



UNIVERSIDADE DA CORUÑA

FACULTAD DE INFORMÁTICA

DEPARTAMENTO DE
TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

**COMPUTACIÓN EVOLUTIVA PARA EL
PROCESO DE SELECCIÓN DE VARIABLES EN
ESPACIOS DE BÚSQUEDA MULTIMODALES**

Tesis Doctoral

Directores

Dr. Julián Dorado de la Calle

Dr. Juan Ramón Rabuñal Dopico

Doctorando

Marcos Gestal Pose

A Coruña, Marzo 2009

D. **JULIÁN DORADO DE LA CALLE**, Profesor Titular de Universidad en el área Ciencias de la Computación e Inteligencia Artificial de la Universidade da Coruña y D. **Juan Ramón Rabuñal Dopico**, Profesor Contratado Doctor en el área Ciencias de la Computación e Inteligencia Artificial de la Universidade da Coruña, pertenecientes al Departamento de Tecnologías de la Información y las Comunicaciones

HACEN CONSTAR QUE:

La memoria **“COMPUTACIÓN EVOLUTIVA PARA EL PROCESO DE SELECCIÓN DE VARIABLES EN ESPACIOS DE BÚSQUEDA MULTIMODALES”** ha sido realizada por D. **MARCOS GESTAL POSE**, bajo nuestra dirección en el Departamento de Tecnologías de la Información y las Comunicaciones, y constituye la Tesis que presenta para optar al Grado de Doctor en Informática de la Universidade da Coruña.

A Coruña, 29 de Enero de 2009

Fdo: Dr. Julián Dorado de la Calle

Fdo: Dr. Juan Ramón Rabuñal Dopico

Agradecimientos

En primer lugar agradecer sinceramente a mis directores de Tesis, Julián Dorado y Juan Ramón Rabuñal, los sabios consejos y la dedicación prestada sin las cuales difícilmente habría sido posible la realización de este trabajo.

A todos los miembros del grupo RNASA-IMEDIR, desde su director hasta la última persona en llegar, por todo el apoyo brindado de manera desinteresada y por lograr que ir a trabajar todos los días se convierta en un placer.

A Julia por sus impagables e incontables gestiones.

Gracias a todos los miembros del IEETA, en Aveiro, por todo lo aprendido durante el tiempo que he pasado con ellos.

No puedo dejar de agradecer a Noemí sus continuas palabras de ánimo, su contagioso optimismo, su simple presencia... en los buenos y en los malos momentos. Y a Iria que, únicamente con una sonrisa, logra que se desvanezcan todos los problemas.

A mis padres, por la paciencia, la ayuda y el apoyo durante tantos años. A ellos, que a buen seguro disfrutarán tanto o más que yo de este momento.

A todos aquellos que, de una u otra manera, han hecho posible que el presente trabajo haya llegado a buen puerto, muchas gracias.

A mi familia

*Algo he aprendido en mi larga vida:
que toda nuestra ciencia,
contrastada con la realidad,
es primitiva y pueril;
y, sin embargo,
es lo más valioso que tenemos"*

Albert Einstein

Índice de contenido

<i>Introducción</i>	1
1.1 Objetivos	4
1.2 Estructura de la tesis	6
<i>Fundamentos</i>	9
2.1 Selección de Variables	9
2.1.1 Definición	10
2.1.2 Ventajas.....	12
2.1.3 Clasificación de Técnicas de Selección de Variables.....	13
2.1.3.1 Filter.....	13
2.1.3.2 Wrapper	15
2.1.3.3 Embedded	16
2.1.4 Estrategias de Búsqueda para Selección de Variables.....	17
2.1.4.1 Exponenciales	17
2.1.4.2 Secuenciales	18
2.1.4.3 Estocásticas	19
2.2 Selección Multimodal de Variables	20
2.2.1 Problemas Multimodales	20
2.2.1.1 Definición.....	20

2.2.1.2	Problemas Multimodales Clásicos.....	21
2.2.1.2.1	Función de Ackley.....	21
2.2.1.2.2	Función de Rastrigin.....	22
2.2.1.2.3	Función de Schwefel.....	23
2.3	Algoritmos Genéticos.....	24
2.3.1	Orígenes. Computación Evolutiva.....	24
2.3.2	Bases Biológicas.....	28
2.3.3	Codificación de Problemas.....	28
2.3.4	Algoritmo Principal.....	30
2.3.5	Operadores Genéticos.....	33
2.3.5.1	Selección.....	34
2.3.5.1.1	Selección por Ruleta.....	34
2.3.5.1.2	Selección por Torneo.....	35
2.3.5.2	Cruce.....	36
2.3.5.2.1	Cruce de 1 Punto.....	38
2.3.5.2.2	Cruce de 2 Puntos.....	38
2.3.5.2.3	Cruce Uniforme.....	39
2.3.5.2.4	Cruces Específicos de Codificaciones no Binarias.....	40
2.3.5.3	Algoritmos de Reemplazo.....	41
2.3.5.4	Copia.....	42
2.3.5.5	Mutación.....	42
2.3.6	Evaluación.....	44

2.4 Redes de Neuronas Artificiales	47
2.4.1 Fundamentos y Conceptos Básicos.....	51
2.4.2 Función de activación.....	53
2.4.3 Tipos de neuronas	54
2.4.4 Flujo de datos en las Redes de Neuronas Artificiales	55
2.4.4.1 Redes alimentadas hacia delante.....	55
2.4.4.2 Redes con Retroalimentación total o parcial.....	57
2.4.5 Información en las Redes de Neuronas Artificiales.....	57
2.4.6 Tipos de problemas.....	60
2.4.7 Fase de Entrenamiento	60
2.4.7.1 Aprendizaje supervisado.....	62
2.4.7.2 Aprendizaje no supervisado	63
2.4.8 Selección del juego de ensayo.....	64
2.4.9 Fase de ejecución	65
<i>Estado de la Cuestión.....</i>	67
3.1 Métodos Matemáticos de Selección de Variables.....	68
3.1.1 Análisis de Componentes Principales	68
3.1.2 PLS.....	70
3.1.3 K-means	73
3.1.4 Limitaciones	74
3.2 Métodos Evolutivos de Selección de Variables.....	75
3.2.1 Algoritmos Genéticos Clásicos.....	75
3.2.2 Messy Genetic Algorithms.....	77

3.2.3 Algoritmos de Estimación de Distribuciones	79
3.2.4 Limitaciones	83
3.3 Selección Multimodal de Variables: Tratamiento de la Multimodalidad con Algoritmos Genéticos.....	83
3.3.1 Fitness Sharing.....	87
3.3.2 Fitness Scaling.....	90
3.3.3 Crowding.....	91
3.3.4 Clearing	93
3.4 Motivación.....	94
<i>Hipótesis de Trabajo</i>	<i>96</i>
<i>Selección de Variables en Espacios de Búsqueda Multimodal Mediante Técnicas de Computación Evolutiva</i>	<i>98</i>
5.1 Metodología Propuesta	98
5.2 Tratamiento de la multimodalidad	103
5.2.1 Algoritmo Genético de Dos Poblaciones.....	104
5.2.1.1 Introducción	104
5.2.1.2 Esquema General de Funcionamiento	105
5.2.1.3 Descripción del Sistema	107
5.2.1.3.1 Piscina Genética.....	107
5.2.1.3.2 Población Secundaria.....	109
5.2.1.3.3 Operadores Genéticos.....	110
5.2.2 Algoritmo Genético de Dos Poblaciones con Regiones Variables.....	115
5.2.2.1 Introducción	115

5.2.2.2	Genes Activos y no Activos.....	117
5.2.2.3	Agrupación de Subregiones de Búsqueda.....	121
5.2.2.4	División de Subregiones de Búsqueda.....	125
5.2.2.5	Estabilidad Subregiones.....	127
5.2.3	Agrupación en Especies con Regiones de Rechazo	128
5.2.3.1	Exploración del espacio de búsqueda mediante especies	129
5.2.3.2	Regiones de rechazo	131
5.2.3.3	Algoritmo Genético basado en especies con regiones de rechazo	132
5.2.3.3.1	Creación de especies	133
5.2.3.3.2	Obtención de los centroides	134
5.2.3.3.3	Creación de las regiones de rechazo	135
5.2.3.3.4	Maximización vs. Minimización	136
5.2.3.3.5	Aplicación de los operadores genéticos	138
5.2.3.3.6	Ubicación de la descendencia	138
5.2.3.3.7	Eliminación de especies.....	140
5.2.3.3.8	Modificación del tamaño de las especies	141
5.2.3.3.9	Esquema general de funcionamiento.....	142
	Resultados	145
	6.1 Introducción	145
	6.2 Funciones Clásicas	147
6.2.1	Configuraciones iniciales	147
6.2.2	Localización de soluciones globales y parciales.....	150
6.2.2.1	Función de Ackley	150

6.2.2.2	Función de Rastrigin	155
6.2.2.3	Función de Schewefel.....	157
6.3	Clasificación de Zumos Naturales	159
6.3.1	Introducción.....	159
6.3.2	Clasificación de muestras.....	162
6.3.3	Descripción de los datos.....	164
6.3.4	Selección de Variables.....	166
6.3.4.1	Modelos de Referencia	168
6.3.4.1.1	Clasificación con la totalidad de variables.....	168
6.3.4.1.2	Clasificación con variables seleccionadas por Procrustes Rotation.....	172
6.3.4.2	Selección multimodal de variables.....	174
6.3.4.2.1	Selección multimodal de variables mediante Algoritmo Genético de dos Poblaciones	177
6.3.4.2.2	Selección multimodal de variables mediante Algoritmo Genético con Agrupación en Especies	179
6.3.4.3	Consideraciones acerca de las pruebas.....	181
	Conclusiones.....	182
	Futuros Desarrollos.....	186
	Bibliografía.....	189

Índice de ilustraciones

<i>Figura 2.1: Selección de Variables: Técnica Filter</i>	14
<i>Figura 2.2: Selección de Variables: Técnica Wrapper</i>	15
<i>Figura 2.3: Representación 3D y 2D de una función multimodal tipo</i>	21
<i>Figura 2.4: Funciones Multimodales: Función de Ackley</i>	22
<i>Figura 2.5: Funciones Multimodales: Función de Rastrigin</i>	22
<i>Figura 2.6: Funciones Multimodales: Función de Schwefel</i>	23
<i>Figura 2.7: Ecuación Evolutiva</i>	25
<i>Figura 2.8: Soft Computing</i>	26
<i>Figura 2.9: Individuo genético binario</i>	29
<i>Figura 2.10: Codificación de una red neuronal</i>	30
<i>Figura 2.11: Pseudocódigo algoritmo genético</i>	31
<i>Figura 2.12: Cruce de 1 punto</i>	38
<i>Figura 2.13: Cruce de 2 puntos</i>	39
<i>Figura 2.14: Cruce uniforme</i>	40
<i>Figura 2.15: Esquema básico del trabajo con Redes de Neuronas Artificiales</i>	48
<i>Figura 2.16: Neurona Artificial</i>	52
<i>Figura 2.17: Función de transferencia umbral</i>	55
<i>Figura 2.18: Función de transferencia sigmoide</i>	55
<i>Figura 2.19: Función de transferencia hiperbólica-tangente</i>	55

<i>Figura 2.20: RNA alimentada hacia delante.....</i>	<i>56</i>
<i>Figura 2.21: RNA con retroalimentación</i>	<i>57</i>
<i>Figura 3.1: Homogeneización de la población genética.....</i>	<i>84</i>
<i>Figura 3.2: Evolución del ajuste ante una situación de homogeneización</i>	<i>85</i>
<i>Figura 5.1: Evaluación mediante redes de neuronas artificiales</i>	<i>101</i>
<i>Figura 5.2: Ejemplo de configuración de la piscina genética</i>	<i>108</i>
<i>Figura 5.3: Estructura de un individuo de la piscina genética</i>	<i>109</i>
<i>Figura 5.4: Construcción del representante de la piscina genética</i>	<i>111</i>
<i>Figura 5.5: Operador de Cruce</i>	<i>112</i>
<i>Figura 5.6: Pseudocódigo de la operación de mutación.....</i>	<i>113</i>
<i>Figura 5.7: Estructura modificada de un individuo de la piscina genética.....</i>	<i>118</i>
<i>Figura 5.8: Piscina genética y mejor solución</i>	<i>119</i>
<i>Figura 5.9: Modificaciones en la piscina genética tras mejorar solución actual</i>	<i>120</i>
<i>Figura 5.10: Agrupación de subregiones: Paso 0.....</i>	<i>124</i>
<i>Figura 5.11: Agrupación de subregiones: Paso 1.....</i>	<i>125</i>
<i>Figura 5.12: División de subregiones: Paso 0.....</i>	<i>126</i>
<i>Figura 5.13: División de subregiones: Paso 1.....</i>	<i>127</i>
<i>Figura 5.14: Centroide de una especie</i>	<i>134</i>
<i>Figura 5.15: Tamaño de la región de rechazo.....</i>	<i>136</i>
<i>Figura 5.16: Maximización vs. Minimización</i>	<i>137</i>
<i>Figura 5.17: Diagrama general de funcionamiento AG basado en especies.....</i>	<i>143</i>
<i>Figura 5.18: Pseudocódigo detallado AG basado en especies</i>	<i>144</i>

<i>Figura 6.1: Ackley: Evolución del número de soluciones encontradas</i>	<i>151</i>
<i>Figura 6.2: Algoritmo Genético de 2 Poblaciones: Número de soluciones encontradas sin el control de subregiones</i>	<i>152</i>
<i>Figura 6.3: Ackley: Evolución del error</i>	<i>153</i>
<i>Figura 6.4: Ackley: Porcentaje de soluciones encontradas tras 100 generaciones</i>	<i>154</i>
<i>Figura 6.5: Rastrigin: Evolución del número de soluciones encontradas</i>	<i>156</i>
<i>Figura 6.6: Rastrigin: Evolución del error</i>	<i>156</i>
<i>Figura 6.7: Rastrigin: Porcentaje de soluciones encontradas tras 100 generaciones</i>	<i>157</i>
<i>Figura 6.8: Schewefel: Evolución del número de soluciones encontradas</i>	<i>158</i>
<i>Figura 6.9: Schewefel: Evolución del error</i>	<i>158</i>
<i>Figura 6.10: Schewefel: Porcentaje de soluciones encontradas tras 100 generaciones</i>	<i>159</i>
<i>Figura 6.11: Espectros de zumos diluidos</i>	<i>162</i>
<i>Figura 6.12: Espectro característico de una muestra</i>	<i>166</i>
<i>Figura 6.13: Evolución de los errores de clasificación en entrenamiento y validación</i>	<i>171</i>
<i>Figura 6.14: Clasificación muestras: función fitness</i>	<i>175</i>

Índice de ecuaciones

Eq. 1: Función de Ackley	21
Eq. 2: Función de Rastrigin	23
Eq. 3: Función de Schwefel	23
Eq. 4: Fitness puro: $r(i,t)$	45
Eq. 5: Fitness estandarizado: $s(i,t)$	45
Eq. 6: Fitness ajustado: $a(i,t)$	46
Eq. 7: Fitness normalizado: $n(i,t)$	46
Eq. 8: Red de Neuronas Artificiales: valor NET_i	52
Eq. 9: Red de Neuronas Artificiales: activación	53
Eq. 10: PCA: Puntuaciones factoriales	70
Eq. 11: Distancia euclídea	74
Eq. 12: Distancia Manhattan	74
Eq. 13: UMDA: estimación distribución probabilidad	81
Eq. 14: MIMIC: factorización probabilidad	82
Eq. 15: Fitness shared: f_i	87
Eq. 16: Fitness shared: m_i	88
Eq. 17: Fitness scaling: función de compartición $sh(d_{ij})$	88
Eq. 18: Fitness scaling: f_i'	90
Eq. 19: Absorbancia	161

Índice de tablas

<i>Tabla I: Configuración Algoritmo Genético clásico</i>	<i>149</i>
<i>Tabla II: Configuración Algoritmo Genético de dos poblaciones (regiones variables)</i>	<i>149</i>
<i>Tabla III: Configuración Algoritmo Genético con agrupación en especies</i>	<i>149</i>
<i>Tabla IV: Muestras concentraciones bajas (2%-20%).....</i>	<i>164</i>
<i>Tabla V: Muestras concentraciones altas (20%-100%).....</i>	<i>165</i>
<i>Tabla VI: Concentraciones Bajas: Errores de clasificación empleando 176 variables.....</i>	<i>169</i>
<i>Tabla VII: Concentraciones Altas: Errores de clasificación empleando 176 variables</i>	<i>170</i>
<i>Tabla VIII: Concentraciones Bajas: Errores de clasificación empleando las dos variables seleccionadas por Procrustes Rotation</i>	<i>172</i>
<i>Tabla IX: Concentraciones Altas: Errores de clasificación empleando las dos variables seleccionadas por Procrustes Rotation</i>	<i>173</i>
<i>Tabla X: AG 2 Poblaciones: Errores Clasificación en Concentraciones Bajas</i>	<i>178</i>
<i>Tabla XI: AG 2 Poblaciones: Errores Clasificación en Concentraciones Altas.....</i>	<i>179</i>
<i>Tabla XII: Agrupación en Especies: Errores Clasificación en Concentraciones Bajas.....</i>	<i>180</i>
<i>Tabla XIII: Agrupación en Especies: Errores Clasificación en Concentraciones Altas.....</i>	<i>180</i>

Capítulo 1

Introducción

Cada vez es mayor el número de problemas que requiere un proceso previo de selección de variables para poder ser abordado de manera eficiente. En este tipo de problemas interesa conocer, de entre un conjunto de variables, cuáles son las que aportan mayor información a la hora de resolver el citado problema. Cuando las soluciones pueden ser posibles múltiples combinaciones válidas de variables, interesa conocer qué combinaciones aportan una información similar.

Los paradigmas imperativo y declarativo son ejemplos de tipos de programación convencionales que no suelen ser cómodos para trabajar con esta clase de problemas. El análisis exhaustivo de una casuística muy extensa y/o compleja es una característica común de tales problemas, característica que no es abordada de una forma sencilla con los paradigmas citados. En este tipo de situaciones pueden emplearse, entre otras, aproximaciones basadas en Inteligencia Artificial como los Sistemas Expertos, las Redes de Neuronas Artificiales o los Sistemas Adaptativos.

La resolución de un problema empleando técnicas de programación convencional suele requerir un elevado tiempo de programación. Como contraprestación, una vez implementado el método, las soluciones se obtienen de manera prácticamente inmediata. Al contrario de lo que ocurre con la programación tradicional, con el uso de técnicas no convencionales se requiere escaso conocimiento del problema, lo que da lugar a que se precise poca configuración. Sin embargo se requiere más tiempo para la obtención de soluciones mediante procesos de simulación o entrenamiento. Dentro de este último tipo de técnicas no tradicionales son especialmente interesantes los Algoritmos Genéticos, principalmente en problemas que requieren la optimización de un conjunto de parámetros.

Ante esta situación compensa el uso de técnicas de Computación Evolutivas (D. B. Fogel, 2000) puesto que el tiempo requerido para la simulación se amortiza con creces gracias a una fase de implementación mucho más reducida. Entre este tipo de técnicas, que evolucionan soluciones a partir de una población de individuos, destacan principalmente los Algoritmos Genéticos (Goldberg, 1989b).

No obstante, en el tipo de problemas anteriormente mencionado que requiere un proceso selección de variables, este tipo de técnicas evolutivas presentan una limitación. Tradicionalmente, las técnicas evolutivas en general, y los Algoritmos Genéticos en particular, han mostrado una especial eficiencia en la resolución de los más diversos problemas. Sin embargo, puede achacarse su tendencia a proporcionar una única solución al problema sobre el que se aplican. Durante la búsqueda de la mejor solución pueden llegar a descartarse otras que, aún no siendo soluciones globales al problema, sí lo serían locales. Así, si un problema presenta múltiples combinaciones de variables que pueden ser empleadas como solución, este tipo de técnicas no suele ser eficaz a la hora de proporcionarlas. En cualquier campo, pero particularmente en el de la selección de variables, existen dos buenas – y prácticas – razones por las que pueda ser deseable localizar los múltiples óptimos de un problema. Por un lado, forzando al Algoritmo Genético, o a cualquier otro método de búsqueda, a localizar múltiples óptimos las posibilidades de localizar el óptimo global del problema se ven consecuentemente aumentadas. Por otro lado, proporcionar múltiples soluciones válidas permite, como es obvio, escoger entre diferentes opciones, lo que ayuda a incrementar el conocimiento acerca del problema.

Sin embargo, el paralelismo intrínseco de las técnicas de computación evolutiva y, por lo tanto, de los Algoritmos Genéticos, indica que debería ser capaz de localizar, no sólo la solución global a un problema sino también las diferentes soluciones parciales.

1.1 Objetivos

Hasta la actualidad se han venido empleando diferentes técnicas de selección de variables. Algunas basadas en métodos matemáticos, otras basadas en métodos evolutivos con el fin de explotar su funcionamiento intrínsecamente paralelo.

Sin embargo, ambos grupos de técnicas presentan una limitación a la hora de trabajar con problemas que presentan un espacio de búsqueda multimodal, es decir, que presentan más de una solución global, o bien una solución global y múltiples soluciones parciales. En estos casos, tanto los métodos matemáticos, como los métodos evolutivos clásicos tienden a proporcionar una única solución válida. Si se desea obtener varias soluciones, una opción es la ejecución del proceso de selección tantas veces (variando en aquellos casos en los que sea necesario los parámetros iniciales) como soluciones distintas se deseen, con el consiguiente incremento de costes, tanto en el plano de los recursos como del tiempo.

En cambio, son prácticamente nulos los métodos que traten de manera simultánea el proceso de selección de variables y la multimodalidad del espacio de búsqueda. Ésta es la principal causa por la que se ha planteado la realización de esta tesis doctoral: buscar un mecanismo de selección que permita determinar, no sólo aquellas que aportan una mayor cantidad de información de la muestra que describen, sino el máximo número posible de combinaciones diferentes de variables que permitan obtener, sino el mismo nivel de bondad que la solución global, sí uno lo más cercano posible. De esta manera la búsqueda no se centra únicamente en el entorno de la solución global, sino que abarca también a las soluciones parciales.

Como resultado de esta búsqueda multimodal de variables, pueden extraerse de manera directa e inmediata las siguientes ventajas:

- Reducción del coste de adquisición de datos: como resultado directo del proceso de selección de variables, se necesita una menor cantidad de datos para caracterizar una muestra, con lo cual el tiempo de adquisición se ve significativamente reducido. Además, al realizarse una búsqueda multimodal, puede escogerse de entre las soluciones aquel conjunto de variables cuya adquisición lleve asociada una menor complejidad o coste.
- Incremento en la eficiencia del sistema clasificador: por lo general los procesos de selección de variables son el paso previo a la clasificación de una muestra. En estos casos, a menor cantidad de información de entrada del sistema clasificador, menor tiempo requerido para su procesado.
- Mejora en la comprensión del modelo clasificatorio: aquellos modelos que empleen menor cantidad de información para realizar una misma tarea serán más fácilmente comprendidos. Adicionalmente, si el sistema de selección de variables ofrece diferentes soluciones, será posible la construcción de diferentes modelos, es decir, podrá explicarse lo mismo de diferentes maneras, con lo cual el conocimiento acerca del dominio del problema se ve incrementado.
- Mejora en la eficacia: en ocasiones, dotar de demasiada información a los sistemas clasificadores, en vez de mejorar su rendimiento, produce una pérdida en su capacidad de generalización. Nuevamente, al

disponer de un conjunto de soluciones, en vez de una única solución, será posible emplearlas todas ellas con el objetivo de determinar cuál es la más eficiente bajo cada circunstancia concreta.

Por lo tanto, se plantea la obtención de un sistema que aborde el proceso de selección teniendo en cuenta la posible multimodalidad presente en el espacio de búsqueda. Se buscará, además, aportar una metodología clara y concisa sobre las diferentes tareas que deben llevarse a cabo en el citado proceso.

Se comprobará el correcto funcionamiento del sistema propuesto, así como su metodología, con casos experimentales basados en problemas genéricos, representativos dentro de los problemas multimodales. Una vez verificado y contrastado su correcto funcionamiento en dichos problemas, se probarán casos reales basados en tareas de clasificación, contrastándose los resultados obtenidos con otras técnicas existentes.

Concretamente, se mostrará el funcionamiento del sistema propuesto en un problema, con una alta tendencia a la multimodal, basado en la clasificación de muestras en atención a su concentración real en zumo.

1.2 Estructura de la tesis

La presente tesis está estructurada en varios capítulos de forma que, partiendo de una serie de datos sobre los métodos y técnicas existentes actualmente para la realización del proceso de selección de variables, sea posible analizar las diversas formas de conseguir los diferentes objetivos que se acaban de presentar.

El presente capítulo sirve para introducir el ámbito del problema sobre el que se desarrollará el trabajo, así como para formular los objetivos que se pretenden alcanzar.

En el capítulo 2, comenzará por explicarse en qué consiste el proceso de selección de variables y las ventajas que este aporta a la hora de resolver un problema. Dicha explicación incluirá además una taxonomía en la que pueden clasificarse las diferentes técnicas o métodos de selección de variables, así como una descripción de las estrategias de búsqueda válidas para la realización del proceso de selección de variables. Posteriormente, se focalizará la atención en el aspecto multidimensional de la selección de variables, mostrando las peculiaridades de este tipo de problemas, así como una serie de problemas clásicos en este ámbito. Finalmente dentro de este capítulo, se mostrarán los conceptos básicos de los Algoritmos Genéticos, base del sistema de selección de variables propuesto en la presente tesis.

En el capítulo 3, se mostrarán los métodos de selección de variables más comúnmente empleados hasta el momento. Se dará una clasificación de los mismos donde se indicarán las características más relevantes de cada uno y los principales problemas que puedan plantear. De esta forma, se puede evitar cometer los mismos errores que muchos de ellos tienen y así disponer de las ventajas que puedan tener y la forma de poder incorporarlas al sistema que se propondrá posteriormente.

Partiendo del análisis y estudio de los métodos anteriormente expuestos, en el capítulo 4 se presenta la hipótesis de trabajo en la que se sustenta el trabajo desarrollado en la presente tesis.

En el capítulo 5 se sientan las bases de desarrollo del sistema de extracción de reglas mediante técnicas de computación evolutiva. Se presenta

en primer lugar la metodología propuesta, junto con la descripción de las dos aproximaciones presentadas.

En el capítulo 6 se muestran los resultados de las pruebas que se han llevado a cabo una vez desarrollado el sistema. Se expondrán los diferentes resultados ordenados por nivel de complejidad. Así, se comenzará por casos de laboratorio y de reducidos espacios de búsqueda mediante los cuales se podrá comprobar el correcto funcionamiento de las propuestas realizadas en comparación con otros métodos existentes. Posteriormente, se expondrán los resultados de casos reales comparados con los resultados obtenidos por otros métodos.

El capítulo 7 recoge las conclusiones que se han obtenido del desarrollo de la presente tesis, tanto de la metodología expuesta como de la creación del sistema de selección de variables y de los resultados obtenidos.

En el capítulo 8 se mostrarán las diferentes líneas de investigación futura que se abren a partir del trabajo realizado a raíz del desarrollo de la presente tesis y del sistema expuesto.

Finalmente se encuentra la sección de Bibliografía, que tal y como su nombre indica, recoge las referencias bibliográficas empleadas en las diferentes etapas del desarrollo de la presente Tesis.

Capítulo 2

Fundamentos

2.1 Selección de Variables

En esta sección se va a presentar la problemática relacionada con la selección de variables, proceso que, por lo general, actúa de paso previo a la elaboración de modelos clasificatorios.

Primeramente se realizará una introducción al problema, seguida de una serie de ventajas que pueden obtenerse tras la aplicación de dicho proceso de selección de variables. Una vez enunciados los beneficios de

realizar la tarea de seleccionar el subconjunto de variables adecuado, se detallarán las diferentes filosofías desde las que es posible abordar dicho problema.

2.1.1 Definición

El problema de la selección del subconjunto de variables para la inducción de un modelo clasificador, suele denominarse en la bibliografía FSS (Feature Subset Selection) (Yang & Honavar, 1998), aunque este concepto es más genérico que la selección de variables propiamente dicha. La necesidad de métodos de selección de variables surge motivada por la no monotonicidad de los modelos clasificatorios en relación con el número de variables predictivas, así como por la existencia de ciertas variables predictivas que pueden llegar a ser irrelevantes o incluso redundantes. A continuación se comentan más en detalle estos conceptos.

En primer lugar, la no monotonicidad de la probabilidad de éxito de un sistema clasificador se debe al hecho –constatado empíricamente, y demostrado matemáticamente para algunos paradigmas– de que no por construir un modelo clasificador con una variable añadida a las ya existentes, la probabilidad de éxito que se va a obtener con este nuevo modelo clasificador deba superar a la del modelo actual. Es más, la probabilidad de éxito del nuevo modelo puede llegar a ser incluso inferior.

Por lo que respecta a la irrelevancia de las variables, se considera que una variable predictiva es irrelevante cuando el conocimiento del valor de la misma no aporta nada al conjunto original de variables. Un caso concreto de variables predictivas son las variables redundantes. Concretamente, una variable predictiva se dice redundante cuando su valor puede ser determinado partir de otras variables predictivas.

Por lo tanto, el objetivo que debe plantearse todo método de selección de variables debe ser el de detectar todas aquellas variables que son irrelevantes y/o redundantes para un problema clasificatorio dado. Con esto lo que se pretende es crear modelos parsimoniosos, guiados por el principio que en Estadística se denomina principio Occam Razor (Ariew, 1976) y que en Aprendizaje Automático se conoce como KISS (Keep It as Simple as possible, Stupid). Es decir, la idea subyacente a la modelización parsimoniosa es que si existen dos modelos que explican suficientemente bien un conjunto específico de datos, se debe escoger el modelo más simple de los dos.

Desde un punto de vista computacional la búsqueda del subconjunto de variables es un problema NP-Hard (Cotta, Sloper, & Moscato, 2004; Kohavi, 1995), es decir, no existe ningún método que garantice la solución óptima en un tiempo polinomial en el tamaño de problema. Esto significa que, en la práctica, encontrar la solución óptima cuando el tamaño del problema es grande es inviable.

Un proceso general de selección de variables puede ser formalizado en base a los siguientes conceptos básicos:

- Sea A un conjunto de N objetos, $A = \{X_i, i=1..N\}$. Cada uno de estos objetos X_i puede ser descrito por un conjunto V formado por M variables, bien cuantitativas o cualitativas, de tal forma que V_i será el conjunto de variables que describa al objeto i , $V_i = \{Q_j, j=1..M\}$.
- Si C es el conjunto formado por todas las clases o categorías que pueden establecerse y C_i es la clase a la que la muestra X_i pertenece, entonces el objeto X_i puede ser completamente descrito mediante el par $\{V_i, C_i\}$.

- El objetivo principal de todo procedimiento de selección de variables será determinar el subconjunto mínimo de variables S ($S \subset V$) que permita describir de manera satisfactoria todas las muestras pertenecientes al conjunto A .

2.1.2 Ventajas

A continuación se indican alguno de los beneficios directos derivados de la aplicación de métodos de selección de variables (Liu & Motoda, 2007).

En primer lugar, se produce una reducción en el coste necesario – tanto a nivel económico, temporal o de complejidad – para realizar la adquisición de los datos. Esto es debido a que el volumen de información a manejar para inducir el modelo es menor.

Relacionado con lo anterior, se produce una reducción en el número de muestras necesarias para aplicar posteriormente cualquier técnica de clasificación o agrupamiento, puesto que a menor número de características descriptivas de cada muestra, menor número de ejemplos requeridos por los algoritmos de clasificación para alcanzar una solución óptima.

Por otra parte mejora la comprensión del modelo clasificatorio, ya que el modelo de clasificación no se va a inducir a partir de un número de variables demasiado elevado. Además esta inducción del modelo clasificatorio será más rápida, debido precisamente a que el algoritmo de inducción va a trabajar con un número más reducido de variables.

Adicionalmente, una vez realizada la selección de variables es previsible una mejora en la bondad del modelo clasificatorio, derivada del principio de no monotonicidad explicado anteriormente.

Por último, y no por ello menos importante, se produce un aumento en el conocimiento del campo de aplicación, puesto que es posible extraer reglas acerca de la cantidad de información que cada variable aporta al conjunto de manera individual.

2.1.3 Clasificación de Técnicas de Selección de Variables

Las técnicas de selección de variables pueden ser esencialmente clasificadas en función de si el enfoque empleado es directo o indirecto. Se estaría hablando respectivamente de métodos *filter* o métodos *wrapper* (Guyon, Elisseeffi, & Kaelbling, 2003; Kohavi, 1995).

A grandes rasgos, las técnicas *filter* seleccionan subconjuntos de variables en un paso de pre-procesamiento de manera independiente del proceso de clasificación. Por su parte, los métodos *wrapper* utilizan aprendizaje automático para asignar una puntuación a los subconjuntos de muestras en función de su capacidad predictiva.

Por último, existe otro grupo de técnicas, denominadas *embedded* que realizan la selección de variables durante el propio proceso de aprendizaje o clasificación de las muestras. Los árboles de decisión CART, por ejemplo, realizan un proceso de selección de variables interno (Brieman, Friedman, Olshen, & Stone, 1984).

2.1.3.1 Filter

Esta aproximación establece una medida indirecta de bondad de la selección de las variables, habitualmente un ranking entre las variables predictoras en base a un criterio previamente fijado de relevancia entre una variable predictora y la variable clase a la que pertenece una muestra (ver Figura 2.1).

El criterio de relevancia puede estar construido en base a la cantidad de información mutua, el incremento en la verosimilitud del modelo teniendo o no en cuenta cada una de las variables individualmente, factores de correlación, test Chi-cuadrado, etc. Por lo tanto, dicho criterio determina la aportación de cada variable a la clasificación final.

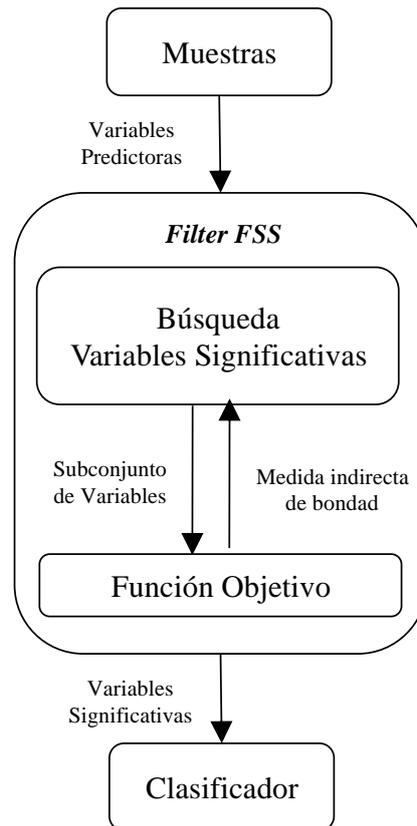


Figura 2.1: Selección de Variables: Técnica Filter

A partir de dicha medida de relevancia es posible asignar un orden a las variables predictoras, por ejemplo de mayor a menor relevancia respecto de la variable clase. De esta manera es posible seleccionar las k ($k < n$) primeras para inducir con ellas el modelo clasificador. Nótese que la medida de relevancia a partir de la que se han ordenado las variables predictivas no tiene en cuenta el paradigma con el que se va a inducir el modelo clasificador (árboles de clasificación, Naive-Bayes, etc.). Es por ello por lo

que este tipo de aproximación al problema puede verse como una selección indirecta.

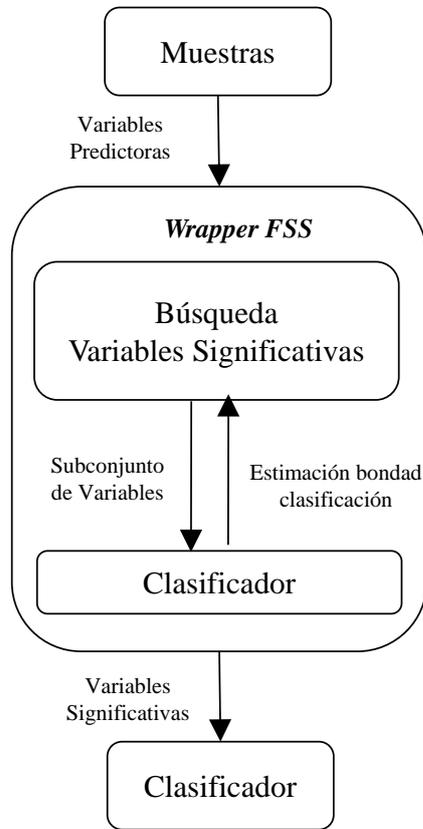


Figura 2.2: Selección de Variables: Técnica Wrapper

2.1.3.2 Wrapper

En la aproximación directa o *wrapper*, cada posible subconjunto de variables candidato es evaluado por medio del modelo clasificatorio inducido –con el paradigma utilizado– a partir del subconjunto que contiene exclusivamente las variables seleccionadas junto con la variable que determina la clase a la que pertenece dicha muestra. Esta evaluación puede realizarse en base al porcentaje de muestras correctamente clasificadas, en base al área bajo la curva ROC, etc.

Así, con esta aproximación, el problema de selección de variables puede ser visto como un problema de búsqueda en un espacio de cardinalidad 2^n , motivo por el cual los métodos heurísticos estocásticos de optimización se han venido aplicando con gran éxito en la presente tarea (Kohavi & John, 1997).

En la Figura 2.2 puede verse de forma gráfica el proceso previamente descrito.

En comparación con la aproximación *filter*, este tipo de aproximación es computacionalmente mucho más exigente lo que la convierte en difícilmente aplicable en muchas situaciones (Cantú-Paz, Newsam, & Kamath, 2004)

2.1.3.3 *Embedded*

La tercera técnica para la realización de selección de variables, denominada *embedded*, es sustancialmente diferente a las dos anteriormente comentadas. En este caso, la búsqueda del subconjunto óptimo de variables se realiza dentro del proceso de construcción del sistema clasificador.

Al igual que la aproximación basada en la técnica *wrapper*, esta aproximación está ligada a un algoritmo de aprendizaje específico. Sin embargo, tiene la ventaja de incluir la interacción con el modelo de clasificación a la vez que requiere un menor número de recursos computacionales que la anterior (Saeys, Inza, & Larranaga, 2007).

2.1.4 Estrategias de Búsqueda para Selección de Variables

Una vez definido el tipo de técnica de selección de variables a emplear, es necesario establecer un mecanismo que permita realizar la búsqueda de variables significativas.

Idealmente, la selección podría realizarse en base a la totalidad de los subconjuntos de variables que puedan formarse. Evidentemente se trata de una aproximación inviable puesto que implicaría analizar una cantidad ingente de combinaciones, lo que requeriría un tiempo de cálculo inasumible en la mayoría de aplicaciones. Por lo tanto, debe proporcionarse un criterio o estrategia de búsqueda simplificado y que proporcione resultados lo más próximos posibles al óptimo global.

Para evitar la explosión de combinaciones derivada de una búsqueda exhaustiva se han desarrollado diferentes métodos que exploran el conjunto de variables de una manera más racional. Estos métodos pueden agruparse en tres grandes grupos: exponenciales, secuenciales y aleatorios.

2.1.4.1 Exponenciales

Este tipo de técnicas realizan una búsqueda cuya complejidad crece exponencialmente a medida que aumenta la cardinalidad del conjunto de variables. Entre los métodos exponenciales el método de ramificación y acotación (*branch and bound*) (Land & Doig, 1960), que garantiza encontrar el subconjunto óptimo de un tamaño especificado, siempre y cuando la función de evaluación tenga un comportamiento monótonico. Es decir, si un sistema que emplee un subconjunto de variables de entrada realiza una clasificación peor que otro sistema que emplea un subconjunto de variables diferente, se

asume que ninguna combinación de las variables presentes en el primero de los subconjuntos conducirá a un mejor éxito en la clasificación y, por lo tanto, se abandonará la búsqueda entre esas variables.

2.1.4.2 Secuenciales

Los algoritmos de búsqueda secuenciales (o deterministas) siguen estrategias que reducen el número de estados a analizar mediante la aplicación de técnicas de búsqueda local. Los métodos más comunes son la búsqueda hacia adelante (*Forward Selection*), la eliminación hacia atrás (*Backward Elimination*) y un método que engloba a ambos, la selección paso a paso (*Stepwise Selection*). Todos estos métodos se detallan a continuación (Guyon et al., 2003; Mao, 2004):

- **Búsqueda hacia adelante (Forward Selection):** este método comienza con un conjunto vacío al que se van agregando variables hasta que el criterio de selección haya alcanzado un mínimo o bien se hayan añadido todas las variables.

El procedimiento comienza considerando cada una de las variables de manera individual y seleccionando la variable z que mejor comportamiento obtiene cuando se emplea individualmente a la entrada del clasificador. Una vez fijada la variable z , el proceso se repite considerando nuevamente de manera aislada el resto de variables. En cada paso se escogerá aquella cuya inclusión en el subconjunto de variables selecciones haga disminuir en mayor medida el error global del sistema. El proceso finalizará cuando la inclusión de nuevas variables no produzca una reducción en dicho error, o bien ya se hayan añadido todas las variables.

- **Eliminación hacia atrás (Backward Elimination):** este método funciona de manera inversa al anterior. Inicialmente todas las variables son consideradas como significativas y, paso a paso, se van eliminando aquellas cuya exclusión degrada en menor medida el sistema clasificador.
- **Selección paso a paso (Stepwise Selection):** Este método consiste en el encadenamiento de fases de los dos métodos anteriores. Se comienza considerando un conjunto vacío de variables, al igual que en el método de selección hacia adelante, añadiéndose en cada paso una nueva variable significativa. La diferencia estriba en que, tras la inclusión de una nueva variable se coteja si alguna de las variables que ya están presentes puede ser eliminada sin afectar al rendimiento del sistema, para lo que se emplea la eliminación hacia atrás.

El proceso termina cuando ninguna de las variables que todavía no ha sido seleccionada, tiene la importancia suficiente como para ser introducida en el conjunto de variables significativa.

2.1.4.3 Estocásticas

Los métodos no estocásticos comentados en los puntos anteriores permiten en muchas ocasiones obtener las variables más significativas de un conjunto de variables original. Sin embargo, cuando la cardinalidad de dicho conjunto inicial crece, tienden a quedarse atrapados en el entorno de soluciones subóptimas. En estos casos, suele ser interesante abordar el problema empleando métodos estocásticos, como pueden ser los Algoritmos Genéticos.

Éstos permiten realizar búsquedas locales alrededor de soluciones prometedoras pero adicionalmente poseen la componente de aleatoriedad necesaria para explorar otras soluciones en el espacio de búsqueda.

2.2 Selección Multimodal de Variables

Por norma general un problema de selección de variables puede ser resuelto de más de una manera. Son varios los conjuntos de variables que ofrecen una caracterización, sino idéntica, muy similar acerca de un conjunto de muestras. En este caso se habla de problemas multimodales.

Bajo estas circunstancias es deseable obtener el mayor número posible de soluciones válidas. Por una parte, esto ayuda a conocer mejor el propio problema, puesto que es posible extraer información de la interrelación existente entre las diversas soluciones. Por otra parte, facilita el paso último de resolución de un problema pues, al darse la posibilidad de escoger entre múltiples soluciones, inevitablemente alguna de ellas será más sencilla de comprender, menos compleja de implementar, etc.

A continuación se hace una breve introducción a este tipo de problemas, junto con una serie de ejemplos típicos.

2.2.1 Problemas Multimodales

2.2.1.1 Definición

Básicamente, un problema multimodal es aquel en cuyo espacio de búsqueda pueden encontrarse múltiples óptimos globales y/o múltiples óptimos locales, tal y como puede verse en la Figura 2.3. Por supuesto, en situaciones de este tipo interesa que los algoritmos de búsqueda encuentren el mayor número posible de dichos óptimos. Esto es especialmente importante cuando los óptimos locales presentan una mayor cuenca de atracción (son más amplios) que el óptimo global, con lo que se dificulta la localización de este último; cuando se necesita proporcionar más de una solución puesto que el espacio

de búsqueda contiene más de un óptimo global o bien porque se necesita un conjunto de soluciones dispersas.

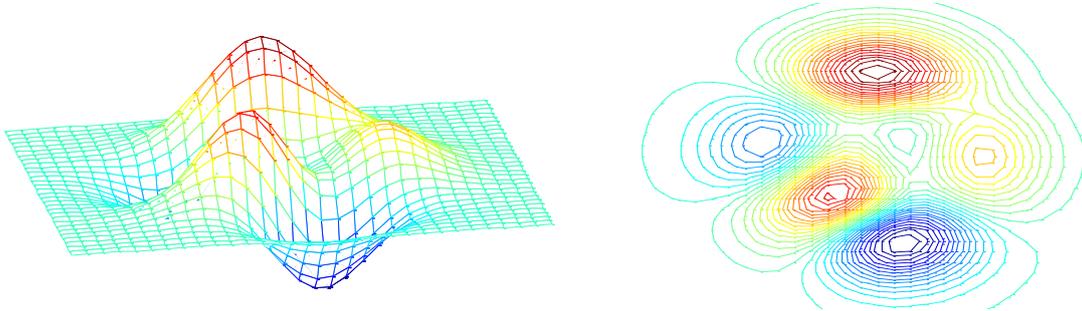


Figura 2.3: Representación 3D y 2D de una función multimodal tipo

2.2.1.2 Problemas Multimodales Clásicos

Son múltiples las funciones multimodales que pueden encontrarse (Hedar, 2009). A continuación se muestra una selección representativa de las funciones más comúnmente empleadas en el campo de los problemas multimodales.

2.2.1.2.1 Función de Ackley

Formalmente, este problema puede ser descrito como la búsqueda de un conjunto de variables $X = \{x_1, x_2, \dots, x_N\}$ de tal forma que, verificándose $x_i \in (-32.768, 32.768)$, se minimice la ecuación:

$$f(x) = 20 + e - 20 \cdot e^{-0.2 \cdot \sqrt{\frac{1}{n} \sum_i x_i^2}} - e^{\frac{1}{n} \sum_i \cos(2\pi x_i)} \quad (\text{Eq. 1})$$

Originalmente este problema fue definido para dos dimensiones ($n=2$) (Ackley, 1987), pero ha sido generalizado a N dimensiones (Bäck, 1996). Principalmente empleada en problemas de minimización, la función de Ackley se trata de una función multimodal con un único mínimo global y múltiples óptimos locales. Escogiendo $n=2$ para permitir su representación

gráfica en 3D, el mínimo local se encuentra en (0,0) y los valores máximos se sitúan en los límites exteriores (ver Figura 2.4).

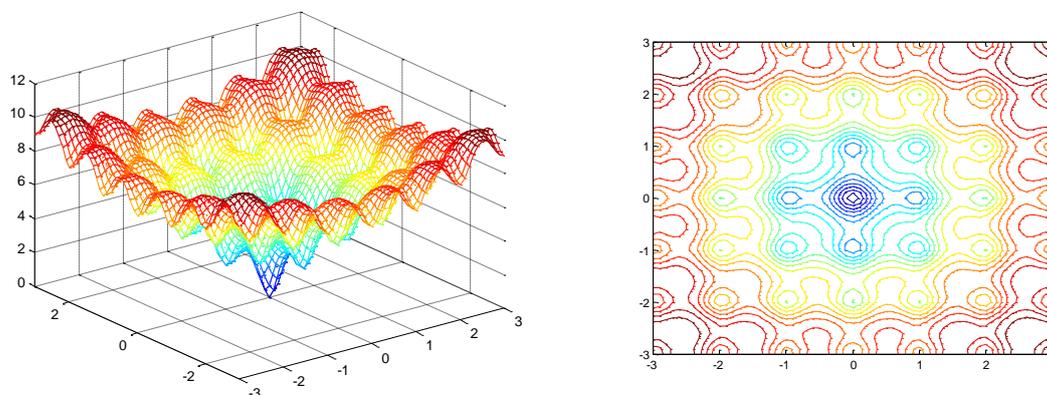


Figura 2.4: Funciones Multimodales: Función de Ackley

2.2.1.2.2 Función de Rastrigin

La función de Rastrigin (ver Figura 2.5) es un ejemplo típico de función multimodal no-lineal Propuesta por primera vez por Rastrigin (Törn & Zilinskas, 1989) para espacios de búsqueda bidimensionales, ha sido posteriormente generalizada para espacios n-dimensionales por Mühlenbein (Mühlenbein, Schomisch, & Born, 1991). La gran complejidad de esta función reside en la gran cantidad de mínimos y máximos globales que presenta y en su gran proximidad.

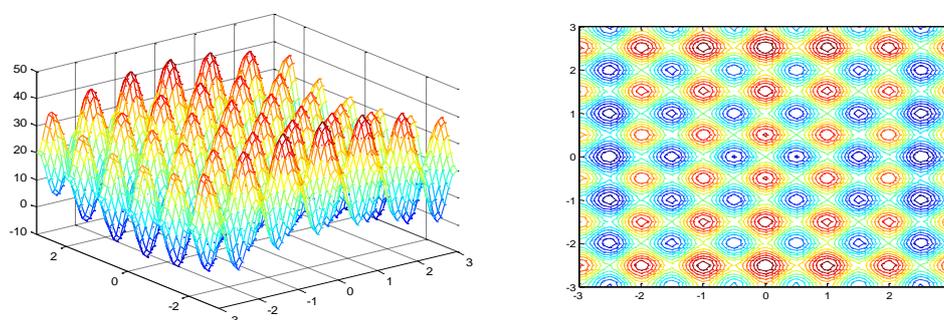


Figura 2.5: Funciones Multimodales: Función de Rastrigin

La formulación matemática de la función de Rastrigin es la mostrada en la ecuación 2, debiendo verificarse que $x_i \in (-5.12, 5.12)$:

$$f(x) = A \cdot n + \sum_{i=1}^n x_i^2 - 10 \cdot \cos(\omega x_i) \quad (\text{Eq. 2})$$

$$A = 10, \quad \omega = 2\pi$$

La complejidad computacional de la función de Rastrigin es de $O(n \cdot \log(n))$, siendo n la dimensión del problema. La superficie de la función está determinada por las variables externas A y ω , que controlan la amplitud y la frecuencia de modulación respectivamente.

2.2.1.2.3 Función de Schwefel

La función de Schwefel (Schwefel, 1981) cumple que el mínimo global está geoméricamente distante en el espacio de búsqueda de cualquiera de los siguientes mejores mejores mínimos globales (ver Figura 2.6). Por lo tanto, es fácil que los algoritmos de búsqueda converjan en la dirección equivocada.

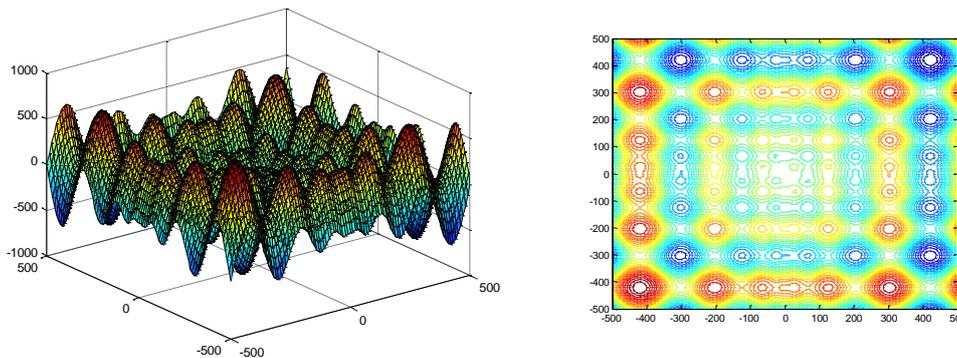


Figura 2.6: Funciones Multimodales: Función de Schwefel

La formulación matemática de la función de Schwefel (siendo $-500 \leq x_i \leq 500$) es la siguiente:

$$f(x) = \sum_{i=1}^n -x_i \cdot \sin(\sqrt{|x_i|}) \quad (\text{Eq. 3})$$

2.3 Algoritmos Genéticos

Los Algoritmos Genéticos son métodos adaptativos, generalmente usados en problemas de búsqueda y optimización de parámetros, basados en la reproducción sexual y en el principio supervivencia del más apto (Goldberg, 1989b; Grefenstette, 1992; Holland, 1975).

Más formalmente, y siguiendo la definición dada por Goldberg, “los Algoritmos Genéticos son algoritmos de búsqueda basados en la mecánica de selección natural y de la genética natural. Combinan la supervivencia del más apto entre estructuras de secuencias con un intercambio de información estructurado, aunque aleatorizado, para constituir así un algoritmo de búsqueda que tenga algo de las genialidades de las búsquedas humanas” (Goldberg, 1989b).

Para alcanzar la solución a un problema se parte de un conjunto inicial de individuos, llamado población, generado de manera aleatoria. Cada uno de estos individuos representa una posible solución al problema. Estos individuos evolucionarán tomando como base los esquemas propuestos por Darwin (Darwin, 1859, 2007) sobre la selección natural, y se adaptarán en mayor medida tras el paso de cada generación a la solución requerida.

2.3.1 Orígenes. Computación Evolutiva

Si algo funciona bien, ¿por qué no imitarlo? La respuesta a esta pregunta lleva directamente a los orígenes de la computación evolutiva (Michalewicz & Fogel, 2000). Durante millones de años las diferentes especies se han adaptado para poder sobrevivir en un medio cambiante. De la misma manera se podría tener una población de potenciales soluciones a un problema de las que se irían seleccionando las mejores hasta que se adaptasen perfectamente

al medio, en este caso el problema a resolver. En términos muy generales se podría definir la computación evolutiva como una familia de modelos computacionales inspirados en la evolución.

Más formalmente el término de computación evolutiva se refiere al estudio de los fundamentos y aplicaciones de ciertas técnicas heurísticas basadas en los principios de la evolución natural (Alba & Tomassini, 2002). Estas técnicas heurísticas podrían clasificarse en 3 categorías principales dando lugar a la ecuación evolutiva reflejada en la Figura 2.7.

$$\text{Computación Evolutiva} = \text{Algoritmos Genéticos} + \text{Estrategias de Evolución} + \text{Programación Evolutiva}$$

Figura 2.7: Ecuación Evolutiva

A continuación se detallarán un poco más los orígenes de cada una de las disciplinas participantes en la ecuación.

El desarrollo de los Algoritmos Genéticos se debe en gran medida a John Holland, investigador de la Universidad de Michigan. A finales de la década de los 60 desarrolló una técnica que imitaba en su funcionamiento a la selección natural. Aunque originalmente esta técnica recibió el nombre de “planes reproductivos”, a raíz de la publicación en 1975 de su libro “Adaptation in Natural and Artificial Systems” (Holland, 1975) se conoce principalmente con el nombre de Algoritmos Genéticos. A grandes rasgos un Algoritmo Genético consiste en una población de soluciones codificadas de forma similar a cromosomas. Cada uno de estos cromosomas tendrá asociado un ajuste, valor de bondad, ajuste o *fitness*, que cuantifica su validez como solución al problema. En función de este valor se le darán a cada individuo más o menos oportunidades de reproducción. Además, con cierta probabilidad se realizarán mutaciones de estos cromosomas.

Las bases de las Estrategias de Evolución fueron apuntadas en 1973 por Rechenberg en su obra “Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution” (Rechenberg, 1973). Las dos Estrategias de Evolución más empleadas son la $(\mu+\lambda)$ -ES y la (μ,λ) -ES. En la primera de ellas un total de μ padres producen λ descendientes reduciéndose nuevamente la población a μ individuos (los padres de la siguiente generación) por selección de los mejores individuos. De esta manera los padres sobreviven hasta que son reemplazados por hijos mejores que ellos. En la (μ,λ) -ES la descendencia reemplaza directamente a los padres, sin hacer ningún tipo de comprobación.

La Programación Evolutiva surge principalmente a raíz del trabajo titulado “Artificial Intelligence Through Simulated Evolution” de Fogel, Owens y Walsh, publicado en 1966 (L. J. Fogel, Owens, & Walsh, 1966). En este caso los individuos, conocidos aquí como organismos, son máquinas de estado finito. Los organismos que mejor resuelven alguna de las funciones objetivo obtienen la oportunidad de reproducirse. Antes de producirse los cruces para generar la descendencia se aplica el operador de mutación sobre los padres.

A su vez la computación evolutiva puede verse como uno de los campos de investigación de lo que se ha dado en llamar Soft Computing (Figura 2.8).

$$\textit{Soft Computing} = \text{Computación Evolutiva} + \text{Redes Neuronas Artificiales} + \text{Lógica Difusa}$$

Figura 2.8: Soft Computing

Como se ha comentado anteriormente la computación evolutiva tiene una fuerte base biológica. En sus orígenes los algoritmos evolutivos

consistieron en copiar procesos que tienen lugar en la selección natural. Este último concepto había sido introducido, rodeado de mucha polémica, por Charles Darwin (Darwin, 1859) y Alfred Wallace (Wallace, 1855). A pesar de que aún hoy en día no todos los detalles de la evolución biológica son completamente conocidos, existen algunos hechos apoyados sobre una fuerte evidencia experimental:

- La evolución es un proceso que opera, más que sobre los propios organismos, sobre los cromosomas. Estos cromosomas pueden ser considerados como herramientas orgánicas que codifican la vida, o visto al revés, una criatura e puede ser *creada* decodificando la información contenida en los cromosomas.
- La selección natural es el mecanismo que relaciona los cromosomas con la eficiencia respecto al medio de la entidad que representan. Otorga a los individuos más adaptados al medio un mayor número de oportunidades de reproducirse.
- Los procesos evolutivos tienen lugar durante la etapa de reproducción. Aunque existe una larga serie de mecanismos que afectan a la reproducción los más comunes son la mutación, causante de que los cromosomas de la descendencia sean diferentes a los de los padres, y el cruce o recombinación, que combina los cromosomas de los padres para producir la descendencia.

Sobre estos hechos se sustenta el funcionamiento de la Computación Evolutiva en general, y de los Algoritmos Genéticos en particular.

2.3.2 Bases Biológicas

En la Naturaleza, los individuos de una población compiten constantemente con otros por recursos tales como comida, agua y refugio. Los individuos que tienen más éxito en la lucha por los recursos tienen mayores probabilidades de sobrevivir y generalmente una descendencia mayor. Al contrario, los individuos peor adaptados tienen un menor número de descendientes, o incluso ninguno. Esto implica que los genes de los individuos mejor adaptados se propagarán a un número cada vez mayor de individuos de las sucesivas generaciones.

La combinación de características buenas de diferentes ancestros puede originar en ocasiones que la descendencia esté incluso mejor adaptada al medio que los padres. De esta manera, las especies evolucionan adaptándose más y más al medio a medida que transcurren las generaciones (Beasley, Bull, & Martin, 1993)

Pero la adaptación de un individuo al medio no sólo está determinada por su composición genética. Influyen otros factores como el aprendizaje, en ocasiones adquirido por el método de prueba y error, en ocasiones adquirido por imitación del comportamiento de los padres. Para imitar esta adquisición de conocimiento se han desarrollado variantes como, por ejemplo, el Ajuste Fino (Gestal, 2002) o los Algoritmos Genéticos culturales.

2.3.3 Codificación de Problemas

Cualquier solución potencial a un problema puede ser presentada dando valores a una serie de parámetros. El conjunto de todos los parámetros (*genes* en la terminología de Algoritmos Genéticos) se codifica en una cadena de valores denominada *cromosoma*.

El conjunto de los parámetros representado por un cromosoma particular recibe el nombre de *genotipo*. El genotipo contiene la información necesaria para la construcción del organismo, es decir, la solución real al problema, denominada *fenotipo*. Por ejemplo, en términos biológicos, la información genética contenida en el ADN de un individuo sería el genotipo, mientras que la expresión de ese ADN (el propio individuo) sería el fenotipo.

Desde los primeros trabajos de John Holland la codificación suele hacerse mediante valores binarios. Se asigna un determinado número de bits a cada parámetro y se realiza una discretización de la variable representada por cada gen. El número de bits asignados dependerá del grado de ajuste que se desee alcanzar. Evidentemente no todos los parámetros tienen porque estar codificados con el mismo número de bits. Cada uno de los bits pertenecientes a un gen suele recibir el nombre de *alelo*.

En la Figura 2.9 se muestra un ejemplo de un individuo binario que codifica 3 parámetros, cada uno de ellos con una diferente precisión en cuanto al número de *alelos* empleados.

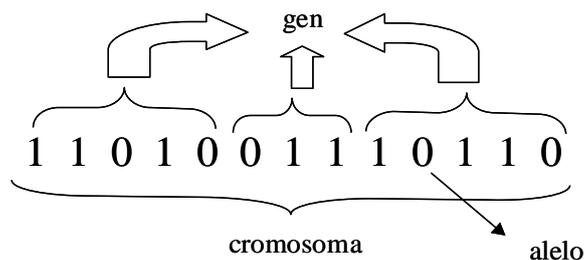


Figura 2.9: Individuo genético binario

Sin embargo, también existen representaciones que codifican directamente cada parámetro con un valor entero, real o en punto flotante (Wright, 1991). A pesar de que se acusa a estas representaciones de degradar el paralelismo implícito de las representaciones binarias, permiten el

desarrollo de operadores genéticos más específicos al campo de aplicación del Algoritmo Genético.

A continuación, la Figura 2.10 muestra un ejemplo típico de la aplicación de los Algoritmos Genéticos: la optimización de los pesos de una red neuronal.

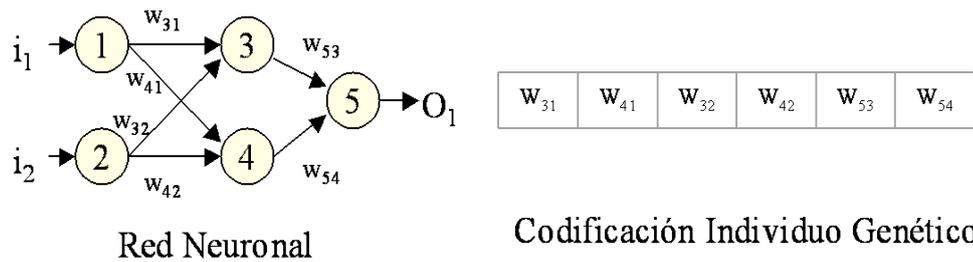


Figura 2.10: Codificación de una red neuronal

Codificar una red neuronal en forma de cromosoma es tan sencillo como asignar un gen del cromosoma a cada uno de los pesos de la red. También se podrían añadir genes que indicasen el número de capas de la red, y el número de elementos de procesamiento en cada capa.

2.3.4 Algoritmo Principal

Los Algoritmos Genéticos trabajan sobre una población de individuos. Cada uno de ellos representa una posible solución al problema que se desea resolver. Todo individuo tiene asociado un ajuste de acuerdo a la bondad con respecto al problema de la solución que representa (en la naturaleza el equivalente sería una medida de la eficiencia del individuo en la lucha por los recursos).

```
Inicializar población actual aleatoriamente  
MIENTRAS no se cumpla el criterio de terminación  
  crear población temporal vacía  
  MIENTRAS población temporal no llena  
    seleccionar padres  
    cruzar padres con probabilidad  $P_c$   
    SI se ha producido el cruce  
      mutar uno de los descendientes con probabilidad  $P_m$   
      evaluar descendientes  
      añadir descendientes a la población temporal  
    SINO  
      añadir padres a la población temporal  
    FIN SI  
  FIN MIENTRAS  
aumentar contador generaciones  
establecer como nueva población actual la población temporal  
FIN MIENTRAS
```

Figura 2.11: Pseudocódigo algoritmo genético

El funcionamiento genérico de un Algoritmo Genético puede apreciarse en el pseudocódigo de la Figura 2.11.

Una generación se obtendrá a partir de los individuos genéticos pertenecientes a la anterior generación por medio de la aplicación de diferentes operadores de reproducción. Dentro de estos operadores genéticos pueden diferenciarse 2 tipos:

- **Cruce.** Se trata de una reproducción de tipo sexual. Se genera una descendencia a partir del mismo número de individuos (generalmente 2) de la generación anterior. Existen varias maneras de aplicar el operador de cruce, que se comentarán con mayor detalle en un punto posterior.

- Copia. Se trata de una reproducción de tipo asexual. Un determinado número de individuos pasa sin sufrir ninguna variación directamente a la siguiente generación.

Una vez generados los nuevos individuos se realiza la mutación con una probabilidad P_m . La probabilidad de mutación suele ser muy baja, por lo general entre el 0.5% y el 2%.

Se sale de este proceso cuando se alcanza alguno de los criterios de parada fijados. Los más usuales suelen ser:

- Los mejores individuos de la población representan soluciones suficientemente buenas para el problema que se desea resolver.
- La población ha convergido. Un gen ha convergido cuando el 95% de la población tiene el mismo valor para él, en el caso de trabajar con codificaciones binarias, o valores dentro de un rango especificado, en el caso de trabajar con otro tipo de codificaciones. Una vez que todos los genes alcanzan la convergencia se dice que la población ha convergido. Cuando esto ocurre la media de bondad de la población se aproxima a la bondad del mejor individuo.
- Se ha alcanzado el número de generaciones máximo especificado.

Sobre este algoritmo inicialmente propuesto por Holland se han definido numerosas variantes.

Quizás una de las más extendidas consiste en prescindir de la población temporal de manera que los operadores genéticos de cruce y mutación se aplican directamente sobre la población genética. Con esta variante el proceso de cruces varía ligeramente. Ahora no basta, en el caso de

que el cruce se produzca, con insertar directamente la descendencia en la población.

Puesto que el número de individuos de la población se ha de mantener constante, antes de insertar la descendencia en la población se le ha de buscar una ubicación. Existen para ello diversas opciones, que se mostrarán en detalle en la sección 2.3.5.3.

Evidentemente cuando se trabaja con una única población no se puede decir que se pase a la siguiente generación cuando se llene la población, pues ésta siempre está llena. En este caso el paso a la siguiente generación se producirá una vez que se hayan alcanzado cierto número de cruces y mutaciones. Este número dependerá de la tasa de cruces y mutaciones especificadas por el usuario y del tamaño de la población. Así con una tasa de cruces del 90%, una tasa de mutaciones del 0.02% y trabajando con 100 individuos se pasará a la siguiente generación cuando se alcanzasen 45 cruces (cada cruce genera 2 individuos con lo que se habrían insertado en la población 90 individuos, esto es el 90%) o 2 mutaciones.

2.3.5 Operadores Genéticos

Para el paso de una generación a la siguiente se aplican una serie de operadores genéticos. Los más empleados son los operadores de selección, cruce, copia y mutación. En el caso de no trabajar con una población intermedia temporal también cobran relevancia los algoritmos de reemplazo. A continuación se verán en mayor detalle cada uno de estos procesos.

2.3.5.1 Selección

Los algoritmos de selección serán los encargados de escoger qué individuos van a disponer de oportunidades de reproducirse y cuáles no. Puesto que se trata de imitar lo que ocurre en la naturaleza, se ha de otorgar un mayor número de oportunidades de reproducción a los individuos más aptos. Por lo tanto la selección de un individuo estará relacionada con su valor de ajuste.

No se debe sin embargo eliminar por completo las opciones de reproducción de los individuos menos aptos, pues en pocas generaciones la población se volvería homogénea (Cantú-Paz, 2000; Goldberg & Deb, 1991).

Una opción bastante común consiste en seleccionar el primero de los individuos participantes en el cruce mediante alguno de los métodos expuestos a continuación y el segundo de manera aleatoria.

2.3.5.1.1 Selección por Ruleta

Propuesto por DeJong, es posiblemente el método más utilizado desde los orígenes de los Algoritmos Genéticos.

A cada uno de los individuos de la población se le asigna una parte de una ruleta proporcional a su ajuste, de tal forma que la suma de todos los porcentajes sea la unidad. Los mejores individuos recibirán una porción de la ruleta mayor que la recibida por los peores. Generalmente la población está ordenada en base al ajuste por lo que las porciones más grandes se encuentran al inicio de la ruleta. Para seleccionar un individuo basta con generar un número aleatorio del intervalo $[0..1]$ y devolver el individuo situado en esa posición de la ruleta. Esta posición se suele obtener recorriendo los individuos de la población y acumulando sus proporciones de ruleta hasta que la suma exceda el valor obtenido.

Es un método muy sencillo pero ineficiente a medida que aumenta el tamaño de la población (su complejidad es $O(n^2)$). Presenta además el inconveniente de que el peor individuo puede ser seleccionado más de una vez.

En mucha bibliografía se suele referenciar a este método con el nombre de Selección de Montecarlo.

2.3.5.1.2 Selección por Torneo

La idea principal de este método consiste en realizar la selección en base a comparaciones directas entre individuos (Blickle & Thiele, 1995). Existen dos versiones de selección mediante torneo:

- Determinística
- Probabilística

En la versión determinística se selecciona al azar un número p de individuos (generalmente se escoge $p=2$). De entre los individuos seleccionados se selecciona el más apto para pasarlo a la siguiente generación.

La versión probabilística únicamente se diferencia en la forma de seleccionar al individuo ganador del torneo. En vez de escoger siempre el mejor se genera un número aleatorio del intervalo $[0..1]$, si es mayor que un parámetro p (fijado para todo el proceso evolutivo) se escoge el individuo más alto y en caso contrario el menos apto. Generalmente el parámetro p toma valores en el rango $0.5 < p \leq 1$.

Variando el número de individuos que participan en cada torneo se puede modificar la presión de selección. Cuando participan muchos individuos en cada torneo, la presión de selección es elevada y los peores individuos apenas tienen oportunidades de reproducción. Un caso particular

es el *elitismo global*. Se trata de un torneo en el que participan todos los individuos de la población con lo cual la selección se vuelve totalmente determinista. Cuando el tamaño del torneo es reducido, la presión de selección disminuye y los peores individuos tienen más oportunidades de ser seleccionados.

Elegir uno u otro método de selección determinará la estrategia de búsqueda del Algoritmo Genético. Si se opta por un método con una alta presión de selección se centra la búsqueda de las soluciones en un entorno próximo a las mejores soluciones actuales. Por el contrario, optando por una presión de selección menor se deja el camino abierto para la exploración de nuevas regiones del espacio de búsqueda.

Existen muchos otros algoritmos de selección. Unos buscan mejorar la eficiencia computacional, otros el número de veces que los mejores o peores individuos pueden ser seleccionados. Algunos de estos algoritmos son muestreo determinístico, escalamiento sigma, selección por jerarquías, estado uniforme, sobrante estocástico, brecha generacional, etc.

2.3.5.2 Cruce

Una vez seleccionados los individuos, éstos son recombinados para producir la descendencia que se insertará en la siguiente generación (Booker, Fogel, Whitley, & Angeline, 1997). Tal y como se ha indicado anteriormente el cruce es una estrategia de reproducción sexual.

Su importancia para la transición entre generaciones es elevada puesto que las tasas de cruce con las que se suele trabajar rondan el 90%.

Los diferentes métodos de cruce podrán operar de dos formas diferentes. Si se opta por una estrategia destructiva los descendientes se

insertarán en la población temporal aunque sus padres tengan mejor ajuste (trabajando con una única población esta comparación se realizará con los individuos a reemplazar). Por el contrario utilizando una estrategia no destructiva la descendencia pasará a la siguiente generación únicamente si supera la bondad del ajuste de los padres (o de los individuos a reemplazar).

La idea principal del cruce se basa en que, si se toman dos individuos correctamente adaptados al medio y se obtiene una descendencia que comparta genes de ambos, existe la posibilidad de que los genes heredados sean precisamente los causantes de la bondad de los padres. Al compartir las características buenas de dos individuos, la descendencia, o al menos parte de ella, debería tener una bondad mayor que cada uno de los padres por separado. Si el cruce no agrupa las mejores características en uno de los hijos y la descendencia tiene un peor ajuste que los padres no significa que se esté dando un paso atrás. Optando por una estrategia de cruce no destructiva se garantiza que pasen a la siguiente generación los mejores individuos. Si, aún con un ajuste peor, se opta por insertar a la descendencia, y puesto que los genes de los padres continuarán en la población –aunque dispersos y quizás levemente modificados por la mutación–, en posteriores cruces se podrán volver a obtener estos padres, recuperando así la bondad previamente perdida.

Existen multitud de algoritmos de cruce. Sin embargo los más empleados son los que se detallarán a continuación:

- Cruce de 1 punto
- Cruce de 2 puntos
- Cruce uniforme

2.3.5.2.1 Cruce de 1 Punto

Es la más sencilla de las técnicas de cruce. Una vez seleccionados dos individuos se cortan sus cromosomas por un punto seleccionado aleatoriamente para generar dos segmentos diferenciados en cada uno de ellos: la cabeza y la cola. Se intercambian las colas entre los dos individuos para generar los nuevos descendientes. De esta manera ambos descendientes heredan información genética de los padres.

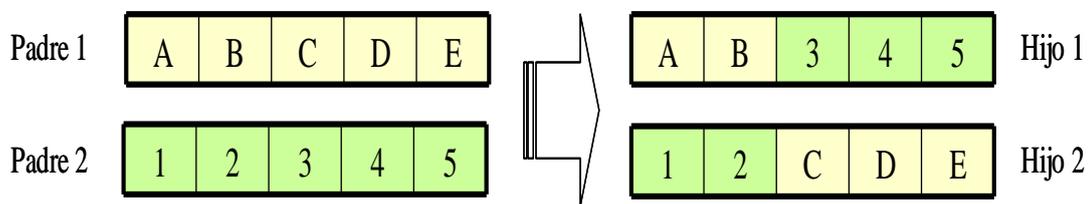


Figura 2.12: Cruce de 1 punto

En la Figura 2.12 se puede ver con claridad el proceso, al que en la bibliografía suele referirse con el nombre de SPX (Single Point Crossover).

2.3.5.2.2 Cruce de 2 Puntos

Se trata de una generalización del cruce de 1 punto. En vez de cortar por un único punto los cromosomas de los padres como en el caso anterior se realizan dos cortes. Deberá tenerse en cuenta que ninguno de estos puntos de corte coincida con el extremo de los cromosomas para garantizar que se originen tres segmentos. Para generar la descendencia se escoge el segmento central de uno de los padres y los segmentos laterales del otro padre.

Generalmente se suele referir a este tipo de cruce con las siglas DPX (Double Point Crossover). En la Figura 2.13 se muestra un ejemplo de cruce por dos puntos.

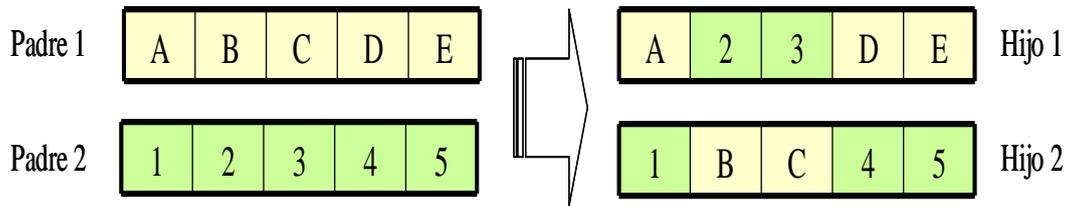


Figura 2.13: Cruce de 2 puntos

Generalizando se pueden añadir más puntos de cruce dando lugar a algoritmos de cruce multipunto. Sin embargo existen estudios que desaprueban esta técnica (De Jong, 1975; Spears, De Jong, & Rawlins, 1991). Aunque se admite que el cruce de 2 puntos aporta una sustancial mejora con respecto al cruce de un solo punto, el hecho de añadir un mayor número de puntos de cruce reduce el rendimiento del Algoritmo Genético. El problema principal de añadir nuevos puntos de cruce radica en que es más fácil que los segmentos originados sean corrompibles, es decir, que por separado quizás pierdan las características de bondad que poseían conjuntamente (De Jong & Spears, 1992). Sin embargo no todo son desventajas y añadiendo más puntos de cruce se consigue que el espacio de búsqueda del problema sea explorado más a fondo.

2.3.5.2.3 Cruce Uniforme

El cruce uniforme es una técnica completamente diferente de las vistas hasta el momento. Cada gen de la descendencia tiene las mismas probabilidades de pertenecer a uno u otro padre (Spears et al., 1991).

Aunque se puede implementar de muy diversas formas (Bate & Jones, 2008), la técnica implica la generación de una máscara de cruce con valores binarios. Si en una de las posiciones de la máscara hay un 1, el gen situado en esa posición en uno de los descendientes se copia del primer padre. Si por el

contrario hay un 0 el gen se copia del segundo padre. Para producir el segundo descendiente se intercambian los papeles de los padres, o bien se intercambia la interpretación de los unos y los ceros de la máscara de cruce.

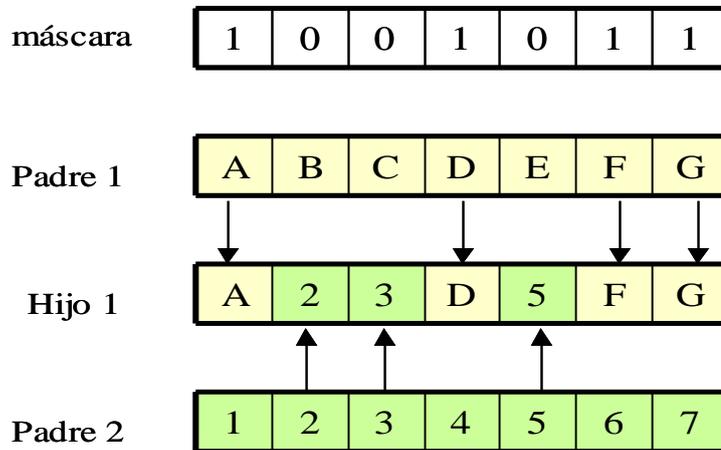


Figura 2.14: Cruce uniforme

Tal y como se puede apreciar en la Figura 2.14, la descendencia contiene una mezcla de genes de cada uno de los padres. El número efectivo de puntos de cruce no es fijo pero será por término medio $L/2$, siendo L la longitud del cromosoma (número de alelos en representaciones binarias o de genes en otro tipo de representaciones).

La máscara de cruce no permanece fija durante todo el proceso evolutivo, sino que se genera de manera aleatoria para cada cruce.

2.3.5.2.4 Cruces Específicos de Codificaciones no Binarias

Los tres tipos de cruce vistos hasta el momento son válidos para cualquier tipo de representación que se emplee en el genotipo de los individuos genéticos. Si se emplean genotipos formados por valores enteros o por valores reales pueden definirse otro tipo de operadores de cruce específicos para estas configuraciones (Deb, Sindhya, & Okabe, 2007; Herrera, Lozano, &

Sánchez, 2005). Entre estos tipos de cruce específicos, cabe destacar los siguientes:

- **Media:** el gen de la descendencia toma el valor medio de los genes de los padres. Tiene la desventaja de que únicamente se genera un descendiente en el cruce de dos padres.
- **Media geométrica:** cada gen de la descendencia toma como valor la raíz cuadrada del producto de los genes de los padres. Presenta el problema añadido de qué signo dar al resultado si los padres tienen signos diferentes.
- **Extensión:** se toma la diferencia existente entre los genes situados en las mismas posiciones de los padres y se suma al valor más alto o se resta del valor más bajo. Solventa el problema de generar un único descendiente.

2.3.5.3 Algoritmos de Reemplazo

Cuando en vez de trabajar con una población temporal se hace con una única población, sobre la que se realizan las selecciones e inserciones, deberá tenerse en cuenta que para insertar un nuevo individuo deberá de eliminarse previamente otro de la población. Existen diferentes métodos de reemplazo:

- **Aleatorio:** el nuevo individuo se inserta en un lugar cualquiera de la población.
- **Reemplazo de padres:** se obtiene espacio para la nueva descendencia liberando el espacio ocupado por los padres.
- **Reemplazo de similares:** una vez obtenido el ajuste de la descendencia se selecciona un grupo de individuos (entre seis y diez) de la población

con un ajuste similar. Se reemplazan aleatoriamente los que sean necesarios.

- **Reemplazo de los peores:** de entre un porcentaje de los peores individuos de la población se seleccionan aleatoriamente los necesarios para dejar sitio a la descendencia.

2.3.5.4 Copia

La copia es la otra estrategia reproductiva para la obtención de una nueva generación a partir de la anterior. A diferencia del cruce, se trata de una estrategia de reproducción asexual. Consiste simplemente en la copia de un individuo en la nueva generación.

El porcentaje de copias de una generación a la siguiente es relativamente reducido, pues en caso contrario se corre el riesgo de una convergencia prematura de la población hacia los individuos copiados. De esta manera el tamaño efectivo de la población se reduciría notablemente y la búsqueda en el espacio del problema se focalizaría en el entorno de los citados individuos.

Lo que generalmente se suele hacer es seleccionar dos individuos para el cruce, y si éste finalmente no tiene lugar, se insertan en la siguiente generación los individuos seleccionados.

2.3.5.5 Mutación

La mutación de un individuo provoca que alguno de sus genes, generalmente uno sólo, varíe su valor de forma aleatoria.

Aunque se pueden seleccionar los individuos directamente de la población actual y mutarlos antes de introducirlos en la nueva población, la

mutación se suele utilizar de manera conjunta con el operador de cruce. Así, en primer lugar se seleccionan los individuos de la población para realizar el cruce. Si el cruce tiene éxito entonces uno de los descendientes, o ambos, se muta con cierta probabilidad P_m . Se imita de esta manera el comportamiento que se da en la naturaleza, pues cuando se genera la descendencia siempre se produce algún tipo de error, por lo general sin mayor trascendencia, en el paso de la carga genética de padres a hijos.

La probabilidad de mutación es muy baja, generalmente menor al 1%. Esto se debe sobre todo a que los individuos suelen tener un ajuste menor después de mutados. Sin embargo se realizan mutaciones para garantizar que ningún punto del espacio de búsqueda tenga una probabilidad nula de ser examinado.

Tal y como se ha comentado, la mutación más usual es el reemplazo aleatorio. Éste consiste en variar aleatoriamente un gen dentro del un cromosoma. Si se trabaja con codificaciones binarias consistirá simplemente en negar un bit. También es posible realizar la mutación intercambiando los valores de dos alelos del cromosoma. Con otro tipo de codificaciones no binarias existen otras opciones:

- Incrementar o decrementar a un gen una pequeña cantidad generada aleatoriamente.
- Multiplicar un gen por un valor aleatorio próximo a 1.

Aunque no es lo más común, existen implementaciones de Algoritmos Genéticos en las que no todos los individuos tienen los cromosomas de la misma longitud. Esto implica que no todos ellos codifican el mismo conjunto de variables. En este caso existen mutaciones adicionales como puede ser la inclusión de un nuevo gen o la eliminación de uno ya existente.

2.3.6 Evaluación

Para el correcto funcionamiento de un Algoritmo Genético debe de proporcionarse un método que indique si los individuos de la población representan o no buenas soluciones al problema planteado. Por lo tanto para cada tipo de problema que se desee resolver deberá derivarse un nuevo método, al igual que ocurrirá con la propia codificación de los individuos.

De esto se encarga la función de evaluación, que establece una medida numérica de la bondad de una solución. Esta medida recibe el nombre de ajuste o fitness. En la naturaleza el ajuste (o adecuación) de un individuo puede considerarse como la probabilidad de que ese individuo sobreviva hasta la edad de reproducción y se reproduzca.

En el mundo de los Algoritmos Genéticos se empleará esta medición para controlar la aplicación de los operadores genéticos. Es decir, permitirá controlar el número de selecciones, cruces, copias y mutaciones llevadas a cabo.

La aproximación más común consiste en crear explícitamente una medida de ajuste para cada individuo de la población. A cada uno de los individuos se les asigna un valor de ajuste escalar por medio de un procedimiento de evaluación bien definido. Tal y como se ha comentado, este procedimiento de evaluación será específico del dominio del problema en el que se aplica el Algoritmo Genético. También puede calcularse el ajuste mediante una manera 'co-evolutiva'. Por ejemplo, el ajuste de una estrategia de juego puede ser determinado mediante la aplicación de la estrategia codificada por un individuo contra la población entera (o en su defecto una muestra) de estrategias de oposición.

Se pueden diferenciar cuatro tipos de ajuste o fitness (Koza, 1992, 1999), que a continuación pasan a explicarse:

- **Fitness Puro:** $r(i,t)$

Es la medida de ajuste establecida en la terminología natural del propio problema. Por ejemplo, supóngase una población de hormigas que deben llenar la despensa de cara al invierno. La bondad de cada hormiga será el número de piezas de comida llevadas por ella hasta el hormiguero.

$$r(i,t) = \sum_{j=1}^{N_c} |s(i,j) - c(i,j)| \quad (Eq. 4)$$

$S(i,j)$: valor deseado
 $C(i,j)$: valor obtenido por el individuo
 N_c : número de casos

En los problemas de maximización, como sería el de las hormigas mencionado anteriormente, los individuos con un fitness puro elevado serán los más interesantes. Al contrario, en los problemas de minimización interesarán los individuos que presenten un fitness puro más reducido.

- **Fitness Estandarizado:** $s(i,t)$

Para solucionar esta dualidad ante problemas de minimización o maximización se modifica el ajuste puro de acuerdo a la ecuación Eq. 5.

$$s(i,t) = \begin{cases} r(i,t) & \text{minimización} \\ r_{max} - r(i,t) & \text{maximización} \end{cases} \quad (Eq. 5)$$

En el caso de problemas de minimización se emplea directamente la medida de fitness puro. Por el contrario, si el problema

es de maximización se resta de una cota superior r_{\max} del error el fitness puro.

Empleando esta métrica la bondad de un individuo será mayor cuanto más cercano esté a cero el valor del ajuste. