades - 1 hacer
    valor = valor + matriz[lista1,lista2]
    lista1 = lista2
    lista2 = siguiente elemento lista
Fin para
Fitness = valor
Para i desde 2 hasta número de ciudades hacer
    valor = 0
    lista = siguiente lista
    lista1 = lista
    lista2 = siguiente elemento lista
    Para i desde 1 hasta numero de ciudades - 1 hacer
        valor = valor + matriz[lista1,lista2]
        lista1 = lista2
        lista2 = siguiente elemento lista
    Fin para
    Siguiete fitness = valor
Fin para
    
```

Al igual que en la función anterior las instrucciones con mayor complejidad son una anidación de dos ciclos Para por lo que la complejidad de esta función es  $O(n^2)$ .

Complejidad de Selección\_elit

Menor = (Primer fitness)

**Para** j desde 2 **hasta** número de ciudad **hacer**

**Si** fitness[j] < menor y no se ha seleccionado **hacer**

Menor = fitness[j]

**Fin si**

**Fin para**

elemento = menor

**Para** j desde 1 **hasta** número de ciudades **hacer**

Nuevo individuo = nuevo individuo + elemento

elemento = siguiente elemento

**Fin para**

Para esta función se tienen dos ciclos Para simples por lo que la complejidad de la función es  $O(n^2)$ .

Complejidad de Selección\_MUE

**Para** j desde 1 **hasta** tamaño población **hacer**

porcentaje[j] = 0

individuo = lista

**Para** j desde 1 **hasta** tamaño población **hacer**

porcentaje[j] = porcentaje (individuo)

primero = valor aleatorio (100)

cuantos = 100 / generar

**Para** j desde 1 **hasta** generar **hacer**

**Si** primero <= porcentaje[1] **entonces**

elemento = lista

**Para** k desde 1 **hasta** numero de ciudade **hacer**

Individuo = individuo + elemento

elemento = siguiente elemento

**Fin para**

primero = primero + cuantos

**Fin si**

**Sino**

siguiente = 2

**Mientras** primero > porcentaje [siguiente] **hacer**

siguiente = siguiente + 1

**Fin mientras**

elemento = lista

**Para** k desde 1 **hasta** numero de ciudade **hacer**

Individuo = individuo + elemento

elemento = siguiente elemento

**Fin para**

primero = primero + cuantos

**Fin sino**

**Si** primero > 100 **entonces**

primero = primero - 100

**Fin para**

Para esta función se tienen varios ciclos Para cuya evaluación es  $n$ , además a diferencia de las funciones anteriores se cuenta con un ciclo mientras que tiene un comportamiento logarítmico, pero como se maneja como un ciclo finito se puede evaluar igual que un ciclo Para, pero al estar anidado dentro de un Para el resultado es  $n^2$ , al igual existe un ciclo Para anidado en otro cuya complejidad es  $n^2$ , por esta razón como la mayor complejidad es  $n^2$  la complejidad de la función es  $O(n^2)$ .

Complejidad Cruza individuo

```

x = valor al azar (número de ciudades)
i = 1
Mientras Individuo1[i] <> x hacer
    i = i + 1
j = 1
Mientras Individuo2[j] <> x hacer
    j = j + 1
nuevo individuo = x
i = i - 1
j = j + 1
Mientras (individuo1 > 0 o individuo2 <= número de ciudades) hacer
    Si Individuo1[i] no se encuentra en nuevo individuo entonces
        nuevo individuo = individuo1[i] + nuevo individuo
    Si i>0 entonces
        i = i -1
    Si Individuo2[j] no se encuentra en nuevo individuo entonces
        nuevo individuo = nuevo individuo + individuo2[j]
    Si j<número de ciudades entonces
        j = j + 1
Fin mientras
Para j desde 1 hasta número de ciudades hacer
    Si individuo1[j] no esta en nuevo individuo entonces
        Nuevo individuo = nuevo individuo + individuo1[j]
    
```

En esta función se tienen tres ciclos mientras y un ciclo Para, pero al igual que en la función anterior el comportamiento de los ciclos mientras es finito su evaluación es n y como la mayor complejidad es n la función tiene una complejidad O(n).

Complejidad de la función Mutar

x = numero al azar (tamaño de población)

individuo = lista

**Para j desde 1 hasta x hacer**

    individuo = siguiente individuo

x = valor al azar (número de ciudades)

elemento1 = individuo

**Para j desde 1 hasta x hacer**

    elemento1 = siguiente (elemento1)

x = valor al azar (número de ciudades)

elemento2 = individuo

**Mientras valor de elemento2 <> x hacer**

    elemento2 = siguiente (elemento2)

    respaldo = elemento1

    elemento1 = elemento2

    elemento2 = Respaldo

Esta función cuenta con dos ciclos Para y un mientras con el mismo comportamiento que en los anteriores y no existe ninguna anidación por lo que la complejidad de la función es  $O(n)$ .

Complejidad función limpiar

Individuo1 = lista

**Para j desde 1 hasta tamaño de la población hacer**

    individuo2 = siguiente individuo1

    elemento = individuo1

**Para i desde 1 hasta numero de ciudades hacer**

        individuo1 = siguiente elemento

        borrar elemento

        elemento = individuo1

**Fin para**

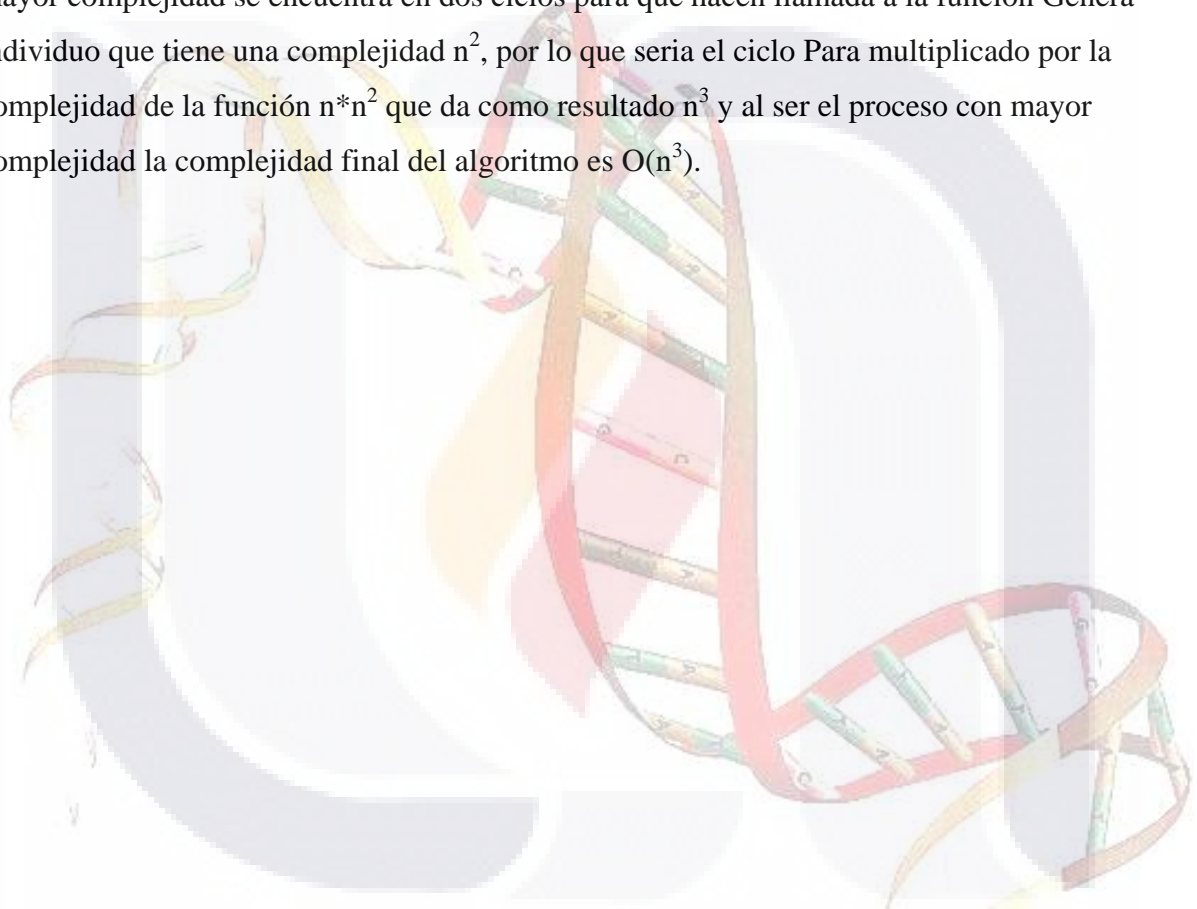


individuo1 = individuo2

**Fin para**

En la función Limpiar se tiene una anidación de dos ciclos Para cuya complejidad es la mayor en la función por lo que esta tiene una complejidad  $O(n^2)$ .

Tomando en cuenta los costos de la funciones y basado en al algoritmo principal se tien que la mayor complejidad se encuentra en dos ciclos para que hacen llamada a la función Genera individuo que tiene una complejidad  $n^2$ , por lo que seria el ciclo Para multiplicado por la complejidad de la función  $n*n^2$  que da como resultado  $n^3$  y al ser el proceso con mayor complejidad la complejidad final del algoritmo es  $O(n^3)$ .



## CAPÍTULO 5

### **REPORTE Y DISCUSIÓN DE RESULTADOS**

---

Como se comentó anteriormente, para probar la herramienta de software desarrollada en su primera versión, se realizaron 120 ejecuciones con parámetros de entrada entre 5 y 30 ciudades que es el valor máximo que permite la primera versión del software.

Después de las 120 ejecuciones, los resultados arrojados fueron, que si se utilizan como parámetros de entrada números de ciudades pequeños, los resultados tanto del AG clásico como del modelo generado son iguales.

De las ejecuciones con parámetro de entrada igual o mayor de 10 ciudades se detectó una mejora significativa.

Con el nuevo modelo se proponía mejorar las soluciones obtenidas por el AG simple y en base a los resultados obtenidos podemos comprobar que en un diez por ciento (aproximadamente) de las ejecuciones esto se cumplió.

Aún cuando en un tres por ciento (aproximadamente) se empeoró el resultado del AG clásico se considera que la propuesta del nuevo modelo es buena.

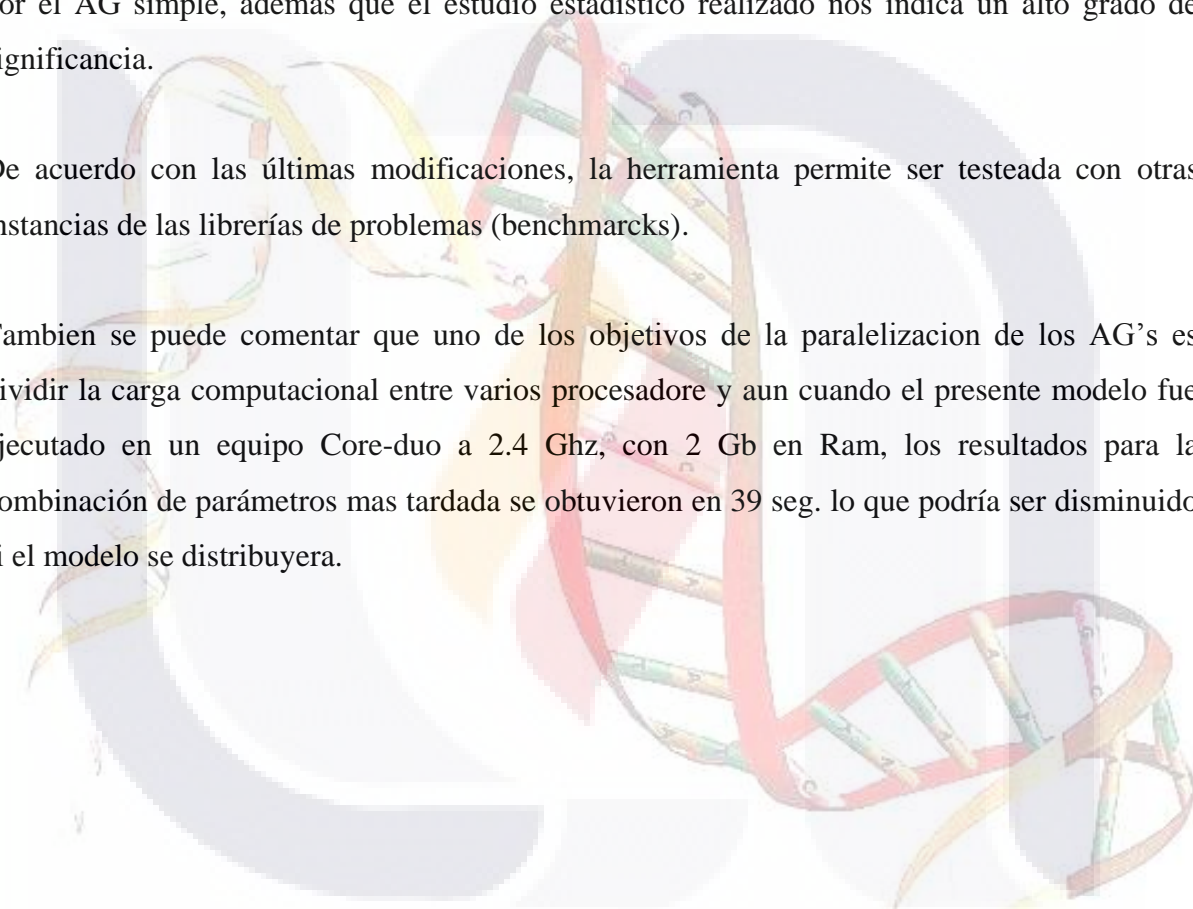
Además, una mejora que no se tenía contemplada y que en base a los resultados ha sido evidente es que en aproximadamente un noventa y nueve por ciento de las ejecuciones, el número de generaciones necesarias para converger a un resultado fue menor que con el AG simple y en aproximadamente un ochenta y tres por ciento de las ejecuciones se obtuvo el mismo resultado.

En una segunda versión de la herramienta se realizaron modificaciones significativas que permitieron probar la herramienta con un mayor número de ciudades, además de que se modificaron algunos factores como fueron el tipo de selección, una mejora en el cruzamiento y la inserción del una población paralela de la cual ir migrando individuos.

Los resultados que se obtuvieron de esta segunda versión fueron muy significativos ya que en aproximadamente un setenta por ciento de los casos, los resultados mejoraron a los obtenidos por el AG simple, además que el estudio estadístico realizado nos indica un alto grado de significancia.

De acuerdo con las últimas modificaciones, la herramienta permite ser testada con otras instancias de las librerías de problemas (benchmarks).

También se puede comentar que uno de los objetivos de la paralelización de los AG's es dividir la carga computacional entre varios procesadores y aun cuando el presente modelo fue ejecutado en un equipo Core-duo a 2.4 Ghz, con 2 Gb en Ram, los resultados para la combinación de parámetros más tardada se obtuvieron en 39 seg. lo que podría ser disminuido si el modelo se distribuyera.



## CAPÍTULO 6

### CONCLUSIONES Y TRABAJO FUTURO

---

#### 6.1. CONCLUSIONES

---

Tomando en cuenta los resultados obtenidos, consideramos que el modelo propuesto obtuvo buenos resultados en comparación con el AG simple por lo que esta modificación se podría hacer a AG más avanzados que cuentan actualmente con un mejor rendimiento y generar AG más competitivos.

Aunque como se comentó anteriormente, el objetivo de nuestra propuesta era mejorar las soluciones (lo cual con el software realizado se demostró), encontramos otra opción de mejora para llegar a los mismos resultados pero con un menor número de iteraciones, esto produciría un AG más competitivo contra otras herramientas heurísticas que se caracterizan por su rapidez.

Al crear un algoritmo que produzca resultado de buena calidad y en menos tiempo comparado con otras heurísticas, generaría que muchos de los problemas que se resuelven actualmente con estas heurísticas migraran al modelo propuesto, ya que si se tienen herramientas que llegan a resultados iguales o similares el siguiente parámetro que se toma en cuenta es la velocidad con que se obtienen dichos resultados y esto es lo que ha generado que aún cuando los AG's obtienen muy buenos resultados y son sencillos de adaptar a casi cualquier problema, el número de ejecuciones que realiza en varios problemas es mayor al de otras herramientas.

Otro factor importante es que la selección adecuada de herramientas permite una mejora significativa en el funcionamiento del algoritmo, pues permite que se evite la carga innecesaria de trabajo sobre el mismo.

Conociendo las ventajas y desventajas de cada herramienta que se utiliza para cada uno de los factores del AG (selección, cruza, mutación y migración), se puede tener una combinación de los mismos que por sí sola mejore el rendimiento del mismo.

El aprendizaje que se obtuvo fue gradual. Inicialmente se tuvo que realizar un estudio sobre el proceso biológico que produce la evolución y que me sirvió de base para seleccionar la migración como proceso de mejora de la diversificación de las poblaciones [Ornelas2007].

Una vez que se selecciono la migración como operador para incrementar la diversificación en los AG's, se realizo un estudio para determinar cuál era la mejor forma de agregarlo como herramienta dentro del AG simple [Ornelas2008].

Por último, se detecto que no solamente el agregar el operador de migración mejoraba la diversificación, sino que la elección adecuada de las herramientas que se utilizan para el proceso de selección, mutación y cruza, permite alcanzar mejores resultados aprovechando sus características [Ornelas2009].

## ***6.2. TRABAJO FUTURO***

---

Dentro del trabajo futuro se tiene contemplado continuar nuestra investigación sobre esta propuesta, además de una mejora al software para aplicarla a problemas reales dentro de la región.

También se realizará un estudio comparativo del modelo propuesto contra herramientas que utilizan EDAS o ACO para resolver el mismo problema y que están altamente sustentados.



Por ejemplo trabajos realizados por Valdez, Hernández y Botello así como Berlanga, Patricio, García y Molina, donde se muestra distintas configuraciones de las herramientas que mejores resultados obtienen dentro de los EDAS [Valdez2010][Berlanga2010] y los trabajos de Ochoa, Hernandez, Cruz, Ponce, Montes, Li y Janacek [Ochoa2010], los de Cong, Jia y Deng [Cong2010] y los de Reza, Rahimi, Shah-Hosseini [Reza2008] sobre la aplicación del Algoritmo de Colonia de Hormigas (y algunas otras técnicas) para resolver el TSP.

Se planea utilizar el modelo generado dentro de otras aéreas como lo es minería de datos para comprobar la eficacia de la propuesta. Algunos trabajos que actualmente trabajan con Minería de Datos Híbrida así como con migración para mejorar la clasificación de los esquemas son: Janssens, Zhang, Glänzel [Janssens2009], donde utilizan cauterización híbrida para mejorar la clasificación de esquemas y Ponce, Hernández, Ochoa, Padilla, Padilla, Álvarez y Ponce de León [Ponce2009], donde aplican minería de datos híbrida en aplicaciones Web.

Se podría utilizar el modelo generado y la minería de datos para diversificar la multiculturalidad en un sistema de recomendaciones inteligente.

También se tratará de llevar la propuesta a un multiprocesamiento que creemos podría mejorar aun más los resultados obtenidos.

Otra vertiente que nos interesaría investigar en el futuro es el introducir la teoría de esquemas, por medio de bloques constructores que se puedan aplicar en las distintas fases del algoritmo. Se ha identificado que una de las características para la generación de bloques constructores al utilizarlo para el TSP es la relación entre las ciudades y no la posición de las mismas en la cadena.

## ANEXO A

### PSEUDOCODIGOS

---

#### A1. GENERA INDIVIDUO

---

**Para** i desde 1 **hasta** número de ciudades **hacer**

Lista [i] = i

elemento = valor al azar (num ciudades)

Lista1 = Lista

**Para** i desde 1 **hasta** elemento **hacer**

Lista1 = siguiente elemento

Individuo = Lista1

borra Lista1

**Para** i desde número de ciudades-1 **hasta** 2 **hacer**

elemento = valos al azar (i)

Lista1 = Lista

**Para** j desde 1 **hasta** elemento **hacer**

Lista1 = siguiente elemento

Individuo = Individuo + Lista1

borrar Lista1

**Fin para**

## A2. EVALUAR INDIVIDUO

---

```

valor = 0
lista1 = lista
lista2 = siguiente elemento lista
Para i desde 1 hasta numero de ciudades – 1 hacer
    valor = valor + matriz[lista1,lista2]
    lista1 = lista2
    lista2 = siguiente elemento lista
Fin para
Fitness = valor
Para i desde 2 hasta número de ciudades hacer
    valor = 0
    lista = siguiente lista
    lista1 = lista
    lista2 = siguiente elemento lista
    Para i desde 1 hasta numero de ciudades – 1 hacer
        valor = valor + matriz[lista1,lista2]
        lista1 = lista2
        lista2 = siguiente elemento lista
    Fin para
    Siguiete fitness = valor
Fin para
    
```

## A3. SELECCIÓN\_ ELIT

---

```

Menor = (Primer fitness)
Para j desde 2 hasta número de ciudad hacer
    Si fitness[j] < menor y no se ha seleccionado hacer
    
```

Menor = fitness[j]

**Fin si**

**Fin para**

elemento = menor

**Para** j desde 1 **hasta** número de ciudades **hacer**

Nuevo individuo = nuevo individuo + elemento

elemento = siguiente elemento

**Fin para**

#### A4. SELECCIÓN\_MUE.

**Para** j desde 1 **hasta** tamaño poblacion **hacer**

porcentaje[j] = 0

individuo = lista

**Para** j desde 1 **hasta** tamaño poblacion **hacer**

porcentaje[j] = porcentaje (individuo)

primero = valor aleatorio (100)

cuantos = 100 / generar

**Para** j desde 1 **hasta** generar **hacer**

**Si** primero <= porcentaje[1] **entonces**

elemento = lista

**Para** k desde 1 **hasta** numero de ciudade **hacer**

Individuo = individuo + elemento

elemento = siguiente elemento

**Fin para**

primero = primero + cuantos

**Fin si**

**Sino**

siguiente = 2

**Mientras** primero > porcentaje [siguiente] **hacer**

siguiente = siguiente + 1

**Fin mientras**

elemento = lista

**Para** k desde 1 **hasta** numero de ciudades **hacer**

Individuo = individuo + elemento

elemento = siguiente elemento

**Fin para**

primero = primero + cuantos

**Fin sino**

**Si** primero > 100 **entonces**

primero = primero - 100

**Fin para**

#### A5. CRUZAR INDIVIDUO.

---

x = valor al azar (número de ciudades)

i = 1

**Mientras** Individuo1[i] <> x **hacer**

i = i + 1

j = 1

**Mientras** Individuo2[j] <> x **hacer**

j = j + 1

nuevo individuo = x

i = i - 1

j = j + 1

**Mientras** (individuo1 > 0 o individuo2 <= número de ciudades) **hacer**

**Si** Individuo1[i] no se encuentra en nuevo individuo **entonces**

nuevo individuo = individuo1[i] + nuevo individuo



**Si  $i > 0$  entonces**

$i = i - 1$

**Si Individuo2[j] no se encuentra en nuevo individuo entonces**

nuevo individuo = nuevo individuo + individuo2[j]

**Si  $j < \text{numero de ciudades}$  entonces**

$j = j + 1$

**Fin mientras**

**Para j desde 1 hasta numero de ciudades hacer**

**Si individuo1[j] no esta en nuevo individuo entonces**

Nuevo individuo = nuevo individuo + individuo1[j]

#### A6. MUTAR

---

x = numero al azar (tamaño de población)

individuo = lista

**Para j desde 1 hasta x hacer**

individuo = siguiente individuo

x = valor al azar (número de ciudades)

elemento1 = individuo

**Para j desde 1 hasta x hacer**

elemento1 = siguiente (elemento1)

x = valor al azar (número de ciudades)

elemento2 = individuo

**Mientras valor de elemento2  $\neq$  x hacer**

elemento2 = siguiente (elemento2)

respaldo = elemento1

elemento1 = elemento2

elemento2 = Respaldo

**A7. LIMPIAR**

---

Índividuo1 = lista

**Para** j desde 1 **hasta** tamaño de la población **hacer**

    individuo2 = siguiente individuo1

    elemento = individuo1

**Para** i desde 1 **hasta** número de ciudades **hacer**

        individuo1 = siguiente elemento

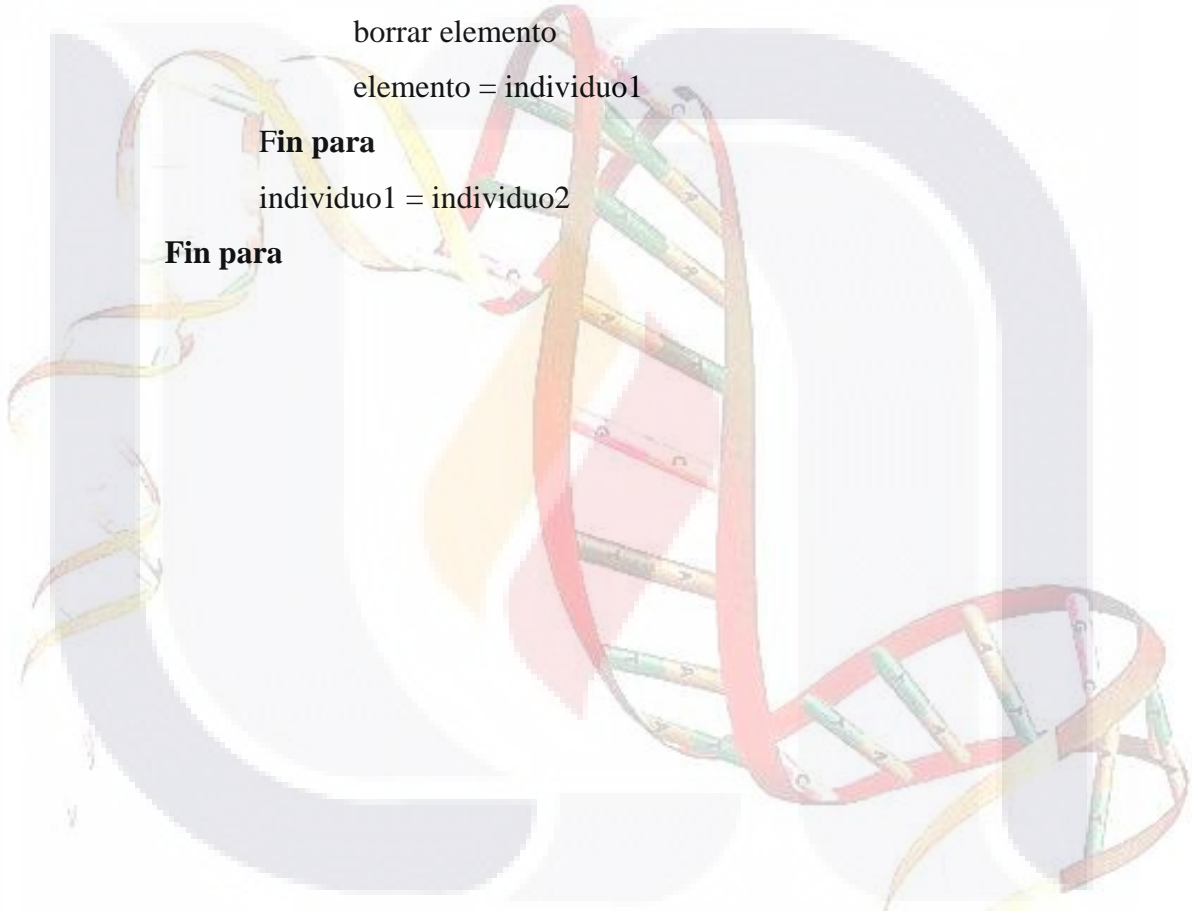
        borrar elemento

        elemento = individuo1

**Fin para**

    individuo1 = individuo2

**Fin para**



## ANEXO B

### ARTÍCULOS

---

#### B1. MICAI '09

---

Genetic Algorithm Using Migration and Modified GSX as Support in the Diversification of Populations.

Francisco Ornelas, Alejandro Padilla, Alberto Ochoa, Eunice Ponce de León, Felipe Padilla

**Abstract.** This research presents a study of some of the operators that have been used in recent years to diversify populations during the evolutionary process of genetic algorithms. This allows us to determine the migration and proper selection and crossing, significantly enhance the quality of solutions obtained, because it prevents premature convergence, resulting from the loss of changes in the genetic material of individuals by inbreeding between them and significantly prevents the migration.

**Keywords.** Genetic Algorithms, Migration, GSX,

#### 1 Introduction

Genetic Algorithms (GA) are search procedures based on natural selection and genetics, which have proven successful in various applications involving search, optimization and machine learning [1], [2] and its use has been on increasing in both academics and industry [6].

The (GA) working with populations of individuals (possible solutions) that compete to preserve their characteristics from generation to generation. The preservation process is given by some operators who are: evaluation, selection, crossover and mutation.

The evaluation is to qualify each individual according to how well solves the proposed problem. That is, if it comes to maximizing a function with a higher score qualify individuals

who obtain a higher result, while minimizing it in the case of a higher score is given to individuals who receive fewer results.

The selection is a process that involves selecting a number of individuals (usually defined depending on the type of problem and the number of individuals) that best solve the problem to be solved (this type of selection is called elitist selection, but there other types such as roulette, tournament, etc.).

The cross is the process which combines the genetic material of two individuals selected (usually from among the best) and usually occurs one or more new individuals that are expected to better solve the problem. There are several different types of crosses that are also used based on the characteristics of the problem, among which crosses a point, multi-point crosses, GSX, among others.

Finally, in the simple GA have the mutation that is a process through which it converts part of the genetic material of one or more individuals that results in new solutions that can enable us to explore different areas of our search space, but not necessarily produce better results than those already obtained.

Unfortunately, this operator is recommended to be used in very small percentages [1], because it is a highly random and does not assess whether the solution to be modified is one of the best results achieved so far or the solution in which turn is better quality.

This paper intends to show some evaluation results to operators which are related to the diversification of populations and migration operator (which is implemented in some models GA) to propose some improvements to this process and increase the performance of the GA.

## **2 Assessment and Selection in the GA**

The simplicity of the GA allows it to be easily adapted to almost any problem. The most difficult thing is to create an evaluation function for taking into account all factors involved to solve the problem and to represent them through a function.

The evaluation function is central to the functioning of the GA.

Once you define a good evaluation function is a selection of individuals who were part of the next generation and will have opportunity to cross for part of their genetic material is preserved.

To make the selection process there are several models proposed (we assess three that are elitism, roulette wheel and stochastic universal sampling), the best known is the elitist, that consists in selecting the best individuals of each generation.

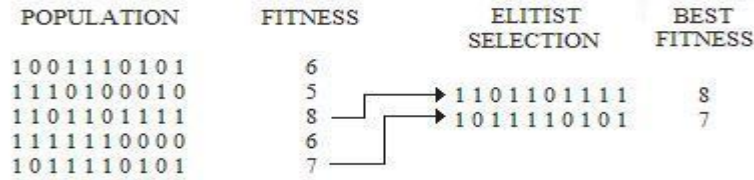


Image 1 Elitist selection.

The main problem we encountered in our research with this type of selection is the fact that some populations are so-called super individuals taking full control of the population and its quality is higher than the rest of the population (though not necessarily the best solution to find), which causes all individuals quickly begin to be part of the genetic material of this super individual that results in rapid convergence (typically a local optimum).

Another type of selection is the method of roulette that is allocated within an imaginary roulette each individual area of population is proportional to the quality with which solves the problem, so individuals get better results in the evaluation function will have a greater area within the spinner. Once this is set to 100% of roulette random values are obtained they determined that individuals are selected.

The selection by this method allows greater diversification than elitist method even though the fittest individuals have a greater chance of being selected.

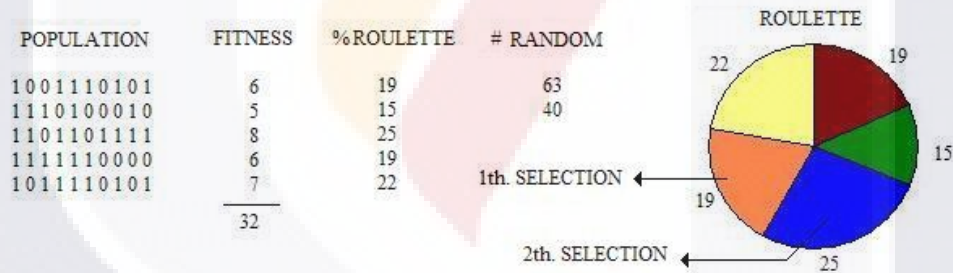


Image 2 Selection by the method of roulette

Finally the stochastic universal sampling selection [4] [5], is as in the previous selection in generating an imaginary roulette represent based on the area assigned to each individual's quality that it solves the problem, but main the main difference is that only the first individual is randomly selected and the rest are selected by dividing the wheel between the number of missing individuals which allows for greater uniformity in the selection and accepting both good and bad individuals.



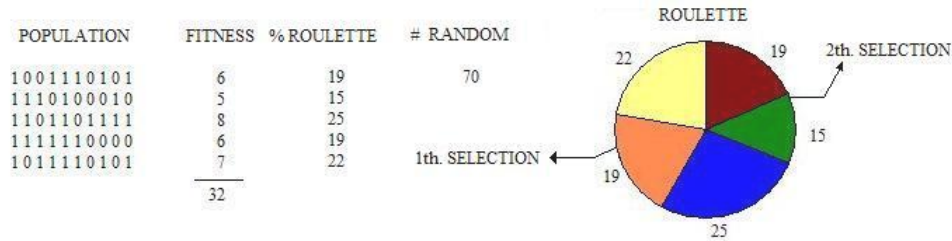


Image 3 Stochastic universal sampling selection.

Based on the study we did make use of an elitist approach to select not more than 10% of individuals to pass on to the next generation (and not lose good solutions that have thus far), and supplement it with other individuals obtained through a uniform universal sample selection (to provide greater diversification).

### 3 Cross Operator

As in the selection operator, the operator of crosses has different models that are used according to the characteristics of the problem and the type of representation with individuals (binary, alphabetic, alphanumeric, etc.).

To make the crosses are taken two individuals at random from the previously selected set of solutions in the previous process.

The most simple and well across is the crossing of a point. The new solution will consist of the elements containing one of the selected individuals (parents) from the start of it to a position (usually chosen at random) as a reference and take the other parent from this reference point until the end of same. This type of cross I may produce one or more new individuals taking the various parts of the parent solutions.

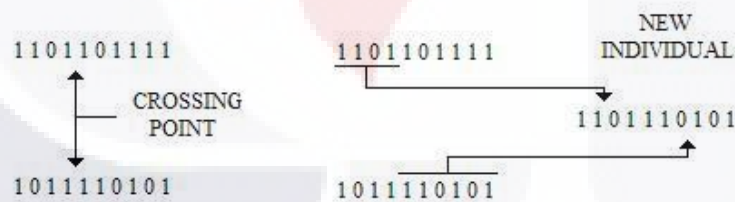


Image 4 Crosses from one point

The other type of crosses that were evaluated was the GSX (Greedy subtour Crossover) [6], which consists of randomly selecting a common element of both parents, and this element becomes part of the new individual. As a next step element is shifted to the left of the first parent and to the right of the second, and assess if current element is not already contained in the new individual, this process continues until some element and the content on the new individual or until no more elements are in one or both parents. Finally completed the new individual with the other elements that are not yet in the same either randomly or based on one of the parents and are inserted as they appear in it.

This type of cross subtours preserves that some problems are of vital importance such as the traveling salesman problem TSP.

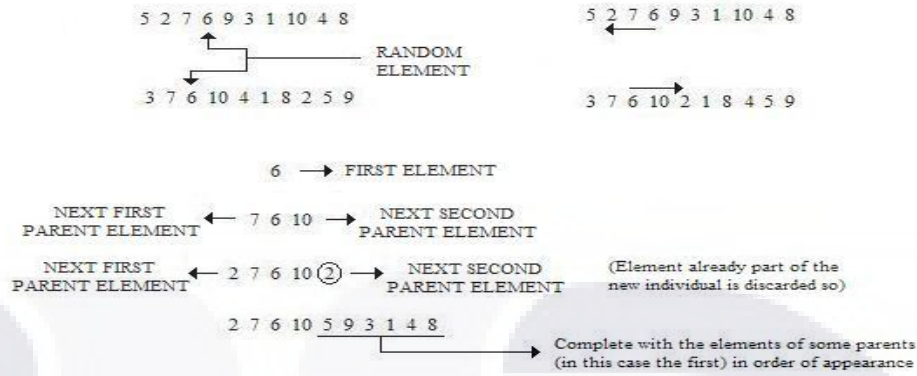


Image 5 GSX

In our research we believe that the GSX is a very important tool that has demonstrated excellent performance. However, a problem that we see is that sometimes the initial element is at the end of the father that causes almost immediately stop the construction of new individual and end up using a random process that occurs in most cases very different individual parents or used as a base to one parent, in which case the new individual would be almost identical to this one.

We propose a modification in the operation of the GSX and described below.

#### 4 GSX Modified (GSXM)

As discussed above the GSX is a great tool for generating new individuals by crossing it allows parties retain complete genetic code from both parents. Unfortunately sometimes the way he works the GSX these pieces of code are very small and you end up with very different individuals parents (losing the desirable characteristics of the goods) or too equal to one parent (which also is desirable).

In nature, individuals are composed of equal parts of both parents' genetic code, which is why we try to modify the GSX to treat it as close as possible to this model. To this end we determined that the first thing we should change is that at least we had the chance to get half of each parent. For these reasons we select the most near the middle of one parent and evaluated against the other parent route also from closer to the middle element of it in the opposite direction as it will cross to begin forming the new individual, this until we find an equality between elements.

Once you have determined the item on the launch of the cross is the same procedure as in the GSX with two main variants. The first is that this process does not stop until it is reaches the end of one or both parents and if at this point there are still elements missing in the new

individual, these are taken from one parent in the order they appear therein. The second is closely related to the first so that if an element is repeated this simply is not taken into account and continues into the following elements.

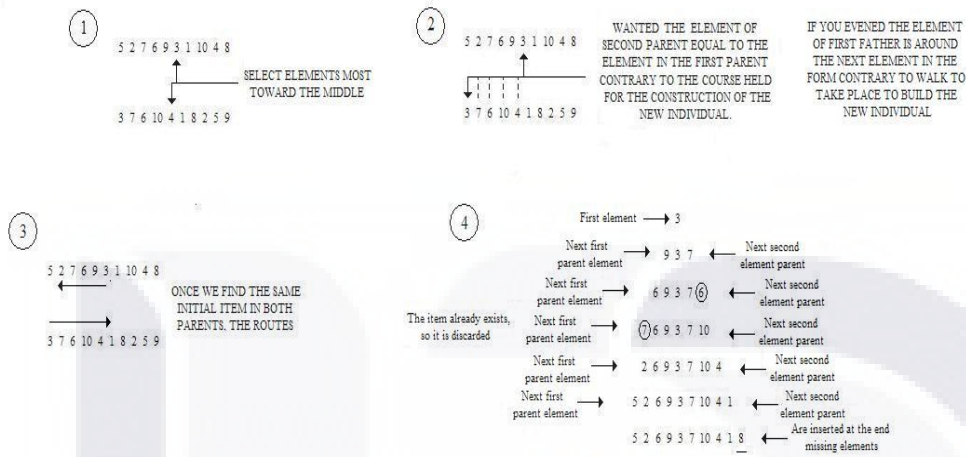


Image 6 GSXM

This change allows us to maintain subtour larger and larger number of subtour occasions that problems like the TSP is important.

## 5 Migration

Another operator we use in our research is migration.

Migration is a concept used in population genetics that is the area of genetics that deals with the study of the evolution of populations in nature and the factors involved in this process.

There is some work done with various proposals on the use of this operator in GA, one of them is the island model of dividing the initial population in subpopulations that evolve in parallel and each given number of generations send a copy of their best individuals of other stocks, that on certain criteria, some of which are similar to network topologies. Examples of such criteria are [4]: communication in star, which has a central population receives the best individuals from other populations and feedback with their best individuals, communication in mesh, which allows all people share their best individuals and communication ring, where people with the following individuals share with the people who have communication and receive feedback from the previous population in the connection.

The model we're using consists of two populations evolving in parallel. One is the main population that is receiving the best individuals of the population and in turn the secondary school population updates its population by introducing new individuals.

The migration process is held every nine generations, this so that individuals are completely discarded or strengthen their genetic material within the population.



We do not make the exchange of genetic material between the two populations based on our experiences because we have found that if a super individual not only takes over the population of which is generated, but to be the best evaluated moves to the other population and also takes control of it and even though the type of evaluation and selection we use largely avoids this situation if you can get to take control of it.

The new individuals enter in the second population can not stop the evolution of populations on search sites which include, but is followed by an exploration of it.

The migration rate we use is of a maximum of 10% to ensure that new individuals only for improvements on the genetic material without removing the original characteristics of the population.

## 6 Findings

In this paper we present some recommendations on the contractors carrying out the process of diversification in the evolutionary process in GA.

First, based on our research we found that an elitist selection causes a rapid loss of genetic diversity, because the better or best individuals take control in the evolutionary process and the operator of mutation is not sufficient to avoid this situation, what we use elitism in a very small amount and the complement to a process of selection and stochastic universal sampling has yielded good results.

It has also been found that the mutation operator is necessary but must be supplemented to improve genetic diversity. From the above we use the migration operator to comply with this process.

As part of research is developing a tool that meets all the characteristics mentioned above for verification if together give the same good results that modifications or improvements separately.

## Bibliography

[1]Goldberg, D. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley. 1989.

[2]Goldberg, D. The Design of Innovation: Lessons from and for Competent Genetic Algorithms. Kluwer Academic Publishers. 2002.

