

13
2ej.

007431



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESTADO DE ACATLAN
ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
ACATLAN
DIVISION DE MATEMATICAS E INGENIERIA
PROGRAMA DE MATEMATICAS APLICADAS
Y COMPUTACION

DEPARTAMENTO DE ESTUDIOS
PROFESIONALES
Y CREDITACION

“SIMULACION DE PROCESOS EVOLUTIVOS
MEDIANTE ALGORITMOS GENETICOS”

T E S I S

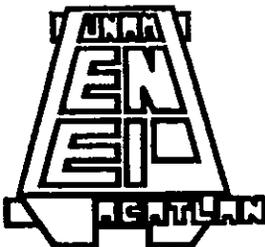
QUE PARA OBTENER EL TITULO DE
LICENCIADO EN MATEMATICAS APLICADAS

P R E S E N T A :

ALAN GARDUÑO RIVERA

ASESOR DE TESIS:

M. EN I. VICTOR PALENCIA GOMEZ



ACATLAN EDO. DE MEXICO

1998

TESIS CON
FALLA DE ORIGEN

259026



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES ACATLAN
DIVISION DE MATEMATICAS E INGENIERIA
PROGRAMA DE MATEMATICAS APLICADAS Y COMPUTACION

SR. ALAN GARDUÑO RIVERA
Alumno de la carrera de
Matemáticas Aplicadas y Computación,
P r e s e n t e .

De acuerdo a su solicitud presentada con fecha 27 de noviembre de 1997, me complace notificarle que esta Jefatura tuvo a bien asignarle el siguiente tema de tesis: "SIMULACION DE PROCESOS EVOLUTIVOS MEDIANTE ALGORITMOS GENETICOS", el cual se desarrollará como sigue:

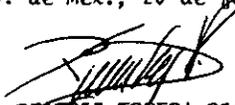
Introducción.

Capítulo 1.- Introducción a los Algoritmos Genéticos.
Capítulo 2.- Introducción y Ejemplos del AG Básico.
Capítulo 3.- Fundamentos Teóricos de los AG's.
Capítulo 4.- Limitaciones y Mejoras de Los AG's.
Capítulo 5.- Placeres Existenciales de los AG's.
Conclusiones.
Bibliografía.

Asimismo fue designado como Asesor de Tesis el M. en I. Victor - Palencia Gómez

Ruego a usted tomar nota que en cumplimiento de lo especificado en la Ley de Profesiones, deberá prestar servicio social durante un tiempo mínimo de seis meses como requisito básico para sustentar examen profesional, así como de la disposición de la Coordinación de la Administración Escolar en el sentido de que se imprima en lugar visible de los ejemplares de la tesis el título del trabajo realizado. Esta comunicación deberá imprimirse en el interior de la tesis.

A T E N T A M E N T E
"POR MI RAZA HABLARA EL ESPIRITU"
Acatlán, Edo. de Méx., 20 de febrero de 1998


LIC. BEATRIZ TRUEBA RÍOS
JEFE DEL PROGRAMA

BTR* yhm.

AGRADECIMIENTOS

Es increíble la enorme satisfacción que puede provocar un simple GRACIAS; no le niegues a nadie la oportunidad de escucharlo de ti.

A.G.R.

Quiero expresar mi más sincero testimonio de agradecimiento a todas aquellas personas que me ayudaron a la realización de este trabajo y, en especial, a todos aquellos que me han brindado su amistad y que de una forma u otra han dejado su huella a lo largo de mi vida. En particular quisiera expresárselo a las siguientes personas:

A Mamá y a Papá por darme la oportunidad de la vida, por apoyarme siempre, por quererme tanto y por haberme hecho quien soy. Sin ustedes esto no hubiera sido posible.

A mis abuelitas Esperanza y Rosa, porque han sido un ejemplo de esfuerzo y lucha continua. Mi cariño sincero para ustedes.

A mi abuelito Franz y a mi tío Ramón con especial cariño, a quienes estoy seguro les hubiese gustado ver llegar este momento, pero estoy cierto que, estén donde estén, me estarán viendo y se sentirán orgullosos.

A Aldo, Sean, Mayda, Liliana y Glen, mis hermanos, por compartir su vida conmigo, y porque, a pesar de todo, sé que cuentan conmigo y yo con ellos. A Karina, Anilú, Gaby y Wendy que forman parte ya de mi familia.

A mis tías y tíos que siempre han estado pendientes de mí: Cuca, Laura, Esperanza, Paty, Lupe, Arturo y Mario.

A Angie, por ser la culpable de la elección del tema, por la presión, a veces necesaria, para terminar el trabajo, pero especialmente por compartir los buenos y malos momentos de mi existencia, por que sé que siempre cuento contigo, por ser el motivo que me impulsa a seguir adelante y por todo el amor que me has dado.

Al Sr. Fernando y la Sra. María, por haberme aceptado como soy y porque sé que existe un gran aprecio mutuo.

Al Maestro en I. Víctor Palencia por sus valiosas sugerencias, por haberme asesorado y por haber sido mi mejor maestro de la licenciatura. Mi respeto y admiración hacia él.

A la Ingeniera Beatriz Clavel Díaz por sus comentarios, por su confianza en mí, por haberme brindado sus enseñanzas y especialmente por haber hecho, indirectamente, que eligiera este tema y no otro. Mi reconocimiento y respeto por su gran valía como profesional y como persona.

A mis compañeros de estudio, con quienes he tenido el gusto de convivir épocas maravillosas y quienes se han convertido en amigos entrañables: Elba de la Torre, Laura Castañeda, Lisset Baños, Fernando González, Jaime Vergara y Carlos Luna.

A mis excompañeros de trabajo, de quienes guardo un aprecio enorme y especialmente a quienes considero mis amigos: Blanca de la Rosa, Minerva Camacho y Vicente Hernández.

A mis actuales compañeros de trabajo, un especial agradecimiento por seguirme soportado hasta la fecha, por haber emprendido juntos un proyecto, que ha caminado, a pesar de ser 'virtual': "El Departamento de Redes y Soporte Técnico", y porque espero que, algún día, me consideren su amigo: Bety Estrada, Alex Rubio, Oscar Heras, Chucho Ruiz, Paco Manóatl, Mauricio Rico, Antolín Obregón y Agustín Lira.

Asimismo, vayan mis agradecimientos a compañeros de estudio y trabajo, y amigos de otras épocas, de quienes guardo gratos recuerdos: Claudia Aquino, Arturo de la Rosa, Magdalena Cuevas, Gerardo Rubio (El Lobo), Lorenzo Mendiola, Rubí Villegas, Guillermo Cabrera, Luisa Cortés, Heric Hernández, Areli Pineda, Erika Cervantes, Jorge Colín, Mario Espina, Pedro Álvarez, José Luis Medina y Ricardo Munguía.

Agradezco al Dr. Carlos Coello por sus especializados conocimientos y su influencia, que pueden encontrarse esparcidos a lo largo de todo este trabajo.

Si inadvertidamente dejé a alguien fuera de la lista, lo siento de verdad y, también, gracias.

No puedo dejar pasar esta oportunidad para agradecer a la UNAM por todo lo que me ha dado.

Finalmente mi infinita gratitud a DIOS.

PREFACIO

La selección del tema de tesis es un proceso que para unos es inmediato y no muy trascendente, al fin y al cabo constituye un trámite más, el último, pero al fin y al cabo, un trámite más. Sin embargo, para otros se convierte en una búsqueda persistente y en ocasiones más larga de lo deseado; el objetivo debe ser claro, no se trata de descubrir el hilo negro, pero sí de sentirse satisfecho con el trabajo realizado, con la esperanza de que se considere un documento útil y que sirva mínimamente como bibliografía para investigaciones posteriores.

El resultado de este trabajo está dirigido a todos aquellos interesados en conocer: qué son los Algoritmos Genéticos (AG's), cómo funcionan, para qué sirven, cómo implementarlos y que hay más allá de su cualidad como solucionadores de problemas, que los han convertido en un tema de gran atractivo.

Desafortunadamente en México la investigación sobre algoritmos genéticos es mínima. No obstante, los pocos que trabajan con ellos, son gente de gran capacidad y se espera que pronto contribuyan a difundirlos más y a que formen parte de los planes de estudio de diversas carreras. Por ahora solamente se estudian en maestrías, particularmente en el TEC de Monterrey, en la UAM y en el Politécnico. También existen algunas investigaciones realizadas en el IIMAS.

En cuanto a la bibliografía, en inglés es basta, pero el acceso es un tanto difícil, afortunadamente Internet ha roto barreras y permite conseguir mucha información al respecto; en español, prácticamente no existe, salvo algunas investigaciones que se han publicado en revistas científicas o especializadas.

La motivación para escribir sobre el tema nació al leer un artículo que mostraba las cualidades de los AG's como herramienta de optimización. Se trata de un método que utiliza modelos matemáticos y que a su vez necesita de la computación como herramienta. Identifiqué entonces que me encontraba ante algo novedoso y que reunía los elementos necesarios, así que decidí que ése sería mi tema de investigación.

La intención de este trabajo es mostrar y proponer; no presentar una aplicación, sino dar los elementos para que alguien más los aplique.

Para una implementación en computadora, se requieren conocimientos de programación; el pseudo-código presentado es suficiente para que se intente crear los programas, esto servirá para que experimenten y jueguen con los algoritmos genéticos; si prefieren ahorrarse el esfuerzo, pueden encontrar una gran cantidad de código en el Laboratorio de Algoritmos Genéticos de Illinois (IlliGAL) en <http://gal4.ge.uiuc.edu>.

Sólo me resta aclarar que no es suficiente leer el trabajo una vez para entenderlo, es necesario un mínimo de dos lecturas, crear los programas y jugar una y otra vez con las corridas de los mismos, y no es porque los algoritmos sean muy complicados, sino porque es difícil tratar de entender como funcionan. Bienvenido pues, al mundo de los AG's.

Alan Garduño R.
Febreo de 1998

+

El origen de la vida es un misterio, que sigamos sobreviviendo es otro misterio mayor aún.

Anónimo

Me temo que la principal conclusión que se desprende de la lectura de este libro, a saber, que el hombre descende de una forma orgánica de rango inferior, irritará grandemente a muchas personas. Sin embargo, no cabe de que somos la progenie evolucionada de criaturas primitivas. Jamás olvidare la sensación de pavor que me invadió al contemplar por vez primera a un grupo de fueguinos en una bravía y desolada zona costera, porque en seguida me vino a la mente la semejanza de nuestros antecesores con aquellos salvajes. Iban completamente desnudos, la piel pintarrajeada, el largo del cabello enmarañado y echaban espumarajos por la boca a causa de la turbación que experimentaban. Su semblante, sobresaltado y receloso, tenía expresión de ferocidad. No conocían arte ni oficio alguno y, cual alimañas, vivían de lo que cazaban. No poseían organización social y se mostraban implacables para con todos los que no pertenecieran a su reducida tribu. Quien haya visto a un salvaje en su entorno natural no sentirá excesivo rubor si se ve obligado a reconocer que por sus venas fluye la sangre de criaturas de un orden más bajo. En lo que a mí concierne, prefiero descender del heroico monito que se plantó ante su más temido enemigo para salvar la vida de su cuidador o del viejo babuino que descendió de la montaña y arrebató triunfalmente a un camarada más joven de los dientes de una sorprendida jauría que de un salvaje que halla placer en torturar a sus enemigos, que ofrece sacrificios sangrientos, comete infanticidios sin el menor escrúpulo, trata a sus mujeres como esclavas, no conoce el decoro y es víctima de las más necias supersticiones.

Al hombre se le puede disculpar que experimente cierto orgullo por haber escalado, aunque no con su esfuerzo, la cúspide de la jerarquía orgánica. Por otra parte, el hecho de haya ascendido a dicho puesto, de que no se encontrase en él desde un buen principio, le permite concebir esperanzas de alcanzar en un futuro lejano objetivos aún más encumbrados. Pero lo que ahora importa no son las esperanzas ni los temores, sino solamente la verdad, en la medida en que nuestra razón nos permita desvelarla. He procurado presentar pruebas recogidas lo mejor que he sabido, y en mi opinión, resulta forzoso reconocer que el hombre, a pesar de las nobles cualidades que le adornan, de la compasión que muestra hacia los más menesterosos, de su bondad no sólo para con los otros hombres, sino también para con las criaturas más insignificantes, de su intelecto divino y de que ha llegado a elucidar los movimientos y su constitución del sistema solar, a pesar de todo ello, digo, el hombre aún lleva impresa en su estructura corpórea la huella indeleble de su humilde origen.

Charles Darwin. El origen del hombre.



Mosaico II por M. C. Escher

INTRODUCCIÓN

Según Empédocles, las primeras formas de vida sobre la tierra fueron árboles. Sucedió mientras dominaba el amor. En su mundo recién creado, a los árboles se sumaron las primeras formas animales: una mezcla de partes corpóreas -torsos sin cabeza, criaturas de innumerables manos, cuerpos de toro y cabezas de hombres-. Algunas eran simples miembros con vida autónoma. Estas formas extrañas se combinaron lo mejor que pudieron, y con el tiempo solo sobrevivieron las formas más eficaces. Así en este aspecto, Empédocles se anticipó a la teoría de la supervivencia del más apto de Darwin.

Para entender con exactitud el alcance de este trabajo es necesario, en primera instancia, analizar el título: "Simulación de procesos evolutivos mediante algoritmos genéticos", para ello es menester considerar conceptos básicos tales como: simulación, procesos evolutivos y algoritmos genéticos.

La simulación permite investigar los modos de crear sistemas alternos que se asemejen con gran exactitud a un sistema real mediante modelos matemáticos¹. Una vez modelados, los sistemas se pueden simular sin afectar al sistema real, lo que permite observar diferentes comportamientos sin consecuencias funestas o costosas.

Para comprender qué son los procesos evolutivos será necesario adentrarnos al concepto de evolución. La evolución es un cambio gradual de los seres vivos que requiere mucho tiempo para producirse, a menudo millones de años. Los procesos evolutivos son pues, todos aquellos procesos que permiten que una especie evolucione, y no se

¹ Sistemas de Investigación de Operaciones – Aplicaciones de Matemáticas y Software. www.sio-ams.com (1997)

puede hablar de evolución de especies sin tener presente a Charles Darwin.

Charles Darwin fue el creador de la *teoría de la evolución* por selección natural. Dicha teoría formulada en "El origen de las especies" (1859) sostiene que los miembros de las especies que más se adaptan al medio son los que acaban sobreviviendo y se reproducen con más facilidad. Al tener que competir por el alimento, el refugio o la pareja sexual, los mejor adaptados salen beneficiados y se reproducen más. Darwin llamó a este proceso "selección natural". Por lo tanto, de una a otra generación las especies evolucionan y se adaptan mejor a su entorno.

Ahora bien, ¿qué significa simular procesos evolutivos? y ¿qué relación tienen con las matemáticas aplicadas y con la computación? En términos generales se ha mencionado que existen poblaciones de especies que, a través de varias generaciones, evolucionan adaptándose mejor a su entorno o, en resumidas cuentas, se trata de la supervivencia del más apto o del más fuerte. Esto permitió deducir que la evolución natural es un mecanismo que puede ser simulado.

Simular procesos evolutivos significa: crear modelos que simulen los procesos de la evolución a fin de encontrar las soluciones óptimas (más aptas) a un problema determinado. Para lograrlo se hará uso de los algoritmos genéticos, y para encontrar las soluciones a los problemas que se plantean es necesario implementarlos en computadoras.

En este trabajo se describen detenidamente las características de dichos algoritmos, se formulan algunas propuestas y se muestran algunas aplicaciones en diversos campos.

Los objetivos que se pretende alcanzar son:

- ❖ Realizar un estudio coherente y pormenorizado de los métodos y procedimientos propios de los algoritmos genéticos.
- ❖ Definir el rol que juegan los algoritmos genéticos como herramienta de optimización.
- ❖ Analizar los fundamentos matemáticos de los algoritmos genéticos.
- ❖ Mostrar las cualidades de los algoritmos genéticos, que los hacen tan atractivos.
- ❖ Proponer mejoras en las operaciones de los algoritmos genéticos para acelerar la convergencia y dar mayor diversidad en las poblaciones y en su caso mitigar los efectos de sus limitaciones.
- ❖ Analizar a los algoritmos genéticos más allá de simples solucionadores de problemas.

Para la consecución de los objetivos planteados, se utilizó la siguiente estructura:

- ❖ Capítulo 1. Introducción a los algoritmos genéticos.
- ❖ Capítulo 2. Implantación y ejemplos del AG básico.
- ❖ Capítulo 3. Fundamentos teóricos de los AG's.
- ❖ Capítulo 4. Limitaciones y mejoras de los AG's.
- ❖ Capítulo 5. Placeres existenciales de los AG's.

Se puede resumir el contenido de los capítulos de la siguiente manera:

- ❖ Capítulo 1: En él se presenta a los algoritmos genéticos, su definición, su estructura genérica, su historia, su relación intrínseca con la teoría de la evolución, así como una breve introducción a los

términos genéticos que permiten realizar una analogía con la naturaleza. También se describen los principios básicos y los criterios necesarios para su implantación.

- ❖ Capítulo 2: En él se describen la estructura del AG básico y los criterios necesarios para implantarlos. Lo anterior se ilustra mediante dos ejemplos prácticos. El primer problema es de optimización paramétrica multimodal, y el segundo un problema de asignación. Se trata, por una parte, de aclarar y concretar todo lo discutido en el capítulo 1, y por otra parte, de facilitar la comprensión del estudio teórico del siguiente capítulo.
- ❖ Capítulo 3: En él se realiza un análisis teórico de los algoritmos genéticos a fin de explicar su funcionamiento interno y justificar las razones de su eficiencia añadida. Se abordan los conceptos básicos para analizar sus fundamentos teóricos, se presenta el teorema fundamental de los algoritmos genéticos, y se explica porqué funcionan los AG's.
- ❖ Capítulo 4: En él se enumeran las limitaciones e inconvenientes que plantea el uso de algoritmos genéticos. Se explican detalladamente las diversas alternativas que existen para resolver o mitigar tales inconvenientes, así como también se presentan y proponen mejoras a los procedimientos y operadores.
- ❖ Capítulo 5: Por último, se hace un análisis de los AG's, desde distintos enfoques, se trata de ver a los algoritmos genéticos más allá de simples solucionadores de problemas.

Es deseo de quien esto escribe que este trabajo sirva como base para futuras investigaciones y aplicaciones, y que no quede olvidado en un estante, pues se ha tratado de plantearlo de la manera más completa y coherente que ha sido posible.

INDICE

| | |
|-----------------|-----|
| Agradecimientos | v |
| Prefacio | vii |
| Introducción | xi |

1

3

| | | | |
|---|-----------------------------|---|----|
| Introducción a los algoritmos genéticos. | 1.1 | Definición de algoritmo genético. | 4 |
| | 1.2 | Historia. | 5 |
| | 1.3 | Genética. | 6 |
| | 1.3.1 | La genética del sexo. | 8 |
| | 1.3.2 | Teoría de la evolución. | 10 |
| | 1.3.2.1 | Selección. | 10 |
| | 1.4 | Introducción a los algoritmos evolutivos. | 11 |
| | 1.5 | Principios básicos. | 12 |
| | 1.6 | Características intrínsecas de los AG's. | 16 |
| | 1.7 | Estructura y componentes básicos. | 17 |
| 1.7.1 | Analogía con la naturaleza. | 17 | |

| | | | |
|--|-------|--|----|
| | 1.7.2 | Estructura básica. | 18 |
| | 1.7.3 | Métodos y criterios necesarios para implantar un AG. | 19 |
| | 1.8 | Procedimientos básicos de un AG. | 21 |
| | 1.8.1 | Mecanismos de selección de individuos. | 21 |
| | 1.8.2 | Mecanismos de reproducción. | 23 |
| | 1.9 | Convergencia. | 26 |

2

29

| | | | |
|---|-------|--|----|
| Implantación y ejemplos del AG básico. | 2.1 | Criterios en la implantación. | 31 |
| | 2.2 | Ejemplo práctico de implantación de un AG. | 33 |
| | 2.3 | Problema de asignación. | 44 |
| | 2.3.1 | Resultados. | 47 |

3

49

| | | | |
|--|-----|---|----|
| Fundamentos teóricos de los AG's. | 3.1 | Definiciones y conceptos básicos. | 51 |
| | 3.2 | La ecuación de crecimiento de los esquemas. | 54 |
| | 3.3 | El teorema fundamental de los algoritmos genéticos. | 61 |

| | | |
|-------|--|----|
| 3.3.1 | Limitaciones prácticas del análisis teórico. | 64 |
|-------|--|----|

| | | |
|----------|--|-----------|
| 4 | | 67 |
|----------|--|-----------|

Mejoras y limitaciones de los AG's.

| | | |
|---------|---|----|
| 4.1 | El problema de la debilidad de los AG's. | 69 |
| 4.2 | El problema de la diversidad de los AG's. | 70 |
| 4.3 | Convergencia prematura por problemas con la diversidad. | 71 |
| 4.4 | El problema de la representación y la codificación. | 72 |
| 4.5 | Mejoras a los operadores genéticos. | 75 |
| 4.5.1 | Convergencia en los AG's y criterios de determinación. | 75 |
| 4.5.2 | Elitismo. | 77 |
| 4.5.2.1 | Reiniciación periódica. | 78 |
| 4.5.3 | Elección de las funciones de evaluación y de aptitud. | 79 |
| 4.5.3.1 | Desplazamiento y escalado de la función de evaluación. | 79 |
| 4.5.4 | Variantes del operador de cruce. | 81 |
| 4.5.4.1 | Alternativas al método de cruce clásico. | 82 |
| 4.5.5 | Alternativas a la mutación clásica. | 85 |
| 4.6 | Parametrización de los AG's. | 85 |

| | | | |
|--|-----|--|-----|
| | 4.7 | Hibridación. | 87 |
| | 4.8 | Algoritmos genéticos paralelos. | 88 |
| | 4.9 | Algoritmos genéticos diploides. | 90 |
| 5 | | | 95 |
| Placeres existenciales de los AG's. | 5.1 | AG's como solucionadores de problemas. | 98 |
| | 5.2 | AG's como una metodología efectiva de invención. | 99 |
| | 5.3 | AG's como las bases para sistemas de aprendizaje artificial. | 101 |
| | 5.4 | AG's como una filosofía guía. | 103 |
| | 5.5 | AG's como un modelo para todo. | 105 |
| | | Conclusiones | 108 |
| | | Glosario | 110 |
| | | Bibliografía | 112 |



J. Burgess/S.P.L./Cosmos

El dilema fundamental de la humanidad, el problema que subyace a todos los demás y que destaca por encima de ellos, es precisar qué puesto corresponde al hombre en la naturaleza y cuál es su relación con el cosmos. El origen de nuestra especie, los límites que condicionan nuestro influjo sobre la naturaleza y el de ésta sobre nosotros, el objetivo que pugnamos por alcanzar, son problemas que se presentan una y otra vez, con indeclinable vigor, a todos los seres vivos de la Tierra

T.H. Huxley, 1863

1.1 Definición de algoritmo genético.

Los algoritmos genéticos (en lo sucesivo AG's) son técnicas de búsqueda basadas en la teoría de la evolución de Darwin¹. Se les conoce también como métodos adaptativos los cuales pueden ser utilizados para resolver problemas de optimización. Los AG's se basan en los procesos genéticos de organismos biológicos, de acuerdo a los cuales a través de diversas generaciones y según los principios de selección natural "sobreviven los más aptos", es decir, aquéllos que son capaces de adaptarse a cambios que se puedan producir en su entorno. Como sabemos, estos cambios se efectúan en los genes (unidad básica de codificación de cada uno de los atributos de un ser vivo), y algunas características se heredan a sus descendientes en la reproducción sexual.

