

01190

ALGORITMOS GENÉTICOS INTELIGENTES Y SU
APLICACIÓN A LA INVESTIGACIÓN DE
OPERACIONES

TESIS DOCTORAL

QUE PARA OBTENER EL GRADO DE
DOCTOR EN INGENIERÍA

P R E S E N T A
JAIME MORA VARGAS

DIRECTOR DE TESIS:
DR. CHRISTOPHER R. STEPHENS
CO-DIRECTOR DE TESIS:
DR. HENRI WAELEBROECK G

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
MÉXICO, D.F.
OCTUBRE 2000

204381



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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 esta 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.

Contenido

Contenido	iv
Indice de Figuras	v
Resumen	xi
Agradecimientos	xii
Aporte de la tesis	1
Introducción	4
1 Evolución	11
1.1 Evolución, competencia y adaptación	11
1.2 Requerimientos para la existencia de evolución	14
1.3 Mecanismos biológicos	15
1.4 Aptitud y paisajes de aptitud	17
2 Computación evolutiva	21
2.1 Algoritmos evolutivos	22
2.1.1 Algoritmos genéticos	28
2.1.2 Estrategias evolutivas	37
2.1.3 Programación evolutiva	39
2.2 Otras variantes de algoritmos evolutivos	40
2.2.1 Programación genética	40
2.2.2 Sistemas clasificadores	41
3 Teorías de algoritmos genéticos	43
3.1 Teorema de esquemas, hipótesis de bloques constructores	43
3.2 Análisis por cadenas de Markov	46

3.3	Enfoque de estrategias evolutivas	49
3.4	Grados de libertad efectivos	50
3.4.1	Paisajes de aptitud	51
3.4.2	Ecuaciones de evolución para cuerdas	54
3.4.3	Aptitud efectiva	56
3.4.4	Aportes	61
4	Algoritmos genéticos inteligentes	71
4.1	Mapeo genotipo-fenotipo	71
4.2	Simetría de sinónimos	75
4.3	Rompimiento de simetría inducido	76
4.4	Ejemplos analíticos	80
5	Experimentos y resultados	89
5.1	Dependencia de tamaño para bloques constructores	90
5.2	Autocodificación de tasas de mutación y cruza	93
5.3	Modelo del virus	106
5.4	Modelo de girafas	114
5.5	Implementación para Agente viajero	118
5.6	Manejo de portafolios de inversión	123
5.6.1	Sistema para manejo de portafolios.	126
5.6.2	Desarrollo del sistema.	127
6	Conclusiones	154
	Bibliografía	159

Índice de Figuras

4.1	Gráfica de la aptitud efectiva como función del tiempo para el modelo de un bit para diferentes tasas de mutación.	82
4.2	Gráfica de aptitud efectiva para en $t = 0$ para el modelo de dos bits. .	84
4.3	Gráfica de aptitud efectiva en los ultimos tiempos para el modelo de dos bits.	85
4.4	Gráfica de aptitud efectiva vs p ($\times 100$) en el límite del estado estable para el paisaje NIAH.	88
5.1	Gráfica de $M(t)$ vs t en modelo unitario con $p_c = 1$	92
5.2	Gráfica de $F(t)$ vs tiempo para el modelo unitario de la Figura 5.1. .	93
5.3	Gráfica de un paisaje multimodal vs genotipo. Para una cuerda binaria de 6 bits se tienen 63 genotipos posibles. En el eje y se presenta la aptitud de cada genotipo.	95
5.4	Gráfica de aptitud vs genotipo para un paisaje “deceptivo” (el entrecruzamiento de los óptimos 0 y 63 producen individuos poco aptos). .	96
5.5	Gráfica de aptitud vs genotipo después de un cambio (una vez que el 60% de la población alcanza el óptimo) en la Figura 5.3	97

5.6	Gráfica que muestra los resultados de la comparación entre parámetros fijos y autocodificados para algoritmos genéticos. La gráfica superior muestre la concentración relativa (CR) de las cuerdas óptimas como función del tiempo. CR-FR es la curva para un AG con tazas fijas, CR-3b para codificación con 3 bits y CR-8b para codificación con 8 bits. En la gráfica inferior se muestre el promedio de las probabilidades de mutación y cruza como función del tiempo para el algoritmo genético autocodificado. Mut3b, Mut 8b, Cross 3b y Cross 8b son las tasas de mutación y entrecruzamiento promedio para las codificaciones de 3 y 8 bits. La población inicial fue elegida aleatoriamente.	98
5.7	Misma que la Figura 5.6 pero con la población inicial concentrada en el punto 49 del paisaje de la Figura 5.3. CR-Cro 3b representa a un algoritmo genético donde la tasa de mutación es fija en 0.01 y la tasa de cruza se codifica con 3 bits. Cros 3b corresponde a la tasa de cruza promedio	99
5.8	Mismos datos que para la Figura 5.6 pero para el paisaje de la Figura 5.4. Población inicial sesgada	100
5.9	La gráfica superior muestra la concentración relativa de las probabilidades de cruza y en la gráfica inferior se presentan las probabilidades de cruza promedio. Ambas como función del tiempo para el paisaje de la Figura 5.4, con 80 individuos por población y una probabilidad de mutación con valor cero. CR-FIT 1 es la concentración relativa del óptimo global en el punto 0, CR-Fit2 es la concentración relativa del óptimo local en el punto 63, Avg Cross es la tasa de cruza promedio y Avg Fit es la aptitud promedio de la población.	101

5.10	Gráfica de la concentración relativa del óptimo (gráfica superior) y probabilidades promedio de cruza y mutación (gráfica inferior) como función de tiempo para el paisaje “jumper” que cambia del paisaje de la Figura 5.3 al de la Figura 5.5 después de que el 60% de la población ha llegado al óptimo de la Figura 5.3. CR-3b y CR-8b son los resultados para los algoritmos genéticos con tasas autocodificadas de 3 y 8 bits, CR-Mut8b es el resultado con sólo la mutación codificada y una probabilidad de cruza de 0 con 40 y 41 siendo la concentración relativa de las cuerdas asociadas con los óptimos locales 40 y 41. Mut 3b, Mut 8b, Cross 3b y Cross 8b son los promedios de las probabilidades para las codificaciones con 3 y 8 bits. La línea para 8b-Mut es la tasa de mutación promedio para el caso de una probabilidad de cruza de 0.	102
5.11	Gráfica de la aptitud promedio poblacional (online fitness) vs tiempo para un paisaje dependiente del tiempo de manera estocástica. La curva continua corresponde al algoritmo con parámetros fijos mientras que la curva punteada corresponde al algoritmo con parámetros codificados	106
5.12	Gráfica de las tasa de mutación y cruza promedio vs tiempo para el paisaje estocástico dependiente del tiempo	107
5.13	Gráfica de la aptitud efectiva vs tiempo para el paisaje “Estocástico” del modelo de la Figura 5.11	108
5.14	Gráfica de mutabilidad promedio sobre 3000 generaciones.	111
5.15	Diversidad de virus por experimento.	112
5.16	Evolución de la población para un experimento.	112
5.17	Gráfica de mutabilidad propia para el experimento de la figura 5.16	113
5.18	Evolución de la población para otro experimento	113
5.19	Gráfica de mutabilidad propia para el experimento de la fig 5.18.	114

5.20	Gráfica del tamaño promedio del cuello de la girafa como función de tiempo para una población de 1000 girafas. La sequía inicia en la generación 240 y dura 30 generaciones. Los valores de los parámetros usados fueron: $p=0.0025$, $p'=10^{-6}$, $\delta=2.0$ y $\epsilon=1.0$	116
5.21	Número de girafas tipo 1 como función del tiempo. Mismos parámetros que en Figura 5.20.	117
5.22	Gráfica de aptitud (longitud de la ruta) en función del tiempo para el problema del agente viajero de 33 ciudades, f2.fit y f2.bes representan el promedio de aptitud y el mejor promedio para un algoritmo genético con tasas fijas mientras que s2.fit y s2.bes para un algoritmo autocodificado con 8 bits.	120
5.23	Gráfica de eficiencia vs tiempo para algoritmos genéticos con parámetros fijos y con parámetros autocodificados para el problema del agente viajero de 33 ciudades. f2.eff representa la eficiencia para el algoritmo con tasas fijas y s2.eff para el algoritmo autocodificado.	121
5.24	Gráfica de aptitud vs tiempo para el problema del agente viajero para el “jumper landscape” donde después de 300 generaciones se añaden 10 ciudades más al conjunto inicial de 23. f2.fit y f2.bes representa la aptitud promedio y la mejor aptitud para un algoritmo con tasa fijas, s2.fit y s2.bes representan los mismos valores pero para el algoritmo autocodificado.	122
5.25	Gráfica de eficiencia vs tiempo para algoritmos genéticos con tasas fijas (fj2.eff) y con tasas autocodificadas (sj2.eff) para el “jumper landscape”.	123
5.26	Gráfica de la tasa de entrecruzamiento vs tiempo para un paisaje plano (s2.cro) de 33 ciudades y para el “jumper landscape” (sj2.cro)	124
5.27	Precios del 17 de marzo al 15 de junio	144
5.28	Precios del 17 de marzo al 14 de abril	145

5.29	Valor de portafolio para diferentes configuraciones, del 17 de marzo al 14 de abril. AG:algoritmo genético. Pm: participación de mercado, 1/3 configuración, 1/3 Banacci O 1/3, Telmex L y 1/3 efectivo. 1/2: 1/2 Banacci O, 1/2 Telmex L	146
5.30	Ganancia diaria porcentual normalizada $G_{dpm}(t) = \frac{(G(t)-G(0))*100}{G(0)}$	147
5.31	Valor de portafolio usando mejor estrategia del periodo anterior contra valor de portafolio usando estrategia generada a partir de los datos de este período del 17 de marzo al 14 de abril	148
5.32	Comparación de ganancia diaria porcentual entre un portafolio que usa la mejor estrategia generada por un algoritmo genético inteligente para el período anterior (Predicción) y un portafolio que usa “Buy & Hold” con una inversión de 1/3 para cada una de las emisoras y el efectivo (B & H 1/3). Período del 17 de marzo al 14 de abril	148
5.33	Precios del 17 de abril al 18 de mayo	149
5.34	Valor de portafolio para diferentes configuraciones, del 17 de abril al 18 de mayo	149
5.35	Ganancia diaria porcentual normalizada $G_{dpm}(t) = \frac{(G(t)-G(0))*100}{G(0)}$ del 17 de abril al 18 de mayo	150
5.36	Valor de portafolio usando mejor estrategia del periodo anterior contra valor de portafolio usando estrategia generada a partir de los datos de este período del 17 de abril al 18 de mayo	150
5.37	Comparación de ganancia diaria porcentual entre un portafolio que usa la mejor estrategia generada por un algoritmo genético inteligente para el período anterior (Predicción) y un portafolio que usa “Buy & Hold” con una inversión de 1/3 para cada una de las emisoras y el efectivo (B&H 1/3). Período del 17 de abril al 18 de mayo	151
5.38	Precios del 19 de mayo al 15 de junio	151
5.39	Valor de portafolio para diferentes configuraciones, del 19 de mayo al 15 de junio	152

5.40	Ganancia diaria porcentual normalizada $G_{dpm}(t) = \frac{(G(t)-G(0)) \cdot 100}{G(0)}$ del 19 de mayo al 15 de junio	152
5.41	Valor de portafolio usando mejor estrategia del periodo anterior contra valor de portafolio usando estrategia generada a partir de los datos de este período del 19 de mayo al 15 de junio	153
5.42	Comparación de ganancia diaria porcentual entre un portafolio que usa la mejor estrategia generada por un algoritmo genético inteligente para el período anterior (Predicción) y un portafolio que usa "Buy & Hold" con una inversión de 1/3 para cada una de las emisoras y el efectivo (B&H 1/3). Período del 19 de mayo al 15 de junio	153

Resumen

La evolución ha demostrado que la habilidad para adaptarse a los cambios en el medio ambiente y permitir que estas adaptaciones sean heredados permite el desarrollo de los individuos. Dada la aparición imprevista de estos cambios, sólo los organismos con estructuras genéticas flexibles y robustas logran sobrevivir.

Esta tesis esta inscrita dentro del campo de la computación evolutiva, especialmente bajo el enfoque de los algoritmos genéticos, propuestos como un modelo de proceso adaptivo, además de ser considerado como un método de optimización.

La tesis aporta resultados tanto teóricos como experimentales para comprender y aplicar conceptos presentes en la evolución como “mapeo genotipo-fenotipo”, “rompimiento de simetría inducido” y “autoadaptación” para mejorar el desempeño de los algoritmos genéticos al aplicarse a problemas cuya función a optimizar es dependiente del tiempo (optimización dinámica) como por ejemplo los sistemas de control de flujo, redes de transporte, líneas de producción entre otros, los cuales son de difícil solución por medio de las técnicas tradicionales de la investigación de operaciones.

También se desarrolla el concepto de “aptitud efectiva” el cual permite entender los enfoques aparentemente encontrados de las principales teorías de la evolución, y definir ecuaciones precisas que expliquen el funcionamiento de los algoritmos genéticos.

Agradecimientos

A mis padres por su ejemplo.

A mi esposa y a mi hija por su ayuda y comprensión.

Al Dr. Christopher R. Stephens por su guía, paciencia y apoyo para la elaboración de esta tesis.

Al Dr. Henri Waelbroeck por sus comentarios y motivación.

A los miembros del Jurado por sus recomendaciones y sugerencias.

A las siguientes instituciones que me apoyaron económicamente durante el transcurso del doctorado:

- UNAM (DGAPA, DGEP)
- CONACYT
- Fundación TELMEX.

Aporte de la tesis

Se hace una crítica al concepto tradicional de paisajes de aptitud y se presenta un concepto alternativo, la aptitud efectiva, que permite un mejor entendimiento de la dinámica de un sistema complejo. Aún cuando en este trabajo se usan algoritmos genéticos los descubrimientos principales pueden ser aplicables a otras áreas de la computación evolutiva (EC por sus siglas en inglés) como las Estrategias evolutivas (ES por sus siglas en inglés) y a la Programación Genética (GP por sus siglas en inglés).

Los resultados de la tesis doctoral permitirán mejorar el entendimiento de los fenómenos que aparecen en algunos sistemas complejos como es el caso de la evolución. Tales fenómenos incluyen: rompimiento de simetría inducido, organización del conjunto genético, emergencia e identificación de comportamientos globales a partir de interacciones locales y autoadaptación a cambios en el ambiente.

Estos resultados podrán ser aplicados a problemas prácticos de importancia para la investigación de operaciones como son aquellos que tratan con la búsqueda de óptimos en topografías dinámicas complejas. Además debido a la estructura de los algoritmos genéticos inteligentes es posible trabajar sobre topográficas dependientes del tiempo, con óptimos no estacionarios y con cambios en los criterios de optimización, aumentando con ello el campo de aplicación de los resultados de la tesis.

El diseño de los algoritmos genéticos “inteligentes” proviene del estudio de los sistemas evolutivos, en especial del manejo en la naturaleza del mapeo genotipo-fenotipo el que provee una manera de obtener comportamientos y estructuras complejas a partir de reglas y estructuras simples.

A pesar de que los algoritmos genéticos han sido ampliamente utilizados para resolver problemas prácticos, en general no está claro como es que funcionan en clases diferentes de problemas, por lo que otro de los aportes de esta tesis es el hacer un análisis teórico sobre los resultados obtenidos por los algoritmos genéticos aplicando la ecuación Stephens-Waelbroeck, demostrando su validez y uso para los problemas considerados.

También es un intento para aplicar el concepto de autoorganización, cuya existencia se observa más fácilmente al hacer uso explícito del mapeo genotipo-fenotipo, las cuales permiten obtener sistemas robustos bajo modificaciones súbitas de las condiciones en las que el sistema se desenvuelve, enfoque que permite la aplicación a problemas de optimización dinámica. Quizás el aporte fundamental de la tesis es la demostración de que los algoritmos genéticos que usan un mapeo genotipo fenotipo no trivial puede llevar a robustez siempre que se logre la autoorganización de este mapeo. La robustez de un sistema permite a este adaptarse a cambios en el entorno, requisito indispensable para atacar problemas de optimización dinámica. Así mismo esta organización del mapeo permite superar el problema al que se enfrenta métodos de optimización estática, denominado “fragilidad” (“brittleness”) que hace que una solución sea válida bajo condiciones específicas del problema.

Los resultados de esta tesis dieron como resultado la publicación de 4 artículos en revistas arbitradas:

- C.R. Stephens and J. Mora, “Effective Fitness as an Alternative Paradigm in Evolutionary Computation I: General Formulism ”, *Genetic Programming and Evolvable Hardware*, Vol. 1 Issue 4. Kluwer (2000).
- C.R. Stephens and J. Mora, “Effective Fitness as an Alternative Paradigm in Evolutionary Computation II: Examples and Applications”, a ser publicado en *Genetic Programming and Evolvable Hardware*, Kluwer (2000).
- J. Mora, C.R. Stephens, H. Waelbroeck and F. Zertuche *Symmetry Breaking and Adaptation: Evidence From a Simple Toy Model of a Virus*, *Biosystems* 51, pp. 1-14 (1999).
- C.R. Stephens, I. García Olmedo, J. Mora Vargas and H. Waelbroeck, *Self-Adaptation in Evolving Systems*, *Artificial Life* 4.2, pp 183-201 (1998).

Introducción

La evolución es el principio unificador del pensamiento moderno en la biología. La teoría clásica de Darwin de la evolución combinada con la selección de Weismann y la genética de Mendel son considerados actualmente como el paradigma neo-Darwiniano [1-8]. El neo-Darwinismo sugiere que la vida es explicada por unos pocos procesos estadísticos actuando sobre poblaciones [9]. Estos procesos son la reproducción, la mutación, la competencia y la selección. La reproducción es obviamente una propiedad necesaria para la vida, la mutación (el cambio genético) esta presente en todo sistema que se reproduce así mismo. La competencia y la selección aparecen como consecuencia de una población creciendo dentro de un espacio finito. La evolución es entonces el resultado de estas interacciones entre procesos fundamentales actuando sobre las poblaciones, generación tras generación [10, 11]. Como se menciona en [12]: “Nada en la biología tiene sentido si no se contempla a la luz de la evolución”.

El pensamiento evolutivo se extiende más allá del estudio de la vida, ya que al considerar a la evolución como un proceso de optimización este se puede aplicar a problemas de ingeniería , por ejemplo, que pueden ser simulados usando una computadora. El interés en tales simulaciones se ha incrementado con el desarrollo de aplicaciones para reemplazar tecnologías convencionales en sistemas de potencia, reconocimiento de patrones, sistemas de control, calendarización en fábricas, diseño

- C.R. Stephens and J. Mora, “*Effective Fitness as an Alternative Paradigm in Evolutionary Computation I: General Formulism*”, *Genetic Programming and Evolvable Hardware*, Vol. 1 Issue 4. Kluwer (2000).
- C.R. Stephens and J. Mora, “*Effective Fitness as an Alternative Paradigm in Evolutionary Computation II: Examples and Applications*”, a ser publicado en *Genetic Programming and Evolvable Hardware*, Kluwer (2000).
- J. Mora, C.R. Stephens, H. Waelbroeck and F. Zertuche *Symmetry Breaking and Adaptation: Evidence From a Simple Toy Model of a Virus*, *Biosystems* 51, pp. 1-14 (1999).
- C.R. Stephens, I. García Olmedo, J. Mora Vargas and H. Waelbroeck, *Self-Adaptation in Evolving Systems*, *Artificial Life* 4.2, pp 183-201 (1998).

de medicinas y otras áreas. En el caso de la investigación de operaciones el desarrollo de las computadoras en los últimos años ha permitido abordar problemas de optimización discreta mediante el desarrollo de técnicas heurísticas, como el recocido simulado, la búsqueda tabú y otros métodos de búsqueda secuencial.

Los algoritmos evolutivos, en especial los algoritmos genéticos, han jugado un papel importante en una gran diversidad de problemas (en [13] se encuentra un resumen de algunas aplicaciones).

Hasta la fecha la mayoría de los trabajos sobre algoritmos genéticos tratan sobre los aspectos técnicos de la implementación computacional, de su codificación y de los resultados como método de optimización, así como su comparación con otros métodos. Estos resultados no siempre son favorables para los algoritmos genéticos ya que estos por su naturaleza poblacional permiten encontrar un conjunto de óptimos a diferencia de otros métodos que se concentran en encontrar al mejor (como es el caso de la búsqueda tabú y del recocido simulado). Sin embargo precisamente por esa misma naturaleza es que tienen un elevado desempeño dentro del contexto de la optimización dinámica. Desde la perspectiva de la evolución natural el concepto de óptimo global como se entiende en la investigación de operaciones pierde validez dado que se trata de un ambiente cambiante donde el óptimo global puede dejar de serlo al modificarse las condiciones en las que se desarrolla el problema. Este enfoque permite aplicar los algoritmos genéticos inteligentes a problemas reales cuyas condiciones cambian en el tiempo como en el caso del manejo de contingencias en problemas de calendarización, asignación de rutas o para el desarrollo de sistemas "inteligentes", i.e. capaces de reaccionar a cambios en su medio (máquinas, robots) entre otros. Como ejemplo imagine el sistema de control de tráfico en su ciudad (semáforos, policía

vial) bajo condiciones normales (sin accidentes automovilísticos, descomposturas de vehículos, manifestaciones o reparaciones viales) uno puede llegar de un lugar a otro de la ciudad sin problemas, pero qué pasa cuando se tiene una contingencia seria que afecte drásticamente el tráfico? Cómo ubicar al personal disponible para coordinar la vialidad o en qué cantidad modificar la frecuencia en las señales de los semáforos para permitir la descongestión del sistema? Desde el punto de vista computacional esto podría hacerse mediante el uso de un sistema experto al que habría que alimentarlo con todos los escenarios posibles y sus respectivas respuestas lo cual implica un sistema ineficiente por la gran cantidad de reglas a manejar. Otra manera sería el implementar un sistema que por sí mismo decidiera las acciones necesarias adaptándose a los cambios en el ambiente. Este sistema bien podría ser un algoritmo genético inteligente.

Un algoritmo genético es una versión computacional idealizada de la evolución natural en la cual los organismos se reproducen a diferentes tasas y los más capaces sobreviven y heredan sus características después de procesos de modificación en su estructura genética como la mutación aleatoria y la recombinación. A diferencia de métodos como el recocido simulado y de la búsqueda tabú que exploran el espacio de soluciones de manera secuencial, los algoritmos genéticos permiten la evolución de un conjunto de individuos que representan una solución al problema. La calidad de cada uno de los individuos es medida mediante una función de aptitud y cada individuo contribuye a la formación de la población siguiente de manera proporcional a dicha aptitud. Esta contribución se ve modificada por los operadores genéticos de mutación y la recombinación, los cuales tienen la función de explorar las nuevas soluciones y por el operador de selección el cual explota dichas soluciones. Para ilustrar esto,

considere una máquina tragamonedas con dos brazos, uno de ellos -no se sabe cual- en promedio tiene más aciertos que el otro, la estrategia con el objetivo de maximizar las ganancias es usar la mayoría de las veces el brazo de alta ganancia pero, cómo hacer si no se sabe cuál es? esto provoca una competencia entre la exploración del conocimiento (identificar el brazo que provee mayores ganancias) y la explotación del conocimiento (usar más veces el brazo con altas ganancias).

La importancia de la investigación en algoritmos genéticos radica en su flexibilidad y adaptabilidad a los problemas, en combinación con un desempeño robusto debido a la manera en que interactúan sus elementos constitutivos. La robustez que caracteriza a los algoritmos genéticos les permite atacar problemas con estructura dependiente del tiempo, es decir aquellos cuya función a optimizar es dinámica. Las técnicas heurísticas tradicionales se enfocan principalmente en problemas estáticos.

Dentro de los métodos de optimización discreta que han dado muy buenos resultados al abordar problemas estáticos se encuentran el recocido simulado técnica propuesta por Kirkpatrick, Gelatt y Vecchi [14] basada en la analogía entre el fenómeno físico de recocido de sólidos (calentamiento del sólido y enfriamiento lento para obtener un estado de cristalización minimizando la energía). Para resolver problemas de optimización combinatoria, es importante mencionar que uno de las etapas más difíciles en la aplicación del recocido simulado es la determinación del calendario de enfriamiento. En Gutiérrez [15], por ejemplo, se hace un análisis detallado de los aspectos teóricos de la técnica y se desarrollan aplicaciones para algunas familias de problemas combinatorios. Otro método de optimización estática mencionado con anterioridad es la búsqueda tabú, desarrollado originalmente por Glover [16] y Hansen [17]. El procedimiento en su forma actual se debe a Glover [18],[19],[20]. En De los Cobos

[21], entre otros, se presentan varias aplicaciones de búsqueda tabú para problemas de calendarización y para el manejo de inventarios multiproducto.

Los métodos de optimización se pueden clasificar en: enumerativos, matemáticos, heurísticos e inteligentes. Los métodos enumerativos son aquellos que enumeran y comparan todas las posibles soluciones como por ejemplo: búsqueda exhaustiva, tienen la ventaja de no verse afectados por la solución inicial. Los métodos matemáticos se basan en la formulación del problema como problema matemático como es el caso de la programación entera, lineal, no lineal, programación dinámica, y métodos de gradiente entre otros. Los métodos heurísticos están basados en reglas o métodos prácticos que buscan una buena solución como recocido simulado, búsqueda tabú, algoritmos genéticos ó ramificación y acotamiento. Finalmente los métodos inteligentes como las redes neuronales o los algoritmos genéticos inteligentes emplean la autoorganización inteligente (no aleatoria) para responder a cambios en el ambiente.

Algunas de las desventajas de estos métodos son:

- La convergencia hacia una solución óptima depende de la solución inicial elegida para el caso de métodos como el de gradiente.
- Muchos de los algoritmos tienden a estancarse en un óptimo local como el caso del gradiente.
- Un algoritmo eficiente en la solución de un problema de optimización puede no ser eficiente en la solución de un problema diferente principalmente en el caso de los métodos heurísticos.
- Los métodos matemáticos en general son más lentos ya que requieren la evaluación de la función objetivo muchas veces.

La existencia de un conjunto de entes que evoluciona (característica principal de un algoritmo genético) permite que este conjunto se organice asimismo en una colectividad con patrones definidos y cuyos elementos intercambian información, permitiendo la adaptación y el aprendizaje mostrando con ello la emergencia de inteligencia definida como la capacidad de un sistema para adaptar su comportamiento para alcanzar sus metas dentro de un medio cambiante [22], como es el caso en los problemas de optimización dinámica.

La capacidad de aprendizaje y adaptación se debe a la organización del conjunto genético a partir de la configuración de los elementos del algoritmo en genotipo (grupo de características esenciales que contienen toda la información del individuo) y fenotipo (expresión final del genotipo) permitiendo la existencia de sinónimos (varios genotipos que representan al mismo fenotipo) que responde de manera diferente a cambios en el ambiente permitiendo la sobrevivencia. Los algoritmos genéticos inteligentes, los cuales se basan en la aplicación explícita del mapeo genotipo-fenotipo permitiendo una alta redundancia de información como aquella que se aprecia en la evolución natural, son un enfoque que poco se ha estudiado y que permite la explotación del procesamiento paralelo, vislumbrando un mejor desempeño de esta estrategia evolutiva.

El fenómeno del rompimiento de simetría inducido y el concepto de aptitud efectiva permiten el desarrollo de los algoritmos genéticos inteligentes. El rompimiento de simetría inducido puede interpretarse como una autoorganización en el mapeo genotipo-fenotipo. Algunos ejemplos son: el fenómeno de bloat en la programación genética -crecimiento incontrolable de individuos que alcanzan su profundidad permitida, la hipótesis más popular sugiere que este crecimiento se debe a segmentos de

código que al tener poca aptitud se mantienen cerca de los bloques con alta aptitud en el código, la autoadaptación en algoritmos genéticos y la emergencia de lenguaje en sistemas genéticos.

La modelación computacional se ha convertido en complemento y en algunos casos en suplemento de los enfoques tradicionales de la teoría matemática así como de la experimentación directa.

El estudio de los algoritmos genéticos va de la mano con el avance de la ciencia computacional, ya que así como la microbiología empezó con el desarrollo del microscopio, la física experimental de altas energías necesito de los aceleradores de partículas, el estudio de los algoritmos genéticos requiere de la computadora como instrumento. Así los algoritmos genéticos se han inspirado en la evolución biológica haciendolo uno de los más prometedores métodos computacionales que se han aplicado a problemas prácticos anteriormente intratables incluyendo seguridad computacional y la creación de sistemas de aprendizaje autónomos. De esta forma los modelos computacionales sirven como exploradores para los sistemas naturales y a su vez estos inspiran nuevas tecnologas computacionales, de ahí la aparición de disciplinas como la "computación adaptiva", "computación neural" o la "vida artificial".

Capítulo 1

Evolución

1.1 Evolución, competencia y adaptación

La evolución es un proceso basado en la competencia y en la adaptación, el cual aparece en diversos ámbitos que van desde los más tangibles como la evolución natural hasta los más abstractos como los mercados financieros. La evolución es un proceso que se presenta en diversas escalas temporales como espaciales, por ejemplo se puede hablar de la evolución de una sociedad, de un país, de un continente o del mundo (cambios de escala espacial) y también de la evolución de los mercados financieros (escala temporal, particularmente a través de los diferentes horizontes de inversión). La característica principal de los individuos que tienen éxito en este proceso se puede medir por su permanencia en las diferentes escalas, así se podría hablar del fracaso de los dinosaurios para responder a cambios súbitos (dentro de la escala de tiempo pertinente) en el ambiente que provocaron su extinción.

De manera simplificada la evolución es el resultado de la interacción entre la creación de nueva información genética, su evaluación y selección, i.e. hace énfasis más en las estrategias que en las características, por ejemplo en los mercados financieros las características de los individuos en función del dinero que estos tengan

dará como resultado la utilización de diferentes estrategias (algunos tomarán sus decisiones al azar, lo que tienen más dinero podrán hacer uso de equipos de investigación o de potentes equipos de cómputo). Un individuo de la población es afectado por otros individuos de la misma (mediante la competencia por la comida, el ataque de predadores o el apareamiento) así como por el ambiente (existencia de alimentos y clima). La oportunidad del individuo para sobrevivir y tener descendencia (la cual heredará la información paterna modificada) dependerá de la manera que este se desenvuelva bajo estas condiciones. En el curso de la evolución, esto lleva a una penetración de la población con la información genética de los individuos mejor adaptados. La naturaleza no determinista de la reproducción lleva a una permanente producción de nueva información genética y por lo tanto a la creación de diferentes descendencias.

La evolución es un ejemplo clásico de un “sistema complejo” caracterizado por una gran cantidad de “grados de libertad” (variables o coordenadas necesarias para describir el estado del sistema), cuyas interacciones son altamente no lineales entre ellas, presentando un comportamiento no aleatorio. La definición de escala elegida para el estudio del sistema permite la identificación de los llamados “agentes autónomos” (para una población de hormigas, el agente estaría representado por cada hormiga, en un mercado financiero el agente es cada uno de los entes dispuestos a negociar sus acciones) sobre los cuales se definen los grados de libertad y sus interacciones no lineales, por ejemplo, en una escala microscópica los agentes autónomos del sistema de evolución natural del ser humano puede ser una célula, pero si lo que interesa es el desarrollo como sociedad el agente autónomo sería la familia. Cuando se trabaja en sistemas artificiales como los relacionados con el estudio de la evolución de los

mercados financieros a estos agentes se les denomina “agentes artificiales”.

Visto como un proceso de aprendizaje, la evolución natural resulta en un aprendizaje de largo plazo de la experiencia colectiva de generaciones de poblaciones de organismos. En otras palabras, cada criatura viva es el resultado de millones de años de aprendizaje de sus ancestros acerca de como sobrevivir en la tierra el tiempo suficiente para reproducirse. Por otra parte la evolución es un problema de optimización dinámica cuya función cambia con el tiempo causado por la interacción de las diferentes elementos de una población con el medio. La definición de éxito evolutivo puede medirse en función de la existencia de una especie como tal a lo largo del tiempo. Algunas de las características principales de los sistemas evolutivos reales:

- Son problemas combinatoriales dada la gran cantidad de elementos que interactúan entre sí, por ejemplo, la organización del ADN para permitir la evolución del ser humano, el cual es una optimización sobre un espacio de 10^{128} posibles combinaciones.
- Gran variación en sus escalas tanto espacial como temporal (evolución de un organismo, de una sociedad, de una especie; desarrollo en el corto mediano y largo plazo).
- Interacciones no lineales, donde un elemento depende de varios otros, por ejemplo la localización de estaciones de transferencia de basura sobre un conjunto de sitios posibles al tratar de maximizar al mismo tiempo el flujo entre las estaciones y la distancia entre ellas.

Un ejemplo de un sistema evolutivo real son los mercados financieros donde los agentes (entes negociando en la bolsa de valores) interactúan indirectamente a través

de sus posturas de compra y venta y directamente al estar en comunicación entre ellos por medio de boletines de prensa. Las interacciones son no lineales y la dinámica es complicada en todas sus escalas (una estrategia de promedio móvil (cálculo del promedio precios sobre N días anteriores, se considera recomendación de compra cuando la gráfica de precios supera la gráfica de los promedios) de que haya funcionado bien los últimos diez días puede llevar a una decisión errónea mañana), ya que no se ha encontrado una cartera de instrumentos financieros que tenga una trayectoria muy predecible.

1.2 Requerimientos para la existencia de evolución

Las condiciones necesarias para la ocurrencia de evolución por selección natural son, de acuerdo a [23] y [24]

- Reproducción de individuos en la población. Para permitir la sobrevivencia de la misma. Las tasas de reproducción varían entre los individuos.
- Variaciones que afecten la sobrevivencia de los individuos, propiciando la modificación en sus características. Estas variaciones pueden ser por mutaciones aleatorias o recombinación sexual, entre otras.
- Herencia de características, mediante la combinación de la información individual de los padres permitiendo adaptación al medio.
- Recursos finitos que provoquen competencia, los que serán consumidos por los individuos mejor adaptados.

Estos factores cambian las características de la población en el tiempo.

1.3 Mecanismos biológicos

La evolución ocurre aún en un sistema no vivo simple como en el caso de ambientes *in vivo* (dentro de tubos de ensayo). Por ejemplo, la evolución puede ser observada en experimentos sencillos usando la enzima $Q\beta$ replicante y ARN. Orgel ha hecho series de tales experimentos [25] observando adaptación a las condiciones iniciales, resultados difíciles de obtener por efecto aleatorio. Asimismo estos resultados indican la existencia de una búsqueda evolutiva (considerando que hay más de 10^{128} posibles estructuras para moléculas de ARN el hacer una búsqueda exhaustiva llevaría más tiempo que la edad del universo). Esto da una idea de la potencia de la evolución como sistema de optimización que como veremos más tarde está basado en la explotación de características como la simetría de sinónimos y su rompimiento inducido. La existencia de redundancia en la información debido a esta simetría de sinónimos se ve reflejado en la naturaleza mediante la existencia de estructuras de ADN que no codifican a proteínas, llamadas introns, los cuales se supone previenen los daños en la estructura genética durante la recombinación; estructuras análogas se presentan en la programación genética como código extra que parece emerger espontáneamente del proceso de evolución como resultado de la longitud variable en las estructuras de programación dinámica [26], estas estructuras suelen tener una aptitud pequeña y tienden a estar cerca de estructuras de mucho mayor aptitud haciendo que los individuos tiendan a crecer sin control hasta alcanzar el máximo permitido, este es el llamado fenómeno de "bloat" [26].

La información aprendida durante la evolución biológica es normalmente almacenada en las bases pares de ADN de manera asombrosamente simple. Cada par representa una parte de la instrucción para la creación de un aminoácido particular.

Una base par esta compuesto de dos ácidos nucleicos que están químicamente unidos. Sólo cuatro diferentes bases aparecen en el ADN: adenina, guanina, citosina y timina, abreviados A,G,C y T respectivamente. Las reglas de unión son simples, A con T y G con C, teniendo con esto cuatro configuraciones posibles $A \leftrightarrow T$, $T \leftrightarrow A$, $G \leftrightarrow C$ y $C \leftrightarrow G$. Estas bases están unidas unas con otras formando una doble hélice. Note la elegancia de esta estructura. Cada uno de los segmentos de ADN es redundante —si C aparece en un segmento G debe aparecer en la misma posición. La molécula entera de ADN puede ser reconstruida a partir de uno de los segmentos. De hecho el ADN tiene muchos mecanismos de reparación que explotan esta redundancia [27]. Un conjunto de tres bases forman un codón el cual es un patrón para la producción de un aminoácido particular o representa una secuencia de terminación. Algunos ejemplos de codones son: ATG codifica la metionina, CAA y CAG la glutamina.

Así se tienen 64 diferentes ($4 * 4 * 4$) codones pero sólo hay treinta aminoácidos y una secuencia de terminación por lo que hay aminoácidos representados por más de un codón. Este es un ejemplo claro de simetría de sinónimos (varios codones codifican al mismo aminoácido, mostrando una codificación claramente redundante) permitiendo la búsqueda de nuevas características mediante los operadores genéticos.