[3]Holland, J. Adaptation in Natural and Artificial Systems. The MIT Press. 1975

[4] Moujahid A., Inza I., Larrañaga P. Tema 2: Algoritmos Genéticos. Universidad del País Vasco. <http://www.sc.ehu.es/isg/>

[5] Gonzalez U., Solanas. A. Proteccion de la Privacidad mediante Microagregación Multivariante basada en Algoritmos Genéticos: Selección por Ruleta vs. Selección Uniforme

[6] Winward P. Fluctuating Crosstalk, GA Scalability, and the Disruption of Optima. University of Illinois at Urbana-Champaign. 2007

**LA MIGRACIÓN COMO PARTE DEL PROCESO EVOLUTIVO EN ALGORITMOS  
GENÉTICOS**

F. ORNELAS, A. PADILLA, F. PADILLA, E. PONCE, A. OCHOA

**Abstract**

Las herramientas que proporciona la Computación Evolutiva (CE) han demostrado ser de gran ayuda en la resolución de problemas con explosión combinatoria. Por su parte los Algoritmos Genéticos (AG) son de gran utilidad por su facilidad para adaptarlos a casi cualquier problema de forma sencilla. Sin embargo algunas otras herramientas obtienen resultados de mejor calidad debido a que los AG convergen prematuramente a óptimos locales. El presente trabajo muestra un modelo de AG, al que se agrega el operador de Migración para mejorar la diversificación de las poblaciones y con ello obtener resultados de mejor calidad. Por último se muestran los resultados obtenidos de una herramienta de software que se desarrollo para comparar un AG básico contra el mismo adicionando la Migración.

**Palabras clave.**

Algoritmos Genéticos, TSP, Mutación, Migración, Greedy Subtour Crossover, Genética de Poblaciones.

**Introducción.**

Desde que el hombre apareció en la Tierra ha buscado herramientas que le permitan facilitar sus tareas diarias y/o facilitar la resolución de los problemas que se le presentan. Derivado de ello se han imitado muchos de los procesos naturales que el hombre ha observado a lo largo de la historia.

La Inteligencia Artificial (IA) surge en el momento que se trataron de crear artefactos que simularan el intelecto humano. La IA se divide en varias áreas dentro de las cuales se tienen a los Agentes Inteligentes, Sistemas Clasificadores, Redes Neuronales, Computación Evolutiva, etc



La Computación Evolutiva por su parte agrupa técnicas basadas en la Evolución Natural propuesta en los trabajos realizados por Charles Darwin y en los descubrimientos realizados por Gregor Mendel en el campo de la Genética. Una de estas técnicas es Algoritmos Genéticos (AG) que son herramientas heurísticas que utilizan técnicas de búsqueda y fueron propuestos en los años 60's por John Holland.

En su forma más básica el AG genera una población inicial aleatoriamente formada por posibles soluciones (individuos) de una función objetivo (problema a resolver) y sobre las que se aplican los operadores de selección, cruza y mutación para evolucionarlas ha soluciones normalmente de mejor calidad [2][5].

Los AG han demostrado ser una muy buena herramienta, ya que es fácil adaptarlos para resolver casi cualquier problema de optimización combinatoria por su sencillez, pero comúnmente convergen prematuramente a óptimos locales debido a que el proceso de selección y cruza provocan que conforme se evoluciona, los individuos que forman parte de la población cada ves se parezcan mas entre si al no existir nuevo “material genético” que permita liberar a la población de áreas muy específicas del espacio de búsqueda.

La única herramienta con la que cuentan los AG para diversificar las características de los individuos y lograr saltar a otras áreas del espacio de búsqueda es el operador de mutación que permite la modificación de parte del material genético. Desafortunadamente este operador se recomienda utilizar en cantidades muy pequeñas ya que al ser un proceso aleatorio puede afectar no solamente a las malas soluciones sino también a las soluciones de buena calidad a las que se haya llegado [2].

### **Algoritmos Genéticos.**

Los AG están basados en la Teoría de la Evolución de las Especies propuesta por Charles Darwin y en los trabajos realizados en Genética por Gregor Mendel.

La Genética de Poblaciones es un área de la Biología que proporciona los “principios teóricos de la evolución” [1]. Según dichos principios, los factores que intervienen en el proceso evolutivo son: la Selección Natural, la cruza, la mutación, la migración y la deriva genética.

De la esencia de los tres primeros factores se desarrollaron los operadores que participan en el AG básico, desafortunadamente el factor de Migración no se aplica en este aun cuando en la naturaleza es parte del proceso evolutivo y apoya a la mutación para generar la diversificación necesaria para que el proceso evolutivo no se detenga.

A continuación se muestran los elementos que forman parte del AG básico.

*Individuo:* Es una posible solución del problema a resolver. Esta formado por elementos que pueden ser de tipo binario, numérico o alfanumérico.

*Población:* Es un conjunto de individuos (posibles soluciones. La población inicial se genera de forma aleatoria.



Fig.1 Representación del individuo y la Población.

*Función objetivo:* Es la función que se pretende resolver. Con esta función se evalúa cada uno de los individuos y al resultado se le llama fitness.

Los operadores que se aplican para evolucionar los individuos a soluciones de mejor calidad son:

*Selección:* El tipo de selección que normalmente se utiliza es la elitista, donde se selecciona basado en el fitness a los individuos que mejor resultado obtuvieron. Por ejemplo si lo que se pretende es minimizar una función los individuos seleccionados serán aquellos que obtuvieron el menor fitness y viceversa.

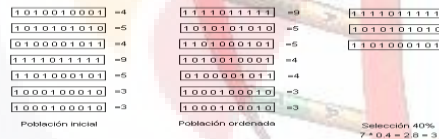


Fig.2 Proceso de selección.

*Cruza:* La cruza que mas se utiliza en el AG básico es la cruza de un punto en la que se toma a dos individuos de los que en la etapa anterior se seleccionaron y se busca aleatoriamente un punto al que se llama punto de cruza. Se toma del primer individuo desde el inicio del mismo hasta el punto de cruza y del segundo desde el punto de cruza hasta el final del mismo y con estas dos partes se genera un nuevo individuo.

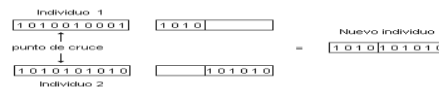


Fig.3 Proceso de cruza de un punto

*Mutación:* La mutación es un proceso en el que se toma aleatoriamente un elemento de un individuo (también tomado al azar) y se modifica (en caso de ser un elemento binario se cambia de 0 a 1 y viceversa). En caso de que sea algún otro tipo de dato normalmente se hace el intercambio entre dos elementos del individuo esto también de forma aleatoria. Se recomienda utilizar este proceso en porcentajes muy pequeños ya que al ser un proceso aleatorio, los buenos resultados obtenidos, no están exentos de que este proceso los afecte.



Fig.4 Proceso de mutación.

Existen algunos modelos que han utilizado la migración como parte de su proceso evolutivo. Algunos de estos son los AG paralelos y los Algoritmos de Grano Grueso y Fino.

Por lo anterior se propuso un modelo al que se introdujo al AG básico el operador de migración como un operador que participara en el proceso de diversificación. Para ello se genera una población paralela que evoluciona de manera independiente y cada cierto número de generaciones migra individuos a la población principal. Este modelo se utiliza para resolver instancias del TSP (Travelling Salesman Problem).

También se generó una herramienta de software que nos permite comparar el modelo propuesto contra el AG básico al resolver instancias del TSP, para identificar si se mejora la calidad de los resultados obtenidos.

**Trabajo desarrollado.**

Se generó una herramienta de software para identificar si la propuesta mejora la calidad de los resultados obtenidos y en su caso aplicar este operador a AG's más poderosos.

La forma en que trabaja la herramienta de software es la siguiente:

Primero se solicita como parámetro el número de ciudades que participaran en el problema a resolver, basándonos en este parámetro se genera una matriz de costos (dando como distancia un valor aleatorio entre 1 y 100) para cada par de ciudades.

En segundo lugar se genera una población principal que contendrá un número de individuos de aproximadamente un 62% del número de ciudades y una población paralela del mismo tamaño.

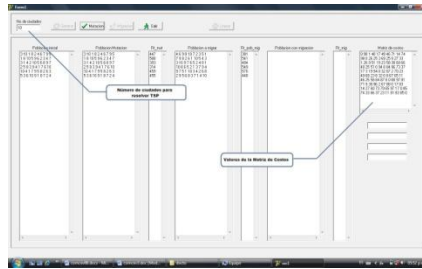


Fig.5 Muestra la captura de número de ciudades y la generación de la matriz de costos.

Se hace un respaldo de de la población inicial para poderla utilizar como entrada tanto en el AG básico como en el modelo propuesto.

Se evalúan todos y cada uno de los individuos de la población y guardamos los resultados en un arreglo. Estos resultados serán el fitness de cada individuo. Como el problema que se quiere resolver es el TSP el objetivo es minimizar el costo de viaje.

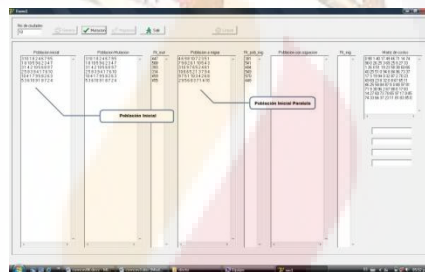


Fig.6 Muestra la población inicial, la paralela, la de respaldo y sus respectivos fitness.

Se seleccionan un 50% de los individuos evaluados utilizando una selección de tipo elitista por lo que los individuos con menor fitness son seleccionados.

Para el AG básico se obtiene el 50% faltante de la población por medio de cruza el tipo de cruza que se utiliza es el Greedy Subtour Crossover (GSX) [4] y su funcionamiento se describe a continuación:

- Se seleccionan dos padres de forma aleatoria.
- Se elige un mismo elemento en ambos padres.
- Este será el primer elemento del nuevo individuo.
- Se hace un recorrido en el primer padre hacia la izquierda y en el segundo hacia la derecha a partir del elemento común.
- Se verifica que cada elemento al que se llegue no forme parte ya del nuevo individuo.



- El proceso de recorrido de los padres es cíclico y finaliza cuando se llegue al inicio o fin del padre (según sea el caso) o cuando algún elemento no sea valido.
- Por ultimo se adicionan al final los elementos que no estén todavía contenidos en el nuevo individuo.

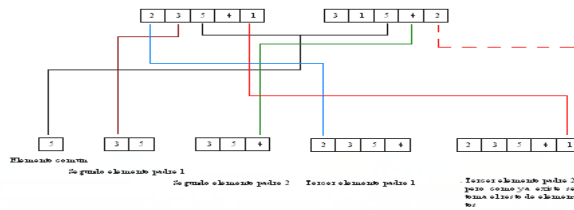


Fig. 7 Proceso de cruce GSX

Este tipo de cruce permite conservar subtours que estén contenidos en los padres seleccionados.

Por último se lleva a cabo el proceso de mutación en donde se selecciona un individuo al azar, de ese individuo se seleccionan 2 elementos del mismo, el primero por el valor del elemento y el segundo por la posición, esto de forma aleatoria y se hace un intercambio entre ambos. Este proceso se hace en el 1% de los elementos, es decir se multiplica el número de ciudades por el número de individuos y el 1% son los intercambios que se realizan.

Este proceso se ejecuta hasta que se llegue a doscientas generaciones o hasta que durante 15 generaciones el mejor resultado obtenido no haya cambiado.

En el AG con Migración el proceso de evaluación, selección y mutación es exactamente el mismo que en el AG básico y durante 8 generaciones el proceso de cruce también es igual. En la novena generación se obtiene solamente el 40% por cruce y el 10% restante se selecciona de la población paralela que ha estado evolucionando también con el proceso del AG básico. Los individuos que se toman de la población paralela son seleccionados de forma elitista.

Este proceso se realiza de forma cíclica y los criterios de parada son los mismos que en el AG básico.

Después de ejecutar ambos procesos sobre una misma población inicial, se obtiene el mejor fitness por cada uno de dichos procesos.



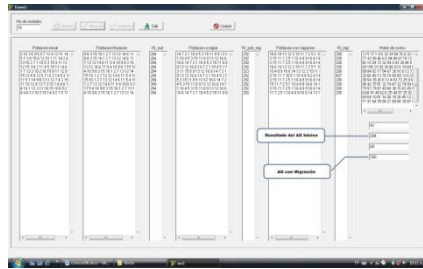


Fig.8 Muestra los resultados de la ejecución

El programa funciona con parámetros de hasta 1000 ciudades y con los parámetros cercanos a este limite el proceso es de aproximadamente 10 minutos.

**Resultados.**

Se realizaron 100 ejecuciones con parámetros de 10, 16, 32, 60, 100, 150, 250, 500, 750 y 1000 ciudades.

En el caso del parámetro igual a 10 ciudades, los algoritmos llegan a resultados diferentes y pero aproximadamente en un 50% de las ocasiones cada uno obtiene mejores resultado. Esto debido a que como el proceso de migración participa solamente como un 10% de la población total, con parámetros menores a 16 esto no sucede.

Con el resto de los parámetros que se utilizaron los resultados obtenidos fueron diferentes casi en el 100% de las ejecuciones.

Con parámetros de 32 y 100 ciudades los mejores resultados obtenidos por el AG básico fueron mayores que en el AG con migración, pero en general los resultados se mejoraron con este último en casi un 70% de los casos.

	10	16	32	60	100	150	250	500	750	1000									
273	304	438	409	546	620	1190	1225	2921	2969	4839	4602	9292	9038	1945	20286	31436	31417	42740	42884
517	443	411	356	640	436	3056	1201	2764	2889	4913	4821	8972	3772	20046	19650	30687	30772	42899	43279
349	393	374	313	410	522	1225	998	2589	2651	4742	4725	9247	3899	20035	19971	31613	31443	43141	43013
326	297	342	439	513	449	1269	1160	3015	2602	4801	4732	9186	9114	20307	20076	31412	31330	42893	42825
234	241	576	245	526	602	1312	1007	2010	2823	4789	4696	9177	8886	19938	19634	31634	31680	43109	42676
459	438	437	359	461	503	1312	1274	2951	2798	4880	4908	9192	9002	20074	19915	31154	31365	43133	42677
282	387	441	396	626	654	1404	1324	3048	2719	4603	4608	9029	9008	20141	20138	31577	31519	43026	42454
267	267	381	389	469	453	1211	1184	2736	2835	4795	4889	9886	9004	20083	19840	31096	31296	42847	43140
309	423	311	316	517	531	1149	1227	2857	2931	4794	4780	9212	8945	20281	19573	31729	31269	43240	42903
324	311	287	350	626	685	1057	1126	2703	2757	4657	4802	8847	9105	19739	19874	31219	31573	43101	42720
5	6	4	6	8	2	4	6	6	4	4	6	2	8	3	7	3	7	3	7

■ Resultados AG básico  
■ Resultados AG con Migración

Tab.1 Tabla de resultados de 100 ejecuciones realizadas con la herramienta de software

### Conclusiones y Trabajo Futuro.

Actualmente se hacen modificaciones a la herramienta de software para probarla con instancias del TSP contenidas en el TSPLIB y basado en los resultados que se obtengan se harán comparaciones con herramientas heurísticas como son: Algoritmos Meméticos y Colonia de Hormigas.

Dependiendo de si los resultados que se obtengan mejoran la calidad de los obtenidos por otras herramientas este modelo se podría aplicar a AG más potentes o a otras herramientas que también forman parte de la CE.

### Bibliografía.

- [1] A. Barbadilla. 'Genética de Poblaciones'. Universidad Autónoma de Barcelona. <http://bioinformatica.uab.es/divulgacio/genpob.html>
- [2] D. Goldberg. 'Genetic Algorithms in Search, Optimization, and Machine Learning'. Addison Wesley. (1989).
- [3] F. Ornelas, A. Padilla, F. Padilla, E. Ponce 'Un Nuevo Operador para Mejorar la Diversificación en Algoritmos Genéticos, Bioinspirado en Genética de Poblaciones'. COMCEV. 2007.
- [4] H. Sengoku & I. Yoshihara. 'A Fast TSP Solver Using GA on JAVA'.
- [5] J. Holland. 'Adaptation in Natural and Artificial Systems'. The University of Michigan Press. (1975).

**UN NUEVO OPERADOR PARA MEJORAR LA DIVERSIFICACIÓN EN ALGORITMOS GENÉTICOS, BIOINSPIRADO EN GENÉTICA DE POBLACIONES.**

**F. Ornelas, A. Padilla, F. Padilla, E. Ponce.**

**Abstract**

Se presenta un modelo en el que se introduce un nuevo operador a los Algoritmos Genéticos, que permite mejorar el proceso de diversificación a la vez que, mejora el rendimiento del mismo. Este nuevo operador esta bioinspirado en un proceso de Genética de Poblaciones llamado *Migración* y se evalúa utilizando el problema del viajante de comercio (TSP).

**Palabras clave.**

Algoritmos Genéticos, TSP, Mutación, Migración, Greedy Subtour Crossover, 2opt, Genética de Poblaciones.

**Introducción.**

Los Algoritmos Genéticos (AG) son herramientas heurísticas que utilizan técnicas de búsqueda y están basadas en la evolución natural y la genética. Forma parte de una de las áreas más prometedoras de la Inteligencia Artificial (IA) conocida como Computación Evolutiva (CE).

Los AG's fueron creados en los años 60 por John Holland y están inspirados en la "Teoría de la evolución de las especies" propuesta por Charles Darwin y los descubrimientos realizados por Gregor Mendel en el campo de la genética.

La forma general en que trabajan los AG's es: generar una población inicial que contiene posibles soluciones (individuos) del problema a resolver, sobre la que se aplican operadores como cruzamiento, selección y mutación para "evolucionarlos" a soluciones normalmente de mejor calidad [1] [2].

Los AG's han demostrado ser una buena herramienta para resolver problemas de explosión combinatoria, ya que logran obtener buenas soluciones en tiempos relativamente cortos. Desafortunadamente su principal problema es que convergen prematuramente a resultados de buena calidad pero no necesariamente el mejor, esto debido a que el operador que utilizan para

diversificar la población debe usarse en porcentajes muy pequeños [1], para que afecte lo menos posible los buenos resultados que se hayan obtenido hasta el momento, ya que este operador funciona de forma aleatoria y puede afectar tanto a buenos como a malos resultados, por lo que al incrementar su porcentaje se gana en diversificación pero se incrementa el riesgo de pérdida de buenos resultado con que se cuente.

Nosotros proponemos un nuevo modelo en el que se adicionará al operador de mutación, un nuevo operador al que llamamos migración y que está inspirado en un proceso utilizado en genética de poblaciones también como diversificador.

### **Genética de Poblaciones.**

“La evolución es ante todo un proceso genético, y la *genética de poblaciones* es la disciplina biológica que suministra los principios teóricos de la evolución.” [4].

Los factores que participan en la evolución son:

*Selección.* “Algunas formas son mas aptas para sobrevivir y reproducirse que otras” [6].

La selección natural conceptualizada por Darwin se centra en el concepto de “la supervivencia de los mas aptos” [4], esto es la conservación de los individuos con mejores características que se transmitirán por medio de la herencia de una generación a otra.

*Mutación.* Es un factor que permite aumentar la diversidad genética de una población. Una mutación es una alteración en el ADN lo que puede generar que un individuo obtenga ventajas o desventajas sobre los demás miembros de la población. En la mayor parte de las ocasiones este cambio genera desventajas pero en los pocos casos en que genera ventajas, esta mutación es heredada a los descendientes de dicho individuo permitiendo conservar las ventajas obtenidas de generación en generación.

*Migración.* Es un intercambio de genes entre poblaciones. La forma en que se realiza dicho intercambio es introduciendo individuos de una población en otra (donde cada población tiene sus características propias), lo que da como consecuencia que al mezclar los diferentes materiales genéticos se genere una mayor diversidad en el material de la nueva población. Este proceso es muy utilizado en la crianza de ganado por algunas razones dentro de las que destacan el hecho de encontrar poblaciones con características deseables en la propia o debido a que en poblaciones cerradas se acaba por incrementar el grado de parentesco entre individuos lo que no es deseable.



*Deriva genética.* Es un factor que se presenta cuando dentro de una población se realiza cruzamiento aleatorio es decir que los gametos que pasaran a los descendientes son aleatorios. Esto puede generar que conforme se evolucione, algunos gametos se pierdan permanentemente. Este factor se presenta con mayor frecuencia en poblaciones pequeñas. La principal consecuencia de la deriva genética es la pérdida de diversificación.

**Algoritmos Genéticos.**

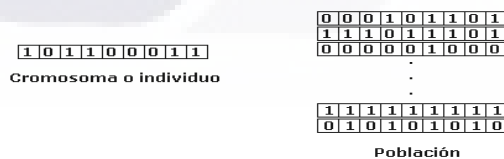
Al igual que en la naturaleza los individuos cuentan con características que les permiten sobrevivir y en algunos casos transmitir dichas características a sus descendientes, además de que algunas de estas características se modifican en el transcurso de los años permitiendo a los individuos ser más competitivo en su entorno, de la misma manera los AG's intentan que los individuos o posibles soluciones evolucionen hasta alcanzar un alto grado de competitividad lo que resultaría en una buena solución al problema que se desea resolver [1][2].

