

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
INGENIERÍA EN INFORMÁTICA

**MODELO DE CODIFICACIÓN Y DESARROLLO DE OPERADORES
GENÉTICOS PARA LA OPTIMIZACIÓN DE REDES DE NEURONAS
DE INTEGRACIÓN Y DISPARO**

Realizado por
D. Francisco José Torres Álvarez

Dirigido por
D. Francisco J. Veredas Navarro

Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA

Málaga, Julio 2003

UNIVERSIDAD DE MÁLAGA
ESCUELA TECNICA SUPERIOR DE INGENIEROS EN INFORMÁTICA
INGENIERÍA EN INFORMÁTICA

Reunido el tribunal examinador en el día de la fecha, constituido por:

Presidente D./D^a. _____

Secretario D./D^a. _____

Vocal D./D^a. _____

Para juzgar el proyecto Fin de Carrera titulado:

MODELO DE CODIFICACIÓN Y DESARROLLO DE OPERADORES
GENÉTICOS PARA LA OPTIMIZACIÓN DE REDES DE NEURONAS DE
INTEGRACIÓN Y DISPARO

del alumno D. *Francisco José Torres Álvarez*

dirigido por D. *Francisco J. Veredas Navarro*

ACORDÓ POR _____ OTORGAR LA CALIFICACIÓN DE _____

Y PARA QUE CONSTE, SE EXTIENDE FIRMADA POR LOS COMPARECIENTES DEL TRIBUNAL, LA PRESENTE DILIGENCIA.

Málaga, a ____ de _____ de _____

El Presidente

El Secretario

El Vocal

Fdo: _____

Fdo: _____

Fdo: _____

Agradecimientos

A D. Francisco J. Veredas Navarro, director de este proyecto, por confiarme la realización del mismo y ser un excelente guía.

A los componentes del GEB, por todo lo que he aprendido de ellos en el tiempo que ha durado esta aventura.

A mis amigos, por su colaboración para hacer más llevadero este largo viaje.

*A mi familia,
por ser mis guías en la vida.*

*A Isabel,
por ser la fuente de mi inspiración.*

Índice

RESUMEN	9
1 INTRODUCCIÓN	11
<i>1.1 Las redes neuronales</i>	<i>13</i>
1.1.1 Bases de la neurofisiología animal	13
1.1.2 Redes neuronales biológicas.....	17
1.1.3 Redes neuronales artificiales	19
1.1.4 Modelado de sistemas neuronales biológicos.....	22
1.1.5 Modelo neuronal de integración y disparo	24
1.1.6 Una herramienta de análisis: el correlograma cruzado	26
<i>1.2 Técnicas evolutivas</i>	<i>28</i>
1.2.1 Las teorías evolutivas	29
1.2.2 Mecanismos de cambio en la evolución	30
1.2.3 Historia de la computación evolutiva	32
1.2.4 Algoritmos genéticos.....	34
2 ESTUDIO DE UN ALGORITMO GENÉTICO BASADO EN CODIFICACIÓN CELULAR	59
<i>2.1 Modelo básico de codificación celular</i>	<i>60</i>
<i>2.2 El modelo de codificación celular adaptado</i>	<i>64</i>
2.2.1 Modelo de red neuronal.....	64
2.2.2 El conjunto de genes.....	66

2.2.3 La estructuración del genoma.....	67
2.2.4 La población inicial	68
2.2.5 Los operadores genéticos	71
2.2.6 La función de evaluación.....	71
2.3 Políticas de selección.....	72
2.4 Pruebas y resultados.....	75
3 CODIFICACIÓN CELULAR ADAPTADO A REDES DE NEURONAS DE INTEGRACIÓN Y DISPARO	89
3.1 Estudio del modelo de neurona de integración y disparo.....	90
3.1.1 El ruido.....	91
3.1.2 La influencia del peso sináptico en la actividad neuronal.....	95
3.2 Adaptación del algoritmo genético	97
3.2.1 El conjunto de genes.....	98
3.2.2 Estructuración del genoma	101
3.2.3 La población inicial	102
3.3 La función de evaluación	103
3.4 Actividad en los circuitos neuronales	106
3.5 Pruebas y resultados.....	106
4 MATERIALES Y MÉTODOS	119
4.1 Estructuración del proyecto.....	119
4.2 Ciclo de vida y metodologías software	122
4.3 Desarrollo de aplicaciones.....	124
4.3.1 Análisis software. Modelo conceptual.....	124

4.3.2 Uso de las aplicaciones.....	130
4.4 Recursos empleados.....	135
5 CONCLUSIONES Y TRABAJOS FUTUROS	137
BIBLIOGRAFÍA	141

Resumen

El presente proyecto propone un método de codificación de redes de *neuronas de integración y disparo* para su aplicación en la optimización de conexionados mediante de algoritmos genéticos.

El estudio de las propuestas de codificación de redes de neuronas artificiales existentes en la literatura actual ha permitido conocer los distintos acercamientos realizados y sus posibles adaptaciones al modelo neuronal de integración y disparo. El modelo propuesto en el presente trabajo denominado modelo de *codificación celular adaptado* parte del modelo de propuesto por F. Gruau en el que el genoma es representado por un árbol de ramas ordenas cuyos nodos representan genes: instrucciones para el desarrollo de una red neuronal a través de un proceso denominado *desarrollo embrionario*. Los operadores genéticos utilizados para la generación de individuos a partir de otros ya existentes son versiones de los clásicos operadores *mutación* y *cruzamiento* adaptados a las nuevas características del genoma.

Un primer prototipo del modelo desarrollado en este trabajo es aplicado en la optimización de topologías de redes binarias para la resolución de problemas lógicos. Sobre este primer prototipo se hace patente la necesidad de aumentar la eficacia del algoritmo mediante la estructuración del genoma, permitiendo la reducción el espacio de búsqueda a sólo topologías de redes válidas. El estudio de la evolución de las poblaciones, producidas por la aplicación del algoritmo genético con diversas políticas de selección y sustitución de individuos, ha permitido evaluar la eficiencia de unos métodos sobre otros y su mayor o menor sensibilidad a la reducción del tamaño de las poblaciones.

Los resultados obtenidos en las pruebas realizadas sobre el modelo de *codificación celular adaptado* para el desarrollo de redes de *neuronas binarias* permiten inferir el potencial del modelo para el desarrollo de modelos neuronales más complejos.

La aplicación del modelo de *codificación celular adaptado* para el desarrollo de redes de *neuronas de integración y disparo* ha requerido un estudio previo del modelo neuronal utilizado, donde se han establecido los valores más adecuados para sus distintos parámetros. La evaluación de los individuos generados se realiza a través de la comparación del *correlograma cruzado*, generado a partir de la actividad de dos de sus neuronas, con otro de referencia aportado por el experimentador.

Capítulo 1

Introducción

En los últimos años son numerosas las ramas de la ciencia que se ayudan de modelos matemáticos y computacionales para representar el comportamiento de los diversos objetos de estudio. Con estas representaciones se facilita el estudio de dichos comportamientos bajo condiciones controladas y se trata de anticipar los resultados que se producirían en condiciones concretas. En la actualidad son numerosos los modelos matemáticos que ayudan a analizar los más diversos sistemas en ámbitos muy distintos de la ciencia.

A su vez, las ciencias computacionales, se han inspirado en numerosas ocasiones en los procesos naturales estudiados en otras disciplinas científicas para encontrar nuevos métodos de resolución de problemas, utilizando las mismas herramientas que “emplea” la naturaleza. Éste es el caso de las ideas aportadas por la Neurofisiología, que produjeron el nacimiento de las redes de neuronas artificiales, basadas en el uso de un conjunto de pequeñas unidades de computación entrelazadas, denominadas *neuronas*, inspiradas en los primeros descubrimientos del funcionamiento del sistema nervioso de los animales y aplicadas a problemas de reconocimiento de caracteres, de reconocimiento del habla, etc. El campo de la genética y las teorías evolutivas ha aportado las bases de las técnicas conocidas como algoritmos evolutivos para la resolución de las más variadas clases de problemas.

En el presente trabajo, se aúnan ambas vertientes, aplicando técnicas de búsqueda inspiradas en paradigmas evolutivos para la optimización de topologías de redes neuronales, que imitan ciertos comportamientos de redes biológicas. Más concretamente, se pretende desarrollar un método de búsqueda que permita optimizar los conexiones de redes de neuronas de integración y disparo, basado en algoritmos genéticos, mediante un modelo de codificación adecuado y sus respectivos operadores genéticos. Para ello se persiguen los siguientes objetivos:

1. Desarrollo de un método de codificación de redes de neuronas de integración y disparo, para su aplicación en algoritmos genéticos, fácilmente portable a modelos neuronales más complejos.

2. Desarrollo de un conjunto de operadores genéticos adecuados que permitan el empleo eficiente del modelo de codificación al que se alude en el objetivo anterior.
3. Optimización del conexionado de redes de neuronas de integración y disparo para la búsqueda de topologías alternativas a configuraciones preestablecidas por un experimentador.

El primer objetivo que se pretende alcanzar consiste en el diseño de un método de codificación adecuado para redes de neuronas que sigan el modelo de integración y disparo. El modelo de codificación diseñado pretenderá evitar los problemas que presentan la mayoría de los métodos de codificación explícitos, fundamentalmente la llamada *disrupción del cruzamiento*, permitiendo la convergencia de la población de los algoritmos genéticos que sigan este modelo de codificación en sus poblaciones de individuos. A partir de este nuevo modelo de codificación será necesario diseñar e implementar, como segundo objetivo fundamental de este proyecto, una serie de operadores genéticos que puedan operar sobre dicho modelo. Básicamente, los operadores necesarios serán los de *cruzamiento* (*crossover*) y *mutación* (*mutation*). Deberán ser diseñados de tal manera que permitan la convergencia eficiente de las poblaciones de los algoritmos genéticos en los que sean utilizados, evitando, sobre todo, el problema de la disrupción del cruzamiento antes mencionado. Por último, el objetivo final consiste en la optimización del conexionado en redes de neuronas de integración y disparo, sobre la base del modelo de codificación referido en el objetivo 1 y a partir de la aplicación de los operadores genéticos diseñados según lo expuesto en el objetivo 2. A partir de los circuitos básicos propuestos por neurofisiólogos como modelos de funcionamiento de subsistemas elementales en determinados sistemas cerebrales, como puede ser la vía visual de los mamíferos, el objetivo final de este proyecto consiste en la búsqueda de topologías alternativas a estos circuitos.

En el presente capítulo se introduce al lector en los principales conceptos y técnicas empleadas en el desarrollo de redes neuronales y algoritmos genéticos y su historia. En el capítulo 2 se estudia la aplicación de un algoritmo genético con codificación implícita para la optimización de redes neuronales artificiales en la resolución de problemas lógicos. El estudio de los diversos operadores necesarios así como de su comportamiento, sirven de base para la aplicación del algoritmo genético en modelos de redes de neuronas de integración y disparo, desarrollado en el capítulo 3. El capítulo 4 desarrolla las fases en las que se ha estructurado el proyecto y las técnicas utilizadas para el desarrollo de las aplicaciones que han hecho posibles los estudios. Finalmente, en el capítulo 5 se reúnen las principales conclusiones a las que se ha llegado con la realización del proyecto y se proponen vías de estudio para trabajos futuros.

1.1 Las redes neuronales

Una parte importante del presente trabajo se enmarca en el campo de las redes neuronales, sistemas de computación inspirados en el funcionamiento del sistema nervioso de los seres vivos. A lo largo de su historia, los descubrimientos en Neurofisiología han influido de forma determinante en el desarrollo de estos sistemas. No obstante, las redes neuronales artificiales han dado a luz a un gran número de enfoques, algunos más alejados que otros de las bases biológicas iniciales. Ciertos modelos, como el de *integración y disparo* usado en este proyecto, buscan un acercamiento al funcionamiento real de los sistemas nerviosos de los seres vivos, imitando ciertas características del comportamiento neuronal tales como: el período refractario, la hiperpolarización de la membrana después de cada disparo, etc.

En los siguientes apartados se introducirán las bases de la neurofisiología animal (apartado 1.1.1) y las redes de neuronas biológicas (apartado 1.1.2). En el apartado 1.1.3 se repasan las principales propuestas de *modelos conexionistas* a lo largo de su historia. El modelado de sistemas neuronales biológicos es abordado en el apartado 1.1.4, y el modelo neuronal de integración y disparo que es adaptado en el presente trabajo es presentado en el apartado 1.1.5. Finalmente se presenta una herramienta de análisis para el estudio de circuitos neuronales denominada *correlograma cruzado* (apartado 1.1.6).

1.1.1 Bases de la neurofisiología animal

El desarrollo de redes neuronales hunde sus raíces en los descubrimientos llevados a cabo en el campo de la neurofisiología. Se puede considerar como padre de las redes neuronales al científico aragonés Santiago Ramón y Cajal, descubridor de la estructura neuronal del sistema nervioso. A finales del siglo XIX la creencia más extendida en la medicina era la teoría reticulista, que sostenía que el sistema nervioso estaba formado por una red de fibras nerviosas (Müller & Reinhardt, 1990). Sin embargo, en 1888 Ramón y Cajal demostró que en realidad estaba compuesto por una red de neuronas individuales, las neuronas, ampliamente interconectadas entre sí. Paradójicamente, Ramón y Cajal realizó su descubrimiento gracias a

la ingeniosa aplicación de una técnica de tinción introducida por un raticulista afamado, Camilo Golgi. Gracias a ella, pudo observar al microscopio los pequeños vacíos que separaban unas neuronas de otras, así como establecer que la información fluye en la neurona desde las dendritas hacia el axón, atravesando el soma. Aunque este descubrimiento fue recibido con escepticismo por la comunidad científica internacional, poco más tarde fue ratificado por otros investigadores, causando una verdadera conmoción en la forma de entender el sistema nervioso. En 1906, se concedió el premio Nobel de medicina a Golgi y a Ramón y Cajal. Hoy día, el trabajo de Ramón y Cajal sigue siendo muy citado, no sólo a causa de su gran descubrimiento, sino también por su monumental obra posterior, en la que llegó a descubrir buena parte del sistema nervioso de los vertebrados.

Gracias al advenimiento de la microscopía electrónica y a la introducción de otras importantes técnicas, se ha llegado a profundizar mucho más en el estudio de la neurona (ver, por ejemplo, [(Kandel, Schwartz, & Jessell, 1999)]). Se estima que el sistema nervioso contiene alrededor de mil millones de neuronas. Este tipo de células pueden presentarse, vistas al microscopio, en múltiples formas, aunque muchas de ellas presentan un aspecto similar, con un cuerpo celular o soma (de entre 10 y 80 micras de longitud), del que surge un denso árbol de ramificaciones (árbol dendrítico) compuesto por las dendritas y del cual parte una fibra tubular denominada axón (cuya longitud varía desde las 100 micras hasta el metro en el caso de neuronas motoras), que también se ramifica en su extremo final para conectar otras neuronas.

Desde el punto de vista funcional, las neuronas constituyen procesadores de información sencillos. Como todo sistema de este tipo, poseen un canal de entrada de información, las *dendritas*, un órgano de cómputo en el núcleo celular, el *soma*, y un canal de salida, el *axón* (figura 1.1). En las interneuronas, el axón envía la información a otras neuronas, mientras que en las neuronas motoras lo hace directamente al músculo. Existe un tercer tipo de neuronas, las receptoras o sensoras, que en vez de recibir información de otras neuronas, las reciben directamente del exterior (tal sucede, por ejemplo, en los conos y los bastones de la retina). Se calcula que una neurona de la corteza cerebral recibe información, por término medio, de unas 10.000 neuronas (convergencia), y envía impulsos a varios cientos de miles de ellas (divergencia).

En la corteza cerebral se aprecia la existencia de una organización horizontal en capas (se suelen señalar unas seis), coexistiendo en una organización vertical en forma de columnas de neuronas. Hay grupos neuronales, compuestos por millones de neuronas pertenecientes a una determinada región del cerebro, que constituyen unidades funcionales especializadas en ciertas tareas (por ejemplo, existen un área visual, un área auditiva, una corteza senso-motora,

etc.); todos los subsistemas juntos conforman el encéfalo. Se tiene evidencia de que el procesamiento en el sistema nervioso involucra la actuación de muchos de tales subsistemas, que intercambian continuamente información.

La unión entre dos neuronas se denomina *sinapsis*. En el tipo de sinapsis más común no existe un contacto físico entre las neuronas, sino que éstas permanecen separadas por un pequeño vacío de unas 0,2 micras. Con relación a la sinapsis, se habla de neuronas presinápticas (las que envían las señales) y postsinápticas (las que las reciben). La sinapsis son direccionales, es decir, la información fluye siempre en un único sentido.

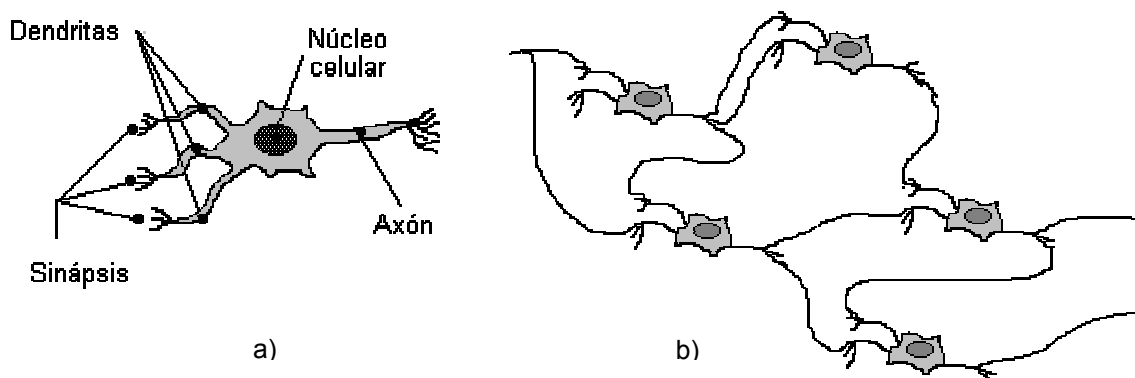


Fig. 1.1. (a) Representación de una neurona biológica y sus diversos componentes. (b) Conexión de un conjunto de neuronas formando un circuito neuronal.

Las señales nerviosas se pueden transmitir eléctrica o químicamente. La transmisión química prevalece fuera de la neurona, mientras que la eléctrica lo hace en el interior. La transmisión química se basa en el intercambio de neurotransmisores, mientras que la eléctrica hace uso de descargas que se producen en el cuerpo celular y que se propagan por el axón.

El fenómeno de generación de una señal nerviosa está determinado por la membrana neuronal y los iones presentes en ambos lados de ella. La membrana se comporta como un condensador que se carga al recibir corrientes debidas a las especies iónicas presentes. La membrana contiene canales iónicos selectivos al tipo de ión, algunos son pasivos (consisten en simples poros de la membrana) y otros activos (poros que solamente se abren ante ciertas circunstancias). En esencia, las especies iónicas más importantes, que determinan buena parte de la generación y propagación del impulso nervioso, son sodio, potasio y calcio, además de los iones de proteínas.

La forma más habitual de comunicación entre dos neuronas es de tipo químico. La neurona presináptica libera unas sustancias químicas complejas denominadas *neurotransmisores* (como el glutamato o la adrenalina), que atraviesan el vacío sináptico. Si la neurona postsináptica posee en las dendritas o en el soma canales sensibles a los neurotransmisores liberados, los fijará, y como consecuencia de ello permitirán el paso de determinados iones a través de la membrana. Las corrientes iónicas que de esta manera se crean provocan pequeños potenciales postsinápticos, excitadores (positivos) o inhibidores (negativos), que se integrarán en el soma, tanto espacial como temporalmente; éste es el origen de la existencia de sinapsis excitadoras y de sinapsis inhibitoras. Si se ha producido un suficiente número de excitaciones, la suma de los potenciales positivos generados puede elevar el potencial de la neurona por encima de los -40 mV (umbral de disparo habitual); en ese momento se abren bruscamente los canales de sodio, de modo que los iones sodio, cuya concentración en el exterior de la membrana es alta, entran masivamente al interior, provocando la despolarización brusca de la neurona, que pasa de un potencial de reposo de unos -70 mV a unos $+10$ mV. A continuación la neurona vuelve a la situación original de reposo de -70 mV; este proceso constituye la generación de un potencial de acción (también llamado disparo de la neurona o *spike*), que al propagarse a lo largo del axón da lugar a la transmisión eléctrica de la señal nerviosa. Tras haber sido provocado un potencial de acción, la neurona sufre un período refractario, durante el cual no puede generarse uno nuevo.

Es importante destacar el hecho de que el pulso así generado es digital, en el sentido de que existe o no existe el pulso, y todos ellos son de la misma magnitud. Por otra parte, una estimulación más intensa hace que disminuya el intervalo entre pulsos, por lo que la neurona se disparará a mayor frecuencia cuanto mayor sea el nivel de excitación que tenga. Es decir, la excitación se codifica en la frecuencia de los pulsos producidos. Por otra parte, no todas las neuronas se disparan generando un tren de pulsos de una frecuencia aproximadamente constante, pues la presencia de otras especies iónicas hace que diferentes tipos de neuronas posean patrones de disparo distintos, en forma de trenes puros, paquetes de pulsos o presentando patrones más complejos (Shepherd, 1990)

1.1.2 Redes neuronales biológicas

Los estudios de mayor relevancia realizados en el campo de la neurofisiología se centran en la vía visual de los mamíferos (por su factible extrapolación posterior al ser humano). Ésta está formada por estructuras de gran complejidad que parten del ojo y finalizan en la corteza cerebral. En cada una de las estaciones intermedias desde la retina hasta la corteza se encuentran diferentes tipos de neuronas, con características fisiológicas y anatómicas muy variables. En este sentido, la mayor complejidad se encuentra en la corteza visual. Los patrones de conexionado entre las diferentes etapas de la vía visual, así como dentro de ellas mismas, están sujetos a un intenso estudio desde diferentes enfoques y en muchos laboratorios a lo largo y ancho del mundo. Aunque existen teorías con amplia aceptación acerca de las pautas de conexionado entre neuronas, como puede ser el llamado modelo jerárquico, continuamente aparecen nuevos estudios y resultados que corroboran o contradicen lo dicho en esas teorías. Una de las mayores dificultades a la hora de sacar conclusiones, consiste en la imposibilidad de aplicar técnicas "directas" de visualización de las conexiones, por lo que se hace necesario emplear métodos estadísticos e indirectos de los que se obtienen resultados con un alto grado de incertidumbre. Los registros neurofisiológicos de los disparos producidos por neuronas de diferentes áreas o capas de la corteza ante la presentación de patrones de estimulación controlada y su posterior análisis con métodos estadísticos (normalmente, computerizados) constituyen una de las principales técnicas empleadas en este terreno. El estudio del conexionado entre neuronas en circuitos locales, por ejemplo en la corteza visual primaria (área 17), tiene como objetivo primordial la comprensión del tipo de procesamiento que realiza el cerebro sobre la base de partida de la información sensorial. La formación de los *campos receptores*, la *dominancia ocular*, la *selectividad a la orientación* o a *patrones en movimiento*, *percepción de contraste y brillo*, la *respuesta a patrones visuales más complejos*, etc., son las características fundamentales que estudian los neurofisiólogos en la actualidad.

Las redes neuronales tuvieron su origen en el conocimiento del funcionamiento básico de las neuronas biológicas y la fisiología de sus sinapsis. Pueden distinguirse, por un lado, las redes neuronales diseñadas para encontrar la solución a problemas concretos de optimización, clasificación, predicción, etc., en muy diversas áreas de aplicación, denominadas redes neuronales artificiales; y, por otro lado, las redes neuronales para modelado de redes de neuronas biológicas. En el primer caso, los modelos de neuronas (continuos o discretos) son mucho más simples y parten de un nivel de abstracción más alto que en el segundo caso. Así, en el modelado de redes neuronales biológicas, se pretende, habitualmente, inferir

conocimiento acerca del comportamiento real de neuronas o arquitecturas de redes neuronales con gran nivel de detalle en el modelado, a una escala más baja y utilizando comúnmente modelos discretos de neuronas. A medio camino entre uno y otro caso, pueden implementarse y diseñarse redes de neuronas artificiales que abstraigan el comportamiento fundamental de las redes y neuronas biológicas, en una escala intermedia, y pretendiendo reproducir en la red artificial las propiedades computacionales de procesamiento de información encontradas o especuladas en la Biología.

A partir del estudio en profundidad de la literatura y de los datos actuales obtenidos de los estudios anatómicos y neurofisiológicos de la vía visual en mamíferos, el proyecto PIVEC (*Processing In Visual Elementary Circuits*)¹ trata del diseño e implementación de un modelo computacional de neurona de integración y disparo que simule el funcionamiento de la neurona biológica. Se hace hincapié en las propiedades electrofisiológicas de la membrana neuronal. Para considerar características funcionales tales como los retrasos presinápticos, desplazamiento, tiempo de subida, etc., es necesario reducir la escala de las simulaciones a nivel de décimas de milisegundo. A partir del modelo de neurona se ha construido un simulador a medida que permita la definición de redes de topología que permitan la construcción de arquitecturas de red y la simulación de sus dinámicas de actividad. Sobre esas redes se han diseñado experimentos destinados a reproducir el comportamiento de circuitos de neuronas observados en el análisis de la vía visual en mamíferos, analizando las propiedades computacionales de dichos circuitos y su utilidad en el procesamiento de la información visual. Utilizando técnicas de optimización, del tipo de los algoritmos genéticos o las propias redes neuronales artificiales, la fase actual del proyecto trata sobre el diseño de experimentos que conduzcan a la localización de circuitos elementales visuales integrados en redes de gran complejidad topológica. Estos circuitos supondrán soluciones similares o alternativas a la circuitería encontrada en el cerebro en la experimentación con animales en laboratorio. El alto grado de complejidad computacional que requiere la utilización de estas técnicas en el ámbito del procesamiento en circuitos neuronales basados en neuronas de integración y disparo, conlleva a la utilización de la programación y el procesamiento paralelos en máquinas de elevada potencia de cómputo. Ello implica la necesidad de un diseño eficiente de los algoritmos y de una planificación eficaz de las tareas paralelizables.

¹ Este proyecto está siendo desarrollado por el *Grupo de Estudios en Biomimética* de la *Universidad de Málaga (UMA)* y el departamento de *Ciencias Biológicas* de la *Universidad Pública de Nueva York (SUNY)*.

1.1.3 Redes neuronales artificiales

Los relevantes descubrimientos llevados a cabo por la neurofisiología, comentados en los apartados anteriores, han llevado a plantear la posibilidad diseñar y construir una máquina que sea capaz de desarrollar facultades cognitivas para el tratamiento de la información, tales como el habla, el razonamiento o la elaboración de conceptos, es uno de los objetivos principales de la investigación científica. Los sistemas actuales de computación han demostrado su incapacidad en estos campos. Pese a ser capaces de realizar un alto número de operaciones matemáticas, de forma muy rápida, no son eficientes para realizar tareas como el procesamiento del habla o el análisis de escenas.

Las redes de neuronas artificiales (RNA), inspiradas en la estructuras del sistema nervioso de los animales, exhiben ciertas propiedades parecidas a las del cerebro, tales como: asociación, reconocimiento de patrones, generalización, clasificación, y sobre todo, adaptación a cambios de entorno.

Ya en los años 40, en los principios de la computación moderna, investigadores como Norbert Wiener y Jonh von Neumann, sugerían en algunos de sus trabajos que la investigación en el diseño de ordenadores inspirados en el cerebro podría ser interesante (Heims, 1980).

Fue el trabajo de Warren McCulloch y Walter Pitts, publicado bajo el título "*A logical calculus of the ideas immanent in nervous activity*" (McCulloch & Pitts, 1943), el que señala el comienzo de la neurocomputación. En dicho trabajo se muestra que incluso las formas más simples de redes neuronales pueden computar, en principio, funciones aritméticas y lógicas. Sus neuronas binarias sólo podían estar en uno de dos posibles estados. Cada unidad de computación o neurona podía recibir entradas de sinapsis inhibitoras y/o sinapsis excitadoras, integrándolas en intervalos de tiempo discretos. Si la suma obtenida superaba un determinado valor de umbral fijado previamente, la neurona se consideraba activa y producía una señal de salida, en caso contrario la neurona se consideraba inactiva. El modelo propuesto por McCulloch y Pitts sería comparado, en 1974, a los sistemas de *momentos magnéticos elementales* o *spin* por William Little (Little, 1974), relacionando la forma de almacenamiento de la información de estas redes a la capacidad de polarización de los materiales magnéticos en niveles elementales.

En 1949, Donald Hebb, psicólogo de la Universidad de McGill, escribió el libro titulado "*The Organization of Behavior*" (Hebb, 1949), donde diseñó la primera regla de aprendizaje

para las redes de neuronas artificiales, que permitía la optimización de los conexiones mediante la reiterada modificación de los pesos sinápticos.

A principios de la década de los sesenta, el psicólogo Frank Rosenblatt introdujo y desarrolló una amplia gama de redes neuronales artificiales basadas en un modelo no lineal de memoria llamadas *Perceptrón* (Rosenblatt, 1962) que en principio crearon bastante entusiasmo. La regla de aprendizaje del Perceptrón utiliza un ajuste iterativo de pesos cuando la respuesta de la unidad es incorrecta, más potente que la regla de Hebb. Por esas fechas Bernard Widrow, ingeniero eléctrico, y su alumno Marcian Hoff (Widrow & Hoff, 1960) desarrollaron una regla de aprendizaje supervisado, muy parecida a la regla de aprendizaje del Perceptrón, donde los cambios en los pesos sinápticos eran proporcionales al error cometido por las unidades conectadas. Esta regla de aprendizaje fue la precursora de la regla de retropropagación para redes multicapa. Denominaron a su red neuronal ADALINE (*Adaptive Linear Element*). La similitud de los modelos de Rosenblatt y Widrow pone de manifiesto la interdisciplinariedad de las redes neuronales.

En 1969, el entusiasmo por las redes neuronales fue apagado por Marvin Minsky y Seymour Papert. En su libro "*Perceptrons*" mostraron las limitaciones del Perceptrón simple para aprender ciertas funciones lógicas no separables linealmente, como la XOR (disyunción exclusiva). A pesar de la poca popularidad de que gozaban las redes neuronales a partir de esa publicación, algunos investigadores siguieron desarrollándolas en trabajos relacionados con los sistemas de memoria asociativa, dispositivos que almacenan y permiten su recuperación basándose sólo en un conocimiento parcial de su contenido y no en el lugar de almacenamiento. Entre ellos se encontraban los trabajos de Teuvo Kohonen, de la *Universidad de Tecnología de Helsinki* (Kohonen, 1984), o Stephen Grossberg, director del *Center for Adaptive Systems* de la *Universidad de Boston* desarrolló, junto a Gail Carpenter, una teoría de redes autoorganizadas, llamada *teoría de resonancia adaptativa* (ART) (Carpenter & Grossberg, 1987; Grossberg, 1987).

No fue hasta la llegada de los trabajos John Hopfield, del *Instituto Tecnológico de California* y premio Nobel en Física, cuando se renovó el entusiasmo por las redes neuronales (Hopfield, 1982). Junto con David Tank, un investigador de *AT&T*, desarrollaron varias redes neuronales con pesos fijos y activaciones adaptativas que pueden resolver problemas de optimización combinatoria con restricciones, como el problema del viajante (Hopfield & Tank, 1985). La dinámica de estas redes está dominada por una tendencia hacia un número determinado de estados estables o *atractores*. Estos atractores se comportan como mínimos locales de una determinada función *energía*, y son generados por cada uno de los patrones usados en el entrenamiento de la red. Las redes de Hopfield exhiben la habilidad, tras su

entrenamiento, de completar y corregir patrones tomados como entrada con información incompleta o incorrecta. Estos trabajos, junto a todas las conferencias que dio Hopfield por todo el mundo, contribuyeron a despertar gran interés por la investigación en redes neuronales.

En los trabajos de Geoffrey Hinton y Terrence Sejnowsky (Hinton & Sejnowski, 1984) se mantiene la estructura básica de las redes de Hopfield (neuronas binarias y función de energía), pero la regla de aprendizaje es sustituida por la regla del *recocido simulado* (el sistema comienza en un estado de alta "temperatura" y progresivamente se enfría hasta alcanzar un estado de mínima temperatura). Con este nuevo modelo, denominado máquina de Boltzmann, se consigue alcanzar mínimos globales de la función de energía, no alcanzables por las redes de Hopfield.

La publicación en 1986 de los dos volúmenes titulados "*Parallel Distributed Processing*" (Rumelhart & McClelland, 1986), editados por David Rumelhart, de la *Universidad de California*, y James McClelland, de la *Universidad de Carnegie-Mellon*, produjo una gran explosión en ese campo. En ellos se recoge un método general efectivo de entrenamiento de una red neuronal multicapa (la regla Delta generalizada) que permite resolver, entre otros, el problema XOR. Este método es muy similar a otros desarrollados independientemente por varios investigadores, como es el caso del algoritmo propuesto por Bryson y Ho en la *teoría de control óptimo* (Bryson & Ho, 1969), o a los propuestos por Werbos (Werbos, 1974), Parker (Parker, 1982) y LeCun (LeCun, 1986).

Otros investigadores inspiraron sus trabajos en los estudios realizados en otras disciplinas, como en el caso del estudio del sistema de visión de los mamíferos realizado por Hubel y Weisel (Hubel & Weisel, 1962), cuyo modelo fue tomado por Fukushima como base de la red neuronal multicapa con fuertes propiedades de auto-organización que denominó *Neocognitrón* (Fukushima, 1988).

En 1987 se celebró en San Diego la primera conferencia abierta sobre redes neuronales (*IEEE International Conference on Neural Networks*), con más de 1700 participantes, y se formó la *International Neural Network Society* (INNS). En 1988 nace la revista *Neural Networks*. Le sigue la revista *Neural Computation* en 1989 y la *IEEE Transaction on Neural Networks* en 1990. Posteriormente han ido apareciendo otras muchas y se han creado Institutos de Investigación y programas de formación en Neurocomputación.

1.1.4 Modelado de sistemas neuronales biológicos

Como se ha comentado en el apartado 1.1.1, las neuronas se comunican unas con otras mediante la generación de disparos (potenciales de acción) separados por intervalos de tiempo variables. Estas series de disparos constituyen el código neuronal que el cerebro usa para procesar la información visual. Aunque parezca sorprendente, todavía se desconoce en qué consiste exactamente ese código visual. Los disparos pueden ser generados según una gran variedad de patrones, a veces de manera casi aislada (es decir, con una frecuencia menor a 1 Hz) y, otras veces, mediante cortos intervalos entre disparos. No está claro aún si el cerebro procesa dicha información simplemente “contando” el número de disparos o haciendo “lecturas” más sofisticadas de los patrones temporales en una especie de “código morse”.

El problema del “código morse” es que requiere una gran precisión en cuanto a la generación de disparos en el tiempo. Sin embargo, cada patrón de disparos significativo necesitará ser reproducido de manera fiable por cada presentación de un evento que se ha codificado en ese patrón. Este problema es particularmente importante en el sistema visual, donde para diferentes presentaciones de un mismo estímulo se ha demostrado que los patrones de disparos que se generan varían considerablemente en latencia y en intervalos entre disparos. Son muchos los argumentos que se han presentado a favor y en contra del papel de la precisión de los disparos en el procesamiento de información visual. Los defensores de la necesidad de una alta precisión en las temporizaciones de los disparos, argumentan que las neuronas pueden disparar de manera sincronizada entre ellas y pueden generar disparos con una precisión de 1 milisegundo en respuesta a ciertos estímulos visuales. Los escépticos, argumentan que las neuronas son, esencialmente, ruido y que el mismo estímulo puede evocar diferentes patrones de disparo de las neuronas, de manera que esta variabilidad se incrementa conforme se avanza en las etapas de procesamiento, desde la retina hasta las capas últimas de la corteza visual.

Para comprender el papel de la precisión de los disparos en el procesamiento visual, es necesario comprender los factores que pueden modificar esa precisión. Es difícil aislar la contribución de cada factor de manera experimental, en el laboratorio y sobre sujetos animales. Sin embargo, puede conseguirse una aproximación muy exacta mediante estudios de modelado computacional. Estos estudios sacan provecho de la creciente cantidad de datos procedentes de la experimentación neurofisiológica y de la anatomía de las conexiones, fundamentalmente monosinápticas. Las conexiones monosinápticas (aquellas que tienen lugar entre dos neuronas, siendo una presináptica y la otra postsináptica) generan disparos precisamente correlacionados entre la neurona de partida (donde se introduce una

estimulación externa) y la neurona diana. Esta correlación determina, de manera amplia, la precisión de sincronía generada entre varias neuronas dianas a partir de un aferente común.

Una pieza clave de todo paradigma de redes neuronales es el modelo matemático de neurona, ya que condiciona fuertemente el funcionamiento de las redes. Se han formulado muchos modelos de neurona, desde los más radicalmente biológicos (MacGregor, 1987; Hodking & Huxley, 1952), altamente parametrizados para representar a las sustancias químicas que afectan a la computación neuronal, hasta la suma elemental ponderada empleada en los modelos conexionistas (McCulloch & Pitts, 1943), pasando por numerosos elementos de proceso inspirados en fenómenos físicos (Hopfield, 1982).

Son numerosos los modelos computacionales propuestos para el modelado de redes neuronales de inspiración biológica. Destaca especialmente los modelos de integración y disparo (Stein, 1967), que muestran los aspectos principales de la excitabilidad neuronal (integración subumbral y generación de disparos por encima de dicho umbral), y los modelos creados a partir de él donde se consideran además aspectos del comportamiento neuronal tales como: el período refractario, la hiperpolarización de la membrana después de cada disparo, el potencial excitador postsináptico (*EPSP: Excitatory PostSynaptic Potential*) y los tiempos de transmisión de los impulsos en las sinapsis neuronales, etc.

El modelo de integración y disparo se caracteriza por considerar a la neurona como un dispositivo biestado. Por defecto, la neurona adopta un estado de reposo en el que realiza la integración espacial y temporal de los impulsos nerviosos que llegan a su árbol dendrítico. Las despolarizaciones recibidas pueden llevar al potencial de membrana a superar el nivel de umbral, en cuyo caso se genera por un breve lapso de tiempo el estado de actividad.

En la siguiente sección se muestra un modelo simplificado de neurona de integración y disparo que trata de emular los principales aspectos de la dinámica del potencial de membrana de las neuronas reales.

1.1.5 Modelo neuronal de integración y disparo

El modelo neuronal que será empleado en el presente proyecto es un modelo de *neurona de integración y disparo*. Dicha denominación se debe al mecanismo usado por la neurona para producir trenes de disparos debido a su actividad.

Los estudios realizados sobre el funcionamiento de las neuronas biológicas han revelado que los impulsos nerviosos procedentes de las neuronas aferentes generan potenciales eléctricos locales en las uniones sinápticas, y éstos son transmitidos hacia el soma o cuerpo celular, que los suma al potencial de membrana existente.

En caso de que el potencial total alcanzado exceda un determinado umbral propio de cada neurona, se genera un potencial de acción que se transmite a través del axón hacia las neuronas eferentes.

Si el potencial de membrana no alcanza el valor umbral, éste decae hasta el *potencial de reposo* en ausencia de estimulación externa. Esto hace que la neurona se comporte como un condensador que se descarga con una constante de tiempo. Este parámetro es fundamental en la computación neuronal, ya que determina la *integración temporal* de los impulsos. La *caída del potencial de la membrana* se define como el tiempo en que el potencial de membrana desciende de un valor V a V/e milivoltios, y se encuentra en el orden de los milisegundos. De este modo, son dos las variables que determinan el valor instantáneo del potencial de membrana: las polarizaciones recientes y la tendencia hacia el potencial de reposo.

Por otro lado, las neuronas presentan una actividad espontánea que no es debida a la estimulación producida por la actividad de otras neuronas de su entrono sino a un factor de naturaleza estocástica, denominado *ruido*, y cuyo papel es fundamental en la computación neuronal.

El modelo que se adapta en este proyecto es el propuesto por F.J. Vico (Vico, 1995) y está definido por tres ecuaciones que evalúan el potencial de membrana (V) y la variable de disparo de la neurona (X) en función de los tres parámetros expuestos (*umbral de disparo*, *potencial de reposo*, *caída del potencial de la membrana*). Este modelo ha sido escogido porque recoge los principios mínimos de la actividad integradora sub-umbral de la neurona y su comportamiento biestable (disparo o ausencia del mismo) de manera suficientemente simple para los propósitos de este proyecto. El modelo queda definido por las ecuaciones 1.1, 1.2 y 1.3,

$$A_i(t) = r_i(t) + \sum_j X_j(t-1)W_{ij} \quad (1.1)$$

$$V_i(t) = \begin{cases} V_i(t-1) + A_i(t) - \gamma[V_i(t-1) - \Phi] & \text{si } X_i(t-1) = 0 \\ \Phi & \text{si } X_i(t-1) = 1 \end{cases} \quad (1.2)$$

$$X_i(t) = \begin{cases} 0 & \text{si } V_i(t) < \Theta \\ 1 & \text{si } V_i(t) \geq \Theta \end{cases} \quad (1.3)$$

donde $A_i(t)$ es el potencial de membrana de la neurona i en el instante t , W_{ij} es el peso de la conexión sináptica entre las neuronas i y j , $r_i(t)$ es una señal ruidosa uniforme que se suma a la actividad neuronal, γ es la constante que regula la caída de potencial de membrana, Φ es el potencial de reposo y Θ es el umbral de disparo. X toma los valores 0 ó 1 para representar los estados de reposo o disparo de la neurona.