¹ En su obra *El Origen de las Especies* (1859).

Capítulo

1

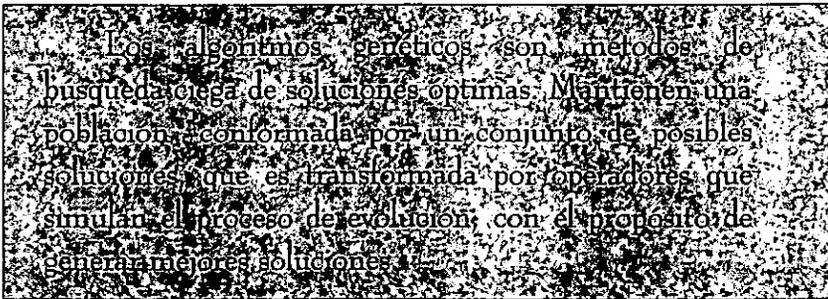
Introducción a los algoritmos genéticos.

Objetivos Particulares:

- Definir qué es un algoritmo genético.
- Apreciar la estructura y los componentes básicos de un AG.
- Analizar la relación intrínseca de los AG's y la teoría de la evolución.
- Establecer la analogía entre los AG's y la naturaleza.
- Reconocer los métodos y procedimientos generales para implantar un AG.

Los algoritmos genéticos son capaces de “evolucionar” soluciones en problemas reales, si son codificados acordemente. Por ejemplo, los AG's pueden ser usados para diseñar estructuras de puentes, para problemas, en general, de asignación para procesos químicos y para una gran variedad de problemas. Una de las características de los AG's es que son robustos, eso precisamente los hace ser aplicables a una gran diversidad de áreas del conocimiento.

Una definición de los AG's podría ser la siguiente:



Los algoritmos genéticos son métodos de búsqueda ciega de soluciones óptimas. Mantienen una población conformada por un conjunto de posibles soluciones que es transformada por operadores que simulan el proceso de evolución, con el propósito de generar mejores soluciones.

1.2 Historia.

Los algoritmos genéticos nacieron en la década de los 60's en la Universidad de Michigan. Su creador fue el investigador John Holland (como se señala en muchos textos). El interés de Holland era lograr que las computadoras aprendieran por sí mismas²; la técnica que inventó la incorporó en un programa de computadora y la llamó en un principio “planes reproductivos” pero, tras publicar su libro³, se popularizó con el nombre “algoritmo genético”.

² Se puede decir que lo logró, pues de sus técnicas nació la Programación Genética, que mucho tiene que ver con el autoaprendizaje de las computadoras.

³ Adaptation in Natural and Artificial Systems (1975).

A partir de entonces se ha notado un creciente interés, por parte de investigadores y académicos, en esta nueva *técnica de búsqueda basada en la teoría de la evolución*.

A pesar de la sencillez del algoritmo, para diversos problemas, muchos investigadores han aportado variantes mejoras o adaptaciones. No es fácil explicar el porqué de ese afán de seguir aportando, pero creemos firmemente que los algoritmos genéticos darán mucho de qué hablar en los próximos años.

Para una mejor comprensión de la analogía que hacen con la teoría de la evolución, es necesario hablar un poco de genética y de los conceptos que maneja dicha teoría.

1.3 Genética.

La genética es la rama de la biología que se ocupa de la herencia y la variación de las especies. Las unidades hereditarias que se transmiten de una generación a la siguiente (herencia) son los denominados *genes*. Los genes se encuentran en una larga molécula llamada ácido desoxirribonucleico (DNA). El DNA se organiza en diferentes unidades denominadas *cromosomas* que se encuentran en el núcleo de las células, a su vez los *genes* están contenidos en ellos. Por lo general los cromosomas son estables, en raras ocasiones se presenta un cambio espontáneo. Este cambio, llamado *mutación*, altera las instrucciones codificadas. El resultado neto de una *mutación* a menudo se manifiesta como cambio en la apariencia física de un ser vivo o

Un gen puede cambiar a dos o más formas alternativas, llamadas *alelos*. Cada gen ocupa una posición específica en un cromosoma, denominado el *locus* genético (loci, en plural). Por tanto, todas las formas alélicas de un gen se encuentran en una posición correspondiente en los *cromosomas*.

A cualquier característica detectable o distintiva de un organismo se le da el nombre de *fenotipo*. La característica puede ser visible, como el color de los ojos o la sedosidad del pelo, o puede requerir pruebas especiales para su identificación, tal es el caso del grupo sanguíneo. El *fenotipo* es pues, el resultado de la manifestación de genes que se expresan en un medio ambiente específico.

Todos los *genes* que posee un ser vivo constituyen su *genotipo*. Para ejemplificar y diferenciar al *genotipo* del *fenotipo*, se tratará un poco sobre dominancia. Los *alelos* se clasifican en *dominantes* o *recesivos* de acuerdo a ciertas características. Para el ejemplo, usaremos letras mayúsculas y minúsculas para designar los *alelos* dominantes y recesivos respectivamente.

Ejemplo: La falta de pigmento en el organismo humano es un carácter recesivo anormal denominado albinismo. Empleando *A* y *a* para representar el *alelo* dominante (normal) y el *alelo* recesivo (albino), respectivamente, es posible tener los siguientes tres *genotipos* y dos *fenotipos*:

| Genotipos | Fenotipos |
|----------------|-----------------------|
| AA (dominante) | Normal (pigmento) |
| Aa | Normal (pigmento) |
| aa (recesivo) | Albino (sin pigmento) |

Tabla 1.1 Genotipos y Fenotipos.

1.3.1 La genética del sexo.

Estamos acostumbrados a considerar el sexo en términos de hombres y mujeres o machos y hembras; sin embargo las plantas también tienen sexo, al menos sabemos que hay partes masculinas y femeninas en una flor. En la mayor parte de los organismos, el número de sexos ha quedado reducido a dos, para el caso carece de importancia que tengan más de dos sexos, o el que éstos residan en un mismo individuo o en individuos diferentes. La importancia de los sexos en sí es que constituyen un mecanismo que fija la numerosa variabilidad genética que caracteriza a la mayor parte de las poblaciones naturales. El proceso evolutivo de selección natural depende de esta variabilidad genética para aportar "la materia prima" a partir de la cual los ejemplares mejor adaptados por lo general sobreviven pasando por la cruda de estos, hasta la *reproducción*.

Para dejar más claro la relación de los genotipos y fenotipos con los procesos de reproducción, cruda y herencia se ilustrará mediante la figura 1.1.

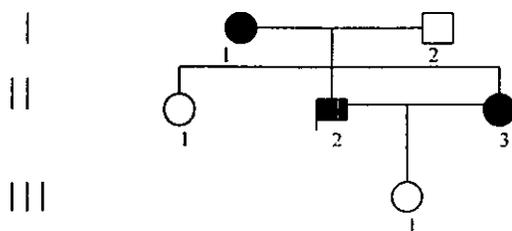


Figura 1.1 Reproducción, cruda y herencia.

Se consideran 4 tipos de individuos descritos en la tabla 1.2.

| | | | |
|-------------|--------------|--------------|---------------|
| ● | ○ | ■ | □ |
| Macho negro | Macho blanco | Hembra negra | Hembra blanca |

Tabla 1.2 Descripción de individuos

En la ilustración se contemplan tres generaciones. En la primera de ellas, se cruzará un macho negro con una hembra blanca. El resultado de esta cruce se ve en la segunda generación conformada por tres individuos, de los cuales dos de ellos se cruzarán para dar origen a la tercera generación. La existencia de individuo blanco descendientes de padres negros se explica a través de la herencia entre generaciones. Todos los detalles de genotipos y fenotipos se presentan en la tabla 1.3.

| Individuos | Fenotipo | Genotipo |
|------------|---------------|----------|
| I1 | Macho negro | Bb |
| I2 | Hembra blanca | bb |
| II1 | Macho blanco | bb |
| II2 | Hembra negra | Bb |
| II3 | Macho negro | Bb |
| III1 | Macho blanco | bb |

Tabla 1.3 Detalle de fenotipos y genotipos.

Lo interesante de esto es ver como se expresan los fenotipos con respecto a los genotipos y la herencia que existe entre las generaciones.

1.3.2 Teoría de la evolución.

La evolución es el proceso por el cual los organismos cambian gradualmente de una forma a otra, en el transcurso de largos periodos llamados eras geológicas.

1.3.2.1 Selección.

Charles Darwin estableció los principios modernos de la teoría de la evolución en su libro, *On the origin of species by means of natural selection or preservation of favored races in the struggle for Life*, publicado en 1859. Darwin observó que los organismos poseen un potencial de reproducción que de hecho excede en mucho al número de individuos en reproducción observado en una población. Más aún el tamaño de la mayor parte de las poblaciones tiende a ser relativamente estable de una generación a otra. En una especie en particular, una hembra podría depositar millones de huevecillos, pero esto no implica un incremento en el tamaño de la población natural en las generaciones sucesivas. La competencia por satisfacer las necesidades más elementales como son: el alimento, el agua, el refugio y el espacio, limita enormemente el número de organismos que pueden ocupar el mismo hábitat o nicho ecológico (es decir, que persiguen la misma forma de vida). Darwin pensó que la diferencia entre el potencial de reproducción y el tamaño de la población real representa la fuerza de la selección natural. Sólo los individuos que estén mejor adaptados o más aptos sobrevivirán para reproducirse.

Con el conocimiento de la genética moderna, la teoría de la evolución ha ganado el status de una ciencia y en la actualidad pueden definirse muchos de sus principios en términos matemáticos.

1.4 Introducción a los algoritmos Evolutivos.

En general, se llama algoritmo evolutivo (en lo sucesivo, AE) a cualquier procedimiento estocástico de búsqueda basado en el principio de evolución. La finalidad es la supervivencia del más apto, el modo de conseguirlo es por adaptación al entorno. Dicho de otro modo, los más aptos tienen más posibilidades de sobrevivir y, como resultado, más oportunidades de transmitir sus características a las generaciones siguientes.

Más concretamente, al ejecutar un AE, una población de individuos que representan a un conjunto de candidatos a soluciones de un problema, es sometida a una serie de transformaciones con las que se actualiza la búsqueda y después a un proceso de selección que favorece a los mejores individuos. Cada ciclo de transformación+selección constituye una generación. Se espera del AE que, tras cierto número de generaciones, el mejor individuo esté razonablemente próximo a la solución buscada.

Dentro de los algoritmos evolutivos destacan de sobremanera los algoritmos genéticos, de entre las razones de tal importancia, están:

- Los AG's son el paradigma más completo, i.e., reúnen todas las ideas fundamentales de dicho enfoque.
- Son muy flexibles, pueden adaptárseles nuevas ideas, de ahí el gran interés de investigadores de proponer o sugerir cambios en su estructura, además es muy fácil hibridar con otros métodos.
- Su base teórica, aunque no sea del todo aceptada, es sencilla, pero posee grandes posibilidades de ampliación.

- Los AG's, para su funcionamiento, no necesitan de conocimiento específico del problema, eso los hace muy versátiles.
- Son fácilmente implantables en computadoras de diversas características.
- Resultan bastante aceptables, sobre todo cuando se trata de problemas que difícilmente se resolverían por otros métodos.
- Los AG's son el paradigma más usado de entre todos los algoritmos evolutivos y, junto con las redes neuronales, los más usados de toda la computación neuronal.

Todo esto nos lleva a plantear que los algoritmos evolutivos son por defecto algoritmos genéticos.

Todas estas afirmaciones se justificarán en los apartados siguientes.

1.5 Principios Básicos.

El primer paso para la implementación de un algoritmo genético es generar una población inicial. En términos genéticos, el conjunto de parámetros representados por un cromosoma en particular es el *genotipo* (véase apartado 1.3). El *genotipo* contiene la información requerida para construir un organismo, el cual es conocido como *fenotipo*. Los mismos términos son usados en los AG's. Por ejemplo en el diseño de un puente (las dimensiones de las vigas), el conjunto de parámetros que especifican un diseño en particular es el *genotipo*, mientras que la construcción finalizada es el *fenotipo*. Estos parámetros (conocidos como genes) se unen para formar una cadena (cromosoma)

de valores. La aptitud de un individuo depende del comportamiento del *fenotipo*, esto puede ser inferido del *genotipo*, es decir, puede ser procesado de los cromosomas, usando la *función de aptitud*.

Para el diseño de un AG, es necesaria una *función de aptitud*, que no es más que la función objetivo de nuestro problema a optimizar. La principal característica que debe tener es de penalizar a las malas soluciones y de premiar a las buenas, de forma que estas últimas sean las que se propaguen con mayor rapidez. La *función de aptitud* retorna un simple valor numérico que viene a ser la "aptitud", a través de la cual se supone una utilidad o habilidad proporcional individual que es representada por los cromosomas. Es necesario diferenciar entre función de evaluación y *función de aptitud*. La función de evaluación provee una medida de comportamiento de cada individuo con respecto a la población entera, en cambio la *función de aptitud* transforma esa medida de comportamiento, de manera que se haga una adecuada distribución de oportunidades de reproducción. Sin embargo, en los AG's canónicos, i.e., los más simples, la función de evaluación es igual a la de aptitud. La evaluación involucra a cada individuo por separado, de manera que es independiente de las evaluaciones de la población en general, y la aptitud necesariamente contempla a toda la población.

El siguiente paso es la codificación, la más común es a través de cadenas binarias, éstas serán construidas sobre un alfabeto binario $V=\{0,1\}$. Como una notación convencional, nos referimos a cadenas como letras mayúsculas y a los caracteres individuales con letras minúsculas subscriptas por su posición. Por ejemplo: la cadena de 8 bits

$$A=01110001$$

puede ser representada simbólicamente como sigue:

$$A=a_1a_2a_3a_4a_5a_6a_7a_8.$$

Aquí cada a_i representa una simple característica (de acuerdo a la analogía natural se llaman genes), donde cada característica puede tomar un valor binario, es decir un 1 ó un 0 (le llamamos *alelos* a los valores de los a_i). En la cadena particular 01110001, a_1 es igual a 0, a_2 es igual a 1, a_3 es igual a 1, etc. Es posible también tener cadenas donde los genes no están ordenados secuencialmente como en la cadena A. Por ejemplo: una cadena A' podría tener el siguiente orden:

$$A' = a_2 a_6 a_4 a_3 a_7 a_1 a_5$$

Una búsqueda genética requiere una población de cadenas y consideramos una población de cadenas individuales A_j ; $j=1,2,\dots,n$, contenidas en la población $P(t)$, donde t es el tiempo o generación t , y la letra en negrita es usada para denotar una población. Las variables, es decir, las cadenas son discretizadas. El rango de esta discretización corresponde a alguna potencia de 2. Por ejemplo, con 10 bits, podemos representar 1024 valores discretos ($2^{10}=1024$).

La mayoría de los problemas a resolver tienen un conjunto de valores exacto, dentro de los cuales se encuentran las soluciones. Por ejemplo, supóngase que se tiene un espacio de búsqueda en donde hay exactamente 1200 valores discretos, los cuales pueden ser representados con la cadena (variable) A_j . Necesitaremos 11 bits para cubrir este rango, en este caso existen 2048 valores discretos ($2^{11}=2048$), i.e., existen 848 valores innecesarios, que dejan de causar ruido con una adecuada *función de aptitud*, como se verá más adelante, pues se penalizará a estos valores. En general para una cadena de longitud L , el tamaño del espacio de búsqueda es 2^L .

Normalmente los espacios de búsqueda son grandes, es común utilizar cadenas de 30 bits o más. Difícilmente se pueden encontrar problemas con una codificación de hasta 400 bits, pero los hay, aunque se trate de espacios de búsqueda exageradamente grandes.

También se han utilizados otros alfabetos para la construcción de cadenas (cromosomas), tales como números reales, o conjuntos de letras. El más convencional, el de cadenas binarias es el más popular por su sencillez de implementar y por haber sido el propuesto originalmente por Holland.

La operación de un AG simple se lleva a cabo generalmente a través del siguiente pseudo-código:

```
Generar aleatoriamente la población inicial, A(0);  
Evaluar A(0);  
t=0;  
repetir  
  t=t+1;  
  generar A(t) usando A(t-1)  
    Crear una nueva población efectuando las operaciones de cruce,  
    reproducción proporcional a la adaptación y mutaciones a los  
    individuos cuyas aptitudes acaban de ser medidas.  
  evaluar A(t);  
hasta encontrar una solución.
```

De acuerdo al pseudo-código, primeramente se generará aleatoriamente la población inicial, que consiste en cadenas de caracteres⁴. Cada vez que se repita, es decir, en cada iteración se creará una nueva generación. Para saber qué tan buena es la solución se aplica la *función de aptitud* a cada uno de los miembros. Una vez que conocemos la aptitud, procedemos a la selección de los que se cruzarán. Se supone que deben ser los mejores, para así ir creando descendientes con mejores características.

Como se puede apreciar en el pseudo-código, el algoritmo es muy fácil de programar e implementar en casi cualquier equipo de cómputo.

⁴ En programación se utilizan tipos booleanos o de carácter de 1 bit sin signo.

1.6 Características Intrínsecas de los AG's.

Los AG's tienen ciertas características que los hacen únicos, son métodos de búsqueda:

- **Ciega**, pues no necesitan conocimiento alguno específico del problema, de tal forma que la búsqueda se basa en los valores de la función objetivo.
- **Codificada**, puesto que no trabajan directamente con el dominio del problema sino con representaciones de sus elementos.
- **Múltiple**, esto significa que buscan entre varios candidatos simultáneamente.
- **Estocástica**, es lo que permite tener un control en la fase de transformación y, evidentemente, en la inicialización de la población.

Las características mencionadas son precisamente las que permiten que los AG's sean robustos, es decir, tienen que ser generales sin perder eficiencia, y eficientes sin perder generalidad.

Los AG's explotan las similitudes entre variables representadas por cadenas, que en conjunto forman una población, esto les da una clara ventaja con respecto a otros métodos de búsqueda, i.e., no trabajan con un solo punto, permitiéndoles esto, mantener una población de puntos bien adaptados y, por consiguiente, se reduce la probabilidad de alcanzar falsos óptimos.

Sin entrar de momento en pormenores, se debe señalar que al procesar las similitudes de las cadenas, se están procesando también los patrones de similitud que contienen, y que son muchos más. Es precisamente esta propiedad la más impórtante y, como veremos en próximos apartados, la que los hace más eficaces que otros métodos de búsqueda.

1.7 Estructura y Componentes Básicos.

1.7.1 Analogía con la naturaleza.

En la naturaleza, los individuos de una población compiten entre sí por los recursos que les permitirán sobrevivir, tal es el caso de la comida, evidentemente, el agua y algún refugio, pero también los miembros de algunas especies frecuentemente compiten para atraer a una pareja, y los que logran conseguirla es porque resultaron más aptos y, por consiguiente tendrán descendientes, mientras que los que resultaron menos aptos tendrán menos descendientes o incluso no tendrán ninguno.

Esto significa que los genes de un individuo altamente adaptado generarán un número mayor de individuos (hijos) en la siguiente generación. La combinación de buenas características de diferentes ancestros puede algunas veces producir hijos "super adaptados" a las características de su medio ambiente.

Los algoritmos genéticos utilizan una analogía directa con los procesos de la evolución en la naturaleza. Existe una población de individuos, cada uno representa una posible solución al problema en cuestión, a cada uno se le asignará una puntuación para calificar su *aptitud*, que dependerá de qué tan buena fue su respuesta al problema. A los que mejor se acercaron a la respuesta, es decir los más aptos, se les concede la oportunidad de *reproducirse*, mediante *cruzamientos* con otros individuos de la *población*, produciendo de esta manera descendientes que tendrán características de ambos padres. Los individuos menos aptos tienen poca probabilidad de ser seleccionados para la reproducción, y desaparecen.

Con la nueva generación, surgen nuevas posibles soluciones, que se espera sean mejores que las anteriores. Cada generación contiene individuos más aptos, dado que heredaron las mejores características de sus antecesores, esto permite recorrer las áreas más prometedoras del espacio de búsqueda. Si el diseño del AG fue adecuado, la población convergerá a una solución óptima o casi óptima al problema.

El papel que juegan los procesos de la *cruza*, la *selección* y *reproducción* son fundamentales en la evolución, sin embargo, al igual que en la naturaleza, existe un factor que no se debe olvidar, y que en ocasiones es clave para lograr la solución, aunque hay quienes debaten su importancia. La *mutación* permite un cambio aleatorio, y permite que todos los puntos del espacio de búsqueda tengan probabilidad de ser examinados, aunque no debe ser utilizada mucho, porque el AG sería una búsqueda al azar.

Para un mejor entendimiento, a los individuos o miembros de una población les llamaremos indistintamente cromosomas o cadenas.

1.7.2 Estructura básica.

Recordemos que el AG conlleva una serie de transformaciones que actualizan la búsqueda y, después, un proceso de selección que debe favorecer a los mejores individuos. De manera muy simplista se puede sintetizar con el siguiente diagrama:

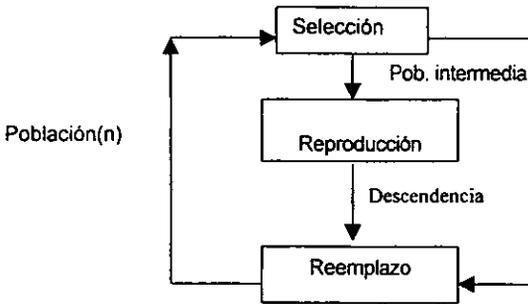


Figura 1.2. Ciclo elemental de un AG.

En el diagrama se muestra una población que consta de n miembros, los cuales pasan por un proceso de selección para crear una población intermedia. De esta población intermedia se extrae un grupo reducido de individuos llamados padres o progenitores, que son los que se cruzarán para pasar a la etapa de reproducción. A través de operadores genéticos, los padres se someten a ciertas transformaciones de alteración y recombinación, en virtud de las cuales se generan nuevos individuos que constituyen la descendencia, y que reemplazarán a sus padres, para formar la nueva población $P[t+1]$.

Para llevar a cabo este procedimiento de búsqueda, el AG necesita de una función de evaluación $f(x)$, y una *función de aptitud*, que en ocasiones resultan ser las mismas.

1.7.3 Métodos y criterios necesarios para implantar un AG.

Se deben seguir los siguientes criterios y métodos al implantar un AG:

- **Criterio de codificación.** Este procedimiento es el que hará corresponder cada punto del dominio con cadenas (cromosomas). En términos biológicos, es el paso del *genotipo* al *fenotipo*.

- **Criterio de penalización.** Frecuentemente será posible establecer una correspondencia punto por punto entre el dominio del problema y el conjunto de las cadenas de tamaño L ; esto da como resultado que existan soluciones no válidas, para ello es necesario crear una penalización, a fin de que no causen ruido.
- **Criterio de iniciación.** Se refiere a cómo se debe generar la población inicial.
- **Criterio de detención.** Se deben establecer las condiciones necesarias para determinar si el AG ha convergido a una solución aceptable o, en su defecto, ha fracasado en la búsqueda y no tiene sentido continuar.
- **Funciones de evaluación y aptitud.** Se debe diseñar una función de evaluación apropiada para el problema, así como la *función de aptitud* que utilizará el AG para resolverlo.
- **Operadores genéticos.** Son los operadores que se encargan de la transformación; los básicos son la cruce y la mutación con sus múltiples variaciones.
- **Criterios de selección.** El proceso debe premiar a los más aptos y castigar a los peores individuos, es muy variable la forma de llevar a cabo esto.
- **Criterios de reemplazo.** Se basan en la selección de los supervivientes básicamente.
- **Parámetros.** Todo AG necesita de ciertos parámetros para su funcionamiento, tales como el tamaño de la población, las probabilidades de cruce y mutación, las tolerancias de la convergencia, etc.

1.8 Procedimientos Básicos de un AG.

Existen dos procedimientos básicos para la implantación de un AG, la selección (de parejas para la cruce), y la reproducción (o transformación).

1.8.1 Mecanismos de selección de individuos.

La selección consiste en escoger de la población inicial, los n elementos de la población intermedia de parejas para realizar la cruce. El criterio concreto de selección depende del problema y del buen juicio del diseñador. Los más usados en la práctica son los métodos aleatorios de torneo y el de la ruleta. Los métodos deterministas se usan muy poco, entre otros motivos porque van contra la filosofía del algoritmo.

Para llevar a cabo la selección, se sigue alguno de los siguientes mecanismos:

- **Determinista.** Se toma un conjunto de individuos de la población siguiendo un criterio fijo, que puede ser “los k mejores”, “los k peores”, se ordenan de mejor a peor, “uno sí otro no”, “el primero con el último, el segundo con el penúltimo”, etc.
- **Estocástico.** Se asignan probabilidades de selección o puntos a los individuos de la población basándose en su aptitud. De entre ellos destacan los siguientes:
 - **Por sorteo.** Se asignan puntuaciones y condiciones, de manera que los más aptos tengan mayores probabilidades de ganar.
 - **Por torneo.** Se agrupan los elementos de una población en subconjuntos tomados al azar y ganan el mejor o los mejores de cada subconjunto.

Torneo. El método del torneo consiste en hacer competir a los cromosomas repartidos en grupos de tamaño predefinido (normalmente es de 2, es decir, compiten en parejas), la idea es muy simple: el más apto gana la competencia, y los competidores se seleccionan al azar. Este método garantiza que el más apto de la población sobrevivirá, pero los que sigan de mayor a menor aptitud, aunque ésta sea alta, pueden perder con el más alto.

- **Por ruleta** En base a la aptitud de cada individuo se les asigna un rango en la ruleta, es decir, entre más aptos sean mayor será su área en la ruleta, y mayor será la probabilidad de ser seleccionado.

La Ruleta⁵. Es muy efectivo y muy simple, consiste en crear una ruleta, en donde cada cromosoma tiene una fracción proporcional a su aptitud. Como ejemplo, la figura 1.3 muestra la proporción asignada de acuerdo la tabla 1.4.

| | | |
|--------|-----|-------|
| 010101 | 21 | 13.2 |
| 111000 | 56 | 35.7 |
| 001101 | 13 | 8.3 |
| 011011 | 27 | 17.2 |
| 000111 | 7 | 4.6 |
| 100001 | 33 | 21.0 |
| -- | 157 | 100.0 |

Tabla 1.4 Valores para mostrar el ejemplo del uso del método de la ruleta.

⁵ Método propuesto por Goldberg en su libro "Genetic algorithms in Search, Optimization and Machine Learning" (1989)

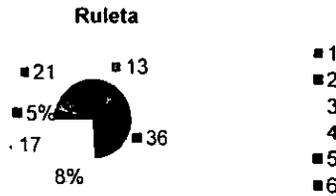


Figura 1.3 Ruleta con fracciones proporcionales a la aptitud.

La ruleta se elabora basándose en los porcentajes de la última columna. Dado que entre más apto sea un cromosoma, mayor será su parte proporcional, entonces se espera que sean seleccionados más veces los que ocupan una mayor área en la ruleta.

1.8.2 Mecanismos de Reproducción.

Reproducción: Aplicando los operadores de transformación (cruza y mutación) sobre ciertos miembros de la población intermedia, es decir, de progenitores se obtiene una descendencia de nuevos miembros.

Los operadores de cruce realizan una recombinación, actúan sobre parejas de individuos y normalmente originan otro par de individuos que combinan características de los progenitores. Dado que normalmente en los AG's los individuos están representados a través de cadenas binarias, la cruce se lleva a cabo por intercambio de segmentos.

Cualquiera que sea el método de selección, una vez concluido éste, se procede a la cruce para que pueda existir reproducción sexual entre los individuos que fueron seleccionados. La idea básica de esta operación es intercambiar material genético para que los descendientes tengan

características de ambos padres. Las dos formas más comunes de reproducción son: utilizar un solo punto de cruce o dos puntos de cruce.

Un solo punto de cruce. Para el intercambio genético, es necesario definir qué parte de los cromosomas de los padres van a heredar sus descendientes. En este sentido el punto de cruce, es precisamente el que va dar esa información. Para ilustrar de una mejor manera veamos la figura 1.4.

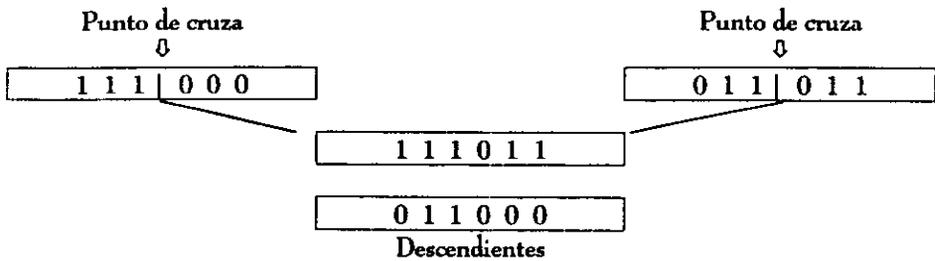


Figura 1.4 Un solo punto de cruce, en este ejemplo el punto fue 3, así que entre el tercer y el cuarto alelo se realiza la cruce.

Dos puntos de cruce. La idea es la misma, sólo que se usarán 2 sitios de cruce y es como en la figura 1.5.

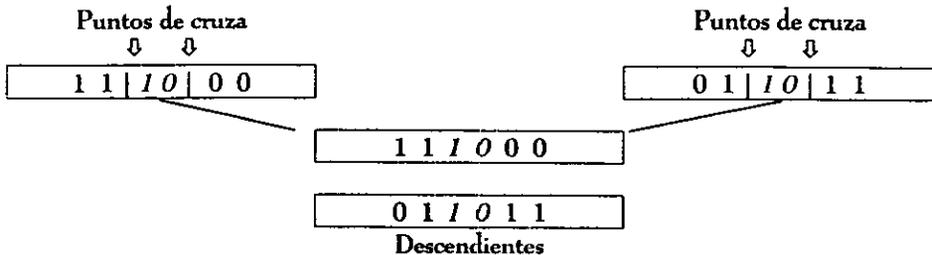


Figura 1.5 Dos puntos de cruce, se mantienen los alelos de los extremos y se intercambian los del centro.

En ambos casos, si el sitio de cruza, que debe ser seleccionado aleatoriamente cae en un extremo, entonces no habrá cruza y los descendientes serán iguales a sus padres.

La cruza normalmente se maneja de acuerdo a un porcentaje, que suele ser alto (60 a 100), esto implica que en ocasiones puede no llevarse a cabo, y habrá algunos cromosomas que pasen intactos a la siguiente generación. Se ha desarrollado una técnica, desde hace varios años en la que el individuo más apto durante varias generaciones no se cruza con nadie, y se mantiene hasta que surge un individuo mejor, que lo desplazará⁶.

El siguiente operador es la mutación, y su función es hacer un cambio a uno de los genes, de un cromosoma, elegido aleatoriamente. La mutación se aplica a cada individuo, con una probabilidad muy baja (0.1 a 0.5), después de la cruza. El bit elegido (gene) cambiará su valor por su complemento (un cero cambia a uno y viceversa). Esta operación permite introducir nuevo material genético, tal como sucede con equivalentes biológicos. Véase la figura 1.6, que ilustra la operación de mutación.

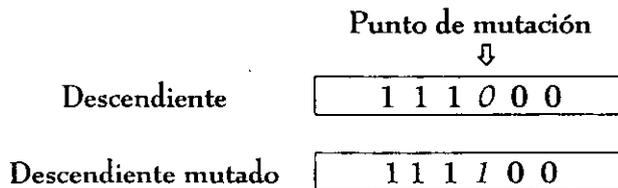


Figura 1.6 Ilustración de la mutación.

⁶ Dicha técnica se llama *elitismo*, y no debe sorprender que haya surgido en Alemania.

1.9 Convergencia.

Si el algoritmo genético ha sido implantado correctamente, la población evolucionará a través de diversas generaciones hacia la mejor aptitud de sus integrantes, y por consiguiente hacia el óptimo global. Se dice que un gene converge cuando el 95% de la población comparte el mismo valor. La población convergerá cuando todos los genes hayan convergido.

Como no sabemos la respuesta que queremos encontrar, no sabemos cuándo detener el algoritmo genético; entonces, se usan dos criterios para la detención: se puede establecer un número máximo de generaciones, o bien detenerlo cuando se haya estabilizado la población, i.e., cuando haya convergido.



Portada del libro "Los Dragones del Edén" de Carl Sagan

Capítulo

2

Implantación y ejemplos del AG básico.

Objetivos Particulares:

- Describir la estructura del AG básico.
- Definir los criterios para implantar un AG.
- Resolver un problema de optimización paramétrica multimodal.
- Ilustrar la forma de implementar un AG para la resolución de un problema de asignación.

Hay miles de millones de galaxias en los cielos, probablemente la mitad de ellas tengan planetas a distancias adecuadas biológicamente del sol local... debe haber otros habitantes en el universo. Por supuesto, los detalles de la evolución no serían los mismos. Incluso si la tierra se iniciase otra vez y solamente las fuerzas operasen al azar no se produciría nada parecido a un ser humano, porque los seres humanos son el producto final de una senda evolucionista exquisitamente complicada, llena de falsos comienzos, de callejones casi sin salida y de accidentes estadísticos... debe haber, creo, muchos lugares en la Galaxia donde haya seres mucho más avanzados que nosotros en ciencia y tecnología, en política, ética, poesía y música

Carl Sagan, 1973

En esta sección se estudia una implantación elemental pero completa del algoritmo descrito anteriormente y se resuelven dos ejemplos prácticos, el primero de optimización paramétrica multimodal, y el segundo un problema de asignación. Se trata, por una parte, de aclarar y concretar todo lo discutido en el capítulo anterior, y por otra parte facilitar la comprensión del estudio teórico del siguiente capítulo.

Dicha implantación fue propuesta por Goldberg en su texto clásico (Goldberg 1989) y es la que, con cambios menores, se usa habitualmente para ilustrar el funcionamiento de los AG's.

2.1 Criterios en la implantación.

El AGB o SGA (simple genetic algorithm) es un algoritmo genético que incorpora los siguientes métodos y criterios:

1. Criterio de codificación: Este criterio es específico de cada problema. Debe hacerse corresponder a cada punto del dominio del problema un elemento del espacio de búsqueda, esto es normalmente a través de cadenas binarias simples.
2. Criterio de tratamiento de los individuos no factibles: Se supone que la codificación se hace de tal manera que todas las cadenas posibles representan a individuos factibles.
3. Criterio de iniciación: La población inicial consta de cadenas binarias elegidas al azar.
4. Funciones de evaluación y aptitud: La función de aptitud coincide con la de evaluación. Preferiblemente la función de evaluación viene dada a través de una función objetivo.
5. Operadores genéticos: Utilización de un solo punto de cruce y mutación bit a bit sobre los individuos codificados.
6. Criterio de selección: Por torneos o por ruleta.
7. Criterio de reemplazo: Inmediato.
8. Criterio de parada: Fijando el número máximo de iteraciones.
9. Parámetros de funcionamiento: Deterministas. Una elección típica es:

| |
|--------------------|
| $Tampob = 30$ |
| $MaxIter = 50$ |
| $P_{cruza} = 60\%$ |
| $P_{mut} = 3.3\%$ |

Donde, $Tampob$ es la variable que representa al tamaño de la población, $MaxIter$ es el número máximo de iteraciones, P_{cruza} es la probabilidad de cruza y P_{mut} es la probabilidad de mutación.

De todas estas especificaciones las únicas que precisan explicación, por ahora, son las referidas a los operadores genéticos. La determinación exacta de qué individuos se van a cruzar y qué bits se van mutar se lleva a cabo mediante ensayos de Bernoulli.

Para la cruza el proceso es así:

Para cada individuo se genera un número aleatorio $r_i \text{ Rand}(0,1)$.

Se comparan los r_i con la probabilidad de cruza, P_{cruza} , que es un parámetro del método.

Son seleccionados para cruzarse todos los individuos v_i para los que:

$$r_i < P_{cruza}$$

Para la mutación se realiza el mismo proceso con cada uno de los $n \cdot L$ bits de la población y usando la probabilidad de mutación P_{mut} .

El emparejamiento de los individuos a cruzar se hace al azar; habitualmente las etapas anteriores a la cruza han introducido el suficiente desorden en la población como para que un emparejamiento sucesivo se pueda considerar aleatorio.

Todavía queda un pequeño problema que resolver para implantar completamente la cruce: ¿Qué hacer cuando el número de individuos a cruzar sea impar? En tal caso, el último individuo quedará sin pareja y lo que se hace es o bien pasarlo directamente a la siguiente generación o bien aparearlo con cualquier otro elegido al azar.

2.2 Ejemplo Práctico de Implantación de un AG.

Se va a resolver el siguiente problema de optimización paramétrica propuesto por R.L. Riolo en (Riolo, 1992).

Hallar

$$x = \arg \max \{f(x) = x + |\sin(32 \pi x)| : x \in [0, 1]\}$$

utilizando los siguientes parámetros:

| |
|--------------------|
| $T_{ampob} = 6$ |
| $P_{cruza} = 30\%$ |
| $P_{mut} = 1\%$ |

con una precisión de media centésima (i.e., $TOL = 5 \cdot 10^{-3}$)

El problema tiene la dificultad de ser fuertemente multimodal, i.e., en el intervalo $x \in [0, 1]$ la función objetivo posee 32 óptimos locales (ver figura 2.1). Por tal motivo, los métodos analíticos de optimización (por ejemplo el de Newton--Raphson) manifiestan una tendencia a caer en alguno de esos 32 subóptimos, y sólo convergen al óptimo global ($x = 0.98447395$, $f(x) = 1.98442447$) si se les da una aproximación inicial muy buena. Por esto se debe recurrir a otro tipo de algoritmos de búsqueda.

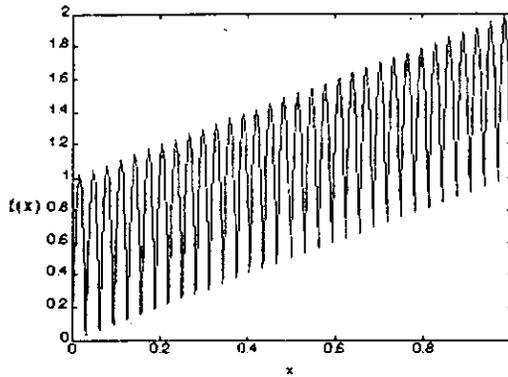


Figura 2.1: Función objetivo.

Lo primero será la codificación; representaremos el dominio del problema a través de números binarios. Lo más sencillo parece ser discretizar el intervalo $[x_{\min}, x_{\max}]$ en una cantidad de puntos 2^{Len} tal que la distancia entre puntos consecutivos sea menor que la tolerancia especificada, i.e.,

$$\frac{x_{\max} - x_{\min}}{2^{Len} - 1} \leq TOL$$

así cada punto del espacio de búsqueda queda representado mediante un binario de longitud Len comenzando por $0...0$ que representa a x_{\min} y terminando en $1...1$ que representa a x_{\max} .

Enseguida, el punto clave es calcular la longitud de los individuos. Si se dispone directamente de la tolerancia TOL el cálculo es inmediato a partir de la fórmula anterior, sólo despejamos Len :

$$Len \geq \left\lceil \log_2 \left(1 + \frac{x_{\max} - x_{\min}}{TOL} \right) \right\rceil$$

Recíprocamente, se puede calcular el punto x que corresponde a un individuo v mediante la siguiente fórmula de decodificación:

$$x(v) = x_{\min} + \text{binadec}(v) \cdot \frac{x_{\max} - x_{\min}}{2^{Len} - 1}$$

para el presente caso se tiene:

$$Len \geq \left\lceil \log_2 \left(1 + \frac{1 - 0}{.005} \right) \right\rceil = 7.651$$

Redondearemos la variable Len al entero inmediato superior, es decir, utilizaremos cadenas binarias de 8 bits de longitud. De hecho dado que con ocho bits de longitud se pueden representar $2^8 = 256$ puntos del intervalo de búsqueda, la precisión será exactamente $1/256 = 0.00390625 (< TOL)$.

El siguiente paso es el de iniciación. En la presente tabla se representa la población inicial de cadenas, elegidas al azar, junto con los valores significativos asociados a cada una:

| i | v_i | x_i | Eval(v_i) | Aptitud(v_i) | Part(v_i) | Partacum(v_i) |
|---|----------|------------|---------------|------------------|---------------|-------------------|
| 1 | 10111010 | 0.72656250 | 1.68308 | 1.68308 | 0.25820 | 0.25820 |
| 2 | 10010111 | 0.58984375 | 1.44967 | 1.44967 | 0.22239 | 0.48059 |
| 3 | 00110101 | 0.20703125 | 0.56235 | 0.56235 | 0.08627 | 0.56686 |
| 4 | 11001101 | 0.80078125 | 1.78708 | 1.78708 | 0.27415 | 0.84101 |
| 5 | 01000100 | 0.26562500 | 0.71500 | 0.71500 | 0.10969 | 0.95070 |
| 6 | 00011110 | 0.11718750 | 0.32126 | 0.32126 | 0.04930 | 1.00000 |

Tabla 2.1 de valores de $P[1]$.

Obsérvese que al principio el individuo más apto es $v_4 = 11001101$ con una aptitud de 1.78708 y el menos apto $v_6 = 00011110$ con una aptitud igual a 0.32126.

Nótese que la función de aptitud, $Aptitud(v_i)$, coincide con la de evaluación, $Eval(v_i)$, y ésta con el objetivo $f(x)$. $Part(v_i)$ representa la aptitud de cada individuo entre la aptitud total de la población.

Los siguientes pasos son:

1. Construir la tabla de valores asociados a la población actual: (Es la que se acaba de calcular). La aptitud global de la población se obtiene sumando las de todos los individuos y vale $Aptitud(P[0]) = 6.51854$.
2. Realizar el sorteo de individuos para la reproducción: En este caso se utiliza el método de la ruleta. Se generarán seis números aleatorios $Rand(0,1)$ y se comparan con las participaciones acumuladas $Partacum(v_i)$. Los números aleatorios obtenidos y los correspondientes individuos seleccionados son:

- 0.52920 $\rightarrow v_3$
- 0.04358 $\rightarrow v_1$
- 0.29492 $\rightarrow v_2$
- 0.04109 $\rightarrow v_1$
- 0.28483 $\rightarrow v_2$
- 0.78687 $\rightarrow v_4$

quedando la población intermedia como:

| i | v'_i |
|---|----------|
| 1 | 00110101 |
| 2 | 10111010 |
| 3 | 10010111 |
| 4 | 10111010 |
| 5 | 10010111 |
| 6 | 11001101 |

Tabla 2.2 Población Intermedia.

3. De entre los individuos anteriores se eligen primero los que se van a cruzar. Como $P_{\text{cruza}} = 30\%$, en promedio se cruzarán $6 * .3 = 1.8$ individuos. Para determinar la cantidad correspondiente en cada iteración se procede así:

Se generan seis números aleatorios $r_i = \text{Rand}[0,1)$, uno para cada individuo, y se eligen para la cruce aquellos para los que $r_i < P_{\text{cruza}}$. Por ejemplo, si los números son:

0.72232 0.31575 0.56156 0.15608 0.86910 0.03070

entonces se seleccionan los individuos v'_4 y v'_6 .

La cruce se llevará a cabo con esos dos individuos. Si hubiera salido un número impar se habría tenido que elegir otro o desechar uno de ellos aleatoriamente.

4. Se cruzan todos los individuos elegidos en un punto al azar ($\text{IntRand}[1, \text{Len} - 1)$). Como hay dos, es trivial hacer la pareja, si hubiera cuatro, se emparejarían arbitrariamente, cada uno con el siguiente.

En el presente caso se ha obtenido como punto de cruce el sexto bit, éste se determinó al azar, lo que hace que v_4 y v_6 se transformen así:

$$\begin{array}{l} 101110|10 \\ 110011|01 \end{array} \rightarrow \begin{array}{l} v'_4 = 101110|01 \\ v'_6 = 110011|10 \end{array}$$

5. Esos nuevos individuos sustituyen a sus progenitores inmediatamente, por lo que la nueva población intermedia es ahora:

| i | v''_i |
|---|----------|
| 1 | 00110101 |
| 2 | 10111010 |
| 3 | 10010111 |
| 4 | 10111001 |
| 5 | 10010111 |
| 6 | 11001110 |

Tabla 2.3 Nueva población intermedia.

6. Se seleccionan ahora los bits (¡no los individuos!) que se van a mutar. Como la población contiene $8 * 6 = 48$ bits y $P_{mut} = 1\%$ en promedio mutarán 0.48 bits, lo cual quiere decir que posiblemente 1 bit sea mutado.

Como todos los bits tienen las mismas probabilidades de ser mutados, para cada uno de ellos se genera un $r_i = \text{Rand}\{0,1\}$ y si $r_i < P_{mut}$ se muta dicho bit.

Se generaron los 48 números aleatorios r_i y son menores que 0.01 el 13 y el 31, es decir, el bit número 5 de la segunda cadena y el bit 7 de la cuarta cadena.

7. Se mutan los bits correspondientes y se obtiene definitivamente la nueva población:

| i | v_i'' |
|---|----------|
| 1 | 00110101 |
| 2 | 10110010 |
| 3 | 10010111 |
| 4 | 10111011 |
| 5 | 10010111 |
| 6 | 11001110 |

Tabla 2.4 La siguiente población intermedia.

Con esto se completa una generación (una iteración). Se deberán repetir estos siete pasos tantas veces como indique el parámetro *MaxIter*. Aquí sólo lo haremos otra vez, para comprobar cómo mejora la aptitud total de la población:

1. Tabla de valores significativos asociados a cada individuo de la población recién hallada:

| i | v_i | X_i | Eval(v_i) | Aptitud(v_i) | Part(v_i) | Partacum(v_i) |
|---|----------|------------|---------------|------------------|---------------|-------------------|
| 1 | 00110101 | 0.20703125 | 0.56235 | 0.56235 | 0.06560 | 0.06560 |
| 2 | 10110010 | 0.69531250 | 1.63441 | 1.63441 | 0.19067 | 0.25627 |
| 3 | 10010111 | 0.58984375 | 1.44967 | 1.44967 | 0.16911 | 0.42538 |
| 4 | 10111011 | 0.73046875 | 1.68896 | 1.68896 | 0.19703 | 0.62241 |
| 5 | 10010111 | 0.58984375 | 1.44967 | 1.44967 | 0.16911 | 0.79153 |
| 6 | 11001110 | 0.80078125 | 1.78708 | 1.78708 | 0.20848 | 1.00000 |

Tabla 2.5 Valores de la siguiente generación.

Ahora el individuo más apto es $v_6=11001110$ y el menos apto $v_1=00110101$. El nuevo valor de la aptitud global es $Aptitud(P[1])=8.5721498$. Debe ser claro la mejora obtenida en tan solo una generación.

2. Sorteo de individuos para reproducirse. Haciendo los mismos cálculos de antes se obtiene:

- 0.45816 $\rightarrow v_4$
- 0.20190 $\rightarrow v_2$
- 0.71533 $\rightarrow v_5$
- 0.00178 $\rightarrow v_1$
- 0.91161 $\rightarrow v_6$
- 0.30624 $\rightarrow v_3$

Ahora la selección no ha tenido ningún efecto sobre la población, por lo que se obtiene como población intermedia la misma del paso anterior:

| i | v_i |
|---|----------|
| 1 | 10111011 |
| 2 | 10110010 |
| 3 | 10010111 |
| 4 | 00110101 |
| 5 | 11001110 |
| 6 | 10010111 |

Tabla 2.5 Población intermedia.

3. Selección de individuos para cruzarse. Los seis números aleatorios $r_i = \text{Rand}(0,1)$, son ahora:

0.27573 0.79741 0.56249 0.89852 0.21859 0.07933

por lo que los elegidos para la cruce serán v_1 , v_5 y v_6 de la nueva población.

Como es una cantidad impar hay que elegir uno nuevo o descartar uno de ellos al azar. En este caso resulta ser descartado v_6 , de modo que sólo se cruzarán v_1 y v_5 .

4. Cruza de los individuos anteriormente elegidos en un punto al azar. En este caso el punto de cruce resulta estar a la altura del tercer bit.

$$\begin{array}{rcl}
 101|11011 & v''_1 = 101|01110 \\
 \rightarrow & \\
 110|01110 & v''_5 = 110|11011
 \end{array}$$

5. Los nuevos individuos sustituyen inmediatamente a sus progenitores. De este modo la población intermedia es ahora:

| i | v''_i |
|---|----------|
| 1 | 10101110 |
| 2 | 10110010 |
| 3 | 10010111 |
| 4 | 00110101 |
| 5 | 11011011 |
| 6 | 10010111 |

Tabla 2.7 Nueva población intermedia.

6. Selección de bits a mutar. Dado que la probabilidad de mutación es muy baja, no hubo en esta ocasión ningún bit para ser mutado, por lo que la población intermedia se mantiene tal cual.
7. Constitución de la nueva población. Que es la misma del quinto paso.

| i | $v_i[2]$ |
|---|----------|
| 1 | 10101110 |
| 2 | 10110010 |
| 3 | 10010111 |
| 4 | 00110101 |
| 5 | 11011011 |
| 6 | 10010111 |

Tabla 2.3 La siguiente población intermedia.

Si se volviesen a calcular los valores significativos se observaría una nueva mejora de las aptitudes.

Se ha resuelto este mismo problema en una computadora utilizando una adaptación del AG básico, con los siguientes parámetros:

| |
|--------------------|
| $T_{ampob} = 30$ |
| $L = 12$ |
| $MaxIter = 50$ |
| $P_{cruza} = 60\%$ |
| $P_{mut} = 1\%$ |

obteniéndose la evolución que se muestra en la gráfica de la figura 2.2. Obsérvese cómo los mejores individuos están cerca del óptimo en muy

pocas generaciones y la evolución exponencial de la aptitud media. Con los resultados del programa, y con parámetros más adecuados se muestra la última generación en forma visual en la figura 2.3.

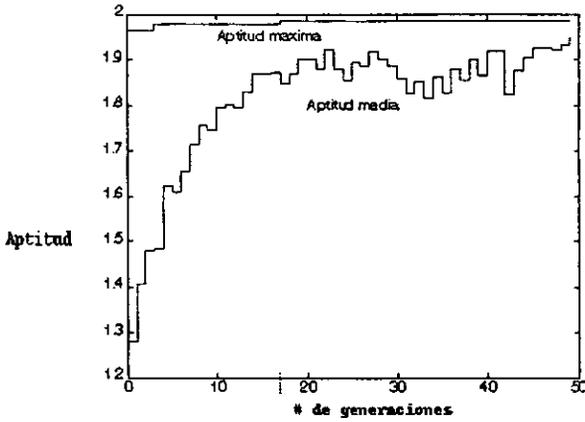


Figura 2.2: Evolución de las aptitudes.

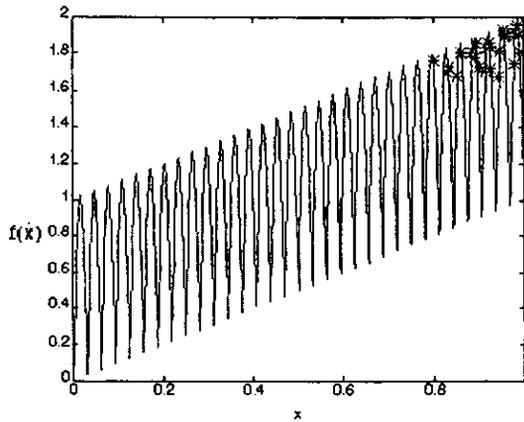


Figura 2.3: Última generación.

2.3 Problema de asignación.

El siguiente ejemplo ilustrará la forma de implementar un AG para la resolución de un problema de asignación; a su vez permitirá mostrar cómo se puede penalizar a valores que caen fuera del dominio del problema. Es un problema imaginario, basado en (Coello, 1995).

Un grupo de financieros mexicanos ha resuelto invertir 10 millones de pesos en la nueva marca de vino "Carta Nueva". Así pues, en 4 ciudades de las principales de México se decide iniciar una vigorosa campaña comercial: México en el centro, Monterrey en el noreste, Guadalajara en el occidente y Veracruz en el oriente. A esas 4 ciudades van a corresponder las zonas comerciales I, II, III y IV. Un estudio de mercado ha sido realizado en cada una de las zonas citadas y han sido establecidas curvas de ganancias medias, en millones de pesos, en función de las inversiones totales (almacenes, tiendas de venta, representantes, publicidad, etc.) Estos datos se ilustran en la tabla 2.9 y en la figura 2.4. Para simplificar los cálculos, supondremos que las asignaciones de créditos o de inversiones deben hacerse por unidades de 1 millón de pesos. La pregunta es: ¿en dónde se deben asignar los 10 millones de pesos de los que se dispone para que la ganancia total sea máxima?

| Inversión(en millones) | Beneficio 1 | Beneficio 2 | Beneficio 3 | Beneficio 4 |
|------------------------|-------------|-------------|-------------|-------------|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0.28 | 0.25 | 0.15 | 0.20 |
| 2 | 0.45 | 0.41 | 0.25 | 0.33 |
| 3 | 0.78 | 0.55 | 0.40 | 0.42 |
| 4 | 0.65 | 0.65 | 0.50 | 0.48 |
| 5 | 0.90 | 0.75 | 0.62 | 0.53 |
| 6 | 1.02 | 0.80 | 0.73 | 0.56 |
| 7 | 1.13 | 0.85 | 0.82 | 0.58 |
| 8 | 1.23 | 0.88 | 0.90 | 0.60 |
| 9 | 1.32 | 0.90 | 0.96 | 0.60 |
| 10 | 1.38 | 0.90 | 1.00 | 0.60 |

Tabla 2.9 Datos de inversiones y beneficios de las 4 ciudades.

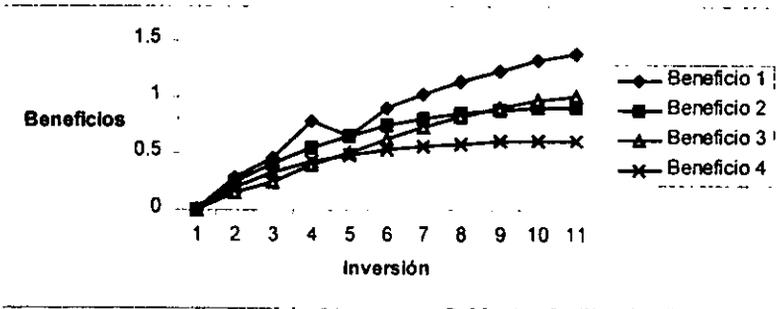


Figura 2.4 Gráfica de valores de la tabla 2.9.

1. Codificación: Necesitaremos 4 bits ($2^4=16$) para representar cada solución, porque cada una admite 11 valores posibles (de 0 a 10). Como existen 4 valores independientes (uno por cada zona de estudio), se requieren entonces 16 bits (4×4) por cada cromosoma. De manera que cada cadena será como se muestra en la figura 2.5.

| | | | | | | | | | | | | | | | |
|----------|---|---|---|----------|---|---|---|----------|---|---|---|----------|---|---|---|
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| Cadena 1 | | | | Cadena 2 | | | | Cadena 3 | | | | Cadena 4 | | | |

Figura 2.5 Cadena representativa de un cromosoma. Las cadenas 1, 2, 3 y 4 corresponden a las cantidades invertidas en las zonas económicas respectivas.

2. Individuos no factibles: Debido a que en este caso los 4 bits utilizados para representar una solución pueden producir más valores de los que se necesitan, se usará una función de ajuste que haga que los resultados producidos siempre se encuentren en el rango válido (ver punto 4).
3. Iniciación: La población inicial consta de 50 cadenas binarias elegidas al azar.

4. Funciones de evaluación y aptitud: La función de aptitud coincide con la de evaluación. Dado que el objetivo es tener las inversiones que sumen 10, y que tengan beneficio máximo, podemos usar la siguiente función de aptitud penalizada:

$$F(x) = \frac{c1 + c2 + c3 + c4}{500 * V + 1}$$

donde $c1$, $c2$, $c3$ y $c4$ son las ganancias por zona, que se calculan de acuerdo a los valores de la tabla 2.9, y V es el valor absoluto de la diferencia entre la suma obtenida de las inversiones y 10. Cuando la suma de las inversiones sea exactamente 10, la función de aptitud no será castigada.

El resultado óptimo contempla el uso de los 10 millones de pesos, ni más ni menos. Así que la suma obtenida de las inversiones, a lo largo de las generaciones, debe ser a lo sumo 10, si el resultado de la suma fuese, en algún momento, diferente de 10, entonces éste no será correcto y se debe penalizar a esa solución, la manera de hacerlo es la siguiente:

Supóngase que en la generación y la suma obtenida fue 15, como se dijo anteriormente, V es igual al valor absoluto de la diferencia entre esta suma y 10, es decir, $V = |15 - 10| = 5$. La función de ajuste es el denominador de la función de aptitud ($500 * V + 1$). En este caso $500 * 5 + 1 = 2501$ es el denominador, y la función de aptitud castigará al resultado obtenido, al dividirlo entre 2501, dando como resultado un beneficio mínimo. Nótese que cuando la suma es 10, V es igual 0, y la suma de ganancias por zona se dividirá entre 1, es decir, no habrá penalización.

5. Operadores: Utilización de dos puntos de cruce y mutación bit a bit sobre los individuos codificados.

6. Selección: Por ruleta.
7. Reemplazo: Inmediato.
8. Detención: Se correrá el algoritmo genético durante 20 generaciones.
9. Parámetros:

| |
|--------------------|
| $T_{ampob} = 50$ |
| $MaxIter = 20$ |
| $P_{cruza} = 80\%$ |
| $P_{mut} = 1.0\%$ |

2.3.1 Resultados.

El resultado obtenido, después de las 20 generaciones fue una ganancia de 1.81 millones de pesos, correspondientes a invertir 4 millones en la zona comercial 1, 3 millones en la zona 2, 1 millón en la zona 3, y 2 millones en la zona 4.

Este mismo problema se resolvió también mediante programación dinámica sin ningún problema, pero ¿qué pasaría si la inversión fuese mucho mayor, y si las cantidades fuesen menores que un millón?, el AG tendría una leve modificación, mientras que el modelo de programación dinámica se volvería inoperable.

Este fue tan sólo un ejemplo sencillo, pero muestra el potencial de los AG's para resolver problemas de optimización con restricciones. Como se vio, la parte más importante fue el diseño del modelo de la función de aptitud, que penalizó de manera adecuada a aquellos individuos no factibles, y permitió evolucionar a los mejores individuos.



Bond of Union por Escher

Capítulo

3

Fundamentos Teóricos de los AG's.

Objetivos Particulares:

- Definir conceptos teóricos básicos.
- Explicar el funcionamiento interno y justificar las razones de la eficiencia añadida de los AG's.
- Analizar la ecuación de crecimiento de los esquemas.
- Presentar el teorema fundamental de los algoritmos genéticos.
- Enunciar la hipótesis de los bloques constructores.

En la antigüedad dominaba la idea de que se podía obtener descendencia a partir de cruces entre organismos de muy distinta naturaleza. La mitología nos dice que el Minotauro muerto por Teseo era fruto de un toro y de y de una mujer, y el historiador romano Plinio manifiesta que el avestruz, recién descubierto por aquel entonces, era el producto del cruce de una jirafa con un mosquito. (Debo suponer que lo sería del apareamiento de una jirafa hembra y un mosquito macho.) Es indudable que muchos apareamientos no deben haber sido intentados en la práctica por una comprensible falta de incentivos.

Carl Sagan, 1977

En los capítulos anteriores se han centrado los esfuerzos en dejar claro cómo funcionan los AG's, la siguiente labor es explicar por qué funcionan. Este análisis permitirá mostrar con claridad qué es exactamente lo que procesan dichos algoritmos.

Por lo visto hasta ahora, pareciera que los AG's realizan una búsqueda meramente aleatoria procesando cadenas binarias, pero el hecho de utilizar precisamente una codificación binaria hace la diferencia. Para aclarar esto, tomemos el caso del ejemplo anterior (el problema de Riolo), el AG diseñado debería funcionar de igual forma con números reales que representasen directamente a las posibles soluciones, únicamente habría que modificar levemente los operadores genéticos, pero con tal representación convergería mucho más lentamente. En general utilizando codificación binaria la convergencia se acelera.

La causa de esto radica en que los AG's no procesan estrictamente individuos, sino similitudes entre ellos, o mejor dicho patrones de similitud entre individuos, y dado que cada individuo encaja en muchos patrones a la vez, la eficiencia de la búsqueda se multiplica.

3.1 Definiciones y conceptos básicos.

- El fundamento teórico de los algoritmos genéticos reside en la noción de *esquema*.

Definición: Un *esquema* es un patrón de similitud que se construye introduciendo el signo "*" que funge como comodín en el alfabeto de alelos. De este modo un esquema representa a todos los individuos que encajan en su representación, en la codificación binaria puede representar un 1 ó un 0.

Vamos a considerar el esquema H tomando 3 caracteres del alfabeto $V+ = \{0,1,*\}$. Por ejemplo, el esquema $H=*1*0$ representa a estas cuatro cadenas binarias:

0100 0110 1100 1110

Naturalmente, el esquema (101110) sólo representa a una cadena y (*****) representa a todas las cadenas de longitud $L=6$.

Proposición: Es inmediato verificar las siguientes propiedades:

- Si un esquema contiene k símbolos comodines entonces representa a 2^k cadenas binarias.
- Una cadena binaria de longitud L encaja en 2^L esquemas distintos.
- Considerando las cadenas binarias de longitud L existen en total 3^L posibles esquemas.

- Una población de Tampob cadenas binarias de longitud L contiene entre 2^L y $(\text{Tampob}) \cdot (2^L)$ esquemas distintos.

No todos los esquemas son creados igual. Algunos son más específicos que otros. Por ejemplo: el esquema 011^*1^{**} es más definido que el esquema 0^{*****} . Además ciertos esquemas abarcan más de la longitud de la cadena que otros. Por ejemplo: el esquema $1^{****}1^*$ abarca una porción más grande de la cadena que el esquema 1^*1^{****} . Para cuantificar estas ideas, introducimos 2 propiedades de los esquemas: orden y definición de longitud.

Dado un esquema H se definen:

- **Orden de un esquema, $O(H)$:** Es el número de posiciones ocupadas en la plantilla (i.e., en el alfabeto binario el número de 0's y 1's) que contiene dicho esquema.

Ejemplos:

el esquema $H = ***00^{**}1^*$ tiene un orden $O(H) = 3$

el esquema $H = *1^*1^*00^*1$ tiene un orden $O(H) = 5$

el esquema $H = *****1^*$ tiene un orden $O(H) = 1$

- **Definición de longitud:** La definición de longitud de un esquema H , denotado por $\delta(H)$ es la distancia entre la primera y la última posición ocupada en la cadena, es decir, la distancia entre las posiciones fijas extremas.

Ejemplos:

el esquema $H = ***00^{**}1^*$ tiene una definición de longitud $\delta(H) = 8-4=4$

el esquema $H = *1^*1^*00^*1$ tiene una definición de longitud $\delta(H) = 9-2=7$

el esquema $H = *****1^*$ tiene una definición de longitud $\delta(H) = 0$

Dado que un esquema H representa a $2^{L-O(H)}$ (2^k) cadenas, cuanto mayor sea el orden del esquema a menos cadenas representa. Por eso se dice que el orden de un esquema da una medida de su especificidad. Se

utiliza este concepto para calcular la probabilidad de supervivencia de un esquema frente a las mutaciones. Por otra parte, la definición de longitud da una medida de la compacidad de la información contenida en el esquema. Nótese que un esquema con una sola posición especificada tiene una definición de longitud de cero. Aquí se usará ese concepto para calcular la probabilidad de supervivencia de un esquema frente a las cruizas.

Dados una población de cadenas binarias $P[t]$ y un esquema H se definen también:

- $\xi(H, P[t])$: Presencia de H en $P[t]$. Es el número de cadenas de $P[t]$ que encajan en el esquema H .
- $Aptitud(H, P[t])$: Aptitud del esquema H en $P[t]$. Es el promedio de las aptitudes de todas las cadenas de la población que encajan en el esquema H en el instante t . Esto es, si se numeran como v_1, \dots, v_p , y si le asignamos a $p = \xi(H, t)$, las cadenas de $P[t]$ que encajan en H , resulta que:

$$Aptitud(H, P[t]) = \frac{1}{p} \sum_{i=1}^p Aptitud(v_i)$$

- $AptProm(P[t])$: Aptitud media de la población en el instante t . Es el promedio de las aptitudes de todas las cadenas de la población en la generación t , o lo que es equivalente, la aptitud del esquema $(*...*)$ en $P[t]$:

$$AptProm(P[t]) = Aptitud((*...*), P[t]) = \frac{1}{Tampob_{v_i \in P[t]}} \sum_{v_i \in P[t]} Aptitud(v_i)$$

- $AptRel[t]$: Aptitud relativa de H en $P[t]$. Es el cociente entre la aptitud del esquema en la población y la aptitud media de la población en cierto instante, i.e.,

$$AptRelProm(H, P[t]) = \frac{Aptitud(H, P[t])}{AptProm(P[t])}$$

Partiendo de estas definiciones ya se puede comenzar el análisis teórico del funcionamiento de los AG's.

3.2 La Ecuación de crecimiento de los esquemas.

La siguiente ecuación resulta esencial para comprender el modo de operación de los algoritmos genéticos. Con ella se muestra cómo evoluciona en promedio un esquema dentro de la población de un algoritmo genético.

Ecuación de crecimiento reproductivo de un esquema en una población: Sea $P[t]$ una población de $Tampob$ elementos de tamaño L (con $Tampob$ lo suficientemente grande) que evoluciona a través de un AG clásico (SGA). Abréviase Ecuación de Crecimiento Reproductivo como [ECR]. La presencia de un esquema H en la población $P[t]$ evoluciona en promedio según la siguiente fórmula:

$$\xi(H, P[t+1]) = \xi(H, P[t]) \cdot f_c \cdot f_s \quad [ECR]$$

donde f_c es el factor de crecimiento de H en la población y f_s es el factor de supervivencia. El factor de crecimiento mide la tendencia del esquema a aumentar su presencia en la población intermedia y el factor de supervivencia mide la probabilidad de que pase a la siguiente

generación. El elemento que hace aumentar la presencia de los esquemas en la población intermedia es la selección; de hecho es muy sencillo ver que, si se usa una selección por sorteo, la presencia de un esquema en la población intermedia $Q[t]$ evoluciona en promedio según:

$$\xi(H, Q[t]) = \xi(H, P[t]) \cdot \text{AptProm}(H, P[t]) = \xi(H, P[t]) \frac{\text{Aptitud}(H, P[t])}{\text{AptProm}(P[t])}$$

de manera que:

$$f_c = \text{AptProm}(H, P[t])$$

En palabras, un esquema en particular se desarrolla a razón de la aptitud promedio del esquema por la aptitud promedio de la población. De otra forma, un esquema con valores de aptitud por encima del promedio de la población recibirá un incremento en el número de muestras en la siguiente generación, mientras que esquemas con valores de aptitud por debajo del promedio recibirán un decremento en el número de muestras. Es interesante observar que este comportamiento esperado es llevado a cabo en cada esquema H contenido en una población particular P en paralelo. En otras palabras todos los esquemas de una población se desarrollan o mueren de acuerdo a los esquemas promedio dentro de la operación de reproducción. En un momento examinaremos por qué esto puede ser bueno.

Por otro parte, la reproducción por sí sola no hace nada por promover la exploración de nuevas generaciones en el espacio de búsqueda, ya que nuevos puntos no son buscados si sólo copiamos viejas estructuras sin cambio. Entonces ¿cómo probar con estructuras nuevas? Aquí es donde entran los pasos de la cruz.

La **cruza** es una estructura de intercambio de información aleatoria entre cadenas, la **cruza** crea nuevas estructuras con un mínimo de ruptura en la estrategia de distribución, dictada solamente por la reproducción. Esto da como resultado un incremento (decremento) exponencial en las proporciones de los esquemas de una población.

Ahora bien, existe la posibilidad de que los esquemas sean destruidos por los operadores genéticos antes de pasar a la próxima generación; es esta posibilidad la que cuantifica el factor de supervivencia f_s . Tal factor es difícil de calcular exactamente, pero se puede acotar mediante:

$$f_s \geq \left(1 - \frac{P_{cruza} \delta(H)}{L-1}\right) \cdot (1 - P_{mut})^{l(H)}$$

El primer factor mide la probabilidad de que H sobreviva a una **cruza**, o más exactamente, la probabilidad de que la **cruza** no ocurra sobre su definición de longitud.

Para ver cuáles esquemas son afectados por la **cruza** y cuáles no, consideramos una cadena, A en particular, de longitud $L=7$, y 2 esquemas representativos de la cadena.

$$\begin{aligned} A &= 0111000 \\ H_1 &= *1****0 \\ H_2 &= ***10** \end{aligned}$$

Claramente los 2 esquemas están siendo representados en A , pero hay que ver el efecto de la **cruza** en los esquemas. Primero volvemos a llamar al procedimiento simple de **cruza** con selección aleatoria de parejas y del sitio de **cruza**, y el intercambio de subcadenas desde el

comienzo de la cadena al sitio de **cruza** inclusive con la correspondiente subcadena escogida como pareja. Supongamos que la cadena A ha sido escogida para hacer pareja y **cruza**. Para esta cadena de longitud 7, supongamos que rolamos un simple dado a fin de escoger el sitio de **cruza** (Existen 6 sitios en una cadena de longitud 7). Además supongamos que el dado cae en 3, significa esto que el corte de **cruza** tomará lugar entre las posiciones 3 y 4. El efecto de esta **cruza** en nuestros 2 esquemas H_1 y H_2 puede ser visto fácilmente en el siguiente ejemplo, donde el sitio de **cruza** ha sido marcado por el símbolo separador “|” :

$$\begin{aligned} A &= 011 | 1000 \\ H_1 &= *1* | ***0 \\ H_2 &= *** | 10** \end{aligned}$$

A menos que las cadenas que formen pareja sean idénticas, las posiciones serán las mismas en el esquema (una probabilidad que conservadoramente se ignoró), el esquema H_1 será destruido porque el “1” en la posición 2 y el “0” en la posición 7 serán puestos en diferentes descendientes (los valores están en lados opuestos por el separador |), es igualmente claro que con el mismo punto de **cruza** el esquema H_2 sobrevivirá, porque el “1” en la posición 4 y el “0” en la posición 5 serían colocados intactos a un descendiente. Aunque hemos utilizado un punto de corte específico para la ilustración, es claro que el esquema H_1 tiene menos probabilidades de sobrevivir en la **cruza** que el esquema H_2 porque en promedio el punto de **cruza** es más probable que caiga entre las posiciones fijas extremas.

Para cuantificar esta observación, notemos que el esquema H_1 tiene una definición de longitud igual a 5. Si el sitio de **cruza** es seleccionado uniformemente al azar entre $L-1=7-1=6$ posibles sitios, entonces claramente el esquema H_1 es destruido con probabilidad

$P_d = \delta(H_1)/(L-1) = 5/6$ (sobrevivir con probabilidad $P_s = 1 - P_d = 1/6$). Similarmente, el esquema H_2 tiene una definición de longitud $\delta(H_2) = 1$, y sería destruido en un evento de seis, donde el sitio de **cruga** sea seleccionado entre las posiciones 4 y 5 tal que $P_d = 1/6$ o la probabilidad de sobrevivir es $P_s = 1 - P_d = 5/6$.

Más generalmente, vemos que el límite inferior de la probabilidad de sobrevivir en la **cruga** P_s puede ser calculado para cualquier esquema. Un esquema sobrevive cuando el sitio de **cruga** cae fuera de la definición de longitud. La probabilidad de sobrevivir bajo una simple **cruga** es $P_s = (1 - (\delta(H_1)/(L-1)))$, ya que el esquema será probablemente roto en cualquier sitio seleccionado de $L-1$ posibles sitios. Si la **cruga** es por sí misma ejecutada por selección aleatoria, sea P_c la probabilidad de un emparejamiento en particular, la probabilidad de sobrevivir puede ser

$$P_s \geq 1 - P_c \cdot \frac{\delta(H)}{L-1},$$

dada por la expresión:

la cual se reduce a la expresión anterior cuando $P_c = 1.0$.

Con ambos, **cruga** y reproducción, el factor depende de dos elementos: si el esquema está por encima o por debajo del promedio de la población, y si el esquema tiene una definición de longitud relativamente corta o larga. Claramente estos esquemas con promedio por encima del observado y definición de longitud corta, están siendo ejemplificados en tasas incrementadas exponencialmente.

El segundo factor mide la probabilidad de que el esquema no se vea afectado por una mutación. Para que un esquema H sobreviva, todas las posiciones especificadas deben sobrevivir. Sin embargo, ya que un simple alelo sobrevive con probabilidad $(1 - P_{mut})$, y ya que cada mutación es independiente, un esquema en particular sobrevive cuando cada una

de las $O(H)$ posiciones especificadas sobreviven con el esquema. Multiplicando la probabilidad de supervivencia $(1-P_{mut})$ por ella misma $O(H)$ veces, obtenemos la probabilidad de sobrevivir en la mutación $(1-P_{mut})^{O(H)}$. Para pequeños valores de P_{mut} ($P_{mut} < 1$), la probabilidad de sobrevivir del esquema puede aproximarse por la expresión $1 - O(H) \cdot P_{mut}$.

En definitiva, siendo P_{cruza} y P_{mut} las probabilidades de cruza y mutación respectivamente del SGA, la presencia de cierto esquema H en la población $P[t]$ evoluciona en promedio según:

$$\xi(H, P[t]) \geq \xi(H, t) \cdot AptProm(H, P[t]) \cdot \left(1 - \frac{P_{cruza} \delta(H)}{L-1}\right) \cdot (1 - P_{mut})^{O(H)}$$

Formulación aproximada: Como habitualmente $P_{mut} < 1$ se puede hacer

$$(1 - P_{mut})^{O(H)} \cong 1 - P_{mut} O(H)$$

de modo que,

$$f_i \geq \left(1 - \frac{P_{cruza} O(H)}{L-1} - P_{mut} O(H)\right)$$

lo que proporciona la siguiente formula aproximada de la ecuación {ECR}:

$$\xi(H, P[t]) \geq \xi(H, P[t]) \cdot ApProm(H, P[t]) \cdot \left(1 - \frac{P_{cruz} \delta(H)}{L-1} - P_{mut} \alpha(H) \right)$$

Concluimos que un esquema en particular H recibe un número esperado de copias en la siguiente generación dependiendo de la reproducción, cruza y la mutación. Esquemas cortos, de orden bajo y con alto promedio, reciben un incremento exponencial de muestras en las subsecuentes generaciones. Esta conclusión es importante, tanto que le daremos un nombre especial: el Teorema del Esquema o el Teorema Fundamental de los algoritmos genéticos.

3.3 El Teorema fundamental de los algoritmos genéticos.

Basándose en los resultados de la sección anterior cabe enunciar el siguiente teorema:

Teorema fundamental de los algoritmos genéticos. En las sucesivas generaciones de un SGA, la presencia de un esquema θ en la población $P(t)$ evoluciona de modo exponencial según la ecuación [ECR].

De la ecuación [ECR] se extraen inmediatamente las siguientes consecuencias prácticas:

- La presencia de un esquema en una población evoluciona en progresión geométrica, progresión cuyo factor está determinado por la aptitud relativa del esquema, su definición de longitud y su orden. Es fundamental darse cuenta que los esquemas con una aptitud por encima de la media (abreviadamente, esquemas aventajados) incrementan exponencialmente su presencia en sucesivas generaciones ($f_c > 1$), mientras que los que tienen la aptitud por debajo de la media (esquemas retrasados) decremantan exponencialmente su presencia en la población ($f_c < 1$).
- La tendencia de los esquemas aventajados a incrementar su presencia en sucesivas generaciones se acentúa cuando el esquema es corto y de bajo orden, pues entonces $f_s \cong 1$.

Desde esta perspectiva, se puede explicar la introducción de los tres operadores del SGA de la siguiente manera:

1. **Selección:** La selección se encarga de incrementar geoméricamente la presencia de los esquemas aventajados y reducir la presencia de los retrasados. No obstante, la selección no introduce nuevos esquemas.
2. **Cruza:** La **cruza** permite el intercambio estructurado de información útil (esquemas cortos, de bajo orden y aventajados) entre individuos. Entonces la **cruza** es un operador esencial para el eficaz funcionamiento de un AG.
3. **Mutación:** La mutación introduce variedad en el juego de esquemas de la población y proporciona un mecanismo de seguridad frente a posibles pérdidas de información valiosa. Vista de esta manera, la mutación es un operador secundario; de ahí que se aplique con bastante menor frecuencia que la **cruza**.

A partir de todos estos resultados prácticos se formula la siguiente hipótesis, que no ha sido completamente probada:

Hipótesis de los bloques constructores. Los algoritmos genéticos exploran el espacio de búsqueda a través de la yuxtaposición de esquemas aventajados, cortos y de bajo orden. A tales esquemas se les llama bloques constructores.

Así quedan respondidas las dos cuestiones formuladas al principio de esta sección:

1. *¿Por qué funcionan los AG's?* Porque dan oportunidades de proliferación exponencialmente crecientes a los esquemas más aptos, y exponencialmente decrecientes a los menos aptos.
2. *¿Qué procesan los AG's?* En sentido amplio los AG's procesan esquemas y en particular los AG's procesan de modo útil bloques constructores.

Además queda justificada la exigencia del uso de representación binaria que se comentó al principio: la representación binaria es la que, para un tamaño dado, proporciona la máxima diversidad de esquemas.

Vista desde otra perspectiva, la hipótesis de los bloques constructores afirma que cuanto más intervenga la construcción por bloques más eficaz será el funcionamiento del AG. Esa *eficacia añadida* se puede cuantificar a través del siguiente resultado, que no se demostrará:

Propiedad del Paralelismo Implícito de los AG's: Como ya quedó dicho, en una generación de Tampob individuos binarios de tamaño L existen entre 2^L y $\text{Tampob} * 2^L$ esquemas distintos. Algunos de ellos, mayormente los bloques constructores, serán procesados de modo útil, esto es, reproducidos exponencialmente y no alterados por los operadores genéticos. Pues bien, el mecanismo subyacente que da eficacia a los AG's a pesar de su generalidad se basa en el hecho de que, aunque el AG sólo procesa Tampob estructuras en cada generación, se puede demostrar que, bajo hipótesis muy generales, se procesan de modo útil al menos $(\text{Tampob})^3$ esquemas.

Es decir, si los AG's son más robustos que otros métodos de búsqueda ciega, ello se debe a que realizan implícitamente una búsqueda en paralelo. Dicho paralelismo implícito se obtiene sin ningún requerimiento añadido de memoria o de potencia de cálculo, y seguramente sea el más destacado ejemplo de una explotación combinatoria que actúa de modo beneficioso.

Se pueden ilustrar estos resultados volviendo al ejemplo descrito en la sección anterior (Riolo). Los esquemas más aventajados son (****100) y (11111***). El primer esquema representa a los puntos más altos de cada pico y el segundo representa a todos los puntos del último pico. Si la población inicial es lo suficientemente variada, en las primeras etapas el AG favorecerá la proliferación de los individuos que

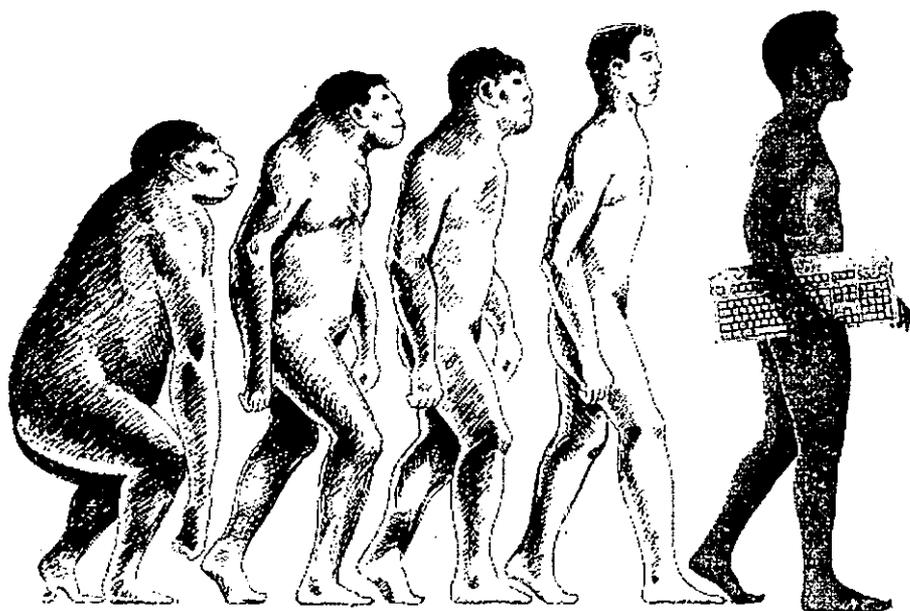
contengan el primer esquema, después dará preferencia a aquéllos que se ajusten más al segundo, hasta finalmente dar con el punto óptimo 11111100, que combina ambos.

3.3.1 Limitaciones prácticas del análisis teórico.

Todos los resultados anteriores están basados en la ecuación [ECR] la cual presupone tres requisitos:

1. Por motivos técnicos, se debe imponer que la función de aptitud devuelva siempre valores no negativos.
2. Se admite que el número de generaciones del AG es infinito. Como esto no puede ser, se supone un número de generaciones suficientemente grande.
3. Se admite que el número de individuos de la población es infinito. Esto tampoco es posible, pero Tampob debe ser lo suficientemente grande como para mantener en todo momento una población variada.

Aún queda sin considerar otra limitación que surge en la práctica: se ha supuesto implícitamente que la codificación es perfecta; esto conlleva, entre otras cosas, que a cada punto del dominio del problema le corresponde uno y sólo un individuo codificado en el espacio de búsqueda. No siempre es este el caso, por lo que se deberán habilitar procedimientos auxiliares que penalicen a valores que no entran en el dominio, que de un modo u otro perjudicaría a la eficiencia del AG.



La evolución del Hombre hasta nuestros días.

Capítulo

4

Mejoras y Limitaciones de los AG's.

Objetivos Particulares:

- Analizar las limitaciones de los AG's (debilidad, diversidad, convergencia prematura, representación y codificación).
- Mostrar distintas variantes o alternativas a fin de mejorar el desempeño del AG.
- Proponer mejoras en las operaciones de los AG's.

La diferencia entre el intelecto humano y el de los animales superiores, con ser grande, es básicamente de grado y no de especie... Si pudiera demostrarse que determinadas funciones mentales de orden superior, como la formación de conceptos generales, la capacidad introspectiva, etcétera, son exclusivas del hombre, cosa que me parece muy dudosa, no es improbable que tales facultades se nos mostraran como simple resultado accesorio de otras cualidades intelectuales superiores que, a su vez, son en buena medida consecuencia del uso continuado de una lengua perfecta.

Charles Darwin, 1866

A la hora de llevar a la práctica el diseño de los algoritmos genéticos para resolver problemas no triviales de optimización se plantean varias dificultades de orden teórico y práctico. Todas ellas surgen como consecuencia de cuatro limitaciones intrínsecas de dicho modelo; por eso para corregir o mitigar sus efectos conviene estudiar antes, con cierto detenimiento, las implicaciones generales de tales limitaciones. Son precisamente estas limitaciones, las que han permitido buscar mejoras a los procedimientos básicos del algoritmo.

4.1 El Problema de la debilidad de los AG's.

El inconveniente más visible de la debilidad es la ineficiencia. Sin embargo, los métodos débiles de resolución de problemas presentan las ventajas de ser muy poco específicos, poco exigentes y notablemente eficaces; no obstante, todos ellos son ineficientes en mayor o menor grado, y especialmente si se les compara con métodos heurísticos. Los AG's sólo se pueden considerar eficientes en comparación con otros métodos estocásticos de búsqueda ciega, pero se puede garantizar que si se encuentra un método heurístico para resolver un problema, éste siempre lo hará mucho más eficientemente que un AG. En algunos casos la única forma de eliminar la debilidad será incorporando conocimiento específico al AG.

Pero eso no es todo, otro de los inconvenientes prácticos de la debilidad es la tendencia al extravío de la búsqueda. El fenómeno es así: el AGB realiza la búsqueda de los mejores puntos utilizando únicamente la aptitud de los individuos para recombinar internamente los *bloques constructores*. En ocasiones ocurre que la información proporcionada resulta insuficiente para orientar correctamente al algoritmo en su búsqueda del óptimo. A esto se le llama desorientación¹. A nivel interno esto se concreta en que no se verifica la hipótesis de los *bloques constructores*, es decir, ciertas combinaciones válidas de buenos *bloques constructores* originan individuos de baja aptitud. En el peor de los casos puede ocurrir que este fenómeno tenga tanta fuerza como para impedir que el AG converja al óptimo global, desviándolo finalmente hacia un óptimo local. Esto es el extravío propiamente dicho. Afortunadamente, aunque un AG se desoriente no necesariamente se va a extraviar, para eso es necesario partir de una mala población inicial o plantear un problema especialmente diseñado para que se extravíe.

¹ En la bibliografía el término desorientación es usado como "deception"

En resumen, para que funcione correctamente el mecanismo básico de los AG's (la construcción por bloques) es necesario proporcionarle una mínima cantidad y calidad de información. Si no se le proporciona ese mínimo, el mecanismo no funcionará con propiedad, y el AG evolucionará incorrectamente.

Como es de suponer, el factor que más contribuye a la existencia de desorientación es la forma de la *función de aptitud*. Típicamente, la desorientación es frecuente en problemas en los que los puntos óptimos están aislados y rodeados de puntos de muy baja aptitud. Sin embargo, conviene señalar que la presencia de desorientación suele estar fuertemente estimulada por una mala codificación que oculte la ya de por sí escasa información disponible. A nivel interno esto se explica por la mayor dificultad para constituir los *bloques constructores*. De modo recíproco, una buena codificación reduce la probabilidad de extravío.

Un caso especialmente desfavorable ocurre cuando hay una fuerte interacción entre dos o más atributos (i.e., genes), de tal forma que la contribución a la aptitud de un individuo que realiza cierto gen depende grandemente de los valores que tomen otros. A este fenómeno se le dice epistasis o acoplamiento y su presencia garantiza la desorientación dado que en esas condiciones resulta muy difícil construir buenos *bloques constructores*.

4.2. El Problema de la diversidad en los AG's.

A efectos prácticos, es fundamental que todo AG verifique lo siguiente:

Requisito de variedad en los AG's: Es esencial para el buen funcionamiento de un AG tener controlada en todo momento la diversidad de la población. Se entiende la diversidad en sentido general

como "diversidad de individuos" y en particular como "diversidad de aptitudes".

La necesidad de que haya diversidad de individuos es fácil de entender. Con poca variedad de individuos hay poca variedad de esquemas, a causa de ello el operador de cruza pierde casi por completo la capacidad de intercambio de información útil entre individuos, y en definitiva, la búsqueda se estanca. La necesidad de tener controlada la diversidad de aptitudes radica en la imposibilidad práctica de trabajar con una población infinita, como enseguida se verá.

Es fácil darse cuenta de que no es conveniente tener poca diversidad de aptitudes ya que en tal caso todos los individuos tendrían más o menos las mismas posibilidades de sobrevivir, la selección reproduciría la situación anterior y todo el peso de la búsqueda recaería en los operadores genéticos, lo que a la larga sería poco más que una búsqueda aleatoria. El resultado final es que la búsqueda se estanca y, como luego se verá, la situación puede empeorar si la población es pequeña. En definitiva, para que la selección sea efectiva, la población debe contener en todo momento una cierta variedad de aptitudes.

Por otra parte, como la población es finita, tampoco se puede tener gran disparidad de aptitudes, pues ello suele afectar muy negativamente a la diversidad de la población, como seguidamente se explica.

4.3. Convergencia prematura por problemas con la diversidad.

Sea un AG básico con población finita y admítase por simplicidad que la *función de aptitud* coincide con la función de evaluación. En algún momento puede ocurrir que un individuo o un grupo de ellos obtengan una aptitud notablemente superior a los demás. Esto es especialmente probable en las fases tempranas de la evolución, en las cuales de entre

una población de individuos mediocres suele surgir un buen candidato por aplicación de los operadores genéticos. En tal circunstancia existe el riesgo de que se produzca una evolución en avalancha, al incrementar los individuos más aptos su presencia en la población, por ser ésta finita la diversidad disminuye, ello hace que en la siguiente generación se favorezca aún más a los individuos más aptos hasta que éstos acaban dominando por completo la población. A esto se le conoce como el *fenómeno de los superindividuos*. Habitualmente ocurre que tales superindividuos sólo son los más aptos en cierto momento, pero no los más aptos absolutos (téngase en cuenta que la falta de diversidad estanca la búsqueda) lo que en último término provoca una convergencia prematura del AG, habitualmente hacia un subóptimo.

4.4. El Problema de la representación y la codificación.

Como ya quedó dicho, los AG's procesan esquemas, esto es, patrones de similitud. Por ello, con la idea de hacer la búsqueda más eficiente, se representan los puntos del espacio de búsqueda mediante cadenas binarias pues ésta es la representación que proporciona la mayor variedad de esquemas para una longitud dada. Esto plantea graves inconvenientes que se resumen en lo siguiente: la representación binaria no resulta útil para la mayoría de problemas de búsqueda en el sentido de que no siempre es posible encontrar una codificación que dé significado a cada una de las posiciones de que consta el genotipo; es más, aunque se encuentre, eso no garantiza que represente con fidelidad el dominio del problema. Ambas cuestiones se estudiarán por separado.

Comenzando por la segunda cuestión (la representación con fidelidad del dominio del problema), es conveniente introducir la siguiente definición: Se dice que una representación es perfecta, desde el

punto de vista de los objetos representados, cuando cumple estas cinco propiedades:

- ❖ **Completez:** Debe poder representar todos los objetos de la clase indicada.
- ❖ **Coherencia:** Únicamente debe representar objetos de esa clase.
- ❖ **Uniformidad:** Todos los objetos deben estar representados por la misma cantidad de codificaciones, i.e., no debe haber objetos "sobrerrepresentados" ni "infrarrepresentados".
- ❖ **Sencillez:** Debe ser fácil de aplicar el mecanismo de codificación objeto \leftrightarrow individuo tanto en sentido directo como inverso.
- ❖ **Localidad:** Pequeños cambios en los individuos se han de corresponder con pequeños cambios en los objetos.

En la práctica suele ser imposible encontrar una representación perfecta del dominio del problema a través de cadenas binarias, lo cual introduce una diferencia más o menos grande entre lo que se está buscando y lo que se quiere buscar.

La otra cuestión, la de la utilidad de la representación, no es trivial. Es muy importante que las L posiciones de que consta el genotipo tengan algún significado concreto; de este modo, si se ordenan apropiadamente dichas posiciones, los *bloques constructores* serán cortos y compactos, lo que dará eficacia a la búsqueda genética.

Toda esta discusión se sintetiza en la siguiente recomendación:

Se aconseja siempre realizar la elección de la representación basándose en estos dos principios:

Principio de los *bloques constructores* con significado: Las zonas del espacio de búsqueda relevantes para el problema deben poderse representar mediante esquemas cortos y de bajo orden (*bloques constructores*).

Principio del alfabeto de símbolos mínimo: Se debe elegir el alfabeto de símbolos más pequeño (preferentemente binario) que permita una expresión natural del problema.

Existe una manera bastante eficaz de estimular la formación de buenos *bloques constructores*, sometiendo a la propia distribución de posiciones al proceso de evolución. Ello se consigue introduciendo un nuevo operador genético llamado *inversión* o también **mutación estructural**.

La *inversión* es un operador unitario que selecciona dos puntos en una cadena e invierte el orden de las posiciones intermedias recordando el significado de cada una. Esto último obliga a identificar los bits en las cadenas, lo cual se lleva a cabo colocando a cada alelo una etiqueta con su posición.

Por ejemplo, al invertir la cadena

$$(0_1 0_2 0_3 1_4 1_5 0_6 1_7 0_8 0_9 0_{10} 1_{11})$$

entre los puntos señalados se obtiene:

$$(0_1 0_2 0_3 1_7 0_6 1_5 1_4 0_8 0_9 0_{10} 1_{11})$$

Nótese que el contenido de la cadena no se altera, tan sólo se reorganiza su estructura; dicho de otro modo, tras la inversión la cadena sigue representando al mismo individuo, sólo que con otra estructura. De ahí el nombre de **mutación estructural**.

4.5 Mejoras a los operadores genéticos.

Se van a describir algunas modificaciones del modelo básico de algoritmos genéticos. Se mantendrá punto por punto la estructura fundamental, introduciendo simplemente variantes y parametrizaciones en las configuraciones.

4.5.1 Convergencia en los AG's y criterios de terminación.

Hasta el momento se ha aceptado por defecto como criterio de terminación de los AG's el del máximo número de iteraciones. Se admite que, dada la hipótesis de la construcción por bloques y una adecuada población inicial, los mejores individuos de la población se acercarán cada vez más al óptimo. Es decir, se asume implícitamente que al llegar a cierto valor de t el algoritmo ha convergido a efectos prácticos. Esta manera de determinar la convergencia es excesivamente arbitraria en muchos casos, lo que lleva a plantear otros criterios de terminación.

Tradicionalmente se distinguen dos tipos de criterios de terminación:

- ❖ Criterios de terminación enfocados al costo: Abreviadamente, criterios de "tipo MAX". Son los que limitan a priori el número máximo de iteraciones ($MaxIter$) o el máximo número de evaluaciones ($MaxNEval$)². Son los más sencillos, pero no siempre tienen significado, ya que normalmente no hay ninguna razón para dar a dichos números máximos uno u otro valor. En tal circunstancia o bien ocurrirá que el AG "se quede corto", es decir, no esté al terminar lo suficientemente cerca del óptimo, o bien consumirá recursos computacionales inútilmente estando ya lo suficientemente cerca del óptimo.

² habitualmente usados cuando la función de evaluación es complicada.

- ❖ **Criterios de terminación enfocados a la calidad:** También se les conoce como criterios incrementales de terminación o, abreviadamente, criterios de tipo *TOL*. Se hace parar al algoritmo una vez que se hayan superado ciertos requisitos de convergencia, independientemente de la iteración en que se encuentre. Dependiendo del criterio práctico que se utilice para evaluar la convergencia se subdividen en dos tipos:

Enfocados a la estructura: Se evalúa la convergencia de la población verificando la cantidad de genes que han convergido. Se considera que un gen del genotipo ha convergido cuando cierto porcentaje de la población (alrededor del 90%) tiene los alelos iguales en dicho gen (o parecidos, si la representación no es binaria). La búsqueda termina cuando el número de genes que han convergido excede cierto porcentaje (alrededor del 95%) del total de genes de que consta el genotipo.

Enfocados al contenido: Miden el progreso hecho por el algoritmo en cierto número de iteraciones; si es menor que cierta tolerancia *TOL*, que es un parámetro del método, la búsqueda ha terminado. Habitualmente se mide el progreso hecho por el algoritmo como el incremento relativo de la aptitud total de la población. También se puede usar la aptitud máxima o la mínima, según sea el problema.

La elección de unos valores u otros para las tolerancias es delicada. No sólo depende del problema, sino que también influye la suerte que se haya tenido en la ejecución del AG (recuérdese que son métodos estocásticos). Además, una pequeña reducción de las tolerancias permitidas puede conllevar un tiempo de ejecución mucho más grande.

Habitualmente, se usa un criterio de terminación híbrido que consta de una tolerancia y un mecanismo de seguridad del tipo *MAX* para cuando se alargue demasiado la búsqueda.

4.5.2 Elitismo.

El método más utilizado para mejorar la convergencia de los AG's es el *elitismo*. Consiste básicamente en realizar la etapa de selección en dos partes:

1. Se selecciona una élite de r miembros de entre los mejores de la población inicial y se incorporan directamente a la población final de cada generación, sin pasar por la población intermedia.
2. La población auxiliar de progenitores se selecciona de entre los $n-r$ restantes miembros de la población inicial.

Comúnmente el tamaño de la élite r es bastante pequeño (1 ó 2 para $n=50$), y el tipo de muestreo es o bien directo, o bien por sorteo.

Bajo ciertas condiciones muy generales, la introducción del *elitismo* garantiza la convergencia teórica al óptimo global; en la práctica, mejora la velocidad de convergencia de los AG's cuando la función de evaluación es unimodal, es decir, cuando no existen subóptimos; sin embargo la velocidad de convergencia empeora con funciones fuertemente multimodales.

Otra propuesta es mantener en forma paralela dos subpoblaciones, la primera será la elitista, mientras que la segunda contendrá al resto de la población, en vez de pasarlos directamente a una población final común. Se aplicarán los operadores a ambas subpoblaciones, la idea es hacer generar un grupo de buenos individuos para combinarlos después con el resto, de manera que compartan las buenas características. Se puede describir en los siguientes puntos:

1. Se selecciona una élite de r miembros, que formarán la subpoblación elitista y una subpoblación de $n-r$ miembros que conforman al resto de la población inicial.

2. Se aplican operadores genéticos a ambas subpoblaciones.
3. Se genera una nueva población combinando la subpoblaciones de manera que se apliquen otra vez los operadores genéticos, pero en esta ocasión para combinar las buenas características de la élite con el resto de la población.
4. Se mantiene la población elitista para la siguiente generación, a excepción de que surja un mejor individuo de la operación anterior, y que a su vez reemplazará al elemento menos apto de la élite.

Dadas las características de este criterio, es conveniente que la subpoblación elitista tenga un tamaño equivalente a un 10% de la población inicial.

4.5.2.1. Reiniciación periódica.

Se propone, para poblaciones pequeñas y en las primeras generaciones del algoritmo, establecer una serie de reiniciaciones periódicas. En cada generación se salvan los mejores individuos, se reinician los demás y se vuelve a comenzar. La reiniciación tiene efectos beneficiosos sobre las prestaciones del método debido a que introduce diversidad, requisito especialmente crítico en los AG's con poblaciones pequeñas. En una población de 50 individuos, sería prudente reiniciar de 5 a 10 veces para seleccionar a 10 ó 5 individuos respectivamente por reiniciación; no sería mucho tiempo destinado a ello, y en cambio sería mucho el logro obtenido en diversidad y aptitud promedio

4.5.3 Elección de las funciones de evaluación y de aptitud.

Como ya se indicó, para poder ejecutar un AG es preciso tener definida una función de evaluación de individuos y una *función de aptitud* que se deriva directamente de ella por desplazamiento y escalado. Es necesario introducir la distinción entre la *función de aptitud* y la de evaluación por dos motivos:

- ❖ Para hacer que las evaluaciones tomen valores positivos.
- ❖ Para controlar la diversidad de la población.

4.5.3.1 Desplazamiento y escalado de la función de evaluación.

Una vez elegida la función de evaluación, se construye la *función de aptitud* desplazándola y escalándola convenientemente.

El desplazamiento tiene como finalidad principal hacer que la *función de aptitud* devuelva valores positivos. El procedimiento más usado para hacer positivas las evaluaciones es el de la ventana de desplazamiento. Consiste en desplazar las evaluaciones así:

$$\text{Aptitud}(v) = F - \text{Eval}(v) \quad (\forall v \in P)$$

donde F es una cota superior para las evaluaciones, esto es,

$$(\forall t) \forall v \in P \text{ Aptitud}(v[t]) \leq F$$

No es habitual conocer a priori el rango de valores que va a tomar la función de evaluación, por lo que es preferible ir adaptando dinámicamente el valor de F del siguiente modo:

ESTO NO DEBE
SALIR DE LA BIBLIOTECA

$$\text{Aptitud}(v(t)) = F_{\max}[t] - \text{Eval}(v(t)) \quad (\forall v(t) \in P(t))$$

donde $F_{\max}[t]$ es la máxima aptitud de cualquier individuo evaluado hasta la iteración inmediatamente anterior, $t-1$.

Ahora bien, si el valor de $F_{\max}[t]$, o en su caso el de F , no está lo suficientemente cerca de la máxima aptitud de la presente generación, la búsqueda se frena e incluso se puede estancar. La causa está una vez más en la pérdida de la diversidad de aptitudes. Efectivamente, si $F_{\max}[t]$ es mucho mayor que cualquier evaluación presente, las correspondientes aptitudes serán todas muy parecidas y la búsqueda se estancará. Supóngase, por ejemplo, que $F_{\max}[t] = 100$ pero que en el presente instante la población sólo contiene individuos cuyas evaluaciones proporcionan valores entre 5 y 10. Evidentemente las aptitudes variarán entre 90 y 95. Sin embargo, si fuera $F_{\max}[t] = 15$ las aptitudes estarían comprendidas entre 5 y 10, y el individuo más apto tendría ahora doble aptitud que el menos apto. En resumen, se ha mejorado la capacidad de discriminación de aptitudes.

En definitiva, resulta conveniente ir actualizando el valor de F_{\max} cada cierto número de generaciones con el fin de evitar una posible pérdida de la diversidad. El periodo de actualización W es un parámetro del método que suele oscilar entre 0 y 5, conviniéndose en que $W = 0$ representa al caso en que F_{\max} nunca se actualiza.

Como ya se comentó, una de las maneras de controlar la diversidad de las aptitudes es definiendo una *función de aptitud* neta, distinta de la función de evaluación. Esto es lo que se conoce como escalado de la función de evaluación. Mediante el escalado se trata de evitar un fenómeno con causas opuestas al anterior pero de similares efectos: es el fenómeno ya descrito de la convergencia prematura hacia superindividuos. No obstante, la finalidad del escalado es más general; siguiendo a Goldberg (Goldberg 1989).

Desde el estudio pionero de De Jong, el escalado de los valores de la función objetivo se ha convertido en una práctica ampliamente aceptada. Se hace esto para mantener los niveles apropiados de competición a lo largo de una simulación. Sin escalado, en las fases tempranas hay una tendencia a que unos pocos superindividuos dominen el proceso de selección. En este caso, los valores de la función objetivo deben ser subescalados para evitar que tales superindividuos ocupen la población. Más tarde, cuando la población ha convergido casi en su totalidad, la competición entre los miembros de la población es menos intensa y la simulación tiende a evolucionar erráticamente. En este caso los valores de la función objetivo deben ser sobreescalados para acentuar las diferencias entre los miembros de la población con el fin de continuar favoreciendo a los mejores miembros.

Es común incluir el desplazamiento, directa o indirectamente, en el propio algoritmo de escalado. En caso de que no se haga así, la fase de desplazamiento es previa a la de escalado.

4.5.4 Variantes del operador de cruza.

En el apartado 3.3 se dio una justificación teórica de los operadores genéticos. Allí se comentaba que la **cruza** se introduce para proporcionar un mecanismo de intercambio estructurado de información útil (*bloques constructores*) entre individuos. Por su parte, la **mutación** permite explorar nuevas zonas del espacio de búsqueda a las que posiblemente nunca se llegaría usando sólo la **cruza**, y además proporciona un mecanismo de seguridad frente a pérdidas de información valiosa.

Ahora se van a presentar posibles mejoras a la búsqueda genética clásica con el fin de hacer más eficientes los mecanismos de procesamiento de esquemas, desde cuya perspectiva los operadores genéticos clásicos adolecen de varios defectos.

La **cruza** clásica no puede realizar ciertas combinaciones de *bloques constructores*. Supóngase por ejemplo que los esquemas $001^{****}01$ y $****11^{****}$ tienen alta aptitud; es evidente que por medio de la **cruza** clásica resulta imposible obtener el esquema que los combina $001*11^{***}01$.

Existe una asimetría entre la aplicación de la **cruza** y la **mutación**: la unidad básica de **mutación** es el gen, la de la **cruza** es el individuo; en consecuencia, la probabilidad de que un individuo se cruce no depende de su longitud, mientras que la probabilidad de que contenga genes mutados es más alta cuanto más largo sea.

4.5.4.1 Alternativas al método de **cruza** clásico.

Se han propuesto muchas alternativas que mejoran, en circunstancias más o menos generales, las prestaciones de la **cruza** clásica. Las alternativas más importantes son:

- ❖ **Cruza multipunto**: Consiste en cruzar los individuos en torno a dos o más puntos. Esta **cruza** mejora la capacidad de procesamiento de esquemas, pero a costa de perder velocidad de convergencia.
- ❖ **Cruza uniforme**: Para cada bit del primer descendiente se decide, con probabilidad P_{unif} , de qué progenitor heredará su valor en esa posición. El segundo descendiente recibe el correspondiente gen del otro progenitor. Dado que la **cruza** uniforme intercambia más bien

genes que *bloques constructores*, se usa para combinar atributos específicos, con independencia de la posición en que han sido codificados. En algunos problemas esta capacidad compensa el riesgo adicional que se introduce al descomponer *bloques constructores*.

- ❖ **Cruzas anulares:** En muchas ocasiones no hay ninguna razón para considerar que el primer y el último bit de la cadena no son adyacentes, de manera que resulta más natural utilizar una representación anular, esto es, con forma de anillo. Ocurre entonces que la **cruza** clásica introduce un sesgo porque obliga a intercambiar segmentos a partir de una posición fija (la que marcan los extremos). Para evitarlo se deben cruzar los individuos en un número par de posiciones. En esto consiste la **cruza** en anillo.
- ❖ **Cruzas con barajado:** Antes de realizar la **cruza** se desordenan aleatoriamente las posiciones de ambos progenitores a la par. Después se realiza la **cruza** y finalmente se deshace el barajado.
- ❖ **Cruza esquemática:** La idea es realizar una adecuada combinación de *bloques constructores*, para ello se definirá cada cuantas generaciones r se aplicará esta **cruza** especial. Para un valor de *MaxIter* de 50, $r=10$ sería conveniente. Se creará una máscara para cada individuo, se trata de verificar el estado de cada bit, si el alelo no ha cambiado durante r iteraciones entonces al bit de la máscara cuya posición es la misma, se le asignará un 1, en caso contrario se le asignará un 0. Al final de las r generaciones que se designen, la **cruza** se realizará dependiendo del número de 1's que tenga la máscara, es decir, se tomarán todas las posiciones de la máscara que tengan un uno, y el valor que tenga el primer progenitor en esas posiciones pasará directamente al nuevo individuo; enseguida se tomarán las posiciones del segundo progenitor, cuyo valor también es uno, y se pasarán los alelos al nuevo individuo, siempre y cuando esa posición no haya sido heredada por el primer progenitor. Para las

posiciones que obtuvieron un cero en ambos padres, se asignarán los valores del primer padre al primer nuevo individuo, y los valores del segundo padre al segundo descendiente.

Por ejemplo, considérese los siguiente esquemas:

Esquema 1: 0 0 1 * * * * * 0 1

Esquema 2: * * * * 1 1 * * * *

El objetivo es combinarlos. Se supone que estos esquemas se formaron después de r generaciones.

Las máscaras de ambos esquemas son las siguientes:

Esquema 1: 0 0 1 * * * * * 0 1

Máscara 1: 1 1 1 0 0 0 0 0 1 1

Esquema 2: * * * * 1 1 * * * *

Máscara 2: 0 0 0 0 1 1 0 0 0 0

la cruce esquemática hará lo siguiente:

Nuevo individuo: 0 0 1 * 1 1 * * 0 1

4.5.5 Alternativas a la mutación clásica.

La **mutación** es un operador de importancia teórica secundaria, lo que se manifiesta en su baja probabilidad de aplicación en comparación con la **cruza**. Ello no obsta para que en la práctica sea un operador imprescindible. De hecho, se han realizado pruebas y un AG puede llegar a funcionar sin realizar cruza ($P_{\text{cruza}}=0$), pero no sin realizar mutaciones.

Mutaciones no estacionarias: Es un hecho admitido que la importancia de la **mutación** es menor en las fases tardías de la búsqueda que en las tempranas. Esto se explica porque es en las primeras etapas cuando un AG debe realizar una búsqueda lo más amplia posible mientras que en las últimas la búsqueda debería ser más local. Por eso en ocasiones conviene ir reduciendo la tasa de **mutación** P_{mut} según progresa el AG. La manera en que se lleva a cabo la reducción depende en gran medida del problema. La más sencilla consiste en multiplicar la tasa de **mutación** por un factor de reducción constante cada cierto número de generaciones.

Mutaciones no uniformes: Consisten en dar distintas probabilidades de **mutación** a cada gen (o a cada bit dentro de un gen), dependiendo de su significado.

4.6 Parametrización de los AG's.

Para que un algoritmo genético pueda funcionar se deben dar valores apropiados a un conjunto de parámetros de entre los que destacan el tamaño de la población ($TamPob$), la longitud de los individuos (L), las probabilidades de **mutación** y **cruza** (P_{mut} y P_{cruza}), el máximo número

de iteraciones ($MaxIter$), la precisión (TOL), el periodo de actualización de la ventana de desplazamiento (W) y otros específicos de cada método como, por ejemplo, los coeficientes de la función de penalización o de la función de asignación de aptitudes. No existen criterios definitivos para dar valores a esos parámetros, por lo que esta tarea depende en gran medida del buen criterio del diseñador. En principio, los algoritmos genéticos están clasificados como técnicas de búsqueda paramétricamente robustas, es decir, funcionan (mejor o peor, pero funcionan) en un amplio rango de valores de sus parámetros. Sin embargo, las diferencias de eficiencia en la práctica son lo suficientemente espectaculares como para solicitar una mayor atención en dicha elección.

Se ha realizado una gran cantidad de pruebas (Davis 1991) sobre una amplia variedad de problemas de búsqueda y optimización con la intención de determinar rangos de buen funcionamiento para los parámetros del SGA. Aunque los resultados (medidos en términos de prestaciones) son dependientes en mayor o menor medida del problema en particular, se pueden extraer las siguientes conclusiones generales:

- ❖ El tamaño de la población $TamPob$ varía habitualmente entre 50 y 100 individuos; valores menores suelen plantear graves problemas de convergencia prematura y valores mayores requieren un gran esfuerzo computacional sin obtener mejoras apreciables.
- ❖ El tamaño de los individuos L suele venir dado como una consecuencia de los criterios de representación y codificación del problema. El único consejo que cabe dar es el de no alargar innecesariamente dicha cantidad, salvo si se tiene la garantía de que al introducir redundancias se van a obtener prestaciones muy superiores.

- ❖ Las mejores prestaciones se obtienen en general con tasas de cruce que varían entre el 20% y el 60% y tasas de mutación entre el 1% y el 5%, aunque no es raro ver otros valores.
- ❖ El máximo número de iteraciones depende de la precisión especificada y varía mucho de un problema a otro; sirvan como valores orientadores $MaxIter \leftrightarrow 50$ para problemas de evaluación compleja y $MaxIter \leftrightarrow 1000$ para problemas de evaluación sencilla. En cuanto a la tolerancia de error se hace notar que está fuertemente condicionada por la representación elegida.

La robustez paramétrica garantiza el buen funcionamiento del AG clásico con esos valores para una gran variedad de problemas de optimización, pero ¿qué ocurre cuando el problema no está dentro de esa gran variedad? Lo común es que una parametrización como la anterior proporcione mediocres prestaciones.

El caso más habitual de esta situación se presenta cuando se usa un AG con una estructura apreciablemente distinta de la del SGA y con nuevos operadores genéticos.

4.7 Hibridación.

La idea general de las técnicas de hibridación consiste en utilizar el AG para la búsqueda global y encargar la búsqueda local a métodos específicos (esto se expresa gráficamente diciendo que el AG se encarga de buscar los picos de aptitud y el otro método de treparlos). En último término se trata de aprovechar las buenas propiedades de cada método.

La mejor manera de llevar a cabo la hibridación es de forma modular. La conexión entre módulos se puede hacer secuencialmente o en paralelo. En la implantación secuencial se deja que el AG converja suficientemente y entonces se ejecuta el método local sobre los mejores puntos obtenidos. En la implantación en paralelo varios procesadores subordinados realizan la búsqueda local y evalúan los individuos obtenidos y un procesador central conectado con todos ellos hace evolucionar la población proporcionada.

4.8 Algoritmos genéticos paralelos.

Posiblemente la propiedad más importante de los AG's sea el paralelismo implícito. Ello no impide realizar una implantación explícitamente paralela aprovechando las facilidades que da la propia estructura del AG. No existe un modelo típico de implantación en paralelo de un AG, entre otras cosas porque hay una gran variedad de estructuras de procesamiento en paralelo. A grandes rasgos, los modos de implantar un AG en paralelo responden a tres modelos:

Modelos centralizados: En ellos un proceso central coordina el funcionamiento de varios procesos subordinados. La implantación es directa y fácil de realizar. El proceso central se encarga de controlar los operadores de selección y reproducción, mientras que los subordinados únicamente evalúan aptitudes. Estos modelos presentan los dos inconvenientes inherentes al centralismo. Es necesario que los tiempos de evaluación de los procesos subordinados sean análogos (sincronía), pues en caso contrario se producirá gran cantidad de tiempos muertos. Por otra parte, el esquema centralizado presenta poca fiabilidad al depender críticamente del buen funcionamiento del proceso central. El requerimiento de sincronía se puede suavizar eligiendo los operadores

adecuados para que el proceso central pueda trabajar con los individuos según van siendo evaluados.

Modelos distribuidos: En ellos un conjunto de procesadores idénticos realizan las operaciones genéticas y las evaluaciones independientemente los unos de los otros, pero accediendo a una zona común de memoria compartida. No existe mayor requisito de sincronía que el de que no se produzcan accesos simultáneos a la memoria compartida. A efectos externos, estos modelos funcionan de la misma manera que los centralizados pero ahora la fiabilidad se ha incrementado grandemente, pues basta que funcione uno sólo de los procesadores para que el AG pueda realizar procesamiento útil.

Modelos en red: En ellos un conjunto de procesadores independientes ejecutan un AG convencional usando cada uno su propia zona de memoria, operadores genéticos y evaluaciones. Además, se establece una topología de comunicación entre los procesadores que permita el intercambio coordinado de información entre poblaciones con el fin de apoyar la búsqueda de mejores soluciones. Habitualmente dicho intercambio se lleva a cabo mediante técnicas de migración (i.e., movimientos de individuos de una subpoblación a otra) y cruza entre individuos de distintas poblaciones.

Es muy frecuente hibridar los mecanismos de procesamiento paralelo de AG's con otras técnicas heurísticas, con el fin de mejorar no sólo la velocidad sino también la precisión de la búsqueda (ver sección 4.7).

En cualquier caso, resulta muy sencillo y natural implantar un AG en paralelo, por lo que no es difícil pronosticar que estas implantaciones se impondrán en cuanto se generalicen los procesadores en paralelo de bajo costo. Además, los nuevos procesadores masivamente paralelos permitirán la creación de nuevos y más complejos modelos de evolución artificial.

4.9 Algoritmos genéticos diploides.

Se ha comentado repetidamente que el hecho de obligar a que los genotipos respondan a la forma de cadenas binarias responde a la necesidad de obtener la máxima capacidad de procesamiento de esquemas. No obstante, es posible añadir nuevas características al genotipo sin perder por ello capacidad de procesamiento.

Tal es el caso de los AG's con representación duplicada, o usando la terminología biológica, diploides. La idea está tomada de los organismos naturales y consiste en hacer que en cada gen haya lugar para dos alelos, de los cuales sólo uno, llamado dominante, se expresará a la hora de calcular la aptitud de cada individuo. En la reproducción, cada individuo hereda en cada gen un alelo al azar de cada uno de sus progenitores.

Como es habitual, la introducción de este mecanismo no responde tanto a la consecución de una mejor emulación de la naturaleza como a la obtención de unas propiedades concretas tendentes a mejorar la búsqueda genética. La analogía con la naturaleza sólo tiene sentido en la medida en que ilustre los procedimientos actuales o guíe el desarrollo de otros nuevos. En el presente caso, Goldberg (Goldberg, 1989) propone la introducción del diploidismo en los AG's basándose en la siguiente analogía:

Un ejemplo muy conocido de la memoria a largo plazo introducida por los mecanismos de diploidismo y dominancia se encuentra en los desplazamientos del equilibrio de poblaciones de la polilla moteada en Gran Bretaña durante la Revolución Industrial. La variedad silvestre (y originalmente la dominante) de este lepidóptero poseía alas blancas con pequeñas motas negras. Anteriormente a la Revolución

Industrial, esa coloración era un camuflaje eficaz contra los pájaros y otros predadores del hábitat natural de la polilla, los árboles cubiertos de musgo. A mediados del siglo XIX se capturaron variedades negras en los alrededores de ciudades industriales. Cuidadosos experimentos llevados a cabo por Kettlewell mostraron que la variedad moteada tenía ventaja en el entorno primitivo, mientras que la variedad negra tenía ventaja en el entorno industrial en el cual la contaminación había exterminado el musgo que cubría los troncos de los árboles. Sucedió que el grado de oscuridad está controlado por un único gen dominante, lo que indicaba que se había producido un desplazamiento de la dominancia. Cuando el equilibrio se desplazó hacia la variedad oscura, ésta se convirtió en dominante y la moteada en recesiva. Nótese que la variedad oscura no era una nueva invención; no era un caso de mutación fortuita que milagrosamente proporcionó la variedad correcta. En lugar de eso, la variedad negra había sido inventada antes, quizá como respuesta a bosques en los que el musgo había desaparecido por causas naturales. Cuando los residuos de la industria originaron la desaparición del musgo, la variedad oscura se reprodujo con mayor facilidad y evolucionó hasta convertirse en la variedad dominante. Con esta solución alternativa en la reserva, la polilla moteada pudo fácilmente adaptarse con rapidez a las presiones selectivas de su entorno cambiante.

En definitiva, la duplicación de los alelos introduce un mecanismo explícito con el que se evita la destrucción prematura de información que pudiera resultar valiosa en el futuro. Eso es especialmente interesante cuando se trabaja con funciones de aptitud cambiantes en el tiempo.

Fácilmente se observa que el punto crítico de los AG's diploides radica en la elección de un criterio apropiado de dominancia, esto es, dados dos alelos de cierto gen ¿cuál de ellos se expresa?

La propuesta es la siguiente:

Considérese el diploidismo como un nuevo parámetro. A fin de no afectar la velocidad de convergencia, se propone aplicar esta propiedad a un solo gen de cada cromosoma, de manera que se manejen dos poblaciones en forma paralela (P y P'), este parámetro tendrá una probabilidad media en el intervalo $[0.4 - 0.6]$, y se sugiere sea aplicado en las primeras etapas del algoritmo. La idea no es la diversidad, sino buscar un aumento considerable en la aptitud promedio. Cuando se aplique dicho operador, en cualquier población P o P' puede haber cromosomas con alelos dominantes y recesivos. Se aplicará la *función de aptitud* a cada A_j y a cada A'_j , el que resulte más apto será el dominante y pasará a formar parte de una población intermedia, digamos P'' . Supóngase que se tienen los siguientes cromosomas:

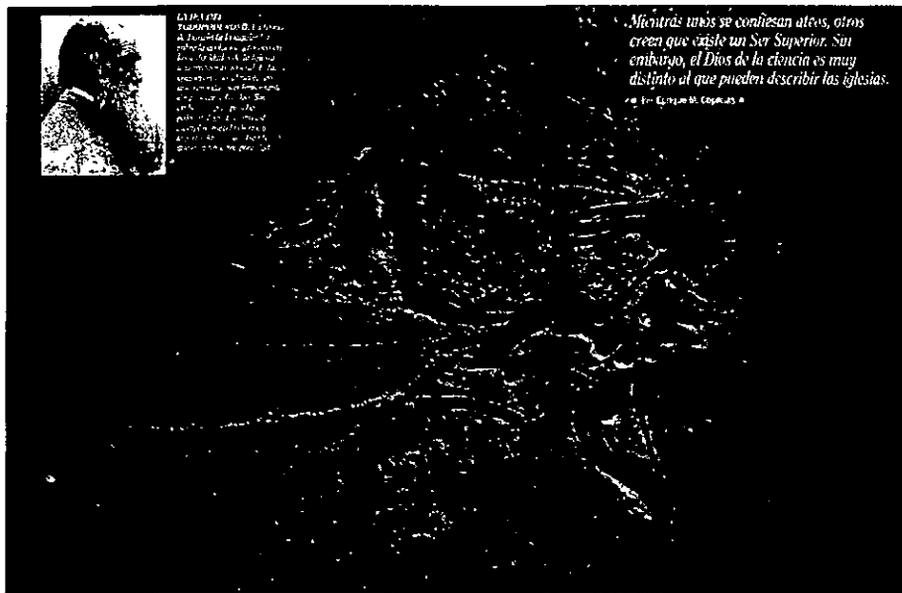
$$A_1 = 101010$$

$$A'_1 = 101000$$

y que la *función de aptitud* señala que A'_1 es mejor, entonces el alelo $a'_5 = 1$ será dominante, y por consecuencia A'_1 pasará a formar parte de la población intermedia.

Al aplicar este nuevo operador se tendrá una mejora considerable a nivel general, pero puede retardar la convergencia, y en las últimas generaciones sólo retendría el proceso.

Es importante recalcar que el diploidismo es muy diferente a una **mutación poblacional**, pues el primero permite mantener el valor original, si es que resulta más favorable, situación que de ninguna manera contempla la **mutación**.



LA IRA DEL TODO PODEROSO. La teoría de Darwin sobre la evolución provocó en las autoridades de la Iglesia una irritación general. Éstas se opusieron frontalmente, en una reproducción lamentable del proceso a Galileo. Sin embargo, ante pruebas paleontológicas como este ejemplar, mezcla de ave y lagarto, las iglesias han tenido que aceptar estos postulados.

Revista Muy Interesante especial "El mundo de las Religiones", 1997

Capítulo

5

Placeres existenciales de los AG's.

Objetivos Particulares:

- Analizar a los AG's más allá de simples solucionadores de problemas.
- Mostrar las cualidades de los AG's que los hacen ser distintos a cualquier otro método.
- Presentar diversas facetas de los AG's que permiten usarlos como modelo para todo.

Para que surjan individuos muy dotados es preciso que su infancia haya discurrido sin apenas presiones y lejos de todo conformismo, que el muchacho o muchacha haya podido cultivar y acrecentar sus intereses por peculiares y extravagantes que pudieran parecer.

Bertrand Russell

...es posible que en un futuro no muy lejano podamos dar vida a un ser humano a partir de una simple célula tomada de prácticamente de cualquier parte del cuerpo donante. Si ello es así, cualquier célula del organismo debidamente preservada hasta el momento de en que la gestación extracorpórea se lleva a la práctica con garantías puede llegar a convertirse en un ser vivo. Por lo demás, ¿como un genocidio si me pincho un dedo y vierto una gota de sangre?

Carl Sagan, 1977

Para analizar a los AG's es necesario verlos más allá de simples solucionadores de problemas, por lo tanto se analizan desde distintos enfoques. El objetivo de este capítulo es dar una idea general de ciertas cualidades y características de los AG's, por lo tanto no se profundizará a detalle el análisis de dichos enfoques.

En el libro "The existential pleasures of engineering (1976)" Samuel Flomman, discute dos aspectos muy poco analizados acerca de la ingeniería: la ingeniería es una profesión multifacética, y la ingeniería es divertida. De la misma forma, el autor de este documento se ha permitido aplicar este análisis a las matemáticas aplicadas, es decir, las matemáticas aplicadas son multifacéticas y divertidas.

Dado que sería imposible tratar en forma general a las matemáticas aplicadas, y más aún si se considera a la computación como herramienta, tal como lo hizo Goldberg¹ en su ensayo², se ha decidido analizar estas características enfocadas a los algoritmos genéticos. Trabajar con AG's es una experiencia enriquecedora. A diferencia de muchos métodos tradicionales de resolución de problemas, es fácil encontrar aspectos interesantes que los hacen multifacéticos y divertidos.

Para los objetivos que se pretende alcanzar, será muy importante destacar la relación entre hacer algo divertido y algo útil. Para los matemáticos aplicados, el simple hecho de crear un modelo y aplicarlo es motivo de orgullo; sin ir lejos, el resolver una ecuación puede ser muy satisfactorio. En el caso de los AG's, el diseñador tiene que crear un modelo y hacer adaptaciones; son tan flexibles que permiten hacer cambios en su estructura, y con ellos se puede resolver una gran diversidad de problemas³.

Normalmente estamos acostumbrados a realizar actividades que nos divierten sin pensar en la utilidad que pudiera tener, pero por qué no pensar mejor en hacer cosas útiles que sean divertidas; ¿y qué podría ser útil? La respuesta es sencilla, el aprendizaje, el trabajo, el estudio, etc., lo importante es ser útil o hacer cosas útiles y divertirse al mismo tiempo, pero hay de diversiones a diversiones, y es necesario destacar la diferencia porque, como se mencionó anteriormente, para algunos es muy divertido y satisfactorio resolver problemas matemáticos, en cambio para la mayoría es todo lo contrario.

¹ David E. Goldberg. Es Doctor en Ingeniería Civil de la Universidad de Michigan en Ann Arbor, EUA. Actualmente es Profesor del Departamento de Ingeniería de la Universidad de Illinois en Urbana Champaign y es director del Laboratorio de algoritmos genéticos de Illinois.

² The existential pleasures of Genetic Algorithms (1995).

³ Se han ajustado los AG's para resolver problemas del tipo de "la mochila" y del "agente viajero", entre muchos otros.

Como se planteó desde el principio de este capítulo se analizará a los algoritmos genéticos en sus múltiples facetas y les hallaremos su aspecto divertido. Existen diversas formas de ver a los algoritmos genéticos. Desde solucionadores de problemas, hasta acertijos técnicos, o hasta como una forma para entender más acerca de la vida misma. Los AG's son mucho más que los problemas que ellos nos ayudan a resolver, mucho más que los individuos involucrados en la solución y mucho más importantes de lo que la mayoría de los algoritmistas genéticos reconocen. Se analizarán desde las siguientes perspectivas:

- ❖ AG's como solucionadores de problemas.
- ❖ AG's como una metodología efectiva de invención.
- ❖ AG's como las bases para sistemas de aprendizaje artificial.
- ❖ AG's como una filosofía guía.
- ❖ AG's como un modelo para todo.

En lo que resta del capítulo examinaremos cada uno de dichos puntos de vista.

5.1 AG's Como solucionadores de problemas.

Viendo a los AG's desde lo más simple, han resultado exitosos. Con ellos se ha solucionado una gran variedad de problemas, en su mayoría problemas que han presentado una gran dificultad para hallar la solución. Enumerar la lista de los distintos problemas a los que han dado solución sería imposible pero, para dar una idea de las múltiples

facetas de los AG's, diremos que hasta han sido utilizados para crear composiciones musicales complejas⁴.

Su intrínseca relación con las computadoras los han hecho ser cada vez más fuertes, es más, a partir de ellos se ha justificado más el procesamiento en paralelo.

Para el modelador o diseñador de AG's, en primer instancia, lo importante es la utilidad que generan, es decir ¿qué tan bien trabajan? ¿son confiables, exactos y veloces? Poco a poco se han ido encontrado respuestas a estas interrogantes, a pesar de que todavía sus fundamentos teóricos no son del todo aceptados. Posiblemente el querer demostrar el porqué funcionan sea la razón por la cual, cada vez más, investigadores y académicos se interesan y quieren aportar.

Dado que los primeros cuatro capítulos de este trabajo han tratado de mostrar el lado más fuerte de los AG's, es decir, su faceta como solucionador de problemas, este último apartado se dedicará a las otras características que hacen de los algoritmos genéticos algo diferente.

5.2 AG's Como una metodología efectiva de invención.

Podríamos considerar a los AG's como una metodología efectiva de invención. Goldberg en uno de sus libros⁵ delineó una conexión entre la metodología de invención que guió a los hermanos Wright al éxito en los cielos a bordo del Kitty Hawk, y la metodología que ha guiado al vuelo exitoso del AG.

Recapitulando brevemente dicho argumento:

⁴ Ver: "Genetic algorithms and computer-assisted music composition" de Hornen y Goldberg (1994).

⁵ "Making genetics algorithms fly: A lesson from the Wright Brothers" (1993).

“Los hermanos Wright triunfaron donde muchos otros fallaron porque descompusieron el difícil problema del vuelo, sustentado por máquinas, en subproblemas casi separados; resolvieron los subproblemas utilizando estudio empírico, modelado por facetas y análisis dimensional e integraron las subsoluciones dentro de la solución del problema completo, considerando mucho los resultados y poco la elegancia, pruebas u otras linduras matemáticas”.

Un adecuado diseño de un algoritmo genético se puede descomponer de la siguiente forma:

1. Saber lo que se está procesando: bloques constructores (en inglés building blocks-BB's).
2. Esperar un adecuado suministro inicial de BB's.
3. Fomentar el crecimiento de BB's necesarios.
4. Asegurar que las decisiones que conciernen a los BB's se hacen bien.
5. Resolver un problema de dificultad acotada en lo concerniente a los BB's.
6. Asegurar que los BB's puedan mezclarse (intercambiarse) apropiadamente.

Coello⁶ hace la siguiente diferenciación entre esquemas y BB's:

“En los AG's los esquemas son subconjuntos (cadenas de caracteres) que tienen una o más características en común y los BB's son esquemas que son consistentemente enfatizados por la selección y respetados e intercambiados por los operadores genéticos”.

⁶“Introducción a los algoritmos genéticos” (1995).

Para más detalles de esta conexión y de cómo ha permitido ir más allá de las teorías de Holland, se debe consultar a (Goldberg 1992a, 1992b, Ipsen 1960 y a Bradshaw 1991).

Es precisamente esta distribución, la que ha permitido progresar hasta el punto donde ahora es posible demostrar analíticamente las limitaciones de los AG's simples por medio de la construcción de *mapas* de control, y donde los mismos métodos hacen posible la creación de algoritmos genéticos desordenados (*messy*) rápidos⁷ que resuelven problemas difíciles en poco tiempo.

5.3 AG's Como las bases para sistemas de aprendizaje artificial.

Al igual que Holland⁸, muchos soñaron que las máquinas aprendieran por sí solas; los AG's se han convertido en la esperanza para lograrlo; ya hay avances, y se espera lograrlo en un futuro cercano. Al parecer, partidarios de la Inteligencia Artificial están cambiando el fondo de la investigación, y todo se debe a los buenos resultados que han dado los AG's.

La implantación más común de sistemas de aprendizaje artificial basados en algoritmos genéticos se hace a través de sistemas clasificadores. Un sistema clasificador es un sistema de aprendizaje inductivo de reglas de decisión sobre espacios basados en atributos con el que se trata de producir respuestas adecuadas a un conjunto de estímulos, a través de un proceso de clasificación de conceptos que van siendo adquiridos empíricamente.

⁷ Para más información consultar: "SEARCH, polynomial complexity, and the fast messy genetic algorithm" de Kargupta (1993).

⁸ Holland plasmó estas ideas en "An Outline for a Logical Theory of Adaptive Systems" (1962).

Para ilustrar el concepto de sistemas clasificadores se ha utilizado una metáfora biológica que muestra su funcionamiento elemental, dicha metáfora es conocida como "Sistema Clasificador Kermit". Kermit es una rana que vive en un entorno perfectamente delimitado (un charco) en el que recibe ciertos estímulos y efectúa ciertas respuestas. Los estímulos son recibidos fundamentalmente a través de sus ojos, los cuales son tan simples que sólo le permiten percibir cinco atributos de los objetos del entorno, estos son: si el objeto es móvil o inmóvil, si vuela o está en el suelo, si es grande o pequeño, si está cerca o lejos y si tiene colores fuertes o débiles. El conjunto de respuestas que Kermit puede efectuar es también bastante limitado, es decir, puede huir, puede ocultarse o puede intentar comerse el objeto en cuestión. De lo que se trata es de proporcionar a Kermit un conjunto mínimo de reglas de decisión que le permitan sobrevivir con éxito en su entorno.

Esas reglas a seguir pueden ser como las siguientes:

- ❖ Si el objeto es grande y se mueve, huir.
- ❖ Si el objeto es pequeño, se mueve, está en el aire y cerca, comérselo.

Evidentemente el conjunto de reglas debe ser lo más reducido posible porque una rápida respuesta puede salvarle la vida, además ese conjunto de reglas ha de ser lo suficiente general como para considerar todas las situaciones importantes que plantee la vida en un charco, y a la vez no debe ser ambiguo. Nótese que Kermit no tiene por qué saber lo que es un mosquito o cualquier otro animal, simplemente tiene que tener muy claro el concepto de peligrosidad, saber que es comestible y saber clasificar los estímulos que le llegan en función de esos conceptos. Las ranas reales han adquirido ese conjunto de reglas con el cual hacer la clasificación a través de un largo proceso de evolución natural. Parece pues, bastante razonable utilizar la evolución artificial para emular el proceso de aprendizaje.

Con esta ilustración de sistemas clasificadores se pretende dar una idea general de como los AG's pueden ofrecer bases para el aprendizaje artificial.

5.4 AG's como una filosofía guía.

Después de involucrarse con los AG's durante un cierto tiempo, uno empieza a relacionarlos con todo. El siguiente paso es que surja un creciente interés por la literatura filosófica de la evolución.

En el libro "Flow: the psychology of optimal experience" de Csikszentmihalyi (1990), se argumenta un conjunto de criterios y un candidato para un nuevo sistema de creencia:

"Parece claro que una mayoría creciente no está siendo ayudada por religiones y sistemas de creencia tradicionales. Muchos son incapaces de separar la verdad por las distorsiones y degradaciones que el tiempo ha agregado, y ya que ellos no aceptan el error, rechazan la verdad también. Otros están tan desesperados por algún orden que se adhieren rígidamente a cualquier creencia que se encuentra a la mano -con todo y sus aspectos negativos- y se convierten en cristianos o musulmanes o comunistas fundamentalistas".

Los criterios para un nuevo sistema de creencia están cambiando:

"Si una nueva fe va a capturar nuestra imaginación, debe ser una que dé cuenta racional de las cosas que conocemos, las cosas que sentimos, las cosas que deseamos y aquellas que tememos. Debe ser un sistema de creencias que encaucen nuestra energía

psíquica hacia metas significativas, un sistema que proporcione reglas para una manera de vida que pueda proporcionar el flujo”.

Esto suena bastante difícil de alcanzar. Csikszentmihalyi dice que una alternativa es darnos vuelta hacia la evolución:

“Una manera de realizar esto es a través del concepto de evolución. Todo lo que nos importa más a nosotros. Preguntas como: ¿de dónde venimos?, ¿hacia dónde vamos?, ¿qué poderes moldean nuestras vidas?, ¿qué es bueno y malo?, ¿cómo nos relacionamos entre nosotros y con el resto del universo?, ¿cuáles son las consecuencias de nuestras acciones? pueden discutirse de manera sistemática en términos de lo que ahora conocemos acerca de la evolución y aún más en términos de lo que vamos a conocer de ella en el futuro”.

Es de llamar la atención este argumento de reemplazar la religión con una fe evolucionista. Esta fe basada en mecanismos puede traer consecuencias graves, al ponderar aspectos individualistas, de manera que exista una lucha por ser el mejor adaptado; sin embargo muchos principios éticos y morales enseñados por las religiones del mundo terminan teniendo sus bases en principios evolucionistas, y su propósito es promover la cohesión social y la supervivencia de grupo, por otro lado las religiones fomentan el cumplimiento de preceptos morales a través de promesas no verificables de vida en el más allá y castigos por malas acciones.

Es importante destacar el pensamiento evolucionista y en este sentido los algoritmistas genéticos se encuentran en una posición especial debido a su experiencia concreta con la simulación de estos sistemas. Tal vez la idea de una fe evolucionista es extrema, no obstante se espera que el pensamiento evolucionista se profundice a fin de acercarse a una mejor comprensión de la esencia de nuestra naturaleza espiritual.

Aunque los aspectos filosóficos de la evolución parecen ser un tema muy alejado de los propósitos, fue necesario destacarlos como parte de las cualidades de los AG's que los hace precisamente innovativos y diferentes a cualquier otro algoritmo matemático.

5.5 AG's como un modelo para todo.

Hasta ahora se han presentado sólo algunas de las múltiples facetas de los algoritmos genéticos. Por mencionar, de manera muy general, un par más de características interesantes, se tratará de relacionarlos con factores de creatividad y de innovación humana, y se harán analogías con otros sistemas.

❖ Modelo de innovación y creatividad.

Varios autores han comparado el poder de los AG's con el poder de la creatividad e innovación humana. ¿Qué es lo que hace que una persona sea innovativa y creativa? Lo que se hace, frecuentemente, es formar conjuntos de características de contextos diferentes, las unimos, les damos forma y obtenemos un resultado.

Resulta un tanto difícil aceptar que un método matemático tenga, de manera inherente, características que más bien están ligadas a las artes. Cuando hablamos de creatividad, inmediatamente la asociamos a la música, a la pintura o a la literatura; sin duda es necesario ser creativo para aportar elementos a estas artes, pero esta creatividad normalmente se lleva a cabo mediante interrelaciones con otros aspectos de la vida. Por ejemplo, una obra pictórica plasma una serie de objetos que tiene un significado en conjunto, o una obra musical nos puede

hacer viajar a un ambiente distinto, y esto se logra al conjuntar características, que nunca nadie había hecho y he ahí el carácter innovativo.

Esta yuxtaposición de creatividad humana es similar a la selección y recombinación de los algoritmos genéticos; específicamente se refiere a la interrelación y conjunción de bloques constructores (BB's). No se profundizará más en esta característica, sin embargo, llama la atención la analogía de los procesos fundamentales de estas dos situaciones.

❖ **Analogía con otros sistemas.**

Basta conocer poco tiempo a los AG's para encontrarles aspectos interesantes. Esa analogía con lo procesos naturales de la evolución incita a buscar razones o simplemente encontrarle nuevos atributos. Si pensamos que simulamos poblaciones, podríamos establecer una enorme serie de analogías: sistemas sociales, sistemas políticos, religiones, organizaciones, etc.

No basta leer este trabajo para darse una idea clara, no tan sólo del potencial de los AG's, sino de las cualidades que los hacen tan atractivos. Es necesario programarlos, diseñarlos y correrlos muchas veces para hacer placentero nuestro trabajo con ellos o simplemente para establecer relaciones con los sistemas antes mencionados.

CONCLUSIONES

Hay miles de millones de galaxias en los cielos, probablemente la mitad de ellas tengan planetas a distancias adecuadas biológicamente del sol local... debe haber otros habitantes en el universo. Por supuesto, los detalles de la evolución no serían los mismos. Incluso si la tierra se iniciase otra vez y solamente las fuerzas operasen al azar no se produciría nada parecido a un ser humano, porque los seres humanos son el producto final de una senda evolucionista exquisitamente complicada, llena de falsos comienzos, de callejones casi sin salida y de accidentes estadísticos... debe haber, creo, muchos lugares en la Galaxia donde haya seres mucho más avanzados que nosotros en ciencia y tecnología, en política, ética, poesía y música.

Carl Sagan, 1973

Al llegar a la finalización de este trabajo, además de los nuevos conocimientos y las satisfacciones que se obtuvieron, se concluye con las siguientes reflexiones:

Fue evidente que el uso de los algoritmos genéticos tiene ciertas limitaciones, aunque dichas limitaciones no son mayores que las planteadas por otros métodos. Es más, al mitigar sus efectos se obtuvieron mejoras en su desempeño.

Existen muchos métodos de optimización que han sido desarrollados en matemáticas e investigación de operaciones, pero en el caso particular de los AG's, el rol que juegan como herramienta de optimización es el de ser métodos de búsqueda global, flexibles y rápidos.

El análisis pormenorizado de los métodos y procedimientos de los AG's ha permitido concluir que, si existe un buen método de optimización especializado para un problema específico, el AG puede no ser la mejor herramienta de optimización para la aplicación. Sin embargo, la utilización de algoritmos híbridos, que combinan otros métodos con los AG's, puede ser la mejor opción.

La mayoría de la gente que se acerca a los algoritmos genéticos lo hace para resolver problemas, pero para algunos el encuentro deja mucho más. Aunque se ha considerado cómo los AG's han resuelto problemas, cómo lo hacen y cómo se han vuelto cada vez más competentes, posiblemente lo más importante e interesante esté aún más adelante, al entender mejor lo que los algoritmos genéticos están tratando de decirnos acerca de la evolución, la innovación, y la creatividad. Más aún, entender la evolución de una manera detallada proporciona intuición en muchos caminos de la vida, haciendo, por lo tanto, el estudio de estas *simulaciones de procesos evolutivos* un hecho de mayor importancia.

GLOSARIO

- **ADN:** Acido Desoxirribonucleico. Una macromolécula doblemente trenzada que tiene una estructura helicoidal. Las 4 bases de nucleótido Adenina (A), Timina (T), Citosina (C) y Guanina (G) son el alfabeto de la información genética. Las secuencias de estas bases en la molécula de ADN determina el plan constructor de cualquier organismo.
- **Alelo:** Cada gen es capaz de ocupar sólo una región en particular de un cromosoma (su "lugar"). En cada determinado lugar pueden existir, en la población, formas alternativas del gen. A estas alternativas se les llama alelos. Por ejemplo, si se usa representación binaria, este valor puede ser 0 ó 1.
- **Aptitud:** Se refiere a la capacidad de adaptación. Frecuentemente se mide como el éxito reproductivo relativo. Los individuos aptos son los que se adaptan a su ambiente y cuyos descendientes se adaptarán también a los ambientes futuros. Un valor asignado a un individuo, que refleja qué tan bueno es éste para resolver una cierta tarea. Una "función de aptitud" se usa para mapear un cromosoma a un valor de aptitud.
- **Bloque Constructor:** Un grupo pequeño y compacto de genes que han co-evolucionado de tal manera que su introducción en cualquier cromosoma tiene una alta probabilidad de incrementar la aptitud de dicho cromosoma.
- **Cromosoma** Cada una de las cadenas de ADN que se encuentra en las células. Los cromosomas contienen genes que se encuentran codificados como una sub-sección de la cadena de ADN. Los cromosomas usualmente están presentes en todas las células de un organismo, aunque sólo una minoría de ellos estarán activos en una célula cualquiera. En AG's son estructuras de datos que contiene una 'cadena' de parámetros de diseño o genes. Esta cadena puede almacenarse, por ejemplo, como una cadena de bits o un arreglo de enteros.
- **Cruza (Reproducción Sexual):** Un proceso complicado que normalmente se efectúa de la manera siguiente: los cromosomas, mientras se encuentran produciendo gametos, intercambian porciones de material genético. Consecuentemente, durante la reproducción sexual los gametos masculinos y femeninos (i.e., el esperma y el ovario) se fusionan para producir una nueva célula diploide con un par de cromosomas. En AG's es un operador de reproducción que forma un nuevo cromosoma combinando partes de cada uno de sus cromosomas padres.
- **Darwinismo:** Teoría de la evolución, propuesta por Charles Darwin, y que dice que la evolución se efectúa a través de variaciones aleatorias de características hereditarias, en conjunción con un proceso de selección natural (supervivencia del más apto).
- **Decepción (Deception):** La condición donde la combinación de buenos bloques constructores nos lleva a una reducción en la aptitud, en vez de un incremento. Este fenómeno fue propuesto originalmente por Goldberg para explicar el porqué el algoritmo genético no funciona satisfactoriamente en algunas tareas.
- **Descendiente:** Un individuo generado por cualquier proceso de reproducción.
- **Diploide:** Se refiere a una célula que contiene 2 copias de cada cromosoma. Las copias son homólogas (i.e., contienen los mismos genes en la misma secuencia).
- **Epistasis:** Un efecto de "enmascarar" entre los genes. Se dice que un gen es epistático cuando su presencia suprime el efecto de un gen que se encuentra en otra posición. Los genes epistáticos son llamados algunas veces genes de inhibición, por el efecto que producen sobre otros genes.
- **Especie:** No hay una definición aceptada universalmente de "especie", pero podemos definirla como una colección de criaturas vivientes que tienen características similares, y que se pueden reproducir entre ellas para producir descendientes que serán similares a sus progenitores. Los miembros de una especie ocupan el mismo nicho ecológico.

- **Elitismo:** Es un mecanismo que se usa en algunos algoritmos evolutivos para asegurar que los cromosomas de los miembros más aptos de la población se pasen a la siguiente generación sin ser alterados por ningún operador genético.
- **Esquema:** Un patrón de valores de genes en un cromosoma que puede incluir estados 'no importa' (don't cares). Por tanto, en un cromosoma binario, cada esquema puede especificarse mediante una cadena de la misma longitud del cromosoma, en la que cada carácter es un elemento del conjunto {0, 1, *}. Un cromosoma en particular se dice que 'contiene' un esquema determinado si corresponde al esquema en cuestión. Por ejemplo, el cromosoma 01101 corresponde al esquema *1*0*. El orden de un esquema es el número de posiciones que no sean 'no importa', mientras que la longitud de definición es la distancia entre las 2 posiciones más alejadas de 'no importa's. Por lo tanto, *1*0* tiene un orden 2 y una longitud de definición de 3.
- **Explotación:** Cuando se atraviesa un espacio de búsqueda, la explotación es el proceso de usar la información obtenida de los puntos visitados previamente en el espacio de búsqueda para determinar qué lugares resulta más conveniente visitar a continuación. Las técnicas de explotación son buenas para encontrar máximos locales.
- **Exploración:** El proceso de visitar completamente nuevas regiones de un espacio de búsqueda, para ver si puede encontrarse algo prometedor. A diferencia de la explotación, la exploración involucra grandes saltos hacia lo desconocido. Los problemas que tienen máximos locales pueden resolverse algunas veces sólo usando este tipo de búsqueda aleatoria.
- **Fenotipo:** Los rasgos específicos de un individuo.
- **Generación:** Una iteración de la medida de aptitud y la creación de una nueva población por medio de operadores de reproducción.
- **Genoma:** La colección completa de genes (y por tanto cromosomas) que posee un organismo.
- **Genotipo:** La composición genética de un organismo (la información contenida en el genoma).
- **Gene:** Una subsección de un cromosoma que (usualmente) codifica el valor de un solo parámetro.
- **Individuo:** Un solo miembro de la población. Cada individuo contiene un cromosoma que representa una solución posible a la tarea en cuestión (i.e., es un solo punto en el espacio de búsqueda). Usualmente se almacena también otro tipo de información en cada individuo (su aptitud).
- **Inversión:** Un operador de reordenamiento que trabaja mediante la selección de puntos de corte en un cromosoma, invirtiendo el orden de todos los genes comprendidos entre ellos.
- **Lamarckismo:** Teoría de la evolución que precedió a la de Darwin. Lamarck creía que la evolución se había llevado a cabo a través de la herencia de ciertas características adquiridas. Esto es, las habilidades o características físicas que un individuo adquiere durante su vida pueden pasarse a sus descendientes. Aunque la herencia Lamarckiana no se lleva a cabo en la Naturaleza, esta idea ha sido aplicada exitosamente en algunas áreas de la computación evolutiva.
- **Mutación:** Un operador de reproducción que forma un nuevo cromosoma a través de alteraciones (usualmente pequeñas) de los valores de los genes de un solo cromosoma padre. La mutación natural se produce normalmente por errores de copiado del ADN.
- **Operador de Reproducción:** Un mecanismo que influencia la forma en que se pasa la información genética de padres a hijos durante la reproducción. Los operadores de reproducción caen en 3 amplias categorías: cruza, mutación y reordenamiento.
- **Operador Genético:** Un operador de búsqueda que actúa sobre una estructura de codificación que es análoga al genotipo de un organismo (e.g., un cromosoma).
- **Población:** Un grupo de individuos que pueden interactuar juntos, por ejemplo, para reproducirse.
- **Progenitor (Padre):** Un individuo que toma parte en la reproducción para generar uno o más individuos a los que se les conoce como descendientes o hijos.
- **Reproducción:** La creación de un nuevo individuo a partir de 2 progenitores (reproducción sexual). La reproducción asexual es la creación de un nuevo individuo a partir de un solo padre.
- **Selección:** Proceso mediante el cual algunos individuos en una población son seleccionados para reproducirse, típicamente en base a su aptitud (los de mayor aptitud son seleccionados).

BIBLIOGRAFÍA

- [1] David E. Goldberg. Genetic algorithms in search, optimization & machine learning. Addison-Wesley Co., Inc, Reading, 1989.
- [2] David E. Goldberg. Sizing populations for serial and parallel genetic algorithms. Universidad de Alabama, 1987.
- [3] David E. Goldberg. First flights at genetic-algorithm Kitty Hawk. IlliGAL Report No. 94008, october 1994.
- [4] David E. Goldberg. Simple genetics algorithms and the minimal, deceptive problem, Londres 1987.
- [5] Zbignew Michalewicz. Genetic algorithms + Data structures = Evolution programs. Springer-Verlag, 1994.
- [6] Lawrence Davis. Handbook of genetic algorithms. Van Nostrand Reinhold, New York, 1991.
- [7] J. Heitkotter & D. Beasley. The Hitch-Hiker's guide to evolutionary computation, enero 1996.
- [8] J. David Schaffer (ed.) Proceedings of the third international conference on genetic algorithms. Morgan Kaufmann Publishers, 1989.
- [9] John H. Holland. Algoritmos genéticos. Investigación y ciencia, septiembre 1992.
- [10] Rick L. Riolo. Los bits mejor dotados. Investigación y Ciencia. septiembre 1992.
- [11] J. H. Holland, Adaptacion in natural and artificial systems, Universidad de Michigan Press, Ann Arbor, MI 1975.
- [12] Georges R. Harik. Learning gene linkage to efficitly solve problems of bounded difficulty using genetic algorithms. IlliGAL Report No.97005, junio 1997.

- [13] Coello, Hernández y Buckless. Estrategias evolutivas: La versión alemana del algoritmo genético, 1998.
- [14] Carlos A. Coello Coello. Algoritmos genéticos y sus aplicaciones, Soluciones Avanzadas, 1996
- [15] Salvatore R. Mangano. A genetic algorithm white paper. Man Machine Interfaces, 1996.
- [16] David B. Fogel. An introduction to simulated evolutionary optimization. IEEE transactions on neural networks vol 5 No. 1 enero 1994.
- [17] Charles Darwin. On the origin of species by means of natural selection or preservation of favored races in the struggle for life. 1859.
- [18] A. Oparin El origen de la vida. Editorial Época 1983.
- [19] Duane T. Gish. Especulaciones y experimentos relaciondos con teorías sobre el origen de la vida (crítica). Libros creación y ciencia, 1971.
- [20] William D. stansfield. Teoría y problemas de genética. McGraw-Hill. Serie Shaum, 1984.
- [21] Lance Chambers. Practical Handbook of Genétic Algorithms:New Frontiers. Lewis Publisher, 1995
- [22] Anselmo Pérez. Una introducción a la Computación Evolutiva. Internet. 1996.
- [23] Carl Sagan. Los dragones del edén. Especulaciones sobre la evolución de la inteligencia humana. Grijalbo, 1977.
- [24] Carl Sagan. La conexión cósmica. Ediciones Orbis, 1973.