Las principales características de los AG's son:

*Población:* Es el espacio en el que se encuentran las posibles soluciones al problema que se pretende resolver. Por esta razón se generan individuos que contengan la información necesaria para resolver el problema dado.

*Cromosoma:* Cada uno de los individuos que pertenece a la población está formado por una cadena ya sea de datos binarios, reales o alfanuméricos. A cada una de estas cadenas se le llama individuo o cromosoma.

Para iniciar el AG se requiere generar una población inicial (aleatoriamente), sobre la que se aplicaran operadores que favorezcan la permanencia de los mejores individuos y los menos capaces desaparezcan. El tamaño de la población se define en base al problema a resolver así como también el tamaño del cromosoma.



Las siguientes poblaciones estarán formadas por los individuos mas aptos (mejores soluciones) de la generación anterior y nuevos individuos generados a partir del cruzamiento de éstos. El número de individuos que pasan de una generación a otra, así como el número de nuevos individuos es un parámetro que hay que definir desde el inicio del AG.



*Grado de adaptación:* Así como en la naturaleza, los individuos más aptos son los que sobreviven, de la misma forma en los AG's cada individuo cuenta con características que pueden ser evaluadas y definir su permanencia dentro de la población y generar descendencia. Para determinar que individuos son los más aptos se requiere calificar a cada uno de los miembros de la población. Esta calificación sería su grado de adaptación. En AG este grado de adaptación es llamado *fitness* y se evalúa en base a una *función objetivo* que califica todas y cada una las características que influyen en la solución del problema a resolver.

*Selección:* Después de haber calificado a cada individuo de la población, existen varias formas de determinar cuales se conservaran en la siguiente generación y serán los progenitores de los nuevos individuos. La forma que utiliza el AG clásico es el *elitismo*, que selecciona los individuos que mejor califican para resolver el problema dado, hasta cubrir el número necesario de individuos que pasará a la siguiente generación.

*Cruza:* En la naturaleza no solo es importante que se existan individuos aptos, sino que los nuevos individuos conserven las cualidades de los anteriores. Dentro de los AG sucede algo similar, ya que solamente los individuos que sean seleccionados como más aptos podrán cruzarse y generar soluciones que contendrán parte del material genético de sus padres a la vez que posibilita el hecho de que tengan mejores características que las de sus progenitores lo que traerá como consecuencia una mejor solución final al problema.

*Mutación:* Muy pocas veces la naturaleza se equivoca, pero cuando sucede alguno de estos errores, el material genético de un individuo es modificado, produciendo un cambio que puede traerle ventajas o desventajas sobre los demás individuos de su especie.

En los AG's la mutación se utiliza para generar un cambio en algún o algunos individuos que permita explorar zonas del espacio de búsqueda que no han sido explorados hasta el momento. La mutación funciona dentro del AG como una herramienta de diversificación.

Dentro del trabajo realizado por David Goldberg [1], se recomienda utilizar porcentajes de mutación muy pequeños ya que en porcentajes altos puede producir un efecto contrario y hacer que el algoritmo pierda soluciones de buena calidad.

Estos son los elementos de un AG básico, pero existen muchas otras variantes, además de híbridos que utilizan las mejores cualidades de AG para resolver problemas de optimización.

### **Modelo propuesto y trabajo realizado.**

Los AG's como se comentó anteriormente, están inspirados en la teoría de la Evolución de las Especies y los trabajos realizados en genética por Mendel, así mismo se comentó que el

principal problema de los AG's es la convergencia prematura a óptimos locales debido a la falta de diversificación necesaria. Por esta razón y tomando en cuenta algunos trabajos realizados en Genética de Poblaciones, proponemos un modelo que toma como base el AG clásico e introduce un operador existente dentro del proceso evolutivo que apoya al operador de mutación en la diversificación de las poblaciones. Este operador es llamado *Migración* y sus características y funciones las comentamos anteriormente.

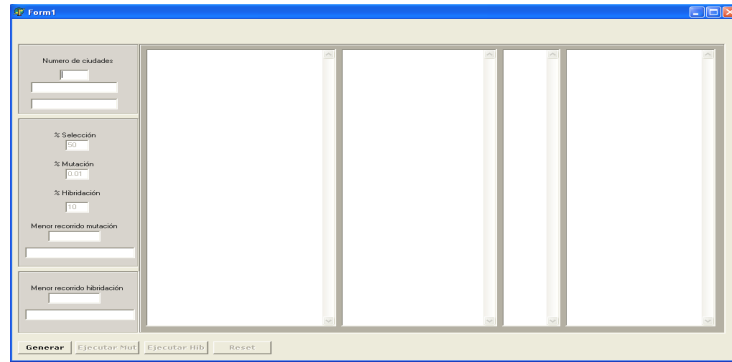
Dentro de los AG's conforme se avanza en el número de generaciones, los individuos tienden a presentar cada vez mayor parecido entre sí (pérdida de diversidad). Por esta razón se necesita mejorar sus mecanismos de diversificación.

El modelo propuesto adiciona el proceso de migración introduciendo individuos con características “frescas”, además antes de que participen en la creación de nuevos individuos son evaluadas sus características al igual que las de los individuos conservados de la generación anterior y compiten por su supervivencia.

La forma en que trabaja el nuevo modelo es:

- Generar una población inicial, de forma aleatoria.
- Se evalúa cada uno de los individuos de la población (posibles soluciones).
- Se seleccionan de forma elitista los individuos cuyo costo del viaje sea más pequeño, hasta tener un 50% que formaran parte de la nueva población la cual deberá tener al final el mismo número de individuos que la población inicial.
- Se generan nuevos individuos en base a la cruce de los individuos que fueron seleccionados en la etapa anterior, esto hasta cubrir al menos el 40% del tamaño de la población inicial.
- Se generan aleatoriamente los individuos necesarios para completar el tamaño de la población.
- Se efectúa el proceso de mutación.
- Se hace un ciclo que regresa al segundo paso hasta que se llegue a cumplir un criterio de parada.

Se desarrollo una herramienta de software para verificar el comportamiento del nuevo modelo en comparación con el AG clásico.



Herramienta de Software para resolver el TSP utilizando un AG clásico y un nuevo modelo de AG utilizando *Migración*.

Esta herramienta resuelve el problema del TSP. Las características de dicha herramienta se describen a continuación.

La herramienta trabaja con grafos planos, completos no dirigidos, se generan aleatoriamente las distancias entre ciudades. El número de ciudades es definido por el usuario hasta un máximo de 30.

Primero se genera una población inicial de 30 individuos con n número de ciudades (n definido por el usuario y menor o igual a 30), así como una matriz de distancias entre ciudades, ambas de forma aleatoria.

2	3	5	4	1
2	4	1	5	3
4	2	1	5	3
3	1	5	4	2

0	12	15	11	5
12	0	5	6	20
15	5	0	5	28
11	6	5	0	16
5	20	28	16	0

Poblacion inicial de 30 individuos y 5 ciudades. Matriz de distancias ciudades.

La población inicial se utiliza para ser resuelta por el AG clásico y por el nuevo modelo.

La función objetivo que se utiliza para evaluar los individuos tanto en el AG como en el modelo propuesto es el costo mínimo del viaje, que se obtiene sumando la distancia entre las ciudades relacionadas, de acuerdo al orden en que se encuentran y agregando la distancia entre la última y primera ciudades.

2	3	5	4	1

Tomando el primer individuo y la matriz de distancias anteriores el costo es 72.

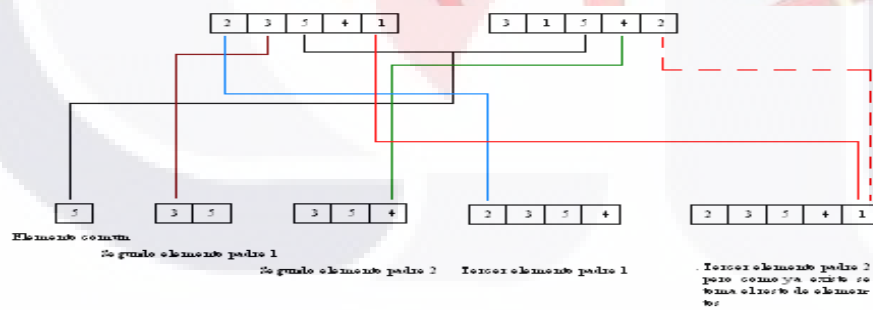
Después de la evaluación se realiza la selección por el método elitista, tomando los individuos que calificaron con menores costos del viaje, hasta cubrir el 50% de la nueva población.

Por medio de cruzamiento se generan el 50% restante de la población para el AG clásico y el 40% para el nuevo modelo tomando como base el tamaño total de la población inicial (100%).

El tipo de cruce utilizada es la propuesta por Sengoku y Yoshihara [2] llamada Greedy Subtour Crossover (GSX).

La forma en que trabaja el GSX es:

- Se seleccionan dos padres de forma aleatoria.
- Se elige un mismo elemento en ambos padres.
- El elemento elegido es el primer miembro del nuevo individuo.
- Se hace un recorrido en el primer padre hacia la izquierda y en el segundo hacia la derecha a partir del elemento común.
- Se verifica cada elemento y son válidos siempre y cuando no estén contenidos ya en el nuevo individuo.
- El proceso de recorrido de los padres es cíclico y se detiene cuando no se pueda seguir recorriendo alguno de los padres o cuando algún elemento no sea válido.
- Por último se adicionan al final los elementos que no estén todavía contenidos en el nuevo individuo.



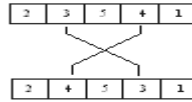
Este tipo de cruce permite conservar subtours que estén contenidos en los padres seleccionados.

El último paso dentro del AG clásico es la mutación. El método que utilizamos fue el 2opt. Este proceso se describe a continuación:

- Se selecciona un individuo.



- Se selecciona un elemento, así como una posición dentro del individuo seleccionado.
- Se hace el intercambio entre el elemento seleccionado y el elemento que se encuentra en la posición seleccionada.



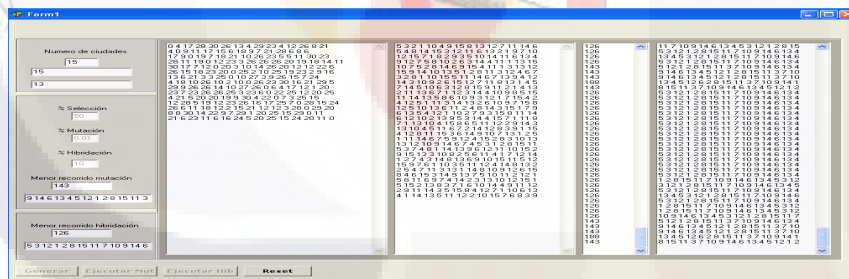
Esta figura muestra el proceso de mutación.

Este proceso se realiza en un 1% del número de elementos contenidos en la población. Esto es el número de individuos (en el software desarrollado 30) multiplicado por el número de ciudades.

En el nuevo modelo se genera el 10% restante de la población insertando nuevos individuos generados aleatoriamente, y se aplica a la población el proceso de mutación de la misma forma que en el AG clásico.

Estos procesos se realizan de forma cíclica hasta llegar a un criterio de parada que nosotros definimos en 1000 generaciones o al llegar a 10 generaciones en las que no se encuentre ningún individuo que mejore al que obtuvo el menor costo.

En la figura siguiente se muestra una ejecución para 15 ciudades:



Se realizaron 120 corridas al software con número de ciudades igual a 5, 10, 15, 20, 25 y 30.

### Resultados.

Los eventos que se tuvieron fueron 120 ejecuciones por el software (20 para cada número de ciudades utilizado como parámetro de entrada), se encontró que si se utilizan 5 ciudades, los resultados tanto del AG clásico como del nuevo modelo son los mismos.



A partir de las ejecuciones con parámetro de entrada igual a 10 ciudades se obtuvo una mejora significativa, aún cuando no fue el esperado, ya que la mejora consiste en que aproximadamente en el 99% de las corridas, el número de generaciones utilizadas por el modelo propuesto, fueron menores a las necesarias en el AG clásico, llegando en un 83% de las ocasiones al mismo resultado, un 13% a mejores resultados y un 4% a peores.

	5	10	15	20	25	30						
1	11 36	11 36	14 90	15 95	17 129	12 129	16 183	12 186	32 195	12 195	20 275	12 275
2	11 68	11 68	14 90	12 90	14 139	12 139	16 162	12 162	18 234	12 234	40 189	12 189
3	11 40	11 40	16 65	12 65	16 118	12 118	19 168	12 168	22 196	12 196	27 210	12 210
4	11 52	11 52	13 100	12 100	17 135	12 135	36 169	12 169	20 211	12 211	20 218	12 218
5	11 52	11 52	14 59	12 59	16 137	12 137	24 135	12 136	23 195	12 196	24 210	12 210
6	11 62	11 62	14 85	12 85	20 108	12 105	20 133	12 133	15 238	12 238	24 210	12 210
7	11 53	11 53	12 117	13 95	25 117	12 117	18 177	12 177	20 175	12 175	18 247	12 247
8	11 64	11 64	16 60	12 60	14 100	12 100	17 140	12 140	16 197	12 203	27 212	12 212
9	11 51	11 51	14 87	12 87	15 112	12 112	15 161	12 161	33 173	12 173	20 218	12 218
10	11 69	11 69	14 94	12 94	19 124	12 124	16 156	12 156	20 180	12 180	29 208	12 208
11	11 59	11 59	13 88	12 88	16 125	12 125	19 169	12 169	14 220	12 220	21 201	12 201
12	11 66	11 66	16 91	12 91	14 111	13 105	21 180	12 180	19 250	12 250	16 262	12 262
13	11 58	11 58	13 84	13 75	13 110	12 110	18 146	12 146	17 172	12 172	21 226	12 226
14	11 68	11 68	13 80	12 80	15 99	12 99	14 127	12 127	26 180	12 180	23 236	12 236
15	11 53	11 53	16 89	12 89	16 141	12 141	23 132	12 132	34 183	12 183	16 233	12 233
16	11 54	11 54	13 70	12 70	13 108	12 108	19 148	12 148	18 231	12 231	20 212	12 212
17	11 56	11 56	14 89	12 89	17 146	12 146	14 133	12 133	17 232	13 203	21 253	12 253
18	11 53	11 53	13 101	12 101	16 157	13 116	13 197	12 197	22 192	12 192	19 199	12 199
19	11 51	11 51	13 45	12 45	16 94	12 94	18 126	12 126	17 208	12 208	21 212	12 212
20	11 75	11 75	14 62	12 62	17 95	12 95	16 189	12 189	26 178	12 178	19 212	12 212

Tabla que muestra los resultados de las 120 ejecuciones, marcadas con verde las ejecuciones donde se mejoraron los resultados y de rojo aquellas en las que se empeoraron.

### 5 Conclusiones y trabajo futuro.

Se considera que se tienen avances significativos, ya que ésta propuesta obtuvo resultados aceptables comparables con el AG clásico, por lo que dicha modificación podría hacerse usando un AG mas avanzado que cuente actualmente con un mejor rendimiento que el AG clásico lo que generaría AG más competitivos.

Aún cuando el objetivo de esta propuesta era mejorar las soluciones (demostrado con el software propuesto), encontramos otra opción de mejora, ya que al llegar a los mismos resultados pero en un menor número de iteraciones, esto produciría un AG más competitivo contra otras herramientas heurísticas que se caracterizan por su rapidez.

Dentro del trabajo futuro se tiene contemplado continuar con dicha investigación sobre esta propuesta, además de una mejora al software para que funcione con un mayor número de ciudades y con grafos no necesariamente completos, ya que esto nos permitiría verificar el funcionamiento de nuestro modelo, utilizando benchmarks de algoritmos y modelos que resuelven el problema del TSP lo cual le daría mas validez a nuestra investigación

### Referencias.

[1] D. Goldberg. ‘Genetic Algorithms in Search, Optimization, and Machine Learning’. Addison Wesley. (1989).

- [2] J. Holland. '*Adaptation in Natural and Artificial Systems*'. The University of Michigan Press. (1975).
- [3] H. Sengoku & I. Yoshihara. '*A Fast TSP Solver Using GA on JAVA*'.
- [4] A. Barbadilla. '*Genética de Poblaciones*'. Universidad Autónoma de Barcelona. <http://bioinformatica.uab.es/divulgacio/genpob.html>
- [5] A. Blasco. '*Seis Horas de Genética de Poblaciones*' <http://www.dcam.upv.es/dcia/Download/GEN-POB.PDF>
- [6] A. Griffiths, S. Wessler, R. Lewontin, W. Gelbart. D. Suzuki & J. Miller. "*An Introduction to Genetic Analysis 8<sup>th</sup> E.*". W. H. Freeman & Company. (2004).



**B4. PRIMER CONGRESO ESTATAL “LA INVESTIGACIÓN EN EL POSGRADO”**

---

**DESARROLLO DE UN MODELO UTILIZANDO HIBRIDACIÓN PARA  
DIVERSIFICAR LA POBLACIÓN EN LA RESOLUCIÓN DE GRAFOS PLANOS NO  
DIRIGIDOS CON CRUCE MÍNIMO DE VÉRTICES, EN COMPUTACIÓN  
EVOLUTIVA**

**Francisco Javier Ornelas Zapata, Alejandro Padilla Díaz, Felipe Padilla Díaz.**

**Introducción.**

La Inteligencia Artificial (IA), surge como una necesidad del ser humano de crear “artefactos” que simulen la inteligencia humana. Dentro de la investigación científica este nombre es dado a un área de las ciencias de la computación encargado de modelar la inteligencia humana en sistemas computacionales. Existen múltiples ramas que se derivan de la IA tales como Agentes Inteligentes, Sistemas Clasificadores, Redes Neuronales, Computación Evolutiva, etc.

La Computación Evolutiva (CE), es una rama de la IA que agrupa algunas técnicas o métodos basados en la evolución natural y la genética. Toma como base la “Teoría de la Evolución de las Especies” propuesta por Charles Darwin y los descubrimientos realizados por Gregor Mendel en el campo de la genética. La CE esta dividida a su vez en tres principales áreas que son: Algoritmos Genéticos, Programación Evolutiva y Estrategias Evolutivas. La forma en que trabaja la computación evolutiva es generando una población, que contiene posibles soluciones (individuos) a los que se aplican operadores como cruzamiento, selección y mutación para “evolucionarlos” a soluciones normalmente de mejor calidad [1] [2].

**Objetivo.**

Se propone la creación de un nuevo modelo, en el cual se utilizará como operador de diversificación de la población la hibridación, para evitar la convergencia prematura de los algoritmos utilizados en CE, ya que actualmente la mutación se usa para dicho fin, pero se recomienda un porcentaje muy bajo [1], debido a que en porcentajes altos puede destruir las buenas soluciones alcanzadas. Este nuevo modelo se aplicara en la resolución de grafos planos no dirigidos con un mínimo cruce de vértices.

**Materiales y Métodos.**

Se utilizarán como base del presente trabajo los modelos de algoritmos genéticos propuestos por John Holland [2] y David Goldberg [1], la inteligencia artificial basada en la evolución de máquinas de estado finitas propuesta por L. J. Fogel, las estrategias evolutivas propuestas por Ingo Rechenberg y Hans Paul Schwefel, la teoría de la evolución de las especies de Darwin, los modelos para el dibujo de grafos propuestos por Kozo Sugiyama [3] y algunas investigaciones que se han realizado dentro del área de biología para mejorar genéticamente las características de algunos animales utilizando la hibridación entre distintas razas.

### **Resultados.**

Los resultados alcanzados hasta este momento son: la propuesta de tesis, su respectivo sustento en base a trabajos que se han realizado anteriormente en CE y dibujo de y que han obtenido buenos resultados y una gran aceptación por parte de la comunidad científica y trabajos dentro del área biológica como son hibridación o mestizaje, que sirven de inspiración del tema propuesto.

### **Conclusiones.**

Hasta este momento, se a realizado revisión de bibliografía y algunos trabajos de autores que cuentan con un alto prestigio dentro de los temas que tratan y que en algunos de los casos dichos trabajos son consideran como la base de dichos temas.

Teniendo ya una buena base teórica se requiere continuar revisando bibliografía para contar con una visión mas clara de las herramientas, modelos o métodos en los que me pueda inspirar o utilizar para lograr el objetivo propuesto.

Falta aún mucho trabajo para lograr llegar con éxito a la meta deseada.

### **Bibliografía.**

- [1] David Goldberg. *'Genetic Algorithms in Search, Optimization, and Machine Learning'*. Addison Wesley. (1989).
- [2] John Holland *'Adaptation in Natural and Artificial Systems'*. The University of Michigan Press. (1975).
- [3] Sugiyama Kozo *'Graph Drawing and Applications'* World Scientific. (2002).
- [4] Timo Eloranta *'TimGA: A Genetic Algorithm for Drawing Undirected Graphs'* *Divulgaciones Matemáticas Vol. 9 No. 2(2001)*.



## **ALGORITMOS GENÉTICOS: MUTACIÓN VS HIBRIDACIÓN.**

**Francisco Javier Ornelas Zapata, Alejandro Padilla Díaz, Felipe Padilla Díaz, Eunice Ponce de León Sentí.**

### **Introducción**

Los Algoritmos Genéticos (AG) son técnicas de búsqueda basadas en la evolución natural y la genética. Forman parte de una de las áreas más prometedoras de la Inteligencia Artificial (IA) conocida como Computación Evolutiva (CE).