Estas secuencias actúan como instrucciones o instrucciones parciales en programas de computadora, mediante la construcción de proteínas y su secuencia de manufactura [28]. Esta naturaleza de programación del ADN, junto con su estructura de longitud variable, explica el atractivo de la evolución biológica como modelo de programación computacional.

La definición explícita de genotipo y fenotipo la cual parece funcionar asombrosamente bien en la biología no se ha aplicado en el contexto de los algoritmos genéticos,

esta tesis contribuye a su desarrollo.

En 1909, Johanssen hizo la distinción entre la apariencia de un organismo y su constitución genética acuñando los términos de *genotipo* y *fenotipo* [29]. La evolución interacciona en forma diferente sobre el genotipo que sobre el fenotipo, tradicionalmente se ha visto que la selección actúa sobre el fenotipo. El genoma o genotipo de un organismo es el ADN de dicho organismo. La mitad del genoma es pasado de padre a hijo. El fenotipo o fenoma es el conjunto de propiedades observables de un organismo.

1.4 Aptitud y paisajes de aptitud

El principio fundamental de la teoría Darwiniana es la “supervivencia de los más aptos”, pero qué significa “apto”. Hay varias maneras de definir la “aptitud” pero todas están de acuerdo en su sentido más amplio que es una medida del “éxito” reproductivo. Matemáticamente se define de una manera más sencilla: $f_Q : Q \rightarrow R^+$ donde Q es el espacio de los fenotipo. El dominio de f_Q puede ser diferente de R^+ ; e.g. los enteros en un intervalo finito.

De manera convencional se considera a la aptitud como una función del fenotipo siendo este la expresión física del genotipo y por lo tanto el objeto sobre el que se percibe el efecto de la selección natural. Para entender la manifestación de la aptitud a nivel del genotipo es necesario el definir un mapeo entre genotipo y fenotipo, $\phi : G \rightarrow Q$, donde G es el espacio de los genotipos. La existencia de las funciones ϕ y f_Q permite definir sobre G una función de aptitud, $f_G = f_Q \circ \phi$, inducida por la acción de ϕ y f_Q .

Siempre que el mapeo genotipo-fenotipo sea no-inyectivo la función f_G será degenerada, muchos genotipos correspondiendo al mismo valor de aptitud, y por lo tanto no inyectiva. La aptitud define una relación de equivalencia sobre G , muchos genotipos equivalentes. A la equivalencia de estos genotipos bajo la acción de la selección reproductiva (la que considera a todos los genotipos como igualmente aptos) se le denomina “simetría” entre ellos.

Uno de los enfoques sobre el concepto de aptitud dentro de la evolución se relaciona con el número de descendientes de un organismo y la habilidad del organismo para llegar a la edad reproductiva, siendo la primera la definición comúnmente aceptada, pero uno de los problemas de esta es su naturaleza tautológica manifiesta en la frase “sobrevivencia del más apto”. Si uno define la aptitud efectiva (aptitud que mide el éxito reproductivo total para un genotipo partícula) mediante una relación del tipo:

$$n_i(t+1) \propto f_{eff}^i n_i(t) \quad (1.4.1)$$

donde $n_i(t)$ es el número de individuos del tipo i presente en la población al tiempo t esto no tiene valor predictivo si el único medio de evaluarlo es el medir n_i en dos generaciones sucesivas a menos que, por ejemplo, se conozca que es constante en el tiempo. Sin embargo, si uno puede desarrollar un algoritmo para calcularlo entonces este se puede aplicar para predecir la dinámica poblacional.

El enfoque adoptado por la computación evolutiva está vinculado con la probabilidad de que el individuo sobreviva hasta su edad reproductiva [30], donde la reproducción puede ser biológica o algorítmica. Esta aptitud reproductiva es una propiedad del individuo, que no depende de otros genotipos aún cuando la función de aptitud puede reflejar efectos ambientales. Esta aptitud se definirá, para un genotipo c_i , por $f(c_i, t)$ o $f(c_i)$, dependiendo si la función es dependiente o no del tiempo, y la

aptitud efectiva por $f_{eff}(c_i, t)$. En términos de operadores genéticos la aptitud reproductiva toma en cuenta solamente al operador de selección mientras que la aptitud efectiva toma en cuenta a todos.

En esta tesis se mostrará que los efectos de los operadores aparte de la selección es tal que con mucha frecuencia el flujo de la población sobre un paisaje de aptitud estándar no puede ser entendido por intuición. Cuando el operador de selección domina sobre el resto de los operadores ambas medidas de aptitud son aproximadamente iguales. Pero en el otro extremo, cuando todos los genotipos tienen la misma probabilidad de alcanzar la madurez, ambos valores son completamente diferentes.

El paisaje de aptitud se puede entender como un paisaje rugoso donde la aptitud es la variable dependiente, definida sobre un espacio D_Q o D_G -dimensional donde D_Q y D_G son las dimensiones de los espacios Q y G respectivamente. Sin embargo esta idea de paisaje rugoso lleva a la concepción de una métrica euclidiana cosa que no siempre sucede en los sistemas evolutivos. Con el fin de mantener la idea gráfica de un paisaje de aptitud es necesario el definir una función de distancia, algo que de idea de cercanía o lejanía para tener una relación de orden entre G y Q .

Una función de distancia muy común es la llamada distancia de Hamming, d_{ij} , entre dos genotipos c_i y c_j . La cercanía es definida por la similaridad bit a bit entre dos cuerdas. Para una representación binaria el dominio de la función de aptitud es un hipercubo donde cada vértice representa una posible cuerda y cada unión entre vértices es un cambio en uno de sus elementos. Al considerar al entrecruzamiento más que a la mutación se puede alegar que la distancia de Hamming no es la métrica natural por lo que se debe elegir una nueva.

Si la función de aptitud es una función con dependencia temporal explícita, el

correspondiente paisaje será dinámico. Hasta el momento la mayoría de los análisis sobre los paisajes de aptitud han estado restringidos a condiciones estáticas.

El efecto de la simetría sobre los paisajes de aptitud en función del genotipo se ve reflejado en la existencia de “direcciones” planas asociadas con la inexistencia de gradientes selectivos entre sinónimos. El concepto de “dirección” es más apropiado para el caso en que los sinónimos son vecinos, que como se ha mencionado anteriormente depende de la métrica seleccionada. Es importante mencionar que es el paisaje sobre G el que permite la existencia de direcciones planas que por definición aparecen si existen sinónimos.

Capítulo 2

Computación evolutiva

La idea de programas de computadora evolutivos es casi tan vieja como la computadora en sí. Pero aunque los orígenes de la computación evolutiva puede trazarse a finales de los '50s (ver los trabajos de Bremmerman [31], Friedberg [32],[33], Box [34]), el campo permaneció relativamente inexplorado para la mayoría de la comunidad científica por al menos tres décadas. Esto fue debido principalmente a la carencia de computadoras poderosas pero también debido a algunas deficiencias metodológicas (ver Fogel [22], p.103).

El trabajo fundamental de Holland [35], Rechenberg [36], y Fogel [37] sirvió para cambiar estas deficiencias durante la década de los 70's y actualmente se observa un marcado incremento en el número de publicaciones (por ejemplo vea la bibliografía en Alender [38]) y conferencias en este campo, una clara demostración de la relevancia científica y económica de este tema.

Los beneficios de la computación evolutiva radican en la ganancia de flexibilidad y adaptabilidad para el desempeño de una tarea determinada en combinación con el desempeño robusto y sus características como optimizadores. De hecho la computación evolutiva debe ser entendida como un concepto general adaptable para la resolución

de problemas, especialmente con un buen desempeño al tratar con problemas de optimización.

La mayoría de las implementaciones actuales de algoritmos evolutivos descienden de tres enfoques fuertemente relacionados pero con desarrollo independiente: *algoritmos genéticos*, *estrategias evolutivas* y *programación evolutiva*. Otras variantes de los algoritmos evolutivos son: los *sistemas clasificadores* y la *programación genética*.

Todos estos métodos de computación evolutiva permiten la existencia de evolución ya que en todos ellos se tiene una reproducción de individuos (cromosomas, máquinas de estado finito, vectores x_i) mediante diversas técnicas (torneo, reproducción (μ, λ) , $(\mu + \lambda)$). La herencia de la información genética y su diversidad es implementada por los operadores genéticos de cada uno de los métodos (inversión y entrecruzamiento de bits en algoritmos genéticos, adición de un número con distribución uniforme para estrategias evolutivas, entre otros). Los recursos finitos que serán consumidos pueden estar representados por la memoria de las computadoras donde se procesan estos métodos.

2.1 Algoritmos evolutivos

La evolución natural ha sido suficientemente poderosa para producir fenómenos tan complejos como los mamíferos y la conciencia humana. Además de su complejidad los sistemas biológicos parecen tan bien adaptados a sus ambientes y por lo tanto bien equipados con “dispositivos” motores y de percepción que hacen pensar en la optimización a propósito. Esto ha causado controversia en la biología sobre si la evolución es de hecho la optimización de estructuras o no [39].

Durante mucho tiempo los ingenieros han aprendido de la naturaleza para aplicar

sus logros en la tecnología. Por lo que la evolución ha estimulado dos preguntas formuladas en el ámbito de la ciencia computacional:

1. El simular la evolución (con algoritmos evolutivos) servirá como herramienta para atacar problemas como la optimización de dispositivos como alas de aviones, motores de vehículos o receptores y sensores para cierto tipo de señales físicas? o procesos como la calendarización de actividades, el diseño de rutas de transporte o la ubicación de plantas nucleoelectricas?
2. La simulación de la evolución en algunos de sus aspectos permitirá encontrar mejores soluciones a las existentes para problemas cuya dificultad es bien conocida?

Una motivación de los algoritmos evolutivos es el contestar estas preguntas. Basados en modelos muy sencillos de la evolución orgánica, estos algoritmos ayudan a comprender los ingredientes básicos de los procesos de evolución natural. Con estos ingredientes los algoritmos evolutivos se aplican a varios problemas como por ejemplo la optimización combinatoria o el aprendizaje de tareas.

Los ingredientes básicos de los algoritmos evolutivos se pueden resumir en:

- Poblaciones de soluciones
- Operadores de innovación
- Operadores de "conservación"
- Diferenciales de calidad
- Selección

Considere un problema de optimización, el cual requiere encontrar un conjunto $\vec{x} \in M$ de parámetros del sistema considerado, tal que un cierto criterio para la evaluación de su calidad $f : M \rightarrow \mathbb{R}$ (función objetivo) sea maximizado (o minimizado).

$$f(\vec{x}) \rightarrow \max \quad (2.1.1)$$

La función objetivo se puede definir para sistemas reales de complejidad arbitraria. La solución al problema global de optimización requiere el encontrar al vector \vec{x}^* tal que $\forall \vec{x} \in M : f(\vec{x}) \leq f(\vec{x}^*) = f^*$. Características como *multimodalidad*, i.e. la existencia de varios óptimos locales \vec{x}^j con

$$\exists \epsilon > 0 : \forall \vec{x} \in M : \rho(\vec{x}, \vec{x}^j) < \epsilon \Rightarrow f(\vec{x}) \leq f(\vec{x}^j) \quad (2.1.2)$$

donde ρ denota la distancia en M , *restricciones* sobre M por funciones $g_j : M \rightarrow \mathbb{R}$ tal que el conjunto de soluciones *factibles* $F \subseteq M$ es sólo un conjunto del dominio de las variables

$$F = \{\vec{x} \in M | g_j(\vec{x}) \geq 0 \forall j\} \quad (2.1.3)$$

y otros factores como la alta dimensionalidad, no linealidad, no diferenciabilidad, y funciones objetivo dependientes del tiempo llevan con frecuencia a tareas de optimización difíciles sino imposibles de resolver [40]. Pero aún en el último la caso la mejora de una nueva solución es un gran éxito para problemas prácticos y en muchos casos los algoritmos evolutivos proveen de un método eficiente para lograr esto.

La primera decisión a tomar es como representar una solución. En algoritmos evolutivos las soluciones se representan por genotipos, genomas o cromosomas. Una

vez que la solución se ha establecido, se puede establecer la calidad de la solución candidato basada en el problema a resolver. La representación permite codificar el problema. La calidad de una instancia puede ser medida mediante algún proceso físico, por la evaluación de una función o por algún juicio subjetivo.

Usualmente los algoritmos evolutivos trabajan con una población de soluciones, con el fin de permitir un proceso de búsqueda paralela. De hecho, la selección correcta del tamaño de la población es decisiva. El tamaño de la población es generalmente un parámetro importante de los algoritmos evolutivos.

Una vez elegida la representación y evaluados los individuos de la población corresponde a los operadores el generar variaciones de dicha solución.

Los operadores de innovación aseguran que se consideren nuevos aspectos del problema. En términos del problema de optimización mencionado anteriormente, esto significa que la prueba de nuevos parámetros en una o más de las posiciones de estos. El operador de innovación es normalmente llamado mutación, y viene con tres parámetros que determinan:

- su fuerza dentro del componente de una solución,
- su aplicación simultánea a los componentes dentro de la solución,
- su frecuencia de aplicación dentro del algoritmo.

Un fuerte operador de mutación puede generar un cambio aleatorio en cualquier posición dentro de la solución. Si se aplica a todas las posiciones dentro de la solución, con $p = 1/2$, generaría una solución completamente no correlacionada con la original.

Los operadores de conservación son empleados para consolidar lo que ya ha sido “aprendido” por los individuos en la población, permitiendo que aquellos individuos

con la mayor aptitud aparezcan un mayor número de veces en la siguiente generación, por lo que en principio la aptitud promedio generacional tiende a aumentar conforme pasan las generaciones. La recombinación de dos o más soluciones, es decir la mezcla de los elementos constitutivos de los individuos para formar nuevos, en unión con la selección ayudan a lograr esto; es importante mencionar que aún cuando los elementos constitutivos se conservan esto puede crear un nuevo individuo con características muy diferentes a las de sus padres pero esperando que las piezas útiles de información de cada individuo pueden mezclarse para obtener mejores individuos, acelerando la búsqueda de la solución óptima.

Hay varias maneras de lograr la mezcla de soluciones. Los métodos de recombinación tradicionales son de un punto, dos puntos o n -puntos de entrecruzamiento para parámetros dinámicos entre dos o más individuos. Estos operadores dan un conjunto de parámetros que controla:

- tipo de recombinación
- frecuencia de aplicación en el algoritmo.

Con la generación de variantes genotípicas se debería esperar la aparición de diferentes comportamientos en el fenotipo. Como el trabajo de Friedberg ha demostrado, esto no necesariamente es cierto. Por lo tanto un aspecto importante de los algoritmos evolutivos es la existencia de una función de aptitud que distinga entre soluciones.

Es en estos diferenciales que la selección funciona. Debido al tamaño del espacio de estados de un problema combinatorial típico no todas las variantes generadas por los métodos mencionados pueden ser almacenadas. Esto obliga a seleccionar de entre las mejores soluciones aquellas que se incluirán en la siguiente generación y aquellas que serán reemplazadas.

A continuación el modelo de un algoritmo evolutivo:

Algoritmo evolutivo

$t := 0$

inicializar $P(t)$

evaluar $P(t)$

mientras no terminación hacer

$P'(t) := \text{variación } [P(t)]$

evaluar $[P'(t)]$

$P(t+1) := \text{selección } [P'(t) \cup Q]$

$t := t+1$

continua

En este algoritmo $P(t)$ denota a la población de μ individuos en la generación t . Q es el conjunto de individuos que pueden ser considerados para la selección $Q = P(t)$. Una población de descendencia $P'(t)$ de tamaño λ es generada por medio de los operadores de variación como mutación o recombinación de la población $P(t)$. Los individuos hijos son evaluados al calcular el valor de la función objetivo $f(\vec{x}_k)$ para cada una de las soluciones \vec{x}_k representadas por individuos en $P'(t)$, y la selección se basa en los valores de aptitud para producir mejores soluciones. Desde el punto de vista de los sistemas dinámicos los operadores de un algoritmo evolutivo desestabilizan la población y el operador de selección la estabiliza. Entonces si se busca una buena solución para un problema de optimización las buenas soluciones tenderán a ser estables bajo selección a diferencia de las malas. El elegir una representación y un conjunto de operadores apropiados es con frecuencia debido a la experiencia y a la intuición.

2.1.1 Algoritmos genéticos

Son uno de los más conocidos algoritmos evolutivos, desarrollado por Holland, sus estudiantes y colegas en la Universidad de Michigan. Es un método inspirado en la evolución poblacional, cuyos individuos representan soluciones posibles a un problema determinado. Este método emplea operadores que permiten diversificar y desarrollar a los individuos en la población. La reproducción de los individuos es proporcional a la aptitud de cada uno de ellos, dicha aptitud es medida de acuerdo a una función previamente establecida (normalmente la función a optimizar), este es uno de los métodos de reproducción más usados existen otros como el torneo y el de valor esperado los cuales se explican más adelante.

Es importante hacer notar que los algoritmos genéticos fueron *inspirados* por la teoría de evolución natural pero de ninguna manera tratan de *duplicar* la evolución biológica. El vocabulario de los algoritmos genéticos es “prestado” de la biología por lo que se llama cromosomas o cuerdas al conjunto de símbolos que codifican una solución al problema a optimizar, cada uno de estas cuerdas esta formada por unidades individuales llamadas genes o bits y los valores que pueden tomar cada una de estos genes se les denomina alelos. Un conjunto de cuerdas forman una población.

Codificación

En los algoritmos genéticos tradicionales, se hace una representación de los individuos mediante cuerdas de tamaño fijo l , con un alfabeto binario $\{0,1\}$, tal que se pueden aplicar a problemas de optimización de la forma

$$f : \{0, 1\}^l \rightarrow R \quad (2.1.4)$$

Considerando la representación binaria, los algoritmos genéticos con frecuencia utilizan funciones de codificación y decodificación $h : M \rightarrow \{0, 1\}^l$ y $h' : \{0, 1\}^l \rightarrow M$ que facilitan el mapeo de soluciones $x \in M$ a cuerdas binarias $h(x) \in \{0, 1\}^l$ y viceversa. Para el caso de problemas de optimización de parámetros continuos, los algoritmos genéticos generalmente representan un vector real $x \in R$ por una cuerda binaria $y \in \{0, 1\}^l$ como sigue: la cuerda binaria se divide en n segmentos de igual longitud l' (i.e. $l = n \cdot l'$), cada segmento se codifica a su correspondiente valor entero y el valor es mapeado linealmente al intervalo $[u_i, v_i] \subset R$ (correspondiente con el i -ésimo segmento de la cuerda binaria) de valores reales.

Sin embargo, dado que muchos de los problemas reales no pueden ser codificados de esta manera se han desarrollado diversas maneras para codificar estos como para problemas combinatorios donde la codificación se basa en el conjunto de permutaciones de los procesos a optimizar, siendo el ejemplo típico el del agente viajero (búsqueda de la ruta más corta para visitar un cierto número de ciudades), considere que se tienen las ciudades 1,2,3 los individuos del algoritmo genético serían 123 (indicando que se visita primero la ciudad 1, después la ciudad 2 y por último la ciudad 3). Por eso es necesario para el adecuado desempeño de los algoritmos genéticos inteligentes que cada genotipo represente una solución potencial al problema; el proceso de evolución se aplica sobre una población de cromosomas o individuos permitiendo la búsqueda en el espacio de soluciones posibles. Los algoritmos genéticos ejecutan una búsqueda multi-direccional, manteniendo una población de soluciones potenciales que se mejoran de generación en generación. El término generación se entiende como la iteración del algoritmo en donde se realizan las operaciones genéticas que producen una nueva población, la cual se espera tenga una mayor cantidad de soluciones "buenas"

que en la población de la generación anterior.

Operadores genéticos

La mutación en algoritmos genéticos fue introducida como “operador de fondo” de poca importancia (ver [41], pp 109-111). La mutación trabaja invirtiendo el valor de los bits con una probabilidad baja $p_m=0.001$ [42], $p_m \in [0.005,0.01]$ [43], o $p_m = 1/l$ (donde l es la longitud del cromosoma o cuerda) [44]; en caso de que se tenga un alfabeto ternario donde los alelos son 0,1 y 2 la mutación es el cambio a uno de los otros estados posibles con probabilidad $1/3$. Estudios recientes han confirmado que tasas de mutación altas que disminuyen conforme el paso del tiempo tienden a mejorar la convergencia y velocidad del algoritmo genético [45]. El cambio del alelo en un bit para codificación binaria i.e. modificación en el bit de 0 a 1 o viceversa es un caso para la mutación binaria cuando se tiene otras codificaciones el operador funciona en forma diferente por ejemplo al trabajar con un cromosoma que codifica el orden de visita para el problema del agente viajero (123 indicando primero visita ciudad 1 después la ciudad 2 y por último la ciudad 3) el operador de mutación puede ser la inversión de alguno de los elementos (123 muta a 132 invirtiendo el orden entre 2 y 3).

El algoritmo genético estándar emplea el llamado entrecruzamiento en un punto, donde dos individuos de la población son elegidos al azar, se elige una posición dentro del cromosoma como punto de entrecruzamiento y la descendencia es generada concatenando la subcuerda izquierda con la subcuerda derecha de cada uno de los padres. Existen numerosas variantes de este operador. La recombinación en algoritmos genéticos es aplicada con probabilidad p_c , con valores recomendados $p_c=0.6$ [41] y $p_c \in [0.75,0.95]$ [43]. Normalmente la cruce se maneja como un porcentaje que

indica la frecuencia con la que se efectuará, lo que implica que no todas las parejas se entrecruzarán permitiendo que algunas de ellas pasen intactas a la siguiente generación.

En algoritmos genéticos, la selección es típicamente implementado como un operador probabilístico, usando una aptitud relativa $p(c_i) = f(c_i) / \sum_{j=1}^n f(c_j)$ para determinar la probabilidad de selección de un individuo c_i . Este método requiere valores positivos de aptitud y una tarea de maximización, pero se usan funciones de escala para transformar los valores, este método se conoce como "rueda de ruleta" la que consiste en crear una ruleta en la que cada cromosoma tiene asignada una fracción proporcional a su aptitud. Esta ruleta se gira (eligiendo un número al azar y sumar la aptitud de cada uno de los cromosomas hasta que esta sea mayor que dicho número el cromosoma seleccionado es el último elegido) un número de veces igual a la cantidad de elementos de la población inicial para determinar los individuos de la nueva generación, ya que a los individuos más aptos se les asigna un área mayor de la ruleta, se espera que sean seleccionados más veces que los menos aptos. La selección por torneo [46] funciona tomando una muestra aleatoria de $q > 1$ individuos en la población y seleccionando al mejor de esos q individuos para que aparezca en la siguiente generación, lo cual genera una población que es de tamaño P/q (donde P es el tamaño de la población), dado lo anterior, el proceso se realiza q veces. Otro método de selección es denominado de valor esperado que considera para cada cromosoma la aptitud de este dividido entre la aptitud promedio, de esta forma se genera un número de copias igual a la parte entera de este número, aceptándose una copia extra con una probabilidad igual a la parte fraccionaria de dicho número.

Funcionamiento de los algoritmos genéticos

La aplicación más común de los algoritmos genéticos ha sido la solución de problemas de optimización en espacios de búsqueda discretos. A continuación algunas consideraciones que se deben tomar en cuenta para el diseño y uso de los algoritmos genéticos.

1. Debe ser posible la definición de una función de aptitud sobre el conjunto de cromosomas que permita medir la calidad de las soluciones proporcionadas por los individuos de la población.
2. Se requiere de una adecuada codificación de las soluciones como conjuntos de genes que formen a cada cromosoma de la población.
3. Los operadores genéticos a utilizar dependerán de la codificación utilizada.

La función de aptitud, para el caso de un problema de optimización, es la función objetivo a maximizar (minimizar). Por su naturaleza, el algoritmo genético maximiza, pero la minimización puede realizarse empleando el recíproco de la función a maximizar (siempre y cuando este no genere una división por cero), por ejemplo. La codificación más común es el empleo de cadenas binarias (propuesto originalmente por Holland,[41]), aunque se pueden emplear números reales y letras, el uso de cromosomas con alelos binarios resulta sencillo de implementar.

Resumiendo un algoritmo genético para su funcionamiento requiere de:

- Codificación, representación de las soluciones al problema de manera que permitan la aplicación de un algoritmo genético. La representación tradicional es la codificación binaria. Para algoritmos genéticos inteligentes se emplean una

codificación que proporcione un mapeo genotipo-fenotipo no trivial, como se verá en el siguiente capítulo.

- Tamaño de la población. El cual dependerá entre otros factores de la tamaño del espacio de búsqueda.
- Población inicial. Puede ser creada de forma aleatoria (si no se tiene ninguna intuición respecto a las posibles soluciones del problema) o determinista (al incorporar información que facilite la búsqueda).
- Operadores genéticos. Diseño de los operadores que se emplearán, depende básicamente de la codificación a emplear.
- Probabilidad de los operadores. Frecuencia con la que se aplicarán estos operadores.
- Función de aptitud. Diseño de función que evaluará la calidad de cada uno de los individuos en la población, para problemas de investigación de operaciones puede ser la función objetivo a optimizar.
- Criterio de paro. Finalización del algoritmo, puede ser por el número de generaciones, convergencia de la población hacia un individuo en particular, estabilidad de la población por un período determinado entre otros.

Considerando estos elementos, la operación de un algoritmo genético tradicional consiste en:

1. Generar aleatoriamente una población inicial, constituida por un conjunto de cromosomas que representan diversas soluciones factibles de un problema de

optimización, a cada uno de estos cromosomas se le aplica la función de aptitud para saber que tan buena es esta solución.

2. Seleccionar a los cromosomas con mejor desempeño para que permanezcan para la siguiente generación.
3. Entrecruzamiento El algoritmo de cruza para dos individuos (individuo1 e individuo2) seleccionados con probabilidad de p_c de entre una población de N individuos con cromosomas de n bits es:

Algoritmo para cruza

selecciona punto de cruza $ppc = \text{Rnd} * n$

para $l = ppc$ hasta n

$\text{temporal}(l-ppc) = \text{individuo1}(l)$

$\text{individuo1}(l) = \text{individuo2}(l)$

$\text{individuo2}(l) = \text{temporal}(l-ppc)$

fin para

Por ejemplo, considerando un alfabeto binario y cuyos individuos a cruzarse sean:

0	0	1	1	0
1	1	0	1	1

haciendo el entrecruzamiento en el punto 2 se tiene:

1	1	1	1	0
0	0	0	1	1

4. Mutación: El algoritmo de mutación para un individuo (*individuo1*) con probabilidad de p_m con n bits en alfabeto binario (0,1) es:

Algoritmo para mutación

para $l=0$ hasta $n-1$

 si $Rnd < p_m$ $individuo1(l) = \text{complemento}(individuo1(l))$

fin para

donde $\text{complemento}(0)=1$, $\text{complemento}(1)=0$.

Por ejemplo: Sea el individuo:

1	1	1	1	0
---	---	---	---	---

y aplicando mutación con probabilidad 1 se tiene:

0	0	0	0	1
---	---	---	---	---

La utilidad de los operadores de entrecruzamiento y mutación es la de permitir al algoritmo genético la búsqueda de nuevas soluciones y conservación de las mejores soluciones parciales encontradas.

Muchos de los algoritmos evolutivos sólo emplean la selección y mutación como el caso de la programación evolutiva [47], pero existe suficiente evidencia para confirmar que los algoritmos genéticos son mejores dado que también emplean al operador de cruza [48].

Algunas de las ventajas y desventajas de los algoritmos genéticos.

No se requiere de conocimientos específicos sobre el problema a resolver.

Los algoritmos genéticos operan de forma simultánea con varias soluciones, en vez de trabajar de forma secuencial como otras técnicas (recocido simulado o búsqueda tabú).

Cuando se emplean para problemas de optimización, resultan menos afectados por los óptimos locales a diferencia de técnicas como el método de gradiente.

La eficiencia del algoritmo depende en cierta medida de los parámetros que se utilicen (tamaño de la población, número de generaciones, etc).

La posible complejidad en la codificación de las soluciones en términos de cromosomas.

Como se ha mencionado antes, uno de los problemas de los algoritmos genéticos radica en la codificación empleada al tratar de solucionar un problema, posiblemente por esto es que en la naturaleza se empleen procesos de decodificación complejos los cuales permiten trasladar el genotipo en fenotipo, a estos procesos se pueden identificar como “intérpretes genéticos”, un ejemplo del ADN, mencionado en el capítulo anterior. Un modelo de intérprete genético artificial es el autómata celular aleatorio $k=2$, debido a Kauffman [81]. En este modelo los alelos están representados por unidades binarias, las cuales son 1 si el alelo se expresa y 0 en caso contrario, además cada alelo recibe señales de otros dos alelos los que pueden estar en cualquier parte del cromosoma y con una regla booleana determinan en valor para el siguiente paso. La actualización se realiza en forma paralela y en caso de que se alcance un punto fijo después de un cierto número de iteraciones entonces se le asigna al individuo (cromosoma) un aptitud que depende del valor del conjunto de unidades binarias en el punto fijo, teniendo una degeneración en el sentido de que muchas cuerdas pueden

llegar al mismo punto fijo.

Es importante hacer explícita la diferencia entre “aptitud interna” (online performance) y “aptitud externa” (offline performance) siendo la primera relacionada con el valor promedio en una generación dada de la aptitud de los individuos, mientras que la “aptitud externa” se refiere a la búsqueda de las mejores soluciones. Como se verá más adelante existen otras medidas para medir el desempeño de un algoritmo genético.

2.1.2 Estrategias evolutivas

Fueron desarrolladas en la década de los 60s, los pioneros en este campo fueron Rechenberg y Schwefel. Su trabajo inicial fue con problemas de hidrodinámica donde usaron la idea de usar elementos al azar para decidir un proceso de optimización. Así las mutaciones discretas fueron la primera variación evolutiva a ser aplicada dentro de las estrategias evolutivas. Debido a las limitaciones de la configuración básica del experimento, solo se podía considerar un objeto a la vez, por lo que la población era de tamaño uno. Pero el proceso de selección mantenía la historia de la aptitud de una configuración experimental y sus variaciones debidas a las mutaciones aleatorias aplicadas.

Con el desarrollo de las computadoras fue posible aplicar las estrategias evolutivas a las variables continuas al representar a los individuos como vectores de variables reales donde la mutación era mediante la adición de números normalmente distribuidos con media 0. En este enfoque las pequeñas variaciones son mucho más frecuentes que las grandes. De hecho es destacado el énfasis hecho sobre la causalidad, i.e. el hecho que grandes mutaciones permita grandes cambios en la aptitud, y mutaciones pequeñas logran pequeñas modificaciones.

Desde hace algunos años en dentro de las estrategias evolutivas ya se emplean poblaciones de individuos permitiendo el uso de diferentes operadores de selección.

Operadores genéticos

En estrategias evolutivas la selección es un operador determinístico el cual elige los $\mu < \lambda$ para conformar la población en la siguiente generación. Las estrategias evolutivas (μ, λ) usan un esquema de selección determinista. La notación (μ, λ) indica que μ padres crean $\lambda > \mu$ descendientes mediante la recombinación y la mutación y los mejores μ individuos creados son determinísticamente seleccionados para reemplazar a sus padres. Este algoritmo permite que el mejor miembro de la generación $t + 1$ pueda desempeñarse pero que el mejor individuo en la generación t , i.e. este método es no elitista permitiendo que la estrategia ayude a escapar en caso de estar en un óptimo local. En contraste la estrategia $(\mu + \lambda)$ selecciona los μ sobrevivientes de la unión de padres y descendientes, de manera que se garantiza un curso de evolución monótono.

En estrategias evolutivas, los individuos consisten de variables $x_i \in R$, ($1 \leq i \leq n$). La mutación se realiza independientemente en cada elemento del vector sumando una valor aleatorio normalmente distribuido con valor esperado 0 y desviación estándar σ , así

$$x'_i = x_i + \sigma \cdot N_i(0, 1) \quad (2.1.5)$$

En las estrategias evolutivas la recombinación es incorporada en el loop principal del modelo de algoritmo evolutivo, mencionado anteriormente, como primer operador y genera una nueva población intermedia de λ individuos, creando de un individuo por aplicación hasta ρ ($1 \leq \rho \leq \mu$) individuos. Normalmente se elige $\rho=2$ o $\rho = \mu$. Los

tipos de recombinación en estrategias evolutivas con frecuencia difieren unos de otros, ejemplos típicos son la recombinación discreta (elección aleatoria de variables para el número de padres, en comparación con el entrecruzamiento uniforme en algoritmos genéticos) y la recombinación intermedia, para mayor información de estos detalles puede consultar [13].

2.1.3 Programación evolutiva

Creada a principios de los 60s por Fogel, Owens y Walsh [49], [50]. La programación evolutiva usa el operador de mutación para cambiar máquinas de estado finito o autómatas. Un autómata es un programa de computación muy sencillo que consiste de una máquina moviéndose en una gráfica de nodos llamados estados. El autómata de estado tiene muchas similitudes con una máquina de Turing. La programación evolutiva emplea creación aleatoria, mutación y reproducción basada en la aptitud.

En su forma original, la programación evolutiva fue usada para resolver problemas de predicción con la ayuda de autómatas. La mutación sobre las máquinas de estado finito (representadas por tablas de transición y estados iniciales) se realiza al permitir los cambios en el número de estados, en el estado inicial, el estado de transición o el símbolo de salida. Como se puede ver la programación dinámica permite la representación simbólica de programas de computadora formulados como autómatas.

Después de su inicialización, todos los P individuos se seleccionan como padres, y son mutados, produciendo P hijos. Estos hijos son evaluados y se eligen P sobrevivientes de los $2P$ individuos, usando selección del tipo torneo. El mejor individuo siempre sobrevive.

Los fundamentos teóricos de este tipo de algoritmo se basan en la prueba de la

convergencia global (con probabilidad 1) para la programación evolutiva. Este resultado se deriva de definir una cadena de Markov sobre el espacio discreto de estados que es obtenido usando los números representados en una computadora digital.

Operadores genéticos

Para la programación evolutiva cada solución de descendientes y padres es comparada contra $q > 1$ otras soluciones elegidas al azar. Para cada comparación se marca un punto a favor si es mayor que la selección de comparación y los individuos con mayor puntuación se mantienen dentro de la población para la siguiente generación. Este es una versión probabilística de la selección $(\mu + \lambda)$ que se vuelve más determinista conforme q aumenta.

Originalmente, la mutación en programación evolutiva fue implementada como un cambio aleatorio (o múltiples cambios) en la descripción de las maquinas de estado finito de acuerdo a cinco diferentes modificaciones: cambio en un símbolo de salida, cambio en un estado de transición, estado de borrado o cambio en el estado inicial. Estas mutaciones se realizaban con probabilidad uniforme y el número de mutaciones para una población era previamente fijado o de acuerdo a una distribución de probabilidad. Normalmente el esquema de mutación más frecuente es muy similar al de las estrategias evolutivas.

2.2 Otras variantes de algoritmos evolutivos

2.2.1 Programación genética

Formalizada por John Koza [51], en su libro "Genetic Programming. On the Programming of Computers by Means of Natural Selection" en 1992, aplica la búsqueda

evolutiva en el espacio de estructuras de árbol las cuales pueden ser interpretadas como programas de computadora en un lenguaje propenso a modificación por mutación o recombinación. El enfoque dominante en programación dinámica usa programas en LISP como espacio de genotipos.

2.2.2 Sistemas clasificadores

Es un algoritmo evolutivo para buscar en el espacio de las *reglas de producción* (con frecuencia codificadas por cromosomas sobre un alfabeto terciario, pero algunas veces usando reglas simbólicas) de un sistema de aprendizaje capaz de inducción y generalización [46],[52]. Típicamente el enfoque *Michigan* y el enfoque *Pittsburgh* son distinguidos de acuerdo a si el individuo corresponde con una regla sencilla del sistema de reglas (Michigan) o con la base de reglas completa (Pittsburgh).

Un sistema clasificador tiene tres componentes: en lenguaje de programación basado en reglas, una economía de mercado simulada y un algoritmo genético. Las reglas son los individuos en el algoritmo genético. Juntas, las reglas pueden ser vistas como un programa realizando una tarea. Cuando ciertas reglas se disparan debido a alguna entrada el sistema genera una salida. Cualquier "recompensa" por la salida acredita una aptitud proporcional a las reglas que contribuyeron. El algoritmo genético opera sobre las reglas. Las reglas resultantes del algoritmo genético construyen el siguiente sistema clasificador modificado. La metáfora es prestada de la economía de mercado donde muchos individuos *cooperan* y *compiten* para lograr una alta eficiencia en alcanzar una meta y cada individuo es recompensado por su aportación al éxito.