La figura 1.2 muestra la evolución del potencial de membrana de una neurona a lo largo de 400 pasos de simulación (cada paso de simulación equivale a un milisegundo). En ella se puede observar como se produce un disparo cada vez que el potencial supera el valor del umbral de disparo (-40 ms). Tras un disparo, el potencial cae hasta el potencial de reposo (-70 ms). Entre cada par de disparos consecutivos el potencial de membrana queda determinado por las polarizaciones y despolarizaciones recientes, el ruido y la tendencia natural hacia el potencial de reposo.

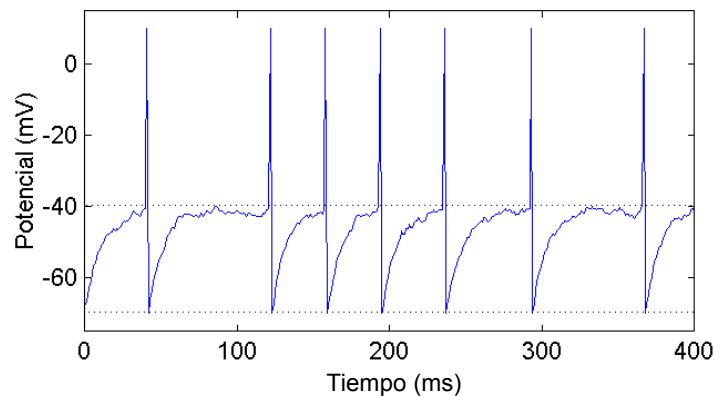


Fig. 1.2. Potencial de membrana de una neurona modelada según las ecuaciones 1.1, 1.2 y 1.3 a lo largo de 400 milisegundos.

1.1.6 Una herramienta de análisis: el correlograma cruzado

Como ya se comentó anteriormente, el estudio de modelos de neuronas inspirados en los biológicos se basan en la creciente cantidad de datos procedentes de la experimentación neurofisiológica y de la anatomía de las conexiones.

Una de las herramientas más usadas para el análisis de la correlación entre los disparos de dos neuronas (ya sean en conexiones monosinápticas, o entre pares de neuronas de un circuito complejo) es el *correlograma cruzado*.

En neurofisiología, el correlograma es un histograma que indica la relación de disparos producidos por una neurona (*objetivo*) en relación con otra (*referencia*), de manera que para cada disparo de la neurona de referencia (*N1*) se contabiliza el número de veces que dispara la neurona objetivo (*N2*) en diferentes instantes anteriores, posteriores y simultáneos en una ventana temporal.

A continuación se detalla el modo en que se elabora un correlograma a partir de los trenes de disparos de un par de neuronas *referencia-objetivo*, a las que se llamarán en adelante *N1* (referencia) y *N2* (objetivo), y se ilustrará con un ejemplo simplificado.

Para construir el correlograma cruzado, se debe, en primer lugar, almacenar de forma simultánea los trenes de disparos del par de neuronas *referencia-objetivo* a estudiar a lo largo de un determinado número de pasos de simulación. En la figura 1.3 se muestra un ejemplo de dos vectores de ceros y unos que equivalen a trenes de disparos de las neuronas *N1* y *N2* (donde 1 significa un disparo y 0 su ausencia).

→

N1 = 10000010001000000100001000000100001000001000001000001000001000010...

N2 = 0000010010001000000100001000101000010000100000100100100000100000101001...

Fig. 1.3. Ejemplo de trenes de disparos de dos neuronas a lo largo de 70 pasos de simulación.

Se define un histograma de tamaño $2\tau + 1$ (ver figura 4.b), donde τ es el tamaño de la ventana temporal del histograma, con una posición destacada $\tau + 1$ (centro de la misma). Los dos trenes de disparos deben disponerse de forma que queden temporalmente alineados. La ventana será desplazará sobre el tren de disparos de *N2*, ocupando su centro únicamente

aquellas posiciones donde se den disparos de la neurona de referencia (ver figura 4.a). Las posiciones de la ventana ocupadas por disparos de N_2 , son contabilizadas y sumadas a la posición correspondiente del correlograma (ver figura 4.b). Al final del proceso es obtenido el número de disparos producidos por N_2 a distancias comprendidas entre $-\tau$ y τ instantes de tiempo de disparos de N_1 . El proceso descrito está formalizado en la ecuación 1.4,

$$C_k = \sum_{t=\tau+1}^M N_1(t) N_2[t - (\tau + 1 - k)] \quad (1.4)$$

donde C_k es el valor del correlograma en la posición k y M es el número de pasos de simulación. Cada una de las $2\tau + 1$ posiciones del correlograma recibe el nombre de *barra*.

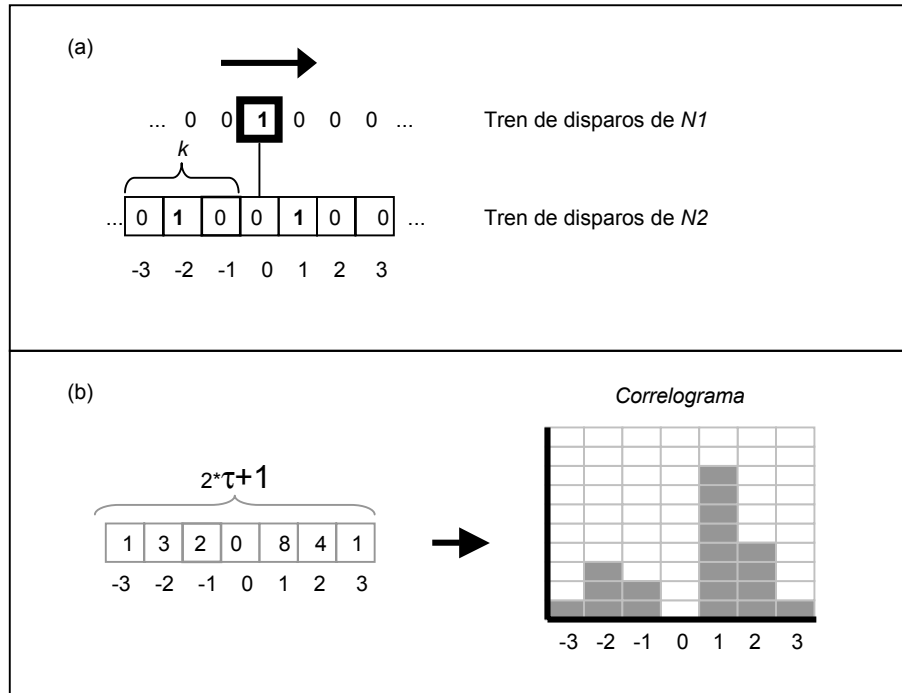


Fig. 1.4. (a) Proceso de contabilización a lo largo del tren de disparos. La ventana se centra en cada posición ocupada por un disparo de N_1 . Las posiciones de la ventana ocupadas por disparos de N_2 son contabilizadas. Cada posición relativa de la ventana representa un desplazamiento temporal con respecto al centro, de esta forma un disparo de N_2 que ocupe la posición -2 indicará que se produjo dos instantes de tiempo antes que el disparo de N_1 que ocupa el centro de la ventana. (b) Gráfica del Correlograma obtenido a partir de un valor de tamaño de ventana de 3 sobre los trenes de disparos de la figura 1.3.

A partir del correlograma se puede estudiar una posible relación entre los disparos de $N1$ y $N2$. Un número relativamente alto de disparos en una barra indica una relación entre los disparos de $N1$ y $N2$ con un desplazamiento temporal equivalente al valor de la barra. Como puede verse en la figura 1.4, el correlograma muestra el valor máximo de 8 en la barra 1, indicando que la mayor parte de los disparos de $N2$ se producen una unidad de tiempo después de disparos de $N1$, lo que podría deberse, con alta probabilidad, a la existencia de una conexión monosináptica, de 1ms de retraso entre $N1$ y $N2$.

Esto nos introduce de lleno en uno de los principales problemas que se plantean en este proyecto: la determinación del tipo de conexionado en un circuito neuronal que se puede inferir a partir de los correlogramas obtenidos a partir de los disparos de dos neuronas.

1.2 Técnicas evolutivas

En el apartado 1.1 se ha introducido los aspectos principales de la neurocomputación, las bases sobre las que se asientan, sus aplicaciones computacionales y los modelos que reproducen características biológicas. En este apartado se aborda la otra gran base sobre la que se asienta el presente trabajo: los algoritmos evolutivos, y más concretamente los *algoritmos genéticos*.

Los algoritmos evolutivos forman parte de los métodos de búsqueda, aquellos que dado el espacio de todas las posibles soluciones a un problema, y partiendo de un conjunto reducido de soluciones iniciales, son capaces de encontrar la mejor solución. Ciertos problemas son fáciles de abordar utilizando algoritmos recursivos o de tipo *voraz* (*Greedy*), sin embargo hay otros cuyo espacio de búsqueda crece de forma exponencial con el tamaño del problema y no pueden ser abordados de esta forma. Este tipo de problemas pueden ser abordados, sin embargo, por otro tipos de métodos como pueden ser los algoritmos evolutivos. Estos últimos realizan búsquedas en el espacio de soluciones partiendo de población iniciales de individuos (posibles soluciones), que evolucionan a lo largo un número determinado de generaciones siguiendo una serie de reglas inspiradas en las teorías evolutivas naturalistas, acercándose paulatinamente a la solución del problema.

1.2.1 Las teorías evolutivas

La teoría de la evolución (que no es tal teoría, sino una serie de hechos aceptados), fue propuesta por Charles Darwin 20 años después de su viaje por las islas Galápagos, en el libro *Sobre el Origen de las Especies por medio de la Selección Natural* (Darwin, 1859). Este libro fue bastante polémico en su tiempo, y en cualquier caso es una descripción incompleta de la evolución. La hipótesis de Darwin, presentada junto con Alfred R. Wallace (Wallace, 1858; Wallace, 1870), naturalista que llegó a las mismas conclusiones en trabajos independientes, afirmaba que la generación de nuevas especies viene producida por pequeños cambios heredables en los seres vivos y un proceso selección inducida por el entorno. Pero Darwin desconocía cual es la base de la herencia, ya que los trabajos de Mendel no serían publicados hasta unos años después. Pensaba que los rasgos de un ser vivo eran como un "fluido", y que los "fluidos" de los padres se mezclaban en la descendencia. El principal problema de esta hipótesis es que, de ser así, al cabo de cierto tiempo, los individuos de una misma población acabarían con iguales rasgos intermedios.

Fue Mendel quien descubrió que los caracteres se heredaban de forma discreta, y que se tomaban del padre o de la madre, dependiendo del carácter dominante o recesivo de aquellos (Mendel, 1865). A estos caracteres que podían tomar diferentes valores se les llamaron *genes*, y a los valores que podían tomar, *alelos*. En realidad, las teorías de Mendel, que trabajó en total aislamiento, se olvidaron y no se volvieron a redescubrir hasta principios del siglo XX. Además, hasta 1930 el biólogo evolucionista inglés Ronald A. Fisher no relacionó ambas teorías (Fisher, 1930), demostrando que los genes mendelianos eran los que proporcionaban el mecanismo necesario para la evolución.

Por la misma época, el biólogo alemán Walther Flemming (Flemming, 1882) describió los cromosomas, como ciertos filamentos en los que se agregaba la cromatina del núcleo celular durante la división; poco más adelante se descubrió que las células de cada especie viviente tenían un número fijo y característico de cromosomas.

Y no fue hasta los años 50 cuando Watson y Crick descubrieron que la base molecular de los genes está en el *ADN* (Watson & Crick, 1953), que forman parte de los cromosomas.

La macromolécula de *ADN* está compuesta por *bases púricas* y *pirimidínicas*, la *Adenina*, *Citosina*, *Guanina* y *Timina*. La combinación y la secuencia de estas bases forma el código genético, único para cada ser vivo, que codifica el conjunto de *proteínas* que forman parte de un ser vivo. Mientras que al código genético se le denomina *genotipo*, al cuerpo que

construyen esas proteínas, modificado por la presión ambiental, la historia vital, y otros mecanismos dentro del cromosoma, se le denomina *fenotipo*.

Todos estos elementos forman hoy en día la teoría del *neo-darwinismo*, que afirma que la historia de la mayoría de la vida está causada por una serie de procesos que actúan en y dentro de las poblaciones: reproducción, mutación, competición y selección. La evolución se puede definir entonces como *cambios en el conjunto genético de una población*.

1.2.2 Mecanismos de cambio en la evolución

Los mecanismos de cambio son las “herramientas” de que dispone el mundo natural para alterar el material genético residente en una población. Entender estos mecanismos es fundamental para su posterior aplicación a modelos de algoritmos evolutivos en la resolución de problemas de ingeniería. Los mecanismos de cambio alteran la diversidad genética en una población, y pueden ser divididos en dos grupos: los que la disminuyen y los que la aumentan.

Los principales mecanismos que disminuyen la diversidad genética son los siguientes:

- *Selección natural*: los individuos que tengan algún rasgo que los haga menos válidos para realizar su tarea de seres vivos, no llegarán a reproducirse, y, por tanto, su patrimonio genético desaparecerá de la población; algunos no llegarán ni siquiera a nacer. Esta selección sucede a muchos niveles: competición entre miembros de la especie (intraespecífica), competición entre diferentes especies, y competición *depredador-presa*, por ejemplo. También es importante la selección sexual, en la cual las hembras eligen el mejor individuo de su especie disponible para reproducirse.
- *Deriva génica*: el simple hecho de que un determinado material genético sea más común en la población que otro, causará que acabe imponiendo en una población aislada, lo cual a veces da lugar a fenómenos de especiación, por ejemplo, por el denominado *efecto fundador*.

Otros mecanismos aumentan la diversidad, y suceden generalmente en el ámbito molecular. Los más importantes son:

- *Mutación*: la mutación es una alteración del código genético, que puede suceder por múltiples razones. En muchos casos, las mutaciones que cambian un nucleótido por otro son letales, y los individuos ni siquiera llegan a desarrollarse, pero a veces se da lugar a la producción de una proteína que aumenta la supervivencia del individuo, y que, por tanto, es transmitida a la descendencia. Las mutaciones son totalmente aleatorias, y son el mecanismo básico de generación de variedad genética. A pesar de lo que se piensa habitualmente, la mayoría de las mutaciones ocurren de forma natural, aunque existen sustancias mutagénicas que aumentan su frecuencia.
- *Poliploidía*: mientras que las células normales poseen dos copias de cada cromosoma, y las células reproductivas una (*haploides*), puede suceder por accidente que alguna célula reproductiva tenga dos copias; si se logra combinar con otra célula *diploide* o *haploide* dará lugar a un ser vivo con varias copias de cada cromosoma. La mayoría de las veces, la poliploidía da lugar a individuos con algún defecto (por ejemplo, el tener 3 copias del cromosoma 21 da lugar al mongolismo), pero en algunos casos se crean individuos viables.
- *Recombinación*: cuando las dos células sexuales, o gametos, una masculina y otra femenina se combinan, los cromosomas de cada una también lo hacen, intercambiándose genes, que a partir de ese momento pertenecerán a un cromosoma diferente. A veces también se produce *traslocación* dentro de un cromosoma; una secuencia de código se elimina de un sitio y aparece en otro sitio del cromosoma, o en otro cromosoma.
- *Flujo genético*: o intercambio de material genético entre seres vivos de diferentes especies. Normalmente se produce a través de un *vector*, que suelen ser virus o bacterias; estas incorporan a su material genético genes procedentes de una especie a la que han infectado, y cuando infectan a un individuo de otra especie pueden transmitirle esos genes a los tejidos generativos de gametos.

En resumen, la selección natural actúa sobre el fenotipo y suele disminuir la diversidad, haciendo que sobrevivan sólo los individuos más aptos; los mecanismos que generan diversidad y que combinan características actúan habitualmente sobre el genotipo.

1.2.3 Historia de la computación evolutiva

Una vez descrito cuales son los mecanismos de la evolución, se introduce a continuación como muchas de esas ideas han pasado de ser conceptos naturalista a ser aplicados a la resolución de problemas de ingeniería.

Las primeras ideas, incluso antes del descubrimiento del ADN, vinieron de Von Neumann, uno de los mayores científicos de este siglo. von Neumann (von Neuman, 1966) afirmó que la vida debía de estar apoyada por un código que a la vez describiera cómo se puede construir un ser vivo, y tal que ese ser creado fuera capaz de autorreproducirse; por tanto, un autómata o máquina autorreproductiva tendría que ser capaz, aparte de contener las instrucciones para hacerlo, de copiar tales instrucciones a su descendencia.

Sin embargo, no fue hasta mediados de los años cincuenta, cuando el rompecabezas de la evolución se había prácticamente completado, cuando Box comenzó a pensar en imitarla para, en su caso, mejorar procesos industriales. La técnica de Box, denominada EVOP (*Evolutionary Operation*) (Box, 1957; Box & Draper, 1969), consistía en elegir una serie de variables que regían un proceso industrial. Sobre esas variables se creaban pequeñas variaciones que formaban un hipercubo, variando el valor de las variables una cantidad fija. Se probaba entonces con cada una de las esquinas del hipercubo durante un tiempo, y al final del período de pruebas, un comité humano decidía sobre la calidad del resultado. Es decir, se estaba aplicando mutación y selección a los valores de las variables, con el objeto de mejorar la calidad del proceso. Este procedimiento se aplicó con éxito a algunas industrias químicas.

Unos años después, Friedberg (Friedberg, 1958) y sus colaboradores pensaron en mejorar, usando técnicas evolutivas, la operación de un programa. Para ello diseñaron un código máquina de 14 bits. Cada programa, tenía 64 instrucciones. Un programa llamado Herman, ejecutaba los programas creados, y otro programa, el *teacher* o profesor, le mandaba a Herman ejecutar otros programas y ver si los programas ejecutados habían realizado su tarea o no. La tarea consistía en leer unas entradas, situadas en una posición de memoria, y debían depositar el resultado en otra posición de memoria, que era examinada al terminarse de ejecutar la última instrucción. Para hacer evolucionar los programas, Friedberg hizo que en cada posición de memoria hubiera dos alternativas; para cambiar un programa, alternaba las dos instrucciones (que eran una especie de alelos), o bien reemplazaba una de las dos instrucciones con una totalmente aleatoria. En realidad, lo que estaba haciendo es usar mutación para generar nuevos programas. Al parecer, no tuvo más éxito que si hubiera buscado aleatoriamente un programa que hiciera la misma tarea. El problema es que la

mutación sola, sin ayuda de la selección, hace que la búsqueda sea prácticamente una búsqueda aleatoria.

Simultáneamente, Bremmerman (Bremmerman, 1962) trató de usar la evolución para *entender los procesos de pensamiento creativo y aprendizaje*, y empezó a considerar la evolución como un proceso de aprendizaje. Para resolver un problema dado, codificaba las variables del problema en una cadena binaria de ceros y unos, y sometía la cadena a mutación, cambiando un bit de cada vez; de esta forma, estableció que la tasa ideal de mutación debía de ser tal que se cambiara un bit cada vez. Bremmerman trató de resolver problemas de minimización de funciones, aunque no está muy claro qué tipo de selección usó, si es que usó alguna, y el tamaño y tipo de la población. En todo caso, se llegaba a un punto, la "trampa de Bremmerman", en el cual la solución no mejoraba; en intentos sucesivos trató de añadir cruzamiento entre soluciones, pero tampoco obtuvo buenos resultados. Una vez más, el simple uso de operadores que creen diversidad no es suficiente para dirigir la búsqueda genética hacia la solución correcta.

El primer uso de procedimientos evolutivos en inteligencia artificial se debe a Reed, Tooms y Baricelli, que trataron de hacer evolucionar un tahúr que jugaba a un juego de cartas simplificado (Reed, Tooms, & Baricelli, 1967). Los intentos posteriores, realizados en los años 60, corresponden a los algoritmos evolutivos modernos, que se han seguido investigando hasta nuestros días. Algunos de ellos son simultáneos a los algoritmos genéticos, pero se desarrollaron sin conocimiento unos de otros. Uno de ellos, la programación evolutiva de Fogel (Fogel, Owens, & Walsh, 1966), se inició como un intento de usar la evolución para crear máquinas inteligentes, que pudieran prever su entorno y reaccionar adecuadamente a él. Para simular una máquina pensante, se utilizó un autómata celular. Un autómata celular es un conjunto de estados y reglas de transición entre ellos, de forma que, al recibir una entrada, cambia o no de estado y produce una salida.

Fogel trataba de hacer aprender a estos autómatas a encontrar regularidades en los símbolos que se le iban enviando. Como método de aprendizaje, usó un algoritmo evolutivo: una población de diferentes autómatas competía para hallar la mejor solución, es decir, predecir cual iba a ser el siguiente símbolo de la secuencia con un mínimo de errores; el 50% peor era eliminado en cada generación, y sustituidos por otros autómatas resultantes de una mutación de los existentes. De esta forma, se lograron hacer evolucionar autómatas que predecían algunos números primos (por ejemplo, uno, cuando se le daban los números más altos, respondía siempre que no era primo; la mayoría de los números mayores de 100 son no primos). En cualquier caso, estos primeros experimentos demostraron el potencial de la evolución como método de búsqueda de soluciones novedosas.

Más o menos a mediados de los años 60, Rechenberg (Rechenberg, 1965) y Schwefel (Schwefel, 1965) describieron las estrategias de evolución, métodos de optimización paramétricos, que trabajan sobre poblaciones de genotipos compuestos por números reales. Hay diversos tipos de estrategias de evolución, que se verán más adelante, pero en la más común, se crean nuevos individuos de la población añadiendo un vector mutación a los cromosomas existentes en la población; en cada generación, se elimina un porcentaje de la población, y los restantes generan la población total, mediante mutación y cruzamiento., siendo calculada adaptativamente la magnitud del vector mutación.

1.2.4 Algoritmos genéticos

Fue a principios de los 60, en la *Universidad de Michigan* en Ann Arbor, donde, dentro del grupo *Logic of Computers*, el profesor John Holland, que impartía un curso titulado *teoría de sistemas adaptativos*, comenzó a desarrollar sus ideas sobre un nuevo tipo de algoritmos basados en las teorías genéticas y evolutivas que acabarían convirtiéndose en los algoritmos genéticos (Holland, 1975). Del libro escrito por R.A. Fisher, titulado *la teoría genética de la selección natural* (Fisher, 1930), aprendió que la evolución era una forma de adaptación más potente que el simple aprendizaje, y tomó la decisión de aplicar estas ideas para desarrollar programas bien adaptados para un fin determinado.

Cuando Holland se enfrentó a los algoritmos genéticos, los objetivos de su investigación fueron principalmente dos: imitar los procesos adaptativos de los sistemas naturales, y diseñar sistemas artificiales (normalmente programas) que retengan los mecanismos importantes de los sistemas naturales.

Unos 15 años después, David Goldberg, ingeniero industrial que trabajaba en el diseño de redes de tuberías, se convirtió en su estudiante, y fue uno de los primeros que trató de aplicar los algoritmos genéticos a problemas industriales. Aunque Holland trató de disuadirle, porque pensaba que el problema era excesivamente complicado como para aplicarle algoritmos genéticos, Goldberg consiguió sus objetivos aplicando un algoritmo genético que el mismo escribió (Goldberg, 1989). Estas y otras aplicaciones creadas por estudiantes de Holland convirtieron a los algoritmos genéticos en un campo con base suficiente aceptado para celebrar la primera conferencia en 1985, ICGA'85. Tal conferencia se sigue celebrando bianualmente.

En los siguientes apartados se repasan los principales conceptos que forman parte de los algoritmos genéticos, así como las pautas a seguir para su uso en la búsqueda de soluciones óptimas de un problema dado.

Esquema de funcionamiento de un algoritmo genético

Los algoritmos genéticos son métodos sistemáticos para la resolución de problemas de búsqueda y optimización que aplican a estos los mismos métodos de la evolución biológica: selección basada en la población, reproducción sexual y mutación. En (Bäck, 1996) se caracterizan los algoritmos genéticos como aquellos métodos de búsqueda con las siguientes propiedades:

- i. Uso del aprendizaje colectivo (o adaptación colectiva) de una población de individuos, donde cada individuo representa un punto en el espacio de posibles soluciones de un problema dado. Cada individuo es representado mediante una expresión denominada *genoma*. Los elementos de información mínimos que componen el genoma reciben el nombre de *genes* y sus posibles valores *alelos*.
- ii. La descendencia de individuos es generada mediante un conjunto de procesos aleatorios denominados operadores genéticos, como son el operador de mutación o de cruzamiento. El operador mutación corresponde a un proceso de replicación de individuos en el que se produce un error, una pequeña variación en el genoma con una frecuencia muy baja. El operador cruzamiento consiste en la recombinación de los genomas de dos individuos, mediante el intercambio de la información contenida en éstos.
- iii. Mediante la evaluación de los individuos de la población, una medida de bondad o valor de *fitness* es asignada a cada individuo. El requisito mínimo exigido para esta medida es que permita la comparación entre dos individuos para decidir cual es mejor y cual peor. En función del valor de *fitness*, un proceso de selección favorecerá a los individuos con mejor puntuación en detrimento de los peor puntuados.

Para la aplicación de un algoritmo genético en un problema de optimización es necesario parametrizar éste y codificar su conjunto de variables, en los llamados genomas. Cada uno de estos genomas codifica a un único individuo (una posible solución del problema). La carrera por

obtener el mejor individuo (la mejor solución) comienza habitualmente con la generación de una población inicial de forma aleatoria, con un número de individuos suficiente para garantizar la variedad genética, a la que se denomina generación 0. En cada generación son evaluados los individuos de la población, y a partir de la puntuación obtenida cada uno de ellos compete para legar parte de su material genético a individuos de la próxima generación. Esto produce, a lo largo de sucesivas generaciones, una población cada vez más "adaptada", cuyos individuos poseen una mayor puntuación y que se traducen en mejores soluciones del problema. La condición de parada se puede establecer al alcanzar un número máximo de generaciones o por la obtención de un conjunto de individuos en alguna de ellas con una puntuación suficientemente buena. Las soluciones obtenidas por un algoritmo genético no son siempre óptimas de la función de evaluación, sino un conjunto de subóptimos que representan soluciones suficientemente buenas.

La aplicación de un algoritmo genético es independiente del problema a resolver, lo cual lo hace un algoritmo robusto (por ser usado para cualquier problema), pero a la vez débil (pues no está especializado en ninguno).

En los siguientes apartados se detallará cada uno de los pasos que deben seguirse para la implementación de un algoritmo genético en la resolución de un problema como son: la elección del modelo de codificación de las variables, la función de evaluación, el modelo de selección de individuos y el conjunto de operadores aplicados para generar diversidad.

Codificación de las variables

Los algoritmos genéticos tradicionales codifican los rasgos destacados de los individuos con un vector de números de una forma explícita. Las primeras aplicaciones de algoritmos genéticos realizadas (Holland, 1975) hicieron uso de una cadena de bits para representar los individuos de una población. La cadena de bits fue una representación general que pudo ser aplicada a una diversidad de problemas y también permitió investigaciones sobre los operadores en algoritmos genéticos. Mientras que las cadenas de bits son adecuadas para muchas aplicaciones, no siempre es el modo más natural o simple de representación. Para problemas de optimización de parámetros donde la precisión es a menudo crítica, los números reales han sido usados de forma efectiva. En otras aplicaciones, como problemas de optimización combinatorial, los números enteros o naturales son a menudo más apropiados. Goldberg

también exploró un esquema de codificación alternativo de cadenas de bits denominado *Messy Genetic Algorithm (mGA)* (Goldberg, Korb, & Deb, 1990). En él eran usados individuos de longitud variable y fue aplicado a una clase de problemas conocidos como "problemas engañosos" (*deceptive problems*). Otra variación inspirada por procesos biológicos e introducida por Dasgupta y McGregor es conocida como el *Structured Genetic Algorithm (sGA)* (Dasgupta & McGregor, 1994). Esta variación usa el principio de activación de genes en estructuras multinivel, ofreciendo muchas ventajas en problemas de búsqueda y optimización. En todos los modelos comentados, cada segmento de esos genomas está relacionado con una característica determinada del individuo, siendo considerados, por ello, modelos de codificación explícita. La diferencia en estos modelos entre el genotipo y fenotipo es prácticamente inexistente.

Evaluación y selección

En cada generación es necesario evaluar los individuos con objeto de obtener su grado de bondad en la solución del problema que se trata de resolver. Para ello se evalúa su genotipo o, si es necesario, se genera su fenotipo a partir de la descodificación de su genoma y se aplica una función de evaluación sobre dicho fenotipo. La puntuación obtenida, a la que se denomina *fitness* del individuo, servirá para establecer órdenes en la selección de individuos para la aplicación de operadores genéticos. En el caso de problemas donde se requiere optimizar varios objetivos, el *fitness* se compone de un conjunto de valores que representan la bondad de cada individuo en diversos aspectos. No obstante, en estos casos se puede recurrir a la suma ponderada de los valores independientes o a la aplicación de alguna otra función que genere un valor único que permita la comparación para responder a la pregunta de cual es el mejor de dos individuos dados.

Los métodos de selección se prestan a numerosas adaptaciones y combinaciones entre ellos. A continuación se esbozan algunos de los métodos de selección más usuales:

- *Proporcional (Roulette)* (Grefenstette, 1997a). Este método usado por primera vez por Holland (Holland, 1975) ha sido ampliamente usado en numerosas aplicaciones de algoritmos genéticos. Consiste en la selección de individuos proporcional al valor de sus *fitness*. Esta selección se realiza para crear la siguiente generación de individuos mediante el uso de operadores genéticos en copias de los individuos seleccionados. Un individuo

con alta puntuación puede ser seleccionado en más de una ocasión para crear "descendencia", aportando en mayor medida su material genético a la siguiente generación que un individuo con una puntuación inferior. Si la proporción hace que un individuo domine la población, es recomendable aplicar una operación de escalado para reducir su presencia. De igual forma debe asegurarse que individuos con puntuaciones malas sean sistemáticamente ignorados.

- *Basada en el rango (Rank-based)* (Grefenstette, 1997b). En la *selección proporcional*, como se ha visto, un "super" individuo puede copar la totalidad de las selecciones si su *fitness* es muy alto con respecto al resto, a no ser que se impongan límites artificiales. En la *selección basada en el rango* no se tiene en cuenta directamente el valor del *fitness* de un individuo para asignar la probabilidad de selección, sino el puesto que ocupa en el ranking de puntuaciones. De esta forma el mejor individuo de cualquier generación siempre posee la misma probabilidad independientemente de la diferencia en el *fitness* con respecto al resto.
- *Torneo (Tournament)* (Blickle, 1997). En este método la selección de individuos se realiza tomando un subconjunto aleatorio de individuos de la población (considerados participantes en el "torneo") y escogiendo, de ellos, el que posea una puntuación más alta. La elección aleatoria de los individuos que compiten asegura que todos los individuos disponen de la misma probabilidad de participar en un torneo. El problema que se plantea en la selección proporcional queda excluido en este tipo de selección, ya que la selección del mejor individuo de la población, para contribuir con su material genético a la siguiente generación, está limitada por el número de veces que es incluido en un torneo.
- *Isolating by distance (Mühlenbein, 1991)*. Está basado en el concepto de "vecindario", y requiere una población distribuida espacialmente sobre una rejilla 2-D. Al igual que en el modelo *torneo*, se selecciona al mejor individuo de un subconjunto, pero a diferencia del anterior, el subconjunto lo componen los individuos localizados a lo largo de un camino de longitud k escogido de forma aleatoria partiendo de una casilla inicial determinada. Aplicado junto con el operador de cruzamiento produce emparejamientos de individuos "relativamente buenos" (cada uno debe ser el mejor de un camino aleatorio) y que se encuentran "relativamente cercanos" (una distancia máxima de $2k$).
- *Crowding* (Mahfoud, 1992). Este modelo, inspirado en las técnicas propuestas por De Jong (De Jong, 1975), mantiene la diversidad de la población imponiendo que un individuo insertado en la población sustituya únicamente a un individuo similar. El concepto de *similitud*, puede asociarse a la cercanía fenotípica o genotípica. Si esto no siempre es fácil medirlo para dos individuos cualesquiera de la población, sí puede considerarse que si un

individuo es generado a partir de otros mediante la aplicación de un operador genético, el primero guarda un alto nivel de similitud con sus predecesores. De esta forma puede establecerse un "torneo" entre pares padre-hijo, siendo el que consiga mejor puntuación el que pasa a la siguiente generación. De esta forma un individuo sólo es sustituido por otro similar (hijo) que mejore su puntuación, por lo que la diversidad es conservada.

El modelo de selección de individuos determinará el comportamiento de la población a lo largo de sucesivas generaciones. En un modelo *elitista* una generación no puede tener puntuaciones máximas inferiores a las obtenidas por los mejores individuos de generaciones anteriores, en el peor de los casos, las puntuaciones deben ser las mismas. La formalización de dicho concepto queda reflejada en la ecuación 1.5,

$$\text{Elitismo} \Rightarrow \max\{\text{fitness}(x) \mid x \in \text{Población}^{(i+1)}\} \geq \max\{\text{fitness}(x) \mid x \in \text{Población}^{(i)}\} \quad (1.5)$$

donde x es un individuo en la población e i es el número de la generación. Una gran desventaja del elitismo es que en casos de no conseguir, durante un determinado número de generaciones, individuos que superen a los mejores de la población, los mejor puntuados habrán tenido la oportunidad de generar una gran descendencia reduciendo la variedad genética del conjunto población a favor de variaciones de los genomas de los individuos mejor valorados, produciendo un bloqueo en la evolución de la población.

En problemas donde no se requiere localizar un sólo punto óptimo, sino un conjunto de ellos, conservar la variedad genética y evitar la convergencia de toda la población a una única solución es un requisito indispensable. Estos problemas requieren, por ello, un tratamiento complementario enfocado a la obtención de diversas soluciones. Dotar a la población de cierta estructura que asegure la diversidad parece ser el camino a seguir. En este sentido son varios los métodos que se proponen en la literatura actual:

- *Métodos de Nicho (Nicheing)* (Mahfoud, 1997). Se dividen principalmente en dos clases, *crowding* y *sharing*. El método de *crowding*, introducido en este apartado como método de selección, puede considerarse también como un método de estructuración de la población donde la búsqueda de soluciones se realiza mediante la escalada progresiva hacia óptimos o subóptimos de la población a lo largo de las generaciones. El método de *sharing*, introducido por Golberg y Richardson (Goldberg & Richardson, 1987), consiste en

fraccionar la población en porciones (nichos) atendiendo a la forma de la función *fitness*, de modo que se formen subconjuntos de individuos agrupados en torno a picos del *fitness* (posibles máximos locales de la función). El número de individuos permitidos en cada porción irá en función de la altura del pico, cuanto mayor sea la altura del pico mayor debe ser el número de individuos que lo exploren.

- *Métodos de Especiación (Speciation)* (Eldredge, 1989). En la naturaleza, el cruzamiento de individuos está limitado al conjunto de individuos de una misma especie (conjunto de individuos fenotípicamente similares), resultando inútiles los cruces entre individuos de especies distintas. Esta idea es adaptada en el método de especiación, donde la población es subdividida en especies entre las que está prohibida el cruce de individuos. Con ello se pretende que cada especie dentro de una población encuentre una solución óptima o subóptima diferente evolucionando de forma independiente.
- *Métodos de Islas (Island)* (Eldredge & Gould, 1972). Basado en las teorías de la evolución, este modelo consiste en la división de la población en número determinado de subpoblaciones (islas) que evolucionarán de forma aisladas. A diferencia del modelo de especiación, los individuos de una subpoblación deben ser genéticamente variados y el cruce de individuos entre subpoblaciones, aunque con una baja frecuencia, es posible. Esta propiedad se denomina *migración*, y consiste en que tras un determinado número de generaciones un individuo perteneciente a una isla puede "migrar" a una isla vecina. En muchos casos puede ser una buena forma de aportar diversidad genética a una población cuya evolución se ha estancado.

La aplicación de métodos de estructuración de la población puede ser especialmente útil en la implementación de algoritmos genéticos en máquinas paralelas de grano grueso en la resolución de problemas multimodales. La evolución independiente de estas subpoblaciones permite que cada una de ellas pueda ser albergada en un procesador distinto, con una mínima necesidad de comunicación entre ellos.

Operadores genéticos

La selección de individuos puede considerarse por si sola como un operador genético, pero para la producción de nuevos individuos para la siguiente generación se requiere la introducción de otros operadores aplicados sobre los seleccionados. De entre esos operadores, los principales son el *cruzamiento (crossover)* y la *mutación (mutation)*. Estos operadores

deben ser adaptados al modelo de codificación del genoma empleado, ya que no puede actuarse de la misma forma para un modelo implementado sobre cadenas de bits, que sobre cadenas de enteros, de números reales, o genomas codificados de forma implícita. A continuación se comentan las características de los principales operadores, algunas variaciones de éstos para modelos implementados sobre cadenas de bits, enteros o reales, así como otros operadores menos usados. Por último se comentará un problema muy usual en la aplicación de operadores genéticos, la *disrupción*.

Cruzamiento

Consiste en la generación de nuevos individuos a partir del intercambio de material genético entre dos individuos de la población. El cruzamiento es el principal operador genético, hasta el punto que se puede decir que no es tal algoritmo genético si no tiene el operador de cruzamiento.

La forma de generar el genoma de un nuevo individuo a partir de los genomas de padres, depende del modo en que está codificado el genoma. Es evidente que la forma de actuar cuando el genoma se encuentra codificado en forma de una cadena de bits, debe ser diferente a cuando lo hace codificado en una cadena de valores reales, y más aun con métodos de codificación más sofisticados, como los árboles gramaticales.

En modelos de codificación basados en cadenas, el intercambio genético se puede llevar a cabo de muchas formas, pero los siguientes son los más usados:

- *Crossover n-puntos*. Los dos genomas se "cortan" por n puntos, y el material genético situado entre ellos se intercambia. Lo más habitual es un cruzamiento de un punto o de dos puntos. Una alternativa al intercambio sistemático de valores entre ambos genomas, para modelos de codificación basados en vectores de enteros o flotantes, es la obtención de valores intermedios para los genes que se encuentran en la zona de combinación, mediante operaciones algebraicas.
- *Crossover uniforme*. Se escoge de forma aleatoria un conjunto de posiciones dentro de los genomas de los padres y se intercambian. Al igual que en el caso anterior, para genomas codificados en vectores de enteros o flotantes, puede ser indicado generar valores intermedios mediante la aplicación de operaciones algebraicas.

- *Crossover especializados*. En algunos problemas, aplicar aleatoriamente el cruzamiento da lugar a cromosomas que codifican soluciones inválidas. En estos casos se hará necesario que el operador de cruzamiento genere siempre individuos contenidos en el subconjunto de soluciones válidas.

Mutación

En la evolución biológica, una mutación es un suceso poco común (sucede, aproximadamente, una de cada mil replicaciones). En la mayoría de los casos las mutaciones son letales, pero en promedio contribuyen a la diversidad genética de la especie. En un algoritmo genético tendrán el mismo papel, y una frecuencia similar (es decir, muy baja).

El operador de mutación consiste en la replicación de un individuo con un determinado error, mediante una pequeña alteración en su genoma.

El operador de mutación es un mecanismo generador de diversidad, y, por tanto, una posible solución para salir de máximos (o mínimos) locales cuando el algoritmo genético está estancado, pero un exceso en su uso puede reducir el algoritmo genético a una búsqueda aleatoria. Es conveniente en esos casos usar otros mecanismos de generación de diversidad, como aumentar el tamaño de la población, o garantizar la aleatoriedad de la población inicial.

El operador de mutación, junto con el de cruzamiento y el método de selección de ruleta, constituyen un algoritmo genético simple, SGA, introducido por Goldberg (Goldberg, 1989).

Operadores de cambio de tamaño del cromosoma

Hasta ahora se han descrito genomas de longitud fija, donde se conoce de antemano el número de parámetros de un problema. Pero hay problemas en los que esto no sucede. En ellos es necesario incluir operadores que aumenten o reduzcan la complejidad de los genomas insertando o eliminando genes. La forma más habitual de añadir un gen es duplicar uno ya existente, el cual sufre mutación, y añadiéndose junto al primero. En este caso, los operadores del algoritmo genético simple (*selección*, *mutación*, *cruzamiento*) funcionarán de la forma

habitual, salvo que en el cruzamiento sólo se realizara recombinación en zonas comunes del genoma.

Otros operadores

Usados en menor medida, existe una gran variedad de operadores que pretenden explorar el espacio de soluciones de una forma ordenada, diseñados para problemas concretos o para aportar diversidad de una forma determinada en los momentos en que una población comienza a estancarse.