Los AG fueron creados en los años 60 por John Holland basándose en la “Teoría de la evolución de las especies” propuesta por Charles Darwin y los descubrimientos realizados por Gregor Mendel en el campo de la genética.

La forma en que trabaja los AG's es generando una población que contiene posibles soluciones (individuos) a los que se aplican operadores como cruzamiento, selección y mutación para “evolucionarlos” a soluciones normalmente de mejor calidad [1] [2].

### **Objetivo.**

Se propone la creación de un nuevo modelo, en el que se introduce un proceso de hibridación que sustituye al operador clásico de mutación para diversificar la población en cada una de las generaciones del AG. Actualmente el operador de mutación se encarga de generar alteraciones en alguno o algunos de los individuos de cada generación para diversificar la población, pero como es un proceso aleatorio se corre el riesgo de eliminar o modificar resultados que califican bien en la resolución del problema propuesto. Por esta razón en la literatura relacionada a este operador se proponen porcentajes de mutación muy pequeños para evitar pérdida de buenas soluciones [1], pero esto a su vez supone que se disminuya en gran medida la diversificación.

Se espera que el proceso de hibridación incremente la diversificación de resultados en el AG, sin perder las buenas soluciones generadas y así evitar que se converja en óptimos locales, esto gracias a que dentro de este proceso se estarán introduciendo a nuestra población individuos con características diferentes a las que se vaya convergiendo dentro del proceso del AG.

Para verificar la eficacia del proceso de hibridación se esta generando un programa para resolver el Traveling Salesman Problem (TSP) utilizando el AG clásico y el AG con hibridación.

### **Materiales y Métodos.**



Se utiliza como base del presente trabajo los modelos de algoritmos genéticos propuestos por John Holland [2] y David Goldberg [1], la teoría de la evolución de las especies de Darwin, algunas investigaciones que se han realizado en el área de biología, para obtener animales o plantas con características específicas utilizando la hibridación entre distintas razas y trabajos realizados sobre el planteamiento matemático del TSP, algunas técnicas de resolución y sobre problemas de optimización combinatoria.

### **Resultados.**

Los resultados que se han obtenido hasta este momento es la propuesta del modelo, así como gran parte del sustento teórico de la misma, basado en trabajos realizados con anterioridad sobre AG, la mutación en AG, hibridación biológica, TSP y problemas de optimización combinatoria.

### **Conclusiones.**

Hasta este momento, se ha realizado revisión de bibliografía y algunos trabajos de autores que cuentan con un alto prestigio dentro de los temas que tratan y que en algunos de los casos dichos trabajos son considerados como la base de dichos temas.

Se cuenta ya con dos propuestas distintas para aplicar la hibridación en los AG y se está trabajando sobre el programa para verificar que la hibridación realmente mejorará los resultados obtenidos por el AG clásico y cual de las dos propuestas es mejor para continuar trabajando sobre ella.

### **Bibliografía.**

- [1] David Goldberg. '*Genetic Algorithms in Search, Optimization, and Machine Learning*'. Addison Wesley. (1989).
- [2] John Holland '*Adaptation in Natural and Artificial Systems*'. The University of Michigan Press. (1975).

**ALGORITMOS GENÉTICOS: HIBRIDACIÓN COMO DIVERSIFICADOR.****Francisco Ornelas, Alejandro Padilla, Felipe Padilla, Eunice Ponce de León.****Abstract**

Se presenta un nuevo modelo para resolver el Traveling Salesman Problem, el cual se basa en Algoritmos Genéticos, introduciendo un nuevo operador para diversificar la población para sustituir al operador de mutación y que esta bioinspirado en un proceso biológico para el mejoramiento de especies animales y vegetales llamado *hibridación*.

**Palabras clave.**

Algoritmos Genéticos, Traveling Salesman Problem, mutación, hibridación, Greedy Subtour Crossover, 2opt.

**1 Introducción.**

Los Algoritmos Genéticos (AG's), son una herramienta heurística que permite resolver problemas de optimización combinatoria que en muchas ocasiones no pueden ser tratados por métodos tradicionales, ya que por su complejidad, el tiempo computacional necesario para resolverlos es prohibitivo.

Uno de estos problemas de optimización es el Traveling Salesman Problem (TSP), en el que el objetivo es que un agente de ventas recorra determinado número de ciudades visitando solamente una vez cada una y por último regresar a la ciudad en que inicio el recorrido, con el menor costo posible.

Este problema incrementa su complejidad conforme crece la cantidad de ciudades a visitar, por lo que al llegar a un determinado número este se vuelve intratable por cualquier método tradicional ya que con cada incremento de ciudades crece su complejidad exponencialmente.

Los AG's han demostrado ser una buena herramienta para resolver problemas de explosión combinatoria, ya que logran obtener buenas soluciones en tiempos relativamente más cortos. Desafortunadamente, su principal problema es que convergen prematuramente a resultados de buena calidad pero no necesariamente el mejor, esto debido a que el operador que utilizan para diversificar la población de posibles resultados debe ser utilizado en porcentajes muy pequeños [1], para que este, afecte lo menos posible los buenos resultados que se hayan

obtenido hasta el momento, ya que este operador funciona de forma aleatoria y puede afectar tanto a buenos como a malos resultados, por lo que al incrementar su porcentaje se gana en diversificación pero se incrementa el riesgo de pérdida de buenos resultado con que se cuenta.

Nosotros proponemos un nuevo modelo en el que se cambia el operador de mutación por un nuevo operador al que llamamos hibridación y que está inspirado en un proceso utilizado en biología para mejorar las características de individuos animales y/o vegetales, que es conocido con este nombre o como mestizaje.

En la sección 2, mostramos las principales características del AG clásico, en la sección 3 describimos el modelo propuesto así como el trabajo realizado, en la sección 4 mostramos los resultados obtenidos, en la sección 5 resultados, en la 6 las conclusiones y trabajo futuro y por último bibliografía.

## **2 Algoritmos Genéticos.**

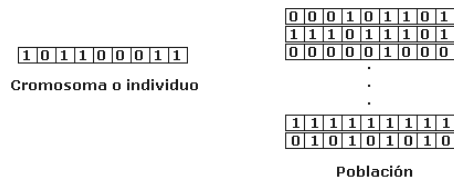
Al igual que en la naturaleza los individuos cuentan con características que les permiten sobrevivir en el entorno en que se desarrollan y además algunas de estas características se modifican en el transcurso de los años permitiendo al individuo ser más competitivo en su entorno, de la misma manera los AG's intentan que los individuos o posibles soluciones evolucionen hasta alcanzar un alto grado de competitividad, lo que resultaría en una buena solución al problema que se desea resolver [1][2].

Las principales características de los AG's son:

*Cromosoma y población.*

Para el AG es importante determinar el espacio en el que se encuentran las posibles soluciones al problema que se pretende resolver. Por esta razón se requiere generar individuos que contengan la información necesaria para resolver el problema dado. Cada uno de los individuos está formado por una cadena ya sea de datos binarios, reales o alfanuméricos. A cada una de estas cadenas se le llama individuo o cromosoma.

Para iniciar el AG requiere que se genere una población de individuos sobre la cual aplicará operadores que favorezcan la permanencia de los mejores individuos y los menos capaces desaparezcan. El tamaño de la población se define en base al problema a resolver así como también el tamaño del cromosoma.



Cada cromosoma o individuo es generado de forma aleatoria, por lo que se espera una gran diversidad en la población inicial aún cuando puede darse el caso de que existan individuos con las mismas característica que otros.

Las siguientes poblaciones se generan en base a los individuos más aptos que pasan a la siguiente generación y serán los padres de los individuos necesarios para completar nuevamente el tamaño inicial de la población. El número de individuos que pasan de una generación a otra, así como el número de nuevos individuos es un parámetro que hay que definir desde el inicio del AG.

*Grado de adaptación o fitness.*

En la naturaleza los individuos más aptos son los que sobreviven. De la misma forma en los AG cada individuo cuenta con características que pueden ser evaluadas y definir su permanencia dentro de la población y generar descendencia.

Para determinar que individuos son los más aptos se requiere calificar a cada uno de los miembros de la población. Esta calificación sería su grado de adaptación. En AG este grado de adaptación es llamado *fitness* y se evalúa en base a una función llamada *función objetivo* la cual se genera en base a las características de las soluciones que se desea obtener.

*Selección.*

Después de haber calificado a cada individuo de la población, existen varias formas de determinar quienes pasaran a la siguiente generación y serán los progenitores de los nuevos individuos. La forma que utiliza el AG clásico es el *elitismo*, donde se selecciona a los individuos que obtengan las calificaciones más altas hasta cubrir el número necesario de individuos que pasara a la siguiente generación.

*Cruza.*

En la naturaleza, no sólo es importante que se conserven los individuos más aptos, sino que existan nuevos individuos que conserven las cualidades de los anteriores.

Dentro de los AG sucede algo similar ya que solamente los individuos que sean seleccionados como más aptos podrán cruzarse y generar nuevos individuos los cuales contendrán parte del



material genético de sus padres lo que dará como resultado una posible nueva solución al problema.

### *Mutación.*

Muy pocas veces la naturaleza se equivoca, pero cuando sucede alguno de estos errores, el material genético de un individuo es modificado produciendo un cambio que puede traerle ventajas o desventajas sobre los demás individuos de su especie.

En los AG's la mutación se utiliza para generar un cambio en algún o algunos individuos, lo cual permita explorar zonas de nuestro espacio de búsqueda que no han sido explorados hasta el momento. La mutación funciona dentro del AG como una herramienta de diversificación.

Dentro del trabajo realizado por David Goldberg [1], se recomienda utilizar porcentajes de mutación muy pequeños ya que en porcentajes altos, puede producir un efecto contrario y hacer que el algoritmo pierda soluciones de buena calidad.

Estos son los elementos de un AG básico, pero existen muchas otras variantes, además de híbridos que utilizan las mejores cualidades de AG para resolver problemas de optimización.

### **3 Modelo propuesto y trabajo realizado.**

El modelo que proponemos toma como base el AG clásico. Sobre este se realiza el cambio del operador de mutación, que es el encargado de realizar el proceso de diversificación de la población, por un nuevo operador al que llamamos hibridación y está inspirado en un proceso de biología conocido con este mismo nombre.

El proceso de hibridación biológica, se realiza para mejorar características tanto en vegetales como en animales.

De manera muy general este proceso consiste en seleccionar individuos de razas distintas que contengan características deseables. Estos individuos deben mostrar un alto grado de la o las características deseadas, para que genéticamente sean transmitidas a sus descendientes.

Dentro de los AG's conforme se avanza en el número de generaciones, los individuos tienden a presentar cada vez mayor parecido entre sí. Por esta razón se necesita un buen mecanismo de diversificación. El modelo propuesto sustituye el proceso común de mutación, por la introducción de individuos, los cuales se espera contengan características frescas, además, antes de que participen en la creación de nuevos individuos, son evaluadas sus características

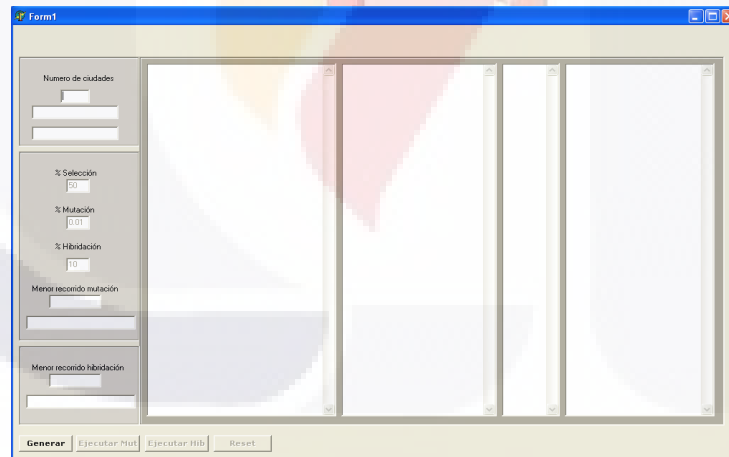


al igual que las de los individuos conservados de la generación anterior y compiten por su supervivencia.

La forma en que trabaja el nuevo modelo es:

- Generar una población inicial, de forma aleatoria.
- Se evalúa cada uno de los individuos de la población (posibles soluciones).
- Se seleccionan de forma elitista los individuos cuyo costo del viaje sea más pequeño, hasta tener un 50% de los individuos que formaran parte de la nueva población la cual deberá tener al final el mismo número de individuos que la población inicial.
- Se generan nuevos individuos en base a la cruce de los individuos que fueron seleccionados en la etapa anterior, esto hasta cubrir un determinado porcentaje el cual debe ser menor del 50% del tamaño de la población.
- Se generan aleatoriamente los individuos necesarios para completar el tamaño total de la población.
- Se hace un ciclo que regresa al segundo paso hasta que se llegue a cumplir un criterio de parada.

Se desarrolló una herramienta de software para verificar el comportamiento del nuevo modelo en comparación con el AG clásico.



Herramienta de Software para resolver el TSP utilizando un AG clásico y un nuevo modelo de AG utilizando hibridación.

Esta herramienta resuelve el problema del TSP. La forma en que se realizó dicha herramienta se describe a continuación.

La herramienta trabaja sobre un grafo plano, completo, no dirigido, para el cual se generan aleatoriamente las distancias entre ciudades. El número de ciudades es definido por el usuario hasta un máximo de 30.

Primero se genera una población inicial de 30 individuos con n número de ciudades (n definido por el usuario y menor o igual a 30), así como una matriz de distancias entre ciudades, ambas de forma aleatoria.

2	3	5	4	1
2	4	1	5	3
4	2	1	5	3
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
3	1	5	4	2

Población inicial de 30 individuos y 5 ciudades.

0	12	15	11	5
12	0	5	6	20
15	5	0	5	28
11	6	5	0	16
5	20	28	16	0

Matriz de distancia entre ciudades.

Esta población inicial se utiliza para ser resuelta por el AG clásico y por el nuevo modelo.

El método de evaluación de los individuos tanto en el AG como en el modelo propuesto es, el costo mínimo del viaje, el cual resulta de sumar la distancia entre las ciudades relacionadas, de acuerdo al orden en que aparecen en el individuo, y agregando la distancia entre la última y primera ciudades del mismo.

2	3	5	4	1
---	---	---	---	---

Tomando el primer individuo de la población y la matriz de distancias anteriores el fitness para este es 72.

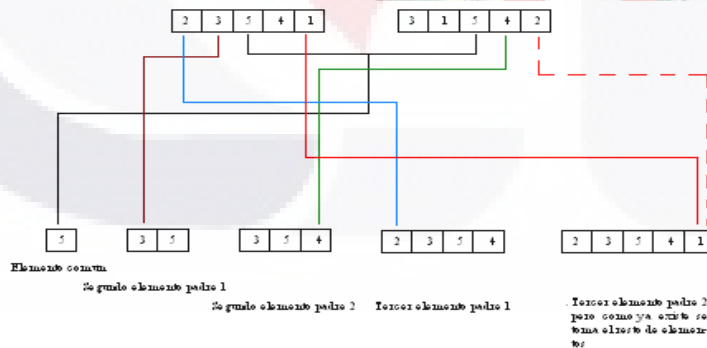
Después de la evaluación se realiza el método de selección que es elitista. Se toman los individuos que calificaron con menores costos del viaje, hasta cubrir un 50% de los individuos de la nueva población.

Para el AG clásico, por medio de cruzamiento se genera el 50% restante de la población. Para el nuevo modelo generamos un 33% de individuos tomando como base el tamaño total de la población inicial como 100%.

El tipo de cruce que utilizamos es el propuesto por Sengoku y Yoshihara [2] al cual llaman Greedy Subtour Crossover (GSX).

La forma en que funciona el GSX es la siguiente:

- Se seleccionan dos padres de forma aleatoria.
- Se elige un mismo elemento en ambos padres.
- El elemento elegido es el primer miembro del nuevo individuo.
- Se hace un recorrido en el primer padre hacia el inicio del mismo y en el segundo hacia el fin.
- Se verifica si los nuevos elementos son válidos para el nuevo individuo. Un elemento es válido siempre que no se encuentre contenido ya en el nuevo individuo.
- El proceso de recorrido de los padres es cíclico y se detiene hasta que alguno de los padres no tenga hacia donde ser recorrido o hasta que algún elemento no sea válido.
- Por último se adicionan al final los elementos que no estén todavía contenidos en el nuevo individuo.

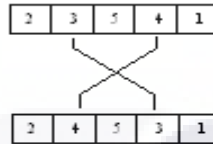


Este tipo de cruce permite conservar subtours que estén contenidos en los padres seleccionados.

El último paso dentro del AG clásico es la mutación. El método que utilizamos fue el 2opt.

Este proceso se describe a continuación:

- Se selecciona un individuo.
- Se selecciona un elemento de este individuo, así como una posición dentro del mismo.
- Se hace el intercambio entre el elemento seleccionado y el elemento que se encuentra en la posición también seleccionada.



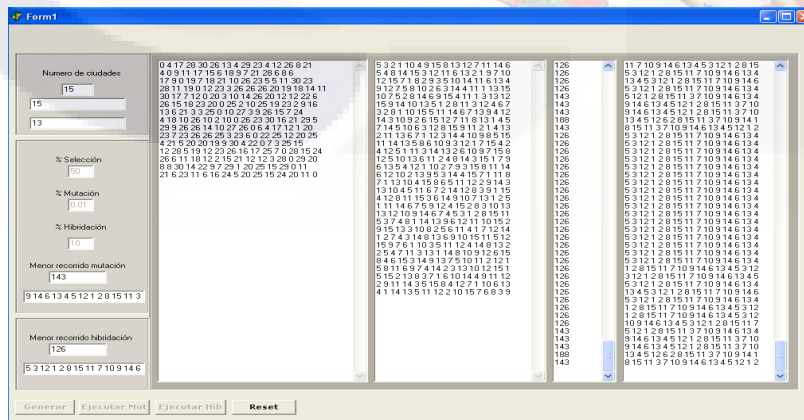
Esta figura muestra el proceso de mutación en el que se elige el elemento con valor = 4 y se intercambia a la posición = 2.

Este proceso se realiza en un 1% del número de elementos contenidos en la población. Esto es, el número de individuos (en el software desarrollado 30) multiplicado por el número de ciudades.

En el modelo que estamos proponiendo, el 22% restante de la población se obtiene insertando nuevos individuos generados aleatoriamente, esto en sustitución del operador de mutación.

Estos procesos se realizan de forma cíclica hasta llegar a un criterio de parada que nosotros definimos en 1000 generaciones o al llegar a 10 generaciones en las que no se haya generado ningún individuo que mejore al que obtuvo el menor subtour de la generación anterior.

En la figura siguiente se muestra una corrida para 15 ciudades:



Se realizaron 120 corridas al software con número de ciudades igual a 5, 10, 15, 20, 25 y 30.

#### 4 Resultados.

Después de 120 ejecuciones del software (20 para cada número de ciudades utilizado), se encontró que si se utilizan 5 ciudades, los resultados tanto del AG clásico como del nuevo modelo son los mismos.

De las ejecuciones a partir de 10 ciudades se detectó una mejora significativa, aún cuando no fué el que esperábamos. Esta mejora consiste en que aproximadamente en el 99% de las corridas, el número de generaciones utilizadas por el nuevo modelo fueron menores a las necesarias en el AG clásico, llegando en un 83% de las ocasiones al mismo resultado, un 13% a mejores resultados y un 4% a peores.

	5	10	15	20	25	30
1	11 36	11 36	17 129	12 129	16 186	12 186
2	11 68	11 68	14 90	12 90	16 162	12 162
3	11 40	11 40	16 65	12 65	16 118	12 118
4	11 52	11 52	13 100	12 100	17 135	12 135
5	11 52	11 52	14 59	12 59	16 137	12 137
6	11 62	11 62	14 85	12 85	20 105	12 105
7	11 53	11 53	17 117	12 117	18 177	12 177
8	11 64	11 64	16 60	12 60	14 100	12 100
9	11 51	11 51	14 87	12 87	15 112	12 112
10	11 59	11 59	14 94	12 94	16 108	12 108
11	11 59	11 59	13 88	12 88	18 125	12 125
12	11 66	11 66	16 91	12 91	18 132	12 132
13	11 58	11 58	13 83	12 83	13 110	12 110
14	11 68	11 68	13 80	12 80	15 99	12 99
15	11 53	11 53	16 89	12 89	16 141	12 141
16	11 54	11 54	13 70	12 70	13 108	12 108
17	11 56	11 56	14 89	12 89	17 146	12 146
18	11 53	11 53	13 101	12 101	13 115	12 115
19	11 51	11 51	13 45	12 45	18 94	12 94
20	11 75	11 75	14 62	12 62	17 95	12 95

Tabla que muestra los resultados de las 120 ejecuciones, así como marcadas con verde las ejecuciones donde se mejoraron los resultados y de rojo aquellas en las que se empeoraron.

#### 5 Conclusiones y trabajo futuro.

Tomando en cuenta los resultados obtenidos, consideramos que vamos por buen camino ya que nuestra propuesta obtuvo buenos resultados en comparación con el AG clásico por lo que ésta modificación se podría hacer a AG's más avanzados que cuentan actualmente con un mejor rendimiento y posiblemente se generarían AG's más competitivos.

Aún cuando el objetivo de nuestra propuesta era mejorar las soluciones (lo cual con el software realizado se demostró), encontramos otra opción de mejora, ya que al llegar a los mismos resultados pero en un menor número de iteraciones, esto produciría un AG más competitivo contra otras herramientas heurísticas que se caracterizan por su rapidez.

Dentro del trabajo futuro tenemos contemplado continuar nuestra investigación sobre esta propuesta, además de una mejora al software para que funcione con un mayor número de ciudades y con grafos no necesariamente completos, ya que esto nos permitiría verificar el



funcionamiento de nuestro modelo utilizando benchmarks de algoritmos y modelos que resuelven el problema del TSP lo cual le daría más validez a nuestra investigación

**Referencias.**

- [1] Goldberg D.: *'Genetic Algorithms in Search, Optimization, and Machine Learning'*. Addison Wesley. (1989).
- [2] Holland J.: *'Adaptation in Natural and Artificial Systems'*. The University of Michigan Press. (1975).
- [3] Sengoku H. & Yoshihara I.: *'A Fast TSP Solver Using GA on JAVA'*.



ANEXO C

DATOS UTILIZADOS EN EL DISEÑO DE EXPERIMENTOS

# Ejecucion	# Generaciones	% de Poblacion	Generaciones entre Miraciones	AG sin Migracion	AG con Migracion
1	500	50	5	61224.93	59672.57
2	500	50	5	60217.03	51837.22
3	500	50	5	60643.84	57376.68
4	500	50	5	56827.49	53463.10
5	500	50	5	55126.71	60116.55
6	500	50	5	69839.99	61464.96
7	500	50	5	56952.54	60057.24
8	500	50	5	55400.52	55588.61
9	500	50	5	64701.24	58423.73
10	500	50	5	59587.59	63287.45
11	500	50	5	56165.83	58917.40
12	500	50	5	61962.48	60156.26
13	500	50	5	56896.79	51937.60
14	500	50	5	53515.03	54146.69
15	500	50	5	62768.49	57613.28
16	500	50	5	61356.10	55730.59
17	500	50	5	57209.47	54156.46
18	500	50	5	55116.26	54497.80
19	500	50	5	64948.59	55443.95
20	500	50	5	56553.22	54626.99
21	500	50	5	54966.57	54845.19
22	500	50	5	60820.29	61533.97
23	500	50	5	55384.96	60837.98
24	500	50	5	62300.71	57224.88
25	500	50	5	54489.61	60297.19
26	500	50	5	56284.49	57421.39
27	500	50	5	57318.79	56261.97
28	500	50	5	63546.51	57672.04
29	500	50	5	59522.86	59663.27
30	500	50	5	58651.97	59530.89
31	1000	50	5	52448.28	47900.45

32	1000	50	5	53841.68	52379.45
33	1000	50	5	53820.00	48519.57
34	1000	50	5	53965.24	47467.62
35	1000	50	5	56091.84	48201.40
36	1000	50	5	50981.02	50108.93
37	1000	50	5	53705.51	48505.08
38	1000	50	5	57790.17	50321.13
39	1000	50	5	49270.07	55074.46
40	1000	50	5	55051.97	53016.18
41	1000	50	5	55665.18	49107.18
42	1000	50	5	49141.35	49329.95
43	1000	50	5	52063.78	47123.27
44	1000	50	5	47480.09	53002.93
45	1000	50	5	52546.74	47028.59
46	1000	50	5	53957.16	50331.62
47	1000	50	5	55566.49	46353.10
48	1000	50	5	49166.98	47555.86
49	1000	50	5	54784.21	47492.09
50	1000	50	5	53299.09	51032.42
51	1000	50	5	49002.49	43812.52
52	1000	50	5	56153.96	46044.94
53	1000	50	5	46611.98	47829.67
54	1000	50	5	47701.49	49501.27
55	1000	50	5	55618.89	49869.12
56	1000	50	5	52358.19	49936.80
57	1000	50	5	51377.57	51444.65
58	1000	50	5	54839.82	50504.11
59	1000	50	5	53794.85	46117.38
60	1000	50	5	52466.31	44242.47
61	2000	50	5	49071.34	49003.31
62	2000	50	5	42874.84	49582.89
63	2000	50	5	45738.18	46967.54
64	2000	50	5	44724.57	45495.05
65	2000	50	5	44649.56	44562.84
66	2000	50	5	43216.28	44854.33
67	2000	50	5	36717.52	41250.60
68	2000	50	5	46144.29	45971.29
69	2000	50	5	46151.82	44168.00
70	2000	50	5	47717.34	38681.00
71	2000	50	5	49688.89	40119.50
72	2000	50	5	47187.32	46715.65

73	2000	50	5	42862.73	43986.45
74	2000	50	5	44413.98	45826.79
75	2000	50	5	50414.14	46469.96
76	2000	50	5	49025.49	44610.52
77	2000	50	5	40027.40	46436.06
78	2000	50	5	45653.38	42354.24
79	2000	50	5	46514.74	43591.78
80	2000	50	5	42201.60	40676.65
81	2000	50	5	44969.01	42212.45
82	2000	50	5	49816.98	43599.64
83	2000	50	5	46598.15	41137.38
84	2000	50	5	46543.48	44780.67
85	2000	50	5	41951.40	42029.49
86	2000	50	5	52274.93	42652.19
87	2000	50	5	45174.25	42611.52
88	2000	50	5	42348.09	39421.76
89	2000	50	5	49169.21	39632.94
90	2000	50	5	44727.11	40661.66
91	3000	50	5	47964.62	41539.79
92	3000	50	5	40683.83	43086.45
93	3000	50	5	38978.92	43590.82
94	3000	50	5	40837.52	38611.92
95	3000	50	5	43275.94	46019.87
96	3000	50	5	42327.60	40191.82
97	3000	50	5	45719.06	45419.96
98	3000	50	5	47627.03	42425.00
99	3000	50	5	42011.25	39254.29
100	3000	50	5	44050.34	44378.30
101	3000	50	5	42174.35	43561.86
102	3000	50	5	44877.87	41165.06
103	3000	50	5	45247.69	37437.99
104	3000	50	5	47597.07	42958.18
105	3000	50	5	41418.51	45023.88
106	3000	50	5	39374.69	40404.54
107	3000	50	5	47304.34	35438.79
108	3000	50	5	40980.19	39264.95
109	3000	50	5	44549.84	40640.26
110	3000	50	5	44091.97	43245.77
111	3000	50	5	44891.51	37421.76
112	3000	50	5	39188.89	39812.44
113	3000	50	5	42105.88	40608.95

114	3000	50	5	40133.44	45449.85
115	3000	50	5	46823.24	39760.42
116	3000	50	5	42750.97	40481.92
117	3000	50	5	42841.79	39308.15
118	3000	50	5	41719.37	40448.71
119	3000	50	5	46547.97	41679.43
120	3000	50	5	46711.06	48528.41
121	500	75	5	55434.90	51108.89
122	500	75	5	59508.86	51371.44
123	500	75	5	49691.24	55617.93
124	500	75	5	55108.84	56204.71
125	500	75	5	52789.98	56632.98
126	500	75	5	63233.00	54867.24
127	500	75	5	56370.82	60163.38
128	500	75	5	57174.00	55836.68
129	500	75	5	50448.94	54772.57
130	500	75	5	51595.35	54385.05
131	500	75	5	59790.92	54278.58
132	500	75	5	47139.79	54701.63
133	500	75	5	62360.35	54564.89
134	500	75	5	58239.98	51300.58
135	500	75	5	58052.28	55975.04
136	500	75	5	62990.80	53325.14
137	500	75	5	59205.28	56146.10
138	500	75	5	61165.79	56747.62
139	500	75	5	52731.71	53694.66
140	500	75	5	56660.03	53343.46
141	500	75	5	62110.68	49442.69
142	500	75	5	55512.66	55086.91
143	500	75	5	56985.85	54004.92
144	500	75	5	62345.48	52740.09
145	500	75	5	59913.24	49063.64
146	500	75	5	58996.12	49700.35
147	500	75	5	56984.92	57644.51
148	500	75	5	60847.41	55547.63
149	500	75	5	59126.59	52412.64
150	500	75	5	57854.67	61436.78
151	1000	75	5	45887.47	42044.55
152	1000	75	5	44178.08	46493.37
153	1000	75	5	51324.39	42099.17
154	1000	75	5	49911.64	45322.87



155	1000	75	5	47772.59	44706.96
156	1000	75	5	44765.82	47557.10
157	1000	75	5	46393.83	47091.27
158	1000	75	5	51389.14	49989.96
159	1000	75	5	47808.02	47545.25
160	1000	75	5	47494.21	43464.90
161	1000	75	5	52621.75	48391.85
162	1000	75	5	43710.86	48097.00
163	1000	75	5	44582.01	47366.33
164	1000	75	5	50831.33	46722.45
165	1000	75	5	49821.77	42280.43
166	1000	75	5	50643.03	49706.10
167	1000	75	5	50937.19	45343.00
168	1000	75	5	57522.67	46458.42
169	1000	75	5	48685.88	43307.77
170	1000	75	5	47824.37	45801.56
171	1000	75	5	47880.25	52215.99
172	1000	75	5	44713.41	44962.08
173	1000	75	5	50853.27	49119.13
174	1000	75	5	47572.41	51573.79
175	1000	75	5	42989.52	49912.75
176	1000	75	5	42771.54	44193.57
177	1000	75	5	45209.52	47913.64
178	1000	75	5	53555.82	44433.96
179	1000	75	5	56186.25	45823.55
180	1000	75	5	51405.11	44844.64
181	2000	75	5	46748.24	41771.98
182	2000	75	5	45139.87	41867.47
183	2000	75	5	40389.78	43123.69
184	2000	75	5	44401.92	42238.86
185	2000	75	5	37614.22	41938.60
186	2000	75	5	42879.87	39155.84
187	2000	75	5	47558.91	39310.77
188	2000	75	5	39826.69	37675.96
189	2000	75	5	45547.72	43766.31
190	2000	75	5	39986.47	39586.26
191	2000	75	5	45057.86	40833.12
192	2000	75	5	45631.67	39781.83
193	2000	75	5	40227.99	40735.47
194	2000	75	5	40968.06	40465.88
195	2000	75	5	42615.36	41539.46

196	2000	75	5	36543.10	37182.93
197	2000	75	5	48457.09	44489.54
198	2000	75	5	43637.02	40022.73
199	2000	75	5	44610.37	41839.03
200	2000	75	5	41548.93	42505.53
201	2000	75	5	43831.21	40022.96
202	2000	75	5	43840.15	41453.63
203	2000	75	5	48329.78	43285.02
204	2000	75	5	40525.03	40210.53
205	2000	75	5	40348.78	41242.23
206	2000	75	5	44810.97	37982.70
207	2000	75	5	44342.34	41818.57
208	2000	75	5	41905.25	38742.65
209	2000	75	5	38826.23	41897.91
210	2000	75	5	39388.46	39730.31
211	3000	75	5	40087.57	45750.33
212	3000	75	5	37540.12	36905.51
213	3000	75	5	40331.78	38513.93
214	3000	75	5	41408.26	36474.17
215	3000	75	5	38085.66	39028.13
216	3000	75	5	41525.37	39511.18
217	3000	75	5	38082.24	38391.36
218	3000	75	5	40273.13	42366.53
219	3000	75	5	44063.19	39344.20
220	3000	75	5	40220.06	33959.87
221	3000	75	5	46478.83	33995.97
222	3000	75	5	45330.13	39650.47
223	3000	75	5	42676.03	37203.09
224	3000	75	5	43482.20	40973.90
225	3000	75	5	40067.20	42860.51
226	3000	75	5	35169.67	37805.93
227	3000	75	5	33193.28	43713.52
228	3000	75	5	40691.49	39115.11
229	3000	75	5	40744.43	38584.44
230	3000	75	5	37706.82	40011.90
231	3000	75	5	34293.66	38874.95
232	3000	75	5	40633.86	39178.46
233	3000	75	5	36183.69	38981.51
234	3000	75	5	43371.05	36686.06
235	3000	75	5	39126.61	39229.47
236	3000	75	5	40131.89	42243.88

237	3000	75	5	38094.38	39321.12
238	3000	75	5	40314.49	43118.44
239	3000	75	5	44607.24	38680.33
240	3000	75	5	43400.85	39321.88
241	500	100	5	60365.66	53361.21
242	500	100	5	50860.04	45923.96
243	500	100	5	49990.48	57029.06
244	500	100	5	57395.83	53428.75
245	500	100	5	52931.43	48748.47
246	500	100	5	53093.65	51881.58
247	500	100	5	54470.53	54025.38
248	500	100	5	56028.07	52880.29
249	500	100	5	51439.23	52706.58
250	500	100	5	52931.55	57092.23
251	500	100	5	53348.97	50361.02
252	500	100	5	56784.55	50245.31
253	500	100	5	57682.46	52208.00
254	500	100	5	52709.18	53208.85
255	500	100	5	50327.82	56725.50
256	500	100	5	59418.15	50315.25
257	500	100	5	53581.99	48198.50
258	500	100	5	49628.10	54508.95
259	500	100	5	47263.36	55646.58
260	500	100	5	49699.42	47155.51
261	500	100	5	49909.44	46041.03
262	500	100	5	58913.24	49891.02
263	500	100	5	50114.85	52681.89
264	500	100	5	47021.09	46956.01
265	500	100	5	52671.03	49796.72
266	500	100	5	53113.07	47889.03
267	500	100	5	55086.33	54541.89
268	500	100	5	49001.40	52986.86
269	500	100	5	48052.53	52931.17
270	500	100	5	50720.18	48228.44
271	1000	100	5	44920.50	49718.85
272	1000	100	5	41992.70	43444.99
273	1000	100	5	43769.97	41297.85
274	1000	100	5	46273.89	46348.91
275	1000	100	5	44130.87	43885.53
276	1000	100	5	44244.48	45272.75
277	1000	100	5	45167.75	42635.04

278	1000	100	5	47438.94	46063.33
279	1000	100	5	42052.17	48266.85
280	1000	100	5	54475.95	46373.62
281	1000	100	5	45505.43	41406.39
282	1000	100	5	45436.61	44316.58
283	1000	100	5	46748.95	43015.14
284	1000	100	5	43493.59	41314.01
285	1000	100	5	45631.08	45020.02
286	1000	100	5	38908.17	43312.89
287	1000	100	5	43438.99	39314.19
288	1000	100	5	44786.00	46939.47
289	1000	100	5	49305.72	43491.42
290	1000	100	5	49814.39	43173.36
291	1000	100	5	44432.31	42094.64
292	1000	100	5	43180.15	40967.00
293	1000	100	5	51236.83	46731.93
294	1000	100	5	45279.30	40967.05
295	1000	100	5	47525.12	40626.12
296	1000	100	5	45310.87	44735.05
297	1000	100	5	42617.06	40891.48
298	1000	100	5	44304.71	48351.07
299	1000	100	5	44265.86	46490.25
300	1000	100	5	42323.58	45199.35
301	2000	100	5	41118.46	37601.47
302	2000	100	5	40123.60	38007.15
303	2000	100	5	44452.97	44294.05
304	2000	100	5	39693.55	40136.83
305	2000	100	5	39921.50	38393.87
306	2000	100	5	44355.82	43167.95
307	2000	100	5	44996.76	40049.39
308	2000	100	5	41078.84	38933.24
309	2000	100	5	38369.30	38999.46
310	2000	100	5	40581.84	40826.68
311	2000	100	5	39380.74	41015.22
312	2000	100	5	39207.44	39402.76
313	2000	100	5	43228.88	39956.60
314	2000	100	5	45043.72	44718.76
315	2000	100	5	35691.65	37185.19
316	2000	100	5	40660.08	39500.43
317	2000	100	5	43848.00	41311.13
318	2000	100	5	41275.50	38127.54

319	2000	100	5	40055.75	37111.83
320	2000	100	5	41721.25	39947.93
321	2000	100	5	40582.28	38144.84
322	2000	100	5	41740.30	41128.53
323	2000	100	5	39605.53	37706.57
324	2000	100	5	45691.85	37555.08
325	2000	100	5	42105.47	41682.69
326	2000	100	5	44851.31	43437.93
327	2000	100	5	45975.41	40881.47
328	2000	100	5	39213.91	40895.42
329	2000	100	5	44345.64	40507.89
330	2000	100	5	36234.40	41390.37
331	3000	100	5	38844.14	36433.19
332	3000	100	5	41106.33	38309.29
333	3000	100	5	37075.92	35800.61
334	3000	100	5	40727.27	35794.85
335	3000	100	5	35785.78	35384.80
336	3000	100	5	36378.12	39650.23
337	3000	100	5	39908.27	39110.03
338	3000	100	5	36814.58	35728.86
339	3000	100	5	39346.00	40896.29
340	3000	100	5	37382.42	39179.63
341	3000	100	5	40076.64	38203.63
342	3000	100	5	35229.37	39575.39
343	3000	100	5	36468.68	37706.91
344	3000	100	5	37584.92	40801.78
345	3000	100	5	43039.68	40357.06
346	3000	100	5	38001.96	35499.17
347	3000	100	5	40345.31	38691.81
348	3000	100	5	41306.79	37216.31
349	3000	100	5	38024.26	39832.36
350	3000	100	5	39978.30	36900.14
351	3000	100	5	37986.70	39202.83
352	3000	100	5	37843.80	35624.12
353	3000	100	5	41221.05	36349.59
354	3000	100	5	37410.05	39843.51
355	3000	100	5	40266.90	36456.29
356	3000	100	5	39745.90	37248.50
357	3000	100	5	43369.11	33768.11
358	3000	100	5	42614.42	37271.86
359	3000	100	5	42567.85	33801.23



360	3000	100	5	40046.47	36019.41
361	500	50	7	59933.30	57821.53
362	500	50	7	57442.80	59020.76
363	500	50	7	59287.08	52029.44
364	500	50	7	57795.28	56525.00
365	500	50	7	59437.24	54979.08
366	500	50	7	65791.82	58134.13
367	500	50	7	55963.12	60963.65
368	500	50	7	59006.54	53860.50
369	500	50	7	59792.29	60313.11
370	500	50	7	53695.77	59603.76
371	500	50	7	64255.20	57178.83
372	500	50	7	55068.78	60659.84
373	500	50	7	57923.67	59698.41
374	500	50	7	56998.22	60402.66
375	500	50	7	56314.51	61258.10
376	500	50	7	61238.53	61968.35
377	500	50	7	63054.66	57360.40
378	500	50	7	62063.65	54950.14
379	500	50	7	56001.64	62801.44
380	500	50	7	55069.99	55253.01
381	500	50	7	63009.89	60193.87
382	500	50	7	56866.19	57224.45
383	500	50	7	56579.29	60339.03
384	500	50	7	55460.36	57624.31
385	500	50	7	66070.62	59902.48
386	500	50	7	61605.40	55707.96
387	500	50	7	59178.75	57365.68
388	500	50	7	60563.50	57910.34
389	500	50	7	60128.77	59018.10
390	500	50	7	60044.89	58788.44
391	1000	50	7	50599.95	47193.26
392	1000	50	7	53396.08	48537.95
393	1000	50	7	54200.51	53295.02
394	1000	50	7	43516.90	53253.70
395	1000	50	7	50785.63	47992.97
396	1000	50	7	52574.51	54736.75
397	1000	50	7	52439.91	53880.69
398	1000	50	7	46631.78	49812.11
399	1000	50	7	52597.41	47633.94
400	1000	50	7	51008.84	44620.56

401	1000	50	7	52153.01	51198.57
402	1000	50	7	53407.14	43646.87
403	1000	50	7	49579.78	47619.61
404	1000	50	7	52176.03	46798.19
405	1000	50	7	46005.51	47860.26
406	1000	50	7	51442.96	47514.89
407	1000	50	7	51028.52	51166.85
408	1000	50	7	53828.91	49617.34
409	1000	50	7	45641.75	51021.88
410	1000	50	7	59946.09	48766.60
411	1000	50	7	49713.14	49470.41
412	1000	50	7	53265.22	50513.82
413	1000	50	7	54809.90	47288.39
414	1000	50	7	51904.24	51694.87
415	1000	50	7	47207.93	45678.41
416	1000	50	7	57893.15	52007.30
417	1000	50	7	47844.83	51414.84
418	1000	50	7	53192.08	52757.47
419	1000	50	7	49787.83	46913.86
420	1000	50	7	54980.83	50495.21
421	2000	50	7	53434.13	47297.79
422	2000	50	7	48147.02	44172.60
423	2000	50	7	52118.87	46679.45
424	2000	50	7	42705.20	38374.60
425	2000	50	7	43576.46	43050.93
426	2000	50	7	48173.81	43713.38
427	2000	50	7	44508.80	39601.34
428	2000	50	7	47757.31	38915.13
429	2000	50	7	49692.24	40339.88
430	2000	50	7	47611.68	40505.14
431	2000	50	7	44543.32	40473.86
432	2000	50	7	40551.68	45567.95
433	2000	50	7	45574.75	44547.43
434	2000	50	7	42757.32	44550.55
435	2000	50	7	41089.92	40547.37
436	2000	50	7	44962.53	44461.87
437	2000	50	7	46499.12	45725.49
438	2000	50	7	44517.26	51784.53
439	2000	50	7	42993.39	38559.23
440	2000	50	7	42736.90	44503.38
441	2000	50	7	50021.47	45581.32

442	2000	50	7	43737.95	41257.50
443	2000	50	7	47763.36	44143.04
444	2000	50	7	43360.27	42797.76
445	2000	50	7	48715.18	53302.67
446	2000	50	7	51300.34	40719.89
447	2000	50	7	45486.82	42149.64
448	2000	50	7	43952.09	42906.90
449	2000	50	7	48523.37	45521.38
450	2000	50	7	46322.26	45074.11
451	3000	50	7	38665.39	41106.12
452	3000	50	7	40965.50	39480.72
453	3000	50	7	42377.26	45051.62
454	3000	50	7	43106.16	45907.96
455	3000	50	7	36574.34	43334.54
456	3000	50	7	43601.39	44180.64
457	3000	50	7	42926.56	35698.59
458	3000	50	7	45581.99	40483.63
459	3000	50	7	46049.16	44994.06
460	3000	50	7	44420.16	42156.05
461	3000	50	7	43221.58	43698.49
462	3000	50	7	47234.96	39789.94
463	3000	50	7	42117.65	45572.52
464	3000	50	7	44517.34	41253.83
465	3000	50	7	42124.33	41229.46
466	3000	50	7	48806.30	43957.65
467	3000	50	7	41637.22	43167.32
468	3000	50	7	39363.28	41072.80
469	3000	50	7	46751.73	36565.79
470	3000	50	7	46279.67	40245.46
471	3000	50	7	39414.35	39929.45
472	3000	50	7	41555.74	44082.20
473	3000	50	7	42813.83	42288.17
474	3000	50	7	48214.49	41019.33
475	3000	50	7	42085.03	39981.75
476	3000	50	7	50281.73	40836.59
477	3000	50	7	43539.11	41749.29
478	3000	50	7	45480.13	42917.72
479	3000	50	7	40437.08	43990.45
480	3000	50	7	45245.00	37738.60
481	500	75	7	56064.40	55939.55
482	500	75	7	55725.23	52761.11

483	500	75	7	59824.76	54524.75
484	500	75	7	59822.08	52779.85
485	500	75	7	58608.46	56905.96
486	500	75	7	58427.45	53571.68
487	500	75	7	54108.56	56114.58
488	500	75	7	55205.37	52696.36
489	500	75	7	58408.53	54864.17
490	500	75	7	58602.56	56194.94
491	500	75	7	56992.18	54689.44
492	500	75	7	57468.89	54476.15
493	500	75	7	51661.64	51466.11
494	500	75	7	56893.73	56781.26
495	500	75	7	58471.74	55886.78
496	500	75	7	51068.79	53291.67
497	500	75	7	56359.10	56602.91
498	500	75	7	52080.90	53100.47
499	500	75	7	60072.51	58973.21
500	500	75	7	59631.15	54771.90
501	500	75	7	56969.73	53550.48
502	500	75	7	59230.78	55577.27
503	500	75	7	57068.86	56231.39
504	500	75	7	66057.09	52871.05
505	500	75	7	56937.75	51281.34
506	500	75	7	55555.84	56020.22
507	500	75	7	59559.43	51297.40
508	500	75	7	55785.83	58175.35
509	500	75	7	61906.27	56618.50
510	500	75	7	54813.29	53902.68
511	1000	75	7	45870.86	45372.76
512	1000	75	7	50909.33	46224.99
513	1000	75	7	43531.11	47931.37
514	1000	75	7	47750.85	44910.51
515	1000	75	7	45478.08	44784.77
516	1000	75	7	41693.55	45450.83
517	1000	75	7	49861.58	41391.54
518	1000	75	7	48727.22	47512.72
519	1000	75	7	42830.68	42405.00
520	1000	75	7	49993.08	45159.10
521	1000	75	7	47248.53	44221.11
522	1000	75	7	49912.84	50068.37
523	1000	75	7	54439.10	49262.63

524	1000	75	7	48816.77	49469.57
525	1000	75	7	49613.20	44191.61
526	1000	75	7	51288.40	48149.01
527	1000	75	7	47674.85	45287.78
528	1000	75	7	44031.90	51274.38
529	1000	75	7	48826.67	44083.12
530	1000	75	7	52600.76	46167.40
531	1000	75	7	51988.04	48764.35
532	1000	75	7	48026.09	43993.82
533	1000	75	7	54033.14	45841.15
534	1000	75	7	54031.95	49459.34
535	1000	75	7	51981.35	49104.34
536	1000	75	7	47385.21	43729.21
537	1000	75	7	47151.35	54479.19
538	1000	75	7	49230.29	51200.55
539	1000	75	7	53347.79	46207.85
540	1000	75	7	51561.58	50953.63
541	2000	75	7	41015.73	39686.43
542	2000	75	7	46397.95	40625.45
543	2000	75	7	44410.80	38113.60
544	2000	75	7	46963.50	36469.37
545	2000	75	7	39396.16	45834.84
546	2000	75	7	42167.87	37872.09
547	2000	75	7	46131.87	42760.56
548	2000	75	7	41564.20	38108.53
549	2000	75	7	47615.89	37705.99
550	2000	75	7	44849.62	40457.66
551	2000	75	7	44374.79	45711.18
552	2000	75	7	43553.66	41457.20
553	2000	75	7	50856.17	39463.90
554	2000	75	7	44652.99	41613.36
555	2000	75	7	42657.24	42208.66
556	2000	75	7	45507.46	42218.94
557	2000	75	7	42955.76	42584.80
558	2000	75	7	45982.80	35314.99
559	2000	75	7	43854.75	44172.94
560	2000	75	7	43614.02	40281.00
561	2000	75	7	46044.88	43701.28
562	2000	75	7	44755.35	41484.53
563	2000	75	7	43052.13	40605.22
564	2000	75	7	40114.71	37481.56



565	2000	75	7	41290.20	43975.64
566	2000	75	7	45142.95	39024.29
567	2000	75	7	45159.63	39758.81
568	2000	75	7	41262.22	42221.92
569	2000	75	7	40926.88	38899.80
570	2000	75	7	40544.23	41706.68
571	3000	75	7	40418.69	38013.52
572	3000	75	7	42647.00	44338.59
573	3000	75	7	45535.68	39976.65
574	3000	75	7	41157.95	36812.04
575	3000	75	7	36025.82	39116.39
576	3000	75	7	42603.63	36723.84
577	3000	75	7	43087.18	37651.21
578	3000	75	7	41550.33	36856.21
579	3000	75	7	43651.15	41974.38
580	3000	75	7	38896.36	41169.50
581	3000	75	7	36042.25	39443.53
582	3000	75	7	41975.09	42513.90
583	3000	75	7	38303.08	40364.70
584	3000	75	7	36233.78	34826.31
585	3000	75	7	35554.55	36345.50
586	3000	75	7	44417.13	40704.04
587	3000	75	7	42821.52	38258.60
588	3000	75	7	37905.75	44350.66
589	3000	75	7	46458.48	35068.67
590	3000	75	7	37913.47	33380.89
591	3000	75	7	44330.33	33337.68
592	3000	75	7	41468.24	42239.71
593	3000	75	7	40212.91	40831.19
594	3000	75	7	42098.94	34757.32
595	3000	75	7	37081.07	42207.54
596	3000	75	7	41577.53	36357.85
597	3000	75	7	32975.38	39631.58
598	3000	75	7	41395.40	33576.57
599	3000	75	7	42320.03	39242.74
600	3000	75	7	40884.83	40060.64
601	500	100	7	51136.52	50064.85
602	500	100	7	52478.83	48864.39
603	500	100	7	53824.92	48597.48
604	500	100	7	53445.03	50304.10
605	500	100	7	51190.92	47870.15

606	500	100	7	60402.95	48944.33
607	500	100	7	55701.29	51568.99
608	500	100	7	57578.49	46972.72
609	500	100	7	51652.94	51828.10
610	500	100	7	47837.02	51935.87
611	500	100	7	54120.27	53680.64
612	500	100	7	50273.17	52988.34
613	500	100	7	54973.98	53433.34
614	500	100	7	52658.74	47032.55
615	500	100	7	52273.72	46950.67
616	500	100	7	54107.39	46455.80
617	500	100	7	55787.36	52175.77
618	500	100	7	56710.16	50222.58
619	500	100	7	52950.99	45720.23
620	500	100	7	54197.66	50253.05
621	500	100	7	59808.39	45546.74
622	500	100	7	57116.91	50494.74
623	500	100	7	62077.40	48883.03
624	500	100	7	53423.94	48210.68
625	500	100	7	53930.10	49566.47
626	500	100	7	51831.95	51907.67
627	500	100	7	53704.41	51739.57
628	500	100	7	51168.88	51310.37
629	500	100	7	50933.85	51439.20
630	500	100	7	52273.36	55360.58
631	1000	100	7	49358.68	41860.76
632	1000	100	7	42883.85	40744.96
633	1000	100	7	45167.99	42936.00
634	1000	100	7	41524.68	41918.77
635	1000	100	7	45272.41	43243.88
636	1000	100	7	39614.34	41201.47
637	1000	100	7	41361.79	43486.86
638	1000	100	7	48143.77	41800.28
639	1000	100	7	44693.04	46006.41
640	1000	100	7	48251.01	45314.81
641	1000	100	7	47275.21	43800.98
642	1000	100	7	43330.42	41647.98
643	1000	100	7	51176.13	41837.85
644	1000	100	7	48559.32	48290.00
645	1000	100	7	49138.99	43890.01
646	1000	100	7	51889.52	41361.19

647	1000	100	7	42870.83	42341.53
648	1000	100	7	48803.61	42044.85
649	1000	100	7	42280.06	42301.75
650	1000	100	7	42636.90	40130.41
651	1000	100	7	41336.47	41236.45
652	1000	100	7	50418.96	43601.53
653	1000	100	7	47561.02	43460.34
654	1000	100	7	49701.31	41112.62
655	1000	100	7	40354.63	47681.29
656	1000	100	7	47229.16	42675.33
657	1000	100	7	41337.93	41868.06
658	1000	100	7	49780.35	43018.61
659	1000	100	7	47696.72	40194.48
660	1000	100	7	45996.51	43180.00
661	2000	100	7	42026.69	42758.53
662	2000	100	7	40721.50	37515.15
663	2000	100	7	44894.95	42306.61
664	2000	100	7	48061.82	44806.70
665	2000	100	7	39870.99	37908.92
666	2000	100	7	38396.37	38035.80
667	2000	100	7	42532.94	38599.18
668	2000	100	7	39675.77	37472.49
669	2000	100	7	37977.29	45003.17
670	2000	100	7	38137.91	40597.58
671	2000	100	7	37430.44	34995.71
672	2000	100	7	44048.98	42978.52
673	2000	100	7	41199.76	41325.67
674	2000	100	7	41974.29	43246.72
675	2000	100	7	38914.40	40661.49
676	2000	100	7	43766.26	38888.22
677	2000	100	7	42810.11	41805.19
678	2000	100	7	40336.47	36767.02
679	2000	100	7	47033.00	42066.61
680	2000	100	7	39923.66	41809.08
681	2000	100	7	40498.25	40820.74
682	2000	100	7	39151.74	39142.94
683	2000	100	7	46404.47	37506.65
684	2000	100	7	40992.36	41018.76
685	2000	100	7	44386.88	36263.59
686	2000	100	7	38091.65	43819.20
687	2000	100	7	45994.38	40046.53

688	2000	100	7	39437.82	38667.43
689	2000	100	7	43467.12	41338.24
690	2000	100	7	35062.06	39799.73
691	3000	100	7	37902.87	36633.64
692	3000	100	7	37901.05	36216.54
693	3000	100	7	38376.89	36444.39
694	3000	100	7	35176.19	41746.90
695	3000	100	7	37159.49	37813.21
696	3000	100	7	34156.68	37978.08
697	3000	100	7	36855.27	35950.33
698	3000	100	7	37649.10	31614.70
699	3000	100	7	48082.97	35502.43
700	3000	100	7	38145.16	39397.94
701	3000	100	7	37001.76	37664.48
702	3000	100	7	39078.52	37450.86
703	3000	100	7	38000.08	39371.16
704	3000	100	7	48229.16	34880.88
705	3000	100	7	39757.63	39301.74
706	3000	100	7	40055.47	34670.92
707	3000	100	7	37951.38	35354.17
708	3000	100	7	41144.92	38066.27
709	3000	100	7	36811.02	35701.30
710	3000	100	7	43259.93	39210.46
711	3000	100	7	40104.29	34125.71
712	3000	100	7	40235.61	34543.84
713	3000	100	7	36747.30	36084.60
714	3000	100	7	41048.87	32660.96
715	3000	100	7	42141.48	38055.01
716	3000	100	7	40138.97	40367.35
717	3000	100	7	43459.06	37855.71
718	3000	100	7	41274.17	38264.33
719	3000	100	7	40615.72	41483.35
720	3000	100	7	36910.51	35526.09
721	500	50	9	54571.78	58172.99
722	500	50	9	55286.79	56280.02
723	500	50	9	63439.45	62103.53
724	500	50	9	64440.86	56530.53
725	500	50	9	62804.13	54122.07
726	500	50	9	55693.00	50523.36
727	500	50	9	61591.67	58876.92
728	500	50	9	60575.54	50425.06

729	500	50	9	55650.70	60623.83
730	500	50	9	62623.03	64292.96
731	500	50	9	56787.86	60622.25
732	500	50	9	61062.77	59369.82
733	500	50	9	60084.95	55842.89
734	500	50	9	66497.43	54633.79
735	500	50	9	63721.76	56884.92
736	500	50	9	61010.40	57453.38
737	500	50	9	54685.22	56212.67
738	500	50	9	59275.72	62846.12
739	500	50	9	59325.33	56192.34
740	500	50	9	57797.97	55124.12
741	500	50	9	62310.78	58799.90
742	500	50	9	58052.50	55531.93
743	500	50	9	57645.59	54561.61
744	500	50	9	65130.96	55638.91
745	500	50	9	63981.99	51763.79
746	500	50	9	64398.87	54261.19
747	500	50	9	59729.38	58649.33
748	500	50	9	55001.73	54275.92
749	500	50	9	56960.66	59401.21
750	500	50	9	54259.98	61530.19
751	1000	50	9	49403.75	49845.59
752	1000	50	9	48495.35	47266.92
753	1000	50	9	46381.25	54208.09
754	1000	50	9	54053.71	48120.40
755	1000	50	9	50258.38	47771.97
756	1000	50	9	55753.78	47794.16
757	1000	50	9	47779.91	47688.75
758	1000	50	9	45910.48	52884.74
759	1000	50	9	55437.79	50566.30
760	1000	50	9	51680.75	44038.00
761	1000	50	9	56779.16	48104.95
762	1000	50	9	55804.31	48585.97
763	1000	50	9	48639.35	48890.95
764	1000	50	9	46390.26	53023.84
765	1000	50	9	54667.68	50298.90
766	1000	50	9	43395.40	51925.55
767	1000	50	9	50052.13	49442.06
768	1000	50	9	50713.62	53567.38
769	1000	50	9	52554.59	50587.33



770	1000	50	9	54226.38	49061.54
771	1000	50	9	56712.43	53565.49
772	1000	50	9	56127.47	51348.62
773	1000	50	9	47427.24	51620.70
774	1000	50	9	51118.07	48953.09
775	1000	50	9	49826.34	46621.46
776	1000	50	9	49266.17	49161.22
777	1000	50	9	53798.30	46724.82
778	1000	50	9	50651.74	52208.64
779	1000	50	9	58886.23	50581.83
780	1000	50	9	56488.31	53586.97
781	2000	50	9	44178.87	40844.87
782	2000	50	9	44427.57	46894.26
783	2000	50	9	43735.78	48215.72
784	2000	50	9	47299.79	47132.82
785	2000	50	9	49725.15	46175.70
786	2000	50	9	43904.08	44910.41
787	2000	50	9	48675.59	40069.87
788	2000	50	9	43262.05	44927.53
789	2000	50	9	47743.71	49793.52
790	2000	50	9	50073.84	47108.48
791	2000	50	9	44201.42	43121.79
792	2000	50	9	36798.56	45237.24
793	2000	50	9	46343.99	48527.05
794	2000	50	9	40397.82	39749.05
795	2000	50	9	48324.74	39249.51
796	2000	50	9	48654.89	45955.42
797	2000	50	9	46583.35	45857.30
798	2000	50	9	43689.83	49484.07
799	2000	50	9	47379.89	42443.14
800	2000	50	9	51570.50	47162.08
801	2000	50	9	41028.15	37640.84
802	2000	50	9	44947.79	38950.64
803	2000	50	9	45925.57	43245.34
804	2000	50	9	45743.97	46140.46
805	2000	50	9	44626.86	43401.33
806	2000	50	9	48026.41	41909.70
807	2000	50	9	49208.34	44008.90
808	2000	50	9	47224.00	43144.02
809	2000	50	9	47542.81	49975.77
810	2000	50	9	49781.04	43234.37

811	3000	50	9	40949.31	48324.75
812	3000	50	9	42503.49	44878.49
813	3000	50	9	45119.85	43736.43
814	3000	50	9	46380.46	40389.70
815	3000	50	9	40975.18	39888.18
816	3000	50	9	39876.13	38867.98
817	3000	50	9	36668.27	45257.12
818	3000	50	9	47278.39	44385.25
819	3000	50	9	46296.81	41829.40
820	3000	50	9	41972.52	39117.60
821	3000	50	9	39145.62	44619.34
822	3000	50	9	41879.07	45108.97
823	3000	50	9	42195.00	39468.32
824	3000	50	9	39972.74	40358.13
825	3000	50	9	39694.76	39390.79
826	3000	50	9	43527.83	46783.96
827	3000	50	9	44975.86	40665.15
828	3000	50	9	44680.38	43308.41
829	3000	50	9	39865.52	44632.67
830	3000	50	9	47910.94	37992.95
831	3000	50	9	40328.21	42928.76
832	3000	50	9	42011.49	41131.21
833	3000	50	9	44855.13	37117.74
834	3000	50	9	41248.93	39943.66
835	3000	50	9	46627.89	43497.36
836	3000	50	9	35951.89	37732.63
837	3000	50	9	43210.27	41828.50
838	3000	50	9	41973.06	39049.03
839	3000	50	9	44749.32	37982.69
840	3000	50	9	45444.77	43215.74
841	500	75	9	61437.07	55588.93
842	500	75	9	61390.08	49246.35
843	500	75	9	56518.73	57811.88
844	500	75	9	54601.96	58849.33
845	500	75	9	63006.28	58708.15
846	500	75	9	55358.72	49118.02
847	500	75	9	49366.83	50475.21
848	500	75	9	56502.96	53921.85
849	500	75	9	53958.82	57359.43
850	500	75	9	54110.75	57668.37
851	500	75	9	56205.99	49978.27

852	500	75	9	52829.10	48140.89
853	500	75	9	54259.43	57676.75
854	500	75	9	58709.80	50368.67
855	500	75	9	53138.14	54526.01
856	500	75	9	58016.68	51143.82
857	500	75	9	62367.50	54751.15
858	500	75	9	59150.80	54271.38
859	500	75	9	58643.59	55867.85
860	500	75	9	53809.38	52854.98
861	500	75	9	57335.61	54796.66
862	500	75	9	61267.00	56647.15
863	500	75	9	55280.96	56693.36
864	500	75	9	60612.70	56215.99
865	500	75	9	59907.91	52843.18
866	500	75	9	65039.08	49802.95
867	500	75	9	56441.58	55124.24
868	500	75	9	58594.98	52906.88
869	500	75	9	52410.65	53705.70
870	500	75	9	53029.98	52707.35
871	1000	75	9	54217.21	47400.88
872	1000	75	9	49186.31	52875.24
873	1000	75	9	44476.78	43383.11
874	1000	75	9	44364.70	43258.96
875	1000	75	9	47519.18	49914.66
876	1000	75	9	49474.15	49896.17
877	1000	75	9	44835.75	49697.71
878	1000	75	9	48199.64	51584.77
879	1000	75	9	47276.08	44779.89
880	1000	75	9	51411.80	43760.75
881	1000	75	9	48273.70	45506.87
882	1000	75	9	49471.65	44029.90
883	1000	75	9	49514.94	45076.88
884	1000	75	9	47482.06	49234.18
885	1000	75	9	44612.85	48331.37
886	1000	75	9	51644.60	47426.09
887	1000	75	9	49324.40	49535.09
888	1000	75	9	50682.60	41099.82
889	1000	75	9	48891.69	47281.73
890	1000	75	9	47929.20	49776.02
891	1000	75	9	51523.30	50635.81
892	1000	75	9	49455.87	50583.12

893	1000	75	9	46294.53	50413.23
894	1000	75	9	46226.11	44361.26
895	1000	75	9	49669.88	46048.87
896	1000	75	9	45029.47	45877.33
897	1000	75	9	51383.01	46762.32
898	1000	75	9	47425.80	43043.58
899	1000	75	9	48512.88	54268.21
900	1000	75	9	49540.38	53503.38
901	2000	75	9	44553.67	38923.54
902	2000	75	9	50499.68	42244.95
903	2000	75	9	42165.37	44239.43
904	2000	75	9	42757.94	39037.87
905	2000	75	9	40254.77	43587.50
906	2000	75	9	39031.00	41976.78
907	2000	75	9	40691.79	37610.92
908	2000	75	9	43052.21	41713.84
909	2000	75	9	48836.09	49982.05
910	2000	75	9	41641.11	36910.14
911	2000	75	9	45301.46	41291.10
912	2000	75	9	38163.17	41079.68
913	2000	75	9	45721.81	45935.70
914	2000	75	9	47090.00	47440.97
915	2000	75	9	49587.37	38561.12
916	2000	75	9	43068.95	40514.41
917	2000	75	9	41145.73	40302.06
918	2000	75	9	40766.18	43059.94
919	2000	75	9	43677.88	41938.76
920	2000	75	9	46096.36	41508.10
921	2000	75	9	44531.91	38409.73
922	2000	75	9	40135.18	44396.23
923	2000	75	9	43169.77	43262.38
924	2000	75	9	41076.39	36561.36
925	2000	75	9	43528.83	41288.06
926	2000	75	9	40409.56	39925.09
927	2000	75	9	50408.76	47570.29
928	2000	75	9	45211.59	38325.99
929	2000	75	9	37244.31	38703.83
930	2000	75	9	43790.05	38935.97
931	3000	75	9	40543.82	38782.96
932	3000	75	9	43470.79	41818.11
933	3000	75	9	41825.42	37209.48

934	3000	75	9	40480.79	41278.51
935	3000	75	9	41530.93	33855.01
936	3000	75	9	37329.90	41644.45
937	3000	75	9	41644.61	38947.77
938	3000	75	9	45403.18	36380.99
939	3000	75	9	42784.23	36806.05
940	3000	75	9	42880.83	42046.03
941	3000	75	9	39451.43	40159.62
942	3000	75	9	40101.13	37700.93
943	3000	75	9	42770.44	40085.46
944	3000	75	9	41944.89	44700.46
945	3000	75	9	36375.23	41525.64
946	3000	75	9	42475.67	38458.77
947	3000	75	9	41320.18	38161.38
948	3000	75	9	40581.63	42241.89
949	3000	75	9	45131.87	37714.55
950	3000	75	9	37541.76	44065.28
951	3000	75	9	42699.06	40072.40
952	3000	75	9	40058.86	39015.02
953	3000	75	9	38731.78	38019.15
954	3000	75	9	42904.95	38810.96
955	3000	75	9	42442.29	37181.12
956	3000	75	9	45668.47	37026.57
957	3000	75	9	43918.20	39922.90
958	3000	75	9	41514.83	44354.57
959	3000	75	9	41277.45	40998.94
960	3000	75	9	42167.04	40273.82
961	500	100	9	51154.25	46512.30
962	500	100	9	53723.93	42861.75
963	500	100	9	57011.36	53027.59
964	500	100	9	56795.84	48815.03
965	500	100	9	51071.94	49071.81
966	500	100	9	58367.88	45851.27
967	500	100	9	58635.97	43767.04
968	500	100	9	48314.09	50402.24
969	500	100	9	52289.32	49768.49
970	500	100	9	52205.87	46691.27
971	500	100	9	55232.48	47427.09
972	500	100	9	50355.42	56937.34
973	500	100	9	52507.58	49032.15
974	500	100	9	45301.12	48273.46



975	500	100	9	57310.95	54387.92
976	500	100	9	53729.06	51983.41
977	500	100	9	51069.66	51694.00
978	500	100	9	53501.69	47994.08
979	500	100	9	52372.40	44988.53
980	500	100	9	56612.84	51078.42
981	500	100	9	59760.51	51685.33
982	500	100	9	44464.11	47415.71
983	500	100	9	55419.79	46945.06
984	500	100	9	53357.13	46405.58
985	500	100	9	49020.66	51691.21
986	500	100	9	54760.02	46635.66
987	500	100	9	58041.18	49464.87
988	500	100	9	49495.21	47607.12
989	500	100	9	50757.41	55175.85
990	500	100	9	57930.10	46025.01
991	1000	100	9	44293.63	45667.51
992	1000	100	9	50575.01	46505.22
993	1000	100	9	40124.80	42293.28
994	1000	100	9	44325.12	46161.93
995	1000	100	9	40433.81	48523.16
996	1000	100	9	44456.09	41173.84
997	1000	100	9	48735.74	41150.23
998	1000	100	9	45597.07	41525.30
999	1000	100	9	47167.93	46702.52
1000	1000	100	9	45292.89	45323.07
1001	1000	100	9	40933.00	41577.86
1002	1000	100	9	41195.56	39367.98
1003	1000	100	9	49474.30	40733.30
1004	1000	100	9	47215.84	43323.78
1005	1000	100	9	43494.19	44845.74
1006	1000	100	9	45880.81	41800.23
1007	1000	100	9	46992.63	42255.61
1008	1000	100	9	39190.50	41793.56
1009	1000	100	9	44014.91	42448.40
1010	1000	100	9	45184.73	40759.62
1011	1000	100	9	49640.55	39661.60
1012	1000	100	9	44700.88	44182.36
1013	1000	100	9	42686.46	45554.97
1014	1000	100	9	46149.54	44302.98
1015	1000	100	9	46604.29	42767.22

1016	1000	100	9	45952.28	41521.81
1017	1000	100	9	45375.57	41582.06
1018	1000	100	9	42930.93	45348.59
1019	1000	100	9	48398.06	45582.16
1020	1000	100	9	45634.71	44914.60
1021	2000	100	9	41136.42	37260.45
1022	2000	100	9	38188.77	41203.41
1023	2000	100	9	35994.15	37821.22
1024	2000	100	9	35874.00	44613.58
1025	2000	100	9	38652.65	40561.40
1026	2000	100	9	42407.13	37945.66
1027	2000	100	9	42113.88	35932.63
1028	2000	100	9	41176.22	42166.88
1029	2000	100	9	36456.72	40289.22
1030	2000	100	9	36599.05	40176.49
1031	2000	100	9	38898.69	40212.70
1032	2000	100	9	41655.10	40250.93
1033	2000	100	9	41469.73	39413.50
1034	2000	100	9	43051.00	33816.11
1035	2000	100	9	40632.77	39926.05
1036	2000	100	9	40205.08	40046.21
1037	2000	100	9	42567.92	34564.10
1038	2000	100	9	40805.82	38578.15
1039	2000	100	9	46072.88	43863.59
1040	2000	100	9	42944.04	36172.41
1041	2000	100	9	39485.78	38826.91
1042	2000	100	9	37242.50	40635.89
1043	2000	100	9	42549.18	40873.91
1044	2000	100	9	49445.07	39188.23
1045	2000	100	9	39578.32	40354.15
1046	2000	100	9	41710.94	40386.95
1047	2000	100	9	39489.48	37882.01
1048	2000	100	9	38952.94	35126.80
1049	2000	100	9	39466.52	35630.67
1050	2000	100	9	35943.15	40266.87
1051	3000	100	9	38654.98	38745.51
1052	3000	100	9	40327.51	38847.37
1053	3000	100	9	39551.85	39952.33
1054	3000	100	9	37888.78	40122.18
1055	3000	100	9	37579.92	41049.95
1056	3000	100	9	45208.71	33414.86

1057	3000	100	9	42973.49	36652.92
1058	3000	100	9	40707.60	36205.56
1059	3000	100	9	37635.59	40350.99
1060	3000	100	9	41252.65	36261.81
1061	3000	100	9	38902.46	40180.86
1062	3000	100	9	39505.58	37366.87
1063	3000	100	9	37366.61	38144.58
1064	3000	100	9	45521.63	37869.82
1065	3000	100	9	39638.12	34237.25
1066	3000	100	9	35402.29	40208.41
1067	3000	100	9	38019.19	37205.58
1068	3000	100	9	37395.14	34030.30
1069	3000	100	9	37903.69	37081.25
1070	3000	100	9	37523.24	42730.92
1071	3000	100	9	38449.24	37744.74
1072	3000	100	9	36650.49	38892.48
1073	3000	100	9	38223.58	40395.93
1074	3000	100	9	39575.04	35895.45
1075	3000	100	9	38628.77	38008.65
1076	3000	100	9	35644.08	37693.17
1077	3000	100	9	41877.34	40380.41
1078	3000	100	9	40374.37	38254.76
1079	3000	100	9	38909.08	35117.05
1080	3000	100	9	39913.51	31997.57

## GLOSARIO

---

**Algoritmos Genéticos.** Son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización. Están basados la Teoria de la Evolucion de las Especies propuesta por Charles Darwin y en los trabajos realizados por Gregor Mendel en el campo de la Genetica.

**Inteligencia Artificial.** Es la rama de las Ciencias de la Computación dedicada al desarrollo de agentes racionales no vivos.

**Computo Evolutivo.** Es una rama de la inteligencia artificial que involucra problemas de optimización combinatoria. Se inspira en los mecanismos de la Evolución biológica.

**Migracion.** Es todo desplazamiento de población que se produce desde un lugar de origen a otro destino y lleva consigo un cambio de la residencia habitual en el caso de las personas o del hábitat en el caso de las especies animales migratorias.

**Inmigracion.** Es la entrada a un país de personas que nacieron o proceden de otro lugar.

**TSP.** El problema del viajante de comercio o TSP por sus siglas en inglés, es uno de los problemas más famosos en el campo de la optimización combinatoria computacional. A pesar de la aparente sencillez de su planteamiento, el TSP es uno de los más complejos de resolver y existen demostraciones que equiparan la complejidad de su solución a la de otros problemas aparentemente mucho más complejos que han retado a los matemáticos desde hace siglos.

**Genética de Poblaciones.** Es la rama de la genética cuya problematica es describir la variación y distribución biológica, con el objeto de dar explicación a fenómenos evolutivos.

## BIBLIOGRAFIA

---

Affenzeller M., Winkler S., Wager S. “*Evolutionary Systems Identification: New Algorithmic Concepts and Applications*”. Advances in Evolutionary Algorithms I-Tech 2008.

Bäck T. “*Evolutionary Algorithms in Theory and Practice*”. Oxford University Press. 1996.

Balman A. “*Applying Parallel Genetic Algorithms to Economic Problems: The Case of Agricultural Land Markets*”. IIFET2000.

Barbadilla A., “*La Genética de Poblaciones*”, Universidad Autónoma de Barcelona. Departamento de Genética y Microbiología. 2008. [http://bioinformatica.uab.es/divulgacio/la\\_genetica\\_de\\_poblaciones.pdf](http://bioinformatica.uab.es/divulgacio/la_genetica_de_poblaciones.pdf)

Berlanga A., Patricio M., Garcia J., Molina J. “*Solving Combinatorial Problems with TimeConstraints using Estimation of Distribution Algorithms and Their Application in Video-Tracking Systems*”. New Achievements in Evolutionary Computation. INTECH 2010.

Blasco A. “*Seis Horas de Genética de Poblaciones*”. Universidad Politécnica de Valencia. 2008.

Brest J., Zerovnik J. “*A Heuristic for the Asymmetric Traveling Salesman Problem*”. The 6th Metaheuristics International Conference. 2005.

Bryant K. y Benjamin A. “*Genetic Algorithms and the Traveling Salesman Problem*”. Harvey Mudd College. 2000.

Buriol L., Franca P., Moscato P. “*A New Memetic Algorithm for the Asymmetric Traveling Salesman Problem*”. Journal of Heuristics. Kluwer Academic Publishers. 2004

Caraviello D. “*Cruzamiento en Ganado Lechero*”. Instituto Babcock. Universidad de Wisconsin. 2004.



Cong S., Jia Y., Deng K. "*Particle Swarm and Ant Colony Algorithms and Their Applications in Chinese Traveling Salesman Problem*". New Achievements in Evolutionary Computation. INTECH 2010.

Darwin C. "*On the Origin of Species by Means of Natural Selection*". Murray, London. 1859.

Dorigo, M. y T. Stützle. "*ACO Algorithms for the Traveling Salesman Problem, Evolutionary Algorithms in Engineering and Computer Science: Recent Advances in Genetic Algorithms, Evolution Strategies, Evolutionary Programming, Genetic Programming and Industrial Applications*", John Wiley & Sons. 1999.

Fogel D. "*An Evolutionary Approach to the Traveling Salesman Problem*". Biological Cybernetics. Springer-Verlag. 1988.

Fogel D. "*Evolutionary Computation. Toward a New Philosophy of Machine Intelligence*". The Institute of Electrical and Electronic Engineers. 1995.

Fogel L., Owens A., Walsh M. "*Artificial Intelligence through Simulated Evolution*". John Wiley & Sons, Inc. 1966.

Futuyma D. "*La Selección Natural: Como funciona la Evolucion*". Entrevista original de ActionBioscience.org. 2004. <http://www.actionbioscience.org/esp/evolucion/futuyma.html>.

Goldberg D. "*Genetic Algorithms in Search, Optimization, and Machine Learning*". Addison Wesley. 1989.

Gomez A., Leal R. "*Implementación de algoritmos genéticos paralelos de grano burdo en redes locales de computadoras*". Congreso Nacional de Instrumentación. Memorias SOMI XV. Guadalajara - Jalisco, México. 2000.

Grant R. "*Approximating the Traveling Salesman Problem with Genetic Algorithms*". 2005.

Griffiths A., Wessler S., Lewontin R., Gelbart W., Suzuki D., Miller J. *“An Introduction to Genetic Analysis”* Freeman 2000.

Gutin G. *“The Travelling Salesman Problem”*. Department of Computer Science, Royal Holloway, University of London. MSC2000.

Gutin G., Glover F. *“Further Extension of the TSP Assign Neighborhood”*. Journal of Heuristics ACM. 2005.

Hidalgo J., Prieto M., Lanchares J., Hermida R. *“A Parallel Genetic Algorithm for solving the Partitioning Problem in Multi-FPGA systems”*. Proceedings of 3rd international Meeting on vector and parallel processing. Portugal 1998.

Hincapié R., Ríos C., Gallegos R. *“Técnicas Heurísticas Aplicadas al Problema del Cartero Viajante (TSP)”*. Scientia et Technica. 2004.

Hinterding, R., Michalewicz, Z., Eiben, A. *“Adaptation in Evolutionary Computation: A Survey”*. Proceedings of the 4th IEEE International Conference on Evolutionary Computation. 1997.

Hoffmann A. *“Arguments on Evolution: A Paleontologist’s Perspective”*. Oxford University Press. 1989.

Holland J. *“Adaptation in Natural and Artificial Systems”*. The University of Michigan Press. 1975.

Holstein D., Moscato P. y Bria O. *“Una Metaheurística Co-evolutiva para el Problema del Viajante de Comercio”* Informe final de trabajo de grado. Universidad Nacional de la Plata. 1998.

Janssens F., Zhang L., Glänzel W. *“Hybrid Clustering for Validation and Improvement of Subject-Classification Schemes”*. Data Mining and Knowledge Discovery in Real Life Applications. I-Tech 2009

Larose D. *“Discovering Knowledge in Data: An Introduction to Data Mining”*. Chapter 1 *Intoduction To Data Mining*. John Wiley & Sons, Inc. 2005.

Londoño N. *“Algoritmos Genéticos”*. Universidad Nacional de Colombia. 2006.

Lu G., Areibi S. *“An Island-Based GA Implementation for VLSI Standard-Cell Placement”*. GECCO 2004. Springer-Verlag (2004).

Magofke J., Garcia X. *“Uso del Cruzamiento entre Razas para Mejorar la Productividad de Animales. VII. Evaluación de Razas Puras”*. 2003

Michalewicz, Z. *“A Perspective on Evolutionary Computation”*. Proceedings of the Workshop on Evolutionary Computation. 1994.

Michalewicz Z. *“Genetic Algorithms + Data Structures = Evolution Programs”*. Springer Verlag. 1996.

Michalewicz, Z., Dasgupta, D., Le Riche, R.G., and Schoenauer, M., *“Evolutionary Algorithms for Constrained Engineering”*. Problems Computers & Industrial Engineering Journal. 1996.

Michalewicz Z., Hinterding R., Michalewicz M. *“Evolutionary Algorithms”*. Kluwer Academic, 1997.

Moscato P. y Tinetti F. *“Blending Heuristics with a Population-Based Approach: A ‘Memetic’ Algorithm for the Traveling Salesman Problem”*, 1994.

Moscato, P., Días A., Glover F., Ghaziri H., González, Laguna M. y Tseng F. “*Optimización Heurística y Redes Neuronales*”. Editorial Paraninfo 1996.

Moujahid A., Inza I. y Larrañaga P. “*Heurísticos en Optimización Combinatorial*”. Universidad del País Vasco-Euskal Herriko Unibertsitatea. Departamento de Ciencias de la Computacion e Inteligencia Artificial. 2005.

Moujahid A., Inza I. y Larrañaga P. “*Algoritmos Genéticos*”. Universidad del País Vasco-Euskal Herriko Unibertsitatea. Departamento de Ciencias de la Computacion e Inteligencia Artificial. 2005.

Moujahid A. “*Métodos Matemáticos en Ciencias de la Computación*”. Universidad del País Vasco. 2007. <http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/temageneticos.pdf>

Nesmachnow S. “*Evolución en el diseño y clasificación de Algoritmos Genéticos Paralelos*”. XXVIII Conferencia Latinoamericana de Informática. Montevideo, Uruguay. 2002.

Nowostawski M., Poli R. “*Parallel Genetic Algorithm Taxonomy*”. Kes99. 1999.

Ochoa A., Hernandez A., Cruz L., Ponce J., Montes F., Li L., Janacek L. “*Artificial Societies and Social Simulation using Ant Colony, Particle Swarm Optimization and Cultural Algorithms*”. New Achievements in Evolutionary Computation. INTECH2010.

Ornelas F., Padilla A., Padilla F., Ponce de León E., Ochoa A. “*Un Nuevo Operador Para Mejorar La Diversificación En Algoritmos Genéticos, Bioinspirado En Genética de Poblaciones*”. Tercer Congreso Internacional de Computación Evolutiva. COMCEV 2007.

Ornelas F., Padilla A., Padilla F., Ponce de León E., Ochoa A. “*La Migración como parte del proceso evolutivo en algoritmos genéticos*”. Cuarto Congreso Internacional de Computación Evolutiva. COMCEV 2008.



Ornelas F., Padilla A., Padilla F., Ponce de León E., Ochoa A. *“Genetic Algorithm using Migration and Modified GSX as Support”*. Hybrid Intelligent Systems Workshop. MICAI 2009.

Ponce J., Hernández A., Ochoa A., Padilla F., Padilla A., Álvarez F., Ponce de León E. *“Data Mining in WEB Applications”*. Data Mining and Knowledge Discovery in Real Life Applications. I-Tech 2009.

Reinelt, G. *“The Traveling Salesman. Computational Solutions for TSP Applications”*. Springer-Verlag, Computer Science Editorial I. 1994.

Reza M., Rahimi M., Shah-Hosseini H. *“Population-Based Optimization Algorithms for Solving the Travelling Salesman Problem”*. Travelling Salesman Problem. I-Tech 2008.

Riol J. *“Gregor Mendel, Las leyes de la herencia”*. Diario de Avisos Universidad de Laguna. 2008.

Santana L., Coello C. *“Una introducción a la computación evolutiva y algunas de sus aplicaciones en Economía y Finanzas”*. Revista de Métodos Cuantitativos para la Economía y la Empresa. 2006

Savla K., Bullo F., Frazzoli E. *“On Traveling Salesperson Problems for Dubins’ vehicle: stochastic and dynamic environments”*. CDC 2005.

Schwefel H. *“Numerical Optimization of Computer Models”*. Wiley, Chichester. 1981.

Seifert J. *“Data Mining: An Overview”*. Analyst in Information Science and Technology Policy. CRS Report for Congress. 2004.

Sengoku H., Yoshihara I. *“A Fast TSP Solver Using GA on JAVA”*. Systems Development Laboratory, Hitachi, Ltd. 1998.



Tzung –Pei H., Wen-Yang L., Shu-Min L., Jiann-Horng L. “*Dynamically Adjusting Migration Rates for Multi-Population Genetic Algorithms*”. Journal of Advanced Computational and Intelligent Informatics. 2007.

Valdez S., Hernandez A., Botello S. “*Efficient Estimation of Distribution Algorithms by using the Empirical Selection Distribution*”. New Achievements in Evolutionary Computation. INTECH 2010.

Vignolo L., Albornoz E. “*Bioingeniería I Algoritmos Genéticos en Paralelo*”. [http://bioingenieria1.wdfiles.com/local--files/descargas/bio1\\_AGparalelos.pdf](http://bioingenieria1.wdfiles.com/local--files/descargas/bio1_AGparalelos.pdf). 2007.

Verdejo A. “*Algoritmos Genéticos*”. 2006-2007.

Wei Y., Gu K. “*Parallel Search Strategies for TSPs using a Greedy Genetic Algorithm*”. Advances in Greedy Algorithms. I-Tech 2008.

Wikipedia1. “*Complejidad Computacional*”. [http://es.wikipedia.org/wiki/Complejidad\\_computacional](http://es.wikipedia.org/wiki/Complejidad_computacional).

Wikipedia2. “*Population genetics*”. [http://en.wikipedia.org/wiki/Population\\_genetics](http://en.wikipedia.org/wiki/Population_genetics).

Whitley D., Rana S., Heckendorn. “*The Island Model Genetic Algorithm: On Separability, Population Size and Convergence*”. Colorado State University. 1998.