Un sistema clasificador no se considera un programa evolutivo dado que el programa completo no evoluciona en los individuos de la población. Las reglas individuales en el sistema clasificador no son capaces de resolver la tarea por ellas mismas. Sin

embargo en 1980, Smith [53] inventó una variante de un sistema clasificador introduciendo cuerdas de tamaño variable como individuos. En este enfoque un individuo es un programa de reglas completo que puede resolver la tarea definido por la función de aptitud.

Capítulo 3

Teorías de algoritmos genéticos

En este capítulo se hace una revisión de las teorías más importantes que tratan de explicar el funcionamiento de los algoritmos genéticos. Las teorías tratadas son el teorema de esquemas y la hipótesis de bloques constructores, que establecen el algoritmo genético evoluciona haciendo uso de pequeños bloques con aptitud elevada; la explicación por cadenas de Markov la cual provee el análisis microscópico de algoritmos genéticos, el enfoque de estrategias evolutivas que sugiere que el funcionamiento de los algoritmos genético se debe a tres principios básicos y por último el enfoque de grados de libertad efectivos el que proporciona una ecuación exacta para el comportamiento de los algoritmos genéticos además de demostrar el efecto constructivo de los operadores genéticos.

3.1 Teorema de esquemas, hipótesis de bloques constructores

Uno de los teoremas fundamentales de la teoría actual de algoritmos genéticos es Debido a Holland [41], en el que manifiesta que el desarrollo de una población en un algoritmo genético se debe al aumento generación tras generación de pequeños

conjuntos de genes con aptitud elevada.

Sin pérdida de generalidad se puede considerar una cuerda basada en un alfabeto binario $V = 0, 1$, a las cuerdas se refiere por una letra mayúscula y al elemento de la cuerda por una letra minúscula. Por ejemplo la cuerda de 8 elementos $A=00011010$ se puede representar simbólicamente por: $A = a_1a_2a_3a_4a_5a_6a_7a_8$ donde cada elemento a_i representa una gen, los alelos posibles son 0 ó 1. La “población” la cual es dependiente del tiempo, se representa por $A(t)$.

Un elemento importante es el llamado “esquema” el cual representa a un subconjunto de cuerdas con elementos en común dentro de una población, para identificarlos es útil el aumentar el símbolo * al alfabeto mencionado anteriormente, con esta modificación el alfabeto será: $V = 0, 1, *$. Este nuevo símbolo hace las funciones de “comodín” representando a todo el resto de elementos del lenguaje en esta posición. Por ejemplo, para una cuerda de 3 genes o “bits” el esquema “0*1” representa a las cuerdas: 001 y 011. En general, para alfabetos con cardinalidad k existen $(k + 1)^l$ esquemas, donde l es el tamaño de la cuerda. En una población de cuerdas con n miembros hay a lo más $n \cdot 2^l$ esquemas ya que cada cuerda por sí misma es representativa de 2^l esquemas.

Sin embargo no todos los esquemas son creados iguales, para identificar esto es necesario definir dos parámetros más: el *orden* del esquema ($o(\xi)$ =orden del esquema ξ) el cual representa el número de bits definidos (diferentes de *) dentro de la cuerda; así el esquema **00*1 es de orden 3, simbólicamente $o(* * 00 * 1) = 3$. El segundo parámetro a definir es la “longitud” del esquema (simbólicamente δ) la que representa la distancia entre la posición del último y primer bits definidos; $\delta(*00 * *0) = 4$, $\delta(0 * * * *) = 0$.

Considerando todos los conceptos anteriores, Holland establece que el número esperado de copias de un determinado esquema ξ para la siguiente generación en un algoritmo genético donde están implementados los operadores de reproducción, mutación y cruza es:

$$P(\xi, t+1) \geq \frac{f(\xi, t)}{\bar{f}(t)} P(\xi, t) \left(1 - p_c \left(\frac{l-1}{N-1} \right) - N_2 p \right), \quad (3.1.1)$$

El factor común de la ecuación se refiere al número de esquemas para la siguiente generación producto del efecto de reproducción. El cociente refleja que los esquemas con aptitud por encima de la aptitud promedio aumentarán en número en la próxima generación. El término entre corchetes representa el efecto de los operadores de mutación y cruza. Es posible calcular el límite inferior para la probabilidad de sobrevivencia para cualquier esquema, debido a que el esquema sobrevive cuando la cruza sucede fuera del segmento definido por la longitud del esquema se tiene: $p_s = 1 - \frac{\delta(\xi)}{l-1}$ y si la cruza se lleva a cabo de manera aleatoria con una tasa p_c se obtendría:

$$p_s \geq p_c \cdot \frac{\delta\xi}{l-1} \quad (3.1.2)$$

Para el caso del operador de mutación, el cual se refiere al cambio de alelo para un gen específico con probabilidad p_m , el alelo sobrevive con probabilidad $(1 - p_m)$ y dado que cada una de las mutaciones es estadísticamente independiente, un esquema particular sobrevive cuando cada uno de los $o(\xi)$ posiciones fijas en el esquema sobreviven. Así la probabilidad de sobrevivencia por mutación es $(1 - p_m)^{o(\xi)}$ en el caso de valores pequeños de p_m ($p_m \leq 1$) la ecuación es: $1 - o(\xi) \cdot p_m$.

La ecuación 3.1.1 lleva a la hipótesis de bloques constructores introducida por

Golberg [46]. “*un algoritmo genético llega al desempeño óptimo a través de la yuxtaposición de esquemas o bloques constructores pequeños, de orden bajo y con alto desempeño*”. Es importante mencionar que en esta ecuación no se considera el efecto constructivo de los operadores de cruce y mutación, dichos efectos si son considerados por la teoría de Stephens-Waelbroeck, la cual será explicada más adelante.

3.2 Análisis por cadenas de Markov

El análisis por cadenas de Markov provee un modelo microscópico exacto para los algoritmos genéticos. Este tipo de análisis hasta la fecha se ha concentrado principalmente en la convergencia de las propiedades (i.e. para $g \rightarrow \infty$, siendo g el número de generaciones del algoritmo genético). Existen varias investigaciones sobre el comportamiento transitorio de los algoritmos [54] pero sin profundizar mucho. En términos de optimización, el interés principal es la tasa de convergencia (y el orden) a un punto óptimo más que a un estado estable en la distribución de probabilidades, i.e. la búsqueda del óptimo global. Un resultado exacto a partir del análisis por cadenas de Markov es que el algoritmo genético propuesto por Holland [41], no converge a un óptimo local o global (para poblaciones finitas) ([55],[56]). Esto puede ser fácilmente explicado si se toma en cuenta que en este tipo de algoritmo genético los individuos de la población se reproducen de manera proporcional a su aptitud y por lo tanto siempre hay una probabilidad no cero para perder cualquier solución encontrada hasta ese momento. La convergencia, sin embargo, puede obtenerse si se usa algún tipo de elitismo (conservar la mejor solución) dentro del algoritmo ([57],[56]). Pero esto todavía deja la pregunta de qué tan rápido el algoritmo alcanza el óptimo. Aunque el análisis por cadenas de Markov ha mostrado que en términos de comportamiento

asintótico el entrecruzamiento no proporciona ninguna ventaja al algoritmo genético (de acuerdo a la que se menciona en estas teorías ya que como se demostrará más adelante esto no es del todo cierto) [55], no se ha tenido éxito al usar este análisis para determinar las tasas para garantizar la convergencia. Este es un requisito si se desea comparar la influencia de los operadores en el desempeño del algoritmo.

La dinámica estadística completa (i.e. la evolución temporal) de la cadena de Markov contiene toda la información del sistema de los algoritmos evolutivos. Desafortunadamente, el tamaño de la matriz de transición crece exponencialmente en función del tamaño del cromosoma. Por lo que sólo hay soluciones disponibles para modelos de "juguete" i.e. que trabajan a una escala pequeña, además de que no se pueden emplear aproximaciones sucesivas.

En [58] se establece que para un modelo de población finita: sea Z una matriz de $N \times R$ donde N es el número de cuerdas posibles y R el número de poblaciones posibles, cuyas columnas representan las posibles poblaciones de tamaño r . La i -ésima columna $\phi_i = \langle Z_{0,i}, \dots, Z_{N-1,i} \rangle^T$ de Z es el vector de incidencia

$Z_{y,i}$ = número de ocurrencias de la cuerda y en la i -ésima población

Dada una enumeración de las poblaciones, Z es bien definida.

Un *modelo exacto* de un algoritmo genético sencillo esta representado por la siguiente cadena de Markov. Los estados estan dados por las índices de las columnas de la matriz Z y las probabilidades de transición estan dadas por la matriz Q donde

$$Q_{i,j} = r! \prod_{y=0}^{N-1} \frac{\{\zeta(\phi_i)_y\}^{Z_{y,j}}}{Z_{i,j}!}$$

donde ζ es el operador de transición de una población a la siguiente.

Note que los estados de esta cadena de Markov pueden ser vistos como punto en el simplex $\Lambda = \{x \in \mathbb{R}^N : x \text{ no negativo y } |x| = 1\}$

mediante la correspondencia $j \longleftrightarrow \phi_i/\tau$.

El derivar una formulación analítica para la dinámica del valor medio y la evolución temporal de las cadenas microscópicas de Markov es un problema abierto hasta la fecha. Dada la complejidad del análisis microscópico fue necesario un cambio a una escala mayor para dar paso al análisis mesoscópico basado en la teoría microscópica pero usando algunas veces “ingredientes” fenomenológicos obtenidos por métodos empíricos. Este siguiente paso consiste en la agrupación en los grados de libertad al considerar los primeros momentos de la función de distribución de la probabilidad de ocupación de estado (probabilidad de encontrar un individuo en cierto estado), Vose y Liepnins (1991) [59], introducen una formulación matemática del algoritmo genético que aproxima la población mediante dicha distribución de probabilidad. Este modelo ha sido usado para investigar ciertas propiedades dinámicas de los algoritmos genéticos estándar (e.g. las rápidas fluctuaciones en la población que pueden ser debidas a la idea de equilibrio puntual). En 1995 Vose y Wright [60], proporcionaron los primeros resultados concernientes a la dinámica del valor esperado de la aptitud poblacional para una clase de funciones. Estos resultados pueden ser usados como aproximaciones para poblaciones finitas.

Otro enfoque que emplea la dinámica del valor medio y la varianza para tratar de explicar el funcionamiento de los algoritmos genéticos emplea la mecánica estadística, este método es propuesto en los trabajos de Shapiro y Prügel [61], [62], escogiendo un conjunto pequeño de cantidades macroscópicas que se asume caracterizan al sistema. Considerando que los algoritmos genéticos son sistemas dinámicos la mecánica estadística ayuda a calcular las propiedades estadísticas de la población que cambia en el tiempo. Las cantidades macroscópicas empleadas son los momentos de la

distribución de aptitud en la población (media y varianza).

3.3 Enfoque de estrategias evolutivas

Debido a Beyer [63], a partir de resultados en estrategias evolutivas establece tres principios que permiten entender el funcionamiento de un algoritmo genético:

- principio de progreso evolutivo (EPP)
- mutación inducida por recombinación (MISR)
- hipótesis de reparación genética (GR)

El EPP como principio básico, identifica al proceso de evolución en el algoritmo genético que proporcione mejores resultados, i.e. identifica al progreso evolutivo como la diferencia entre ganancia de progreso y pérdida del mismo. La reparación genética ayuda a la disminución en pérdidas mediante correcciones estadística de error. Entonces es posible aumentar la probabilidad de mutación con el efecto de un mayor progreso. Este enfoque establece que las mutaciones son no dirigidas, además de que una probabilidad de mutación alta incrementa tanto la aptitud poblacional como los componentes útiles de la población. La selección extrae aquellos individuos producidos por las mutaciones con los mayores componentes de aptitud y el mecanismo de reparación genética disminuye la influencia de sus partes con menor aptitud. Esto permite explicar los beneficios de la recombinación sin hacer referencia a la teoría de bloques constructores.

3.4 Grados de libertad efectivos

Los grados de libertad son las variables que se necesitan para describir el estado de un sistema, dependiendo de su escala relevante cuya definición dependerá de la aplicación que se hará del sistema, por ejemplo para describir un lápiz desplazándose por el espacio se requiere de 6 grados de libertad (3 de ellos para determinar la ubicación de su centro de gravedad en las coordenadas x , y y z , 3 más para describir los correspondientes movimientos angulares) ahora, si este lápiz se analiza a nivel microscópico se tendría que tiene una cantidad de átomos del orden de 10^{23} y cada uno de estos átomos tendrían sus correspondientes grados de libertad. Los grados de libertad por el tipo de acoplamiento se clasifican en microscópicos (con frecuencia fuertemente acoplados) y efectivos (variables en la escala relevante cuya dinámica esta determinada por acoplamientos débiles). Otros ejemplos:

- Convección de Boussinesq. Sistema con muchos grados de libertad, pero a larga escala se reduce a tres modos de Fourier, con una dinámica caótica representada por el sistema de Lorenz.
- Fluido turbulento: los grados de libertad son los vectores de velocidad en los distintos puntos del fluido. Sus interacciones no lineales están dadas por las ecuaciones de Navier-Stokes, cuando el fluido rebasa el número de Reynolds la dinámica se vuelve desordenada con fluctuaciones impredecibles en el campo de velocidad aún en una escala espacial muy fina, grados de libertad efectivos en todas las escalas.
- Condensación de copo de nieve: Proceso de aglomeración de moléculas de agua en una estructura cristalina. Hay muchas moléculas de agua y sus interacciones

son no lineales; a larga escala se desprende una estructura regular de apariencia fractal — el crecimiento de esta estructura emergente es predecible.

Esta teoría se basa en la identificación de los grados de libertad efectivos de un sistema, dichos grados de libertad se caracterizan por tener una interacción mutua débil, i.e. cierto grado de integridad en función de la “escala” temporal en que se desenvuelven, por ejemplo considera en la biología se tienen diversas escalas como la consideración de individuos que a su vez forman familias y estas forman poblaciones. Además este enfoque hace incapie en el efecto de construcción de los operadores genéticos.

3.4.1 Paisajes de aptitud

Los paisajes de aptitud, a pesar de ser un concepto difícil de comprender, principalmente debido a la dificultad de visualizarlos, ha servido como cimiento dentro de la teoría de los sistemas evolutivos y en fechas recientes dentro de los sistemas complejos.

El concepto de paisaje de aptitud fue originalmente aplicado en la biología evolutiva por Wright [2] identificando la evolución como un flujo poblacional sobre una topografía o paisaje. En [64] se puede encontrar la perspectiva histórica y en [65] un recuento reciente sobre el rol de los paisajes de aptitud en la computación evolutiva.

Dada la complejidad para el análisis de un paisaje de aptitud real hasta la fecha se ha trabajado con sistemas relativamente simples. Los paisajes del tipo NIAH (aguja en un pajar, por sus siglas en inglés) considerados por Eigen [66] han sido menospreciados por considerarseles muy simples en comparación con los modelos N_k de Kauffman los cuales identifican a un individuo exitoso al aplicar un intérprete basado en reglas booleanas, en tanto que los modelos NIAH emplean una función con

un sólo óptimo.

El análisis de paisajes de aptitud dentro de la teoría de algoritmos genéticos se ha concentrado en la relación entre la dificultad del problema y la modalidad del paisaje, asumiendo que la modalidad es proporcional a la dificultad. Sin embargo como se ha discutido previamente [67],[68] es posible encontrar varios contra-ejemplos. Por ejemplo el paisaje de NIAH es unimodal pero sin embargo es difícil de resolver para un algoritmo genético y por otro lado una función multimodal, como la función de porcupine [69] puede ser solucionada de manera sencilla. Por lo tanto la modalidad no se debe confundir con rugosidad, que se puede entender como una medida de la densidad de óptimos locales, siendo de mayor importancia el grado de correlación en el paisaje. La correlación estructural es la que reviste la mayor importancia dentro de la búsqueda además de ser una medida gruesa de la información mutua disponible entre dos puntos del paisaje. La longitud de correlación ϵ , en el paisaje es una medida característica del grado y extensión de tales correlaciones. Existen varios métodos para definir esta longitud. Uno de ellos es a través de una función de correlación de dos puntos conectados o varianza definida por $G(i, j) = \langle f(i)f(j) \rangle - \langle f(i) \rangle \langle f(j) \rangle$ donde $\langle \rangle$ representa el promedio sobre el conjunto estadístico de interés. Uno puede definir la correspondiente longitud de correlación como $\epsilon = -\ln G(i, j) / |i - j|$. Obviamente esto requiere de la existencia y definición de una métrica dentro del paisaje. Una amplia discusión sobre paisajes de aptitud puede ser encontrada en [70].

El concepto principal respecto a la distancia de correlación identifica a puntos con una distancia mayor a ϵ como no efectivamente correlacionados y en caso de ser menor como efectivamente correlacionados. En el paisaje NIAH la longitud de correlación es cero indicando que no hay ningún indicio dentro del paisaje de la existencia de

un óptimo global, la longitud cero se debe a que esta se define por $G(i, j)$ y no por $\langle f(i)f(j) \rangle$ es decir esta más relacionada con correlaciones entre fluctuaciones cerca de la media que con la persistencia.

Al identificar la evolución como un flujo poblacional es necesario identificar las características robustas del sistema, aquellas que no cambian aún cuando el ambiente en el que se desarrolla el sistema se modifique. Como anteriormente se ha comentado en el modelo de evolución, esta se ve afectada por los diferentes operadores genéticos. Un operador genético se puede definir como un operador H tal que $P(t+1) = HP(t)$, siendo P la población al tiempo t . El concepto tradicional de paisajes de aptitud se basa en la asociación de un proceso de "hill-climbing" y normalmente se vincula principalmente con la reproducción de los individuos. Sin embargo en la presencia de otros operadores genéticos el fenómeno deja de ser "hill-climbing".

Considerando cualquier nivel de agrupación se verá que se introduce una dependencia de tiempo, sea cualquier configuración microscópica, c_i que pueda ser representada por un conjunto s de N elementos $\{s_k\}, k \in [1, N]$. Tal configuración puede representar, por ejemplo, el genotipo de un organismo, o una posible solución a un problema combinatorio. Se asume que la aptitud de dicha configuración es $f(c_i)$ independiente del tiempo y se considera otra configuración, $c_j \neq c_i$, con aptitud $f(c_j)$. Ya que ambas configuraciones componen el mismo paisaje y considerando que tienen N_2 elementos en común, representados por el conjunto $\{\bar{s}\} \subset \{s_k\}$, es posible cuantificar las características en común al definir:

$$f(\xi, t) = f(c_i)P(c_i, t) + f(c_j)P(c_j, t) \quad (3.4.1)$$

donde $P(c_i, t)$ es la probabilidad de encontrar la configuración c_i al tiempo t .

En algoritmos genéticos ξ es el denominado esquema. El concepto de esquema es

muy importante ya que permite agrupar la información microscópica, de la cual es muy difícil seguir su evolución. Normalmente se buscan variables que proporcionen información más global, como por ejemplo la aptitud de alguna especie, S , la que se puede considerar como:

$$f(S, t) = \frac{\sum_{c_i \in S} f(c_i) P(c_i, t)}{\sum_{c_i \in S} P(c_i, t)} \quad (3.4.2)$$

donde la suma es sobre todos aquellos genotipos miembros de las especies. Esta agrupación respecto a variables más microscópicas introduce una dependencia temporal inducida por el acoplamiento de los grados de libertad efectivos. En general el concepto de un paisaje dinámico es de mayor relevancia que el de uno estático. La aptitud es una función dependiente del tiempo por lo que cualquier paisaje de esta función también será dependiente del tiempo.

Una población $\mathcal{M}(t) \equiv \{g(t)\} \subset G$, donde $\{g(t)\}$ es el conjunto de genotipos presentes al tiempo t , fluye de acuerdo a

$$\mathcal{M}(t + 1) = \mathcal{H}(\{f(c_i)\}, \{\mathcal{M}(t)\}, \{p_k\}, t) \mathcal{M}(t) \quad (3.4.3)$$

donde \mathcal{H} es un operador de evolución que depende del paisaje de aptitud, $\{f(c_i)\}$, la población $\{\mathcal{M}(t)\}$ y el conjunto de parámetros, $\{p_k\}$, que rigen a los otros operadores genéticos; e.g. las probabilidades de mutación y recombinación.

3.4.2 Ecuaciones de evolución para cuerdas

Ecuación considerando sólo al operador de selección

La proporción de un individuo c_i en la siguiente generación de un algoritmos genéticos esta dada por:

$$P(c_i, t + 1) = P'(c_i, t) = \frac{f_i}{\bar{f}} P(c_i, t) \quad (3.4.4)$$

Como se puede apreciar la proporción en el número de cuerdas dependerá de la relación $\frac{f_i}{\bar{f}}$ que al normalizar permite definir el parámetro $\delta f_i = \frac{f_i}{\bar{f}} - 1$ para medir la ventaja selectiva de la cuerda C_i por lo que:

- Si $\delta f_i > 0$, el número de cuerdas C_i aumenta
- Si $\delta f_i = 0$, el número de cuerdas C_i permanece constante
- Si $\delta f_i < 0$, el número de cuerdas C_i disminuye

Ecuación considerando los operadores de selección, mutación y cruza

La ecuación que representa la proporción de cuerdas del tipo c_i en la siguiente generación de un algoritmo genético bajo los operadores de selección, mutación y cruza es:

$$P(c_i, t + 1) = \mathcal{P}(c_i) P_c(c_i, t) + \sum_{c_j \neq c_i} \mathcal{P}(c_j \rightarrow c_i) P_c(c_j, t) \quad (3.4.5)$$

donde los coeficientes $\mathcal{P}(c_i)$ y $\mathcal{P}(c_j \rightarrow c_i)$ representan la probabilidad de que el genotipo c_i permanezca sin cambio y la probabilidad que el genotipo c_j mute al genotipo c_i respectivamente. $P_c(c_i, t)$ es la proporción de cuerdas c_i al tiempo t después de la selección y recombinación. Explícitamente

$$P_c(c_i, t) = (1 - p_c) P'(c_i, t) + \frac{p_c}{N-1} \sum_{k=1}^{N-1} P'(c_i^L(k), t) P'(c_i^R(k), t) \quad (3.4.6)$$

donde $P'(c_i, t) = (f(c_i, t)/\bar{f}(t))P(c_i, t)$, $\bar{f}(t)$ la aptitud promedio generacional. p_c es la probabilidad de cruza, k el punto de cruza sobre el tamaño de la cuerda. Las

cantidades $P'(c_i^L(k), t)$ y $P'(c_i^R(k), t)$ son definidas de forma análoga a $P'(c_i, t)$ pero referidas a variables acopladas, i.e. esquemas, c_i^L y c_i^R . c_i^L es de orden $N_2 = k$ con longitud $l = k$ mientras que c_i^R es de orden $N_2 = N - k$ con longitud $l = N - k$. En este caso se esta tomando la longitud como el número total de bits entre las posiciones más externas del cromosoma. Por lo tanto, $l = \delta(\xi) + 1$ donde $\delta(\xi)$ es la definición estándar. Para el caso de cuerdas completas, $l(c_i^L) = N_2(c_i^L)$ y $l(c_i^R) = N_2(c_i^R)$.

Ecuación de esquemas

A continuación la ecuación para un esquema ξ [71, 72].

$$P(\xi, t + 1) = \mathcal{P}(\xi)P_c(\xi, t) + \sum_{\xi_i} \mathcal{P}(\xi_i \rightarrow \xi)P_c(\xi_i, t) \quad (3.4.7)$$

donde

$$P_c(\xi, t) = P'(\xi, t) - \frac{p_c}{N-1} \sum_{k=1}^{l-1} (P'(\xi_L(k), t) - P'(\xi_R(k), t))P'(\xi(k), t) \quad (3.4.8)$$

y la suma en (3.4.7) es sobre todos los esquemas ξ_i que difieren en al menos un bit de ξ en uno de los N_2 bits definidos de ξ . En otras palabras, cualquier esquema compitiendo con ξ y perteneciendo a la misma partición. Todas las otras cantidades para esquemas son análogas a las definidas por las ecuaciones (3.4.5) y (3.4.6). La suma sobre los puntos de entrecruzamiento es solo sobre los puntos dentro del esquema, i.e sobre la longitud de definición l , del esquema y no sobre toda la cuerda. La ecuación (3.4.7) puede ser interpretada como un teorema de esquemas. [73].

3.4.3 Aptitud efectiva

El concepto de aptitud efectiva fue introducido primero en el contexto de programación genética [74],[75] para tomar en cuenta el efecto destructivo de la cruce y la

mutación en el contexto de evolución inequitativa, i.e. una desigualdad que da sólo un límite inferior para el número esperado de individuos de un cierto genotipo para la siguiente generación y no toma en cuenta la contribución de la creación de nuevas cuerdas. Un concepto similar, "aptitud ajustada" y el correspondiente concepto de un coeficiente de Walsh "ajustado" fue introducido en [76] y [77] en el contexto de paisajes engañosos.

La aptitud efectiva, f_{eff} , fue introducida independientemente en el contexto de los algoritmos genéticos en [71],[72] y discutida formalmente en [73],[80],[78]. La aptitud efectiva fue definida con respecto a la ecuación de evolución exacta para cuerdas 3.4.5 y/o esquemas 3.4.7 tomando en cuenta los efectos de creación de esquemas o cuerdas. A diferencia de la formulación Nordin-Bauzhaf [74],[75] donde $f > f_{eff}$ siempre en la formulación Stephens-Waelbroeck [71],[72] se presenta $f < f_{eff}$. De hecho para cada genotipo donde $f > f_{eff}$ existe otro donde $f < f_{eff}$.

Explícitamente se define la aptitud efectiva para el caso de cuerdas [71, 72, 73, 78]

$$P(c_i, t + 1) = \frac{f_{eff}(c_i, t)}{\bar{f}(t)} P(c_i, t) \quad (3.4.9)$$

y análogamente para el caso de esquemas por

$$P(\xi, t + 1) = \frac{f_{eff}(\xi, t)}{\bar{f}(t)} P(\xi, t) \quad (3.4.10)$$

Se puede entender a la aptitud efectiva como la representación del efecto de todos los operadores genéticos en un factor de selección simple. $f_{eff}(c_i, t)$ es el valor para la aptitud reproductiva al tiempo t requerido para incrementar o disminuir $P(c_i, t)$ mediante selección reproductiva pura por la misma cantidad que con todos los operadores genéticos combinados en el contexto de aptitud reproductiva $f(c_i)$. Si $f_{eff}(c_i, t) > f(c_i, t)$ el efecto de los operadores genéticos aparte de la selección es el

de aumentar la aptitud reproductiva del genotipo c_i . El efecto contrario es cierto cuando $f_{eff}(c_i, t) < f(c_i, t)$.

La forma exacta de la aptitud efectiva depende del conjunto de operadores genéticos usados. Para el caso general de la ecuación (3.4.5) se tiene

$$f_{eff}(c_i, t) = \left(\frac{\mathcal{P}(c_i)P_c(c_i, t) + \sum_{c_j \neq c_i} \mathcal{P}(c_j \rightarrow c_i)P_c(c_j, t)}{P(c_i, t)} \right) \bar{f}(t) \quad (3.4.11)$$

Note que es dependiente del tiempo y de la composición actual de la población. La aptitud efectiva se extiende naturalmente a los esquemas vía las ecuaciones (3.4.10) y (3.4.7) y toma una forma análoga a las anteriores mostrando su invariancia ante la agrupación. De forma explícita:

$$\begin{aligned} f_{eff}(\xi, t) &= \mathcal{P}(\xi) \bar{f}(\xi, t) + \sum_{\xi_i} \mathcal{P}(\xi_i \rightarrow \xi) \frac{P(\xi_i, t)}{P(\xi, t)} \bar{f}(\xi_i, t) \\ &\quad - \frac{p_c}{N-1} \mathcal{P}(\xi) \bar{f}(t) \sum_{k=1}^{l-1} \left(\frac{P'(\xi(k), t) - P'(\xi_L(k), t)P'(\xi_R(k), t)}{P(\xi, t)} \right) \\ &\quad - \frac{p_c}{N-1} \sum_{\xi_i} \mathcal{P}(\xi_i \rightarrow \xi) \bar{f}(t) \sum_{k=1}^{l-1} \left(\frac{P'(\xi_i(k), t) - P'(\xi_{iL}(k), t)P'(\xi_{iR}(k), t)}{P(\xi, t)} \right) \end{aligned} \quad (3.4.12)$$

donde $\bar{f}(\xi, t)$ es la aptitud promedio del esquema ξ . En el límite $p \rightarrow 0$, $p_c \rightarrow 0$ (o en el caso más genérico cuando la fuerza de los operadores aparte de la selección $\rightarrow 0$) $f_{eff}(\xi, t) \rightarrow \bar{f}(\xi, t)$.

Otra definición de aptitud efectiva sigue de separar los términos de la ecuación de evolución para esquemas que en aquellos que son lineales para $P(\xi, t)$ y aquellos términos independientes de ellos. Por ejemplo para el caso de selección y entrecruzamiento se tiene

$$P(\xi, t+1) = \frac{f'_{eff}(\xi, t)}{\bar{f}(t)} P(\xi, t) + j(t) \quad (3.4.13)$$

donde $f'_{eff}(\xi, t) = (1 - p_c \frac{(t-1)}{N-1}) \frac{\bar{f}(\xi, t)}{f(t)}$ and $j(t) = \frac{p_c}{N-1} \sum_{k=1}^{t-1} P'(\xi_L(k), t) P'(\xi_R(k), t)$. El coeficiente de aptitud efectiva es $s'_{eff} = ((1 - p_c \frac{(t-1)}{N-1}) \frac{\bar{f}(\xi, t)}{f(t)} - 1)$. En el límite de evolución temporal continua (3.4.13) puede ser integrado para llegar a

$$P(\xi, t) = e^{\int_0^t s'_{eff}(t') dt'} P(\xi, 0) + e^{\int_0^t s'_{eff}(t') dt'} \int_0^t j(t') e^{-\int_0^{t'} s'_{eff}(t'') dt''} dt' \quad (3.4.14)$$

Siguiendo lo indicado en [71], [72]. Poli [79], recientemente, ha extendido esta definición de aptitud efectiva a la programación genética donde permite entender cualitativamente el fenómeno de bloat. También puede ser usado para entender como la robustez evolutiva, medida por la aptitud efectiva, supera el problema de “fragilidad”.

La aptitud efectiva toma en cuenta todos los operadores genéticos de una manera “democrática”. Cualquier operador genético representado por O mapea una población al tiempo t a otra población en $t+1$ i.e. $n(i, t+1) = On(i, t)$. Esto no quiere decir que los operadores son iguales en cuanto a sus efectos. Una mutación que es neutral en el nivel del paisaje de aptitud puede ser no-neutral en el paisaje de aptitud efectiva. Sólo cuando no hay otros operadores genéticos además de la selección se presenta el caso de que la aptitud y la aptitud efectiva son las mismas. Por ejemplo en el modelo de Eigen [66] la relación es:

$$f_{eff}(i) = \wp(i)f(i) + \sum_j \wp(j \rightarrow i)f(j) \frac{P(j, t)}{P(i, t)} \quad (3.4.15)$$

donde $f(i)$ es la aptitud del individuo tipo i , $\wp(i)$ es la probabilidad de que el genotipo i no mute y $\wp(j \rightarrow i)$ es la probabilidad de que el genotipo j mute al genotipo i . Tanto $\wp(i)$ como $\wp(j \rightarrow i)$ dependen de la tasa de mutación p . Solamente en el límite $p \rightarrow 0$ se tiene $f_{eff} = f(i)$. La aptitud efectiva da significado a la cuestión

de adaptabilidad de los sinónimos: aquellos sinónimos que tienen una mayor aptitud efectiva se adaptarán mejor al ambiente que aquellos con menor aptitud efectiva. La aptitud efectiva es una medida más apropiada para la adaptación en el paisaje de aptitud al representar el crecimiento de un genotipo particular durante varias generaciones tomando en cuenta no sólo su propia ventaja selectiva, asociada con el paisaje de aptitud en sí, sino también su habilidad para producir descendencia con alto grado de adaptación y que a su vez producirá descendencia con alto grado de adaptación y así sucesivamente.

La función de aptitud efectiva toma en cuenta la simetría de sinónimos y provoca presión selectiva que fortalece la producción de mutantes potencialmente exitosos al seleccionar, entre aquellos sinónimos a los que tienen una alta probabilidad de producir descendencias altamente adaptadas. Este concepto implica la existencia de un mecanismo de retroalimentación del medio ambiente hacia el individuo en la búsqueda genética. Se puede pensar en el concepto de “aptitud efectiva” como el modo de combinar la selección natural Darwiniana “tradicional”, asociada con el paisaje de aptitud, con el mecanismo de retroalimentación sugerido por Lamarck, sin embargo lo que se propone no está en ningún sentido en contradicción con el dogma central (como es el caso del enfoque de Lamarck). Lo que se demuestra es que la selección natural actúa al nivel de la “aptitud efectiva” más que sobre el paisaje de aptitud. Esto permite entender el fenómeno de la ortogénesis, la “tendencia de la evolución a ocurrir en ciertas direcciones no por efecto de restricciones en la selección sino por el sistema subyacente prefiere determinadas direcciones en presencia de mutaciones “aleatorias” [81]. Así la ortogénesis no es más que la selección natural entre sinónimos empleando la función de aptitud efectiva.

3.4.4 Aportes

Teorema de esquemas La teoría de esquemas convencional provee un límite inferior para el número esperado de esquemas debido a que no toma en cuenta explícitamente la reconstrucción de estos. La ecuación (3.4.5), toma en cuenta el efecto de reconstrucción de esquemas debido tanto a la mutación como al entrecruzamiento. Junto con la definición de aptitud efectiva en la ecuación (3.4.12) permite establecer un nuevo teorema de esquemas:

Teorema de esquemas

$$P(\xi, t + 1) = \frac{f_{eff}(\xi, t)}{\bar{f}(t)} P(\xi, t) \quad (3.4.16)$$

La interpretación de este esquema es similar a la del teorema tradicional: esquemas mejores que el promedio de *aptitud efectiva* serán ubicados “exponencialmente” mayor número de veces. Se usan las comillas para exponencialmente ya que esto sólo es cierto en casos sencillos como paisajes de aptitud planos. Es importante mencionar que hay una gran diferencia entre la selección estándar basada en aptitud o en aptitud efectiva ya que esta última toma en cuenta el efecto de todos los operadores genéticos. Como se verá en el siguiente capítulo, el hecho de que cuerdas con un valor cero como aptitud de selección tengan un aumento exponencial muestra claramente que la aptitud efectiva es un concepto más relevante. En este sentido este nuevo teorema de esquemas no solamente establece lo obvio — que esquemas aptos preservados por el operador de cruce prosperarán, sino que enfatiza el papel del efecto constructivo del entrecruzamiento. El nuevo elemento a observar es la reconstrucción de esquemas que generalmente es el aporte dominante.

La ecuación de evolución de esquemas tiene muchas características importantes

siendo una de ellas el modo en que relaciona la evolución temporal con niveles diferentes de agrupación (coarse graining). Antes es necesario regresar a la ecuación de evolución para cuerdas (3.4.5). Hasta este punto se han presentado los resultados de la manera más general posible — para cualquier paisaje y tomando en cuenta mutación y cruza. Para el resto de esta sección se hará más énfasis en el entrecruzamiento. La razón para esto es que el considerar la importancia de la longitud del esquema vis a vis la hipótesis de bloques constructores. Siendo la mutación un operador local su consideración no es relevante en este contexto. Note que la ecuación esta escrita totalmente en términos de los grados de libertad fundamentales — las cuerdas. Al pasar a la forma (3.4.6) se ha logrado una agrupación sumando sobre todas las cuerdas que contienen c_i^L sin importar lo que haya al lado derecho del punto de entrecruzamiento, y de manera similar para las cuerdas que contienen a c_i^R . Debido a que la naturaleza del entrecruzamiento sugiere la idea de agrupación (al considerar el intercambio de segmentos entre las cuerdas, i.e. una vez seleccionado el punto de entrecruzamiento se obtienen nuevos individuos uno de ellos formado por el bloque izquierdo del primer padre y el bloque derecho del segundo padre y el segundo individuo con el bloque izquierdo del segundo padre y el bloque derecho del primero, ver ejemplo en capt 2, sección 2.1.1) de ambos para uno y para el otro con el) y más específicamente la idea de esquema, dado que c_i^L y c_i^R definen un esquema de orden y tamaño k y $N - k$ respectivamente. Con el fin de resolver la ecuación (3.4.6) es necesario conocer $P(c_i^L, t)$ y $P(c_i^R, t)$. Las cuales obedecen las ecuaciones de evolución de la forma

$$P(c_i^L, t + 1) = P'(c_i^L, t) - \frac{p_c}{N - 1} \sum_{m=1}^{k-1} (P'(c_i^L, t) - P'(c_i^{LL}, t)P'(c_i^{LR}, t)) \quad (3.4.17)$$

donde c_i^{LL} y c_i^{LR} son las partes izquierda y derecha de c_i^L , izquierda y derecha definida respecto al punto de entrecruzamiento m , donde $m < k$. c_i^{LL} y c_i^{LR} como

esquemas están más agrupados que c_i^L , i.e. son de orden más bajo. Este patrón de comportamiento continua, i.e. para calcular $P(c_i, t+1)$ se requiere $P(c_i^L, t)$ y $P(c_i^R, t)$ que a su vez requieren $P(c_i^{LL}, t-1)$, $P(c_i^{LR}, t-1)$, $P(c_i^{RL}, t-1)$ y $P(c_i^{RR}, t-1)$ etc. Por cada paso atrás en el tiempo se pasa a una agrupación mayor de grados de libertad. c_i considerado como esquema que es de mayor orden que c_i^L or c_i^R , que a su vez son de mayor orden que c_i^{LL} , c_i^{LR} , c_i^{RL} and c_i^{RR} . La mayor agrupación de grados de libertad efectivos es el esquema 1. No es posible partir al esquema 1 por lo que el entrecruzamiento es explícitamente neutral, i.e. el esquema 1 obedece la ecuación

$$P(i, t + 1) = P'(i, t) \quad (3.4.18)$$

Como ejemplo sencillo considere un cuerda de 4 bits $ijkl$. La estructura jerárquica de un posible árbol genealógico puede ser escrita como

t+1	$ijkl$					
t	ijk, l	ij, kl	i, jkl			
t-1	ij, k	i, jk	i, j	k, l	jk, l	j, kl
t-2	i, j	j, k			j, k	k, l

Este árbol muestra sólo el efecto del término de reconstrucción en la ecuación de esquemas sobre el espacio de tres generaciones. Por supuesto que hay muchos otros procesos que contribuyen a la aparición de $ijkl$ al tiempo $t + 1$ que involucran varias combinaciones de destrucción y construcción de esquemas. Al sólo considerar la reconstrucción de esquemas se observa que el esquema 1 juega un papel privilegiado ya que representa el último de los bloques constructores. Para una cuerda de N bits el máximo número de pasos temporales necesarios para que todos los ancestros sean el esquema 1 es $N - 1$.