En los problemas donde haya que restringir las nuevas soluciones generadas por los operadores genéticos, debido a que no todas las soluciones generadas van a ser válidas, se requieren operadores que mantengan la estructura del problema.

Operadores más minoritarios nacen como variaciones de los principales. Éste es el caso del operador *transposición*, similar al cruzamiento, que consiste en el intercambio de dos genes o zonas dentro de un sólo genoma, sin afectar al resto del cromosoma. Otro del mismo estilo es el operador de *eliminación-reinserción*, en el que un gen o zona completa cambia de posición con respecto a los demás.

El problema de la Disrupción

El problema de la disrupción se produce cuando una pequeña variación en el genotipo produce una gran variación en el fenotipo. Esto produce unos efectos indeseables cuando se aplica un operador genético para la generación de un nuevo individuo, ya que se espera que el individuo generado se encuentre genotípica y fenotípicamente cercano a los individuos que lo originaron.

Cuando este problema aparece únicamente en la aplicación del operador de cruzamiento (*disrupción del cruzamiento*) suele deberse a que los individuos que son cruzados son genotípicamente demasiado diferentes entre sí, dando lugar a individuos muy deficientes. La solución a este tipo de disrupción puede venir dada por técnicas de especiación, donde el cruzamiento está limitado a individuos con estructuras genotípicas similares.

Si la disrupción aparece también al aplicar el operador de mutación, puede ser indicativo de que el método de codificación no es el apropiado. Esto se produce principalmente cuando se pretende codificar individuos complejos en genomas no estructurados definidos explícitamente. Una posible solución a esto es la de codificar de forma implícita los genomas de individuos complejos.

En el siguiente apartado se introduce los modelos de codificación implícita junto con un ejemplo de uso en la definición de individuos consistentes en distribuciones de puntos en un espacio bidimensional con estructura fractal.

Codificación Implícita

Los algoritmos genéticos tradicionales representan las soluciones codificando explícitamente los individuos de la población en un vector de números. Para problemas con pocas dimensiones, la estructura de los individuos no es excesivamente compleja y es posible mantener el tamaño de esos genes entre límites razonables, permitiendo la convergencia de los algoritmos. Cuando son considerados problemas de optimización con un alto número de dimensiones, la complejidad de las soluciones aumenta. Individuos complejos codificados mediante métodos de codificación explícitos generan genomas de tamaño excesivamente grande, degradando la eficacia de la búsqueda. Además, es necesario también aumentar el tamaño de la población (número de individuos) para permitir al método de búsqueda la suficiente variedad de individuos que haga converger de forma continua y rápida hacia la solución, añadiendo además el incremento del tamaño y complejidad de las estructuras de datos necesarias para almacenarlos.

La naturaleza hace uso de procesos de desarrollo para transformar genomas en organismos. Durante el desarrollo embrionario, las células sufren divisiones de acuerdo con la información genética. Esto permite que sistemas complejos puedan ser codificados con un código compacto. La información sobre todos los componentes de los individuos es almacenada en el genoma, pero la relación entre una región del genoma y una determinada característica del individuo no existe. Por el contrario, el genoma debe verse como un conjunto de instrucciones que operan sobre un conjunto de datos (almacenados ambos en el genoma). La interpretación del genoma permite, en un proceso denominado desarrollo embrionario o embriogénesis, la maduración de los organismos. Según ciertos estudios, un genoma expresa

su contenido como resultado de una interpretación iterativa de sus genes. Esto significa que los genes son expresados en diferentes momentos durante el desarrollo. Asumiendo este resultado, un modelo simple fue propuesto por Jacob (Jacob & Monod, 1961), conocido como el modelo de regulación *Operon*, el cual integra los principales principios de la genética bacteriana y estableció que los genes son activados y desactivados dependiendo de los productos de otros genes. En este trabajo se propuso que el genoma contiene la información concerniente al individuo, así como el modo de generarlo.

El desarrollo embrionario constituye el mecanismo con el que se genera el fenotipo (estructura del individuo) a partir del genotipo. Siguiendo esta línea de investigación, muchos trabajos han estudiado el modo de codificar estructuras complejas. F. Gruau propuso un mecanismo de desarrollo artificial para generar redes modulares (Gruau, 1995) basados en trabajos previos de Kitano (Kitano, 1990) donde se presentaba un modelo para diseñar modelos de redes neuronales usando algoritmos genéticos. En esta misma dirección, otros métodos de codificación pueden verse en los trabajos de Angeline (Angeline, Saunders, & Pollack, 1994) donde se construyen redes neuronales usando algoritmos evolutivos.

A continuación se presentará un modelo de codificación implícita propuesto por F.J. Veredas (Veredas, Vico, & Guillén, 2000), basado en los mecanismos descritos anteriormente. El modelo se basa en dos tipos de genes operacionalmente distintos: estructurales y reguladores, que implementan una interpretación recursiva del genoma.

En este modelo de caracterización, y en la base de los procesos naturales, una célula es representada como un conjunto de variables que constituye su estado interno. En el comienzo de desarrollo embrionario el individuo consiste en una única célula. Tras sucesivas divisiones (generaciones celulares) el número de células se incrementa, resultando el fenotipo del individuo al final del proceso. Los genes controlan la división celular.

El genoma está compuesto de dos clases de genes: estructurales y reguladores. Un gen estructural es una función matemática aplicada a un argumento específico —las variables de estado de la célula y algunos términos específicos de cada problema que son almacenados juntos con los genes en el genoma—, dando algunos valores de salida que contribuyen a nuevos estados celulares. Los genes reguladores controlan la expresión de estos genes estructurales. Cada gen regulador expresa exactamente un rango de genes estructurales y, además, varios genes reguladores pueden expresar rangos de genes estructurales sobrepuestos. En la primera fase del desarrollo embrionario, los genes estructurales son aplicados a una célula simple, dando dos nuevos conjuntos de variables que constituyen dos nuevas células. El proceso se repite recursivamente, generando el fenotipo del individuo. La aplicación de un gen estructural —o incluso la posible aplicación simultánea de varios genes

estructurales — es llamada la expresión del gen (inspirado en los mecanismos genéticos naturales). El concepto de individuo en este modelo de codificación emerge como un nuevo nivel de abstracción: un conjunto de células constituye una estructura. La figura 1.5 ilustra la representación de los genes reguladores y estructurales, y el modo en que son expresados.

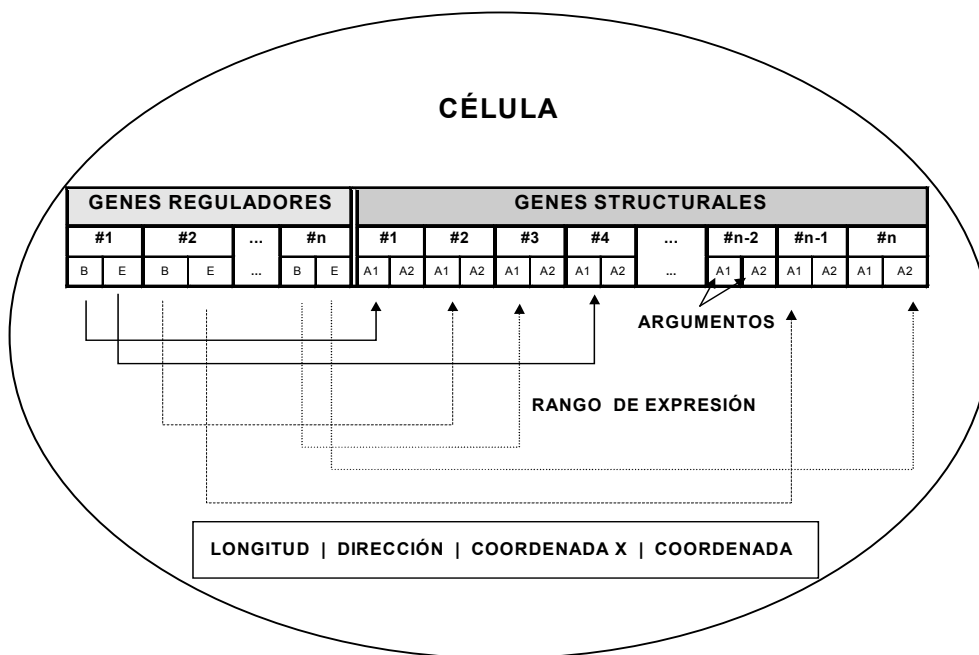


Fig. 1.5. Esquema de codificación genética. Los genes reguladores controlan las expresiones de los genes estructurales. En este ejemplo, la célula consiste en cuatro variables: longitud, dirección del desplazamiento espacial en la división celular, y las coordenadas cartesianas (X e Y) de la posición espacial de la célula. Por cada gen regulador, B y E representan el puntero al primer (Comienzo) y el último (Final) Gen estructural de su campo de expresión. A1 y A2 son los argumentos específicos de cada gen estructural.

Para ilustrar las posibilidades de este modelo de codificación implícita, fue aplicado para codificar individuos consistente en una distribución de puntos en un espacio bidimensional. Una codificación explícita tradicional implementaría el genoma como un vector de coordenadas cartesianas designando la posición de cada punto que constituye el fenotipo del individuo. Para una concreta clase de fenotipos bidimensionales que presentan estructuras repetitivas fácilmente identificables, a diferentes escalas y posiciones, puede ser usado un método de codificación implícita con un procedimiento de desarrollo recursivo. Desde un punto de vista matemático, estos objetos gráficos que presentan características de autoinclusión y autosimilitud son considerados fractales (Mandelbrot, 1977). Desde el punto de vista del modelo de codificación propuesto, la simulación del desarrollo embrionario es llevada a cabo

partiendo de genomas aleatorios. El proceso de desarrollo comienza con una célula con variables que representan la posición de la célula en un espacio cartesiano bi-dimensional, la dirección y el desplazamiento resultando del proceso de división. La expresión del genoma significa la aplicación de operadores matemáticos sobre esas variables y la sucesiva división celular en nuevos conjuntos de células.

La bioforma final muestra similitudes con ciertas formas de vida, o partes de ellas, como conchas. En la figura 1.6 se muestran dos individuos generados de genomas diferentes. El fenotipo del individuo de la figura 1.6(a) muestra un aspecto de concha. Desde una célula inicial, este genoma fue expresado mediante sucesivas generaciones llegando a formar un individuo formado por 1000 células representadas en un espacio bi-dimensional interpretando las coordenadas cartesianas de las variables internas de estas.

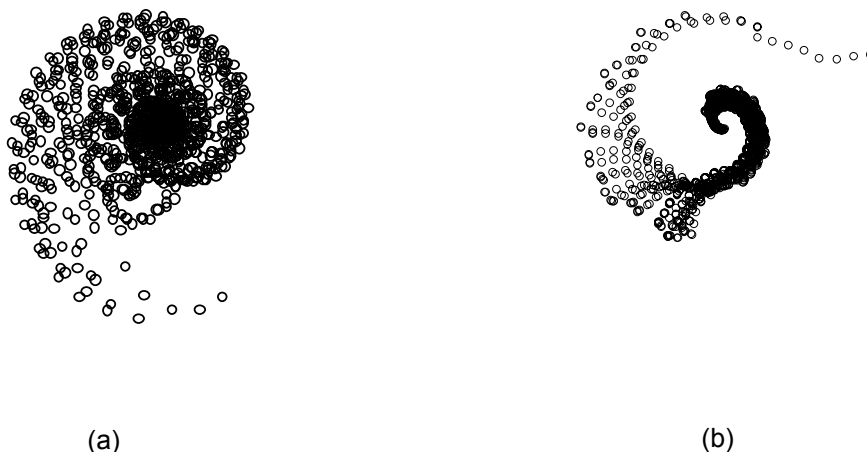


Fig. 1.6. Estructuras de dos individuos en su estado final del desarrollo embrionario. La fisionomía fractal de los individuos puede ser vistas en estas imágenes, especialmente destacable en la primera.

La auto-similitud es la repetición de una estructura a diferentes escalas. Los individuos en la figura 1.6 muestran esta propiedad de auto-similitud: pequeñas formas espirales constituyen la forma completa en espiral del individuo. La naturaleza se beneficia de esta propiedad: patrones de ramificación como en los árboles, helechos, los sistemas circulatorios y respiratorios, o la conectividad del cerebro, siendo almacenado de un modo fractal.

Como es obvio, los operadores genéticos vistos en la sección anterior no son válidos para su aplicación sobre modelos de codificación implícitos, éstos deberán ser desarrollados específicamente para el modelo usado. En el siguiente apartado se podrán ver algunas

propuestas de codificaciones implícitas aplicadas a problemas de redes neuronales incluyendo los operadores genéticos específicos.

Técnicas evolutivas aplicadas al diseño de Redes Neuronales

Una de las principales dificultades encontradas en la aplicación de las redes de neuronas a determinados problemas, consiste en el diseño de la arquitectura neuronal más adecuada. De hecho, en la actualidad, este proceso se lleva a cabo empleando heurísticas y métodos de prueba y error. Los diferentes trabajos de investigación llevados a este respecto, muestran que es posible emplear técnicas basadas en técnicas evolutivas de tal forma que el diseño de redes de neuronas sea automático. Es decir, que exista un algoritmo evolutivo que encuentre la mejor arquitectura neuronal que haga máxima una determinada función de valoración. En este sentido se encaminan los trabajos de un buen número de investigadores. En los siguientes apartados se introducen algunas de sus principales propuestas.

Programación Genética

En el paradigma de la programación genética propuesto por John Koza (Koza & Rice, 1991), los individuos de una población se componen de funciones y terminales apropiados al dominio del problema. El conjunto de funciones típicamente usadas incluyen operadores aritméticos, funciones matemáticas, operadores condicionales lógicos, y funciones específicas del dominio. Cada función del conjunto de funciones debe estar bien definida para cualquier combinación de elementos desde el rango de cada función a los elementos terminales que puedan formar parte de sus argumentos. El conjunto de terminales normalmente usados incluye entradas (sensores) apropiados al dominio del problema y varias constantes. El espacio de búsqueda es el de las posibles composiciones de funciones que puedan ser compuestas de forma recursiva a partir de las funciones disponibles y terminales. La expresión simbólica (*S-expresión*) del lenguaje de programación LISP es especialmente conveniente para el modo de crear y manipular las composiciones y terminales descritos. Esas *S-expresiones* de LISP corresponden directamente al árbol de análisis que es internamente creado por el compilador. Se puede decir por ello que el propio código genético de cada individuo es en sí mismo una función o programa.

Los operadores básicos para la programación genética son dos: la selección proporcional al valor del *fitness* de los individuos (*roulette*) y el cruzamiento. Este último genera dos *S-expresiones* a partir del intercambio de subárboles entre los padres.

El conjunto de funciones empleado para la construcción de una red neuronal basado en un modelo lógico de neurona está compuesta por seis funciones $F=[P, W, +, -, *, \%]$. P es la función *umbral lineal*. W es la función *peso* usada para ponderar la señal que procede de la salida de una función P . Ambas funciones pueden aparecer en el conjunto de funciones con un número variable de argumentos. El conjunto de funciones puede contener también 4 operadores aritméticos: suma (+), resta (-), multiplicación (*), y la división protegida (%) que devuelve 0 si es una división por cero. Estos operadores aritméticos son usados para crear y modificar los pesos de la red neuronal.

El conjunto de terminales (argumentos de la función) contiene las señales de entrada del problema y un posible valor flotante aleatorio. Si para un determinado problema se requiere de dos señales de entrada ($D0$ y $D1$) el conjunto de terminales final es $T=[D0, D1, \text{Valores Flotantes}]$.

El problema del diseño de redes neuronales necesita una serie de restricciones sintácticas para la creación aleatoria de la población inicial. No es posible cualquier combinación de nodos representando funciones y terminales de los conjuntos. Son necesarias reglas de construcción que especifiquen qué estructuras corresponden a verdaderas redes. Las *S-expresiones* deben ser creadas atendiendo a estas reglas de construcción. Además los operadores que modifiquen a estos individuos deben cumplir igualmente las reglas para preservar una estructura de red válida.

Para un problema concreto con una única señal de salida se requiere que el nodo de la raíz del árbol sea una función de umbral lineal P . Las funciones representadas en los nodos inmediatamente por debajo de un nodo P deben ser nodos W . Los nodos por debajo de un nodo W pueden ser elementos terminales, funciones aritméticas o una función P . El nivel por debajo de una función aritmética sólo puede estar formado por valores constantes aleatorios, u otras funciones aritméticas. A su vez, el operador de cruzamiento debe mantener la estructura siguiendo la siguiente restricción: se selecciona un punto de corte en el primero de los padres de entre todos los que forman el árbol de la *S-expresión*, se selecciona el punto de corte en el segundo padre entre los nodos que sean del mismo tipo que la raíz del subárbol seleccionado en el primer padre y se intercambian los subárboles. Para este problema hay 4 tipos de nodos: funciones P , funciones W , señales de entrada (como $D0$ y $D1$) y funciones aritméticas o valores flotantes constantes.

Estas reglas producen un árbol (*S-expression LISP*) que corresponderá con una red neuronal completa. La estructura consiste en elementos de proceso de umbralización lineal (funciones *P*) que procesan entradas ponderadas mediante pesos para producir una salida discreta (0 o 1).

La *S-expresión LISP* mostrada en la figura 1.7, donde las funciones *P* tienen fijado su valor de umbral a 1,0, representa a una red neuronal que resuelve el problema de la función XOR (disyunción exclusiva) para las entradas *D0* y *D1*.

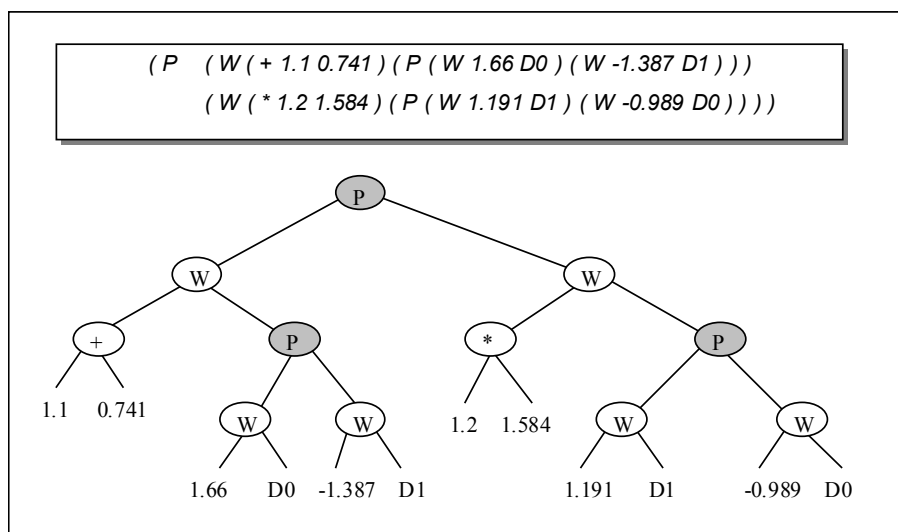


Fig. 1.7. *S-expresión de Lisp que representa una red neuronal que resuelve el problema de la función XOR y su representación en forma de árbol.*

La raíz del árbol contiene una función de umbral lineal (*P*) con dos argumentos. Esto representa a una neurona con dos conexiones sinápticas. Dichas conexiones vienen representadas mediante nodos con funciones peso (*W*). El primer argumento de la función peso de la izquierda es 1.181 (resultado de una operación aritmética de suma sobre dos valores constantes 1,1 y 0,741). El segundo argumento es la salida de una nueva función umbral correspondiente a la salida de una nueva neurona. Esta nueva función umbral posee a su vez dos argumentos correspondientes a sendos nodos de peso donde se ponderan las entradas *D0* y *D1* con 1,66 y -1,387, respectivamente. El segundo argumento de la función umbral raíz es la salida ponderada con 1,901 (resultado de la multiplicación de 1,2 y 1,581) de la función umbral de las señales de entrada *D1* y *D0* ponderadas en 1,191 y -0,989, respectivamente.

Si el problema a resolver posee más de una señal de salida, se recurre a la función *LIST* que devuelve una lista de valores donde cada uno de sus argumentos debe ser una *S-expresión* como las descritas antes. La salida de cada una de esas sub-expresiones corresponde con cada una de las señales de salida, por lo que se requerirán tantas sub-expresiones como señales de salida tenga el problema. El conjunto de funciones disponibles se amplía a $F=[LIST, P, W, +, -, *, \%]$.

Codificación Gramatical

En 1991, Hiroaki Kitano propuso el modelo de codificación de conexiones de redes neuronales que denominó codificación gramatical (*grammatical encoding*) (Kitano, 1990). Éste consistía en un conjunto de reglas de producción que reescriben un símbolo como una matriz 2x2 de símbolos del alfabeto $\Sigma = N \cup P \cup T$, donde $N=\{A,B,...,Z\}$ (conjunto de símbolos no-terminales), $P=\{a,b,...,p\}$ (conjunto de símbolos pre-terminales) y $T=\{0,1\}$ (conjunto de símbolos terminales). La gramática se compone de una parte constante y otra variable. La parte constante es un conjunto de 16 reglas de producción correspondiente a las transformaciones de los símbolos pre-terminales en las 16 posibles matrices 2x2 de símbolos terminales. La parte variable de la gramática es la que se codifica en el genoma y corresponde con las reglas de producción que transforman símbolos no-terminales. La figura 1.8 muestra un ejemplo de genoma (a), el conjunto de reglas que éste codifica (b) y el conjunto de reglas constantes (c).

El proceso de generación del fenotipo comienza con una matriz 1x1, denotada M_0 , formada por un símbolo no-terminal. Sobre este axioma se aplica la regla de producción correspondiente, codificada en el genoma, transformándolo en una matriz 2x2, denotada M_1 . Durante el segundo paso cada símbolo de la matriz M_1 es reescrito en matrices 2x2. Esto produce una nueva matriz 4x4, denotada M_2 . La gramática se interpreta de forma iterativa un número k de iteraciones de forma que, para una red neuronal de n neuronas, 2^{k-1} sea menor que n y 2^k mayor que n . De la matriz resultante M_k de tamaño $2^k \times 2^k$, se extrae una matriz $n \times n$ que será la matriz de conexión de la red.

Los símbolos terminales pueden reescribirse en matrices 2x2 de ceros (en caso de ser el símbolo 0) o de unos (en el caso de ser el símbolo 1). Si, en un momento determinado, no existe una regla para un determinado no-terminal en la matriz, se considera a ese terminal como "muerto" y se aplica la misma regla que para expandir un símbolo terminal 0. Si, por el contrario, hay dos reglas con el mismo símbolo a expandir, la primera de ellas en el genoma

será la regla usada. Si se llega al paso $k-1$ de iteración y aún hay elementos no-terminales éstos son reemplazados por matrices 2x2 de ceros.

(a)

S	A	b	M	D	A	c	p	a	c	D	a	a	A	b	C	a	a	a	a
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(b) $S \Rightarrow \begin{bmatrix} A & b \\ M & D \end{bmatrix} \quad A \Rightarrow \begin{bmatrix} c & p \\ a & c \end{bmatrix} \quad D \Rightarrow \begin{bmatrix} a & a \\ A & b \end{bmatrix} \quad C \Rightarrow \begin{bmatrix} a & a \\ a & a \end{bmatrix}$

(c) $a \Rightarrow \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad b \Rightarrow \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \quad c \Rightarrow \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad p \Rightarrow \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$

(d) $S \Rightarrow \begin{bmatrix} A & b \\ M & D \end{bmatrix} \Rightarrow \begin{bmatrix} c & p & 0 & 0 \\ a & c & 0 & 1 \\ 0 & 0 & a & a \\ 0 & 0 & A & b \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

Fig. 1.8. Codificación Gramatical: (a) Codificación del Genoma. (b) Reglas de producción generadas a partir del genoma. Los símbolos no-terminales se reescriben como matices 2x2 de símbolos no-terminales y pre-terminales. (c) Reglas de producción fijas. Los símbolos pre-terminales se reescriben como matrices 2x2 de símbolos terminales. (d) Proceso de generación de la matriz de conexión a partir de la aplicación iterativa de las reglas de producción sobre el axioma S.

Algoritmo de Adquisición Generalizada de Enlaces Recurrentes

Propuesto en Angeline et. al. (Angeline et al., 1994), el algoritmo GARL (*Generalized Acquisition of recurrent Links*) es un método evolutivo que prescinde del operador de cruzamiento para la generación de descendencia, siendo únicamente el operador mutación el que realiza esta tarea. Se basa para ello en el problema denominado *convención competitiva* (*competing convention*) (Schaffer, Whitley, & Eshelman, 1992), donde, por ejemplo, dos

genomas que producen estructural y computacionalmente la misma red neuronal, pero con sus neuronas asignadas en distinto orden, son cruzados (figura 1.9). Como los genomas son distintos entre sí el cruzamiento puede dar lugar a múltiples copias de una misma neurona y ausencia de otras unidades, haciendo que la red resultante sea computacionalmente peor. En su trabajo, Angeline afirma que el uso de algoritmos genéticos clásicos para el desarrollo de redes neuronales no es apropiado ya que la generación de descendencia se realiza de forma demasiado aleatoria, y poco guiada.

Las redes neuronales sobre las que se aplica el algoritmo GARL están formadas por tres grupos de neuronas, de entrada (que reciben la información de entrada del problema a resolver), ocultas (que procesan esa información) y de salida (cuyos valores constituyen la información de salida de la red neuronal). El algoritmo propuesto por Angeline exige que la arquitectura general de las redes sobre las que trabaja debe tener una serie de restricciones:

- *R1*: No puede haber enlaces a una neurona de entrada.
- *R2*: No puede haber enlaces *desde* una neurona de salida.
- *R3*: Dos nodos *x* e *y* sólo pueden estar conectados mediante un único enlace desde *x* a *y*.

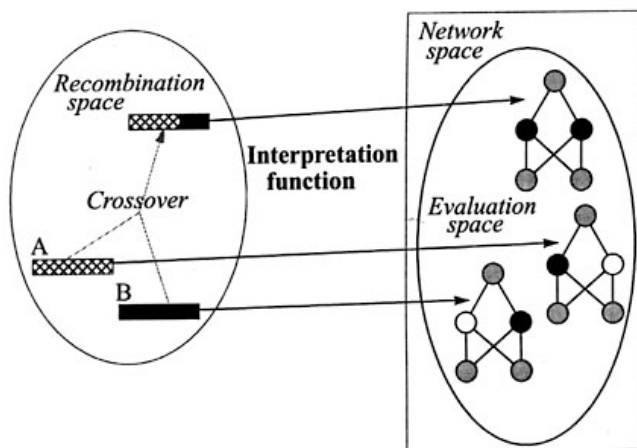


Fig. 1.9. Problema de la convención competitiva expuesto por Schaffer (Schaffer et al., 1992).

El número de neuronas de entrada y de salida vendrá dados por el problema que se pretende resolver, el número de neuronas ocultas debe estar entre 0 y un límite impuesto por el

usuario h_{max} , y el peso de los enlaces son valores reales. Las neuronas, a excepción de las de entrada, poseen una función de activación sigmoide estándar.

GNARL inicializa la población con redes generadas aleatoriamente. El número de nodos ocultos es elegido en una distribución uniforme por un rango dado por el usuario. El número de enlaces iniciales se elige de forma similar. Los nodos sobre los que incide un enlace se elige de acuerdo con la mutación estructural comentada más abajo. Cuando los enlaces han sido elegidos se les asignan un valor real en el rango $[-1,1]$.

En cada generación, las redes son evaluadas por una función *fitness* dada por el usuario. Los individuos pertenecientes al 50% de la población con mejor puntuación son designadas como padres de la siguiente generación, el resto son descartados. La generación de la descendencia se realiza en tres pasos: copia del padre, determinación de grado de mutación y finalmente la mutación de la copia. El grado de mutación que sufrirá la red vendrá dado por lo cerca que se encuentre el valor de *fitness* de la red obtenida al mejor valor esperado, siguiendo la siguiente ecuación 1.6:

$$T(\eta) = 1 - \frac{f(\eta)}{f_{max}} \quad (1.6)$$

donde η representa a una de las neuronas de la red, f es la función de evaluación (*fitness*) y $T(\eta)$ es el grado de modificación aplicable a la neurona (*temperatura*).

Esto permite empezar modificando la red fuertemente en las etapas iniciales y afinando los cambios conforme más cerca esté la red de la deseada.

Angeline divide la mutación de la red en dos clases: *mutación paramétrica* (modificación de los pesos) y *mutación estructural* (modificación del número de nodos ocultos y enlaces). La mutación paramétrica consiste en la perturbación de cada *peso* w de los enlaces de la red mediante la adición de un ruido gaussiano, siguiendo las ecuaciones 1.7 y 1.8:

$$\hat{T}(\eta) = U(0,1)T(\eta) \quad (1.7)$$

$$w = w + N[0, \alpha \hat{T}(\eta)] \quad \forall w \in \eta \quad (1.8)$$

donde $U(0,1)$ es una variable uniforme aleatoria sobre el intervalo $[0,1]$, α es una constante de proporcionalidad definida por el usuario y $N(a,b)$ es una variable gaussiana aleatoria donde a

es la media y b la varianza. El otro tipo de mutación, la estructural, agrupa 4 operadores: *Inserción/ Eliminación* de neuronas ocultas e *Inserción/ Eliminación* de enlaces. Por cada uno de estos operadores se define un rango $[\Delta_{\max}, \Delta_{\min}]$ que limita el número máximo y mínimo de elementos que insertados o eliminados. El número de elementos a añadir o eliminar viene dado por la ecuación 1.9:

$$\Delta_{\min} + \left[U(0,1) \hat{T}(\eta) (\Delta_{\max} - \Delta_{\min}) \right] \quad (1.9)$$

Para evitar un salto radical entre el *fitness* del padre a la descendencia, la mutación estructural debe preservar el conocimiento de la red, por ello cuando se inserta un nuevo enlace éste es inicializado a 0, y cuando se inserta un nuevo nodo, éste no está conectado a ningún otro. En el caso de la eliminación de un nodo no se puede evitar un cambio brusco en el *fitness*, ya que supone la eliminación de todos los enlaces que lo unen a otras neuronas.

Codificación celular

El trabajo de Frédéric Gruau (Gruau, 1995) está inspirado en gran medida por el modelo de árbol gramatical propuesto por J. Koza en su *Programación Genética* (Koza & Rice, 1991). Gruau se basa, al igual que J. Koza, en una codificación del genoma en forma de árbol gramatical pero, a diferencia de éste, el árbol no representa directamente a la red neuronal sino al conjunto de instrucciones que aplicados sobre una célula inicial dará lugar al fenotipo desarrollado. Cada nodo etiquetado del árbol representa a un gen: una instrucción para el desarrollo de una determinada célula y el conexionado con las de su entorno. Estos nodos pueden ser binarios, unarios o terminales dependiendo la naturaleza del gen al que representa. Cada nodo es leído e interpretado por una célula en un momento determinado del desarrollo, comenzando con una única célula inicial (*célula madre*) cuya cabeza de lectura apunta a la raíz del árbol genotípico, y acabando con cada una de las cabezas de lectura del conjunto de células completo de la red final apuntando a un nodo hoja distinto.

El fenotipo de un individuo es generado mediante el proceso denominado *desarrollo embrionario*, consistente en partir de una sola célula inicial conectada a todas las entradas y salidas, denominada *célula madre*, y sobre la cual se aplica de forma iterativa y guiada por el genotipo un conjunto de operadores que producen la duplicación de la célula, la alteración de

conexiones existentes o la modificación de las variables que forman el estado interno de la neurona. El proceso de desarrollo embrionario se ilustra en la figura 1.10.

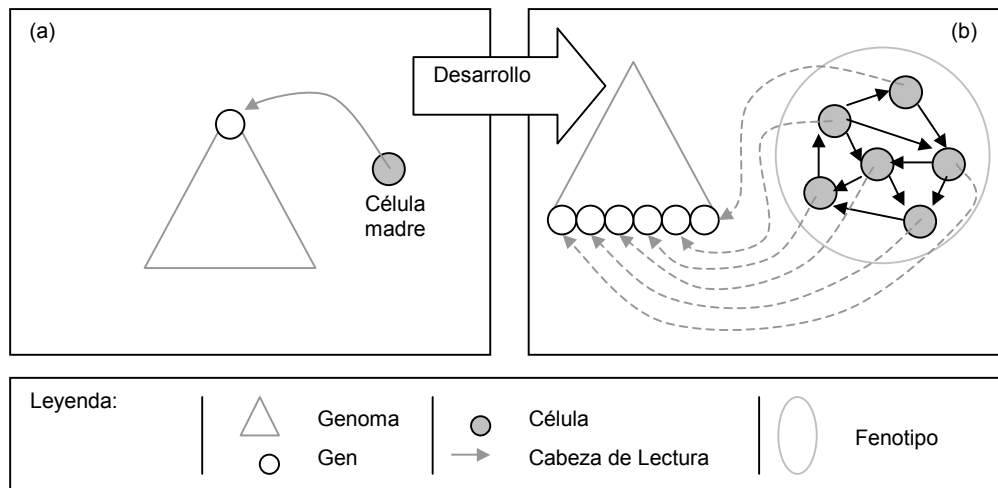


Fig. 1.10. Desarrollo del fenotipo a partir del genoma. (a) Estado inicial. La cabeza de lectura de la célula madre apunta al nodo raíz del árbol. (b) El proceso de desarrollo se da por finalizado cuando la cabeza de lectura de cada una de las células apuntan a un nodo terminal del árbol. El fenotipo se encuentra totalmente desarrollado.

Los genes son clasificados en tres tipos atendiendo a su función en el desarrollo de la red:

- **Genes de división.** Producen la división de la célula. Las células resultantes de la división heredan los valores internos de la célula origen y la cabeza de lectura de éstas apuntan a cada uno de los nodos hijos. Vienen representados por tanto como nodos binarios (con dos hijos).
- **Genes de estado.** Producen una modificación en el conjunto de conexiones sinápticas de la célula, alterando el peso o eliminando una determinada conexión. La cabeza de lectura de la célula pasará al gen cuyo nodo es hijo del que ha sido leído. Estos genes son representados por nodos unarios.
- **Genes terminales.** Indican el final del desarrollo de una célula. El nodo que lo representa es un nodo hoja.

El operador de mutación actúa cambiando uno de los nodos del árbol por otro de igual aridad. El cruzamiento de dos individuos se realiza intercambiando subárboles de cada uno de árboles que forman el genoma de los padres, dando lugar a nuevos individuos. La elección de la altura del corte de los subárboles determinará el nivel de cambio provocado en la estructura cerca de las hojas.

El modelo puede ser extendido para aplicar nuevos conceptos de topologías o modificar el tipo de modelo neuronal de la red que se desea implementar, con sólo añadir genes que modelen la característica deseada. De esta forma, para incluir el concepto de modularidad, como el que exhibe el modelo de Kitano (Kitano, 1990), basta con dividir el genoma en secciones, denominadas *cromosomas*, modelado como un conjunto de árboles que estructuran de forma jerárquica la red. De este modo, el primer cromosoma define la forma de la red neuronal a más alto nivel y el último codifica los elementos a menor escala. Cada una de las hojas de un árbol se enlaza con la raíz de un árbol con un índice estrictamente mayor, salvo el último árbol cuyas hojas son elementos terminales. Esto puede ser aplicado en el desarrollo de estructuras de redes de neuronas que requieran repetición de esquemas comunes (por ejemplo, estructuras columnares en los niveles inferiores del sistema de visión según el modelo de Hubel y Wiesel (Hubel & Wiesel, 1962))).

Capítulo 2

Estudio de un algoritmo genético basado en codificación celular

Tras el estudio de los diversos modelos de codificación implícita introducidos en el apartado 1.2.4, el elegido para el desarrollo de este proyecto es el denominado *codificación celular*, propuesto por Frédéric Gruau (Gruau, 1994). Las principales razones que han llevado a su elección son: el estructurado modo que presenta el genoma en forma de árbol semántico, el proceso con el cual se genera el circuito neuronal, denominado *desarrollo embrionario*, y su facilidad de adaptación para la implementación de cualquier topología de red neuronal y modelo de neurona. Esta última característica del modelo es especialmente importante para el presente proyecto, ya que se desea estudiar la viabilidad del uso de algoritmos genéticos para la búsqueda de topologías óptimas para distintos tipos de redes neuronales, comenzando el estudio con un modelo neuronal lógico, y extendido a continuación a un modelo de neurona de integración y disparo.

Los siguientes subapartados desarrollan el conjunto de elementos que conforman el estudio, empezando por la descripción completa del modelo básico de *codificación celular* y el proceso de generación de la red neuronal a partir de la interpretación del genoma (desarrollo embrionario), en el apartado 2.1. Las adaptaciones llevadas a cabo sobre el modelo básico y el modelo neuronal empleado para la resolución de los problemas lógicos son detallados en el apartado 2.2. Entre las adaptaciones llevadas a cabo destacan la modificación del conjunto de genes, la estructuración del genoma llevado a cabo, los operadores genéticos aplicados y la función de evaluación (*fitness*) adecuada al tipo de problema planteado. En el apartado 2.3 se describen las diversas políticas de selección, asociadas a los operadores genéticos, empleadas en el estudio de la búsqueda de topologías óptimas. Por último, en el apartado 2.4, se presentan los resultados obtenidos de la aplicación del algoritmo genético para la resolución de problemas lógicos.

2.1 Modelo básico de codificación celular

El genoma reúne el conjunto de instrucciones necesarias para la generación de un individuo. En el modelo seguido, inspirado en el modelo de *codificación celular*, el genoma es representado como un árbol con ramas ordenadas cuyos nodos etiquetados representan genes: instrucciones en el proceso de desarrollo embrionario. Una célula es un nodo de un grafo con conexiones ordenadas, con una cabeza de lectura interna que señala a un único nodo del árbol genotípico. Cada célula leerá, por tanto, una posición diferente del genoma, y cada gen, representado por un nodo del árbol, es leído e interpretado por una única célula. Estos genes representan las instrucciones necesarias para el desarrollo celular, modificando el estado de la célula, modificando sus conexiones o dividiéndola en nuevas células. Durante un paso del desarrollo embrionario una célula ejecuta la instrucción representada por el gen leído y mueve su cabeza de lectura al siguiente nodo del árbol (hijo del primero). Sólo tras finalizar el proceso de desarrollo embrionario pueden considerarse a las células como neuronas y al grafo como la red neuronal completa. Cada célula posee un conjunto de parámetros que forman su estado interno. Alguna de estos parámetros son usadas únicamente durante el proceso de desarrollo mientras que otros determinan los valores propios de la futura neurona o conjunto de neuronas que se formen a partir de ella. A este ultimo tipo pertenecen el parámetro *umbral* y los *pesos* de las conexiones.

El desarrollo de una red neuronal comienza con una única célula llamada *célula madre*, considerada a la vez, neurona de entrada y de salida de la red neuronal. La cabeza de lectura de esta célula apunta inicialmente al nodo raíz del árbol genotípico y sus variables de estado se fijan a unos valores por defecto. Cuando la instrucción que este gen lleva asociado se ejecuta sobre la célula ésta será modificada, bien dividiéndose de una forma determinada (si es un gen *de división*) o bien cambiando su estado interno (si es un gen *de estado*). En el primer caso, a cada una de las células generadas de la división se les hace corresponder un nodo hijo del nodo interpretado, y en el segundo caso a la propia célula modificada se le asigna el único nodo hijo del primero. Este proceso se repite hasta que las cabezas de lectura de todas las células apunten a nodos terminales, dando por terminado el desarrollo. Al final del proceso, la red está estructurada en neuronas *de entrada*, *de salida* y *ocultas* (intermedias). El tipo del gen representado en un nodo rige el número de nodos hijos que este posee. De esta forma un nodo que representa un gen de división tendrá varios nodos hijo (tantos como células resulten de la división), mientras que uno que representa un gen de estado poseerá un único nodo hijo, y finalmente, un nodo que represente un gen terminal será un nodo hoja en el árbol del genoma.

El conjunto de genes que forma parte del modelo de básico propuesto por F. Gruau, usado para el desarrollo de un modelo de red neuronal lógica, es clasificado en tres grupos según la función que cumple el gen durante el desarrollo embrionario:

- **Genes de división.** La ejecución de un gen de división genera dos células a partir de una. En una división secuencial (implementada por un gen denotado *SEQ*) la primera célula hija hereda los enlaces de entrada de la célula original y la segunda célula los enlaces de salida. La primera hija es conectada a la segunda con un peso de 1 con un enlace orientado de la primera a la segunda hija. Un nodo que represente un gen de división debe tener aridad dos, es decir tener dos nodos hijos. La primera de las células hijas mueve su cabeza de lectura al nodo raíz del subárbol izquierdo y la segunda al del subárbol derecho. De igual forma en la división paralela (implementada por un gen denotado *PAR*) la célula inicial se divide en dos nuevas células pero en este caso ambas heredan las conexiones de entrada, salida y los valores de los registros internos que forman el estado interno de la célula inicial. En la figura 2.1 puede verse gráficamente el efecto de la ejecución de estos genes.

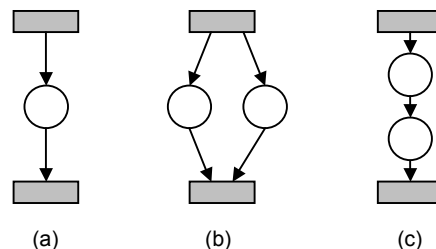


Fig. 2.1. Efecto de los distintos genes de división aplicados sobre una célula inicial. (a) Configuración inicial. (b) Aplicación de un gen de división paralela (PAR). (c) Aplicación de un gen de división secuencial (SEQ).

- **Genes de estado.** La ejecución de un gen de estado produce una modificación en el estado interno de la célula o en el conjunto de conexiones de entrada, tras lo que la cabeza de lectura de la célula pasa al nodo hijo. Estos genes son representados por nodos unarios en el árbol. Los genes denotados como *INCBIAS* y *DECBIAS* producen el incremento y decremento en 1, respectivamente, del valor de umbral de la célula que produce la ejecución del gen. La modificación del peso de una

conexión de entrada a una célula es producida por los genes *VAL+* y *VAL-* que lo incrementa y decremента, respectivamente. La eliminación total de una conexión de entrada es llevada a cabo por el gen *CUT*. Estos últimos genes (*VAL+*, *VAL-*, *CUT*) hacen uso de un parámetro que indica el enlace sobre el que se debe aplicar la acción de modificación o eliminación. El valor de este parámetro es modificado por la acción de los genes *INCLR* y *DECLR* que lo incrementa y decremента, respectivamente, cambiando así el enlace sobre el que los genes *VAL* o *CUT* realizarán su acción. En la figura 2.2 puede verse gráficamente el efecto de la ejecución de estos genes.

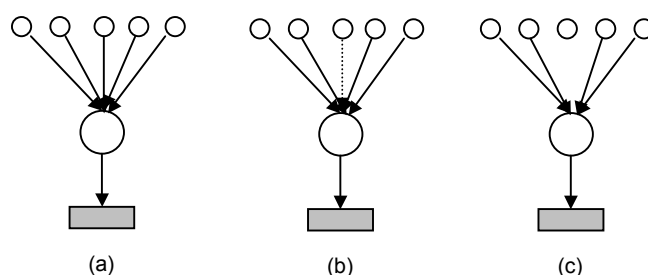


Fig. 2.2. Efecto de los distintos genes de estado que alteran las conexiones de entrada aplicados sobre una célula inicial. (a) Configuración inicial con su conexionado de entrada detallado. (b) Aplicación de un gen de modificación de conexión (*VAL+* o *VAL-*) sobre su tercer enlace. (c) Aplicación de un gen de eliminación de conexión (*CUT*) sobre su tercer enlace.

- **Genes terminales.** La ejecución de un gen terminal (*END*) da por finalizado el desarrollo de una célula. A partir de ese momento, ésta es considerada una neurona y su conexionado de entrada inalterable. Estos genes son representados por nodos hojas del árbol. El número de nodos hojas en el árbol es igual al número de neuronas que serán desarrolladas a partir de la célula madre.

El conjunto final de genes queda formado por los genes *PAR*, *SEQ*, *VAL+*, *VAL-*, *CUT*, *INCBIAS*, *DECBIAS*, *INCLR*, *DECLR* y *END*.

Los operadores genéticos empleados para la generación de nuevos individuos a partir de los ya existentes son el de *cruzamiento* y el de *mutación*. La aplicación del primero consiste en el intercambio de subárboles entre dos genomas de individuos distintos. Dicho intercambio modifica el conjunto de instrucciones contenido en cada genoma, de modo que los nuevos individuos poseen mayoritariamente las características topológicas de uno de sus padres con algún rasgo del otro (fig. 2.3). La elección de los subárboles a intercambiar es aleatoria, con la

única limitación de que el número de nodos hojas que poseen (igual al número de neuronas finales cuyo desarrollo controla) debe ser similar aunque no necesariamente el mismo. La posibilidad de que ambos subárboles a intercambiar posean un número diferente de hojas, producirá una alteración en el número de neuronas que poseen los individuos hijos con respecto a los padres. Esta característica permite que se amplíe la búsqueda de soluciones a topologías con números de neuronas fuera del intervalo usado en la generación de los individuos de la población inicial.

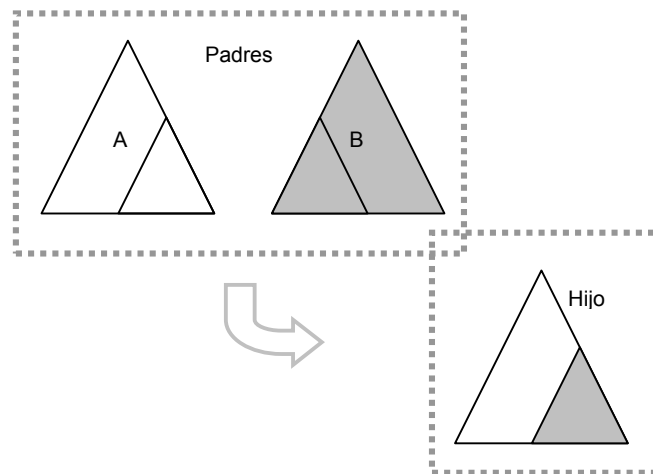


Fig. 2.3. Operador de cruzamiento aplicado sobre dos individuos. El individuo resultante mantiene la estructura básica de uno de sus padres, y cierto rasgo del otro progenitor, producido por el subárbol intercambiado.

El operador de *mutación*, por su parte, consiste en la sustitución de algunos de los genes que forman el conjunto de instrucciones para el desarrollo de un individuo, por otros diferentes pero igual aridad. La limitación en la sustitución de un gen por otro del mismo tipo es necesaria para mantener inalterada la estructura del árbol. Un nodo, que representa a un gen, debe ser sustituido únicamente por otro con la misma aridad, representando a otro gen del mismo tipo. La aplicación del operador sobre un genoma se realiza mutando los genes que lo forman con una probabilidad previamente fijada, de modo que sólo sea mutado un porcentaje determinado de los genes del árbol genotípico.

2.2 El modelo de codificación celular adaptado

El uso del algoritmo genético descrito en la búsqueda de soluciones a un determinado tipo de problemas requiere una serie de modificaciones en el conjunto de genes, para adaptar las instrucciones que representan al modelo de red neuronal y topología requerido para la resolución del problema. De igual forma es deseable trasladar la posible estructuración de las redes definidas al genoma, si ésta existe y es posible. En los siguientes apartados se describe el modelo de red neuronal usado y las adaptaciones llevadas a cabo sobre el conjunto de genes y la propia estructura del genoma.

2.2.1 Modelo de red neuronal

Para iniciar el estudio de la aplicación del algoritmo genético, para la obtención de topologías de redes neuronales óptimas, se ha elegido un modelo de neurona lógico especialmente sencillo pero capaz de ser aplicado para la resolución de problemas lógicos. El objetivo propuesto en esta fase es conseguir que el algoritmo genético encuentre redes capaces de resolver un problema expresado mediante *tablas de verdad*. El modelo básico de neurona elegido es el propuesto por McCulloch y Pitts, expresado en las ecuaciones 2.1 y 2.2, donde cada neurona es un elemento de conmutación binario que puede tomar los valores de cero (si está desactivada o en reposo) o uno (cuando está activada). La red tiene como nodos las N neuronas, que están conectadas entre sí con enlaces sinápticos, y asociado a cada uno de estos enlaces (j,i) un valor w_{ij} , denominado *peso sináptico*, que representa la influencia que tiene el estado de la neurona j sobre el de la neurona i . De este modo, el estado de una neurona está determinado por la influencia total de los estados de las neuronas a las que está conectada.

$$S_i(t+1) = g\left(\sum_j w_{ij} S_j(t) - \theta_i\right) \quad (2.1)$$

$$g(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{si } x \leq 0 \end{cases} \quad (2.2)$$

El estado de la neurona i en el instante $t+1$ ($t=0,1,\dots$) está representado por $S_i(t+1)$. El valor θ_i , asociado a la neurona i , es denominado *umbral* de dicha neurona, y es el valor que debe ser superado por la entrada de la neurona (llamada *potencial sináptico*) para que ésta sea activada. La función de activación $g(x)$, usada en este caso, es la denominada *función de paso* o de *Heaviside*.

La topología de la red, inspirada en modelos multicapas agrupa conjuntos de neuronas según la función que desempeñan. De este modo, las unidades consideradas de entrada (o sensores) forman la capa de entrada, el conjunto de neuronas consideradas de salida forman la capa de salida, y el resto se distribuyen en una o más capas (denominadas ocultas). El uso de capas intermedias se hace necesario para incluir en el conjunto de problemas a resolver aquellos que no son separables linealmente. Un claro ejemplo de la incapacidad de los modelos mono-capa ante este tipo de problemas lo presenta el *Perceptrón*, propuesto por Rosenblatt (Rosenblatt, 1962), y su conocida incapacidad para resolver el problema de la *OR-exclusiva*. Una red con las características propuestas puede resolver cualquier problema expuesto como una tabla de verdad. Valga como ejemplo la red multicapa mostrada en la figura 2.3, que resuelve la función, no separable linealmente, *OR-exclusiva*.

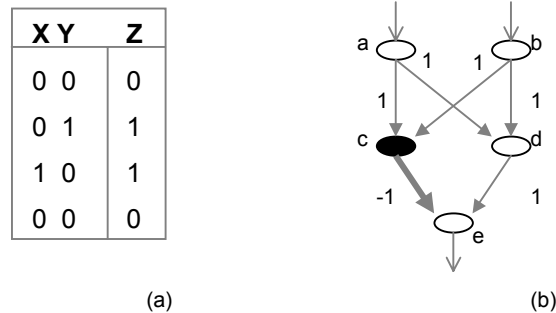


Fig. 2.3. (a) Tabla de verdad de la función OR-exclusiva (XOR). (b) Red neuronal con cinco neuronas que implementa la función OR-exclusiva. Los umbrales de las neuronas a, b, d, y e tienen valor 0 y el de la neurona c a 1. Todos los enlaces tienen peso 1 salvo el de la conexión c-e cuyo peso es -1. La neurona d se activa cuando al menos una de las neuronas a o b se activa. La neurona c sólo se activa cuando a y b están activas simultáneamente, ya que requiere de ambas para que su entrada supere el valor de umbral de 1. La neurona e tiene un enlace sináptico positivo (procedente de d) y otro negativo (procedente de c) y sólo se supera su umbral (y por tanto se activa) cuando d está activa y c no, esto es cuando sólo una de las entradas de la red neuronal es 1. La salida de e corresponde por tanto con el valor Z de la tabla de verdad de la función OR-exclusiva.

La topología de red empleada en este proyecto no sigue fielmente el concepto de capa que puede encontrarse en modelos como el *Perceptrón*, donde cada neurona está conectada con todas las neuronas de las capas anterior y posterior, y con ninguna de las neuronas con la que comparte capa (Fig. 2.4.a). Para el desarrollo del modelo de red se ha considerado un esquema de conexionado más general donde el concepto de *capa de neuronas* se extiende a un concepto más general denominado *nube de neuronas*. En este último, una neurona puede estar conectada a neuronas de la capa precedente, posterior y propia, sin estar obligado a ello en ningún caso (Fig. 2.4.b).

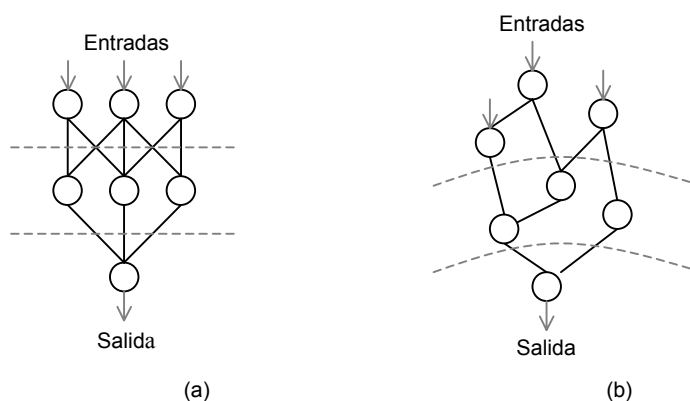


Fig. 2.4: (a) Esquema de conexionado clásico de modelos multicapa, como el *Perceptrón*. Las neuronas de una capa están plenamente conectadas con las neuronas de la capa anterior y posterior. (b) Esquema de conexionado con la extensión del concepto de "capa" al de "nube" de neuronas (menos restrictivo).

El concepto de topologías de redes estructuradas en nubes de neuronas será ampliamente usado en fases posteriores del proyecto, referidas principalmente al estudio del modelo de red de neuronas de integración y disparo y la aplicación del algoritmo genético, por lo que el estudio en esta fase de genomas que permitan desarrollar este tipo de topologías, queda ampliamente justificado.

2.2.2 El conjunto de genes

El modelo neuronal desarrollado por el conjunto de genes que forman el modelo básico de *codificación celular* no limita el rango de valores de los parámetros *umbral* de la neurona, y

peso de la conexión sináptica, pudiéndose alcanzar cualquier valor entero con la sola concatenación de genes que incrementen o decrementen esos parámetros. El modelo de red y neurona, introducido en el apartado 2.2.1, tiene limitado a 0 y 1 los valores posibles del parámetro *umbral*, y a 1 y -1 los valores posibles para los pesos de las conexiones. Estos cambios hacen necesario una modificación sobre el conjunto de genes que permita la generación correcta de fenotipos según las especificaciones del modelo de neurona. Las modificaciones realizadas se centran principalmente en el conjunto de genes de estado, dado la relación directa de éstos con las diferencias entre ambos modelo de neuronas.

Los genes *VAL+* y *VAL-* son sustituidos por un único gen *VAL* que alterna el valor de *peso* de las conexiones sinápticas entre 1 y -1, quedando limitado únicamente a estos dos valores. La misma idea es aplicada sobre el parámetro *umbral*. La sustitución de los genes *INCBIAS* y *DECBIAS* por un único gen *BIAS* que alterna entre los posibles valores del parámetro *umbral* (1 y 0) limita el parámetro a los valores deseados. Por otro lado, el uso de la variable interna de la célula, que selecciona uno de los enlaces de entrada de la célula, puede ser eliminada, y con ella los genes *INCLR* y *DECLR*, si es añadido un parámetro *n* a los genes *VAL* y *CUT*, que cumpla dicha función. De esta forma los genes pasarán a ser denotados *VAL-n* y *CUT-n*, donde *n* es la posición del enlace de entrada de la célula sobre la que se realiza la acción.

El conjunto final de genes queda formado por los genes *PAR*, *SEQ*, *VAL-n*, *CUT-n*, *BIAS* y *END*. Su uso en el desarrollo de redes neuronales asegura la obtención de fenotipos correctos para el modelo neuronal descrito.

2.2.3 La estructuración del genoma

La estructura básica del genoma en forma de árbol de genes puede emplearse para definir, en principio, fenotipos que carecen de una estructura definida en su fenotipo, pero, como puede observarse, el modelo de red neuronal definido en el apartado 2.2.1 posee una estructura de capas claramente definida. Conseguir adaptar esta característica del fenotipo al genotipo, para reducir el espacio de búsqueda a, únicamente, fenotipos estructuralmente válidos, es el siguiente objetivo. Para ello basta con ampliar el concepto de genoma al de una colección de árboles que definan distintas zonas (capas) del fenotipo. Concretamente, para el modelo de red neuronal lógica descrito puede considerarse un genoma como un conjunto de tres árboles de

genes donde cada uno dirija la evolución de una de las capas. Cada uno de estos árboles pueden ser denominado *cromosomas*, por analogía con su homónimo biológico (figura 2.5). De esta forma tenemos genomas divididos en cromosomas, que definen zonas distintas del fenotipo, y cromosomas formados por conjuntos de genes. Esta estructura del genoma permite además controlar el número de neuronas que forman parte de cada una de las capas de la red de forma independiente. Esto es especialmente útil en el caso de la búsqueda de redes que puedan resolver un problema definido mediante una tabla de verdad con un determinado número de entradas y salidas. Con ello se consigue el objetivo de reducir el espacio de búsqueda a únicamente topologías estructuralmente válidas, quedando fuera aquellas que no se ajustan al problema que se desea resolver.

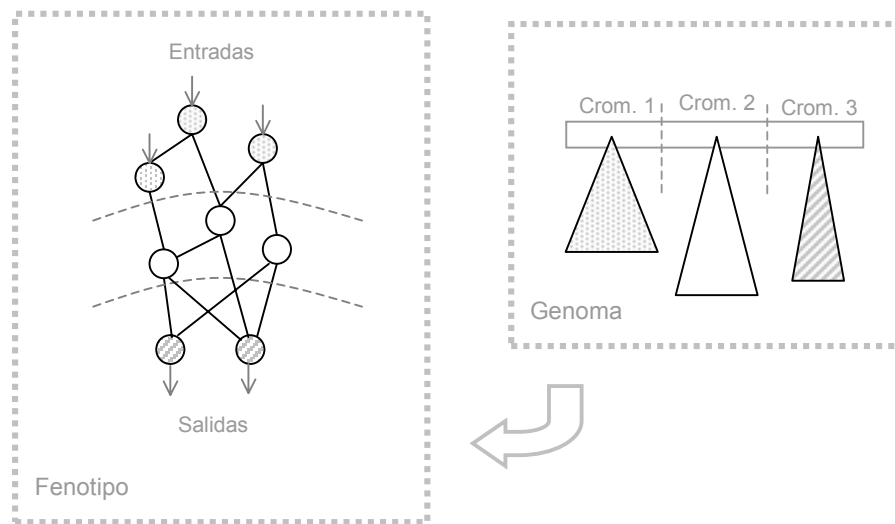


Fig. 2.5: Estructuración del genoma en tres cromosomas que definen sendas capas de la red neuronal que constituye el fenotipo del individuo.

2.2.4 La población inicial

Los individuos que forman la población inicial deben representar de forma homogénea el espacio de búsqueda. El tamaño de los individuos que forman la población es indicado en forma de intervalo que limita el número de neuronas que deben componer las redes neuronales

de la población inicial. Este intervalo debe ser escogido adecuadamente para maximizar las probabilidades de encontrar individuos solución (relación tamaño-complejidad del problema). Los individuos de la población inicial están constituidos por redes cuya capa oculta posee un número de neuronas elegido aleatoriamente dentro dicho intervalo, y cuyas capas de entrada y salida poseen un número de neuronas determinado por el tamaño de los patrones de entrada y salida del problema a resolver. Con el tamaño seleccionado de cada una de las capas que formarán la futura red neuronal se procede a la creación de los cromosomas que forman el genoma del individuo. La generación del árbol de genes, que constituye un cromosoma, comienza con la generación del nodo raíz. A cada nodo que forma el árbol de genes se asigna un valor que indica el número de nodos hojas que debe tener. El valor asignado al nodo raíz es igual al número de neuronas seleccionado para la capa correspondiente al cromosoma al que pertenece. El gen que ocupa una determinada posición en el árbol se elige en dos etapas, inicialmente eligiendo el tipo del gen que debe ocupar la posición y finalmente eligiendo el gen concreto dentro del tipo seleccionado. La primera elección se realiza atendiendo al valor del número de nodos hojas asignado (equivalente al número de neuronas cuyo desarrollo controla el gen), eligiendo entre *genes de división* o *de estado* si el nodo debe poseer más de un nodo hoja, y entre *genes de estado* o *terminal*, si sólo debe poseer un único nodo hoja. Una vez seleccionado el tipo de gen se elige el gen concreto que ocupará la posición en el genoma de entre los disponibles en ese tipo (*PAR* o *SEQ* si es un gen de división, *VAL-n*, *CUT-n* y *BIAS* si es *de estado* y *END*, únicamente, si es un *gen terminal*). La elección de un tipo u otro en la creación del árbol define la estructura definitiva del árbol. La elección de un gen de división produce la inserción de un nodo de aridad 2, es decir, inserta un nodo con dos subárboles, repartiéndose el número de nodos hojas que dependen del nodo insertado entre los nodos raíz de cada uno de los subárboles. La elección de un *gen terminal* para ocupar una posición da por cerrada la rama en la que se inserta, siendo el nodo del *gen terminal* un nodo hoja del árbol. La elección de un *nodo de estado* produce la inserción de un nodo de aridad 1, es decir, con un único hijo que hereda el valor inalterado del número de nodos hojas del padre.

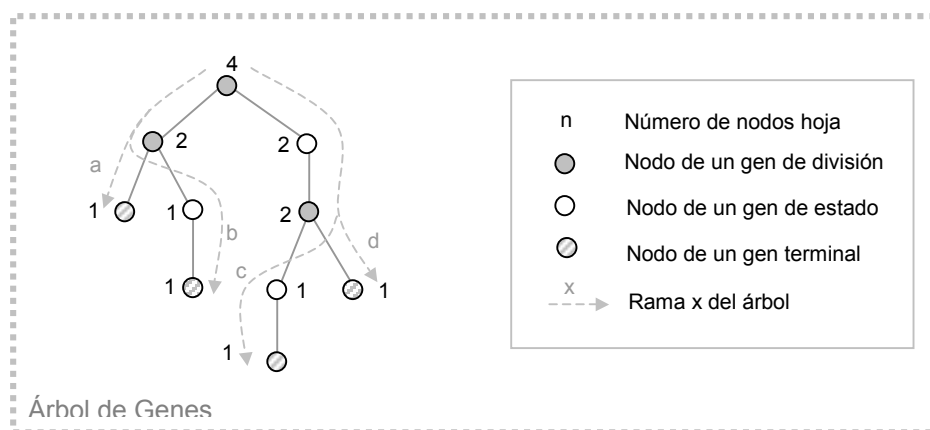


Fig. 2.6: Efecto de la aplicación de diversos tipos de genes en la estructura final del árbol.

Observando cada una de las ramas de un árbol de genes construido de esta forma, se comprueba cómo el valor del número de nodos hojas de los nodos de la rama se comporta como un contador que se decrementa al descender por las ramas del árbol. La excepción a este decremento se produce en los nodos de estado donde el “contador” se mantiene inalterado al pasar del padre al hijo. En la figura 2.6 puede observarse cómo siguiendo la rama etiquetada como *a*, los valores del número de nodos hoja de cada nodo se decrementa al descender por la rama (4-2-1). En la rama *c* puede comprobarse cómo, en la secuencia de valores del número de nodos hojas, existe dos puntos donde el valor no es decrementado, producidos por sendos nodos *de estado* (4-2-2-1-1). Según el *Teorema de los Contadores* (Floyd, 1967) este fenómeno puede llevar a la no terminación del proceso de generación del árbol. Efectivamente, basta con que se encadene una sucesión infinita de genes de estado en una determinada rama del árbol para que el proceso de generación del árbol no termine. Si bien esto no es probable, si puede producir con relativa facilidad cadenas de *genes de estado* que se traduzcan en ramas excesivamente largas, que incluso pueden ver anulados sus efectos entre sí (baste pensar en una secuencia de genes *BIAS*). Para reducir este efecto no deseado se ha estudiado la posibilidad de controlar la elección de los genes mediante la incorporación de un vector de valores *reguladores* que indiquen con qué probabilidad debe ser seleccionado un tipo de gen frente a otro. En este sentido se requiere un regulador que indique la probabilidad de elección de un gen de tipo *terminal* frente a uno de *estado* cuando el número de nodos hojas es igual a 1, y otro que indique la probabilidad de elección de un gen de *estado* frente a uno *de división*. Para la reducción del efecto no deseado, estos valores deben ser: superior a 50 en el primer caso e inferior a 50 en el segundo (representando %). Este concepto es llevado también a la elección de genes concretos dentro de un mismo tipo, lo cual permitirá realizar un posible estudio de la influencia de unos u otros genes en la obtención de soluciones a determinados problemas.

2.2.5 Los operadores genéticos

La estructuración llevada a cabo en el genoma requiere una modificación en la aplicación de los operadores genéticos, especialmente en la del operador cruzamiento. La identificación de cromosomas con zonas determinadas del fenotipo requiere que la recombinación de dos genomas, llevada a cabo por el operador, se haga necesariamente entre zonas análogas de ambos. Es decir, el intercambio de subárboles debe ser llevado a cabo entre *cromosomas* que cumplan la misma función en ambos genomas.

El operador de *mutación* no se ve afectado por la estructuración del genoma, ya que actúa a nivel de gen y no de árboles de genes, pero resulta conveniente que su acción no altere la probabilidad de aparición de cada uno de los genes definida por el *vector regulador*. La elección del nuevo gen debe seguir, por tanto, el mismo criterio usado durante la generación del árbol genotípico. De esta forma se asegura que la proporción de genes que forman parte de los genomas se mantiene inalterable a lo largo de sucesivas generaciones.

2.2.6 La función de evaluación

El uso de una adecuada función para la evaluación de los individuos generados por el algoritmo genético es fundamental para el correcto funcionamiento del algoritmo en la búsqueda de soluciones. El tipo de problemas que se pretende abordar en esta fase del proyecto es, como ya ha sido explicado, la obtención de redes neuronales que resuelvan problemas lógicos expresados mediante *tablas de verdad*. La valoración de un individuo vendrá dada por el número de “aciertos” conseguidos para cada uno de los patrones de entrada. La función empleada para evaluar los individuos obtenidos es expresada en la ecuación 2.3,

$$fitness = \frac{\sum_{i=1}^{2^k} n_i}{2^k s} 100 \quad (2.3)$$

donde k es el número de bits de entrada del problema, s es el número de bits de salida y n_i es el número de bits coincidentes entre la salida de la red y los bits de salida del problema para el

patrón i . Puede definirse p como el número de patrones que debe resolver la red, valor directamente relacionado con el número de bits de entrada ($p = 2^e$). El valor resultante de la ecuación es el porcentaje de acierto de la red neuronal para la totalidad de los pares de patrones que forman el problema que se desea resolver. El valor máximo posible, obtenido por individuos solución, es 100.

2.3 Políticas de selección

Las políticas de selección y sustitución de individuos empleada en el desarrollo de nuevas generaciones puede ser determinante en la obtención de los resultados en la búsqueda de soluciones a un problema. Éstas determinan que individuos deben ser cruzados o mutados para la generación de nuevos individuos, a quienes deben reemplazar estos últimos y bajo que condiciones se hará. Los métodos de selección y sustitución de individuos aplicados en el presente proyecto para el operador cruzamiento son *roulette*, *isolating by distance* y *crowding*, introducidos en el apartado 1.2.4. El primero ha sido elegido por ser un representante clásico de los métodos de selección para el cruzamiento, ampliamente usado en numerosas aplicaciones de algoritmos genéticos desde su comienzo, caracterizado por dar a cada individuo una probabilidad de ser seleccionado proporcional a su *fitness*. Los métodos *isolating by distance* y *crowding* aportan dos nuevos enfoques en la selección de individuos. El primero introduce el concepto de vecindario, definido como el conjunto de posiciones de una matriz cuadrada accesibles desde una determinada posición con caminos de longitud n . La población se distribuye espacialmente por la matriz cuadrada, existiendo un vecindario distinto por cada posición existente. El cruzamiento queda limitado a pares de individuos seleccionados por ser lo mejor valorados de sendos caminos aleatorios en un mismo vecindario. El método *crowding* selecciona a todos y cada uno de los individuos de la población en cada generación para el emparejamiento, independientemente de la valoración de éstos. Los individuos generados sustituyen a sus progenitores sólo en el caso que mejoren sus valoraciones. Esta política de sustitución, asegura que las valoraciones de los individuos de una población es mayor que las de generaciones anteriores, o en el peor de los casos igual. Esta técnica, en la que padre e hijo compiten entre si para ganar un puesto en la siguiente generación, evita que individuos que manifiestan *disrupción* sustituyan a individuos bien valorados, siendo una forma eficaz de eliminación a posteriori del problema de la *disrupción del cruzamiento*. Esta característica

deseable ha sido añadida también al método de *isolating by distance*, realizándose la sustitución por el nuevo individuo sólo si éste mejora al que ocupa la posición que pretende reemplazar.

En las figuras 2.7, 2.8 y 2.9 se muestran, de forma esquemática, el proceso seguido por cada uno de los métodos de selección y sustitución de individuos comentados.

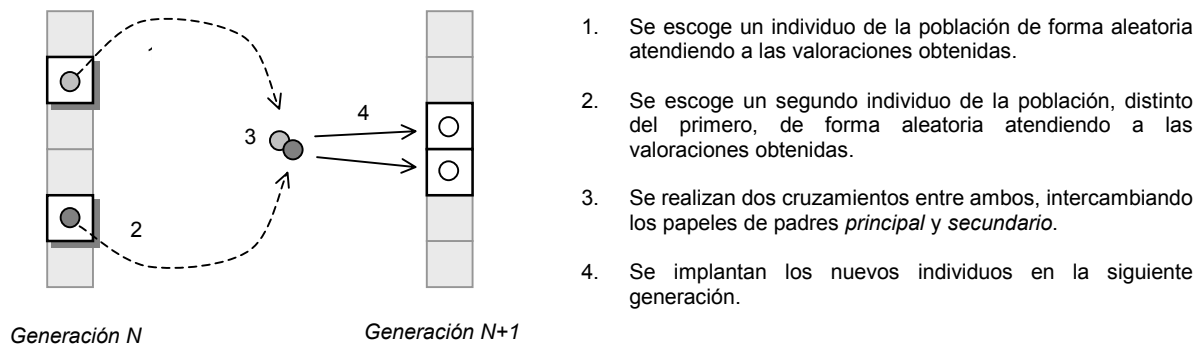


Fig. 2.7: Método de selección de individuos para el cruzamiento "roulette".

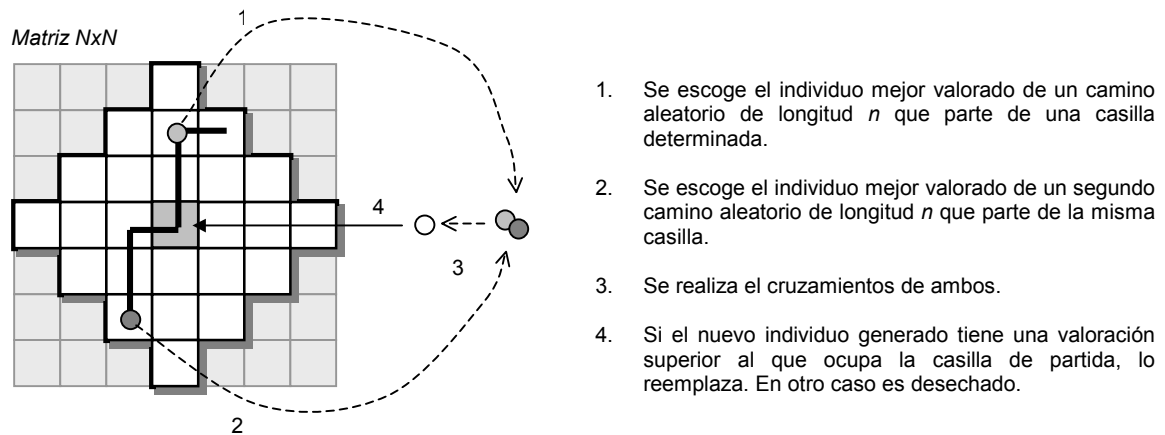


Fig. 2.8: Método de selección de individuos para el cruzamiento "isolating by distance".

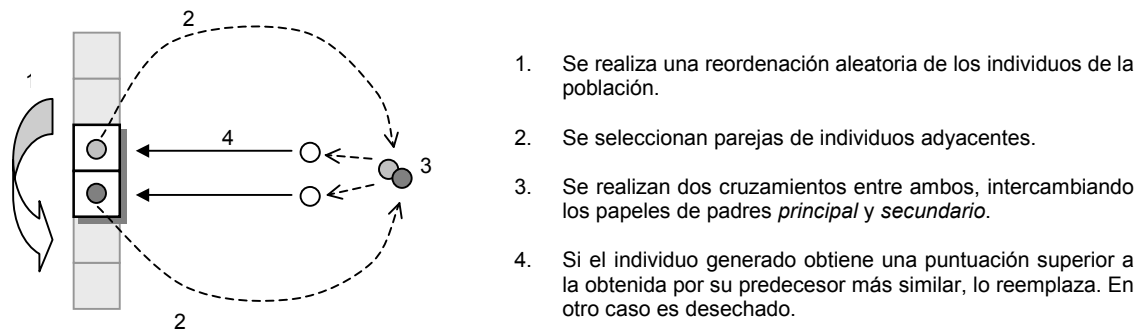


Fig. 2.9: Método de selección de individuos para el cruzamiento "crowding".

Para el operador mutación han sido desarrollados dos nuevos métodos de selección y sustitución de individuos basados en *isolating by distance* y *crowding*. Estos dos métodos, a los que se han denominado respectivamente *caminos aleatorios* y *sustitución positiva*, mantienen una política de sustitución sólo en caso de mejora, es decir, un individuo sólo es sustituido por otro con una puntuación mayor o igual. En las figuras 2.10 y 2.11 se muestran, de forma esquemática, el proceso seguido por los métodos de selección y sustitución de individuos para la mutación.

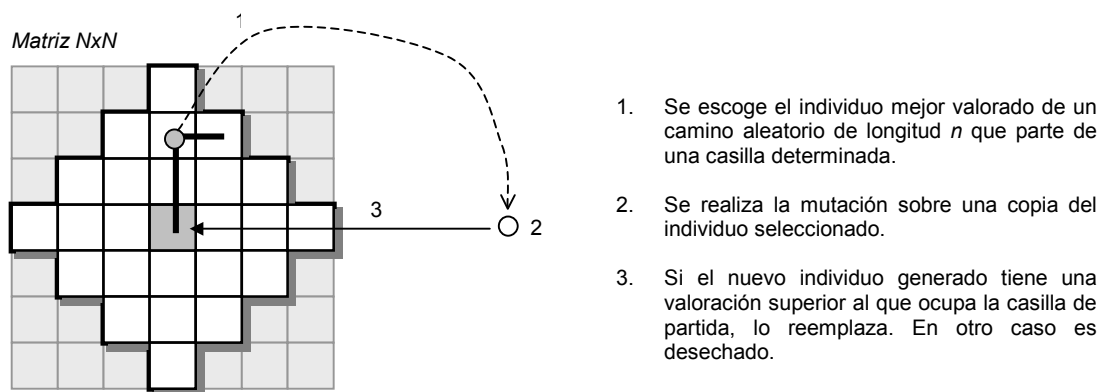


Fig. 2.10: Método de selección de individuos para la mutación "caminos aleatorios".

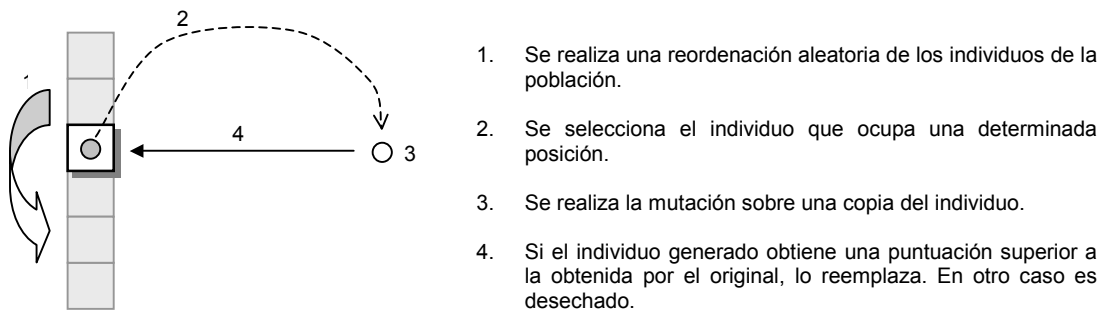


Fig. 2.11: Método de selección de individuos para la mutación “sustitución positiva”.

2.4 Pruebas y resultados

Para la evaluación del algoritmo propuesto con el método de *codificación celular adaptado*, en la búsqueda de soluciones del tipo de problemas planteados, se han realizado repetidas ejecuciones que permiten establecer el comportamiento de las poblaciones ante las diversas políticas de selección y sustitución de individuos propuestas, a lo largo de sucesivas generaciones, de una forma estadística. El primer problema planteado es el denominado “*sumador de dos bits*”, a cuyos patrones de entrada de dos bits (00, 01, 10, 11) de les hace corresponder los patrones de salida de dos bits (00, 01, 01, 10), respectivamente, representando el valor binario de la suma de los bits de entrada. Con este problema J. Koza ilustra su artículo “*Genetic generation of both the weights and architecture for a neural network*” (Koza & Rice, 1991), lo que permitirá hacer un análisis comparativo del rendimiento del algoritmo propuesto con el algoritmo de *Programación Genética*, desarrollado por J. Koza. El siguiente problema estudiado extiende el primero a la función “*sumador de tres bits*” que duplica su complejidad añadiendo un bit más a la entrada, duplicando así el número de patrones de entrada que la red solución debe resolver. En este problema al conjunto de entradas (000, 001, 010, 011, 100, 101, 110, 111) se les hace corresponder las salidas (00, 01, 01, 10, 01, 10, 10, 11). Por último se dan las pautas para la aplicación del algoritmo a la resolución de cualquier otro problema lógico.

Sumador de dos bits

Para abordar el estudio de la búsqueda de solución al problema del *sumador de dos bits* se han realizado conjuntos de 190 ejecuciones con cada uno de los métodos de selección asociados al cruzamiento que han sido descritas en el apartado 2.3. El tamaño de las poblaciones ha sido fijado a los usados por J. Koza en las pruebas descritas en su artículo, esto es poblaciones de 500 individuos y 50 generaciones como máximo. La excepción se produce en la aplicación de *isolating by distance* que requiere poblaciones cuadradas, por lo que en lugar de 500 se ha fijado poblaciones de 484 (22x22) individuos. Las poblaciones iniciales son creadas con individuos aleatorios cuyos fenotipos están formados por un número de entre 10 y 20 neuronas. En los resultados obtenidos por el algoritmo de J. Koza, el 86% de las ejecuciones encuentra una solución en 50 generaciones. La figura 2.12.a refleja la curva de éxito obtenida por J. Koza con la aplicación de su algoritmo en la búsqueda de soluciones.

Para alcanzar una probabilidad de éxito del 99% con cualquiera de los algoritmos aplicados, basta realizar un conjunto de N ejecuciones independientes. La determinación del número de ejecuciones necesarias se lleva a cabo aplicando la ecuación 2.4,

$$N = \left\lceil \frac{\log(1 - P_d)}{\log(1 - P_e)} \right\rceil \quad (2.4)$$

donde P_d es la probabilidad de éxito deseada y P_e la probabilidad de éxito obtenida.

En el caso de los resultados obtenidos por Koza, la probabilidad de éxito del 80% se alcanza en la generación 36. Tomando ésta como probabilidad obtenida ($P_e=0,80$) el número de ejecuciones independientes necesarias para alcanzar una probabilidad deseada del 99% ($P_d=0,99$) es de 3, lo que quiere decir que el 99% de éxito se alcanza con 3 ejecuciones del algoritmo y 36 generaciones como máximo para cada una de ellas. Para hallar el número de individuos desarrollados para alcanzar la probabilidad deseada basta con realizar el producto de los valores del tamaño de la población, el número de generaciones necesitadas y el número de ejecuciones independientes, lo que para los resultados obtenidos da un total de 54.000 (500x36x3) individuos.

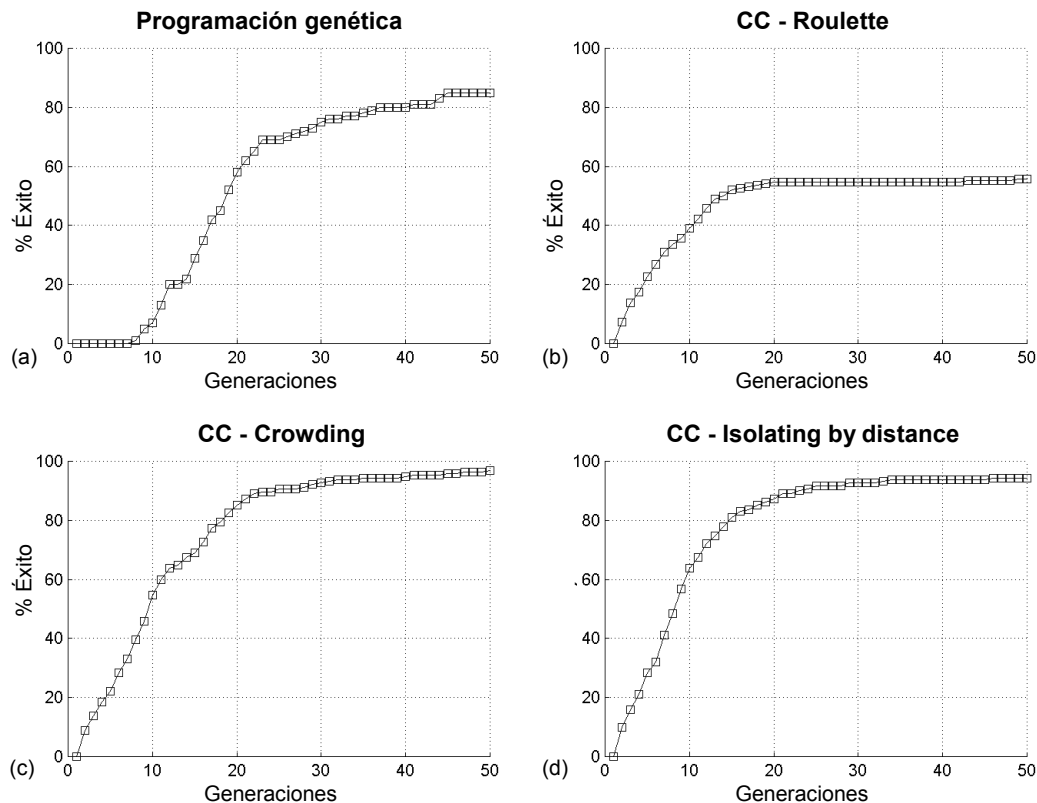


Fig. 2.12. Curvas de probabilidad de éxito en la búsqueda de soluciones al problema del sumador de dos bits a lo largo de 50 generaciones. (a) Algoritmo de programación genética aplicado sobre poblaciones de 500 individuos según los resultados obtenidos por J. Koza (Koza & Rice, 1991). (b) Algoritmo de codificación celular adaptado con el método roulette de selección de individuos para el cruzamiento sobre una población de 500 individuos. (c) Algoritmo de codificación celular adaptado con el método crowding de selección de individuos para el cruzamiento sobre una población de 500 individuos. (d) Algoritmo de codificación celular adaptado con el método isolating by distance de selección de individuos para el cruzamiento, con caminos de longitud 2, sobre una población de 484 (22x22) individuos.