Todo lo anterior se aplica al esquema genérico, ξ , compuesto de los esquemas, ξ_L y ξ_R que a su vez están compuestos por los esquemas ξ_{LL} , ξ_{RL} , ξ_{LR} y ξ_{RR} etc. Como se puede apreciar la idea de bloques constructores se manifiesta en la estructura de las ecuaciones de evolución. ξ_{LL} , ξ_{RL} , ξ_{LR} y ξ_{RR} son bloques constructores para ξ_L y ξ_R que son bloques constructores para ξ . Los últimos bloques constructores son los esquemas 1. En el ejemplo anterior de una cuerda de 4 bits o esquemas los cuatro bloques constructores de orden uno, i , j , k y l se combinan para formar bloques constructores de orden dos ij y kl , etc. que a su vez se combinan con los bloques de orden uno para formar bloques de orden tres, ijk y jkl , etc. Los bloques de orden tres se combinan con los bloques de orden uno, o dos con dos, para formar bloques de orden 4 y así sucesivamente.

En términos de la aptitud efectiva, $f'_{eff}(\xi, t)$

$$P(\xi, t) = e^{\int_0^t s'_{eff}(t') dt'} P(\xi, 0) + \frac{p_c e^{\int_0^t s'_{eff}(t') dt'}}{N-1} \sum_{k=1}^{l-1} \int_0^t P'(\xi_L, t') P'(\xi_R, t') e^{-\int_0^{t'} s'_{eff}(t'') dt''} dt' \quad (3.4.19)$$

Dada la dificultad de hacer un análisis específico sobre un paisaje arbitrario se trabajará con paisajes sencillos. El primero de ellos es el caso de un paisaje plano. En este caso el coeficiente que mide la presión selectiva es $s_{eff} = -p_c(l-1)/(N-1)$, así

$$P(\xi, t) = e^{-p_c \frac{(l-1)}{(N-1)} t} P(\xi, 0) + e^{-p_c \frac{(l-1)}{(N-1)} t} \frac{p_c}{N-1} \sum_{k=1}^{N-1} \int_0^t P'(\xi_L, t') P'(\xi_R, t') e^{p_c \frac{(l-1)}{(N-1)} t'} dt' \quad (3.4.20)$$

Note que la dependencia de la condición inicial, $P(\xi, 0)$, es exponencialmente superada a menos que ξ no sea el esquema 1, la solución para la ecuación del esquema

1 es

$$P(i, t) = P(i, 0) \quad (3.4.21)$$

Teorema de Geiringer Una consecuencia inmediata cuando se considera el término fuente que describe la reconstrucción es que los únicos términos no cero que se necesitan tomar en cuenta son aquellos que provienen del esquema 1, ya que cualquier término de orden alto siempre estará acompañado de un factor de amortiguamiento exponencial. Así la distribución de punto fijo para un algoritmo genético evolucionando en un paisaje plano es:

$$P^*(\xi) = \lim_{t \rightarrow \infty} P(\xi, t) = \prod_{i=1}^{N_2} P(i, 0) \quad (3.4.22)$$

el cuál es básicamente el Teorema de Geiringer [82] en el contexto de distribuciones de esquema y entrecruzamiento simple. Note que esta distribución de punto fijo aparece solamente por los efectos de reconstrucción, cuya ausencia lleva a un amortiguamiento exponencial y el comportamiento no físico $P(\xi) \rightarrow 0$. Esta versión del teorema de Geiringer también funciona para paisajes no planos donde la selección es muy débil, i.e. $\frac{f(\xi, t)}{f(t)} \sim (1 + \epsilon)$ y $\epsilon < \frac{p_c \frac{(l-1)}{(N-1)}}{1 - p_c \frac{(l-1)}{(N-1)}} \quad \forall l > 1$. Bajo tales circunstancias el esquema 1 estará asociado con el factor de amortiguamiento exponencial. La diferencia entre los dos casos es que para el paisaje plano el punto fijo es establecido por las proporciones iniciales de los diferentes esquemas y no hay competencia entre ellos. En el último debido al paisaje no trivial ciertos esquemas 1 son preferidos sobre otros. Un ejemplo concreto de tal paisaje sería $f_i = 1 + \alpha_i$ donde $\sum_i |\alpha_i| \leq \frac{\epsilon}{(2+\epsilon)}$ y f_i es la aptitud del i -ésimo bit.

Hipótesis de bloques constructores De la forma en que la ecuación de evolución de esquemas esta estructurada se ve que la aptitud efectiva en base a los efectos del entrecruzamiento consiste en un término de destrucción y en un término de reconstrucción. Inherente a la estructura del término de reconstrucción esta una forma de hipótesis de bloques constructores — aquellos esquemas de orden elevado están formados por esquemas de esquemas pequeños y de orden bajo. Si $P'(\xi, t) > P'(\xi_L(k), t)P'(\xi_R(k), t)$ entonces los efectos de la destrucción superarán a los de reconstrucción, mientras que si $P'(\xi, t) < P'(\xi_L(k), t)P'(\xi_R(k), t)$ la reconstrucción dominará. Para hacer un análisis más profundo se considerará el caso de esquemas de orden 2 en un paisaje plano, dentro del cual se tiene

$$s_{eff} = -p_c \left(\frac{l-1}{N-1} \right) + p_c \left(\frac{l-1}{N-1} \right) \frac{P(i, 0)P(j, 0)}{P(ij, t)} \quad (3.4.23)$$

Aquí se observa que el efecto de reconstrucción es mayor que el de destrucción si i y j están negativamente correlacionados. Si la reconstrucción es más importante la contribución de este último es maximizado al aumentar la longitud del esquema. En otras palabras los esquemas grandes son los favorecidos.

En general el paisaje de aptitud induce correlaciones entre ξ_L y ξ_R . En este caso existe una competencia entre el efecto anti-correlacionado del paisaje y el efecto de mezcla del entrecruzamiento. Es más frecuente que la selección induzca la anti-correlación entre partes de esquema aptos que una correlación positiva. De hecho, en el caso neutral de paisajes no epistáticos se tiene $1 + \frac{2N_L}{N} \delta f_\xi < (1 + \frac{2N_L}{N} \delta f_{\xi_L})(1 + \frac{2N_R}{N} \delta f_{\xi_R})$ donde δf_ξ , δf_{ξ_L} y δf_{ξ_R} son las desviaciones de aptitud del esquema ξ , ξ_L y ξ_R de la aptitud promedio que se a normalizado a la mitad. Entonces se observa que la selección induce una anti-correlación cuando $\delta f_{\xi_L}, \delta f_{\xi_R} > 0$ y por lo tanto se tiene una población inicial no correlacionada, $P'(\xi, t) \neq P'(\xi_L, t)P'(\xi_R, t)$. Esto

significa que el entrecruzamiento desempeña un papel importante el permitir que ambas partes de un esquema exitosa aparezcan en el mismo individuo. El efecto del entrecruzamiento es debilitar las anti-correlaciones inducidas por la selección y hacer más fácil el encontrar el esquema entero. Es posible mostrar que para un paisaje no epistático la contribución a la aptitud poblacional de todos los esquemas de longitud l , empezando con una población inicial aleatoria al tiempo t , es independiente de l a $t + 1$ y es una función incremental de l para l grandes al tiempo $t + 2$ [72].

Siempre es posible el inventar un paisaje donde haya ventaja de aptitud asociada con bits que están juntos. Sin embargo también es posible el encontrar paisajes para bits que están separados. Los paisajes no epistáticos mencionados arriba son neutrales en este sentido y por lo tanto cualquier resultado acerca de la naturaleza del esquema y de los bloques constructores son un reflejo del efecto geométrico del entrecruzamiento y no están asociados con correlaciones bit a bit del paisaje en sí.

Parámetros de mutación y cruza óptimas En esta sección se muestra que sólo es para paisajes triviales es posible calcular de manera analítica los parámetros óptimos para el funcionamiento de un algoritmo genético. Razón por la cual es necesario un algoritmo genético autoadaptado o inteligente que evite el invertir tiempo experimentado sobre la tasa óptima de algoritmo.

La ecuación a emplear es la propuesta por Stephens y Waelbroeck, [72], [78] 3.4.5, como se mencionó, el término $\mathcal{P}(c_i) = \prod_{k=1}^N (1 - p(k))$ que representa la probabilidad de que la cuerda i permanezca sin cambio, $p(k)$ es la probabilidad de mutación del bit k que se considera constante y

$$\mathcal{P}(c_j \rightarrow c_i) = \prod_{k \in \{C_j - C_i\}} p(k) \prod_{k \in \{C_j - C_i\}_c} (1 - p(k)) \quad (3.4.24)$$

donde $\{c_j - c_i\}$ es el conjunto de bits que son diferentes entre c_j y c_i y $\{c_j - c_i\}_c$ es el complemento de dicho conjunto, i.e. el conjunto de bits que es común en ambas cuerdas. $\mathcal{P}(c_j \rightarrow c_i)$ es la probabilidad que la cuerda j mute a la cuerda i . En el límite donde la tasa de mutación es uniforme, $\mathcal{P}(c_i) = (1 - p)^N$ y $\mathcal{P}(c_j \rightarrow c_i) = p^{d^H} (1 - p)^{N - d^H(i,j)}$, donde $d^H(i, j)$ es la distancia de Hamming entre la cuerdas c_i y c_j . Esta ecuación de evolución toma en cuenta exactamente el número promedio de cuerdas y en el límite de una población grande da la distribución de probabilidad para las cuerdas.

Con esto es posible investigar la evolución de cualquier función. Una de particular interés es el incremento de la aptitud poblacional por generación

$$\delta \bar{f} = \bar{f}(t + 1) - \bar{f}(t) \quad (3.4.25)$$

al emplear la ecuación (3.4.5) se tiene:

$$\delta \bar{f} = \sum_{C_i} f(c_i, t) \mathcal{P}(c_i) P_c(c_i, t) + \sum_{C_i} \sum_{C_j \neq C_i} f(c_i, t) \mathcal{P}(c_j \rightarrow c_i) P_c(c_j, t) - \sum_{C_i} f(c_i, t) P(c_i, t) \quad (3.4.26)$$

Esta ecuación es no-lineal para p pero lineal para la probabilidad de cruza p_c .

Los valores de p y p_c tienen un papel muy importante en el éxito de un algoritmo genético. Se entiende por éxito como el aumento en la aptitud promedio generacional de la población \bar{f} . Como $\delta \bar{f}$ depende sólo linealmente de p_c , el valor de p_c optimiza el aumento de la aptitud en una generación, sin considerar efectos de tamaño finito será 0 o 1. Esto es de alguna manera engañoso sin embargo el cambio al integrar el valor de aptitud, $\bar{f}(t) - \bar{f}(0)$, será altamente no lineal para p_c excepto cuando $t=1$. Para el intervalo entre dos generaciones, al analizar la ecuación para P_c se observa que aún

cuando p_c sea maximizada o minimizada depende de la correlación entre las cuerdas con mayor aptitud. Si la correlación es negativa, i.e. $P'(C_i, t) < P'(C_i^L, t)P'(C_i^R, t)$, entonces la reconstrucción de cuerdas de alta aptitud domina sobre la destrucción de estas y p_c debe ser maximizado. Para ilustrar esto considérese el caso cuando $p=0$.

$$\delta \bar{f} = \Delta \bar{f} - \frac{p_c}{N-1} \sum_{k=1}^{N-1} \sum_{C_i} f(C_i, t) (P'(C_i, t) - P'(C_i^L, t)P'(C_i^R, t)) \quad (3.4.27)$$

donde $\Delta \bar{f}(t) = (\bar{f}^2 - (\bar{f})^2)/\bar{f}$ es la varianza en la aptitud poblacional. Para el caso de una población de cuerdas de 2 bits

$$\delta \bar{f} = \Delta \bar{f} - \frac{p_c}{f^2} (f_{00} + f_{11} - f_{10} - f_{01})(f_{10}P_{10}f_{01}P_{01} - f_{00}P_{00}f_{11}P_{11}) \quad (3.4.28)$$

Para que el entrecruzamiento sea una fuerza positiva se requiere que el segundo término de la ecuación anterior sea positivo. Para un paisaje de aptitud lineal, donde $f_{00} + f_{11} = f_{10} + f_{01}$, la cruce en un problema de 2 bits es neutral, aunque como se mencionó anteriormente esto es para el caso de un paso temporal. Sin embargo, para un paisaje deceptivo de tipo I o tipo II [46] se encuentra que el entrecruzamiento es destructivo. En este caso el tener valores bajos de p_c sería benéfico. Pero en el caso de efecto de tamaño finito valores muy bajos de p_c pueden inhibir la diversidad genética.

En [46] se observa que si en la suma sobre las cuerdas, el paisaje es deceptivo en el sentido que $P'(c_i^L, t)P'(c_i^R, t) < P'(c_i, t)$, el entrecruzamiento inhibirá el aumento en la aptitud, mientras que si $P'(c_i^L, t)P'(c_i^R, t) > P'(c_i, t)$ el aumento en la aptitud se verá favorecido. En el primer caso p_c bajo será benéfico mientras que en el segundo se requerirá que sea alto. En general, la probabilidad de cruce es proporcional a la decepción que presente en el algoritmo genético. Es importante notar que se puede tener decepción respecto a un punto para la cruce, lo que implica que una probabilidad de cruce $p_c(k)$ que dependa de la posición dentro del cromosoma sería

útil. Esto demuestra la dependencia del paisaje a optimizar.

Por lo que respecta a las mutaciones, si por el momento no se considera al operador de cruce y se toma a p como constante se tiene:

$$\delta \bar{f} = \sum_{C_i} \sum_{C_j} f(C_i, t) P'(C_j, t) p^{d^H(i,j)} (1-p)^{N-d^H(i,j)} - \sum_{C_i} f(C_i, t) P(C_i, t) \quad (3.4.29)$$

así

$$\frac{d\delta \bar{f}}{dp} = \sum_{C_i} \sum_{C_j} f(C_i, t) P'(C_j, t) p^{d^H(i,j)-1} (1-p)^{N-d^H(i,j)-1} (d^H(i,j) - pN) \quad (3.4.30)$$

Se puede obtener un estimado del óptimo de p al resolver la ecuación anterior al igualarla a cero en una aproximación de campo medio donde se reemplaza $d^H(i,j)$ por $\langle d^H(i,j) \rangle$, el valor esperado sobre la población. De esta manera se obtiene $p^* = \langle d^H(i,j) \rangle / N$. Para una población aleatoria, donde $\langle d^H(i,j) \rangle \sim N/2$, $p^* = 1/2$. Cerca del límite de una población ordenada donde $\langle d^H(i,j) \rangle \sim 0$, $p^* \sim 0$. Sin embargo p^* será dependiente del tiempo. Aún cuando la tasa de mutación sea o no cero en el límite ordenado o no dependerá de si la población se ha ordenado respecto al óptimo global o a un óptimo local. Si se presenta este último caso una tasa de mutación cero sería lo peor que pudiera pasar.

Con lo anterior queda demostrado que no hay ninguna regla teórica para la optimización de parámetros. Estos dependen tanto del paisaje de aptitud como del ordenamiento de la población. De ahí la importancia de que el algoritmo genético sea capaz de encontrar por sí mismo las tasas de mutación y cruce adecuadas para su funcionamiento.

Capítulo 4

Algoritmos genéticos inteligentes

Hasta la más sencilla criatura se transforma así misma para adaptarse al ambiente en el que se encuentra, siendo la adaptación un requisito indispensable para cualquier sistema cognoscitivo [83].

Tomando la definición de inteligencia propuesta por Fogel [22] como la capacidad de un sistema para adaptar su comportamiento y así alcanzar sus objetivos dentro de un rango de ambientes se tiene el concepto de algoritmos genéticos inteligentes los cuales usan de manera explícita el mapeo genotipo-fenotipo permitiendo al sistema adaptarse a cambios en el ambiente con el fin de alcanzar sus metas.

La adaptación en los algoritmos genéticos inteligentes esta vinculada con el rompimiento de simetría inducido y la aptitud efectiva como veremos en este capítulo.

4.1 Mapeo genotipo-fenotipo

En muchas de las aplicaciones reales se tiene un espacio de búsqueda definido por un conjunto de objetos, e.g. unidades de procesos, bombas, calentadores y enfriadores en una planta química, portafolios de inversión, cada uno de los cuales tiene diferentes parámetros como consumo de energía, capacidad de producción, dinero disponible,

etc. Esos parámetros que son sujetos a optimizar constituye el *espacio de fenotipos*. Por otra parte, los operadores genéticos con frecuencia trabajan con operadores matemáticos abstractos como por ejemplo las cuerdas binarias, estrategias de mercado o la secuencia de trabajos para un problema de calendarización, este espacio constituye el *espacio de genotipos*. Obviamente se necesita un mapeo o función entre el espacio de fenotipos y el de genotipos.

Dentro de los algoritmos genéticos tradicionales estos dos espacios con frecuencia son los mismos, al emplear una codificación uno a uno, i.e. cada genotipo codifica a un sólo fenotipo, por ejemplo en la codificación de una función de optimización del tipo $y = x^2$ donde la variable x es codificada dentro de una cuerda binaria la cual al traducirse a decimal da el valor de la función.

El empleo de organizaciones genotipo-fenotipo, los cuales caracterizan a los algoritmos genéticos inteligentes, se inspira en los resultados de la evolución natural de los individuos, donde los procesos dinámicos fundamentales están conectados a procesos dinámicos en secuencias. El mensaje genético — el genotipo — es codificado por una secuencia de ADN y la totalidad de la dinámica en la célula esta determinada por la interacción de proteínas y polinucleótidos. El fenotipo son todas las propiedades que caracterizan al individuo, donde la asignación de genotipos a fenotipos se le llama mapeo genotipo-fenotipo. El proceso de selección Darwiniano esta basado en la aptitud de los fenotipos.

Las mutaciones aleatorias del genotipo producen mutaciones no aleatorias del fenotipo debido a la existencia de sinónimos y del rompimiento de esta simetría, proceso que incorpora información del medio. Al emplear un algoritmo genético inteligente para atacar problemas combinatorios dinámicos el genotipo consiste en la

codificación representación de una instancia del problema y el fenotipo es el resultado de aplicar el intérprete sobre el conjunto de reglas.

El mapeo genotipo-fenotipo dentro de los algoritmos genéticos inteligentes permite que la selección de los genotipos tome en cuenta tanto la ventaja selectiva al momento como la capacidad del individuo de producir descendientes capaces y así sucesivamente. Ya que la mutación y la recombinación actúan de manera diferente sobre los sinónimos, estos tendrán diferentes descendencias tanto en el aspecto de sobrevivencia del individuo como en la aparición de nuevos individuos.

En algoritmos genéticos inteligentes, sin pérdida de generalidad, se puede considerar la secuencia de reglas o codificaciones como el genotipo de los individuos. Todas las posibles secuencias pueden ser consideradas como elementos del espacio de genotipos G .

Un individuo viene a ser la expresión de un genotipo g . Las características del individuo se manifiestan al construir su fenotipo ϕ perteneciente al espacios de fenotipos Q . El mapa genotipo-fenotipo $\Phi : g \in G \rightarrow \phi \in Q$ asigna a los genotipos g a los fenotipos ϕ . Cada genotipo será evaluado en el proceso de selección al determinar su función de aptitud $F(\phi)$. Cuando esto sucede se obtiene un valor de aptitud $V(g) = F(\phi)$, $\phi = \Phi(g)$. El valor de aptitud $V(g)$ es una cantidad escalar. De manera gráfica se tiene:

$$\begin{array}{ccc}
 G & \xrightarrow{\Phi} & R_+ \\
 \phi \searrow & & \nearrow F \\
 & & Q
 \end{array}$$

La superficie formada por los valores de aptitud en el espacio de secuencias se le llama paisaje de aptitud. Para un problema discreto, el paisaje consiste de un

conjunto discreto de puntos pero para facilitar su interpretación se conectan estos puntos formando una superficie, e.g. mediante una aproximación piezo-lineal.

Una de las principales razones para usar un intérprete en algoritmos genéticos es que la codificación genotípica original puede no ser la más robusta. Este es el caso de la codificación binaria en el escenario estándar con los operadores de selección, mutación y cruza normales. En la literatura se ha demostrado que la codificación Gray [84] que mapea vecinos euclidianos en vecinos en función de la distancia de Hamming es más efectiva [43].

Algunos de los requisitos que debe cumplir el intérprete son:

- Es deseable el uso de relaciones muchos a uno, i.e. es necesario que existan muchos genotipos para cada fenotipo.
- Para cualquier intérprete, existe una función de aptitud inducida, dada por la composición de la función de aptitud del fenotipo mediante el intérprete (la aptitud reproductiva del cromosoma es la aptitud del fenotipo).
- El intérprete debe magnificar la densidad de estado acerca de los óptimos de V

El mapeo genotipo-fenotipo no trivial, i.e. cuyo mapeo no es uno a uno, conlleva a la simetría de sinónimos que no es más que la redundancia de información que le confiere al sistema la robustez necesaria para adaptarse a cambios en el medio en que este se desarrolla. En general este mapeo es no-inyectivo por lo que existe un gran número de sinónimos genotípicos para cada fenotipo.

4.2 Simetría de sinónimos

Como se vió en la sección anterior el mapeo genotipo-fenotipo conlleva una simetría de sinónimos, la representación de un determinado fenotipo por diferentes genotipos, permitiendo que los diferentes operadores genéticos produzcan mutaciones aleatorias a nivel genotípico que se reflejan como mutaciones no aleatorias en el fenotipo permitiendo la autoorganización del conjunto genético.

Es importante mencionar que entre sinónimos por definición no hay directamente una ventaja selectiva para uno o el otro. Sin embargo en la naturaleza se observa que hay algunos que son preferidos a otros. La pregunta es: esto es debido a algún tipo de ventaja selectiva o es solamente un efecto aleatorio?. En la presencia de otros operadores genéticos además del operador de reproducción la respuesta es que hay una ventaja selectiva. El paisaje de aptitud ciertamente no puede dar una medida dicha ventaja pero es posible definir una aptitud efectiva (ecuación 3.4.11 del capítulo anterior).

Un ejemplo simple muestra la no-equivalencia de diferentes sinónimos bajo el operador de mutación. Considérese las palabras en inglés *dead* y *defunct*, una mutación puntual consiste en el cambio de una sola letra. La palabra *dead* puede mutar a *deed*, *bead*, *lead*, *deaf*, *dear*, *read* o *deal* todos ellos con significado, pero es difícil generar una palabra con significado al aplicar una mutación puntual sobre *defunct*. Como se mostrará los sinónimos pueden diferir en su mutabilidad. La sobrevivencia de un organismo dependerá de la existencia de sinónimos de alta mutabilidad que le permitan adaptarse rápidamente a cambios en el ambiente.

La existencia de sinónimos a nivel genotipo también se puede encontrar dentro

de los mercados financieros, identificando como genotipos al conjunto de recomendaciones para el armado de portafolios de inversión y como fenotipo el valor de dicho portafolio, por ejemplo suponga que usted tiene x_A acciones de la emisora A y x_B acciones de la emisora B así como una cantidad X de dinero en efectivo, el precio de cada una de las acciones es v_A y v_B respectivamente, por lo tanto el valor del portafolio será: $x_A * v_A + x_B * v_B + X$ los genotipos que dan el mismo valor de portafolio serán todos aquellas configuraciones que sean solución de la ecuación $k = x_A * v_A + x_B * v_B + X$ donde k es una constante.

4.3 Rompimiento de simetría inducido

La importancia de la distinción entre fenotipo y genotipo en la evolución ha sido sugerida por otros autores [85],[86]. En la naturaleza no todos los sinónimos se observan y aquellos que pueden ser identificados se presentan en diferentes proporciones, i.e. se muestra un rompimiento en la simetría de los sinónimos. Por la teoría de procesos ramificados [87] y por la teoría neutral de la evolución molecular de Kimura [88] se ve que el rompimiento de simetría ocurre en forma espontánea en un conjunto evolutivo, en esta parte de la tesis se demuestra que el *rompimiento de simetría inducido* también ocurre debido a la violación de la simetría de sinónimos por efecto de los operadores genéticos como la mutación y la recombinación. Como ejemplo de rompimiento de simetría imagina una función con dos óptimos con el mismo valor, al evolucionar un algoritmo genético se encuentra la población converge a uno u otro de los óptimos por efecto de los operadores genéticos.

Otro ejemplo de rompimiento de simetría inducido sería: Considere el efecto de

mutación sin entrecruzamiento en el contexto de un modelo que consiste en un esquema de orden 2, 11, 01, 10, 00, donde cada esquema puede mutar o cualquiera de sus dos esquemas adyacentes cuando los estados 11, 10, 00, 01 se colocan de manera circular conforme a las manecillas del reloj. Por ejemplo, 11 puede mutar a 10 o 01 pero no a 00. Este es el límite donde mutaciones de dos bits no tienen importancia comparada con mutaciones de un bit. Asumiendo un paisaje del tipo: $f(11) = f(01) = f(10) = 2$, $f(00) = 1$. En una población aleatoria, $P(11) = \dots = P(00) = \frac{1}{4}$. Si hay una probabilidad uniforme p_m para que cada esquema mute a uno adyacente entonces la ecuación que describe este sistema es

$$P(i, t + 1) = (1 - 2p_m)P'(i, t) + p_m(P'(i - 1, t) + P'(i + 1, t)) \quad (4.3.1)$$

Para $p_m = 0$ el estado estable la población es $P(11) = P(01) = P(10) = 1/3$, $P(00) = 0$. Con esto se ve una la simetría de sinónimos en el paisaje asociada con la degeneración de los estados 11, 10 y 01 no esta rota. Sin embargo, para $p_m > 0$, la distribución de esquemas a $t = 1$, empezando con una población aleatoria en $t = 0$, es $P(11) = 2/7$, $P(01) = P(10) = (2 - p_m)/7$, $P(00) = (1 + 2p_m)/7$. Así se observa que hay un rompimiento de simetría inducido de la simetría de sinónimos por efecto de la mutación.

El rompimiento de simetría puede estar relacionado a la emergencia de un lenguaje algorítmico, al igual que el rompimiento de simetría de permutación de las letras permite la existencia de palabras en el lenguaje natural. Para comprender mejor esto, consideremos al cromosoma como un algoritmo, donde el intérprete es la "computadora" la cual ejecuta este algoritmo produciendo el fenotipo. Así, el rompimiento de simetría esta relacionado con la selección de un "lenguaje", donde las "palabras" o "reglas gramáticas" son seleccionadas si mejoran la búsqueda de buenas soluciones

[90].

Ya que el lenguaje algorítmico está íntimamente relacionado con la evolución del algoritmo, facilitará la búsqueda de soluciones futuras sólo si la descomposición del problema es estable, por lo que la evolución de la topografía debe preservar esta descomposición para que el lenguaje que fue útil en el pasado siga siendo útil en el futuro.

Dado que el rompimiento de simetría es debido en parte a presiones selectivas, es razonable esperar que los genotipos mutantes son producidos de una manera que exhiban un cierto grado de autoorganización. La información del ambiente es incorporada no a través de un sólo individuo sino indirectamente por medio del rompimiento de simetría en el conjunto genético. El rompimiento de simetría respeta las estrategias de mutación correctas bajo el precepto de que estrategias que fueron útiles en el pasado seguirán siendo útiles en el futuro.

Como se verá en el experimento con el modelo de girafas del capítulo siguiente, el desarrollo poblacional no requiere que las mutaciones a nivel del cromosoma sean organizadas. La información que permite a las mutaciones estar organizadas *en el nivel fenotípico* está codificada en la distribución de sinónimos y expresada dentro del mapeo genotipo-fenotipo.

Un ejemplo de un organismo simple donde la estrategia de mutación puede ser seleccionada, y donde la evidencia de la existencia de lenguaje es la evolución de un epítipo de neutralización viral *in vivo* [89, 90, 91]. En este caso con el fin de evadir a la respuesta inducida del sistema defensivo del individuo. Dado que esta estrategia es válida en todo momento, es una situación donde el rompimiento de simetría, el cual refleja la presión selectiva al mutar en el pasado, también se puede esperar que

funcione en el futuro al mejorar la habilidad del epítopo a mutar. En el artículo de Stephens-Waelbroeck [91] se llevó a cabo un análisis de las proteínas *env* que favorece esta teoría. Sin embargo en ese caso otros factores de selección compiten con la necesidad de mutar; por ejemplo la estructura secundaria m-RNA, enzima y t-RNA disponibles en células infectadas, y restricciones para la estructura secundaria juegan el mismo papel. Todavía no es totalmente conocido que segmento(s) de la región del lazo V3 u otras regiones hiper-variables sea reconocido como un epítopo de neutralización por la inmonoglobulina.

Lo anterior muestra la dificultad de identificar el rompimiento de simetría en un sistema biológico real debido a la complejidad de los posibles paisajes de aptitud los cuales dependen de un gran número de variables independientes, es muy difícil saber si uno ha considerado todos los factores o la posibilidad de ser engañado al pensar que dos genotipos son sinónimos sólo porque no se ha entendido la conformación del paisaje de aptitud. Por esto es necesario el emplear modelos sencillos para tratar de entender el fenómeno como requisito para cualquier intento de explicar o demostrar la existencia de procesos análogos en la naturaleza. La demostración del rompimiento de simetría será en el siguiente capítulo a través de tres modelos sencillos, el primero sobre el comportamiento de un virus donde *solamente* el factor selectivo es la necesidad de adaptarse a la respuesta inmune del organismo. En este caso es posible identificar sin ninguna ambigüedad sinónimos con respecto a la función de aptitud. El segundo demuestra la existencia del fenómeno de rompimiento de simetría en sistemas más complicados y esta enfocado en hacer más plausible los comentarios sobre los cuellos de girafas mostrando que es de hecho posible tener una preferencia "ortogénica" por características muy específicas de adaptación —cuellos más largos siendo el último un

modelo de un algoritmo genético que es capaz de adaptarse a cambios de los diferentes paisajes a optimizar.

4.4 Ejemplos analíticos

Estos ejemplos consideran el efecto de los operadores genéticos, además de la selección, actuando sobre una población que evoluciona en paisajes de aptitud simples y analíticamente tratables, observando comportamientos que no son explicables en términos del “escalado de cimas” o “hill-climbing”.

Considerando primero sistemas de uno y dos bits. Sea un sólo gen con dos alelos, 0 y 1, los cuales tienen la misma aptitud reproductiva, f , i.e. el mapeo genotipo-fenotipo es degenerado de grado 2. En la ausencia de mutación las aptitudes reproductivas y efectivas son las mismas. En el límite de población infinita o en el promedio del caso de población finita, $\Delta P(t) = P(1, t) - P(0, t)$ es dependiente del tiempo. Por lo tanto, cualquier desviación de la homogeneidad en la población inicial será preservada. Para una tasa de mutación no cero cualquier inhomogeneidad inicial será eliminada por efectos de las mutaciones. Entonces, si $\Delta P > 0$ la aptitud efectiva del alelo 0 es mayor que la del alelo 1 hasta que la desviación es eliminada. Así, el efecto de las mutaciones es llevar el sistema al equilibrio, i.e. dentro del estado de población homogénea. Durante este acercamiento al equilibrio el alelo menos numeroso, 0, es elegido más que el alelo 1 en tanto que dan más descendencia, aún cuando 0 y 1 tienen la misma aptitud reproductiva.

Si las tasas de mutación para cambiar el alelo 1 al alelo 0 y para cambiar del alelo 0 al alelo 1 no son iguales pero son p_1 y p_2 respectivamente entonces las diferencias entre la aptitud reproductiva y la aptitud efectiva son aún más pronunciadas como

lo muestra

$$\Delta P(t+1) = (1 - 2p_1)\Delta P(t) + 2(p_2 - p_1)P(1, t) \quad (4.4.1)$$

que se obtiene considerando el modelo de población infinita con las ecuaciones

$$P(1, t+1) = (1 - p_1)P(1, t) + p_2P(0, t)$$

$$P(0, t+1) = (1 - p_2)P(0, t) + p_1P(1, t)$$

En este caso $\lim_{t \rightarrow \infty} \Delta P(t) \rightarrow ((p_2 - p_1)/(p_1 + p_2))$. Así se observa que la población fluye de 0 a 1 aún cuando se piensa que no hay un gradiente de aptitud reproductiva en esa dirección.

Para este modelo, en la ausencia de mutaciones se tiene, $p = 0$, $f_{eff}(c_i, t) = f(c_i) = f$, $\forall i = 0, 1$. Para $p \neq 0$

$$f_{eff}(c_i, t) = ((1 - 2p) + (p/P(c_i, t))) f \quad (4.4.2)$$

Con esto se nota que cuando la población es homogénea la aptitud efectiva es la misma que la aptitud reproductiva. Las desviaciones en esta homogeneidad resultan en una mayor aptitud efectiva para un número menor de genotipos. La aptitud efectiva disminuye conforme se acerca al equilibrio.

El paisaje de aptitud efectiva para este modelo cuando las tasas de mutación de 1 a 0 y de 0 a 1 son diferentes se muestran en la figura 4.1. En este caso, $f = 2$, $p_1 = 0.15$, $p_2 = 0.1$ y $P(0, 0) = P(1, 0) = 0.5$. Observe las diferencias significativas en $f_{eff}(c_i, 0)$ resultado de lo cual hay un flujo poblacional de 1 to 0 a lo largo de este gradiente de aptitud efectiva aún cuando no hay un gradiente de aptitud reproductiva. Este gradiente disminuye de manera monótona como función del tiempo por lo que el paisaje de aptitud efectiva se vuelve plano asintóticamente. Esto será una propiedad

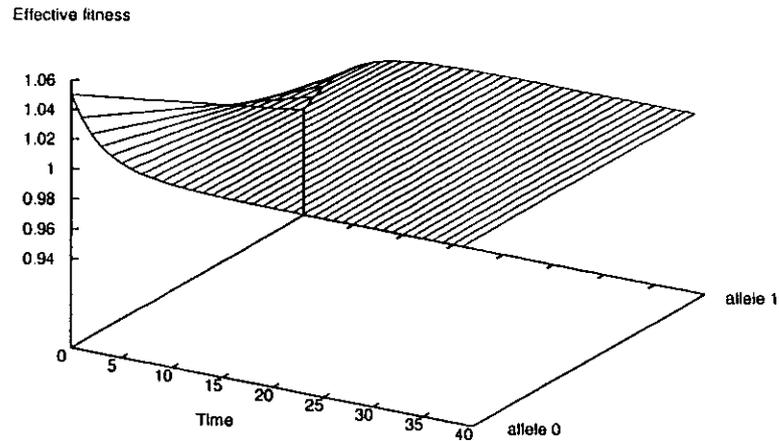


Figura 4.1: Gráfica de la aptitud efectiva como función del tiempo para el modelo de un bit para diferentes tasas de mutación.

de cualquier paisaje de aptitud efectiva, el único estado estable de las ecuaciones de evolución siendo $f_{eff}(c_i, t) = \bar{f}(t) \forall c_i$, or $P(c_i, t) = 0$. Con esto es posible apreciar la utilidad de la aptitud efectiva— indica por su desviación del estado plano que tan cerca esta la población de un estado estable.

Considerando un sistema de dos bits, con dos alelos 0 y 1, con el paisaje de aptitud definido por: $f(00) = f(01) = 1$, $f(11) = 10$, $f(10) = 0.1$. El paisaje de aptitud en este caso es parcialmente degenerado: los estados 00 y 01 tienen el mismo valor de aptitud. Sin embargo, aunque los valores de aptitud reproductiva son iguales, los valores de aptitud efectiva son diferentes. La degeneración en este caso es superada

por el efecto de la mutación como se ve por las ecuaciones

$$P(00, t+1) = \frac{f(00)}{\bar{f}(t)}(1-p)P(00, t) + p^2 \frac{f(11)}{\bar{f}(t)}P(11, t) + \frac{p(1-p)}{\bar{f}(t)}(f(01)P(01, t) + f(10)P(10, t)) \quad (4.4.3)$$

$$P(01, t+1) = \frac{f(01)}{\bar{f}(t)}(1-p)P(01, t) + p^2 \frac{f(10)}{\bar{f}(t)}P(10, t) + \frac{p(1-p)}{\bar{f}(t)}(f(00)P(00, t) + f(11)P(11, t)) \quad (4.4.4)$$

las cuales son equivalentes a la ecuación (3.4.5) en el límite $p_c = 0$.

Para $p < 1/2$, e teniendo una población aleatoria inicial, en términos del número de descendientes la configuración 01 será preferida a 00. La razón es que la configuración más apta es 11 y esta puede mutar más fácilmente a 01 que a 00. Entonces, hay un flujo poblacional de 00 to 01 a pesar del hecho de que no hay un gradiente en el paisaje de aptitud reproductiva que lo induzca. Por el contrario, aún si hubiera un gradiente en la dirección de 01 a 00 si este no fuera muy grande la mutación induciría un flujo de 00 a 01 que lo sobrepasaría, i.e. la población puede fluir hacia arriba contra el gradiente de aptitud reproductiva! Así hay una tendencia del sistema para evolucionar a lo largo de la dirección preferida no por causa de las restricciones de selección sino porque el sistema ha preferido las direcciones en función de las mutaciones *aleatorias*. Este es el fenómeno de ortogénesis definido por Kauffman [81], i.e. cambios genéticos en presencia de mutaciones aleatorias.

La aptitud efectiva de este modelo a tiempo 0 es: $f_{eff}(00, 0) = (1 - 0.9p + 9.9p^2)$ y $f_{eff}(01, 0) = (1 + 9p - 9.9p^2)$ respectivamente, donde las proporciones iniciales de los cuatro estados se consideran iguales. El paisaje de aptitud efectiva se muestra en la figura 4.2 para $t = 0$ con $p = 0.05$, $P(00, 0) = 0.3$, $P(01, 0) = 0.2$, $P(10, 0) = 0.4$ y

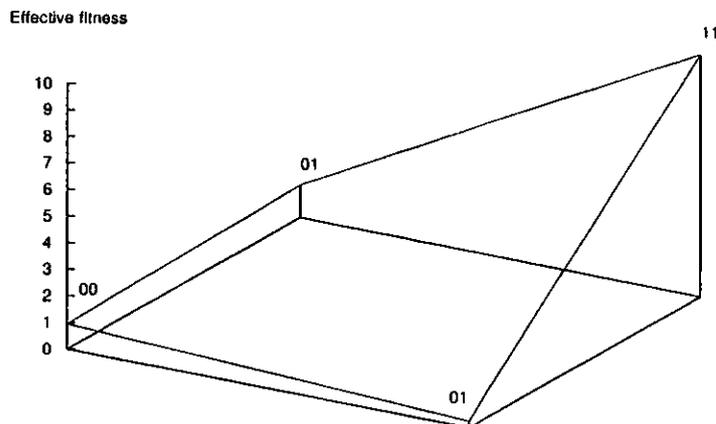


Figura 4.2: Gráfica de aptitud efectiva para en $t = 0$ para el modelo de dos bits.

$P(11, 0) = 0.1$. Note que el gradiente inicial de 00 a 01 que $\rightarrow 0$ como $t \rightarrow \infty$ como se ve en la figura 4.3 el cual muestra el mismo paisaje de aptitud efectiva que el de la figura 4.2 pero en los tiempos finales. Observe la conformación plana del paisaje de aptitud efectiva para los últimos tiempos.

En el caso de un paisaje de aptitud estrictamente plano característico de la evolución neutral se tiene que la aptitud efectiva es

$$f_{eff}(c_i, t) = f \sum_{j=1}^{2^N} \frac{P(c_j, t)}{P(c_i, t)} p^{d_{ij}} (1-p)^{N-d_{ij}} \quad (4.4.5)$$

Para una población homogénea $f_{eff}(c_i, t) = f \forall c_i, t$. Bajo estas circunstancias el paisaje de aptitud efectiva es tan plano como el normal. Pequeñas desviaciones de la homogeneidad se manifestarán en pequeñas arrugas del paisaje de aptitud efectiva las cuales disminuirán conforme la población se homogeneiza. Si el paisaje tiene sólo un subespacio plano el sistema tratará de escapar por la dirección con el mayor gradiente

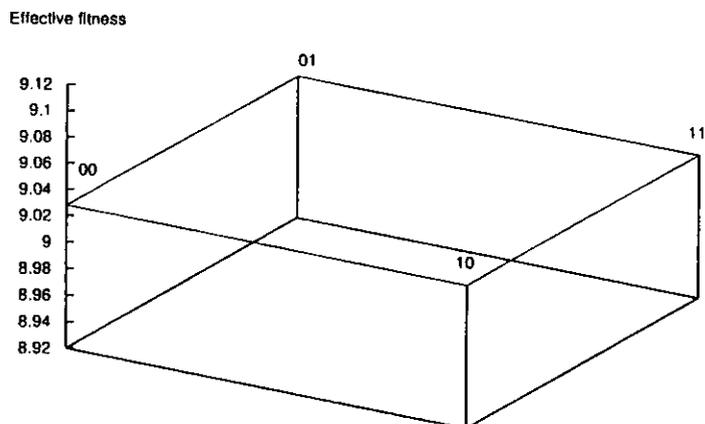


Figura 4.3: Gráfica de aptitud efectiva en los últimos tiempos para el modelo de dos bits.

de aptitud efectiva.

Arriba se ha considerado sólo el efecto de mutación para apoyar la suposición de que la aptitud reproductiva y la aptitud efectiva pueden ser muy diferentes en presencia de otros operadores genéticos. Consideraciones similares se aplican a la recombinación. Tomando el ejemplo extremo de esto se considera al siguiente sistema de dos bits en presencia de selección y recombinación sin mutación, con el paisaje de aptitud: $f(01) = f(10) = 0$, $f(11) = f(00) = 1$. La solución para el estado estable de la ecuación de evolución (3.4.5) es $P(11) = P(00) = \frac{1}{2} (1 - \frac{p_c}{2})$, $P(01) = P(10) = \frac{p_c}{4}$. Para $p_c = 1$ se observa que la mitad de la población en el estado estable esta compuesta por genotipos que tienen aptitud cero. En este caso se observa que la población fluye contra los gradientes de aptitud infinita! Definiendo un gradiente de aptitud entre 11 and 10, por ejemplo, como $(f(11) - f(01))/f(01)$. Esta definición captura el hecho de

que después de una generación sin entrecruzamiento hay cero genotipos del tipo 01 o 10, independientemente de su número en la población anterior. En otras palabras hay una resistencia infinita a su propagación. Note también que este punto fijo en la dinámica poblacional es estable. Para los sistemas de dos bits mencionados antes $P(00,1) = (1 - (9.9p_c/12.1))P(00,0)$, $P(01,1) = (1 + (9.9p_c/12.1))P(01,0)$. Una vez más se observa una degeneración en la aptitud reproductiva que es rota por el efecto de operador de recombinación con el consecuente diferencial en la aptitud de la descendencia.

Otro ejemplo es el paisaje NIAH en presencia de mutación y selección. El paisaje es $f(c_i) = f_0$, $i = opt$, $f(c_i) = f_1$, $i \neq opt$, opt siendo el genotipo óptimo. Es posible usar como medida de orden en la población la concentración relativa del genotipo óptimo, $P(c_{opt}, t)$; y en particular en el límite a largo plazo, $P(c_{opt}, \infty)$, donde un estado estable es alcanzado — el cuasi-especie. En este caso es bien conocido [28] que $P(c_{opt}, \infty)$ disminuye en forma monótona como función de la tasa de mutación p hasta que una tasa crítica, p_{cri} , es alcanzada más haya de la cual $P(c_{opt}, \infty) = 1/2^N$.

Es importante observar que el paisaje de aptitud es constante en su comportamiento. Así, para $p = 0$, i.e. en la ausencia de mutaciones, la población entera alcanza el pico de aptitud hasta $\bar{f}(t) = f_{opt}$, donde para $p \geq p_{cri}$ la población es distribuida de manera uniforme dentro del paisaje. Como se aprecia no es posible intuir los flujos poblacionales a partir del paisaje en sí. En uno de los límites la selección domina, en el otro no tiene efecto. Ya sea que la población alcanza o no el pico de aptitud debido a la selección reproductiva depende del operador genético de mutación.

Lo anterior sirve como un buen ejemplo de rompimiento de simetría inducido. En el paisaje NIAH hay sólo dos fenotipos, Q_0 y Q_{opt} donde el fenotipo Q_0 corresponde

a todos los genotipos c_{opt} . Con esto se tiene una degeneración de dimensión $(2^N - 1)$ asociada con el paisaje de aptitud reproductiva. Por lo que, en términos de la aptitud reproductiva se podría esperar encontrar todos los genotipos excepto el óptimo en iguales proporciones. Sin embargo, el estado estable de la población – el cuasi-especie – exhibe una preferencia por aquellos genotipos sub-óptimos que están más cerca en distancia de Hamming del óptimo. Siendo el efecto de la mutación el superar la degeneración del paisaje. Permaneciendo una simetría residual debida al hecho de que, al menos empezando con una población homogénea, todos los genotipos de una distancia de Hamming dada al óptimo tendrán la misma probabilidad de éxito reproductivo, i.e. la misma aptitud efectiva.

En este caso la aptitud efectiva de la cuerda óptima es

$$f_{eff}(c_{opt}, t) = f_0(1-p)^N + f_1 \sum_{C_j \neq c_{opt}} \frac{P(C_j, t)}{P(c_{opt}, t)} p^{d_{optj}} (1-p)^{N-d_{optj}} \quad (4.4.6)$$

En la figura 4.4 se muestra se grafica la aptitud efectiva del genotipo óptimo del modelo de Eigen en el estado estable como función de p . En este caso $f_0 = 10$ y $f_1 = 1$. Para $p = 0$, $f_{eff}(c_{opt}, \infty) = f_0$ i.e. las aptitudes reproductiva y efectiva son las mismas. Observe como el umbral de error se manifiesta en términos de la aptitud efectiva — en y por arriba del umbral $f_{eff}(c_{opt}, t) \rightarrow \bar{f}(t) \approx f_1$. i.e. una vez más el paisaje de aptitud efectiva será plano. Así la aptitud efectiva puede servir como un parámetro de orden para distinguir el régimen de selección dominante al del régimen dominado por la mutación, i.e. selección fuerte de selección débil.

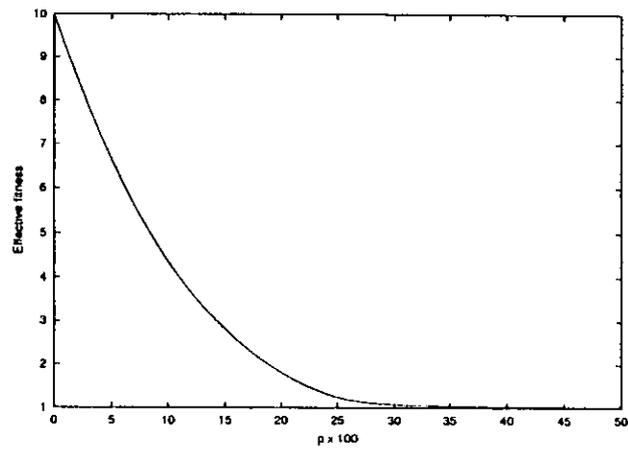


Figura 4.4: Gráfica de aptitud efectiva vs p ($\times 100$) en el límite del estado estable para el paisaje NIAH.

Capítulo 5

Experimentos y resultados

El rompimiento en la simetría de los sinónimos genéticos debido a la acción de operadores genéticos como la mutación pueden aumentar la adaptabilidad de los individuos a cambios en el ambiente y en particular aumentar su robustez. Para el caso del modelo de virus mostrado en este capítulo, el cambio en la codificación del codón (conjunto de tres nucleótidos) del epítopo de neutralización mejora la habilidad del virus para generar mutantes que evaden la respuesta del sistema inmune. La ventaja selectiva de una alta mutabilidad lleva a la dominancia de codones que favorecen mutaciones hacia codones no sinónimos.

Otro ejemplo de esto es el modelo presentado de una red regulatoria genética que lleva a la evolución adaptiva en una población de jirafas por medio de un rompimiento de simetría inducido más que a través de cualquier ventaja selectiva.

Los modelos utilizados (virus y jirafas) muestran la emergencia espontánea de estructura en la producción de mutantes a pesar de la naturaleza aleatoria y no direccional de las mutaciones puntuales a nivel genotipo. Esto muestra que el proceso de autoorganización mejora la habilidad de los sistemas para adaptarse a cambios en el ambiente. En el caso del virus el promedio de la mutabilidad propia con un período

largo de vida *in vivo* esta por arriba de la tasa de mutación promedio reflejando el dominio de codones más capaces en el cromosoma.

Por otra parte, se hace un análisis experimental y teórico de los efectos de la autoadaptación en sistemas evolutivos sencillos. Específicamente es estudiada a través de la codificación de las tasas de mutación y cruza dentro del cromosoma para un algoritmo genético optimizando sobre ciertos paisajes de aptitud. Dado que este mapeo es degenerado los operadores genéticos inducen un rompimiento de simetría hacia los genotipos que tiene un mayor potencial evolutivo permitiendo al sistema autoadaptarse en un ambiente dependiente del tiempo. También se mencionan los resultados que demuestran el efecto del rompimiento en esquemas de orden 2, al favorecer la reproducción de los esquemas de mayor tamaño.

Como último experimento se presentan los resultados de usar un algoritmo genético inteligente para el manejo de un portafolio de inversión compuesto por acciones de las emisoras TELMEX L y BANACCI O las que cotizan en la Bolsa Mexicana de Valores.

5.1 Dependencia de tamaño para bloques constructores

En este apartado se muestra un ejemplo donde existe la preferencia por esquemas grandes, debido a su aptitud efectiva. En contraste con la hipótesis de bloques constructores que manifiesta lo contrario.

Se considera la evolución de esquemas de orden 2 (al que se hará referencia como esquemas-2) con varias longitudes de definición en cuerdas de tamaño 8 para el caso de paisajes del tipo “contando unos”, donde la aptitud del individuo dependen del

número de 1 en la cuerda, en presencia de selección y recombinación pero sin mutación [92]. En ausencia de entrecruzamiento no hay una preferencia para un esquema de una longitud u otra, i.e. la aptitud efectiva y la aptitud reproductiva para esquemas-2 de longitudes diferentes es la misma.

Para analizar esta situación considere la cantidad $M(l)$ donde $M(l) \equiv (n_{opt}(l) - n_{opt}(8))/n_{opt}(8)$. Aquí $n_{opt}(l)$ es el número de esquemas-2 óptimos con longitud de definición l normalizado por el número total de esquemas-2 de longitud l por cuerda, i.e. $9 - l$. Por esquemas-2 óptimo se entiende que al esquema que contiene al óptimo global 11. $n_{opt}(8)$ es el número de esquema-2 óptimo con longitud de definición 8. Considerando un paisaje de aptitud, el cual es subespacio del paisaje de aptitud completo, que grafique a la aptitud como función del promedio de esquemas estáticos para las siete direcciones que corresponden al esquema óptimo, 11*****,...,1*****1. En este caso el paisaje es plano, i.e. hay una degeneración de grado 7 en el paisaje, en otras palabras para cada fenotipo de esquema-2 le corresponden 7 genotipos de esquema-2 todos con la misma aptitud. Así, para cada esquema-2 óptimo hay siete genotipos óptimos de esquema-2.

Se consideró una población de tamaño 5000 con individuos de 8 bits. La Figura 5.1 muestra un promedio sobre 30 diferentes experimentos de $M(l)$ vs tiempo con $p_c = 1$. Como se mencionó, sin entrecruzamiento no hay preferencia por esquemas de un tamaño determinado, cualquier diferencia experimental es resultado de ruido estadístico. La inclusión de entrecruzamiento lleva a un cambio significativo: la prevalencia del esquema se ordena monótonamente con respecto a la longitud pero favoreciendo a los esquemas grandes. Así, aunque no hay preferencia en términos de aptitud reproductiva para la longitud de un esquema u otro, no sucede lo mismo

respecto a la aptitud efectiva. Se observa que, contrario a la hipótesis de bloques

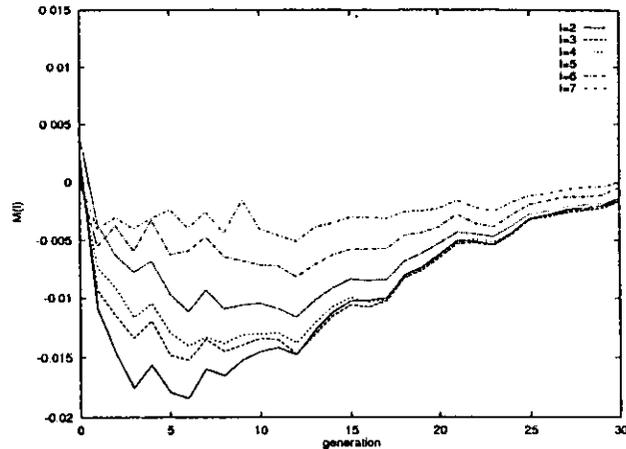


Figura 5.1: Gráfica de $M(l)$ vs t en modelo unitario con $p_c = 1$.

constructores, hay una preferencia reproductiva para esquemas grandes. Es decir, hay flujos poblacionales en el paisaje de aptitud reproductiva a lo largo de direcciones planas, con el entrecruzamiento rompiendo la degeneración genotipo-fenotipo.

En la gráfica que se presenta a continuación (Figura 5.2) se muestra $F(l) \equiv (f_{eff}(l) - f_{eff}(8))/f_{eff}(8)$, donde $f_{eff}(l)$ es la aptitud efectiva del esquema-2 óptimo de tamaño l y $f_{eff}(8)$ es la cantidad análoga para el esquema-2 de tamaño 8. Observe que la aptitud efectiva de los esquemas grandes es mayor que la de los pequeños para las primeras 6 generaciones aproximadamente. Después de 6 generaciones las curvas en la Figura 5.1 empiezan a converger lo que coincide con que la aptitud efectiva es mayor para los esquemas pequeños. Se puede pensar que la aptitud efectiva es una medida de los gradientes de las curvas en la Figura 5.1.

Se observa un tipo de rompimiento de simetría inducido dado que en términos del promedio de los esquemas estáticos no hay preferencia entre esquemas-2 óptimos de un tamaño determinado. El entrecruzamiento rompe la simetría dando una ventaja

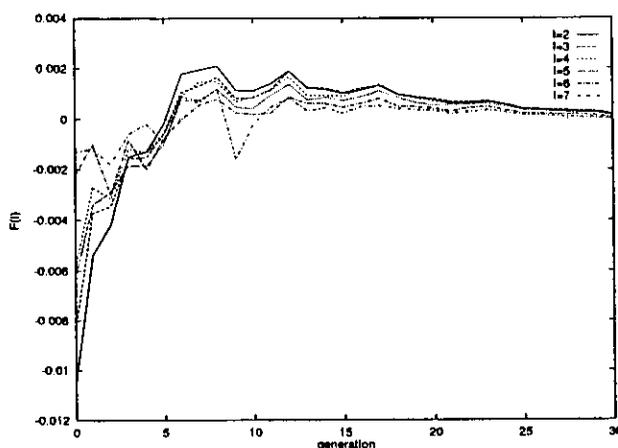


Figura 5.2: Gráfica de $F(l)$ vs tiempo para el modelo unitario de la Figura 5.1.

reproductiva a esquemas grandes ya que pueden ser fácilmente creados usando esquemas óptimos de orden 1. La aptitud efectiva mide la fuerza de este rompimiento de simetría inducido.

5.2 Autocodificación de tasas de mutación y cruza

La autoadaptación apareció primero en el contexto de las estrategias evolutivas [93] y ha sido estudiado por Bäck y colaboradores [94] tanto en el contexto de algoritmos genéticos como de estrategias evolutivas. Uno de los puntos de interés fue la derivación de tasas de mutación óptimas en una cuerda por cuerdas bases en el caso de ciertos paisajes sencillos. No se consideró la codificación de la probabilidad de entrecruzamiento (en [95] se hace un primer intento considerando un paisaje no epistático).

En un algoritmo genético es de principal importancia el identificar cuáles son los valores óptimos para las tasas de mutación y cruza de ahí que mucha de las investigaciones están dirigidas en este sentido [42], [96], [97]. Sin embargo se ha

demostrado [98] que las tasas óptimas dependen de la función específica a optimizar.

Para ilustrar el fenómeno de cómo el rompimiento de simetría se presenta en algoritmos genéticos con parámetros codificados que redundan en un mejor desempeño comparado con un algoritmo genético de parámetros fijos se eligieron diferentes paisajes de aptitud: una función genérica multimodal no deceptiva; una función deceptiva y una función dependiente de tiempo. El método de selección utilizado es el de la rueda de ruleta para los casos donde no se menciona.

La codificación de parámetros se hace extendiendo el tamaño del cromosoma aumentando un segmento que corresponde a la tasa de mutación y otro a la tasa de cruce. Siguiendo con el concepto de “simetría de sinónimos” tenemos que para cada segmento que codifica al individuo en el paisaje de aptitud tiene asociados una gran variedad de probabilidades de mutación obteniendo con ello el mapeo genotipo-fenotipo degenerado que permitirá estudiar el rompimiento de simetría inducido. Considerando que se tienen N_c bits para codificar la cruce y N_m bits para la mutación la degeneración del mapeo genotipo-fenotipo es $2^{N_c+N_m}$.

La función multi-modal simple se muestra en la Figura 5.3. El diseño de la función fue mediante la asignación de un valor a cada entero entre 0 y 63. Esta función es no deceptiva ya que el entrecruzamiento entre las cuerdas óptimas o próximas al óptimo no producen cuerdas poco aptas, i.e. el entrecruzamiento de las cuerdas cerca de 10 o 40 no tienden a producir cuerdas entre 20 y 30, o entre 50 y 63. El tamaño de la población es de 500 individuos. El cromosoma básico es una cuerda de 6 bits con codificación binaria representando los enteros entre 0 y 63. La codificación de las tasas de mutación y/o cruce fueron de 3 hasta 16 bits además dependiendo de si las tasas eran o no implementadas o de si se ocupaban 3 u 8 bits para representarlas.

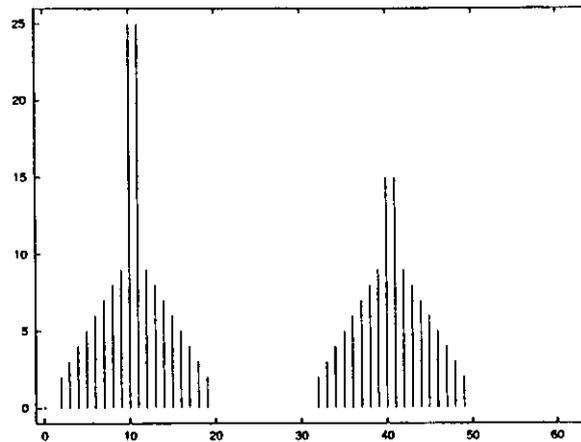


Figura 5.3: Gráfica de un paisaje multimodal vs genotipo. Para una cuerda binaria de 6 bits se tienen 63 genotipos posibles. En el eje y se presenta la aptitud de cada genotipo.

El usar mayor número de bits permitió una selección más fina en las tasas de los operadores.

El segundo paisaje utilizado para investigar se muestra en la Figura 5.4. El paisaje es engañoso en el sentido que el entrecruzamiento entre cuerdas aptas asociados con los óptimos en $000000 = 0$ y $111111 = 63$ producen cuerdas poco aptas.

El tercer paisaje elegido incluye la dependencia temporal. El paisaje inicial se muestra en la Figura 5.3 que tiene un óptimo global en 10 y 11. Después de que el 60% de la población alcanza el óptimo global el paisaje cambia al mostrado en la Figura 5.5 donde el óptimo global original es sólo un óptimo local y un nuevo óptimo global aparece en 62. Este tipo de paisaje se le denominó "jumper landscape". Para los experimentos con este tipo de paisaje se usó selección por torneo de tamaño 5 y el valor mínimo para la probabilidad de mutación fue 0.005.

Con el fin de comparar resultados de experimentos con parámetros fijos contra parámetros autocodificados se hicieron 50 pruebas para cada par de parámetros fijos

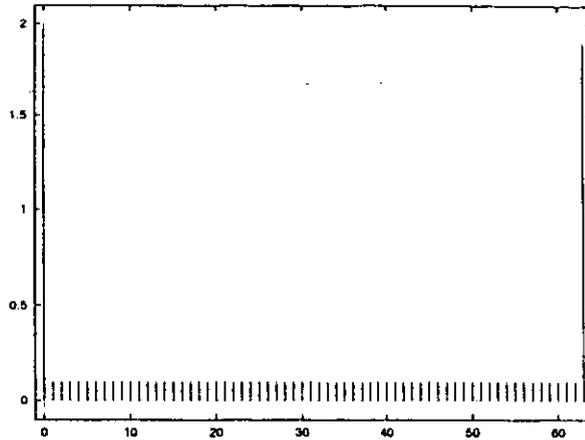


Figura 5.4: Gráfica de aptitud vs genotipo para un paisaje “deceptivo” (el entrecruzamiento de los óptimos 0 y 63 producen individuos poco aptos).

p y p_c . p_c fue cambiando entre 0 y 1 con incrementos de 0.1 mientras que p varió entre 0 y 0.1 con incrementos de 0.01. También se experimentó con $p=0.003$. Por lo que para cada paisaje se hicieron pruebas para 121 puntos en el espacio de parámetros. El desempeño del algoritmo genético con parámetros fijos fueron comparados con los de los algoritmos genéticos autocodificados. Para este último sólo se consideraron codificaciones de 8 y 3 bits para los parámetros. El análisis fue hecho principalmente sobre desempeño por cada generación.

Los paisajes anteriores fueron elegidos para ilustrar claramente las ventajas de la codificación de los parámetros en el contexto de un óptimo claramente identificado.

Para el caso de los paisajes de las Figuras 5.3-4 en términos del desempeño offline (mejor óptimo en el momento) los cromosomas con altas tasas de mutación son preferidos al permitir escapar de los óptimos locales, pero el desempeño on-line (aptitud promedio de la población) es bajo. Enfatizando el desempeño online sin prejuicio para el offline se encontró que al promediar sobre muchos experimentos tasas de 0.01

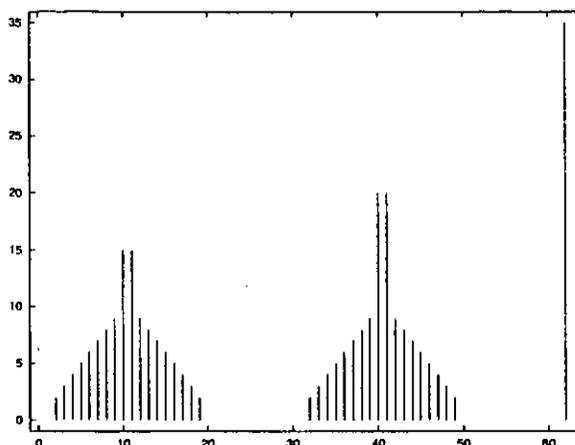


Figura 5.5: Gráfica de aptitud vs genotipo después de un cambio (una vez que el 60% de la población alcanza el óptimo) en la Figura 5.3

para mutación y 0.8 para cruza daban los mejores resultados, siendo estos los mejores valores recomendados en la literatura [42],[96],[97]. Un algoritmo genético con estos valores fue utilizado para comparar a los algoritmos genéticos autocodificado. El análisis para las Figuras 5.3-5.5 se promediaron los resultados de 30 experimentos.

En la Figura 5.6 se muestran los resultados de varios algoritmos genéticos con el paisaje de la Figura 5.3. La gráfica muestra en la mitad superior la frecuencia relativa para la cuerda óptima como función del tiempo y en la mitad inferior los valores promedio de los parámetros codificados del algoritmo genético como función del tiempo. La población inicial fue generada al azar. El tamaño de la población es grande comparado con el del espacio de estados ya que la intención fue la de investigar los efectos de la codificación de parámetros sin considerar los efectos de tamaño finito debido a errores muestrales. Lo más notable de la Figura 5.6 es el comportamiento de los parámetros codificados. Para las mutaciones se observa que el sistema empieza a “enfriarse” por si mismo conforme la población comienza a

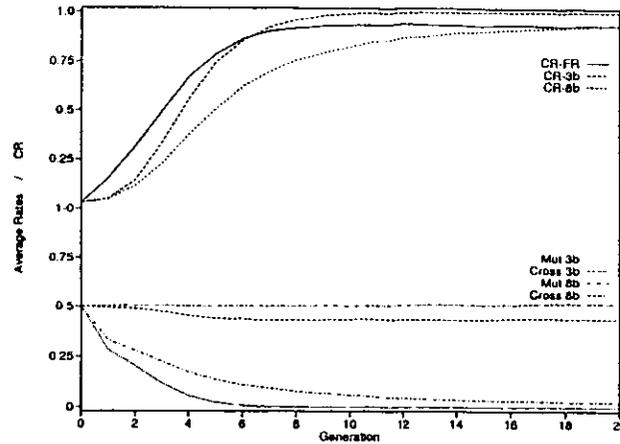


Figura 5.6: Gráfica que muestra los resultados de la comparación entre parámetros fijos y autocodificados para algoritmos genéticos. La gráfica superior muestra la concentración relativa (CR) de las cuerdas óptimas como función del tiempo. CR-FR es la curva para un AG con tasas fijas, CR-3b para codificación con 3 bits y CR-8b para codificación con 8 bits. En la gráfica inferior se muestra el promedio de las probabilidades de mutación y cruce como función del tiempo para el algoritmo genético autocodificado. Mut3b, Mut 8b, Cross 3b y Cross 8b son las tasas de mutación y entrecruzamiento promedio para las codificaciones de 3 y 8 bits. La población inicial fue elegida aleatoriamente.

ordenarse. Es claro que no hay una ventaja selectiva en cualquier generación para una mutación u otra, sin embargo, se aprecia claramente los efectos del rompimiento de simetría sobre las cuerdas con bajas tasas de mutación pueden producir mejores descendencias. Este rompimiento de simetría se vuelve más pronunciado como función de tiempo. Con respecto a la cruce se nota que no hay beneficio selectivo de una tasa de cruce vs otra. Comparando con el algoritmo genético con parámetros óptimos se observa que una codificación de 8 bits toma más tiempo tratando de encontrar las tasas de mutación y cruce óptimas ya que tiene que buscar en un espacio más grande. Observe que la población en el estado estable para una tasa de mutación codificada será superior a la de la tasa fija porque la población sólo puede estar estrictamente

ordenada cuando $p=0$, i.e. un algoritmo genético codificado eventualmente tiende a un mejor desempeño online.

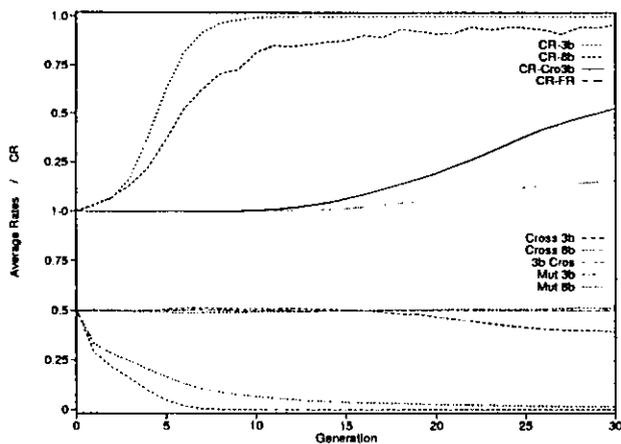


Figura 5.7: Misma que la Figura 5.6 pero con la población inicial concentrada en el punto 49 del paisaje de la Figura 5.3. CR-Cro 3b representa a un algoritmo genético donde la tasa de mutación es fija en 0.01 y la tasa de cruce se codifica con 3 bits. Cro 3b corresponde a la tasa de cruce promedio

En la Figura 5.7 se muestra que pasa con el paisaje de la Figura 5.3 pero en el caso de que toda la población inicial esta concentrada en el punto 49. La razón de esto es que en un paisaje dependiente del tiempo la población puede converger a un punto en específico pero al presentarse un cambio en el paisaje el punto anterior se convierte en óptimo local y el sistema debe encontrar el nuevo óptimo. Si el paisaje cambia sólo cuando la población se ha ordenado respecto a un punto entonces en términos de evolución en el nuevo paisaje el sistema empieza con una condición inicial especial. Esta capacidad de los algoritmos genéticos inteligentes motiva su aplicación a problemas reales como el manejo de contingencias en redes de comunicaciones o de transporte, calendarización de actividades o procesos todos ellos problemas tratados por la investigación de operaciones.

Considerando los resultados explícitos se nota que el desempeño de los algoritmos genéticos de parámetros fijos es particularmente malo debido a que el entrecruzamiento no actúa de manera eficiente para favorecer la diversificación cuando se inicia con una población ordenada. Entonces la mutación debe jugar este papel, si la probabilidad de mutación es baja entonces se lleva mucho tiempo para encontrar el óptimo. Es claro que codificar la tasa de cruce y no la de mutación no ayuda mucho. Al codificar la mutación los resultados son mucho mejores. Una vez más se ve como el sistema se enfría por si mismo después de que el algoritmo genético ha alcanzado el óptimo. El operador de cruce ayuda en la búsqueda del nuevo óptimo y sólo en el caso de que la población esta totalmente ordenada su efecto es neutral.

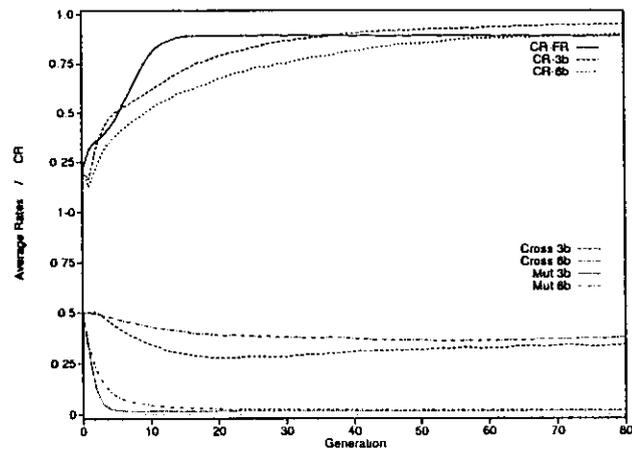


Figura 5.8: Mismos datos que para la Figura 5.6 pero para el paisaje de la Figura 5.4. Población inicial sesgada

La Figura 5.8 muestra los resultados asociados con el paisaje deceptivo de la Figura 5.4. La población inicial fue sesgada al colocar 100 individuos en cada óptimo para hacerlo más deceptivo. En varios de los experimentos los algoritmos genéticos con parámetros óptimos fueron incapaces de mantener a la población en el óptimo global.

El algoritmo genético codificado con 8 bits, en contraste, aumentó sin problemas la concentración relativa del óptimo global. Al usar una codificación de 3 bits algunas veces alcanzó el óptimo global. Esto con 30 experimentos. Una vez más el sistema se enfría por sí mismo. El resultado más interesante es la tendencia a disminuir de la tasa de cruza. Sin embargo se aprecia un cambio inicial muy marcado. Esto es debido al efecto deceptivo, la cruce de las cuerdas 0 y 63 producen cuerdas poco aptas y como el sistema inicia con una elevada cantidad de ellas son preferidas las que tienen la menor tasa de cruza. Conforme el sistema comienza a ordenarse cerca del óptimo global el entrecruzamiento pierde su efecto destructivo y la tasa aumenta.

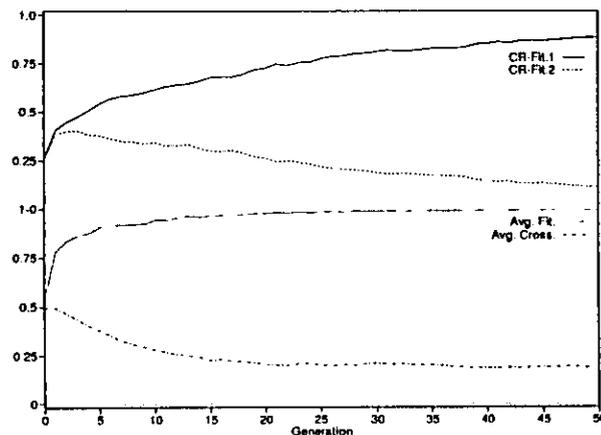


Figura 5.9: La gráfica superior muestra la concentración relativa de las probabilidades de cruza y en la gráfica inferior se presentan las probabilidades de cruza promedio. Ambas como función del tiempo para el paisaje de la Figura 5.4, con 80 individuos por población y una probabilidad de mutación con valor cero. CR-FIT 1 es la concentración relativa del óptimo global en el punto 0, CR-Fit2 es la concentración relativa del óptimo local en el punto 63, Avg Cross es la tasa de cruza promedio y Avg Fit es la aptitud promedio de la población.

La Figura 5.9 contiene los resultados para el paisaje deceptivo cuando $p=0$. En este caso se empezó con una población inicial de 80 individuos, 20 de los cuales están

localizado en un óptimo, 20 en el otro y los restantes 40 distribuidos aleatoriamente. Una vez más debido a las grandes poblaciones asociadas con los dos óptimos, la cruce que incluye cuerdas de ambos óptimos resultará en una descendencia poco apta.

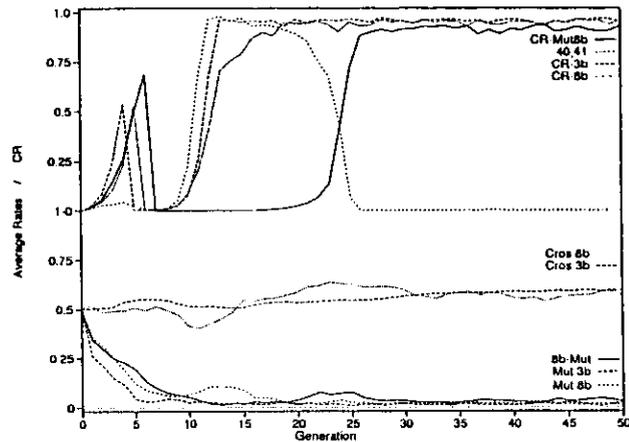


Figura 5.10: Gráfica de la concentración relativa del óptimo (gráfica superior) y probabilidades promedio de cruce y mutación (gráfica inferior) como función de tiempo para el paisaje “jumper” que cambia del paisaje de la Figura 5.3 al de la Figura 5.5 después de que el 60% de la población ha llegado al óptimo de la Figura 5.3. CR-3b y CR-8b son los resultados para los algoritmos genéticos con tasas autocodificadas de 3 y 8 bits, CR-Mut8b es el resultado con sólo la mutación codificada y una probabilidad de cruce de 0 con 40 y 41 siendo la concentración relativa de las cuerdas asociadas con los óptimos locales 40 y 41. Mut 3b, Mut 8b, Cross 3b y Cross 8b son los promedios de las probabilidades para las codificaciones con 3 y 8 bits. La línea para 8b-Mut es la tasa de mutación promedio para el caso de una probabilidad de cruce de 0.

En la Figura 5.10 se observa lo que pasa en el “jumper landscape”. En esta gráfica se muestra el desarrollo de uno de los experimentos. Las curvas superiores muestran las frecuencias relativas del óptimo usando 8 y 3 bits y lo que pasa cuando $p_c=0$ y cuando solamente la mutación es codificada. Hay varios resultados notables: primero el algoritmo genético con tasas fijas no fue capaz de encontrar el nuevo óptimo mientras que el algoritmo genético codificado no tuvo problemas. Para el caso de $p_c=0$

la curva 40,41 muestra la frecuencia relativa de las cuerdas asociadas con el óptimo en 40 y 41. Antes de que el paisaje cambie este es un óptimo local siendo menos apto que el óptimo global en 10 y 11. Después del salto este es más apto pero menor que el óptimo global 63 que es un sólo punto.

Con esto se ve que el óptimo fue encontrado en un proceso de dos pasos después de que el paisaje cambia. Primero las cuerdas encuentran el óptimo en 40 y 41 antes de moverse al verdadero óptimo global, 63. Inmediatamente después del cambio la población efectiva en el nuevo óptimo es aproximadamente cero. El número de cuerdas asociados con 40 y 41 empieza a crecer a expensas de las cuerdas 10 y 11. En su máximo, el número de cuerdas óptimas es todavía limitado. Conforme pasa el tiempo el algoritmo genético identifica al óptimo global y este empieza a crecer a expensas del resto de las cuerdas. Un resultado notable se encuentra al comparar los cambios en las frecuencias relativas con los cambios en la tasa promedio de mutación, especialmente en el caso $p_c=0$, observándose una alta correlación. Primero, mientras la población se ordena por si misma en el óptimo original, hay una selección efectiva contra altas tasas de mutación la cual se puede observar en la disminución de la tasa de mutación. Después del cambio hay un incremento notable en la probabilidad de mutación en el caso de la codificación con 3 bits. Ya que el óptimo global es un sólo punto aislado es más fácil el encontrar cuerdas asociadas con 40 y 41. En este momento la población esta concentrada en este óptimo local y se empieza a enfriar hasta descubrir que este no es el óptimo global por lo que el sistema se vuelve a calentar para tratar de encontrar este nuevo óptimo, a diferencia del recocido simulado donde el calendario de enfriamiento debe modificarse externamente. Este incremento en la tasa de mutación es lo más notable cuando $p_c=0$. Es claro que hay un pequeño retraso

entre los cambios en la población y los cambios en la tasa de mutación. Esto se esperaba dado que no hay una ventaja selectiva en una determinada generación para una tasa particular de mutación. La ventaja selectiva de un genotipo más mutable sobre otro sólo se puede obtener mediante un mecanismo de retroalimentación. Es precisamente esta retroalimentación la que es descrita y medida por la función de aptitud efectiva. Dicha retroalimentación es la que permite al algoritmo genético autoorganizar la búsqueda del nuevo óptimo, un fenómeno análogo resultado de aplicar este concepto a la técnica tradicional de recocido simulado implicaría la capacidad del sistema para autoconfigurar un calendario de enfriamiento.

La tasa de mutación promedio también crece debido a otro efecto que condiciona el óptimo global a tasas elevadas de mutación que después crecerá debido a su ventaja selectiva. Entonces las cuerdas con valores altos de p dominan de manera natural la evolución temprana del óptimo global. Después de encontrar el óptimo global es una desventaja el tener una tasa elevada de mutación por lo que las cuerdas con bajas tasas dominan en la población.

Otro experimento para demostrar la utilidad de los algoritmos genéticos autocodificados consiste en emplear un paisaje del tipo "contando unos" que cambia estocásticamente a un paisaje de "contando ceros" (la aptitud del individuo es igual al número de ceros que contiene). En otras palabras ciertas generaciones fueron elegidas aleatoriamente (generaciones 100, 200, 210, 220, 230, 240, 400, 700 y 900) donde se cambió el paisaje de contando unos a contando ceros. En cada uno de los experimentos los cambios se hicieron en las mismas generaciones.

La comparación se hace entre cromosomas con tasas de mutación y cruza codificadas y cromosomas con tasas fijas. Para los últimos los valores elegidos fueron

$p = 0.01$ y $p_c = 0.8$. Para los cromosomas con tasas codificadas se emplearon 9 bits para codificarlas. Se emplearon 8 bits para representar la aptitud reproductiva. Los resultados representan promedios sobre 10 diferentes experimentos con una población de tamaño 500. La tasa de mutación mínima para las cuerdas codificadas fue de 0.001. En la figura 5.11 se muestra la aptitud promedio de la población (online fitness) como función del tiempo. La curva punteada representa a la población codificada. Tanto los algoritmos genéticos codificados como los no codificados fueron capaces de seguir los cambios en el paisaje y alcanzar una composición poblacional asociada con el cuasi-especie centrada en el óptimo global 8. Observe que el desempeño online de los algoritmos genéticos codificados es siempre superior al de los no codificados ya que el primero se puede auto-organizar de manera que la mutación promedio para cuerdas con todos 1s es menor que la mutación promedio para la población entera, o para aquella de las cuerdas no codificadas. Para una frecuencia alta de cambios en el paisaje ningún algoritmo fue capaz de seguir los cambios. Esto es los cambios en el paisaje fueron tales que la exploración no podía llevar a una fase de explotación.

En la Figura 5.12 se muestran las tasas de mutación y cruza promedio en la población codificada. La curva inferior corresponde a la tasa de mutación. Note la coincidencia exacta entre los cambios grandes en la tasa de mutación y en el paisaje, la primera cambiando en más de un orden de magnitud en respuesta a los cambios en el paisaje. Es claro que hay un alto grado de auto-organización en el mapeo genotipo-fenotipo presente en este modelo. Como se ha enfatizado previamente, en términos de aptitud reproductiva no debería haber preferencia por una u otra tasa. Es claro que el grado de correlación en el paisaje cambia por lo que la tasa de cruza es pequeña.

En la gráfica 5.13 se muestra la aptitud efectiva para este modelo como función

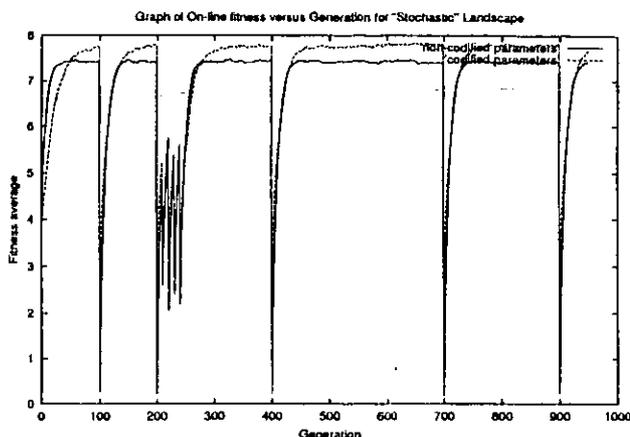


Figura 5.11: Gráfica de la aptitud promedio poblacional (online fitness) vs tiempo para un paisaje dependiente del tiempo de manera estocástica. La curva continua corresponde al algoritmo con parámetros fijos mientras que la curva punteada corresponde al algoritmo con parámetros codificados

del tiempo. Se observa que la aptitud efectiva de las cuerdas codificadas es mayor que la de las no codificadas. Los cambios drásticos en las tasas de las cuerdas codificadas permiten a la población codificada adaptarse rápidamente a los cambios en el paisaje.

5.3 Modelo del virus

En esta sección se presenta un modelo de “juguete” para ilustrar las ideas mencionadas anteriormente en la sección 4.3 (rompimiento de simetría). El modelo representa un virus, representado por un epitope de 6 “aminoácidos”. Existen tres posibles aminoácidos (a, b, c) en cada posición representada por un “codón” de tres bits. Por lo que el genotipo es una cuerda de 18 bits ($6 \cdot 3$). Los posibles valores para cada “nucleótido” (bit en la cuerda) son 0 o 1 por lo tanto hay 2 bases y ocho posibles nucleótidos (2^3). El número total de posibles genotipos es $2^{18} = 262144$. El fenotipo es una palabra de 6 letras, por ejemplo $aabacb$, cada palabra representa a un virus

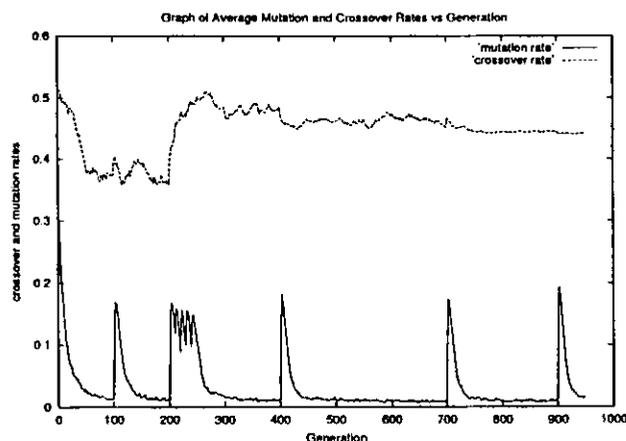


Figura 5.12: Gráfica de las tasa de mutación y cruza promedio vs tiempo para el paisaje estocástico dependiente del tiempo

diferente. El número total de fenotipos es $3^6 = 729$. Como hay 8 posibles codones y sólo tres aminoácidos el mapeo genotipo-fenotipo será degenerado. Una manifestación de esto es la diferencia entre el número total de genotipos y de fenotipos. Si se piensa en dicho mapeo, ϕ , como un “intérprete” entonces ϕ es no-inyectiva. Especificando el intérprete al nivel de codones y aminoácidos tenemos:

$$\phi(000) = \phi(001) = \phi(010) = a$$

$$\phi(011) = \phi(100) = \phi(101) = b$$

$$\phi(110) = \phi(111) = c$$

Cada codón tiene una diferente mutabilidad propia, i.e. hay un número de diferentes aminoácidos que es alcanzado por cualquier mutación puntual dentro del codón. La relación entre codón, aminoácido y mutabilidad propia se muestra en la siguiente tabla:

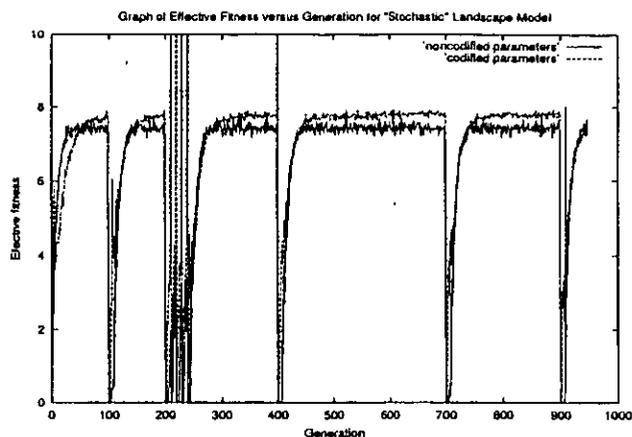


Figura 5.13: Gráfica de la aptitud efectiva vs tiempo para el paisaje “Estocástico” del modelo de la Figura 5.11

Aminoácido	Codón	Mutabilidad propia	Aminoácido alcanzado
a	000	1	b(100)
a	001	2	b(101),b(011)
a	010	2	b(011),c(110)
b	011	3	a(010),b(001),c(111)
b	100	2	a(000),b(101)
b	101	2	a(001),c(111)
c	110	2	a(010),b(100)
c	111	2	b(011),b(101)

Como se puede ver la mayoría de los codones tienen mutabilidad propia 2, el codón 000 tiene mutabilidad propia de 1 y para el codón 011 la mutabilidad propia es 3. La mutabilidad propia del cromosoma es por definición la suma de las mutabilidades propias de sus 6 codones constituyentes. En una población inicial aleatoria la mutabilidad propia es 12 en promedio y se encuentra en un rango de 6 (todos los codones 000) y 18 (todos los codones del tipo 011).

La aptitud, $f_i(n)$, del i -ésimo virus en una generación n es la medida de que tan

bien el virus evade al sistema inmunológico. La aptitud es asociada con el fenotipo por lo que para calcular la aptitud del genotipo es necesario convertirlo a su fenotipo.

Por ejemplo

Genotipo	Número de genotipo	Fenotipo	Número de fenotipo
000010111110110101	12213	aacceb	79
101111000001101001	192617	bcaaba	408

La aptitud de una cuerda particular i se obtiene de la ecuación:

$$f_i(n+1) = f_i(n) - \gamma V_i(n) \quad (5.3.1)$$

donde $V_i(n)$ es la cantidad de virus i presente al tiempo n y γ es una constante que representa el éxito del sistema inmunológico en reconocer al virus. El decaimiento en la aptitud de un virus en proporción a su abundancia es una representación de la acción de los macrófagos que consumen virus y activa células T que reconocen el epítopo específico del virus atacado.

Para la evolución de $V_i(n)$ se usó un operador que reemplaza $V_i(n)$ por la parte entera de $f_i V_i(n) R \zeta$ donde R es un parámetro reproductivo y ζ es un número aleatorio uniformemente distribuido en un intervalo unitario. Después de la selección se implementó la mutación con una probabilidad p por bit, la cual actuada al cambiar el valor del bit dentro del genotipo (cambio de 0 a 1 y viceversa). La probabilidad de mutación es la misma para todo el experimento.

El sistema se inicializa con un sólo virus con aptitud 1. Una "generación" consiste de: evaluación de la aptitud de cada virus, reproducción para cada virus y finalmente mutación. Conforme evoluciona la población el valor inicial en la aptitud de cada virus que no estaba presente con anterioridad se le asigna 1. Cuando la aptitud alcanza 0 significa que el virus no puede sobrevivir más por efecto de las respuesta

del sistema inmunológico y el virus desaparece de la población. Es posible que la desaparición se deba a algún efecto aleatorio dentro del conjunto genético. Un virus que ha sido eliminado en una generación puede reaparecer más tarde a partir de una mutación en un cromosoma diferente, con una aptitud de 1.

Este modelo trata de representar la esencia del fenómeno que se alega existe en la naturaleza [71]. Un virus, identificado con el fenotipo en este modelo, consiste de una cadena de sólo seis aminoácidos que es aproximadamente el tamaño de un epítopo de neutralización. Desde el punto de vista del sistema inmunológico esto es lo que caracteriza a un virus. Cuando la cantidad total de virus excede un determinado número M , la persona infectada muere y la evolución se detiene. Si el total de virus no excede de M en G generaciones el paciente se recupera. Los parámetros particulares para el experimento fueron: $p=0.001$, $\gamma=0.001$, $M=1000$ y $G=3000$. El período largo de existencia se alcanza “debilitando” al paciente cuando su cuenta de virus alcanza un valor determinado y “fortaleciéndolo” cuando esta es muy alta, para evitar tanto la muerte como la cura del paciente. Si el total de virus es mayor a 600 el parámetro de reproducción R , se cambia a 3 y si esta por debajo de 5 cambia a 8. Para valores intermedios el valor es $R=4$. El código fue diseñado para guardar la información de las infecciones que duraron las 3000 generaciones.

El hecho que $\gamma=0.001$ implica que si 1000 copias de un virus particular han sido detectadas en un período de tiempo, la respuesta inducida por el sistema inmune es perfecta y el virus es necesariamente eliminado y su aptitud va a cero. Con el valor 4 para el factor de reproducción el tiempo esperado para que la respuesta inmune elimine la infección es de aproximadamente 10 generaciones; si la infección tiene una duración de 3000 generaciones implica que se observarán cerca de 300 mutaciones.

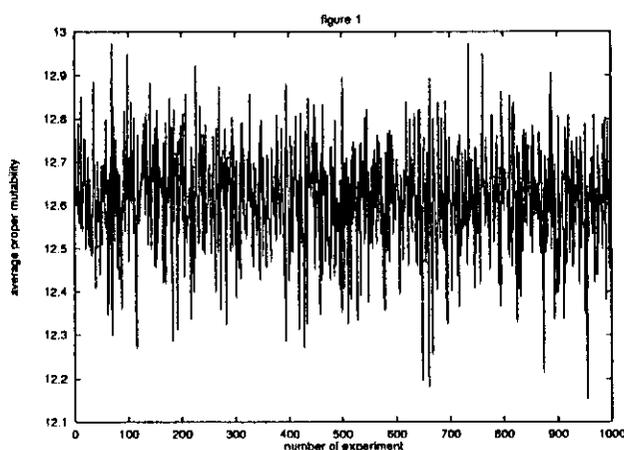


Figura 5.14: Gráfica de mutabilidad promedio sobre 3000 generaciones.

La mutabilidad propia fue promediada durante las 3000 generaciones de cada experimento. En la Figura 5.14 los resultados de 1000 experimentos son representadas. La gráfica muestra que todos los experimentos tuvieron un mutabilidad propia promedio mayor a 12. Esto es evidencia de la retroalimentación del medio donde el ambiente es representado por el sistema inmune de manera que disminuye la aptitud del virus cuando es reconocido. Ya que el virus es constantemente forzado a mutar a nuevas formas, que por definición empieza con una aptitud de 1 para evitar la exterminación, una estrategia es seleccionada donde los codones que mutan mas frecuentemente a objetivos no-sinónimos son preferidos. Como se mencionó arriba esto implica el uso más frecuente del codón 011, con mutabilidad propia de 3 y menos frecuente el codón 000, muestra clara del efecto de rompimiento de simetría.

En la Figura 5.15 se muestra la diversidad promedio de virus durante la infección para cada uno de los 1000 experimentos. En todos los casos la relación de virus presente respecto al total es de menos del 10%. Dado que la mutación actúa en cada gen, si la población crece el número de virus crece también por lo que alta diversidad

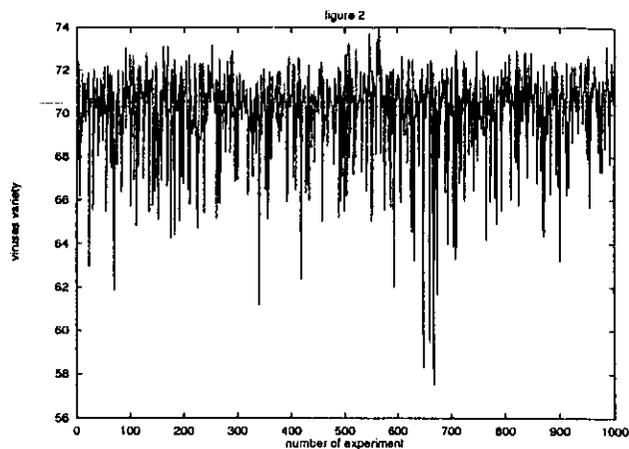


Figura 5.15: Diversidad de virus por experimento.

esta relacionada a un incremento en la población. Las figuras 5.16-5.19 muestran dos experimentos *in extenso*.

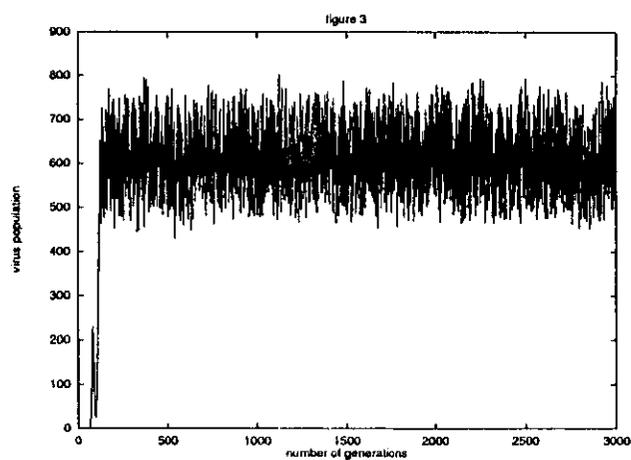


Figura 5.16: Evolución de la población para un experimento.

En la fig. 5.16 es posible ver que el paciente casi elimina al virus completamente en las primeras generaciones: la mutabilidad propia eventualmente aumenta a intervalo excepcionalmente alto, por arriba de 13 (fig 5.17). Las figs 5.18 y 5.19 representan un

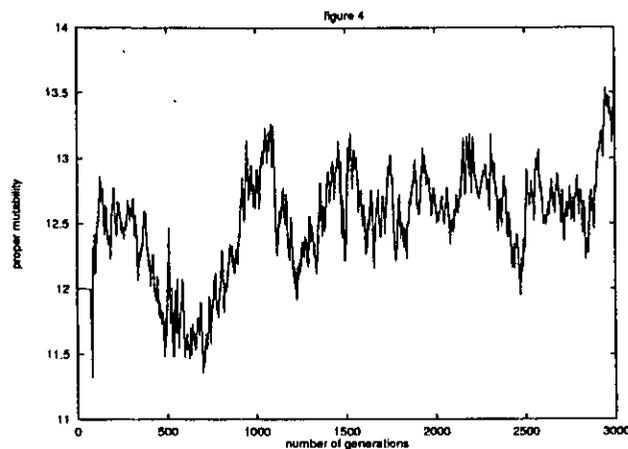


Figura 5.17: Gráfica de mutabilidad propia para el experimento de la figura 5.16 comportamiento más genérico. Es posible notar las fluctuaciones de la infección total durante el proceso donde el número total en la población viral esta entre rangos de 500-750. Las figs. 5.17 y 5.19 muestran que la mutabilidad propia esta por arriba de 12 para casi todas las generaciones, confirmando la hipótesis acerca del rompimiento de simetría.

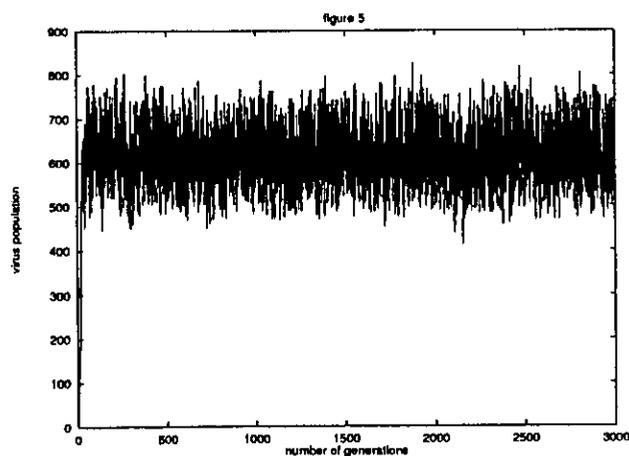


Figura 5.18: Evolución de la población para otro experimento

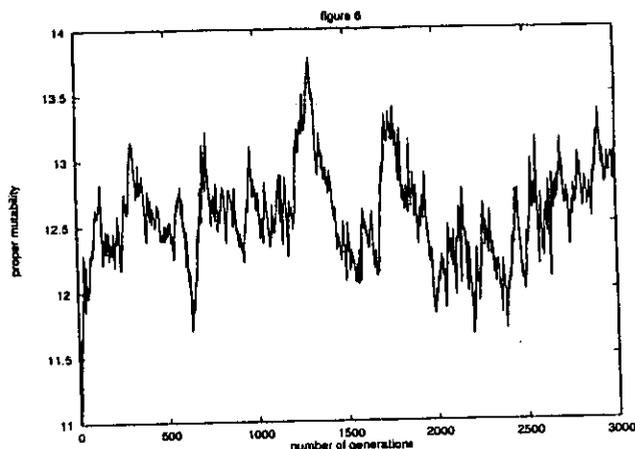


Figura 5.19: Gráfica de mutabilidad propia para el experimento de la fig 5.18.

El modelo del virus, aunque muy sencillo, ilustra perfectamente el fenómeno de rompimiento de simetría inducido. Aunque la elevada mutabilidad no está directamente seleccionada por el paisaje de aptitud es favorecido por una alta aptitud efectiva.

5.4 Modelo de girafas

Este modelo consiste de una población de 1000 genotipos sujeta a mutaciones aleatorias. Un genotipo consiste de un automata celular con elementos binarios el que representa al tamaño del cuello de la girafa, i.e. un fenotipo, dado por el número de automatitas que están “prendidos” en el estado estable de la dinámica del automata. Como hay muchos diferentes automatitas que pueden evolucionar al mismo punto fijo el mapeo genotipo-fenotipo es degenerado. Un gen “maestro” en particular juega un papel especial dado que gobierna la manera en que las reglas booleanas mutan en el esquema evolutivo. Modelos análogos, introducidos en diferentes contextos por Kauffman [99], [81] fueron introducidos para modelar la maquinaria de regulación

genética en la naturaleza, siendo la idea que la tasa de expresión de un gene, en este caso la síntesis de una hormona de crecimiento, esta regulada por un complejo sistema de procesos bioquímicos que tienen propiedades estadísticas similares que aquellas de una red booleana o de autómata celular aleatorio.

Cada miembro de la población es seleccionado para la siguiente generación con probabilidad $P_i = \frac{\text{aptitud}_i}{\sum_j \text{aptitud}_j}$. Al inicio hay un período de abundancia en el cual las girafas encuentran fácilmente comida de árboles grandes y pequeños y el único criterio es que las girafas prefieren una pareja que tenga un tamaño de cuello similar. Para modelar este paisaje de "presión social", se define la aptitud de la i -ésima girafa como una función del tamaño de su cuello n_i y el promedio poblacional del tamaño de los cuellos, $\langle n \rangle$, con valor de 1 si $\langle n \rangle - \delta < n_i < \langle n \rangle + \delta$ y 0 en cualquier otro caso. Aquí, $\delta > 0$ es la ventana de tolerancia. El paisaje de aptitud depende sólo del tamaño del cuello por lo que todos los genotipos que corresponden al mismo punto fijo (fenotipo) tienen la misma aptitud. Por lo tanto no hay una ventaja selectiva directa para un genotipo u otro. Para introducir la dependencia temporal en el paisaje se impone un corto período de "sequía" donde el alimento sólo esta disponible en árboles altos. Este período es modelado asignando una aptitud $f_i = 1$ si $\langle n \rangle - \delta + \epsilon < n_i < \langle n \rangle + \delta + \epsilon$, donde ϵ es un parámetro para controlar la presión, y cero de otra manera. Terminada la "sequía" el paisaje vuelve a su estado original.

El gen "maestro" divide la población en dos categorías genéticas, tipo 0 y tipo 1 las cuales pueden mutar una en otra debido al efecto de mutaciones aleatorias que tienen una probabilidad p excepto para el gen maestro el cual tiene una tasa p' . Los cromosomas tipo 0 por la naturaleza de las reglas de evolución dinámica estan

asociados con la descendencia de cuellos cortos, mientras que los tipo 1, cuando son expresados, tienden a dar descendencia con cuellos más grandes. Antes de la sequía hay un período durante el cual el tipo 1 no se expresa. Después de un cierto período de tiempo se expresa y después la sequía comienza. El paisaje de presión social implica dos posibles atractores: todos tipo 1 o todos tipo 0. El efecto de la sequía es el cambio entre uno y otro.

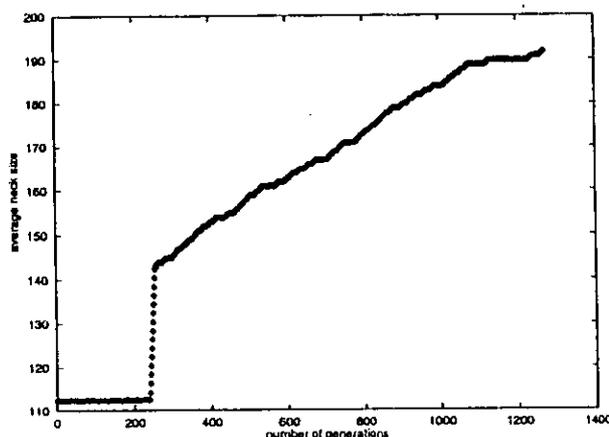


Figura 5.20: Gráfica del tamaño promedio del cuello de la girafa como función de tiempo para una población de 1000 girafas. La sequía inicia en la generación 240 y dura 30 generaciones. Los valores de los parámetros usados fueron: $p=0.0025$, $p'=10^{-6}$, $\delta=2.0$ y $\epsilon=1.0$

Un experimento típico produce los siguientes resultados: en el período inicial de la evolución, antes de la sequía en promedio la longitud de los cuellos es corta. Después del inicio de la sequía el tamaño crece muy rápido. Terminando esta el tamaño sigue creciendo a una menor tasa hasta alcanzar el estado estable. Este comportamiento se ve en la Figura 5.20. Estos resultados se explican de una manera muy sencilla. Durante el período previo a la sequía y antes de la expresión, el tipo de cromosomas 1 se incrementa debido a la tendencia neutral. Después de la expresión ellos son

efectivamente seleccionados otra vez debido a su tendencia para producir girafas con cuellos más largos que sus antepasados que sobrepasan los umbrales de tolerancia y por lo tanto ya no pueden reproducirse. Antes de la sequía la aptitud efectiva de los cromosomas tipo 1 es baja. Sin embargo debido al efecto de las mutaciones los cromosomas tipo 1 no son eliminados totalmente pero forman del 1–5% del total de la población. Inmediatamente después de la sequía la aptitud efectiva de los cromosomas tipo 1 se ve incrementado substancialmente, dado que ellos permiten la aparición de cuellos más largos. El resultado es que la población se ve dominada por cromosomas tipo 1 con una pequeña proporción de cromosomas tipo 0 que permanecen por efectos de mutación. Después de la sequía como los cromosomas tipo 1 tienden a producir cuellos más largos hasta que se alcanza el estado estable y ya no puede crecer más. En la Figura 5.21 el número de cromosomas tipo 1 se muestra como función de tiempo.

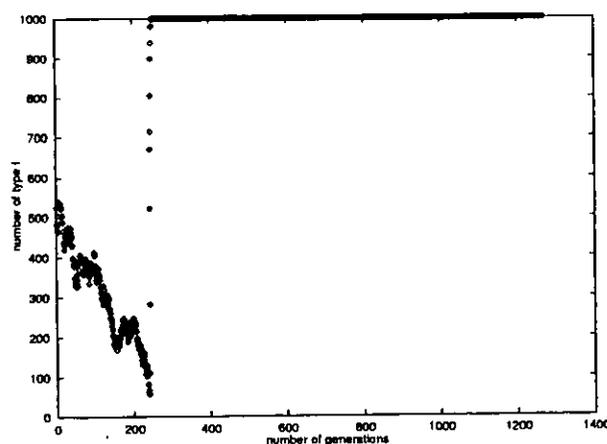


Figura 5.21: Número de girafas tipo 1 como función del tiempo. Mismos parámetros que en Figura 5.20.

Tanto el modelo del virus como el de las girafas exhiben el mismo fenómeno: el rompimiento de simetría inducido aumenta la presión selectiva efectiva para ciertos

sinónimos. En el modelo de girafas no hay diferencia selectiva directa entre cromosomas de tipo 0 y 1. La única ventaja de uno respecto al otro esta en la habilidad para producir poblaciones bien adaptadas.

5.5 Implementación para Agente viajero

Se eligió el problema del agente viajero como un paisaje experimental de los algoritmos genéticos autocodificados como un ejemplo de un sistema donde el óptimo global no se conoce a priori y donde el tamaño del espacio de estados es muy grande considerando 33 ciudades.

El problema del agente viajero es el principal exponente de los problemas que tratan con la explosión combinatoria. Este problema consiste en determinar la ruta más corta para visitar un conjunto de ciudades una sólo vez.

El problema del agente viajero es un problema típico NP-Completo que es fácil de explicar pero difícil de resolver. En este trabajo se comparó el desempeño de un algoritmo genético con parámetros fijos contra un algoritmo genético autocodificado. El objetivo fue mostrar la propiedad de optimizador global del algoritmo genético, pero sobre todo demostrar su utilidad ante problemas dinámicos, donde la dinámica fue asociada con el aumento en el número de ciudades a visitar después de un cierto tiempo. Es bien sabido que los algoritmos genéticos son bastante competitivos con otras técnicas de optimización [100] cuando la codificación es la adecuada.

Considerando lo anterior se eligió la codificación más sencilla donde las ciudades se enlistan en el orden de visita. Los operadores genéticos usados fueron: "mutación" (permutación de dos ciudades al azar) y "entrecruzamiento" (inversión de las ciudades entre dos puntos elegidos aleatoriamente dentro del cromosoma). Por ejemplo: para

un problema de 6 ciudades la mutación en 0 y 3 de la posible ruta 134520 lleva a 534120. La inversión entre los puntos 1 y 5 en la misma ruta produce la nueva ruta 132540. No se evitaron las permutaciones cíclicas para una ruta determinada ya que la idea principal fue compara los desempeños poblacionales de los algoritmos genéticos. Para comparar con el algoritmo genético de parámetros fijos se hicieron varios experimentos variando las tasas de “mutación” y “cruza” entre 0 y 1 en incrementos de 0.1 probando 100 puntos dentro del espacio de parámetros. Las codificaciones de los operadores se hicieron usando 8 bits para cada uno de los operadores resultando un cromosoma de 49 bits ($33 + 8 + 8$). Se estableció un valor mínimo de mutación de 0.001. Se compararon los desempeños on-line y off-line para el caso del agente viajero para ambos experimentos (fijos y autocodificados).

La función de aptitud para el problema del agente viajero de n -ciudades es la distancia total de la ruta

$$R = \sum_{i=0}^{n-1} d(c_i, c_{i+1}) + d(c_0, c_{n-1}) \quad (5.5.1)$$

donde $d(x, y)$ es la distancia entre las ciudades x y y , con la restricción de que todas las ciudades deben ser visitadas una sola vez.

Este problema también fue estudiado considerando su dependencia temporal donde se empezó con 23 ciudades a optimizar y en la generación 3000 se agregaron 10 ciudades más teniendo que encontrar una nueva ruta diferente de la inicial. Para comparar los desempeños de los algoritmos genéticos en el problema del agente viajero se usó la función:

$$F = \sum_{i=j}^{j+100} \frac{R_{min}(i) + R_{av}(i)}{R_{min}(i) - R_{av}(i)} \quad (5.5.2)$$

donde la suma es sobre un intervalo de 100 generaciones y j representa el período de tiempo para el cual se evalúa F . En un período de 6000 generaciones se tuvieron 60 puntos para evaluar F . $R_{min}(i)$ es la ruta más corta encontrada en la generación i y $R_{av}(i)$ es el promedio de las rutas en la generación i . Esta función enfatiza la noción de la aptitud poblacional que la del mejor individuo. En términos de pura optimización combinatoria sabiendo que existe un óptimo único el objetivo sería encontrar al mejor individuo en el menor tiempo posible. Sin embargo en muchos otros problemas, como es el caso de la evolución y principalmente en problema de dependencia temporal es más importante la aptitud del conjunto poblacional. La ecuación anterior toma en cuenta tanto la velocidad de convergencia como la aptitud poblacional.

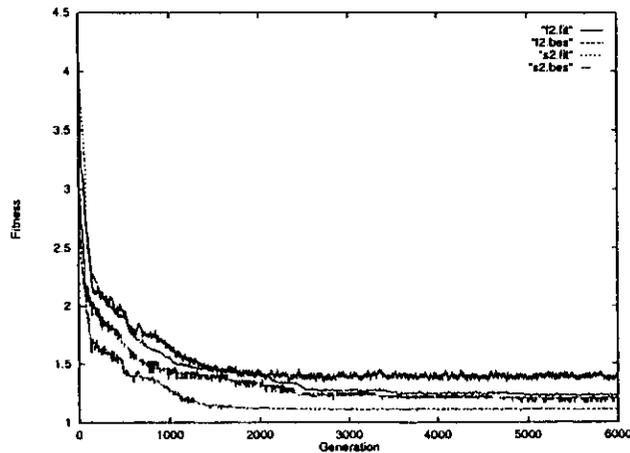


Figura 5.22: Gráfica de aptitud (longitud de la ruta) en función del tiempo para el problema del agente viajero de 33 ciudades, f2.fit y f2.best representan el promedio de aptitud y el mejor promedio para un algoritmo genético con tasas fijas mientras que s2.fit y s2.best para un algoritmo autocodificado con 8 bits.

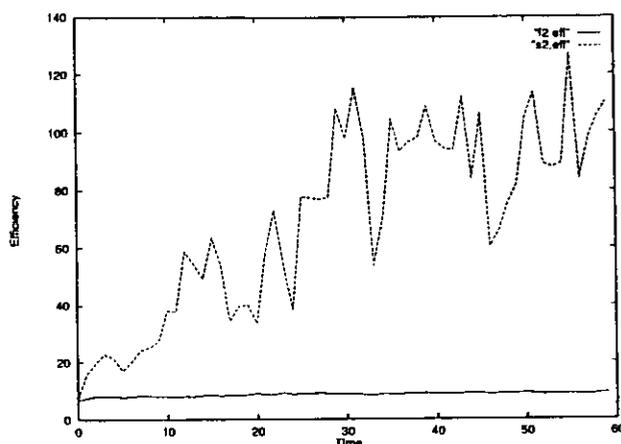


Figura 5.23: Gráfica de eficiencia vs tiempo para algoritmos genéticos con parámetros fijos y con parámetros autocodificados para el problema del agente viajero de 33 ciudades. f2.eff representa la eficiencia para el algoritmo con tasas fijas y s2.eff para el algoritmo autocodificado.

Se probaron varias tasas fijas encontrando que la relación óptima entre desempeño online y offline se presentaba con $p=0.1$ y $p_c=0.2$. Para valores más altos los indicadores de desempeño tuvieron una drástica caída. Es con los parámetros óptimos contra los que se comparó el algoritmo genético autocodificado, en la Figura 5.22 se puede apreciar dicha comparación. En términos de velocidad de convergencia los dos son comparables pero en términos de la función de aptitud del algoritmo genético codificado funciona mucho mejor debido a que la aptitud promedio es casi tan buena como la mejor encontrada en la generación. La convergencia entre desempeño offline y online es una característica importante de los algoritmos genéticos autocodificados implicando que estos no sacrifican desempeño online para tener un buen desempeño offline. Tales desempeños se muestran en la Figura 5.23. La ruta más corta para este problema reportada en [101] es 10930. El mejor resultado de los algoritmos genéticos

fue de 11040 para el que usó tazas fijas y 11662 para el que usó parámetros codificados. También se descubrió que el operador de inversión tiene un mayor efecto para alterar el cromosoma que el operador de mutación.

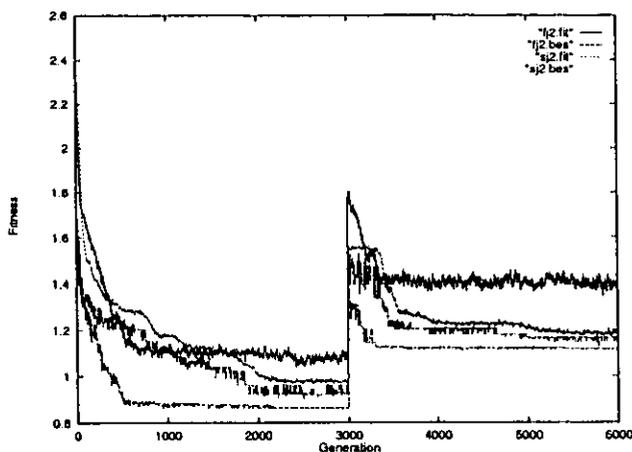


Figura 5.24: Gráfica de aptitud vs tiempo para el problema del agente viajero para el “jumper landscape” donde después de 300 generaciones se añaden 10 ciudades más al conjunto inicial de 23. $f2.fit$ y $f2.bes$ representa la aptitud promedio y la mejor aptitud para un algoritmo con tasa fijas, $s2.fit$ y $s2.bes$ representan los mismos valores pero para el algoritmo autocodificado.

En la Figura 5.24 se muestran los resultados para el “jumper landscape” para el problema del agente viajero y en la Figura 5.25 se comparan las eficiencias de ambos tipos de algoritmos genéticos (tomando muestras cada 50 generaciones). Para la Figura 5.24 se puede apreciar que tan pronto se presenta el cambio en $t=3000$ la tasa de cruza empieza a subir hasta alcanzar su valor máximo en $t=3500$ donde ha casi duplicado su valor. Con esto se demuestra que el sistema responde al cambio en el paisaje de aptitud modificando el valor de sus parámetros, de manera explícita esto se ve en la Figura 5.26.

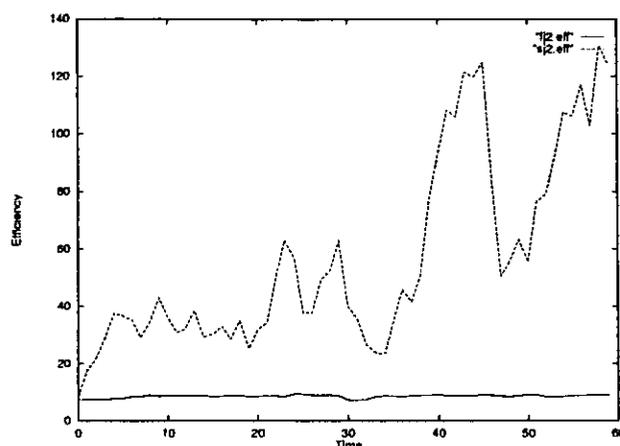


Figura 5.25: Gráfica de eficiencia vs tiempo para algoritmos genéticos con tasas fijas (fj2.eff) y con tasas autocodificadas (sj2.eff) para el “jumper landscape”.

5.6 Manejo de portafolios de inversión

Se le denomina portafolio de inversión a la combinación de diferentes instrumentos bursátiles en determinadas proporciones, por ejemplo se puede conformar un portafolio con 20% del dinero en acciones de TELMEX L, 25% en acciones de TELECOM A1, 35% en GFBB O y el resto en efectivo, otra opción podría ser 50% en CETES, 25% en dólares y el resto a tasa fija en un banco. La idea de manejar un portafolio de inversión es el maximizar las ganancias diversificando el riesgo. En esta tesis se presentan resultados para un portafolio de inversión formado por emisoras que cotizan en la Bolsa Mexicana de Valores.

Las bolsas de valores son instituciones que surgen para facilitar el contacto entre entidades superavitarias y deficitarias de recursos monetarios. La Bolsa Mexicana de Valores (BMV) ha fomentado el desarrollo de México ya que ha contribuido a canalizar el ahorro hacia la inversión productiva, fuente de empleo y crecimiento del país.

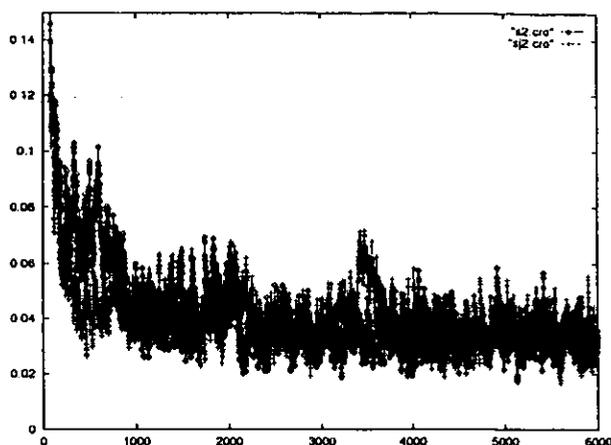


Figura 5.26: Gráfica de la tasa de entrecruzamiento vs tiempo para un paisaje plano (s2.cro) de 33 ciudades y para el “jumper landscape” (sj2.cro)

Algunos beneficios de la existencia de la BMV:

1. Fortalecimiento del ahorro interno y la inversión
2. Disminución por endeudamiento externo
3. Equilibrio para las empresas entre pasivos y capital
4. Diversificación de la propiedad industrial en el país.

La inversión en el mercado bursátil es una opción desaprovechada por el público en general, pero que puede ayudar a incrementar o mantener el patrimonio de familias, algunas de las razones por las que no se usa pueden ser la ignorancia de su funcionamiento, la disponibilidad de dinero necesario para invertir, su alto riesgo comparado con el de otros instrumentos, entre otros.

El *mercado de valores* es el lugar donde los corredores (representantes de casa de bolsa) realizan acciones de compra-venta a nombre de terceros (público en general),

las posiciones de compradores o vendedores son intercambiables; la mercancía con la que se negocia son los recursos monetarios. El comprador acude a este mercado con sus excedentes de recursos monetarios, busca al representante de un vendedor (el cual tiene un déficit de recursos monetarios) ya que desea hacer inversiones en proyectos que prometen generar riqueza suficiente tanto para el comprador como para el vendedor. Este cambio de recursos por derechos sólo sucede eventualmente y es conocido como oferta primaria o *mercado primario*. Al intercambio de estos derechos entre el público permite la existencia de un mercado permanente conocido como *mercado secundario*.

Las acciones constituyen un conjunto de títulos que se encargan de representar parte del capital correspondiente a determinada empresa. La tenencia de acciones otorga a sus compradores los derechos de participar de los beneficios de la empresa ya que cada comprador es un socio más.

Se denomina *lote* a la cantidad mínima de títulos que convencionalmente se intercambian en una transacción. Para el mercado accionario un lote se compone de mil títulos.

Se denomina *puja* al importe mínimo en que puede variar el precio unitario de cada título, y se expresa como una fracción del precio de mercado o valor nominal de dicho título. Para que sea válida una postura que pretenda cambiar el precio vigente de cualquier título, lo debe hacer cuando menos por el monto de una puja.

Se denomina *pico* a la cantidad de títulos menor a la establecida para un lote. Las transacciones con picos deben efectuarse al último precio y están sujetas a reglas particulares de operación.

5.6.1 Sistema para manejo de portafolios.

Las características combinatoriales de un portafolio de inversión (configuración en las cantidades invertidas en diferentes instrumentos bursátiles), y la posibilidad de establecer reglas que llevan al mismo resultado (existencia de un mapeo genotipo-fenotipo no trivial, i.e. muchos a uno) motivan el empleo de un algoritmo genético inteligente, cuyo objetivo es maximizar la ganancia diversificando el riesgo, al invertir en diferentes instrumentos, sus entradas son, para este trabajo, las enviadas por el sistema de predicción de mercado "Adaptrader Advantage 2.0" desarrollado por la empresa "Adaptive Technologies S.A. de C.V."

El sistema administrador de portafolios de inversión emplea un algoritmo genético, en donde el resultado no se lee directamente del vector resultante sino a través de un intérprete que "traduce" el conjunto de reglas que componen al vector genético. El conjunto de reglas que usa el algoritmo genético esta distribuido en tres etapas: decisión, cantidad y manejo de riesgo. El resultado final es la lista de emisoras operadas, junto con la posición recomendada así como la cantidad de acciones a negociar. Como se puede apreciar este es un tipo de sistema clasificador en el sentido de usar un conjunto de reglas que junto con las recomendaciones del Adaptrader Advantage 2.0 conforma el portafolio. Es importante mencionar que es posible emplear recomendaciones de cualquier otro método como por ejemplo al usar herramientas de análisis técnico. Para este trabajo se uso el Adaptrader Advantage 2.0 dados sus buenos resultados obtenidos desde que salió al mercado.

El sistema de predicción Adaptrader Advantage 2.0 usa un equipo de agentes artificiales entrenados y dedicados específicamente para producir recomendaciones para una acción en particular por un período limitado. Dichos agentes artificiales son

creados por un sistema maestro dedicado a buscar burbujas de predictibilidad para ciertas emisoras, una vez que encuentra una, crea uno de estos agentes artificiales que se dedica a explotarla para sacar ganancias, además el sistema supervisa los agentes y declara un agente “muerto” cuando la burbuja de predictibilidad se desvanece.

5.6.2 Desarrollo del sistema

El genotipo del algoritmo es un conjunto de reglas clasificadoras que resumen la información proporcionada por las salidas del Adaptrader Advantage. Están asociadas con las cuatro etapas del proceso. Además del genotipo se tiene al “intérprete” genético el cual determina la aptitud asociada con cada individuo en la población y obtiene el fenotipo resultante.

El sistema administrador de portafolios de inversión consta de tres etapas (decisión, cantidad y manejo de riesgo) cada una de ellas, a su vez, constituida por otro conjunto de procedimientos y operadores. Todas estas etapas se integran al algoritmo genético como un conjunto de cromosomas.

Parte 1. Decisión

A continuación se presentan las señales proporcionadas por el Adaptrader Advantage, sus rangos y explicación.

Regla	Salida AA	Rangos
A	Predictibilidad	[0,10]
B	pfi	+,n,-
C	Fuerza1	[-1,1]
D	Fuerza2	[-10,10]
E	P.objetivo	[0,100]%
F	Riesgo	[0,100]%
G	min50	[0,100]%

- **Predictibilidad:** es una función sobre el canal de negociación de un agente que define su desempeño. Varía entre 0 y 10. Un buen valor es arriba de 5.
- **Pfi:** Price Feedback Indicator. Mide como un agente o grupo de agentes reaccionarán ante cambios en el precio. Si la recomendación es COMPRA y el indicador es “+” significa que si el precio sube la recomendación seguirá siendo COMPRA. Si el precio baja el agente tratará de vender. De la misma manera, si la recomendación es VENTA y el Pfi es “+” si el precio baja la recomendación seguirá siendo VENTA mientras que si sube el agente cambiará a comprar. Por otro lado si el pfi es “-” indica que el agente buscará una oportunidad para tomar ganancias: si la recomendación es COMPRA y el precio sube, dependiendo del tamaño del cambio puede hacer que el agente cambie de opinión, si la recomendación es VENTA el agente puede cambiar a COMPRA. Si el pfi es “n” (neutral) significa que no es posible anticipar la reacción del agente ante un cambio en el precio.
- **Fuerza1:** Basado en una fuerza de consenso (usando una fórmula de reducción de volatilidad) de las fuerzas actuales de los agentes de Adaptrader Advantage. Si es mayor a 0.5 la recomendación es COMPRA, si es menor a -0.5 la

recomendación es VENTA, en cualquier otro caso es NEUTRAL.

- Fuerza2: Basado en una fuerza de consenso de las fuerzas actuales de Adaptrader Advantage, donde la fuerza de cada agente esta pesada por su predictibilidad. Su interpretación en términos de recomendaciones es como la Fuerza 1.
- P. objetivo: Variación entre precio objetivo de corto y largo plazo
- Riesgo: Variación entre riesgo de corto y largo plazo
- Min50: Variación entre el precio actual y el precio mínimo en los ultimos 50 dias. $x = \frac{\text{precio}_{\text{actual}} - \text{min}_{50}}{\text{max}_{50} - \text{min}_{50}}$

A continuación se presentan la discretización de las señales proporcionadas por el Adaptrader Advantage:

Regla	Valor	Salida
A	< 3	0
	3-4	1
	4-5	2
	≥ 5	3
B	+	0
	n	1
	-	2
C	< 0.5	0
	0.5-0.8	1
	0.8-0.999	2
	1	3
D	< 0.5	0
	0.5-1.0	1
	≥ 1.0	2
E	ganancia < 10%	0
	10-20 %	1
	20-35 %	2
	$\geq 35\%$	3
F	pérdida < 10%	0
	10-20 %	1
	20-35 %	2
	$\geq 35\%$	3
G	< 0.3	0
	0.3-0.7	1
	>0.7	2

El algoritmo de decisión esta basado en una cadena de filtros, alimentada por una fuente. El valor de la fuente es 0,1,2 dependiendo de la recomendación del Adaptrader Advantage; cada filtro recibe una entrada que es igual a 0,1 o 2 cuya salida dependen

de la discretización de los resultados de Adaptrader Advantage y de la salida del filtro precedente. La salida final es 1 (aceptar la decisión) o 0 (rechaza la decisión).

El proceso para la decisión es de la siguiente forma:

$$C+D \rightarrow A \rightarrow E+F \rightarrow B+G \rightarrow 0/1$$

Las flechas representan el flujo de un elemento al filtro siguiente.

Esta agrupación indica que se construyen tablas de verdad para los conjuntos de salidas C+D, A, E+F y B+G. Los resultados de estas tablas son generados aleatoriamente y constituyen los genotipos que empleará el algoritmo genético. Todas las tablas excepto C+D tienen como salida un operador el cual modifica la señal de entrada. Los operadores son α , β , γ , δ definidos como:

Operador	Entrada	Salida
α	2	2
	1	1
β	2	1
	1	1
γ	2	1
	1	0
δ	2	0
	1	0

Es importante mencionar que un valor de 0 en cualquier punto de la traducción rechaza la decisión.

En seguida se presenta una posible combinación de tablas producidas por un elemento del algoritmo genético para cada una de las etapas, un ejemplo más detallado se presenta en la siguiente sección.

C	D	C+D	A	A_{op}	E	F	$(E + F)_{op}$	B	G	$(B + G)_{op}$
0	0	0	0	δ	0	0	γ	0	0	β
0	1	0	1	γ	0	1	δ	0	1	β
0	2	0	2	β	0	2	δ	0	2	β
1	0	0	3	α	1	0	α	1	0	β
1	1	1			1	1	β	1	1	γ
1	2	1			1	2	γ	1	2	δ
2	0	1			1	3	δ	2	0	δ
2	1	1			2	0	α	2	1	δ
2	2	2			2	1	α	2	2	δ
3	0	1			2	2	β	2	0	δ
3	1	2			2	3	β	2	1	δ
3	2	2			3	0	α	2	2	δ
					3	1	α			
					3	2	α			
					3	3	α			

Las salidas intermedias de cada una de las etapas son un indicador del resultado final, este indicador tiene valores 0,1 y 2.

La traducción de las señales del Adaptrader Advantage dependen de la composición y dimensión de las tablas de decisión y de la discretización de las señales de entrada. La tabla C+D tiene como componentes a C y D con dimensiones C=4 y D=3, la tabla E+F tiene componentes E y F con dimensiones E=4 y F=4, la tabla B+G tiene a B y G como componentes cuyas dimensiones son 3 y 3 respectivamente, finalmente la tabla A tiene dimensión 4. Para determinar que elemento del cromosoma le corresponde a una determinada discretización de las salidas del Adaptrader Advantage se emplea la ecuación:

$$ut = d(A_1) \cdot D(A_0) + D(A_1) \quad (5.6.1)$$

donde ut identifica al elemento dentro del cromosoma que representa la salida de la tabla de decisión. $d(A_1)$ es la dimensión del componente A_1 . $D(A_0)$ es la discretización del componente A_0 . $D(A_1)$ es la discretización del componente A_1 .

Note que esto no es necesario para la tabla A ya el valor se lee directamente de la discretización, i.e. si este es 0 corresponde al primer renglón, si es 1 corresponde al segundo y así sucesivamente.

Para ilustrar el empleo de estas tablas se presentan los siguientes ejemplos: considere los resultados de la tabla anterior que corresponde a $C+D$, suponga que la discretización de C y D fue 3 y 2 respectivamente por lo tanto el valor de salida en la tabla será el correspondiente a $3 \cdot 3 + 2 = 11$, que en la tabla tiene un valor de 2, 11 significa el onceavo renglón de la tabla considerando que el primer renglón corresponde a cero. Otro ejemplo: sea la discretización de 2 y 1 la discretización para C y D respectivamente el renglón de interés será el séptimo ($3 \cdot 2 + 1 = 7$) cuya salida es 1. Un último ejemplo: sea los valores 1 y 2 la discretización de E y F , el renglón de interés es el sexto ($4 \cdot 1 + 2 = 6$) cuya salida es γ . Es importante mencionar que la referencia al número de renglón corresponde al mismo bit al considerar la codificación como cromosoma lo cual se verá más adelante.

Teniendo definida la decisión a tomar, se procede a la etapa 2 (cantidad)

Parte 2. Cantidad

El proceso para la determinación de la cantidad es:

$$I+DD \rightarrow DDE \rightarrow B \rightarrow \text{Cantidad a negociar}$$

donde I corresponde representa la diferencia entre la cantidad máxima de acciones que se puede comprar en esta emisora y el inventario actual. Por ejemplo si el dato es igual al 10% significa que falta un 10% del inventario actual para alcanzar la máxima cantidad de acciones permitidas para una determinada emisora. DD representa el dinero disponible (liquidez) para comprar acciones y es también un porcentaje en función del capital total. La discretización es de la siguiente manera:

Regla	Valor	Salida
I	< 20 %	0
	20-40 %	1
	40-60%	2
	60-80%	3
	\geq 80%	4

Estas configuraciones son para una señal de compra, para una señal de venta se tiene:

Regla	Valor	Salida
I	< 20 %	4
	20-40 %	3
	40-60%	2
	60-80%	1
	\geq 80%	0

Las opciones a tomar son:

- 0: compra/venta hasta el máximo posible
- 1: C/V la mitad del porcentaje o el máximo
- 2: C/V 1/3 o el máximo
- 3: C/V 1/4 o el máximo
- 4: nada

Parte 3. Manejo de riesgo

Las etapas para la determinación de límites se componen de:

1. Capitalización de mercado de la emisora
2. Coeficiente Beta estimado para un conjunto de 20 días.

Capitalización de mercado La capitalización de mercado se refiere al número de acciones que de una determinada emisora multiplicadas por su precio, que están en el mercado para su negociación. Proporciona el orden en que se harán las negociaciones empezando por la emisora con mayor participación.

Coeficiente Beta estimado El observar en el mercado de valores el comportamiento de las acciones (cuando el mercado sube muchas de las acciones tienden a incrementar su precio y cuando el mercado baja muchas tienden a disminuirlo) sugiere una respuesta común de las acciones a cambios en el mercado. Por lo que se puede relacionar la ganancia de una emisora con la ganancia del mercado mediante:

$$R_i = \alpha_i + \beta_i R_m + e_i \quad (5.6.2)$$

donde:

R_i =Ganancia de la acción i

R_m =Ganancia del mercado

α_i =Valor esperado del componente de la ganancia independiente de la ganancia del mercado

e_i =Representa el elemento aleatorio de la ganancia de la emisora independiente de la ganancia del mercado

β_i = Constante que mide el cambio esperado en R_i dado un cambio en R_m . Es decir una medida de la sensibilidad de la emisora a movimientos del mercado.

Es importante hacer notar que tanto R_m como e_i son variables aleatorias con su correspondiente distribución de probabilidad y desviación estandar (σ_{ei} y σ_m respectivamente).

Es conveniente que e_i (distribución normal con media 0) sea no correlacionada con R_m , es decir:

$$\text{cov}(e_i, R_m) = E[(e_i - 0)(R_m - \overline{R_m})] = 0 \quad (5.6.3)$$

Esta ecuación indica que la ganancia de la emisora es independiente de las otras ganancias excepto la del mercado.

Es necesario calcular la ganancia sobre la inversión durante un período determinado de tiempo conocido como período de posesión. Es medida por la suma del cambio en el precio de mercado de una acción más cualquier ingreso recibido sobre el período de pertenencia, dividido por el precio de la acción al inicio de este.

Otro supuesto importante es que $E(e_i e_j)$ es igual a cero para toda i y j , lo que implica que la única razón por la que las emisoras cambian en forma conjunta es debido a un movimiento común con el mercado.

Por definición:

Varianza de la parte aleatoria, e_i , del componente independiente en la ganancia de la emisora i , $\sigma_{ei}^2 = E(e_i)^2$

Varianza del índice del mercado, R_m , $\sigma_m^2 = E(R_m - \overline{R_m})^2$

Ganancia promedio de la emisora i , $\overline{R_i} = \alpha_i + \beta \overline{R_m}$

Varianza de la ganancia para la emisora i , $\sigma_i^2 = \beta^2 + \sigma_{ei}^2$

Covarianza de las ganancias entre emisoras i y j , $\sigma_{ij} = \beta_i \beta_j \sigma_m^2$

Proporción del capital invertido en la emisora i , X_i .

La ganancia esperada del portafolio es:

$$\overline{R_P} = \sum_{i=1}^N X_i \alpha_i + \sum_{i=1}^N X_i \beta_i \overline{R_m} \quad (5.6.4)$$

y su varianza:

$$\sigma_P^2 = \sum_{i=1}^N X_i^2 \beta_i^2 \sigma_m^2 + \sum_{i=1}^N \sum_{i=1, j \neq i}^N X_i X_j \beta_i \beta_j \sigma_m^2 + \sum_{i=1}^N X_i^2 \sigma_{ei}^2 \quad (5.6.5)$$

Características del modelo de un índice:

Sea la Beta del portafolio el promedio ponderado de cada una de las Betas individuales en función de la porción del portafolio invertido en cada emisora, si N es el número de emisoras en el portafolio:

$$\beta_P = \sum_{i=1}^N X_i \beta_i \quad (5.6.6)$$

$$\alpha_P = \sum_{i=1}^N X_i \alpha_i \quad (5.6.7)$$

$$\overline{R_P} = \alpha_P + \beta_P \overline{R_m} \quad (5.6.8)$$

Si el portafolio P tiene la misma configuración que el portafolio de mercado (todas las acciones en la misma proporción que tienen en el índice) se debe cumplir que $\overline{R_P} = \overline{R_m}$ lo cual sucede si $\alpha_P = 0$ y $\beta_P = 1$

El riesgo del portafolio del inversionista es:

$$\sigma_P^2 = \beta_P^2 \sigma_m^2 + \sum_{i=1}^N X_i^2 \sigma_{ei}^2 \quad (5.6.9)$$

Asumiendo que el inversionista invierte la misma cantidad de dinero en cada una de las N emisoras ($X_i = 1/N$) se tiene:

$$\sigma_P^2 = \beta_P^2 \sigma_m^2 + \frac{1}{N} \left(\sum_{i=1}^N \frac{1}{N} \sigma_{ei}^2 \right) \quad (5.6.10)$$

Se observa que al aumentar el número de emisoras el riesgo residual promedio disminuye. Entonces

$$\sigma_P = \beta_P \sigma_m = \sigma_m \left[\sum_{i=1}^N X_i \beta_i \right] \quad (5.6.11)$$

A la cantidad σ_{ei}^2 se le conoce como riesgo diversificable, β_i es el riesgo no diversificable de la emisora i , y ya que el riesgo diversificable puede ser disminuido al aumentar el número de emisoras en el portafolio β_i es usado como la medida de riesgo de la emisora.

La estimación de β_i se hace por regresión lineal de la ecuación 5.6.2, donde la distribución de datos en la grafica R_{it} vs R_{mt} depende del valor de σ_{ei}^2 y es necesario asumir que α_i , β_i y σ_{ei}^2 asumiendo que son constantes en el tiempo.

También se puede estimar a partir de:

$$\beta_i = \frac{\sigma_{im}}{\sigma_m^2} \quad (5.6.12)$$

$$\alpha_i = \overline{R_{it}} - \beta_i \overline{R_{mt}} \quad (5.6.13)$$

Considerando lo anterior tenemos

$\frac{\beta_{nuevo} - \beta_{ultimo}}{\beta_{ultimo}}$	Opción
$\gg 0.25$	0
0.25-0.5:1	1
0.5-0.75	2
>0.75	3

Las opciones a tomar son:

- 0: compra/venta hasta el máximo posible
- 1: C/V la mitad del porcentaje o el máximo
- 2: C/V 1/3 o el máximo posible
- 3: C/V 1/4 o el máximo posible

Por máximo posible se entiende la cantidad máxima a negociar en función de su capitalización de mercado. El producto resultante de la capitalización de mercado por el porcentaje del portafolio invertido en la emisora determina los límites máximo y mínimo entre los que puede estar el portafolio después de la transacción. En la sección correspondiente al algoritmo genético se presenta un ejemplo.

Codificación y operadores La codificación del algoritmo es de la siguiente manera: cada individuo de la población está formado por un conjunto de una matriz de datos de $M \times N$ donde M es el número de emisoras a considerar y N es el número total de posibilidades de las tablas de decisión mencionadas anteriormente, las cuales dependen de la discretización empleada. En este caso para la tabla C+D se tiene que

N es 12, para A N es 4, y para B+G N es 9. Además hay dos vectores correspondientes al manejo de la cantidad y riesgo. La dimensión de dichos vectores es de 5 y 4 respectivamente. Las secciones del cromosoma para decisión son para cada emisora mientras que las referentes a cantidad y riesgo son para el portafolio en su conjunto.

Se emplean los operadores normales de mutación (cambio en uno de los bits por otro perteneciente al mismo alfabeto) y entrecruzamiento en una posición para cada etapa (elección de dos padres, se elige un punto dentro de cada uno de los elementos del cromosoma (decisión, cantidad y riesgo) y se intercambian los elementos a la derecha con los del otro padre).

Implementación y resultados El caso práctico que se consideró fue para las emisoras TELMEX L y BANACCI O dentro del período del 7 de mayo de 1999 al 14 de julio de 2000. Se hicieron experimentos con diversos valores en las tasas de mutación y cruza y aquellas que dieron los mejores resultados fueron de 0.001 para mutación y 0.8 para cruza.

Algoritmo genético Se generó una población inicial de 50 individuos. Un ejemplo de individuo dentro de la población fue:

B+G	C+D	E+F	A	Cantidad	Riesgo
231112232	112221122122	3212212101300122	2320	33334	1121
223323121	211212212112	1212202032012030	1231		

Las primeras cuatro columnas son los elementos de la parte de decisión. Estas columnas corresponden a las columnas de salida de las tablas anteriores (C+D, A, B+G, E+F). Por facilidad en la implementación computacional los operadores α , β , γ y δ se codifican como 0, 1, 2 y 3 respectivamente dentro del cromosoma. El primer renglón corresponde a TELMEX L y el segundo a BANACCI O.

La interpretación de dicho individuo sería de la siguiente manera: sea la discretización de los valores del Adaptrader Advantage 2.0 para TELMEX L del día 3 de julio de 2000 :

Entrada	Valor	Discretización
A	5.5	3
B	n	1
C	1	3
D	1.97	2
E	0.047	0
F	-0.17	1
G	$\frac{30.7-21.55}{21.55} = 0.42$	1

Primero se traduce la porción C+D, cuyos valores son 3 y 2 respectivamente que de acuerdo a la ecuación 5.6.1 corresponden al elemento 12 de la sección C+D del cromosoma, dicho elemento tiene un valor de 2. La discretización de A es 3 indicando que el operador que se usará es α (tercer elemento de la sección A del cromosoma), el operador para E+F, considerando las discretizaciones de E y F (0 y 1) corresponde al valor del bit 1, usando 5.6.1, en la sección E+F del cromosoma, dicho valor es 2 (operador γ). Finalmente para B+G se tiene que es el elemento 5 de la sección correspondiente en el cromosoma, cuyo valor es 2 (operador γ), también usando 5.6.1 con $A_0=B$ y $A_1=G$. Este ejemplo fue para la primera línea del cromosoma (TELMEX L). Es muy importante mencionar que la numeración de los elementos del cromosoma empieza en 0.

El resultado se obtiene :

$$C+D=2 \rightarrow \alpha(2) = 2 \rightarrow \gamma(2) = 1 \rightarrow \gamma(1) = 1 \rightarrow 1$$

aceptándose la decisión del sistema. Esto se hace para cada una de las emisoras del portafolio. Una vez determinadas las decisiones a tomar se calcula la cantidad que se negociará para cada una de las emisoras.

Para la determinación de la cantidad se procede de la siguiente forma: la cantidad máxima a negociar será la proporción del portafolio para esa emisora en función de su participación de mercado (ejemplo: si se tiene el 33% del portafolio invertido en una determinada emisora, cuya participación de mercado es del 27% la cantidad en el portafolio después de la transacción estará entre los límites definidos por 33 ± 8.91 ($8.91 = 33 * 0.27$), i. e. si se va a comprar la cantidad máxima en el portafolio es de $33 + 8.91 = 41.91$ y si se va a vender la cantidad mínima en el portafolio será $33 - 8.91 = 24.09$. Una vez establecidos los límites se determina la cantidad en función de la distribución de portafolio discretizado y leyendo el valor correspondiente en la porción del cromosoma correspondiente a la cantidad. El manejo de riesgo se hace considerando 20 días anteriores al día en cuestión, calculando sus ganancias en 5 días y calculando los coeficientes beta correspondientes con los coeficientes beta calculados se extrapola pronosticar el siguiente coeficiente. Discretizando la variación de este respecto al ultimo calculado se encuentra la acción a tomar.

Para el funcionamiento de los operadores de mutación y cruza: la mutación cambia en valor para el bit dentro de la sección correspondiente por otro dentro del mismo alfabeto.

Cálculo de aptitud:

El conjunto de reglas que representa cada individuo es aplicado durante los 20 días anteriores a la fecha actual. Para cada día se discretizan sus valores respectivos y se determina la posición y la cantidad a negociar. Se empieza con un portafolio inicial distribuido en forma igualitaria entre las emisoras y el efectivo disponible (para este caso particular la distribución es 1/3, 1/3 y 1/3) que evoluciona de acuerdo a las reglas codificadas de cada individuo. Se calcula la ganancia neta $\frac{\text{valor de portafolio final} - \text{valor de portafolio inicial}}{\text{valor de portafolio inicial}}$. Esta cantidad es la aptitud de cada individuo. Este valor es el que se emplea para el funcionamiento del algoritmo genético.

El proceso de reproducción, mutación y cruce se repite durante 25 generaciones.

Resultados El análisis de resultados se hizo para los periodos del 17 de marzo al 14 de abril, del 17 de abril al 18 de mayo y del 19 de mayo al 15 de junio, todos del año 2000.

Los mejores individuos para cada una de las etapas fueron:

Para la etapa entre el 17 de marzo y el 14 de abril:

B+G	C+D	E+F	A	Cantidad	Riesgo
112213233	111222111102	2000003301331020	2112	20131	2121
223223121	221221212122	3311212222113211	1101		

Para la etapa entre el 17 de abril y el 18 de mayo:

B+G	C+D	E+F	A	Cantidad	Riesgo
121121212	111111221112	0322112323131001	3020	24240	0121
232223121	211112112222	3121311113130113	0312		

Para la etapa entre el 19 de mayo y el 15 de junio:

B+G	C+D	E+F	A	Cantidad	Riesgo
122331112	221221111221	2020121112202323	1322	32113	2233
332231331	11222221211	122202133202233	0232		

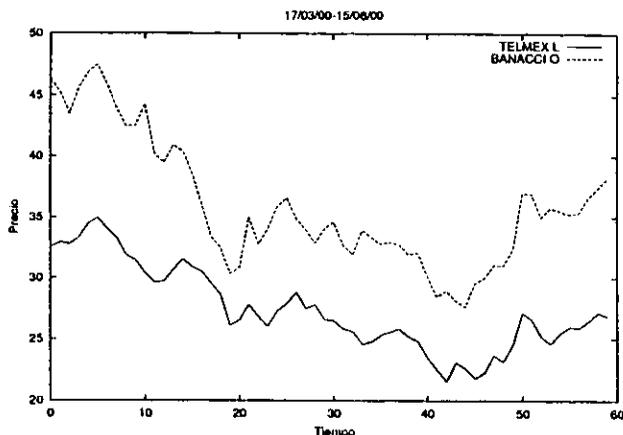


Figura 5.27: Precios del 17 de marzo al 15 de junio

Análisis A pesar de que este experimento es un modelo muy sencillo del manejo de portafolios de inversión ya que entre otras cosas no considera: costos de transacción y restricciones debidas a la normatividad de operación (cantidad máxima permitida de cada emisora, mínimo en el efectivo disponible, etc.) en la Bolsa Mexicana de Valores.

Período del 17 de marzo al 14 de abril: El manejo de las decisiones para TELMEX como para BANACCI condicionan una decisión fuerte a que los valores de la “Fuerza2” estén entre 0.5 y 0.8 y débil en los casos menores a 0.5 y entre 0.8 y 1, para BANACCI y para TELMEX en los casos menores a 0.5 y entre 0.8 y 1, la aparición de elementos que usen el operador de destrucción (δ) es limitada en el resto de los filtros, por lo que respecta a la cantidad a negociar este es menor al 50% de la máxima permitida permitiendo la configuración de un portafolio conservador que a pesar de la caída en los precios que se aprecia en la gráfica 5.22 este no se ve muy afectado.

Período del 17 de abril al 18 de mayo: Decisiones con poca fuerza para TELMEX y con una mayor fuerza para BANACCI el filtro correspondiente a la predictibilidad de

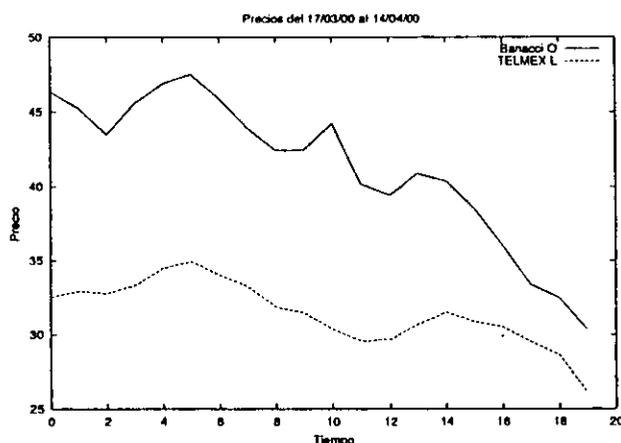


Figura 5.28: Precios del 17 de marzo al 14 de abril

TELMEX tiende a favorecer a aquellas con predictibilidad media y para BANACCI la preferencia aumenta conforme la predictibilidad. El portafolio generado por este individuo es altamente conservador dado su preferencia por negociar menos del 25% de sus acciones disponibles permitiendo una mayor ganancia al aprovechar el alza en las acciones de TELMEX hacia finales del periodo.

Período del 19 de mayo al 15 de junio:

En este período se aprecia una clara tendencia a la alza en las acciones que configuran el portafolio. El portafolio correspondiente al mejor individuo no presenta un aumento en las ganancias comparado con el período anterior. Dada la tendencia alcista de las emisoras el sistema se organiza favoreciendo la elección de operaciones de alto riesgo (manejo arriba de 75% del valor posible). Las decisiones tienden a la aceptación para BANACCI y en menor medida para las de TELMEX.

Las comparaciones de los portafolios para cada etapa ubican a los generados por el algoritmo genético un lugar atrás de una configuración 50/50 que para los periodos estudiados resultó la mejor.

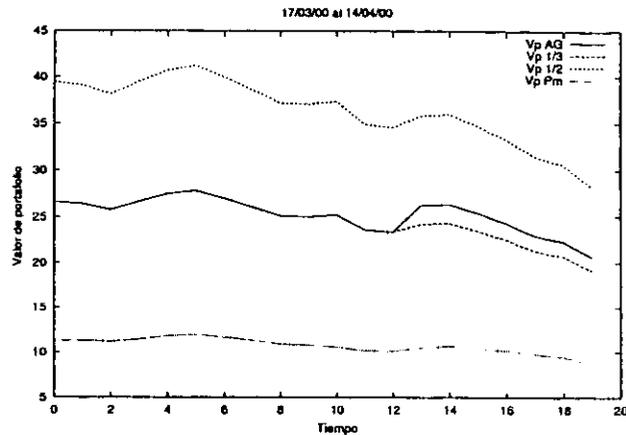


Figura 5.29: Valor de portafolio para diferentes configuraciones, del 17 de marzo al 14 de abril. AG: algoritmo genético. Pm: participación de mercado, 1/3 configuración, 1/3 Banacci O 1/3, Telmex L y 1/3 efectivo. 1/2: 1/2 Banacci O, 1/2 Telmex L

Para investigar capacidad predictora del algoritmo genético inteligente se hizo la comparación de las ganancias diarias porcentuales normalizadas (G_{dpm}) entre la mejor estrategia encontrada por el algoritmo genético en el período anterior y la estrategia de “Buy & Hold” (comprar y mantener) con proporción de 1/3 para cada una de las emisoras y el efectivo. Esta estrategia consiste en tener una tercera parte del portafolio invertido en TELMEX L y otra tercera parte en BANACCI O teniendo el resto del capital en efectivo, manteniendo la cantidad de acciones durante el período a comparar. En las gráficas se identifica como “Predicción” a los resultados generados por el algoritmo genético y como “B & H 1/3” a los generados por la estrategia de “Buy & Hold”. A pesar de que esta comparación no es totalmente justa ya que el uso de estrategias que funcionaron bien en el pasado no garantiza que lo harán en el futuro, dados la variación en los componentes determinísticos y estocásticos de una serie de precios. Se observa que en dos de los tres períodos considerados (17/03/00 a 14/04/00 y 19/05/00 a 15/06/00) las ganancias obtenidas por el algoritmo genético

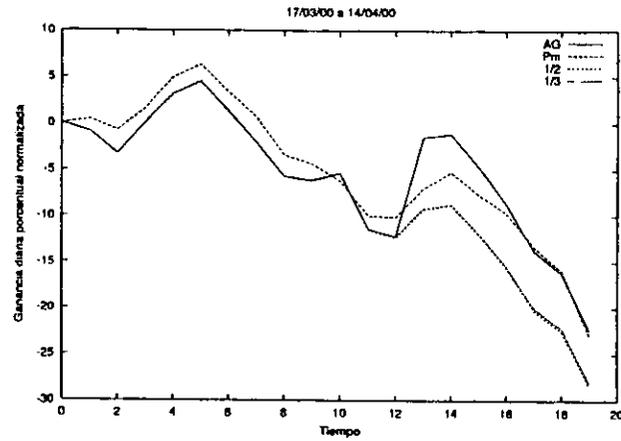


Figura 5.30: Ganancia diaria porcentual normalizada $G_{dpn}(t) = \frac{(G(t)-G(0)) \cdot 100}{G(0)}$

son superiores a las proporcionadas por “Buy & Hold” motivando el estudio y empleo de los algoritmos genéticos inteligentes para sistemas predictivos de manejo de portafolios.

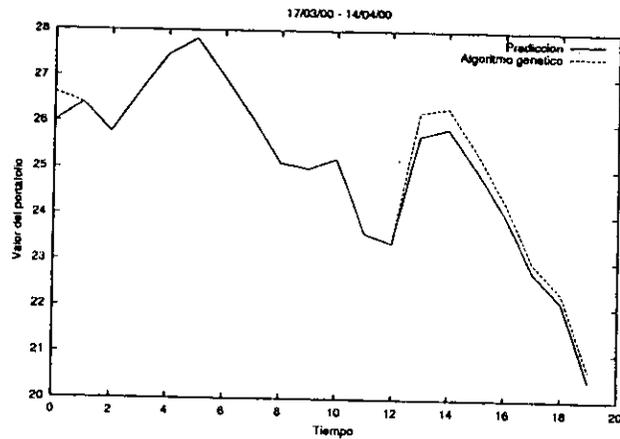


Figura 5.31: Valor de portafolio usando mejor estrategia del periodo anterior contra valor de portafolio usando estrategia generada a partir de los datos de este período del 17 de marzo al 14 de abril

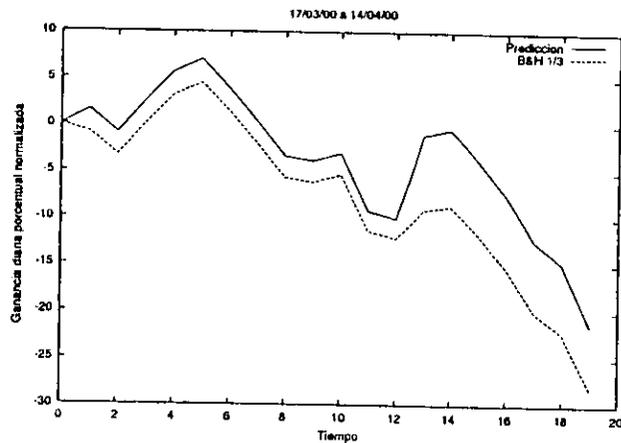


Figura 5.32: Comparación de ganancia diaria porcentual entre un portafolio que usa la mejor estrategia generada por un algoritmo genético inteligente para el período anterior (Predicción) y un portafolio que usa "Buy & Hold" con una inversión de 1/3 para cada una de las emisoras y el efectivo (B & H 1/3). Período del 17 de marzo al 14 de abril

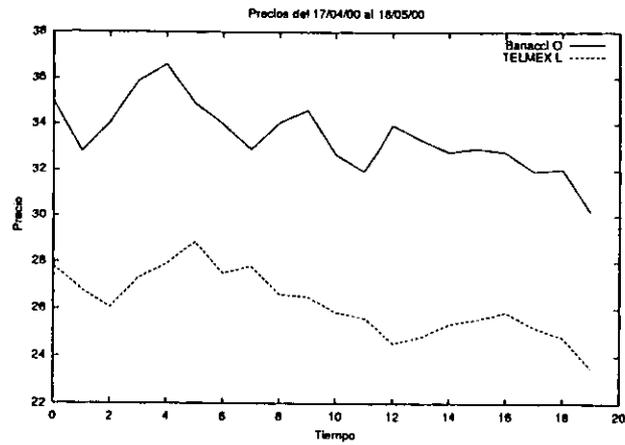


Figura 5.33: Precios del 17 de abril al 18 de mayo

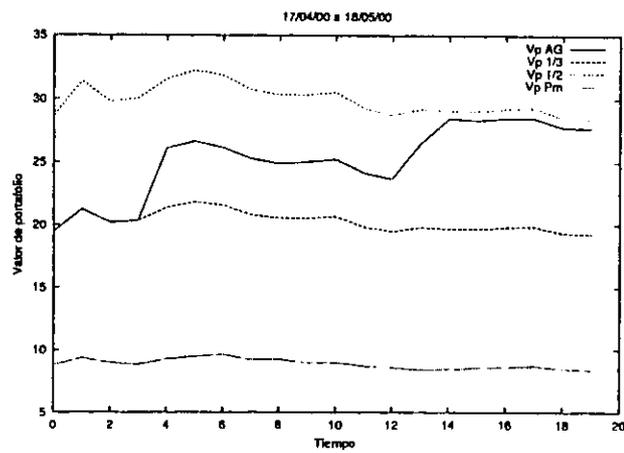


Figura 5.34: Valor de portafolio para diferentes configuraciones, del 17 de abril al 18 de mayo

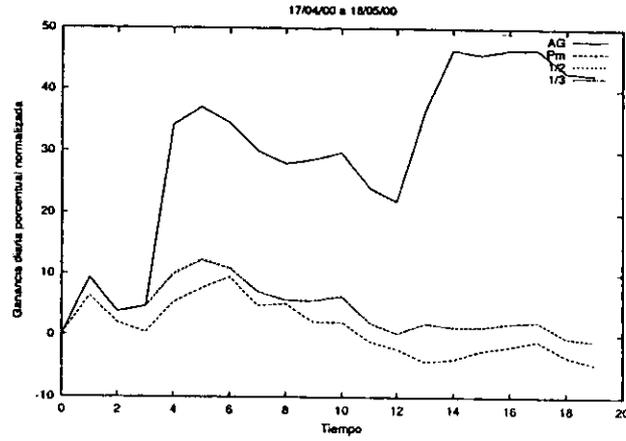


Figura 5.35: Ganancia diaria porcentual normalizada $G_{dpn}(t) = \frac{(G(t)-G(0))*100}{G(0)}$ del 17 de abril al 18 de mayo

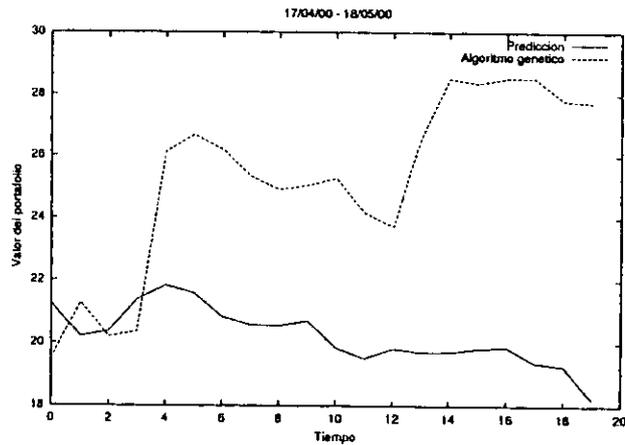


Figura 5.36: Valor de portafolio usando mejor estrategia del periodo anterior contra valor de portafolio usando estrategia generada a partir de los datos de este período del 17 de abril al 18 de mayo

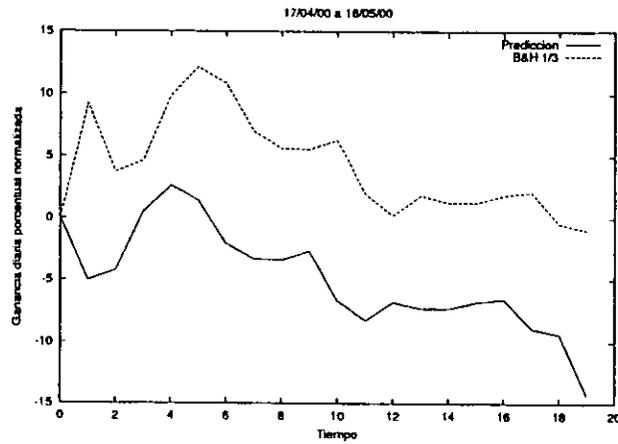


Figura 5.37: Comparación de ganancia diaria porcentual entre un portafolio que usa la mejor estrategia generada por un algoritmo genético inteligente para el período anterior (Predicción) y un portafolio que usa "Buy & Hold" con una inversión de 1/3 para cada una de las emisoras y el efectivo (B&H 1/3). Período del 17 de abril al 18 de mayo

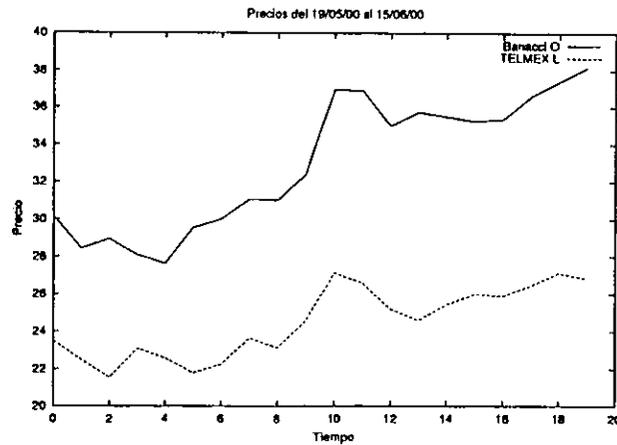


Figura 5.38: Precios del 19 de mayo al 15 de junio

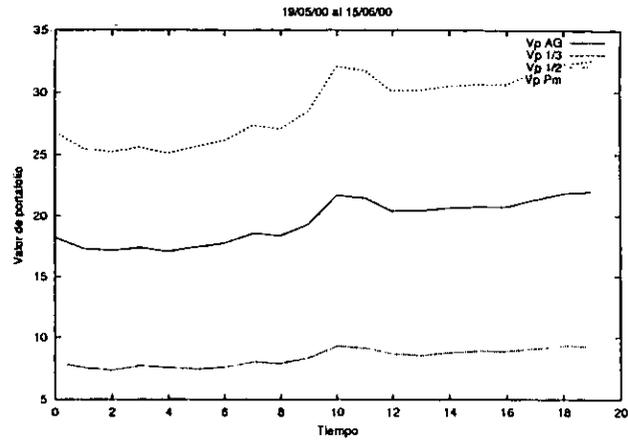


Figura 5.39: Valor de portafolio para diferentes configuraciones, del 19 de mayo al 15 de junio

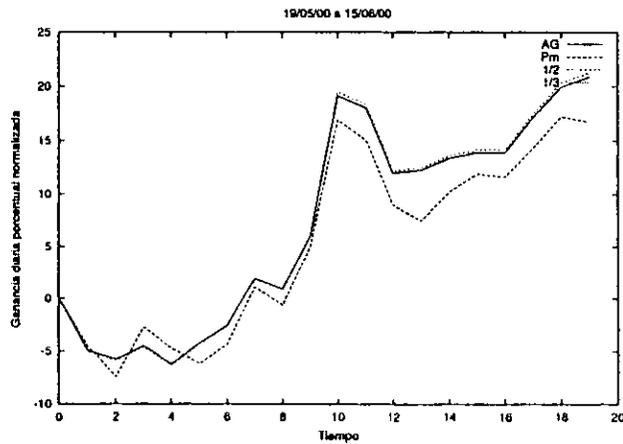


Figura 5.40: Ganancia diaria porcentual normalizada $G_{dpn}(t) = \frac{(G(t)-G(0))*100}{G(0)}$ del 19 de mayo al 15 de junio

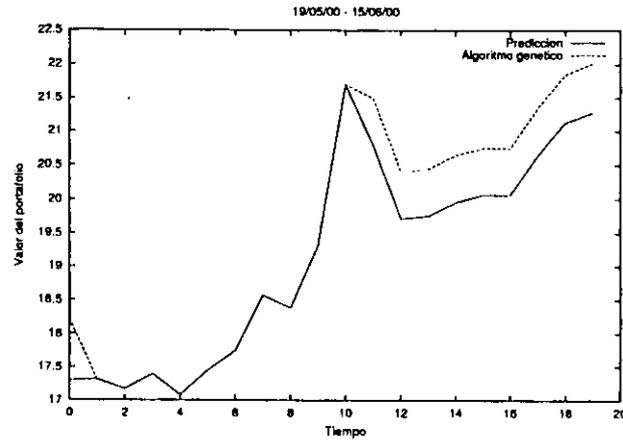


Figura 5.41: Valor de portafolio usando mejor estrategia del periodo anterior contra valor de portafolio usando estrategia generada a partir de los datos de este período del 19 de mayo al 15 de junio

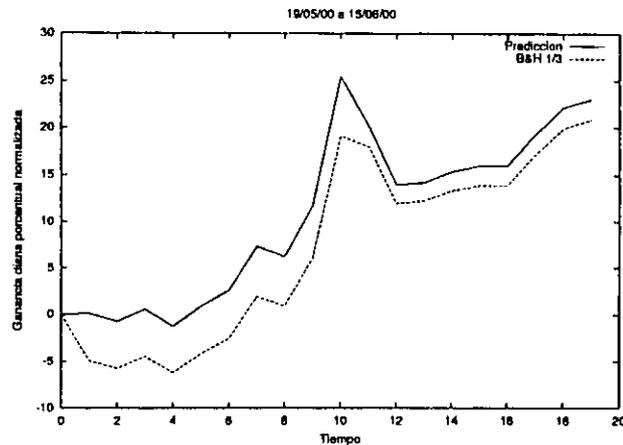


Figura 5.42: Comparación de ganancia diaria porcentual entre un portafolio que usa la mejor estrategia generada por un algoritmo genético inteligente para el período anterior (Predicción) y un portafolio que usa "Buy & Hold" con una inversión de 1/3 para cada una de las emisoras y el efectivo (B&H 1/3). Período del 19 de mayo al 15 de junio

Capítulo 6

Conclusiones

Esta tesis presentó las bases para el desarrollo de los algoritmos genéticos inteligentes al definir, explicar y aplicar conceptos como: mapeo genotipo-fenotipo, rompimiento inducido en la simetría de sinónimos y aptitud efectiva. Como se demuestra en esta tesis la importancia del mapeo genotipo-fenotipo consiste en permitir la redundancia en la información dada la existencia de sinónimos que responde de manera diferente a la acción de los operadores genéticos. Esta acción induce una jerarquía entre los sinónimos como resultado del rompimiento de simetría ya que algunos sinónimos son más robustos.

En este trabajo se demostró que en muchas situaciones el término de aptitud reproductiva no necesariamente corresponde al de aptitud efectiva siendo este último el que determina el éxito reproductivo total.

La principal característica de los algoritmos genéticos inteligentes es su capacidad de adaptación a cambios en el ambiente lo cual los hace una herramienta poderosa para atacar problemas combinatorias dinámicos, principalmente el manejo de contingencias. Tal adaptación se debe a la capacidad del algoritmo para autoorganizarse en el espacio de los genotipos por efecto de los operadores genéticos y la interacción del

fenotipo con la función a optimizar. El éxito de este tipo de algoritmos depende en gran medida de una codificación que permita un mapeo genotipo-fenotipo de muchos a uno para que los operadores genéticos puedan redireccionar la búsqueda del óptimo ante cambios en la función objetivo.

Tanto el modelo de virus como el de girafas muestran la emergencia espontánea de estructura en la producción de mutaciones a nivel fenotípico además de la naturaleza aleatoria y no direccional de las mutaciones a nivel del genotipo. Esto refleja la existencia de un proceso de autoorganización que mejora la habilidad del sistema para adaptarse a cambios en el ambiente. Para el caso del virus la mutabilidad propia promedio para un virus *in vivo* con un período de vida largo esta por arriba del promedio, manifestando el dominio de los codones más mutables dentro del cromosoma. Entonces se ha demostrado que un virus puede “organizar” sus mutaciones con el fin de evitar la respuesta del sistema inmunológico mediante la elección de mutaciones no-sinónimas. De todos los posibles epitopes que se pueden representar en ese modelo sólo una pequeña porción, $\approx 10\%$ se hacen presentes durante la infección. La única ventaja selectiva entre un epitope y otro esta dada por la capacidad del sistema inmune de reconocerlo. Así, cualquier mutación sobre uno del 85% restante todavía no reconocida por el sistema inmune, es favorecida. La ventaja efectiva para la selección a largo plazo de alta mutabilidad propia lleva al dominio de codones que favorecen las mutaciones hacia no-sinónimos.

El rompimiento de simetría inducido que resulta al favorecer codones mutables esta íntimamente relacionado a la propiedad no-inyectiva del intérprete que lleva a cabo el proceso de traducción del genotipo. Esto implica que los sinónimos que existen difieren por su evolución bajo el efecto de los operadores genéticos. Esta

acción induce una jerarquía entre los sinónimos como resultado del rompimiento de simetría. Para el caso del virus el operador genético relevante es la mutación puntual y la jerarquía se establece por la mutabilidad propia. El intérprete utilizado fue trivial y su única función fue establecer los tres estados jerárquicos posibles dependiendo de la mutabilidad propia (en este caso los estados fueron 1,2 y 3). Ya que el sesgo en los codones ocurre en este modelo sencillo permite concluir que la mutabilidad propia por si misma esta siendo seleccionada. El intérprete es independiente del paisaje de aptitud el cual depende sólo del fenotipo.

Por supuesto, es posible considerar el punto de vista de que no hay ninguna diferencia entre una función de aptitud no-inyectiva (i.e. muchos a uno) en los genotipos (i.e. codificación directa con un paisaje de aptitud "simétrico") y un intérprete no-inyectivo. Sin embargo, al considerar esta opción se pierde de vista que es más fácil comprender la evolución al saber como funciona el intérprete. En este caso, identificando al intérprete permite entender que la simetría de la función de aptitud inducida en los genotipos hace que diferentes codones pueden codificar para el mismo aminoácido pero con diferente mutabilidad. Esto permite al sistema producir los cambios necesarios para la adaptación.

En el modelo de el cuello de la girafa la adaptación que emerge durante el rompimiento de simetría inducido por el efecto de la mutación fue mucho más específico que en el modelo del virus. Es notable que la adaptación emergió en un modelo mucho más complicado. Estos modelos han probado que el rompimiento de simetría mejoran la adaptabilidad de las especies a cambios en el medio ambiente.

Los conceptos de rompimiento de simetría inducido y aptitud efectiva permiten

entender el fenómeno de la ortogénesis dentro de una interpretación darwiniana “estándar”: la ortogénesis no es más que la acción de la selección natural al nivel de la función de aptitud efectiva. En los trabajos del modelo neuro-genético presentado por Angeles [105] la ortogénesis se presenta al efecto de un operador mucho más complejo que no es la mutación. A pesar de que los modelos presentados están muy lejos de representar el mundo real ofrecen evidencia que el rompimiento de simetría inducido puede jugar un papel importante en los sistemas reales.

La relevancia en el empleo de los algoritmos genéticos inteligentes para un problema del campo de la investigación de operaciones radica en que requiere de pocas entradas que le permitirán reaccionar ante cambios en el sistema sin necesidad de alguna modificación por parte del usuario, especialmente en problemas reales donde los cambios se pueden presentar de manera aleatoria como por ejemplo: en la configuración de portafolios de inversión, la aparición de fallas en líneas de producción, modificaciones de rutas de transporte ante accidentes de tránsito o en la planeación de actividades dentro de una organización.

En cuanto al sistema de manejo de portafolios de inversión, la aplicación del algoritmo genético inteligente probó que aún cuando el sistema no obtuvo los mejores rendimientos, sus rendimientos fueron superiores a los de la mayoría de los portafolios contra los que se comparó además de no requerir ninguna modificación o información fuera de la señal de posición y del comportamiento histórico de los precios. En cuanto a su propiedad de predicción del comportamiento del mercado esta fue de corto plazo (menos de una semana) dada la volatilidad presente en el mercado. Motivados por los resultados respecto al desempeño de los algoritmos genéticos en este campo y por su posible éxito comercial se tiene como trabajos futuros el manejo de portafolios de

inversión con un mayor número de emisoras y que puedan usar diferentes instrumentos financieros como divisas y deuda, además de aplicar los resultados a otros mercados financieros como NASDAQ100 o S&P 500 en los Estados Unidos. Las implicaciones prácticas de esto están vinculadas con la adquisición de los datos y el desarrollo de reglas para su implementación genotípica.

Las líneas de investigación que abre este trabajo son: 1) aplicación de los algoritmos genéticos inteligentes a diversos problemas reales de optimización combinatoria dinámica (calendarización de actividades, portafolios de inversión, redes de transporte), 2) investigación teórica sobre los paisajes de aptitud de tales problemas, 3) aplicación de otros conceptos de la evolución natural como la co-evolución.

Bibliografía

- [1] R. A. Fisher, "*The Genetical Theory of Natural Selection*", Oxford, UK: Clarendon, (1930).
- [2] S. Wright, *The roles of mutation, inbreeding, cross breeding and selection in evolution*, En D. F. Jones, editor, *Proceedings of the sixth international congress on genetics*, Vol 1, pp 356-366, Brooklyn Botanic Garden, New York (1932).
- [3] J. B. S Haldane, *The Causes of Evolution*, Londres: Longman (1932).
- [4] T. Dobzhansky, *Genetics and the Origin of Species*, Nueva York: Columbia Univ. Press, (1937).
- [5] J. S. Huxley, *Evolution: The Modern Synthesis*, Londres: Allen y Unwin,(1942).
- [6] E. Mayr, *Systematics and the Origin of Species*, Nueva York: Columbia Univ. Press, (1942).
- [7] G. G. Simpson, *Tempo and Mode in Evolution*, Nueva York: Columbia Univ. Press, (1944).
- [8] G. L. Stebbins, *Variation and Evolution in Plants*, Nueva York: Columbia Univ. Press, (1950).

- [9] A. Hoffman, *Arguments on Evolution: A Paleontologist's Perspective*, Nueva York: Oxford Univ. Press, (1989).
- [10] J. Huxley, *The evolutionary process*, en J. Huxley, A. C. Hardy y E. B. Ford, editores *Evolution as a Process*: Nueva York, Collier, pp.9-63, (1963).
- [11] D. E. Wooldridge, *The Mechanical Man: The Physical Basis of Intelligent Life*, Nueva York: McGraw-Hill, (1968).
- [12] T. Dobzhansky, F. J. Ayala, G. L. Stebbins, y J. W. Valentine, *Evolution*. San Francisco, CA: W. H. Freeman, (1977).
- [13] T. Bäck, *Evolutionary Algorithms in Theory and Practice: evolution strategies, evolutionary programming, genetic algorithms*, Oxford Univ. Press (1996).
- [14] S. Kirkpatrick, C. D. Gelatt y M. P. Vecchi, *Optimization by Simulated Annealing*, *Science*, 220, pp. 671-680, (1983).
- [15] M. A. Gutiérrez, *La técnica del recocido simulado y sus aplicaciones*, tesis doctoral, DEPMI-UNAM, México, (1991).
- [16] F. Glover, *Future Paths for Integer Programming and Links to Artificial Intelligence*, *Computers and Operations Research*, 13, pp 533-549, (1986).
- [17] P. Hansen *The Steepest Ascent Mildest Descent Heuristic for Combinatorial Programming* en *Congress on Numerical Methods in Combinatorial Optimization*, Capri, Italia. (1986).
- [18] F. Glover, *Tabu Search, Part I*, *ORSA Journal on Computing*, 1:3, pp 190-206, (1989).

- [19] F. Glover, *Tabu Search, Part II*, ORSA Journal on Computing, 2:1, pp 4-31, (1990).
- [20] F. Glover, *Tabu Search: A Tutorial*, Interfaces, 20:4, pp 74-94, (1990).
- [21] S. G. De los Cobos, *La técnica de la búsqueda tabú y sus aplicaciones*, tesis doctoral, DEPMI-UNAM, México, (1994).
- [22] D. B. Fogel, *Evolutionary computation: Toward a New Philosophy of Machine Intelligence*. Piscataway, NJ. IEEE Press, (1995).
- [23] C. Darwin, *On the Origin of Species by Means of Natural Selection or the Preservation of Favoured Races in the Struggle for Life*, Murrau, Londres, UK. (1859).
- [24] J. Maynard-Smith, *Evolutionary Genetics*, Oxford Univ. Press, Oxford, UK (1994).
- [25] L. Orgel, *Selection in vitro*, Proc. Royal Soc. of London (1979).
- [26] W. Banzhaf, P. Nordin, R. E. Keller y F. D. Francone, *Genetic Programming, An Introduction*, Morgan Kauffman Publishers, Inc. San Francisco, CA. (1998).
- [27] J. D. Watson, N. H. Hopkins, J. W. Roberts, J. A. Steitz, A. M. Weiner, *Molecular Biology of the Gene*, Benjamin-Cummings, Menlo Park, CA. (1987).
- [28] M. Eigen, *Steps Toward Life*, Oxford Univ. Press, Oxford, UK (1992).
- [29] W. Johannsen, *The genotype conception of heredity*, The American Naturalist, 45:129-159, (1911).

- [40] A. Törn y A. Žilinskas, *Global Optimization*, Lecture Notes in Computer Science, vol 350. Berlin: Springer, (1989).
- [41] J. Holland, *Adaptation in natural and artificial systems*. MIT press, Cambridge, MA (1975).
- [42] K. De Jong, *An Analysis of the Behaviour of a Class of Genetic Adaptive Systems*, tesis doctoral, Univ. of Michigan, (1975).
- [43] R. A. Caruna y J. D. Schaffer, *Proceedings of the 5th Int. Conf. on Machine Learning*, 153 (Morgan Kaufmann 1988).
- [44] , H. J. Bremermann, M. Rogson y S. Salaff, *Global properties of evolution process*, eds H. H. Pattec, E. A. Edelsack, L. Fein y A. B. Callahan, *Natural Automata and Useful Simulations*, Washington DC: Spartan, (1966).
- [45] T. Back, *Optimal mutation rates in genetic search*, S. Forrest, Ed. *Proc. 5th. Int. Conf on Genetic Algorithms*, San Mateo CA: Morgan Kauffman, (1989).
- [46] D. E. Goldberg, *Genetic Algorithms in search, optimization and machine learning*, Addison Wesley, Reading, MA (1989).
- [47] D. B. Fogel, *System Identification through Simulated Evolution: A Machine Learning Approach to Modeling*, (Ginn Press, Needham, MA, 1991).
- [48] J. H. Holland, *Scientific American*, Julio (1992).
- [49] L. Fogel, A. Owens y M. Walsh, *Artificial Intelligence through a simulation of evolution*, M. Maxfield, A. Callahan y L. Fogel, editores, *Biophysics and Cybernetic Systems* pp 131-155, (1965).

- [59] M. D. Vose y G. E. Leipnins, *Punctuated equilibria in genetic search*, *Complex syst.* 5 (1), 31-44 (1991).
- [60] M. D. Vose, A. H. Wright, *Simple genetic algorithms with linear fitness*, *Evolut. Computat.* 2(4). 347-368 (1995).
- [61] J. Shapiro y A. Prüguel-Bennett, *Genetic algorithm dynamics in a two-well potential*, *Foundations of Genetic Algorithms 4*
- [62] A. Prüguel-Bennett y J. Shapiro, *The dynamics of a genetic algorithm for simple random Ising systems*, *Physica D*, (1997).
- [63] H. Beyer, *An alternative explanation for the manner in which genetic algorithms operate*, *Biosystems* 41, 1-15, (1997).
- [64] D. B. Fogel, *Evolutionary Computation: The Fossil Record*, IEEE Press (1998).
- [65] T. C. Jones, *Evolutionary Algorithms, Fitness Landscapes and Search*, tesis doctoral, Univ. of New Mexico (1995).
- [66] M. Eigen *Self-Organization of Matter and the Evolution of Biological Macromolecules*, *Naturwissenschaften* 58: 465 (1971).
- [67] J. Horn and D. E. Goldberg, *Genetic Algorithm Difficulty and the Modality of Fitness Landscapes*, In D. Whitley and M. D. Vose, Editores, *FOGA 3*, pp 243-271, Morgan Kaufmann, San Francisco (1995).

- Workshop on Genetic Programming: From Theory to Real World Applications*, Tahoe City, CA, USA, pp. 6-22 (1995).
- [76] D. E. Goldberg, *Genetic Algorithms and Walsh Functions: Part I a Gentle Introduction*, *Complex Systems* 3, pp 123-152 (1989).
- [77] D. E. Goldberg, *Genetic Algorithms and Walsh Functions: Part II deception and its analysis*, *TCGA Report* No. 89001, Univ. of Alabama, Tuscaloosa (1989).
- [78] C. R. Stephens, *Effective Fitness Landscapes in Evolution*, In P. Angeline, Editor, *CEC99*, pp. 703-714, IEEE Press (1999).
- [79] R. Poli, *Exact Schema Theorem and Effective Fitness for GP with One-Point Crossover*, Proceedings of GECCO 2000.
- [80] C. R. Stephens, *Effect of Mutation and Recombination on the Genotype-Phenotype Map*, en W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honovar, M. Jakiela and R. E. Smith, Eds, *GECCO99*, pp. 1382-1390, Morgan-Kaufmann (1999).
- [81] S. A. Kauffman, *The Origins of Order*, Oxford University Press, Oxford (1993).
- [82] L. B. Booker, "Recombination Distributions for Genetic Algorithms", *Foundations of Genetic Algorithms 2*, ed. D. Whitley, (Morgan Kaufmann, San Mateo) 29-45.(1993).
- [83] H. A. Simon *Cognitive Science: The Newest Science of the Artificial*, *Cognitive Science* 4:33-46 (1980).
- [84] A. H. Wright, *Foundations of Genetic Algorithms*, 205 (Morgan Kaufmann 1991).

- [94] T. Bäck, *Self-Adaptation in Genetic Algorithms*, en F. J. Varela y P. Bourguine editores, *Proceedings of the First European Conference on Artificial Life*, MIT Press, Boston, (1992).
- [95] D. Bowen, *A Study of the Effects of Internally Determined Crossover and Mutation Rates on Genetic Algorithm Optimization*, Univ. of Alabama, Tuscalosa, (1986).
- [96] J. J. Grefenstette, *IEEE Transactions on Systems, Man and Cybernetics*, SMC-16(1), 122 ,1986.
- [97] J. D. Schaffer, R. A. Carauna, L. J. Eshelman and R. Das, *Proceedings of the Third Int. Conf. on Genetic Algorithms and their Applications*, 51-60, Morgan Kaufmann, (1989).
- [98] W. E. Hart and R. K. Belew, *Proceedings of the Fourth Int. Conf. on Genetic Algorithms and their Applications*, 190-195, Morgan Kaufmann, (1991).
- [99] S. A. Kauffman, *Adaptation on rugged fitness landscapes*, en: *Lectures in the Sciences of Complexity*. D. Stein (ed.) (Reading, MA, Addison-Wesley) pp. 527-618, (1989).
- [100] A. Homaifar, S. Guan y G. E. Liepins, *A new approach on the Travelling Salesman Problem by Genetic Algorithms*, en *Proceedings of the 5th Int. Conf. on Genetic Algorithms and their Applications*, 460-466, Morgan Kaufmann, (1993).
- [101] R. L. Karg y G. L. Thompson *A Heuristic Approach to Solving Travelling Salesman Problems*, *Management Science*, 10, 225, (1964).