La aplicación del modelo de *codificación celular adaptado*, en la resolución del problema, con el método *roulette* de selección de individuos para el cruzamiento sobre poblaciones de 500 individuos, produce unos resultados inferiores a los obtenidos por J. Koza. Tras 50 generaciones el valor de éxito se establece en un 56%, y con una tendencia a crecer muy limitadamente en generaciones posteriores. La curva mostrada en la figura 2.12.b permite establecer que la aparición de individuos solución se produce en las 20 primeras generaciones. Si una ejecución sobrepasa esta generación sin haber hallado una solución, la probabilidad de que esta se produzca en las siguientes es nula. Este resultado es achacable a la política de

sustitución de individuos llevada a cabo por *roulette*, ya que en ella todos los individuos de una generación son sustituidos sistemáticamente por los generados a partir de ellos mediante cruzamiento, sin que se produzca una evaluación previa de los nuevos individuos. Tomando el 55% como el valor de éxito obtenido en la generación 20, el número de ejecuciones independientes para alcanzar el 99% de éxito queda establecido en 6, y el número de individuos que deben ser desarrollados en 60.000 ($500 \times 20 \times 6$), un 10% más que con la aplicación de *Programación Genética*.

Los resultados obtenidos en la aplicación del método de selección *crowding* producen la curva mostrada en la figura 2.12.c. Tras 50 generaciones el algoritmo alcanza el 97% de éxito. La tendencia ascendente de la curva, en las últimas generaciones, indican el posible aumento del valor de éxito más allá de la generación 50, pudiéndose alcanzar posiblemente el 100% de éxito con sólo aumentar el número de éstas. Tomando el valor del 80% de éxito alcanzado en la generación 18, se concluye que el número de individuos necesitados por el algoritmo para alcanzar el 99% de éxito es de 27.000 ($500 \times 18 \times 3$), lo que representa un 50% menos, de individuos desarrollados que los necesitados por J. Koza.

La curva generada a partir de la aplicación de *isolating by distance* (2.12.d) mantiene una pendiente ascendente más pronunciada que las producidas por los otros métodos, alcanzando el 80% de éxito en tan sólo 15 generaciones, pero a partir de la generación 30 el aumento del valor de éxito es prácticamente nulo, consiguiendo unos valores finales, tras 50 generaciones, ligeramente inferiores a los conseguidos por *crowding*. El número de individuos que es necesario desarrollar para alcanzar el 99% de éxito es por tanto de 21780 ($484 \times 15 \times 3$), un 60% menos que los requeridos por *Programación Genética*.

A la vista de los resultados obtenidos, *crowding* e *isolating by distance* se perfilan como los mejores candidatos entre los métodos de selección y sustitución de individuos, para la resolución del tipo de problema a resolver con el modelo de *codificación celular adaptado*. El siguiente conjunto de pruebas pretende establecer de qué modo afecta la reducción del tamaño de la población al éxito en la obtención de soluciones al problema planteado. Para ello se han realizado ejecuciones con poblaciones de 250 individuos para *crowding* y de 256 para *isolating by distance*. Las curvas de éxito obtenidas se muestran en la figura 2.13.

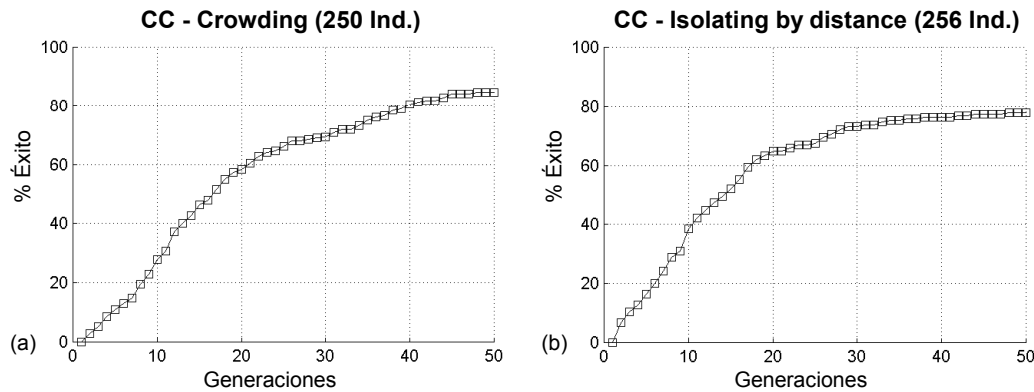


Fig. 2.13. Curvas de probabilidad de éxito en la búsqueda de soluciones al problema del sumador de dos bits a lo largo de 50 generaciones. (a) Algoritmo de codificación celular adaptado con el método crowding de selección de individuos para el cruzamiento sobre una población de 250 individuos. (b) Algoritmo de codificación celular adaptado con el método isolating by distance de selección de individuos para el cruzamiento, con caminos de longitud 2, sobre una población de 256 (16x16) individuos.

Las aplicaciones de ambos métodos alcanzan el 80% de éxito antes de la finalización del algoritmo. En el caso de *crowding* esto se consigue en la generación 40, mientras que en *isolating by distance* se retrasa hasta la generación 50, lo que sitúa el número de individuos necesarios para alcanzar el 99% de éxito en 30.000 (250x40x3) para *crowding* y 38.400 para *isolating by distance*. Estos resultados indican que el segundo es más sensible a la reducción del tamaño de la población, que el primero. La explicación a esto se encuentra en el modo que se distribuye el material genético en la población. El método *isolating by distance* produce la sustitución de individuos por los hijos de los que consiguen mejor puntuación en posiciones cercanas. De esta forma el material genético de los mejores individuos se acaba imponiendo en los alrededores de la posición que ocupan estos, creando una serie de “castas” dominantes en la población. Si el tamaño de la población es bajo, estas castas se imponen sobre toda ella en un menor número de generaciones, produciendo el estancamiento de la población debido a la reducción de la diversidad genética. La diversidad genética de la población se mantiene inalterada, por el contrario, mediante el uso de *crowding*. En éste la sustitución sólo se realiza entre individuos que comparten el mismo material genético, padres e hijos, por lo que la reducción del tamaño de la población no condiciona la obtención de resultados, sólo retrasa el número de generaciones necesarias para ello.

La falta de diversidad genética puede paliarse con la adecuada aplicación del operador *mutación*. Este generador de diversidad puede ser especialmente adecuado para aquellas políticas elitistas que producen un empobrecimiento sistemático de la diversidad, como

isolating by distance. Cada generación es producida por la aplicación de una política de selección de individuos asociado a un operador genético. El uso de *cruzamiento* y *mutación* en una misma ejecución deben ser aplicados, por tanto, en generaciones distintas. El número de generaciones en las que se aplica un operador u otro debe ser indicado explícitamente en la ejecución. La probabilidad usada en las ejecuciones realizadas, donde se combinan ambos operadores (independientemente de la política de selección asociada a cada uno de ellos), es del 80% para el operador *cruzamiento* y del 20% para el operador *mutación*. También se ha fijado para este último, una probabilidad del 10% en la mutación de cada uno de los genes que componen un genoma, para la creación de un nuevo individuo a partir de uno existente.

Han sido realizados varios conjuntos de ejecuciones para el estudio de los efectos de la aplicación combinada de los métodos de selección de individuos para el cruzamiento y la mutación. El método *sustitución positiva* ha sido aplicado junto con *crowding* e *isolating by distance*, mientras que el método de *camino aleatorios* ha sido aplicado únicamente con *isolating by distance*, ya que requiere, al igual que éste, un tamaño de población de $N \times N$ y una distribución espacial de los individuos en una matriz cuadrada. Los resultados obtenidos son mostrados en la figura 2.14.

En la gráfica 2.14.a se representa las curvas correspondientes a las ejecuciones realizadas utilizando el método *crowding* con y sin la aplicación conjunta de *sustitución positiva*. Como puede observarse, la aplicación del operador de mutación no mejora los resultados obtenidos, llegando incluso a retrasar en algunas generaciones el 80% de éxito. De hecho este valor es alcanzado en la generación 45, 5 generaciones más que sin la aplicación de mutación, lo que sitúa el número de individuos necesarios para alcanzar el 99% de éxito en 33.750 ($250 \times 45 \times 3$). Por el contrario, los resultados obtenidos tanto en la aplicación de *sustitución positiva* como de *camino aleatorios* junto a *isolating by distance* (figura 2.14.b) consiguen mejores resultados a los obtenidos sin su aplicación. Los resultados obtenidos en las pruebas realizadas con *sustitución positiva* fijan el 80% de éxito en 30 generaciones, y en 43 las realizadas con *camino aleatorios*. El número de individuos necesarios para alcanzar la probabilidad de éxito deseada es de 23.040 ($256 \times 30 \times 3$) individuos en el primer caso, y de 33.024 ($256 \times 43 \times 3$) individuos en el segundo. Estos resultados confirman lo esperado: la aplicación combinada del operador de mutación con el de cruzamiento, mejoran los resultados de aquellas políticas de selección y sustitución que reducen la diversidad genética.

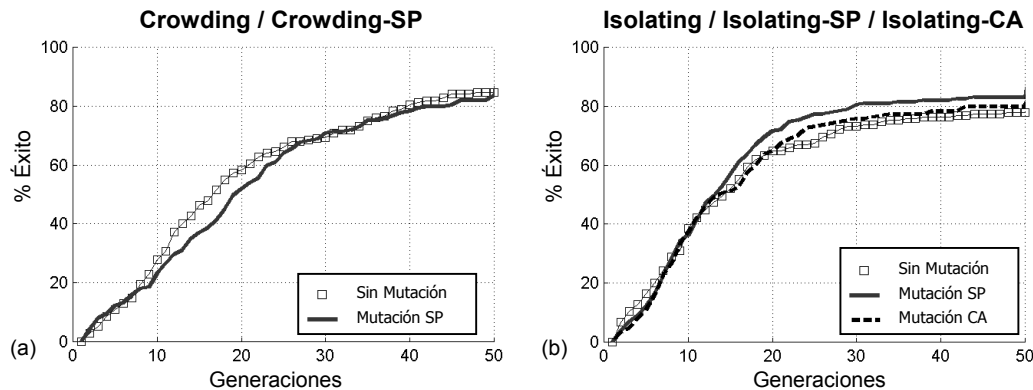


Fig. 2.14. Curvas de probabilidad de éxito en la búsqueda de soluciones al problema del sumador de dos bits a lo largo de 50 generaciones con el uso combinado de los operadores cruzamiento y mutación. (a) Algoritmo de codificación celular adaptado con el método crowding de selección de individuos para el cruzamiento sobre una población de 250 individuos. Uso conjunto del método sustitución positiva asociado a la mutación con una probabilidad de aplicación del 20% sobre el total de las generaciones. (b) Algoritmo de codificación celular adaptado con el método isolating by distance de selección de individuos para el cruzamiento, con caminos de longitud 2, sobre una población de 256 (16x16) individuos. Uso conjunto de los métodos sustitución positiva y caminos aleatorios, asociados a la mutación, con una probabilidad de aplicación del 20% sobre el total de las generaciones.

La tabla 2.1 resume los resultados obtenidos en las pruebas con los diferentes métodos de selección en las búsqueda de soluciones al problema contador de dos bits. Cabe destacar, tras los resultados obtenidos, que la búsqueda de individuos solución no se ha limitado necesariamente a redes neuronales con el número de neuronas prefijadas. Si bien la población inicial está formada por redes con un número de entre 10 y 20 neuronas, la aplicación del operador cruzamiento puede llevar a ciertos individuos de la población fuera de estos límites, permitiendo la búsqueda de soluciones más allá del espacio de búsqueda inicial. Ello asegura que pese a la errónea elección del tamaño de los individuos de la población inicial, el algoritmo pueda extender la búsqueda a topologías con tamaños más adecuados. De cualquier modo, la elección del intervalo de tamaños entre 10 y 20 neuronas ha sido establecido tras pruebas de evaluación previas que permitieron establecer ese intervalo como el más adecuado para la obtención de individuos solución.

Método Selección <i>cruzamiento</i>	Método Selección <i>mutación</i>	Tamaño población	Generación (80% Éxito)	Nº Individuos 99% (% Prog. Gen.)
Roulette	-	500 Ind.	-	60.000 (110%)
Crowding	-	500 Ind.	18	27.000 (50%)
Isolating by distance	-	484 Ind.	15	21.780 (40%)
Crowding	-	250 Ind.	40	30.000 (55%)
Isolating by distance	-	256 Ind.	50	38.000 (70%)
Crowding	Sustitución positiva	250 Ind.	45	33.750 (62.5%)
Isolating by distance	Caminos aleatorios	256 Ind.	43	33.024 (61%)
Isolating by distance	Sustitución positiva	256 Ind.	30	23.040 (42.7%)

Tabla 2.1 Resumen de los resultados obtenidos en las pruebas realizadas sobre el modelo de codificación celular adaptado en la búsqueda de soluciones al problema contador de dos bits, con los distintos métodos de selección propuestos. La última columna muestra el número de individuos necesarios para conseguir el 99% de éxito en cada método, siendo relacionados con los 54.000 individuos necesitados por el algoritmo programación genética según las pruebas realizadas por J. Koza. Han sido destacados los tres mejores resultados conseguidos, correspondientes a crowding, isolating by distance e isolating by distance con sustitución positiva.

Como ejemplo de individuo solución, se muestra en la figura 2.15 uno de los obtenidos en las pruebas realizadas sobre el algoritmo genético aplicado al problema del “Sumador de 2 bits”. El individuo es expresado a través de su genoma, dividido en tres cromosomas, que representa de forma codificada a la red neuronal. Tras el proceso de desarrollo embrionario, la interpretación del genoma da lugar al fenotipo, o lo que es lo mismo la red de neuronas binarias, que resuelve el problema planteado. En este caso la red está formada por 13 neuronas, dos de las cuales pertenecen a la capa de entrada, dos a la capa de salida y el resto a la capa oculta. La matriz de conexionado de la red neuronal, los umbrales de cada neurona y la capa a la que pertenecen son mostrados en la figura.

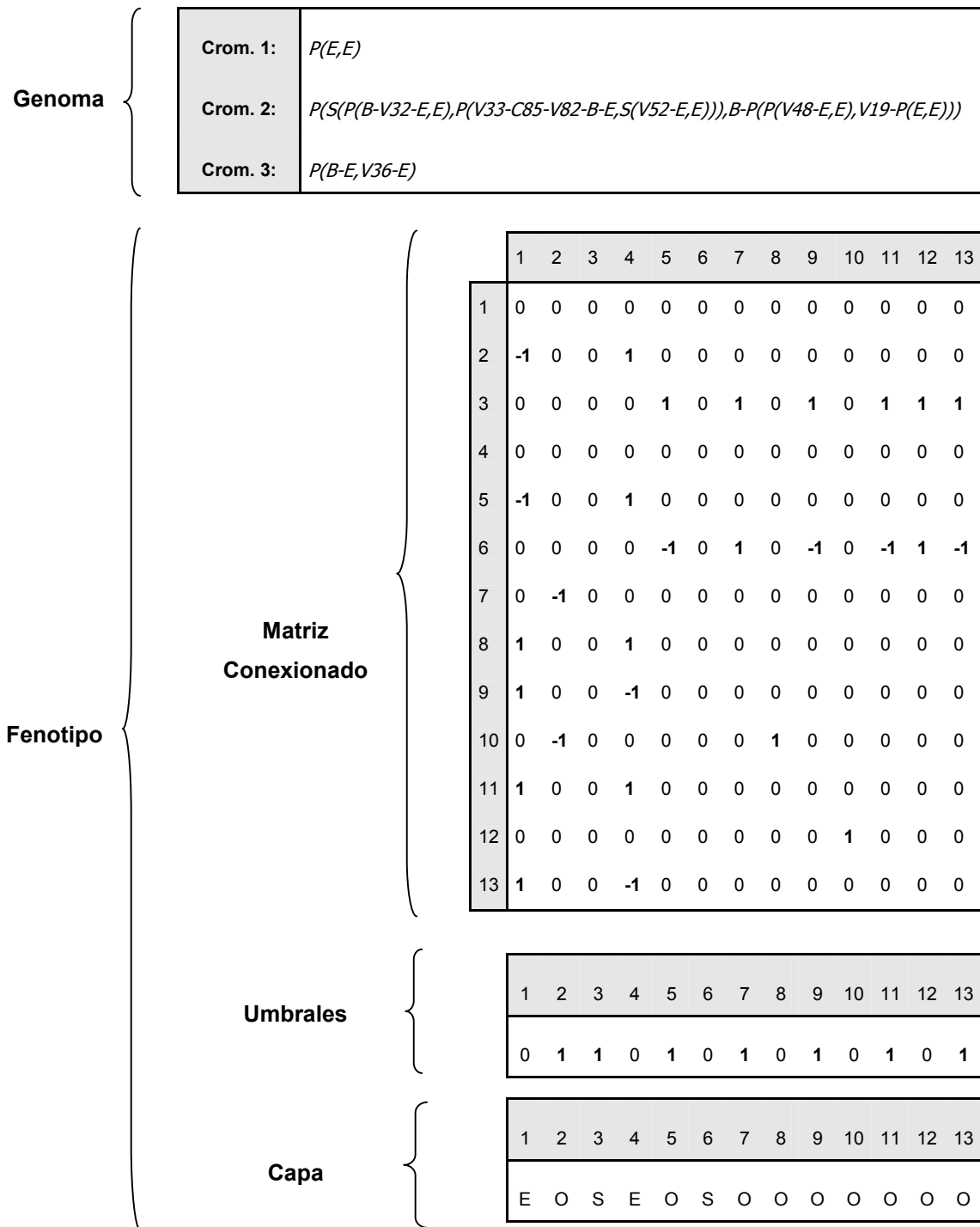


Fig. 2.15 Individuo solución al problema Contador de dos bits. Los nombres de los genes que constituyen el genoma han sido abreviados: PAR=P, SEQ=S, BIOS=B, CUT=C y VAL=V. De igual forma las capas a las que pertenecen las distintas neuronas han sido abreviadas: Entrada=E, Salida=S y Oculta=O.

Sumador de tres bits

Para estudiar la escalabilidad del algoritmo en la búsqueda de soluciones a problemas más complejos se ha extendido el problema anterior a la suma de un tercer bit. De este modo el número de patrones a resolver por la red, pasa de 4 a 8, duplicando con ello la complejidad del mismo. Al conjunto de patrones de entradas (000, 001, 010, 011, 100, 101, 110, 111) se les hace corresponder las salidas (00, 01, 01, 10, 01, 10, 10, 11), representando con ello la suma del número de bits de la entrada a '1'. Los métodos seleccionados para la búsqueda de la solución a este problema han sido los que mejores resultados obtuvieron en el conjunto de pruebas anteriores, *crowding* (sin *mutación*), *isolating by distance* (sin *mutación*) e *isolating by distance* con *sustitución positiva*. Dado la mayor complejidad del problema, se ha dotado del doble de neuronas a las redes neuronales que forman las poblaciones iniciales, siendo su tamaño de entre 20 y 40 neuronas. Las poblaciones se han establecido en 500 para *crowding* y 484 para *isolating by distance*, permitiendo la suficiente variedad genética inicial para una óptima búsqueda de soluciones. Se han realizado para cada uno de los métodos 190 ejecuciones independientes, con un número de generaciones máximo de 500, para permitir analizar la evolución de la búsqueda de soluciones más allá de las 50 generaciones estudiadas en el problema anterior. La figura 2.16 muestra las curvas de éxito obtenidas en los tres conjuntos de pruebas.

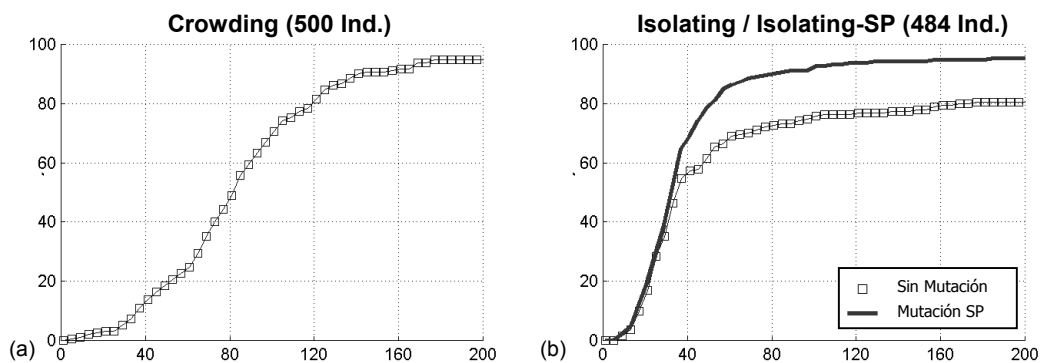


Fig. 2.16. Curvas de probabilidad de éxito en la búsqueda de soluciones al problema del sumador de tres bits a lo largo de 200 generaciones. (a) Algoritmo de codificación celular adaptado con el método *crowding* de selección de individuos para el cruzamiento sobre una población de 500 individuos. (b) Algoritmo de codificación celular adaptado con el método *isolating by distance* de selección de individuos para el cruzamiento, con caminos de longitud 2, sobre una población de 484 (22x22) individuos. Uso conjunto del método *sustitución positiva*, asociado a la *mutación*, con una probabilidad de aplicación del 20% sobre el total de las generaciones.

En los tres conjuntos de ejecuciones el 80% de éxito es alcanzado antes de la generación 200. Las pruebas llevadas a cabo con *crowding*, cuya curva de éxito se muestra en la figura 2.16.a, fijan el 80% de éxito en torno a la generación 120. Esto supone que el número de individuos generados necesarios para asegurar el 99% de éxito es de 180.000 ($500 \times 120 \times 3$). Cabe destacar que el 100% de las ejecuciones realizadas con este método han encontrado individuos solución antes de alcanzar la generación máxima, establecida en 500.

La aplicación de *isolating by distance* sin mutación produce mejores resultados que *crowding* en las 110 primeras generaciones, reduciéndose en gran medida su efectividad a partir de ese punto. El 80% de éxito es alcanzado en la generación 168, lo que implica un número de individuos necesarios de 243.936 ($484 \times 168 \times 3$) para alcanzar el 99% de éxito. Tras 500 generaciones, sólo el 83% de las ejecuciones han encontrado una solución. La aplicación conjunta del método de selección de individuos para la mutación *sustitución positiva* e *isolating by distance* produce una espectacular mejora de los resultados con respecto a los obtenidos por este último en solitario. El 80% de éxito es alcanzado, en este conjunto de ejecuciones, en tan sólo 52 generaciones, situando el número de individuos generados, necesarios para alcanzar la probabilidad de éxito deseada, en 75.504. Sólo el 2% de las ejecuciones realizadas con el uso combinado de estos métodos no ha encontrado una solución al problema en 500 generaciones. Las curvas de éxito correspondiente a los resultados obtenidos con la aplicación de *isolating by distance* con y sin *sustitución positiva* son mostradas en la figura 2.16.b.

Como conclusiones se puede extraer que los métodos *crowding* e *isolating by distance* son dos excelentes métodos de búsqueda aplicados sobre el modelo de *codificación celular adaptado*. El primero se perfila como un “gran corredor de distancias largas”, ya que asegura la obtención de soluciones con sólo ampliar el número de generaciones, es por tanto un método exhaustivo para la búsqueda de soluciones. *isolating by distance* es, por el contrario, un gran “*sprinter*”. La probabilidad de éxito se dispara en las primeras generaciones perdiendo, de forma considerable su eficacia, cuando se alcanza cierto número de generaciones. El estancamiento es tal que resulta inútil la ampliación del número de generaciones para aumentar la probabilidad de éxito. Este efecto ya conocido, achacable a la pérdida de diversidad genética en la población, queda prácticamente anulado con la aplicación del método *sustitución positiva*, que convierte al combinado en el campeón en todas las distancias, ya que prácticamente iguala a cada uno de los métodos anteriores en su propio campo. El método *sustitución positiva* se perfila como un complemento ideal para la política elitista de *isolating by distance*.

La tabla 2.2 resume los resultados obtenidos en las pruebas con los diferentes métodos de selección en las búsqueda de soluciones al problema *contador de tres bits*.

Método Selección <i>cruzamiento</i>	Método Selección <i>mutación</i>	Tamaño población	Generación (80% Éxito)	Nº Individuos 99% Éxito
Crowding	-	500 Ind.	120	180.000
Isolating by distance	-	484 Ind.	168	243.936
Isolating by distance	Sustitución positiva	484 Ind.	52	75.504

Tabla 2.2 Resultados obtenidos en las pruebas realizadas sobre el modelo de codificación celular adaptado en la búsqueda de soluciones al problema Contador de tres bits, con los distintos métodos de selección propuestos. La última columna muestra el número de individuos necesarios para conseguir el 99% de éxito en cada método. Ha sido destacado el mejor resultado conseguido, correspondientes a la aplicación de isolating by distance con sustitución positiva.

En la figura 2.17 se muestra el genoma codificado de uno de los individuos solución obtenidos, que resuelven el problema del contador de 3 bits, formado por 20 neuronas.

Crom. 1:	$P(E, P(E, E))$
Crom. 2:	$P(C75-P(B-P(P(E, E), P(E, V19-E))), P(S(B-E, E), P(E, C75-B-E))), P(V51-P(E, V48-B-P(E, E)), B-P(P(E, E), P(C40-E, E))))$
Crom. 3:	$B-P(E, V86-B-E)$

Fig. 2.17 Genoma de un individuo solución al problema Contador de dos bits. Los nombres de los genes que constituyen el genoma han sido abreviados: PAR=P, SEQ=S, BIOS=B, CUT=C y VAL=V.

Aplicación a otros problemas

La aplicación del algoritmo genético para la resolución de cualquier otro problema lógico requiere del estudio previo del tamaño óptimo de los individuos, así como del número de ellos que deben formar la población, de forma que se asegure la diversidad genética necesaria. El aumento de la complejidad del problema, relacionado directamente con el número de patrones que incluye el problema y el tamaño de éstos, debe ir acompañado de un aumento del número

de neuronas que forman las redes neuronales. De igual forma, dicho aumento de tamaño de los fenotipos implica un mayor espacio de búsqueda, y con ello, la necesidad de aumentar el número de individuos que forman la población para asegurar la diversidad necesaria.

Capítulo 3

Codificación celular adaptado a redes de neuronas de integración y disparo

Los resultados obtenidos con el modelo *codificación celular adaptado* para la optimización de redes de neuronas binarias demuestran el potencial del método desarrollado para nuevas aplicaciones. El siguiente objetivo consiste en la adaptación del modelo para el desarrollo de redes neuronales de integración y disparo, cuyo modelo es descrito en el apartado 1.1.5. Su aplicación se realizará para la búsqueda de topologías de redes neuronales cuyo conexionado represente una alternativa al aportado por un experimentador. Con la aplicación del algoritmo se pretende concluir si es posible encontrar topologías alternativas, para el modelo neuronal dado, mediante la búsqueda exhaustiva de individuos candidatos en el espacio de búsqueda. Para ello se evaluarán los resultados obtenidos en la aplicación de los diversos métodos de selección para los operadores de cruzamiento y mutación ya usados, a excepción del método *roulette* que demostró ser ineficaz en las pruebas realizadas sobre el modelo binario.

La totalidad de las pruebas realizadas sobre el algoritmo genético se centrarán en la búsqueda de conexionados alternativos al *monosináptico*. Este conexionado es el más simple que puede modelarse, ya que consiste en dos neuronas conectadas por un único enlace sináptico, y exhibe un *correlograma cruzado* muy característico que es identificado directamente con él.

La relativa complejidad del modelo neuronal para la que va a ser adaptado el modelo de codificación, obliga a realizar un detallado estudio de la influencia de los valores de los parámetros en la actividad presentada por las neuronas, para adecuarlos al comportamiento deseado. En primer lugar, se presenta dicho estudio que permitirá una adaptación óptima del modelo de codificación. Los siguientes apartados describen dichas adaptaciones, el conjunto de pruebas realizado sobre el algoritmo genético y los resultados obtenidos.

3.1 Estudio del modelo de neurona de integración y disparo

El modelo neuronal de integración y disparo adaptado en el presente trabajo es el introducido en el apartado 1.1.5 y definido en las ecuaciones 1.1, 1.2 y 1.3. El conjunto de valores de los parámetros que rige la actividad neuronal debe estar acotado en unos rangos que aseguren un comportamiento similar al observado en los sistemas neuronales biológicos que se pretende simular. Ello hace necesario un estudio de los parámetros que caracterizan el modelo, para que el modelo reproduzca el comportamiento deseado.

Los estudios neurofisiológicos han fijado algunos de los valores de los parámetros, como pueden ser: el *potencial de reposo*, fijado en -70 mV y el *potencial de disparo*, fijado en -40 mV. La *caída del potencial de la membrana celular* es fijada en 0,1.

Los valores que debe tomar la señal de ruido ($r_i[t]$) y el conjunto de pesos sinápticos (W_{ij}) deberán ajustarse a partir de la experiencia empírica sobre el modelo neuronal. La necesidad de realizar el estudio viene dada por la importancia de estos parámetros en la dinámica de la actividad neuronal. Los estudios desarrollados a continuación permiten conocer los valores adecuados para estos parámetros y de que modo afectan modificaciones en la actividad global de la red. Este conocimiento permitirá una correcta adaptación del modelo de codificación genética para el desarrollo de redes de neuronas que ofrezcan comportamientos adecuados. En primer lugar se realiza un estudio sobre la actividad neuronal, en ausencia de señales procedentes de otras neuronas, debida al *ruido*. Este estudio se realiza para dos modelos diferentes de ruido: *uniforme* y *gaussiano*. El segundo estudio analiza principalmente la influencia de la actividad de una neurona sobre la de otra a la que se encuentra conectada dependiendo del *peso* del enlace.

3.1.1 El ruido

El ruido influye en la actividad de las neuronas incluso en ausencia de otras entradas. Todas las neuronas poseen una actividad de baja frecuencia debida a este factor. En el modelo usado en el presente proyecto, las neuronas con un mayor nivel de ruido, y por ello una mayor actividad, son principalmente las iniciadoras de la actividad en los circuitos neuronales. Este tipo de neuronas son denominadas neuronas *driver*.

El modelo de distribución *gaussiano*, es el que se puede encontrar en cualquier aspecto de la naturaleza, y por ello el más empleado en sistemas que pretenden modelar mecanismos naturales. Es a priori, la distribución que deberían seguir los valores de ruido empleado en el presente modelo neuronal, pero su alto coste computacional hace plantear la posibilidad de usar otro modelo de distribución computacionalmente menos pesado, que consiga reproducir los mismos resultados a un menor coste.

En el presente trabajo se han estudiado el modelo de *ruido gaussiano* y el modelo de *ruido uniforme*, más eficiente computacionalmente. El modelo de *ruido gaussiano* se caracteriza por la distribución en forma de campana de Gauss que sigue la función y cuya expresión analítica está definida por la ecuación 3.1,

$$y = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad (3.1)$$

siendo μ la media de la distribución y σ la desviación típica de la misma. La distribución seguida por el modelo de *ruido uniforme* está definida por un intervalo $[\lambda_{min}, \lambda_{max}]$ donde todos los valores son equiprobables. Para permitir un mayor acercamiento entre los dos modelos puede redefinirse el intervalo a partir de los parámetros μ (media) y r (radio), donde $\lambda_{min}=\mu-r$ y $\lambda_{max}=\mu+r$, siendo el intervalo resultante $[\mu-r, \mu+r]$.

En la figura 3.1 se pueden observar ambas distribuciones y los parámetros que las definen.

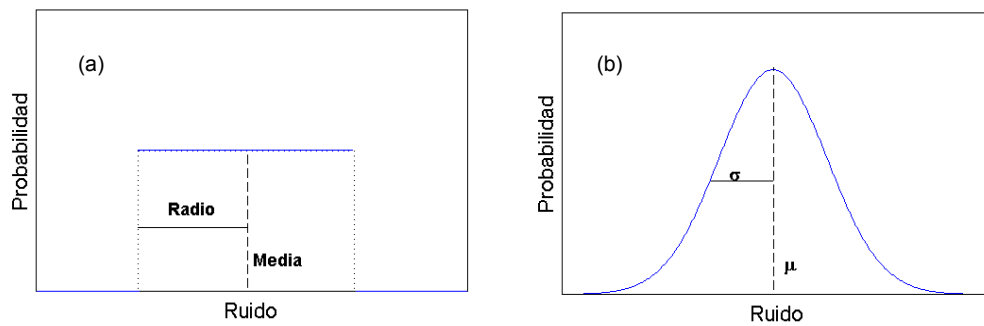


Fig. 3.1. Distribuciones del ruido. (a) Distribución uniforme del ruido con los parámetros media y radio para definir el intervalo. (b) Distribución Gaussiana del ruido con los parámetros media (μ) y desviación típica (σ).

Para estudiar la influencia que la forma del ruido tiene sobre la actividad neuronal se han realizado una serie de pruebas donde se ha hecho variar la media del ruido, para comprobar la forma en que es afectada la frecuencia de disparo de la neurona. Estas pruebas se han realizado tanto en el modelo uniforme como en el gaussiano, para un conjunto determinado de valores de radio y desviación típica, respectivamente. La media del ruido se ha hecho variar entre 2mV y 4mV, en ambos modelos, para valores de 0.3, 0.5, 0.7 y 0.9 de radio y desviación típica, respectivamente (Fig. 3.2.a y 3.2.b). El objetivo es estudiar si es posible encontrar parámetros que reproduzcan el mismo comportamiento en la actividad neuronal para ambos modelos con una mínima diferencia. En la figura 3.2 puede observarse las distintas distribuciones que son usadas en las pruebas y los resultados que producen en la actividad neuronal, medida ésta en frecuencia de disparo (s^{-1}).

Como puede observarse en las gráficas 3.2.c y 3.2.d, para frecuencias superiores a 40 disparos por segundo, la forma de la distribución del ruido no afecta de forma determinante, siendo la media del ruido el único valor que determina la frecuencia. Dicha frecuencia es alcanzada cuando la media de ambos modelos alcanza los 3,4 mV, siendo el valor a partir del cual convergen todas las curvas. Las diferencias entre las curvas de las distintas distribuciones sólo se hacen evidentes en frecuencias inferiores a 40 disparos por segundo.

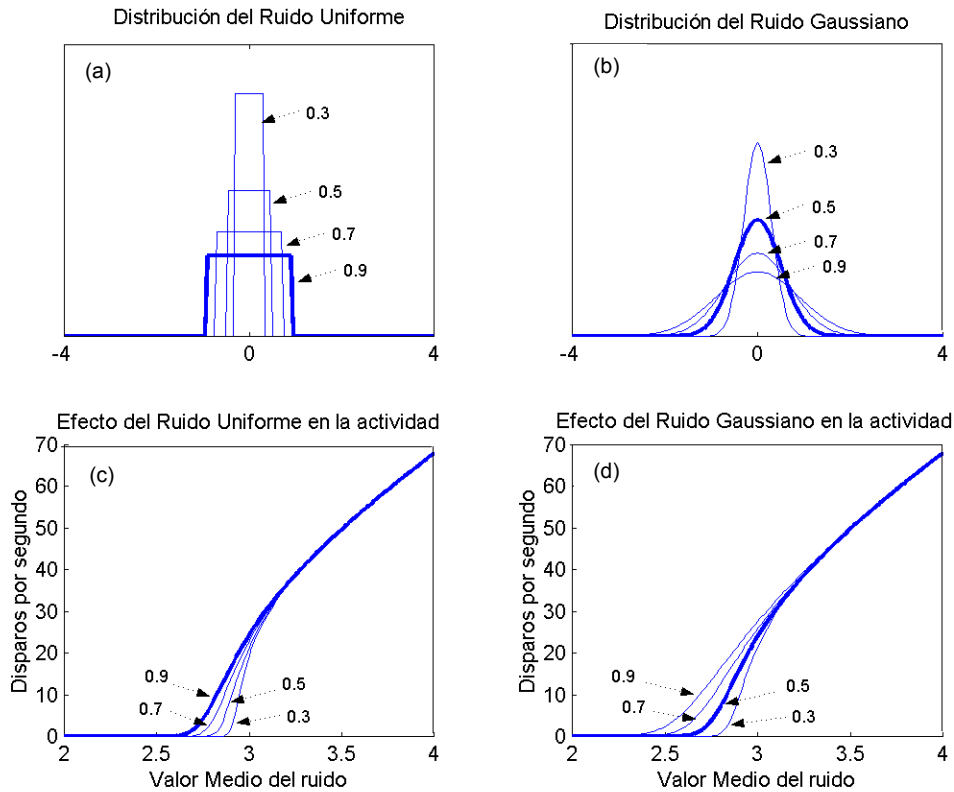


Fig. 3.2 (a) Función de densidad de probabilidad uniforme del ruido con los parámetros media a 0 y radio a 0,3, 0,5, 0,7 y 0,9. (b) Función de densidad de probabilidad gaussiana del ruido con los parámetros μ (media) a 0 y σ (desviación típica) a 0,3, 0,5, 0,7 y 0,9. (c) Efecto del ruido uniforme en el número de disparos por segundo de la neurona con una media variable entre 2 mV y 4 mV. (d) Efecto del ruido gaussiano en el número de disparos por segundo de la neurona con una media variable entre 2 mV y 4 mV. Se han rotulado con un trazo más grueso en las gráficas a y b aquellos valores donde las distribuciones generan prácticamente la misma curva, esto es un valor de 0,9 en el radio de la distribución uniforme y 0,5 en la desviación típica de la distribución gaussiana.

La cuestión que se planteaba antes de la realización de las pruebas era si es posible encontrar parámetros para los cuales el efecto del ruido fuera el mismo para ambos modelos. Por ello el estudio se ha centrado en encontrar un valor de radio en el modelo uniforme cuyo efecto en la actividad neuronal sea indistinguible del que se produce en el modelo gaussiano para un valor de desviación típica igual a 0,5 mV. En las pruebas realizadas, la curva del modelo uniforme que más se ajusta es la que corresponde a un valor de radio de 0,9 mV. Un segundo grupo de pruebas se ha realizado para ajustar el valor de radio minimizando la distancia entre ambas curvas, con valores de radio en el rango comprendidos entre 0,8 mV y 1 mV, y pasos de 0,01 mV. Tras la evaluación de los resultados el valor para el que la diferencia media es mínima se ha establecido en 0,85 mV. El error medio cometido entre ambas curvas se establece en 0,076 disparos, prácticamente despreciable. De esta forma podemos concluir

que un valor de radio de 0,85 mV en el modelo uniforme genera la misma actividad que un valor de 0,5 mV de desviación típica en el modelo gaussiano para los mismos valores de media.

El siguiente paso es determinar los valores de media que van a ser usados en las futuras pruebas, atendiendo a las frecuencias deseadas que deben presentar las neuronas. Los circuitos neuronales que serán usados en las pruebas dispondrán de dos tipos de neuronas con un nivel de actividad diferente. Un grupo de neuronas, que recibirán el nombre de *neuronas driver*, tendrán una actividad superior a un segundo grupo, denominadas *neuronas no driver*. Las neuronas del primer tipo serán, generalmente, las iniciadoras de la actividad en los circuitos neuronales que serán desarrollados. El número de disparos deseables para las neuronas *driver* estará en torno a los 15 disparos por segundo. En el caso de neuronas *no driver* su actividad es reducida a 5 disparos por segundo. Los valores de media que serán usados para obtener las frecuencias de disparos deseadas, a partir del valor de radio del intervalo de 0,85 mV, son de 2,89 mV para neuronas *driver* y de 2,77 mV para neuronas *no driver*. Hay que recordar que la actividad fijada es en ausencia total de entradas y, en los circuitos neuronales que serán desarrollados, la actividad de estas neuronas aumentará o se reducirá en relación a la influencia externa que posean. La figura 3,3 muestra la relación entre el valor de la media del intervalo de ruido de una neurona con ruido uniforme para un radio de 0,85 mV y el número de disparos por segundo que produce.

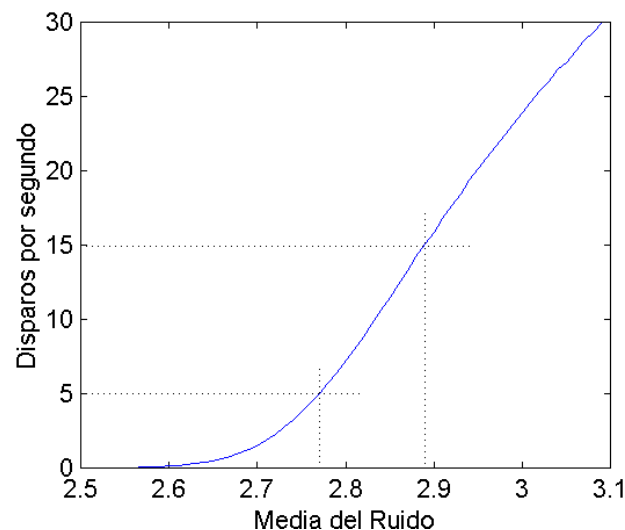


Fig. 3.3. Efecto de la media del ruido en la frecuencia de disparo. Para frecuencias de 15 y 5 disparos por segundo son requeridos valores de media establecidos en 2,89 mV y 2,77 mV, respectivamente, para un modelo de ruido gaussiano con un valor de desviación típica de 0,5, o un modelo de ruido uniforme con un radio de 0,85.

Las pruebas realizadas sobre el modelo de neurona de integración y disparo usaran, a partir de ahora, el modelo de ruido uniforme con los parámetros estudiados. La figura 3.4 muestra el comportamiento de una neurona *driver* en ausencia de entradas durante 400 milisegundos.

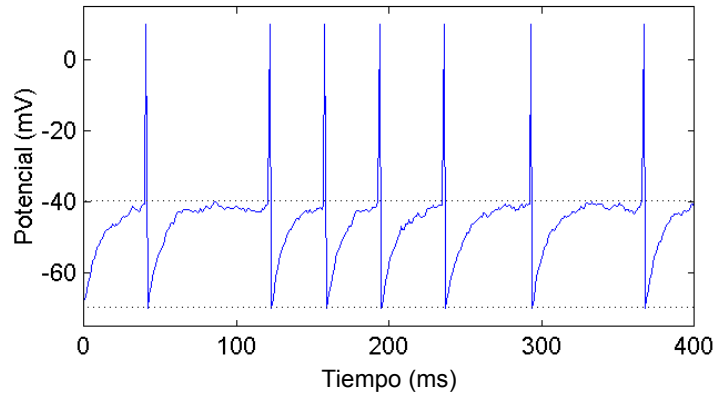


Fig. 3.4. Efecto del ruido uniforme en el potencial de membrana de una neurona con una actividad de 15 s^{-1} en ausencia de entradas. La media del ruido para el cual se muestra la actividad es de $2,89 \text{ mV}$ y su radio de intervalo de $0,5 \text{ mV}$. El valor del potencial de reposo es fijado en -70 mV y el del potencial de disparo en -40 mV .

3.1.2 La influencia del peso sináptico en la actividad neuronal

Para medir la influencia del peso de las conexiones sobre la actividad neuronal, se recurre a los conceptos de *eficacia* y *contribución sináptica*. La eficacia sináptica, mide el número de disparos de la neurona postsináptica, a la que denominamos $N2$, producidos a continuación de disparos de la neurona presináptica, que denominamos $N1$, en relación con el número total de disparos producidos por esta última. Con esto se pretende medir que número de disparos de $N2$ que son producidos directamente por la actividad de $N1$. El concepto de contribución sináptica es análogo al anterior pero desde el punto de vista de la neurona postsináptica, esto es el número de disparos de $N2$ producidos a continuación de disparos de $N1$ en relación con el número total de disparos producidos por la primera.

Para el estudio de la eficacia y la contribución sináptica en ambos modelos se ha diseñado una red con dos únicas neuronas y una única conexión entre ellas (conexión *monosináptica*). La neurona presináptica posee una actividad de ruido de 15 s^{-1} (*driver*) y la postsináptica de 5 s^{-1} (*no driver*). Los valores de los parámetros son de: 0,5 en la desviación típica del modelo gaussiano y 0,85 en el radio del modelo uniforme para ambas neuronas, 2,89 en la media del ruido de $N1$ y 2,77 en la media del ruido de $N2$ para ambos modelos.

Las simulaciones se han realizado para pesos comprendidos entre 0 y 10 y como condiciones de parada para cada simulación 20.000 disparos de $N2$ o 10.000.000 pasos de ejecución (10.000 segundos). Las gráficas obtenidas a partir de los resultados se muestran en la figura 3.5.

Es conveniente fijar unos límites a la máxima eficacia y contribución permitida en un circuito neuronal, de forma que no existan neuronas demasiado "influyentes", ni "influenciables". Unos valores máximos que parecen adecuados son un 7% de eficacia y un 16% de contribución, esto es que la neurona presináptica produce un aumento máximo del 16% en la actividad de la neurona postsináptica equivalente al 7% de su propia actividad. Estos valores de eficacia y contribución se alcanzan con un peso sináptico de 1. Por tanto el rango de valores que podrá tomar el peso de una conexión sináptica se encontrará entre 0 y 1.

Los resultados de las pruebas sobre la eficacia y la contribución confirman que ambos modelos de ruido, producen el mismo resultado en la actividad neuronal para los valores estudiados. Con esto se extiende la equivalencia de los valores de los modelos de ruido a nuevos aspectos, lo que permitirá extrapolar, en cierta medida, los resultados obtenidos con uno de los dos modelos para unos valores determinados, al otro modelo para valores equivalentes.

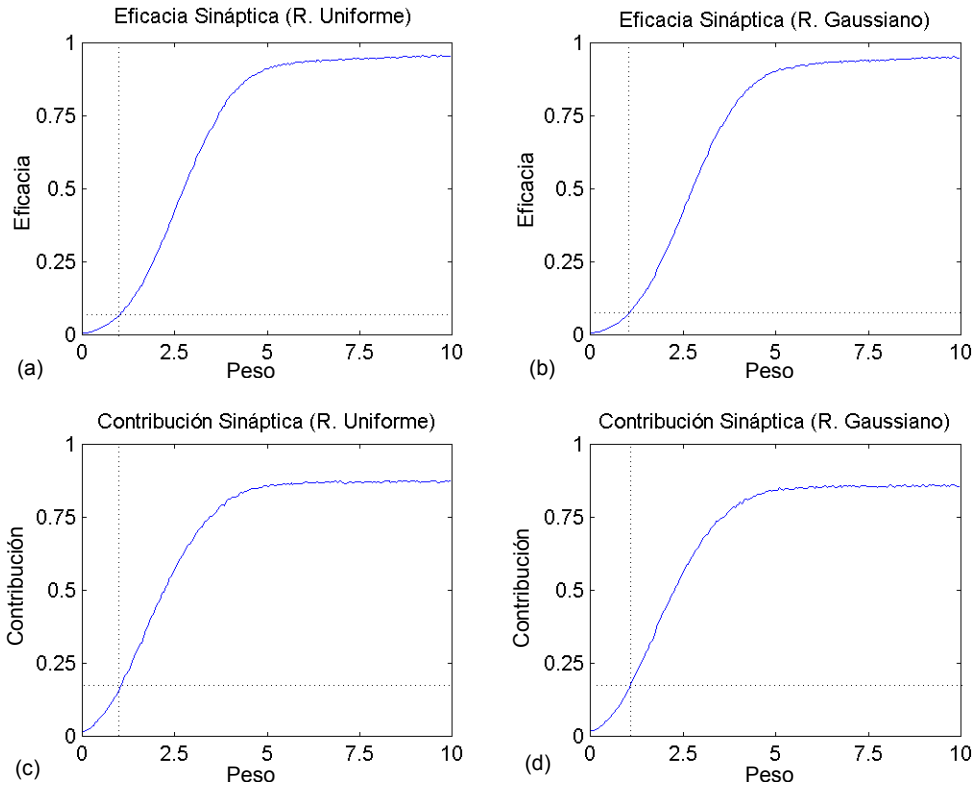


Fig. 3.5. Las gráficas (a) y (b) muestran los resultados de la eficacia sináptica para los modelos con ruido uniforme y gaussiano respectivamente para los valores estudiados. Las gráficas (c) y (d) muestran la contribución sináptica en ambos modelos. Para un peso de 1 la eficacia queda fijada en torno al 7% y la contribución al 16% en ambos modelos de ruido lo que confirma la equivalencia para los parámetros estudiados.

3.2 Adaptación del algoritmo genético

El estudio del conjunto de parámetros, que definen la dinámica del modelo de neuronal de integración y disparo empleado, a permitido establecer el conjunto de adaptaciones que son necesarias aplicar, sobre el modelo de codificación, para permitir un modelado completo y óptimo de las redes neuronales a desarrollar. Las modificaciones necesarias se centran principalmente en el conjunto de genes, dado que éstos son los principales responsables de la descripción de la futura red neuronal. Por otro lado, la estructuración del genoma en tres cromosomas, propuesta e implementada en el modelo binario, se presenta también como una

buena solución para el caso biológico, aunque la identificación *cromosoma-capas* no es aplicable en este último, dado que las redes desarrolladas carecen de una estructura definida en capas. Todos estos aspectos son desarrollados en los siguientes apartados.

3.2.1 El conjunto de genes

Como cabe esperar, la implementación de un nuevo modelo neuronal, por parte del modelo de codificación, implica una revisión profunda en el conjunto de genes. El modelo de codificación tomado de partida es el adaptado y desarrollado en el apartado 2.2. Si bien la clasificación en tres tipos de genes (de división, de estado y terminales) permanece inalterada, se hace necesaria la modificación de las instrucciones que representan, así como la eliminación e inserción de determinados genes. La alteración del conjunto de parámetros internos de la neurona implica modificaciones en los genes de estado, así como la modificación de la topología de las redes neuronales que se pretende desarrollar implica la modificación del conjunto de genes de división.

Uno de los parámetros más importantes en el modelo de neurona lógico es el *Umbral* de activación de la neurona, que toma los posibles valores 0 y 1, y es controlado por el gen *BIOS*. Este parámetro variable desaparece en el modelo de neurona de integración y disparo, y es sustituido por un parámetro constante que recibe el nombre de *umbral de disparo* (Φ), lo que hace innecesaria la existencia de dicho gen en el nuevo conjunto de genes.

Los genes de división empleados en el modelo binario, *SEQ* y *PAR* permiten la generación de redes neuronales con una estructura en capas donde son evitados conexiones cíclicas. En el modelo de integración y disparo, por el contrario, una neurona puede estar conectada a cualquier otra que forme parte del circuito neuronal, sin establecerse una ordenación entre ellas. Esto hace necesario la inserción de un nuevo gen que contribuya a generar las topologías que son excluidas por el único uso de *SEQ* y *PAR*.

Una nueva característica del modelo neuronal impone la obligación de que el signo de todas las conexiones de salida de una neurona debe ser el mismo, de forma que si estas son positivas la neurona se considera *excitadora*, y si son negativas *inhibidora*. Para modelar esta característica, se ha preferido considerar pesos sin signo en todo el proceso de desarrollo embrionario, y sólo al final de éste es impuesto un signo positivo o negativo en las conexiones

de salida de la neurona dependiendo el tipo de neurona que represente. El gen *VAL* trabajará, por tanto, sobre pesos sin signo, siendo el gen terminal *END* el que asumirá, en el nuevo conjunto de genes, la decisión del signo de las conexiones de salida de la neurona (decidiendo con ello el tipo de neurona que representa).

A continuación se describe las modificaciones sobre el conjunto de genes que han sido necesarias para el desarrollo de redes de neuronas biológicas, clasificado en tres grupos según la función que cumple el gen durante el desarrollo embrionario:

- **Genes de división.** En una división secuencial (implementada por un gen denotado *SEQ*) la primera célula hija hereda los enlaces de entrada de la célula original y la segunda célula los enlaces de salida. La primera hija es conectada a la segunda con un peso cuyo valor es la media entre el peso máximo y mínimo permitido para la simulación. De igual forma en la división paralela (implementada por un gen denotado *PAR*) la célula inicial se divide en dos nuevas células, pero en este caso ambas heredan las conexiones de entrada y salida y los valores de los registros internos que forman el estado interno de la célula inicial. El conjunto de genes de división es completado con la inserción de un nuevo gen, denominado *TOT*, que produce la división de la célula madre en dos nuevas células que heredan las conexiones de entrada y salida de la célula madre, así como los valores de los registros internos, al igual que hace el gen *PAR*, pero interconectando mutuamente ambas células con nuevos enlaces. El resultado obtenido de la aplicación del gen *TOT* es equivalente a la aplicación de un gen *PAR* y dos *SEQ* de forma simultánea. La figura 3.6 representa los diversos conexiones generados a partir de la aplicación de cada uno de los genes de división.

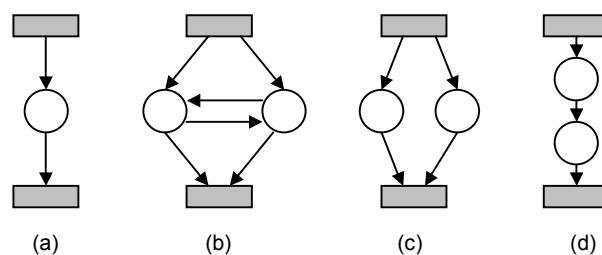


Fig 3.6: Efecto de los distintos genes de división aplicados sobre una célula inicial. (a) Configuración inicial. (b) Aplicación de un gen de división total (*TOT*). (c) Aplicación de un gen de división paralela (*PAR*). (d) Aplicación de un gen de división secuencial (*SEQ*).

- Genes de estado.** Como ya es sabido, la ejecución de un gen de estado produce una modificación en el estado interno de la célula o en el conjunto de conexiones de entrada. La eliminación del gen *BIAS* viene provocado por la desaparición de la variable *Umbra1* en el estado interno de la neurona. En el nuevo modelo de neurona los pesos de las conexiones aumentan el rango de valores posibles, pasando de los valores enteros 1 y -1 del modelo binario, a un intervalo de valores reales positivos comprendidos dentro de un intervalo cuyos límites son previamente definidos. Para permitir que el gen que actúa sobre el valor de los pesos, *VAL-n*, realice mayores o menores modificaciones sobre un determinado enlace es necesario dotarlo de un segundo parámetro *m* que defina la cuantía de la modificación que lleva a cabo, resultando un nuevo gen denotado *VAL-n-m*. Por el contrario el gen *CUT-n* no sufre modificación alguna ya que su función permanece inalterada. La figura 3.7 representa los diversos conexionados generados a partir de la aplicación de cada uno de los genes de estado.

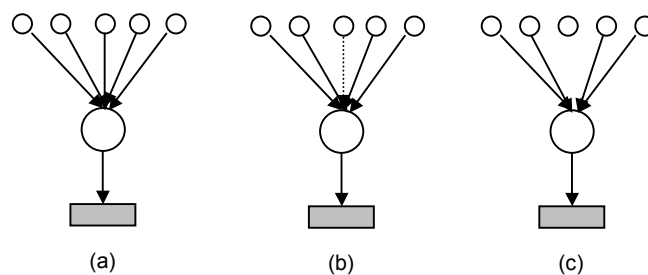


Fig. 3.7. Efecto de los distintos genes de estado que alteran las conexiones de entrada aplicados sobre una célula inicial. (a) Configuración inicial con su conexionado de entrada detallado. (b) Aplicación de un gen de modificación de conexión (*VAL-n-m*) sobre su tercer enlace. (c) Aplicación de un gen de eliminación de conexión (*CUT-n*) sobre su tercer enlace.

- Genes terminales.** La ejecución de un gen terminal da por finalizado el desarrollo de una célula, considerándose a partir de ese momento una neurona totalmente desarrollada. El nuevo papel asumido por este gen, donde decide el tipo de neurona que representa (*excitadora* o *inhibidora*), requiere de la inserción de un parámetro *t* que toma dos posibles valores (*e*, *i*), con el que impone el signo al conjunto de conexiones de salida de la neurona. El nuevo gen es denotado *END-t*.

El conjunto final de genes, para el desarrollo de redes de neuronas biológicas, queda formado por los genes *PAR*, *SEQ*, *TOT*, *VAL-n-m*, *CUT-n*, y *END-t*.

3.2.2 Estructuración del genoma

Como ha sido comentado, el modelo de red de neuronas de integración y disparo estudiado carece de una estructura definida, debido a que el conjunto de neuronas que forma la red neuronal se encuentra formado por una única nube de neuronas, donde toda neurona puede estar plenamente conectada a las demás. Esto puede llevar a pensar que no es posible realizar una estructuración válida del genoma. Tras el análisis de las restricciones que plantea el problema a estudiar y el subconjunto de topologías que las cumplen, es posible encontrar una estructura común a todas ellas. La restricción que plantea el problema consiste en la obligada ausencia de conexiones directas entre las dos neuronas sobre la que se realiza el estudio, *N1* y *N2*. Las topologías con esta restricción pueden considerarse estructuradas de tres partes diferenciadas, una primera compuesta por la neurona *N1*, una segunda compuesta por *N2*, y una tercera formada por la nube de neuronas a las que están conectadas las dos primeras (figura 3.8). Ésta estructuración del fenotipo hace posible la estructuración del genoma en tres cromosomas equivalente al llevado a cabo en el modelo binario.

El proceso de desarrollo embrionario parte de tres células madre iniciales cuyas conexiones establecen desde el principio las restricciones del problema. Dos de las células madres darán lugar a las neuronas *N1* y *N2*, mientras que la tercera célula madre dará lugar a la nube de neuronas completa que se intercala entre ambas. No por ello los cromosomas 1 y 2, que dirigen la evolución de *N1* y *N2*, dejan de ser importantes ya que éstos controlan el conexionado de entradas de ambas neuronas así como el signo de sus conexiones de salida (dependiendo si son excitadoras o inhibitorias).

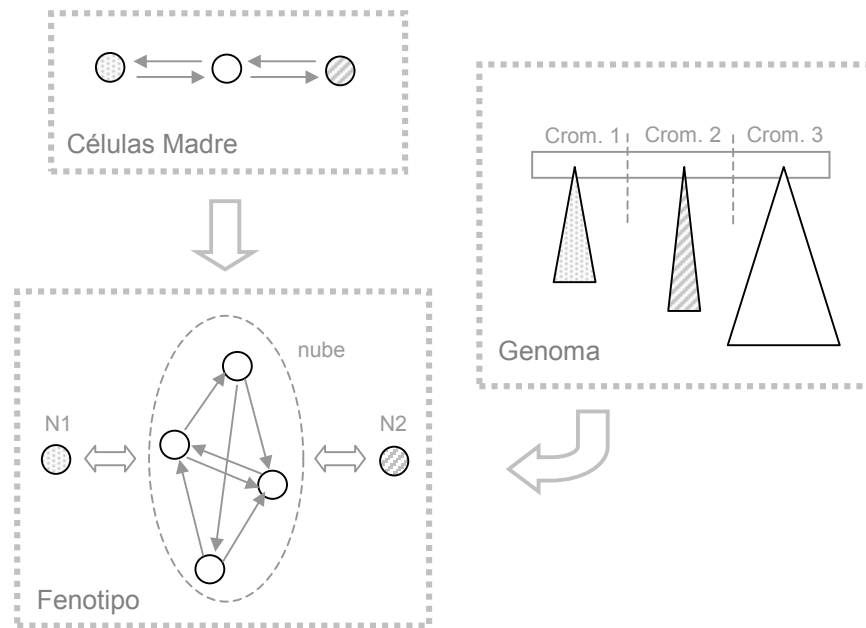


Fig. 3.8: Estructuración del genoma en los tres cromosomas que definen la estructura de la red neuronal tras su aplicación sobre tres células madre iniciales.

3.2.3 La población inicial

Al igual que sucede en la aplicación de cualquier algoritmo genético, los individuos que forman la población inicial deben representar de la forma más homogénea posible el espacio de búsqueda. El número de neuronas que deben constituir las redes es indicado en forma de un intervalo, que representa el número de neuronas que forman la nube de neuronas que interconectan a $N1$ y $N2$. El número de nodos hojas del cromosoma 1 y 2 es siempre de 1 mientras que el número de hojas del cromosoma 3 es elegido aleatoriamente entre los valores posibles del intervalo seleccionado. El proceso de generación seguido para la generación de los árboles de genes es el mismo al descrito para el modelo binario (apartado 2.2.4).

La elección de los genes que forman parte de los cromosomas se controla, al igual que en el modelo binario, mediante la incorporación de un *vector regulador* que indica la probabilidad de aparición de unos genes con respecto a otros. Las modificaciones realizadas

en el conjunto de genes empleados debe verse reflejada igualmente en el *vector regulador* aportando probabilidad de aparición al nuevo gen *TOT* y eliminando la relacionada con el desaparecido gen *BIAS*. De igual forma se requiere un nuevo valor en el *vector regulador* que fije la probabilidad de aparición de neuronas excitadoras frente a inhibidoras, incidiendo en los valores que tomará el parámetro de los genes *END*. Con todo ello se posee pleno control sobre el tamaño y la estructura de los árboles de genes que forman los genomas y las estructuras finales de las redes generadas.

3.3 La función de evaluación

El objetivo planteado en este estudio es la obtención de redes que posean conexiones alternativas al monosináptico para redes de integración y disparo, si esto es posible. La herramienta utilizada para la evaluación del individuo es correlograma cruzado, introducido en el apartado 1.1.6, generado por dos de sus neuronas denominadas *N1* y *N2*. El correlograma generado debe poseer características similares al pasado como referencia, que en las pruebas realizadas es el generado por dos neuronas conectadas monosinápticamente. La limitación que se impone a las redes generadas es que no exista conexión sináptica directa entre las dos neuronas destacadas. El modo de determinar la similitud entre dos correlogramas con el fin de determinar la distancia que los separa es el objeto de este apartado.

Para determinar el *fitness* de un individuo deben realizarse una serie de pasos que comienza con el desarrollo del fenotipo a partir del genoma del individuo a evaluar. La red neuronal desarrollada debe ser ejecutada permitiendo que evolucione la actividad que presentan sus neuronas durante un determinado periodo de tiempo. A partir de la actividad presentada por las dos neuronas destacadas *N1* y *N2* se genera el correlograma cruzado y se calcula la distancia con respecto al pasado como referencia. Este conjunto de pasos, necesarios para la evaluación de un individuo, conducen a una alta *no linealidad* que añade complejidad al problema planteado. Para establecer un valor de distancia, que permita establecer una ordenación entre una serie de correlogramas según sus distancias al correlograma de referencia, es necesario identificar sus características principales y compararlas con las de éste último. Una caracterización suficientemente completa de los

correlogramas permitirá una comparación objetiva de la distancia entre ellos. Como es sabido, un correlograma producido por dos neuronas conectadas monosinápticamente presenta un pico claramente definido sobre la basal posicionado en la barra 1 ms. Ésta es la principal característica que se pretende conseguir en las redes generadas a partir de la aplicación del algoritmo genético, pero no es la única. Otras características del pico, como el número de barras que lo forman, y características de la basal, como su altura media, la desviación estándar y la relación entre ésta y la altura del pico, permiten generar un *vector de caracterización* con el que pueden ser determinadas las diferencias entre correlogramas, considerando múltiples aspectos. A continuación se describe cada una de estas características y el modo en que son valoradas para un correlograma determinado.

- **Posición del Pico.** El pico del correlograma se obtiene considerando como tal el conjunto de barras que superan el 25% de la diferencia entre la barra máxima y la altura media de la basal. Se considera la posición del pico como la media ponderada de las posiciones de las barras que forman el pico, con la altura de los mismos. El valor generado puede ser un valor real que no coincida con una barra concreto sino a un valor intermedio.
- **Anchura del pico al 25%.** Supone el número de barras que superan el 25% de la diferencia entre la barra máxima y la altura media de la basal, no necesariamente adyacentes. Define el conjunto de barras que pueden considerarse parte del pico.
- **Anchura del pico al 50%.** Supone el número de barras que superan el 50% de la diferencia entre la barra máxima y la altura media de la basal.
- **Altura de la basal.** Representa la altura media alcanzada en la zona basal de la gráfica. Su cálculo se realiza con la media de una muestra de la basal correspondiente a las 10 primeras barras del correlograma.
- **Desviación de la basal.** Representa la desviación típica de la basal. Su cálculo se realiza, al igual que en el caso anterior, sobre una muestra correspondiente a las 10 primeras barras del correlograma.
- **Relación Pico-Basal.** Representa la relación proporcional entre la altura de la barra máxima del correlograma con la altura media de la basal.

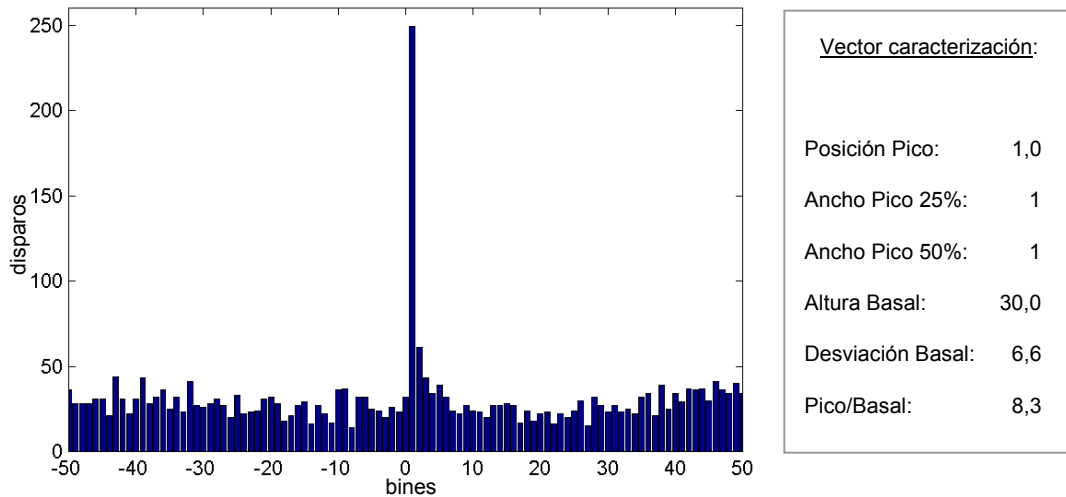


Fig. 3.9: Correlograma generado dos neuronas driver conectadas monosinápticamente y valores de caracterización correspondientes.

La evaluación del correlograma produce el denominado *vector de caracterización* con los valores obtenidos en cada característica. La comparación entre dos correlogramas se limita a la distancia entre sus dos vectores, pudiéndose reducir a un sólo valor numérico mediante la distancia euclídea entre ambos vectores. La diferente importancia de cada característica, y los diferentes rangos de valores que cada una puede tener, hace necesario que éstas sean escaladas previamente mediante diferentes pesos para cada una de las características. El conjunto de pesos para las diversas características forman el *vector de pesos*, que debe ser determinado de forma empírica asignando pesos mayores a las características que desee dotar de mayor importancia en el cálculo final. Para las pruebas realizadas se ha empleado un *vector de pesos* de [10, 1, 1, 5, 5, 5], correspondiente respectivamente a las características: *Posición del pico*, *Altura de la basal*, *Desviación de la basal*, *Relación Pico-Basal*, *Ancho del pico al 25%* y *Ancho del pico al 50%*. La figura 3.9 muestra los valores obtenidos por cada uno de estos parámetros en un correlograma generado a partir de dos neuronas *driver* conectadas por una conexión monosináptica.

3.4 Actividad en los circuitos neuronales

El estudio del ruido sobre el modelo de neurona de integración y disparo, descrito en el apartado 3.1.1, ha permitido obtener dos conjuntos de valores para los parámetros que regulan el nivel de ruido, en función de la actividad que debe presentar la neurona en ausencia total de entradas. Las denominadas neuronas *driver*, con una actividad de unos 15 disparos por segundo, poseen un nivel de ruido definido en el intervalo centrado en 2,89 mV y un radio de 0,85 mV mientras que las *no driver*, con una actividad de tan sólo 5 disparos por segundo, poseen un nivel de ruido cuyo intervalo está centrado en 2.77 mV y un radio de 0.85 mV. De igual forma el estudio ha permitido establecer ciertos límites en el nivel de influencia que una neurona pueda ejercer sobre las demás, atendiendo a los conceptos de *eficacia* y *contribución sinápticas* (apartado 3.1.2), estableciendo a 1 el valor máximo posible del peso de las conexiones sinápticas.

Previa a la aplicación del algoritmo genético, se ha estudiado la actividad que presentan diversos circuitos neuronales dependiendo de que tipo son las neuronas que lo forman. El estudio ha revelado que el uso de neuronas *no driver*, unido al bajo nivel de influencia aportado por los pesos sinápticos, producen una caída de la actividad del circuito neuronal, que se traducen en niveles de basal excesivamente bajos en los correlogramas generados, alejándolos de los resultados deseados. Por ello la aplicación del algoritmo genético se realizará únicamente sobre topologías cuyos conjuntos de neuronas estén formados íntegramente por neuronas *driver*.

3.5 Pruebas y resultados

Para evaluar el comportamiento del algoritmo con cada uno de los métodos de selección de individuos para el cruzamiento y la mutación se han realizado repetidas pruebas con diversos conjuntos de parámetros. De esta forma se pretende analizar el efecto que cada uno de estos métodos produce sobre la evolución de la población en la búsqueda de un conexionado alternativo al monosináptico, para diferentes tamaños de redes neuronales.

Pruebas con nubes de 3 a 5 neuronas

El primer conjunto de pruebas sobre el algoritmo genético evalúa los diversos métodos de selección sobre circuitos neuronales compuestos por las neuronas $N1$ y $N2$ y una nube de entre 3 y 5 neuronas. Con estas pruebas se pretende evaluar el potencial de los métodos de selección desarrollados en la búsqueda de topologías óptimas de redes neuronales de integración y disparo. El cálculo del *fitness* de los individuos se realiza atendiendo a la distancia entre el correlograma que genera y el correlograma de referencia generado por dos neuronas *driver* conectadas monosinápticamente. Cada prueba se ha realizado sobre poblaciones de 100 individuos a lo largo de 100 generaciones. La ejecución del algoritmo, con el mismo conjunto de valores para los parámetros de configuración, a sido realizado 10 veces para establecer estadísticas de comportamiento de la población para dichos valores. Las gráficas de las figuras 3.10, 3.11, 3.12 y 3.13 representan la evolución de las poblaciones a lo largo de 100 generaciones para cada uno de los métodos aplicados. En las gráficas 3.10.a, 3.11.a, 3.12.a y 3.13.a se agrupan los resultados de las 10 ejecuciones realizadas para cada uno de los métodos de selección empleados, representando los valores más altos y más bajos de *fitness* obtenidos en cada generación por cada una de las ejecuciones. En cada una de las gráficas se representan la media de los tres valores más altos y la media de los tres resultados más bajos de *fitness* de cada una de las ejecuciones en cada una de las generaciones realizadas de cada ejecución (puntos verdes y amarillos, respectivamente). La representación negativa del *fitness* en las gráficas tiene únicamente la intención de expresar la evolución del *fitness* a través de curvas ascendentes. La media de los mejores y peores resultados de todas las ejecuciones son representados en forma de líneas continuas (azul y roja, respectivamente), permitiendo evaluar el comportamiento general de la evolución de las poblaciones para cada uno de los métodos de selección empleados. Las gráficas 3.10b, 3.11.b, 3.12.b y 3.13.b muestran la evolución en el número de sustituciones, y la media de todos ellos, producidas por cada uno de los métodos de selección empleados.

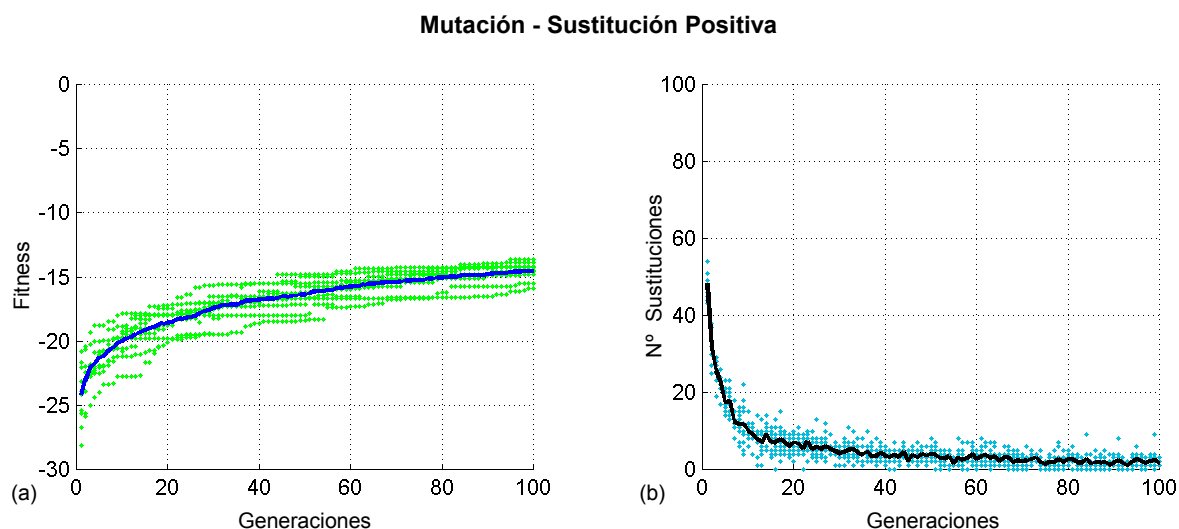


Fig. 3.10: Aplicación del método de selección de individuos sustitución positiva para el operador mutación a lo largo de 100 generaciones sobre una población de 100 individuos. (a) Evolución de los fitness de la población. (b) Evolución del número de individuos sustituidos.

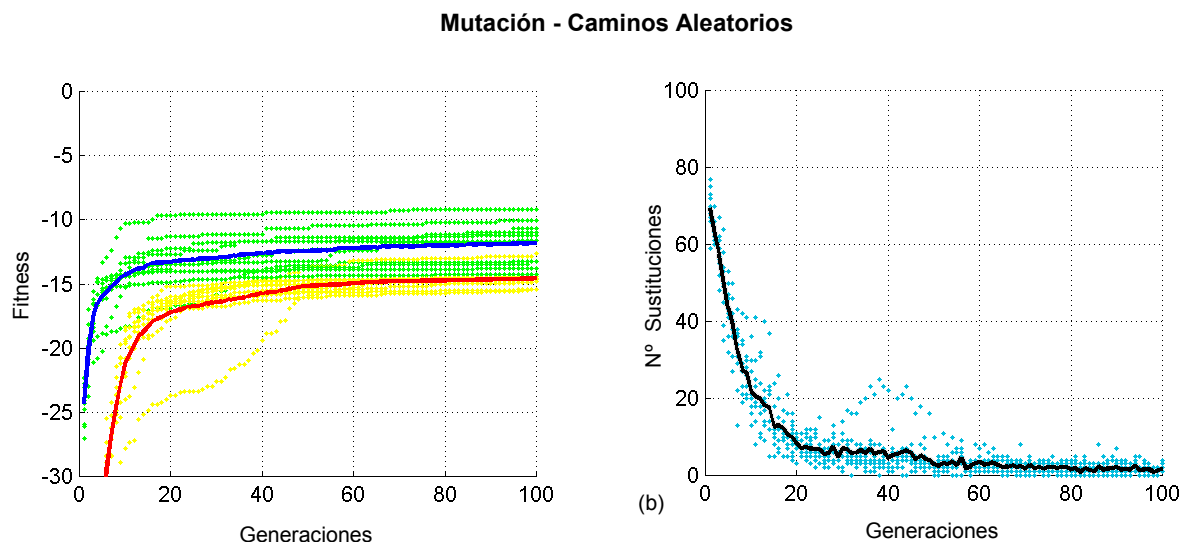


Fig. 3.11: Aplicación del método de selección de individuos caminos aleatorios para el operador mutación a lo largo de 100 generaciones sobre una población de 100 individuos. (a) Evolución de los fitness de la población. (b) Evolución del número de individuos sustituidos.

Las gráficas mostradas en la figura 3.10 y 3.11, correspondientes a los métodos de selección de individuos para la mutación *sustitución positiva* y *caminos aleatorios*, muestran la evolución de la población y el índice de sustitución obtenido a lo largo de 100 generaciones. En

ambos métodos la media de los mejores individuos de las poblaciones iniciales se sitúan en torno a un valor de *fitness* de -25 . La evolución de la población se realiza de una forma mucha más activa en *camino aleatorios* donde el valor -15 es superado por los mejores individuos en tan sólo 7 generaciones, requiriendo el método *sustitución positiva* de al menos 70 generaciones para alcanzar esa cota. Trascurridas las 100 generaciones, los mejores individuos obtenidos con *sustitución positiva* se sitúan en torno a -14 mientras que con *camino aleatorios* la cota alcanzada se sitúa en -12 . El nivel de sustituciones llevado a cabo por *camino aleatorios* la convierte también en ganadora en este aspecto ya que se mantiene, durante las 60 primeras generaciones, por encima del obtenido por *sustitución positiva*, y permanece igualada a partir de ésta. Otro factor a tener en cuenta es la evolución de los individuos peor valorados de las poblaciones para ambos métodos. En el caso de *camino aleatorios*, éstos se sitúan a partir de la generación 20 a menos de 5 puntos de los mejor valorados, lo que implica que la totalidad de la población se compone de individuos con una alta valoración. En *sustitución positiva* los individuos peor valorados de cada población no superan en ningún momento una valoración de -30 , lo que de nuevo lo sitúa como perdedor en las comparativas con *camino aleatorios*.

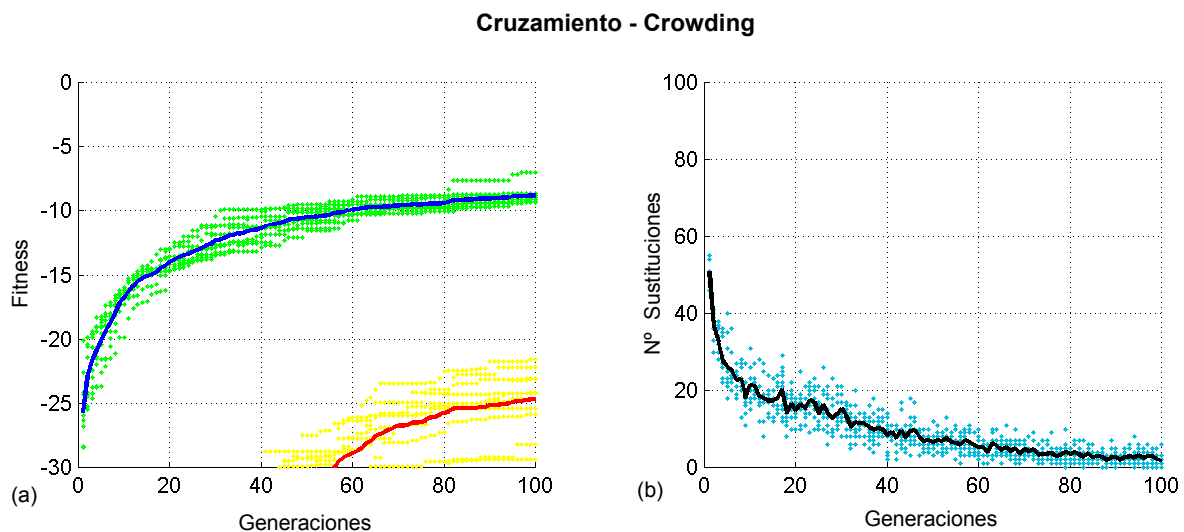


Fig. 3.12: Aplicación del método de selección crowding para el operador cruzamiento a lo largo de 100 generaciones sobre una población de 100 individuos. (a) Evolución de los fitness de la población. (b) Evolución del número de individuos sustituidos.

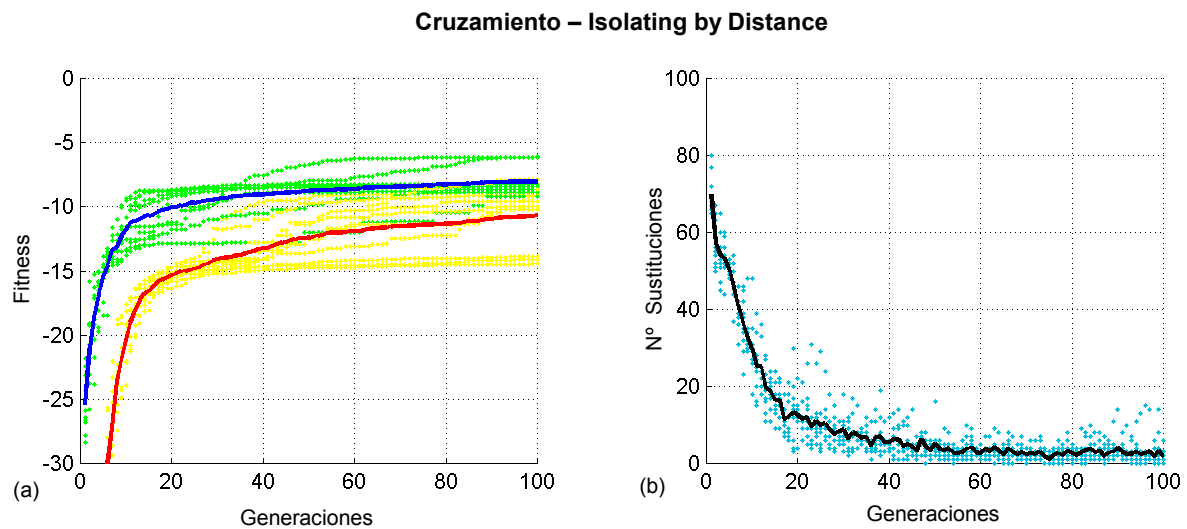


Fig. 3.13: Aplicación del método de selección *isolating by distance* para el operador cruzamiento a lo largo de 100 generaciones sobre una población de 100 individuos. (a) Evolución de los fitness de la población. (b) Evolución del número de individuos sustituidos.

La evolución de las poblaciones y el índice de sustituciones obtenidos por los métodos de selección de individuos para el cruzamiento *crowding* e *isolating by distance* son mostrados en las gráficas de la figura 3.12 y 3.13. En ellas se puede observar como, a pesar de la diferente evolución de las poblaciones, los valores finales de los mejores individuos se sitúa en torno a -8 en ambos métodos. La búsqueda más “voraz” en las primeras generaciones de *isolating by distance* se ve seriamente frenada a partir de la generación 40, mostrando a partir de entonces una tímida mejora. Si bien la curva generada por *crowding* con respecto a sus individuos mejor valorados es menos pronunciada en las primeras generaciones que *isolating by distance*, se muestra más constante y eficiente en las generaciones que siguen alcanzando los valores obtenidos por este en la generación 100. Los resultados obtenidos por ambos métodos con respecto a los individuos peor valorados son muy distintos, llegando en el caso de *isolating by distance* a una distancia de tan sólo 3 puntos con respecto a los mejor valorados, y a 15 en el caso de *crowding*. En el aspecto del nivel de sustituciones llevado a cabo por ambos métodos las gráficas 3.12.b y 3.13.b confirman lo ya comentado. *isolating by distance* mantiene durante las primeras 20 generaciones un nivel muy superior en el número de sustituciones de individuos al llevado a cabo por *crowding*, estando en la primera generación en torno al 70% frente al 50% de *crowding*. Este alto índice de sustituciones lleva a la población a aumentar rápidamente la valoración de sus individuos, pero el ritmo de sustituciones de *isolating By distance* cae en pocas generaciones para situarse por debajo del obtenido por *crowding*.

Para permitir una comparación objetiva del uso de los operadores genéticos y la política de selección de individuos, en la búsqueda de soluciones al problema planteado, se ha realizado un nuevo conjunto de pruebas, donde han sido sustituida la generación de individuos mediante la aplicación de operadores genéticos por la generación de individuos de forma aleatoria. Para establecer que individuos forman parte de una nueva generación se realiza una competición entre individuos que formaban parte de la población anterior y nuevos individuos generados aleatoriamente. Esta competición entre pares de individuos se realiza por cada una de las posiciones de la población, ocupando la posición en la siguiente generación el ganador (mejor valorado). Los resultados obtenidos, mostrados en las gráficas de la figura 4.14, son claramente inferiores a los obtenidos con el aplicación de los métodos de selección de individuos asociados a los operadores genéticos, a excepción de los resultados generados por la aplicación del método de selección de individuos para la mutación *sustitución positiva*, que prácticamente iguala.

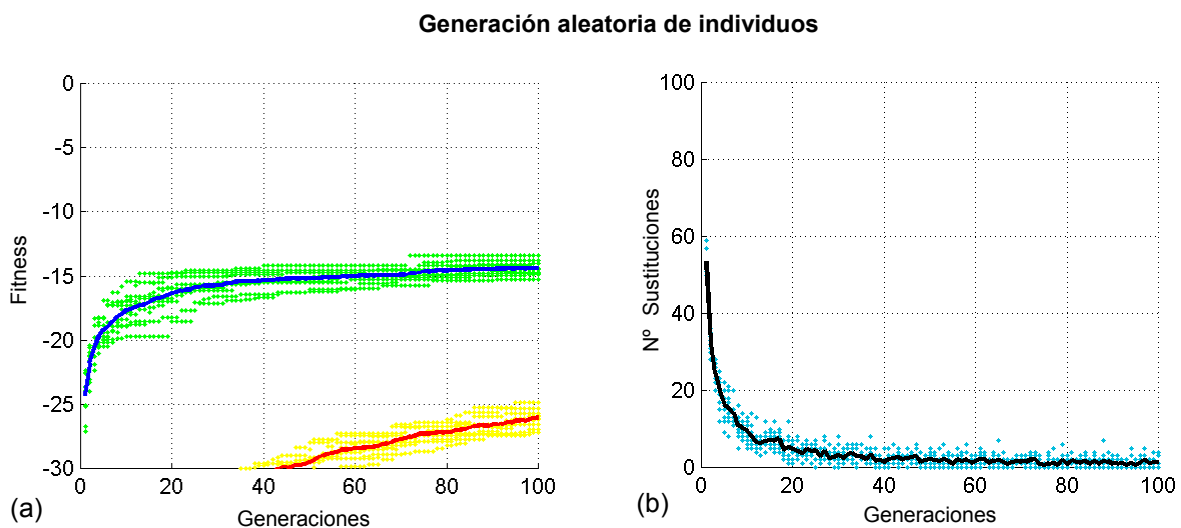


Fig. 4.14: Generación de Individuos aleatorios a lo largo de 100 generaciones sobre una población de 100 individuos. (a) Evolución de los fitness de la población. (b) Evolución del número de individuos sustituidos.

La totalidad de las pruebas llevadas a cabo con el algoritmo genético y el empleo de los diferentes métodos de selección de individuos, muestran que los mejores individuos, tras 100 generaciones, distan en el mejor de los casos un mínimo de 7 puntos del individuo objetivo. El análisis de los individuos mejor valorados (obtenidos con los métodos de cruzamiento) ha revelado que éstos representan redes compuestas por 12-15 neuronas (con nubes 10-13 neuronas). Esto es posible debido a que los individuos generados mediante cruzamiento no están limitados al número de neuronas que poseen sus padres permitiendo la búsqueda de individuos fuera del intervalo inicial de tamaños permitidos. A lo largo de sucesivas

generaciones la búsqueda de individuos mejor valorados puede salir del rango inicial y ampliar la búsqueda a redes con tamaños superiores o inferiores al inicial. Esta característica se presenta como una garantía en la búsqueda de resultados, ya que no compromete la obtención de éstos a intervalos mejor o peor seleccionados.

Pruebas con nubes de 10 a 20 neuronas

Las pruebas anteriores demuestran que el intervalo de tamaños de las nubes de neuronas usado en la población inicial (entre 3 y 5) excluye a los mejores individuos, cuyas nubes están formadas por entre 10 y 13 neuronas. Estos individuos sólo han sido accesibles con la aplicación de los métodos de selección asociados al operador cruzamiento, dada la capacidad del operador para ampliar la búsqueda más allá de los tamaños de los individuos de la población inicial. Por esa razón se ha visto necesario la realización de un nuevo conjunto de pruebas reubicando el intervalo de tamaños entre 10 y 20 neuronas, permitiendo así a los métodos de selección asociados a la mutación trabajar en un espacio de búsqueda que incluye los mejores individuos obtenidos en las pruebas anteriores. De igual forma se comprobará si los métodos asociados al cruzamiento pueden mejorar los resultados obtenidos en las pruebas anteriores. Las figuras 3.15, 3.16, 3.17 y 3.18 muestran la evolución de las poblaciones a lo largo de 100 generaciones para cada uno de los métodos de selección aplicados.

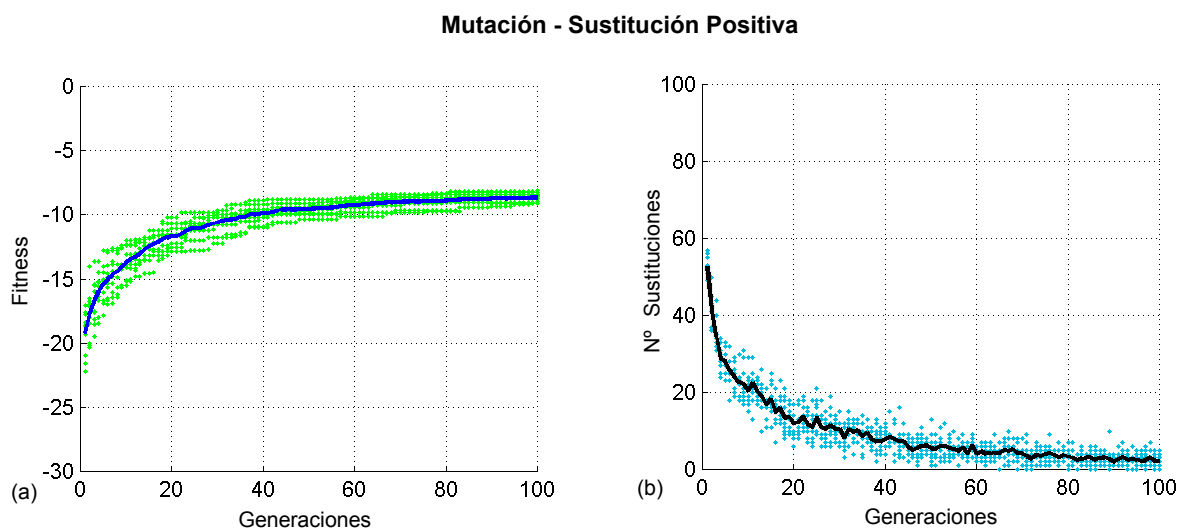


Fig. 3.15: Aplicación del método de selección sustitución positiva para el operador mutación a lo largo de 100 generaciones sobre una población de 100 individuos. (a) Evolución de los fitness de la población. (b) Evolución del número de individuos sustituidos.

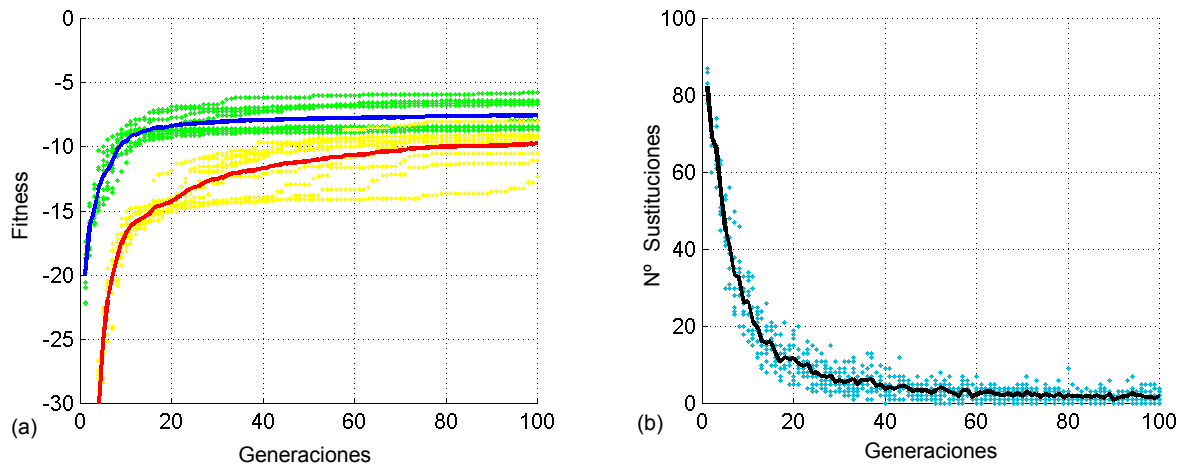
Mutación – Caminos Aleatorios

Fig. 3.16: Aplicación del método de selección caminos aleatorios para el operador mutación a lo largo de 100 generaciones sobre una población de 100 individuos. (a) Evolución de los fitness de la población. (b) Evolución del número de individuos sustituidos.

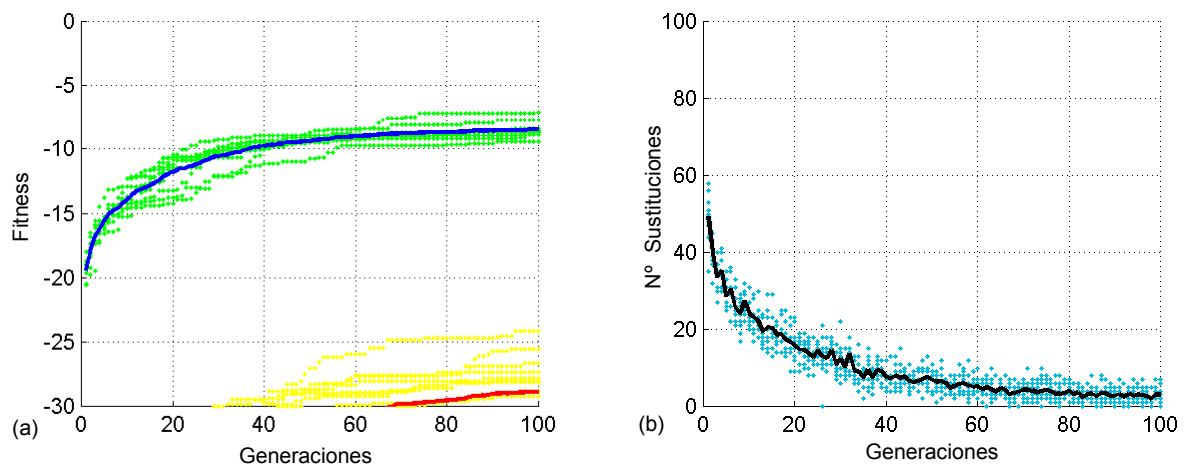
Cruzamiento - Crowding

Fig. 3.17: Aplicación del método de selección crowding para el operador cruzamiento a lo largo de 100 generaciones sobre una población de 100 individuos. (a) Evolución de los fitness de la población. (b) Evolución del número de individuos sustituidos.

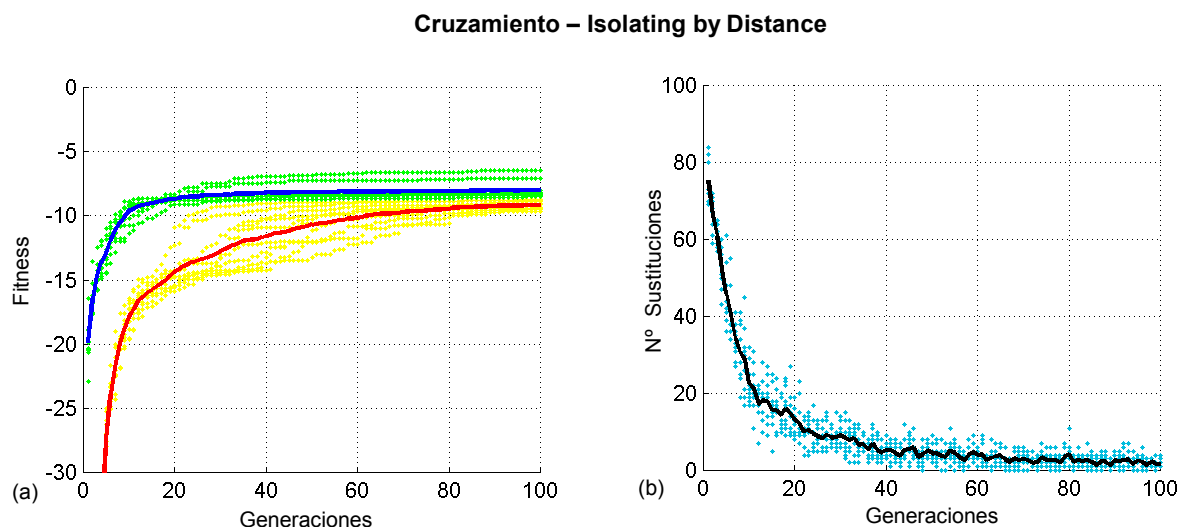


Fig. 3.18: Aplicación del método de selección *isolating by distance* para el operador *cruzamiento* a lo largo de 100 generaciones sobre una población de 100 individuos. (a) Evolución de los *fitness* de la población. (b) Evolución del número de individuos sustituidos.

El nuevo intervalo, usado para determinar el tamaño de los individuos de la población inicial, ha permitido a los métodos asociados al operador *mutación* alcanzar los resultados obtenidos por los métodos de *cruzamiento* en el conjunto de pruebas anterior, suponiendo una mejora de entre 3 y 5 puntos. Por el contrario, los métodos asociados al *cruzamiento* no han conseguido una sustancial mejora en el valor de *fitness* de los mejor valorados, aunque si puede observarse que estos valores son alcanzados en un menor número de generaciones. En esta ocasión la búsqueda, llevada a cabo por los métodos asociados al operador *cruzamiento*, no se ha extendido fuera del intervalo fijado para la población inicial, lo que supone que es en ese intervalo donde se encuentran los individuos mejor valorados.

Las mejores soluciones encontradas por todos los métodos se sitúan en torno a los 7 puntos de distancia con respecto al conexionado del correlograma de referencia es mostrado en la figura 3.19. La figuras 3.20, 3.21, 3.22 y 3.23 reúnen los correlogramas correspondientes a algunos de los mejores individuos obtenidos con cada uno de los métodos de selección empleados en el algoritmo genético, junto con sus *vectores de caracterización*.

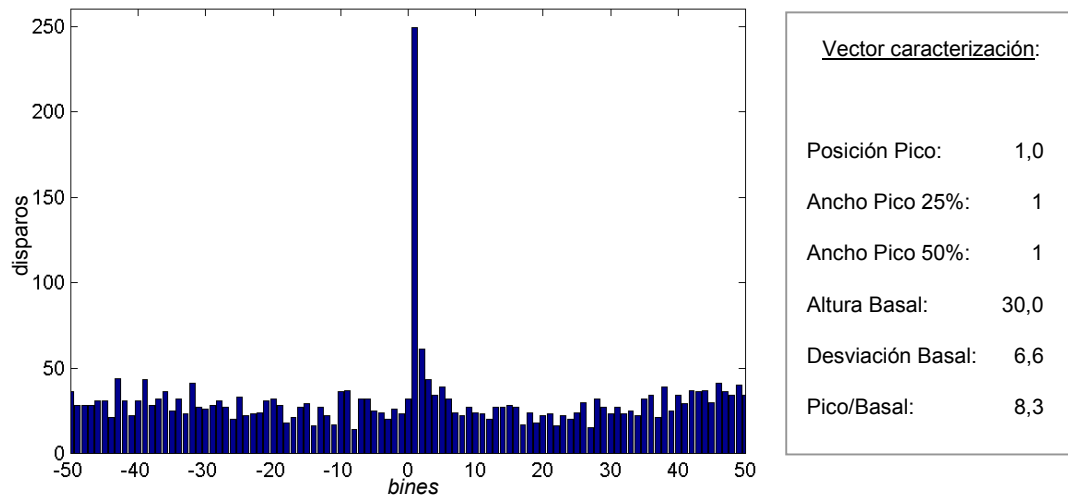


Fig. 3.19: Correlograma generado por un circuito neuronal compuesto por dos neuronas driver conectadas monosinápticamente, usado como correlograma de referencia en las ejecuciones del algoritmo genético.

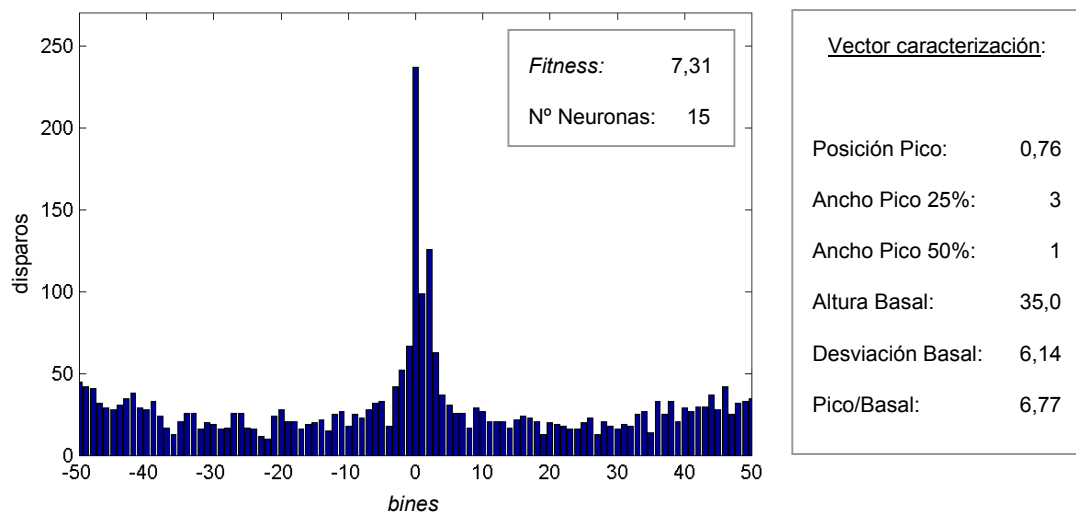


Fig. 3.20: Correlograma y vector de caracterización correspondiente a una red neuronal formada por 15 neuronas (nube formada por 13 neuronas) y seleccionada como mejor individuo en una ejecución donde el método aplicado es sustitución positiva. La distancia que lo separa del correlograma de referencia es de 7,31.

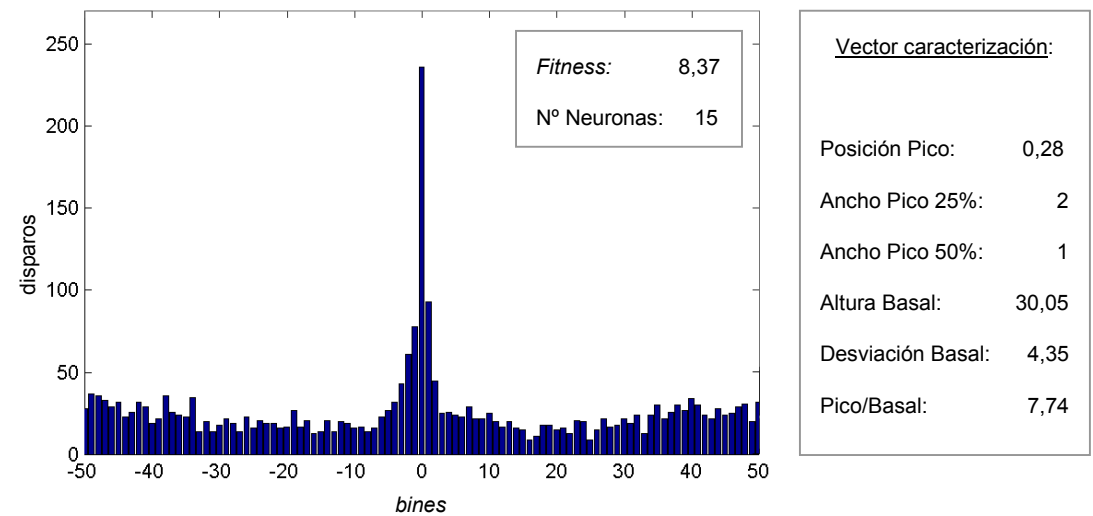


Fig. 3.21: Correlograma y vector de caracterización correspondiente a una red neuronal formada por 15 neuronas (nube formada por 13 neuronas) y seleccionada como mejor individuo en una ejecución donde el método aplicado es caminos aleatorios. La distancia que lo separa del correlograma de referencia es de 8,37.

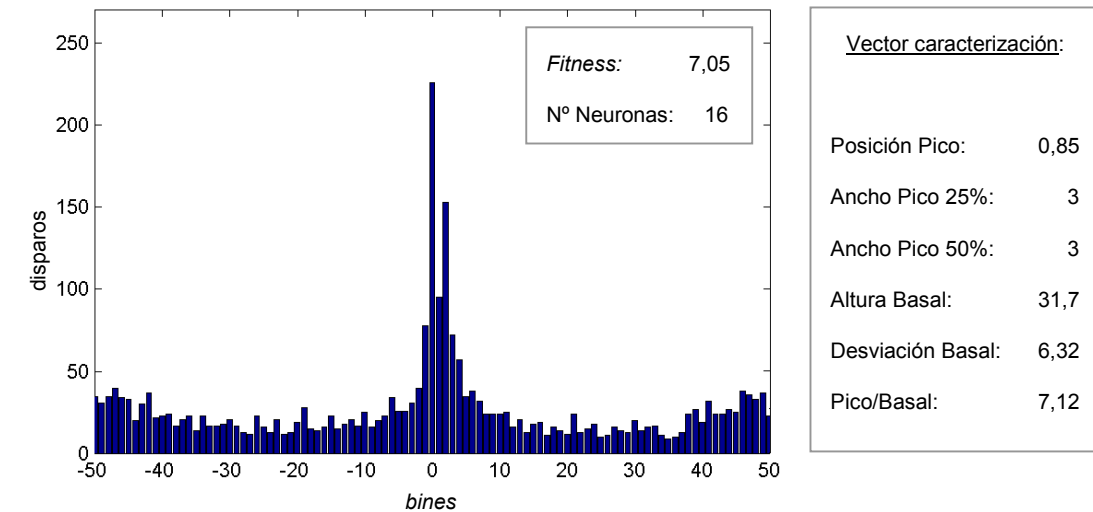


Fig. 3.22: Correlograma y vector de caracterización correspondiente a una red neuronal formada por 16 neuronas (nube formada por 16 neuronas) y seleccionada como mejor individuo en una ejecución donde el método aplicado es crowding. La distancia que lo separa del correlograma de referencia es de 7,05.

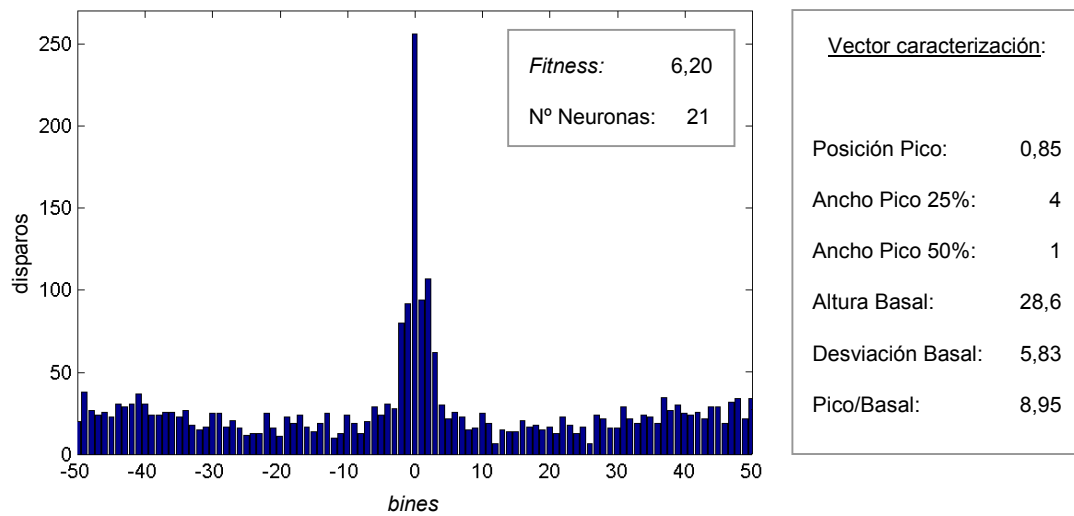


Fig. 3.23: Correlograma y vector de caracterización correspondiente a una red neuronal formada por 21 neuronas (nube formada por 19 neuronas) y seleccionada como mejor individuo en una ejecución donde el método aplicado es *isolating by distance*. La distancia que lo separa del correlograma de referencia es de 6,20.

Las repetidas búsquedas realizadas con la aplicación del algoritmo genético y las diversas políticas de selección aplicadas no han conseguido la generación de redes que produzcan correlogramas que reduzcan prácticamente a 0 la distancia de cada una de las características que forman el vector de caracterización. Dadas las múltiples dimensiones en las que se descompone el valor de *fitness*, y la imposibilidad de reducir simultáneamente la distancia en todas ellas, el algoritmo busca acercamientos al individuo solución a través de la minimización de la distancia de unas de características en detrimento de otras, condicionadas principalmente por el *vector de pesos*.

El *vector de pesos* seleccionado para todas las ejecuciones del algoritmo genético da una prioridad de 10 a la *Posición del pico*, de 5 a la *Relación Pico-Basal*, al *Ancho del pico al 25%* y al *Ancho del pico al 50%*, y de 1 a la *Altura de la basal* y la *Desviación de la basal*. La alta prioridad dada a la posición del pico viene justificada por ser la principal característica de un correlograma generado por dos neuronas conectadas monosinápticamente (un pico claramente definido en la barra 1). Esto induce al algoritmo a priorizar la minimización de la distancia en la característica “centro del pico”, en detrimento de otras como “anchura del pico al 25%” o la relación *pico-basal*, como puede observarse en los correlogramas mostrados en la figuras 3.20, 3.21, 3.22 y 3.23. Por ello la mayor parte de las aproximaciones pasan por generar un pico más ancho de lo deseado pero cuyo centro se sitúa en torno a 1, minimizando el valor final del *fitness*.

Conclusiones

Todos los métodos empleados, sobre poblaciones iniciales con redes cuyas nubes están formadas por entre 10 y 20 neuronas, alcanzan una distancia mínima en torno a 7. La capacidad demostrada por los métodos de selección asociados con el operador *cruzamiento* en la optimización de topologías, encontrando individuos con una mejor valoración fuera del intervalo de tamaños fijados para la población inicial, garantiza que de existir mejores individuos fuera de este intervalo la búsqueda no se limitaría a los límites establecidos en la población inicial. La no obtención de conexiones alternativas a un conexionado monosináptico con las limitaciones planteadas, es un paso en favor de la tesis que afirman la no existencia de dichas topologías.

Capítulo 4

Materiales y métodos

Las aplicaciones software desarrolladas a lo largo del presente proyecto han permitido realizar un completo estudio sobre el modelo de codificación implícita *codificación celular*, su aplicación para el desarrollo de redes neuronales, los operadores genéticos adecuados y su uso final en la búsqueda de topologías óptimas para diferentes modelos de neuronas. El conjunto de pruebas realizadas con las aplicaciones a permitido realizar estudios comparativos de los métodos de selección de individuos, así como de la evolución de las poblaciones con cada uno de ellos. A lo largo de este apartado se describen las dos aplicaciones desarrolladas y su empleo para la realización de las numerosas pruebas.

4.1 Estructuración del proyecto

El presente proyecto ha sido estructurado en ocho fases, atendiendo a los objetivos planteados y su desarrollo. La aplicación de metodologías de Ingeniería del Software sólo tiene sentido en cinco de las ocho fases del proyecto, dedicadas al desarrollo de las aplicaciones que implementan el algoritmo genético. Las aplicaciones desarrolladas, que implementan el algoritmo genético para dos modelos neuronales distintos, han permitido estudiar y analizar el potencial de los operadores genéticos y los métodos de selección para la resolución de problemas concretos en sus respectivos ámbitos de aplicación. A continuación se describen cada una de las ocho fases:

1. Estudio comparativo de diferentes modelos de codificación para redes de neuronas artificiales, donde se analizan y comparan las diversas propuestas de codificación existentes en la literatura actual y su posible adaptación a modelos de neuronas con fuerte inspiración biológica.

2. Diseño e implementación de un modelo de codificación prototipo, a partir de los estudiados, para la resolución de problemas de tipo lógico mediante redes de neuronas binarias.
3. Diseño, implementación y pruebas de operadores genéticos sobre la base del modelo de codificación obtenido en la fase anterior.
4. Estudio de resultados obtenidos por el algoritmo genético en la búsqueda de soluciones a problemas lógicos con el uso de los diversos operadores genéticos.
5. Diseño e implementación de un modelo simple de neurona de integración y disparo. Se estudian los parámetros que generen el comportamiento deseado del modelo, para la obtención de correlaciones de disparos entre neuronas que sigan un patrón establecido.
6. Optimización del modelo de codificación diseñado en la segunda fase para la codificación de redes de neuronas que siguen el modelo neuronal obtenido en la fase anterior.
7. Diseño, implementación y pruebas de operadores genéticos sobre la base del modelo de codificación obtenido en la fase 6.
8. Optimización de topologías de redes de neuronas de integración y disparo para la obtención de conexiones alternativas a los propuestos por un experimentador.

La primera fase, consistente en el estudio y análisis de los diversos modelos de codificación implícita aplicados al desarrollo de redes neuronales propuestos por en la actualidad, da como resultado la elección del método de *codificación celular* como el mejor candidato para su desarrollo y aplicación en el presente proyecto.

El conjunto formado por las fases 2 y 3 dan lugar al desarrollo de una aplicación software que implementa el algoritmo genético, basado en *codificación celular*, para la optimización de redes de neuronas binarias. Esta aplicación es usada en la fase 4 para realizar un estudio empírico de la eficacia de los diversos métodos de búsqueda de topologías de redes neuronales binarias para la resolución de problemas lógicos. El ciclo de vida aplicado a dicho desarrollo software ha sido el *modelo espiral*, que ha permitido el sucesivo refinamiento de las especificaciones sobre prototipos del modelo de codificación y operadores genéticos.

La aplicación del modelo de codificación y operadores genéticos llevada a cabo en las fases 2 y 3, y el estudio sobre los resultados obtenidos en la fase 4 han servido como punto de partida al conjunto de fases 5, 6 y 7, para realizar la adaptación del modelo de codificación al desarrollo de redes neuronales de integración y disparo y su aplicación en la optimización de topologías.

La última fase utiliza el software desarrollado para analizar la eficacia de los diversos métodos de búsqueda en la búsqueda de topologías de redes neuronales alternativas a un conexionado monosináptico, a través de la comparación de sus respectivos correlogramas cruzados.

La distribución temporal de las fases así como el desarrollo conjunto de algunas de ellas están representados en la figura 4.1.

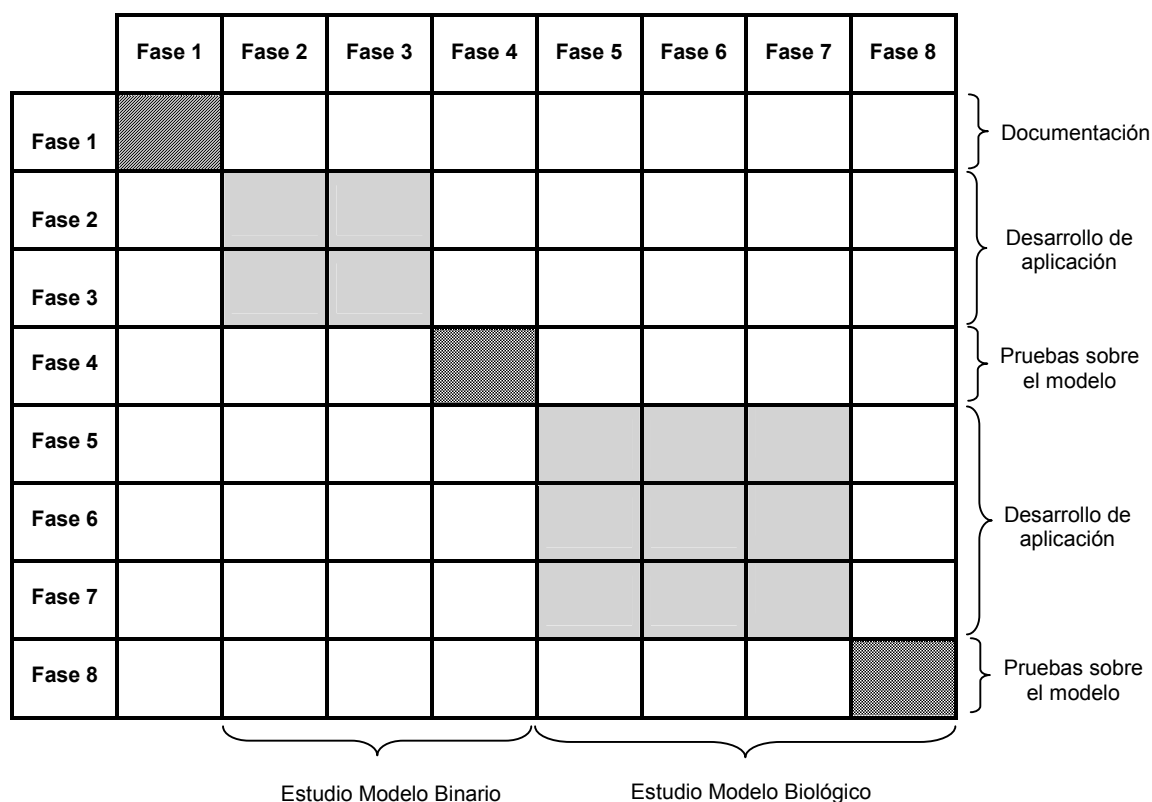


Fig. 4.1. Distribución temporal de las fases del proyecto.

4.2 Ciclo de vida y metodologías software

Para el desarrollo de las aplicaciones software incluidas en el presente proyecto, que implementan el algoritmo genético y han permitido el estudio comparativo entre los diversos métodos de búsqueda aplicados a los diferentes modelos neuronales, se ha empleado el paradigma orientado a objetos.

La orientación a objetos consiste en concebir el sistema como una colección cooperativa de objetos, cada uno de los cuales representa una instancia de alguna clase, y se caracteriza fundamentalmente por las siguientes características:

- El problema real se transforma en un conjunto de objetos que interactúan entre si para resolverlo
- El estado del sistema queda descentralizado en los diferentes objetos que lo componen
- La realidad es modelada de un modo más cercano a la perspectiva del usuario.
- Los componentes son reutilizables, ampliables y modificables sin afectar al resto de la estructura del sistema ya que los objetos son vistos como una caja negra que ofrece una serie de servicios a través de sus métodos.

Por todo ello el paradigma orientado a objetos es indudablemente el más adecuado para el desarrollo de las aplicaciones del presente proyecto. Conceptos ampliamente usados a lo largo de todo el proyecto, como “población”, “individuo”, “fenotipo”, “genoma”, “gen” y “neurona”, adquieren presencia por si mismos en el paradigma orientado a objetos, permitiendo modelar con total precisión cada uno de estos conceptos y la relación existente entre ellos. Herramientas como la herencia permiten, además, modelar la especialización de ciertos tipos de objetos a través de la definición de subclases. Un ejemplo de la aplicación especialización se encuentra en el concepto de “gen” que se especializa a diversas clases como son “gen de estado”, “gen de división” y “gen terminal”, y éstos a su vez en genes concretos con comportamientos y características muy determinadas, estableciéndose una jerarquía de especialización en tres niveles.

Los lenguajes que dan soporte al paradigma orientado a objetos son los denominados lenguajes orientados a objetos. De entre todos ellos, las características de C++ lo han

convertido en el elegido para el desarrollo de las aplicaciones del presente proyecto. El principal argumento usado a su favor es el mayor rendimiento obtenido por aplicaciones compiladas para un sistema informático concreto que aplicaciones interpretadas que sacrifican parte de su rendimiento para permitir portabilidad entre distintos sistemas, como es el caso de Java.

La elección del lenguaje de programación orientado a objetos empleado en el proyecto se ha basado fundamentalmente en la necesidad de eficiencia computacional. El gran número de pruebas a realizar sobre las aplicaciones y el tamaño de éstas requerían un uso eficiente de los recursos, por lo que su optimización ha sido un objetivo constante en el diseño e implementación de las aplicaciones. Si bien Java plantea ventajas de portabilidad entre diversas plataformas, su menor eficiencia frente a C++ lo han descartado como candidato válido, y han hecho de este último el lenguaje elegido para en el desarrollo de las aplicaciones.

El uso de un ciclo de vida en espiral proporciona un modelo evolutivo para el desarrollo de sistemas software complejos. En él se realizan cuatro tipos de actividades que se repiten cíclicamente permitiendo la sucesiva depuración del software, que corresponden a las citadas a continuación:

- *Planificación.* Consiste en la determinación de los objetivos, posibles alternativas y restricciones.
- *Análisis de riesgo.* Donde se han analizado los riesgos, principalmente técnicos, inherentes a las decisiones adoptadas en la planificación. Se ha hecho especial hincapié en la optimización de los recursos que no produzca limitación en la escalabilidad del sistema.
- *Ingeniería.* Consistente en el desarrollo del un prototipo ajustado a los requisitos.
- *Evaluación.* Donde se evalúa el prototipo mediante pruebas.

Estas cuatro tareas son repetidas un número determinado de iteraciones, cada una de las cuales da lugar a sucesivas versiones del software, más completas y más cercanas a los objetivos finales del proyecto. El ciclo de vida concluye con la aplicación totalmente desarrollada.

La aplicación del ciclo de vida en espiral a permitido englobar las dos aplicaciones en un mismo ciclo de vida final. El desarrollo de la primera aplicación, que da lugar a una aplicación completa que implementa el algoritmo genético para la búsqueda de topologías óptimas de redes de neuronas binarias, es tomado como punto de partida para el desarrollo de la segunda aplicación, con un modelo neuronal de integración y disparo. La aplicación inicial se convierte en el prototipo de partida de la segunda, y la mayor parte de sus componentes son adaptados en sucesivos refinamientos para dar lugar a una versión más compleja y adaptada a los nuevos objetivos.

4.3 Desarrollo de aplicaciones

Las aplicaciones desarrolladas han sido los medios que han permitido estudiar de una forma empírica los resultados obtenidos de la aplicación del algoritmo genético a distintos problemas de optimización de redes neuronales. El estudio de la evolución de las poblaciones en las numerosas pruebas realizadas a permitido analizar las capacidades y limitaciones de cada uno de los métodos usados en las situaciones planteadas (distintos tipos de problemas a resolver, tamaños de población o complejidad de individuos). En los siguientes apartados se da una visión general de las aplicaciones desarrolladas a través del modelo conceptual y el conjunto de parámetros que permiten al experimentador modelar las pruebas a realizar con el algoritmo genético.

4.3.1 Análisis software. Modelo conceptual

El conjunto de conceptos que forman parte del algoritmo genético es representado en este apartado a través de un diagrama de estructura estática UML, denominado modelo conceptual. El objetivo de este modelo es la representación del dominio de la aplicación, no de los componentes software que la componen, cerca de la visión del usuario. De esta forma el modelo expresa una propuesta de organización del dominio del problema que finalmente ha

servido para, con algunos matices, dar origen al diagrama final de clases de las aplicaciones. Ambas aplicaciones comparten una estructura común, que permite la reutilización de componentes entre ellas, quedando las diferencias localizadas en el conjunto de genes, la red neuronal y el problema a resolver. A continuación se presentan los modelos conceptuales de las aplicaciones, describiendo inicialmente la estructura común a ambas y continuación las particularizaciones necesarias para completar cada una de ellas.

Modelo conceptual. Estructura común

La estructura común a las aplicaciones desarrolladas consiste en el conjunto de elementos que configuran el algoritmo genético. Los conceptos a los que en tantas ocasiones se ha hecho referencia a lo largo del desarrollo del proyecto son los que conforman la estructura básica.

El puesto más alto del esquema es ocupado por el concepto *Población*, definido como una colección de *Individuos* que evoluciona en el tiempo (generaciones) mediante la aplicación de diferentes métodos de generación y selección de individuos con el objetivo de encontrar soluciones a un problema dado.

Cada uno de los individuos que forman parte de la población está compuesto por un genotipo y un fenotipo. El primero, representado por el concepto *Genoma*, constituye la descripción codificada del individuo, y el segundo, representado por el concepto *Fenotipo*, constituye el conjunto de caracteres que definen al individuo.

El *Genoma* se encuentra formado por una colección ordenada de instrucciones denominadas genes y representadas por el elemento *Gen*. Como ya ha sido comentado, el modelo propuesto estructura el genoma en tres árboles de genes (común en ambas aplicaciones) por lo que estos genes no sólo deben representar instrucciones codificadas de la representación del individuo, sino también a nodos de un árbol de ramas ordenadas. Estas dos visiones de un *Gen* deben considerarse complementarias, ya que la primera, en la que el *Gen* es visto como una instrucción codificada que determina el modo en que debe actuarse durante el proceso de desarrollo embrionario, representa el “Cómo y Qué”, la visión del elemento como un “nodo de un árbol ordenado” determina el momento y el área de actuación, representando el “Cuándo” y “Dónde”. La especialización de genes en tres clases, definidas por los elementos *Gen_División*, *Gen_Estado* y *GenTerminal*, permite definir tipos de nodos con distinta aridad y

con funciones claramente diferenciadas. La creación de nuevos genes y la mutación de unos en otros son regidas por un elemento denominado *Regulador*, que determina la probabilidad de aparición de cada tipo de gen.

El *Fenotipo* construye la red neuronal asociada, representada por el elemento *Red*, durante el desarrollo embrionario del individuo mediante la interpretación de las instrucciones contenidas en el genoma del individuo. La evaluación del individuo se realiza atendiendo a los resultados obtenidos por el elemento *Red* ante las condiciones definidas por el elemento *Problema*. De esta forma un fenotipo ajustado a la solución de un determinado problema aportará la valoración más alta al individuo que representa.

El elemento *Red*, que representa a la red neuronal que configura al individuo, es considerada como un conjunto de elemento Neurona interconectadas entre sí mediante elementos *Conexión*.

Los elementos *Neurona* y *Conexión* deben ser adaptados al modelo neuronal que se implemente en cada aplicación, al igual que el elemento *Problema* debe caracterizar correctamente el tipo de problema para el que el algoritmo genético debe encontrar solución. Por último el conjunto de clases en que se especializa el elemento *Gen*, así como los genes concretos deben ser definidos de forma que permitan modelar el modelo neuronal usado sin aportar restricción alguna.

El significado concreto y comportamiento de los elementos presentados puede depender de la aplicación concreta que se haga del algoritmo genético. Un ejemplo de ello puede encontrarse en el elemento *Problema*, que aunque común a todas las aplicaciones su definición es distinta en cada uno de ellos.

La figura 4.2 muestra el conjunto de elementos, y relaciones entre ellos, que forman el modelo conceptual de la estructura común a las dos aplicaciones desarrolladas.

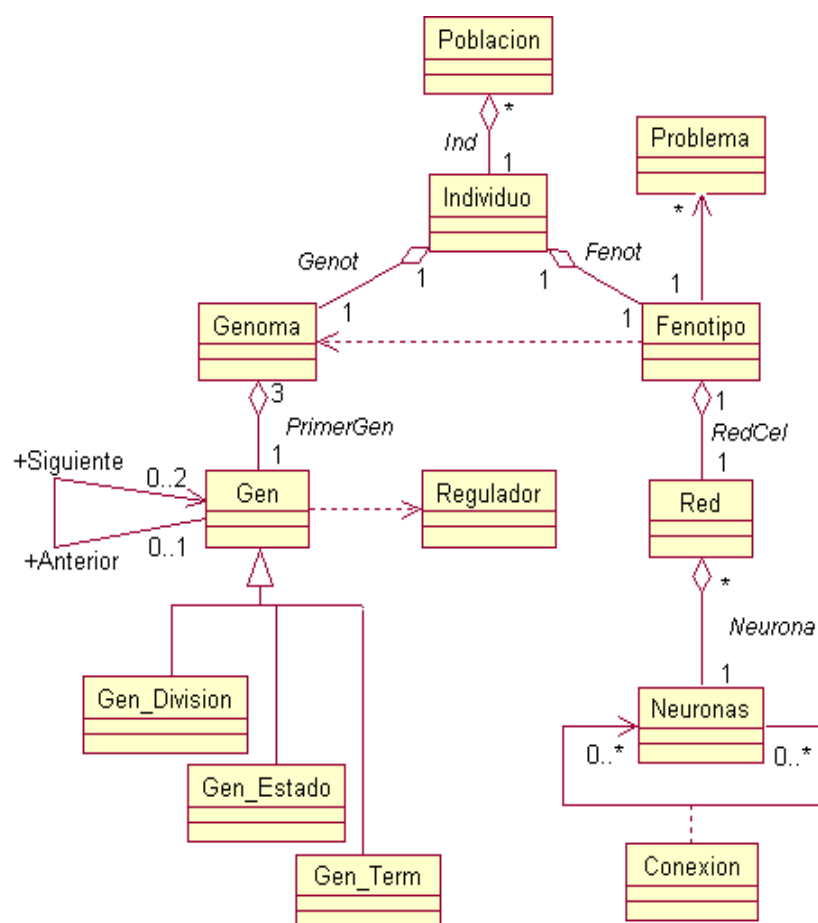


Fig. 4.2. Modelo conceptual correspondiente a la estructura común de las aplicaciones desarrolladas

Modelo conceptual. Adaptación al modelo neuronal binario

El modelo conceptual común, descrito en el apartado anterior, debe ser completado para la aplicación concreta que es modelada. El conjunto de genes disponibles en el algoritmo genético no se encuentra definido totalmente en el modelo conceptual descrito ya que está estrechamente relacionado con el modelo neuronal que se desea modelar y eso lo hace dependiente del modelo de neurona empleado. Por ello el conjunto de genes debe ser definido para un determinado modelo neuronal y no otro. Los elementos *Gen_División*, *Gen_Estado* y *Gen_Terminal* no describen genes como tal sino que representan una clasificación de genes común a ambas aplicaciones. El estudio del modelo neuronal binario utilizado ha dado lugar a

la configuración del conjunto de genes mostrado en la figura 4.3. En dicha figura se muestra la especialización del elemento *Gen_División* en los genes *Gen_PAR* y *Gen_SEQ*, genes que representan nodos de aridad 2 en el árbol genotípico e instrucciones de división en el desarrollo embrionario. El elemento *Gen_Estado* se especializa en tres genes, *Gen_VAL*, *Gen_BIAS* y *Gen_CUT*, que modifican el estado interno de las neuronas y sus conexiones con otras. Por último el elemento *Gen_Term* se especializa en un único gen, denominado *Gen_END*, que pone fin al desarrollo de una determinada neurona.

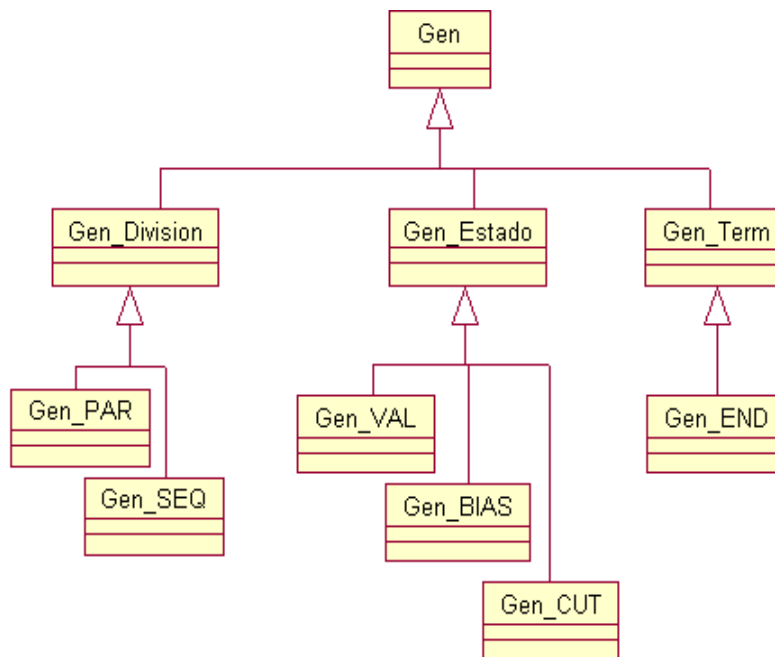


Fig. 4.3. Detalle de la adaptación del conjunto de genes en el Modelo Conceptual para el modelado de redes de neuronas binarias.

Modelo conceptual. Adaptación al modelo neuronal de integración y disparo

La aplicación del algoritmo genético al modelo de neurona de integración y disparo requiere el modelado del nuevo conjunto de genes y las nuevas características del problema que se pretende resolver. Al igual que para la aplicación del modelo binario de neurona, el conjunto de

genes debe ser definido y ajustado al modelo de neurona de integración y disparo. Los tipos básicos de genes *Gen_División*, *Gen_Estado* y *Gen_Term* son especializados en genes concretos para el nuevo modelo de neurona. El estudio del modelo neuronal de integración y disparo utilizado ha dado lugar a la configuración del conjunto de genes mostrado en la figura 4.4. En dicha figura se muestra la especialización del elemento *Gen_División* en los genes *Gen_PAR*, *Gen_SEQ* y *Gen_TOT*, genes que representan nodos de aridad 2 en el árbol genotípico e instrucciones de división en el desarrollo embrionario. El elemento *Gen_Estado* se especializa en tres genes, *Gen_VAL* y *Gen_CUT*, que modifican el estado interno de las neuronas y sus conexiones con otras. Por último el elemento *Gen_Term* se especializa en un único gen, denominado *Gen_END*, que pone fin al desarrollo de una determinada neurona.

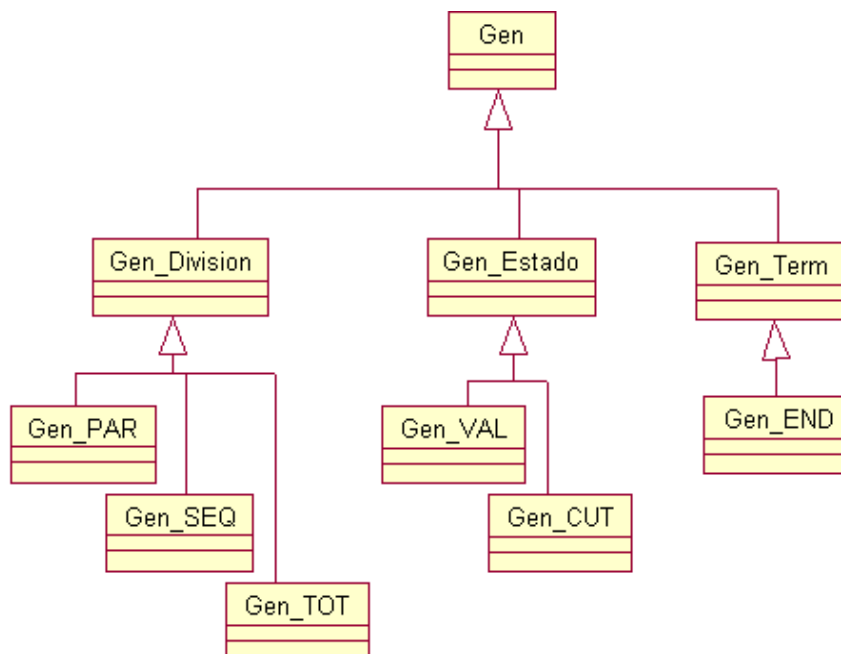


Fig. 4.4. Detalle de la adaptación del conjunto de genes en el modelo conceptual para el modelado de redes de neuronas de integración y disparo.

La repetida aparición de un nuevo e importante concepto, denominado *correlograma cruzado*, asociado al problema y a la actividad de las redes neuronales de integración y disparo hace necesaria su incorporación al modelo conceptual. Cada red neuronal genera un correlograma, que debe ser comparado con el correlograma de referencia, que forma parte del problema, para obtener la valoración final del individuo. El elemento *Problema* pasa a convertirse prácticamente en contenedor del correlograma de referencia con el que deben ser

comparado los generados por los individuos. La figura 4.5 muestra la inserción del nuevo elemento *Correlograma* y las relaciones que mantiene con los ya existentes.

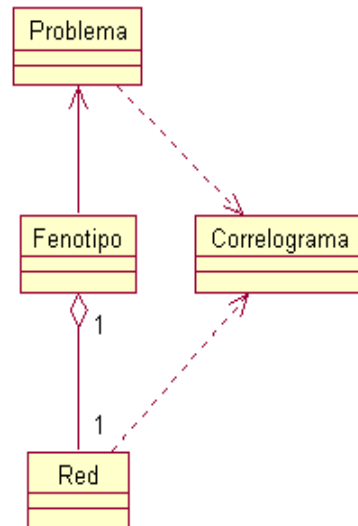


Fig. 4.5. Detalle de la inserción del elemento Correlograma y sus relaciones con los elementos Problema y Red.

4.3.2 Uso de las aplicaciones

Un objetivo tenido en cuenta durante el desarrollo de las aplicaciones realizadas ha sido el de permitir la mayor flexibilidad posible al experimentador para el modelado de una gran variedad de pruebas. Para ello se ha dotado a las aplicaciones de un completo conjunto de parámetros, diferentes para cada una de ellas, que permiten controlar todos los aspectos del uso del algoritmo genético, los individuos generados y los problemas a resolver. De igual forma toda la información relevante generada por la ejecución es mostrada y almacenada para permitir su posterior estudio. A continuación se describen los conjuntos de parámetros usados en las aplicaciones y la información que estas generan.

Codcel binario

La aplicación que implementa el algoritmo genético, basado en el modelo de *codificación celular adaptado* sobre redes de neuronas binarias para la resolución de problemas lógicos, es controlada por un conjunto de parámetros cuyos valores son determinados por el experimentador. La aplicación desarrollada dispone de una serie de funciones lógicas predefinidas sobre las que se centran la gran parte de las pruebas a realizar (“Contador2”, “Contador3”, “EXOR”, “Menos1”), e incluye la posibilidad de seleccionar funciones generadas aleatoriamente (“Automático”) haciendo necesario para ello indicar también el tamaño de entrada del mismo (tamaño de salida es invariablemente 1). El tamaño de los individuos que componen la población inicial son indicados mediante límites máximo y mínimos. El resto de parámetros son los propios del algoritmo genético indicando el tamaño de la población, el número de generaciones máximo y la elección de los métodos de selección empleados. A continuación se detallan la totalidad de parámetros que modelan una ejecución.

Parámetro	Valor	Descripción
FUNCIÓN	Cadena de caracteres	Nombre de la función a resolver.
TAMAÑO_POBLACIÓN	Entero	Número de individuos que forman la población.
NUM_GENERACIONES	Entero	Número de generaciones máximo.
TAM_MÁXIMO	Entero	Tamaño máximo de la red neuronal.
TAM_MÍNIMO	Entero	Tamaño mínimo de la red neuronal.
BALANCE	Entero	% de generaciones de <i>cruzamiento</i> .
PROB_MUT	Entero	% probabilidad de mutación de un gen.
MET_CRUZAMIENTO	Cadena de caracteres	Método de selección de <i>cruzamiento</i> usado.
MET_MUTACIÓN	Cadena de caracteres	Método de selección de <i>mutación</i> usado.
ARCH_SALIDA	Cadena de caracteres	Nombre del archivo generado.
LONG_MEJOR	Entero	Longitud del camino para buscar el mejor individuo *
LONG_PEOR	Entero	Longitud del camino para buscar el peor individuo *
TAM_ENTRADA	Entero	Tamaño de entrada del problema **

*(para “*isolating by distance*” o “*caminos aleatorios*”) ** (para funciones aleatorias)

El conjunto de parámetros es pasado como una lista de argumentos en la invocación de la aplicación y en el orden en que han sido expuestos. Un ejemplo de parámetros usados en una ejecución de la aplicación es el mostrado a continuación

> *codcelbin CONTADOR2 100 50 15 5 80 20 CROWDING SUSTPOSV salida_01.txt 5 5 10*

donde la función a resolver es “Contador de 2 bits”, y para lo que se define un tamaño de población de 100 individuos y 50 generaciones. El número de neuronas posibles para los individuos de la generación inicial oscila entre 5 y 15 neuronas. La proporción de generaciones de *cruzamiento* frente a generaciones de *mutación* es del 80% con lo que 2 de cada 10 generaciones son de *mutación* y el resto de *cruzamiento*. El método de selección para el cruzamiento elegido es *crowding*, y *sustitución positiva* para el de mutación. La probabilidad de mutación de cada gen se ha fijado en un 20%, por lo que cabe esperar que, en el proceso de mutación de un individuo, 2 de cada 10 genes resulten mutados. El fichero de salida generado para almacenar la información referente a las puntuaciones obtenidas por los individuos generados a lo largo de la ejecución tendrá como nombre “salida_01.txt”. El resto de argumentos no son relevantes para esta ejecución ya que hacen referencia a valores de parámetros propios de los métodos de selección *isolating by distance* y *camino aleatorios*, y al tamaño de entrada para problemas a resolver generados aleatoriamente.

La ejecución de la aplicación da lugar a un fichero donde figuran los valores de fitness de todos los individuos generados a lo largo de las generaciones. Esta información es utilizada posteriormente para realizar estudios estadísticos sobre la evolución de la población para los parámetros de entrada usados. De igual forma, durante la ejecución, se genera una salida por pantalla que informa de la generación actual en desarrollo y el número de sustituciones llevadas a cabo en ella. También se muestra en pantalla información sobre el problema a resolver, expresado a través de su *tabla de verdad*, y datos sobre el individuo solución como son: su genoma, el número de neuronas que lo componen y su matriz de conexionado. La aplicación se detiene en el momento en que encuentra un individuo solución o es alcanzado el número máximo de generaciones establecido.

Codcel integración y disparo

La aplicación que implementa el modelo de *codificación celular adaptado* sobre redes de neuronas de integración y disparo, posee un mayor número de conceptos parametrizables que en el caso anterior. El propio modelo neuronal requiere de un conjunto propio de parámetros relacionados con la actividad de los distintos tipos de neuronas (*driver* y *no driver*) o el tipo de distribución del ruido empleado en ellas. La generación de correlogramas cruzados, para la valoración de individuos, requiere igualmente un conjunto de parámetros que permitan su definición y el cálculo de la distancia con el usado de referencia.

Dado el gran número de parámetros necesarios para definir totalmente una ejecución se ha considerado oportuno reunirlos en un fichero de configuración cuyo nombre es pasado como argumento en la ejecución. A continuación se detallan los parámetros incluidos en el fichero organizados según el ámbito de su aplicación.

Parámetros de caracterización de la Neurona

Parámetro	Valor	Descripción
TIPO_RUIDO	Cadena de caracteres	Tipo de Ruido Neuronal (<i>Uniforme</i> o <i>Gaussiano</i>)
MEDIA_RUIDO_DRIVER	Real	Media de Ruido para neurona <i>Driver</i>
MEDIA_RUIDO_NODRIVER	Real	Media de Ruido para neurona <i>No Driver</i>
DESV_RUIDO_DRIVER	Real	Radio/Desviación de ruido para neuronas <i>Driver</i>
DESV_RUIDO_NODRIVER	Real	Radio/Desviación de ruido para neuronas <i>No Driver</i>
MIN_CONEXIÓN	Real	Peso mínimo de una conexión
MAX_CONEXIÓN	Real	Peso máximo de una conexión

Parámetros para la generación de Individuos y Correlogramas

Parámetro	Valor	Descripción
NUM_NEURO_MAX	Entero	Tamaño máximo de la nube de neuronas
NUM_NEURO_MIN	Entero	Tamaño mínimo de la nube de neuronas
NUM_DISPAROS_N2	Entero	Número de disparos de N2 requeridos

NUM_PASOS	Entero	Numero Máximo de pasos de simulación
REGULADOR	Vector 6 Enteros	Probabilidad de aparición de genes
INIC_CONEXIÓN	Real	Peso de las conexiones iniciales.
TAM_VENTANA	Entero	Tamaño de la ventana del correlograma
VECTORPESOS	Vector 6 Enteros	Vector de pesos para la caracterización del correlograma

Algoritmo Genético

Parámetro	Valor	Descripción
TAM_POBLACION	Entero	Número de individuos en la población
NUM_GENERACIONES	Entero	Número máximo de generaciones
MET_SELCRUCE	Cadena de caracteres	Método de selección de <i>cruzamiento</i>
MET_SELMUTAC	Cadena de caracteres	Método de selección de <i>mutación</i>
BALANCE	Entero [0-100]	% de generaciones de <i>cruzamiento</i>
PROB_MUT	Entero [0-100]	% probabilidad de mutación de un gen
LONG_MEJOR	Entero	Longitud del camino para buscar el mejor individuo *
LONG_PEOR	Entero	Longitud del camino para buscar el peor individuo *

* (para "isolating by distance" o "caminos aleatorios")

Entrada / Salida

Parámetro	Valor	Descripción
ARCH_CORRELOG	Cadena de caracteres	Archivo con el correlograma de referencia
ARCH_SALIDA	Cadena de caracteres	Nombre de los archivos generados

La ejecución de la aplicación requiere, además del fichero de configuración, la existencia de un fichero que contenga el correlograma cruzado usado como referencia y cuyas características se pretenden emular. Un ejemplo de llamada a la aplicación cuyos parámetros están contenidos en un fichero llamado "*config.txt*" es el mostrado a continuación

```
> codcel config.txt
```

La ejecución de la aplicación da lugar a un conjunto de ficheros que contienen toda la información relevante de las pruebas realizadas. En el fichero “[nombre]_Generac.txt” se almacena los *fitness* de todos los individuos pertenecientes a la población a lo largo de las *N* generaciones. El fichero “[nombre]_Sustit.txt” recoge el número de individuos sustituidos en cada una de las generaciones. Los ficheros “[nombre]_Genoma.txt”, “[nombre]_Conexion.txt” y “[nombre]_Correlograma.txt” contienen el genoma del individuo con una valoración más alta, la matriz de conexionado de la red neuronal que desarrolla y el correlograma cruzado obtenido, respectivamente. De igual forma, durante la ejecución, se genera una salida por pantalla que informa del desarrollo de las pruebas

4.4 Recursos empleados

Casi la totalidad del proyecto ha sido realizado utilizando la equipación aportada por el IPI (Instituto de Procesamiento de Imagen) perteneciente a la UMA. Los recursos utilizados han consistido en un PC Pentium III con conexión a Internet via LAN y sistema operativo Linux (*RedHat 7.0*).

Las aplicaciones software realizadas han sido desarrolladas con la herramienta de gestión de proyectos software *Kdevelop 1.2* y el compilador *gcc* (C++).

Las estadísticas realizadas y sus respectivas gráficas, incluidas en este documento, han sido generadas con *Matlab 6.0* a partir del procesamiento de la información generada en las diversas pruebas.

El desarrollo de los modelos UML correspondientes a las etapas de análisis y diseño han sido realizados con la herramienta *Rational Rose 2000* (sobre Windows 2000).

La elaboración del presente documento ha sido realizado íntegramente en *Word 2000*, y las referencias bibliográficas contenidas en él almacenadas y gestionadas con *Procite 5.0*.

Capítulo 5

Conclusiones y trabajos futuros

La optimización de circuitos neuronales artificiales ha recaído tradicionalmente en el uso de reglas de aprendizaje que modifican el conexionado y los parámetros internos de las neuronas que los componen. Estas reglas no siempre son aplicables en modelos de neuronas complejos o que reproducen características biológicas. El presente proyecto propone el uso de algoritmos genéticos en la optimización de redes neuronales de integración y disparo. Para ello se ha realizado un amplio estudio de las propuestas realizadas en este sentido en la literatura actual, y se ha seleccionado el modelo *codificación celular* como el mejor candidato para su implementación y estudio. Este modelo de codificación implícita, y el proceso de desarrollo embrionario asociado, se ha mostrado enormemente flexible para el modelado de los más diversos paradigmas neuronales.

Un problema patente en el uso de algoritmo genéticos, donde los individuos generados poseen cierto nivel de complejidad, es el de la *disrupción*. Este fenómeno, especialmente asociado al operador cruzamiento, hace que se generen individuos no validos o con una baja valoración a partir de individuos con características bien valoradas. Las técnicas de *especiación*, que limitan el cruzamiento a individuos con características similares (considerados de la misma especie), no parecen adecuadas para su uso con redes neuronales, por cuestiones de eficiencia computacional. Por ello, los esfuerzos para evitar la *disrupción* se han centrado en la estructuración de los genomas, y el uso de reglas en la generación de individuos por recombinación. Esto, si bien ha reducido en gran medida la aparición de este fenómeno, no lo ha evitado por completo. Ha sido necesario el uso de políticas de selección de individuos que no permiten la sustitución de individuos por otros con una peor valoración, eliminando así el fenómeno de la *disrupción a posteriori*.

La primera aplicación del algoritmo genético desarrollada pretendía evaluar el potencial del algoritmo, el modelo de codificación y los métodos de selección, en la búsqueda de redes neuronales binarias que resuelven problemas lógicos. Los resultados de las pruebas realizadas han mejorado obtenidos por otros autores, como es el caso de los resultados obtenidos con *programación genética* por J.Koza.

La segunda aplicación del algoritmo genético se basaba en un modelo neuronal de integración y disparo y su aplicación para la búsqueda de conexionados alternativos a uno propuesto por el experimentador. El uso de un modelo neuronal de *integración y disparo*, mucho más complejo que el usado para el caso binario y con características estocásticas han requerido un estudio por si mismo que permitiera establecer valores óptimos para sus parámetros. Dicho estudio del modelo neuronal a generado interesantes resultados. En primer lugar se han fijado varios conjuntos de valores que producen unos determinados niveles de actividad neuronal. Han sido establecidos los límites entre que no deben superar los pesos sinápticos para mantener unos correctos valores de eficacia y contribución sináptica en los circuitos neuronales. Por último, el resultado más importante obtenido en dicho estudio ha sido la equivalencia demostrada entre determinados valores de los parámetros de modelos de ruido con distribución *uniforme* y *gaussiana* empleados. Esta equivalencia permite hacer uso de un modelo de ruido con baja carga computacional (distribución uniforme) y considerarlos como obtenidos por un modelo de ruido con mayor carga (distribución *gaussiana*), con el consiguiente ahorro computacional.

El objetivo final del proyecto ha consistido en la aplicación del modelo de *codificación celular adaptado* en la optimización de conexionados de redes de *neuronas de integración y disparo*. Las pruebas sobre el modelo han consistido en la aplicación de las distintas políticas de selección y sustitución de individuos en la búsqueda de conexionados alternativos al monosináptico. Los individuos mejor valorados de las distintas pruebas realizadas son aproximaciones al individuo objetivo, pero distan lo suficiente de él como para no poder ser considerados alternativos. Los resultados obtenidos confirman la tesis de que estos conexionados no son posibles, es decir, no existen circuitos neuronales para los que dos neuronas determinadas exhiban un comportamiento idéntico al presentado por dos neuronas conectadas monosinápticamente, al menos para el modelo neuronal utilizado de integración y disparo y las limitaciones que incluye. Las diferentes aproximaciones realizadas por el algoritmo demuestran la correcta y exhaustiva búsqueda, llevada a cabo por el algoritmo genético, en el complejo espacio de soluciones.

Cabe destacar por último la importancia que han tomado las políticas de selección y sustitución de individuos en los resultados obtenidos en ambas adaptaciones, llegando a ser un factor decisivo a la eficacia del algoritmo.

Trabajos Futuros

Son numerosas las líneas que quedan abiertas en el presente trabajo para ampliaciones futuras. El modelo neuronal de integración y disparo empleado es relativamente simple y no captura características complejas deseables como son los *EPSP*, *delays* o *jitters*, entre otros. La inclusión de un modelo neuronal con un mayor nivel de complejidad que incluya dichas características, y la adaptación del modelo de codificación asociado, puede ser una posible vía de trabajo futuro. De igual forma, sería posible la inclusión de nuevos métodos de caracterización de correlogramas cruzados que permitan definir nuevas características a tener en cuenta para la evaluación de individuos. Es posible, igualmente, un estudio de la aplicación de distintas funciones de distancia entre correlogramas que permita una mayor optimización de los conexiones obtenidos. Por otro lado, el uso de nuevas técnicas de generación de la población inicial, como sería el caso del uso de semillas, o nuevas técnicas de selección y sustitución de individuos, podría dar lugar a mejoras en los resultados. Por último, y dada la importancia de la eficiencia computacional en aplicaciones de las escalas empleadas, sería conveniente el estudio de la posible optimización del modelo neuronal empleado, que permita la realización de ejecuciones en un menor tiempo.

Bibliografía

1. Angeline, P. J., Saunders, G. M., & Pollack, J. B. (1994). An evolutionary algorithm that construct recurrent neural networks. IEEE Trans. Neural Networks, 5(1), 54-65.
2. Bäck, T. (1996). Evolutonary Algorithms in Theory and Prattice. New York: Oxford University Press.
3. Blicke, T. (1997). Tournament selection. T. Bäck, D. B. Fogel, & Z. Michalewicz (Handbook of Evolutionary Computation). New York: Oxford University Press.
4. Box, G. E. P. (1957). Evolutionary operation: A method for increasing industrial productivity. Applied Statistics, 6, 81-101.
5. Box, G. E. P., & Draper, N. R. (1969). Evolutionary operation. A statistical method for process improvement. John Wiley & Sons.
6. Bremmerman, H. J. (1962). Optimization through evolution and recombination. Self-organizing Systems. Washington, D.C.: Spartan Books.
7. Bryson, A., & Ho, Y.-C. (1969). Applied Optimal Control. New York: Blaisdell Publishing Co.
8. Carpenter, G. A., & Grossberg, S. (1987). A massively parallel architecture for a self-organizing neural pattern recognition machine. Computer Vision, Graphics, and Image Processing, 37, 54-115.
9. Darwin, C. (1859). The origin of species by means of natural selection. London: John Murray.

10. Dasgupta, D., & McGregor, D. (1994). A more biologically motivated algorithm: the model and some results. Cybernetics and Systems, 25(3), 447-69.
11. De Jong, K. A. (1975). An Analysis of the Behavior of a Class of Genetic Adaptive Systems. Doctoral Dissertation, University of Michigan, 36.
12. Eldredge, N. (1989). Macro-evolutionary Dynamics: Species, Niches and Adaptive Peaks. New York: McGraw-Hill.
13. Eldredge, N., & Gould, S. J. (1972). Punctuated equilibria: an alternative to phyletic gradualism. T. J. M. Schopf Models of Paleobiology (pp. 82-115). San Francisco: Freeman, Cooper.
14. Fisher, R. A. (1930). The Genetical Theory of Natural Selection. Oxford: Clarendon Press.
15. Flemming, W. (1882). Zellsubstanz, Kern und Zelltheilung (Cell Substance, Nucleus, and Cell Division). Leipzig: FCW Vogel.
16. Floyd, R. (1967). Assigning meaning to programs. Mathematical Aspects of Computer Science (pp. 19-31). American Mathematical Society.
17. Fogel, L. J., Owens, A. J., & Walsh, M. J. (1966). Artificial Intelligence through Simulated Evolution. New York: NY: John Wiley.
18. Friedberg, R. M. (1958). A Learning Machine: Part I. IBM Journal, January 1958.
19. Fukushima, K. (1988). Neocognitron: A Hierarchical Neural Network Capable of Visual Pattern Recognition. Neural Networks, 1.
20. Goldberg, D. (1989). Genetic algorithms in search, optimization, and machine learning. Addison-Wesley Publishing Company, Inc.
21. Goldberg, D., Korb, B., & Deb, K. (1990). Messy genetic algorithms: Motivation, analysis

and first results. Complex Systems, 3, 493-530.

22. Goldberg, D., & Richardson, J. (1987). Genetic algorithms with sharing for multimodal function optimization. 2nd. Int. Conf. on Genetic Algorithms Cambridge: Lawrence Erlbaum Associates.
23. Grefenstette, J. (1997a). Proportional selection and sampling algorithms. T. Bäck, D. B. Fogel, & Z. Michalewicz (Handbook of Evolutionary Computation). New York: Oxford University Press.
24. Grefenstette, J. (1997b). Rank-based selection. T. Bäck, D. B. Fogel, & Z. Michalewicz (Handbook of Evolutionary Computation). New York: Oxford University Press.
25. Grossberg, S. (1987). Competitive learning: from interactive activation to adaptative resonance. Cognitive Science, 11, 23-63.
26. Gruau, F. (1994). Neural Network Synthesis using Cellular Encoding and the Genetic Algorithm. Thesis Doctoral, L'Universite Calude Bernard-Lyon I.
27. Gruau, F. (1995). Automatic definition of modular neural networks. Adaptive Behavior, 3(2), 151-183.
28. Hebb, D. O. (1949). The organization of behaviour. New York: J. Wiley & Sons.
29. Heims, S. J. (1980). John Von Neumann and Norbert Wiener: From Mathematics to the Technologies of Life and Death. Cambridge: MIT Press.
30. Hinton, G., & Sejnowski, T. (1984). Boltzmann machines: Constraint satisfaction networks that learn. (Report No. CMU-CS-84-119). Pittsburgh:
31. Hodking, A. L., & Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. Journal of Physiology, 117, 500-544.

32. Holland, J. H. (1975). Adaptation in Natural and Artificial Systems. Ann Arbor: MA: MIT Press.
33. Hopfield, J. J. (1982). Neural Networks and Physical Systems with Emergent Collective Computational Abilities. National Academy of Sciences. USA, 79, 2554-2558.
34. Hopfield, J. J., & Tank, D. W. (1985). Neural Computation of decisions in optimization problems. Biological Cybernetics, 52.
35. Hubel, D. H., & Weisel, T. N. (1962). Receptive Fields, Binocular Interaction, and Functional Architecture in the Cat's Visual Cortex. Journal of Physiology, 160, 106-123.
36. Jacob, F., & Monod, J. (1961). Genetic Regulatory Mechanics in the Synthesis of Proteins. Journal of Molecular Biology, 3, 318-356.
37. Kandel, R., Schwartz, H., & Jessell, T. M. (1999). Essentials of Neural Science and Behavior. McGraw Hill.
38. Kitano, H. (1990). Designing Neural Networks using Genetic Algorithms with Graph Generation Systems. Complex Systems, 4, 461-476.
39. Kohonen, T. (1984). Self-Organization and associative memory. Springer Verlag.
40. Koza, J., & Rice, J. P. (1991). Genetic Generation of Both the Weights and Architecture for a Neural Network. International Joint Conference on Neural Networks, 397-404.
41. LeCun, Y. (1986). Learning processes in an asymmetric threshold network. E. Bienenstock, F. Fogelman-Soulié, & G. Weisbuch Disordered systems and biological organization (pp. 233-240). Les Houches, France: Springer-Verlag.
42. Little, W. A. (1974). The existence of persistent states in the brain. Mathematical Biosciences, 19(101), 120.

43. MacGregor, R. J. (1987). Neuronal and brain modelling. New York: Academic Press, Inc.
44. Mahfoud, S. W. (1992). Crowding and preselction revisited. R. Männer, & B. Manderick
Parallel Problem Solving From Nature (Vol. 2pp. 27-36). Amsterdam: Elsevier.
45. Mahfoud, S. W. (1997). Niching methods. T. Bäck, D. B. Fogel, & Z. Michalewicz
(Handbook of Evolutionary Computation). New York: Oxford University Press.
46. Mandelbrot, B. B. (1977). Fractals: Form, Chance, and Dimension. San Francisco: W.H.
Freeman.
47. McCulloch, W. S., & Pitts, W. (1943). A Logical Calculus of the Ideas Immanent in
Nervous Activity. Bulletin of Mathematical Biophysics, 7, 115-133.
48. Mendel, G. (1865). Experiments in Plant Hybridisation. London: Oliver and Boyd.
49. Mühlenbein, H. (1991). Darwin's continent cycle theory and its simulation by the
prisoners dilemma. Complex System, 459-478.
50. Müller, B., & Reinhardt, J. (1990). Neural Networks. An introduction. Springer-Verlag.
51. Parker, D. B. (1982). (Report No. 581-64). Stanford University: Office of Technology
Licensing.
52. Rechenberg, I. (1965). Cybernetic Solution Path of an Experimental Problem. Royal
Aircraft Establishment, 1122.
53. Reed, J., Tooms, R., & Baricelli, N. (1967). Simulation of Biological Evolution and
Machine Learning. J. Theo. Biol., 17, 319-342.
54. Rosenblatt, F. (1962). Principle of Neurodynamics: Perceptron and the theory of Brain
Mechanisms. Washington, D.C.: Spartan Books.

55. Rumelhart, E., & McClelland, J. L. (1986). Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Cambridge: MIT Press.
56. Schaffer, J. D., Whitley, L. D., & Eshelman, L. J. (1992). Combinations of genetic algorithms and neural networks: A survey of the state of the art. International Workshop on Combinations of Genetic Algorithms and Neural Networks California: IEEE Computer Society Press.
57. Schwefel, H. P. (1965). Kybernetische Evolution als Strategie der Experimentellen Forschung in der Stromungstechnik. Diploma Thesis Technical, University of Berlin.
58. Shepherd, G. M. (1990). The synaptic organization of the brain. Oxford Press.
59. Stein, R. B. (1967). Some models of neuronal variability. Biophysical Journal, 7, 37-68.
60. Veredas, F. J., Vico, F. J., & Guillén, M. J. (2000). Genetic Algorithms with Implicit Encoding for Solving High-Dimensional Optimization Problems. RealLife. Evolutionary Design Optimization. PPSN' 2000 Paris.
61. Vico, F. J. (1995). Concepción y modelado de un paradigma biológico de redes neuronales artificiales. Tesis Doctoral, Universidad De Málaga.
62. von Neuman, J. (1966). Theory of Self-Reproducing Automata. Langton: University of Illionios Press.
63. Wallace, A. R. (1858). On the tendency of varieties to depart indefinitely from the original type. Ternate.
64. Wallace, A. R. (1870). Contributions to the Theory of Natural Selection. A Series of Essays. London & New York: Macmillan & Co.
65. Watson, J. D., & Crick, F. H. C. (1953). Molecular Structure of Nucleic Acids. A Structure for Deoxyribose Nucleic Acid. Nature, 171, 737.

66. Werbos, P. (1974). Beyond Regression: New Tools for Predictions and Analysis in the Behavioral Science. Ph. D. Thesis, Harvard University.
67. Widrow, B., & Hoff, M. E. (1960). Adaptive switching circuits. Ire Wescon Conv. Rec., 4, 96-104.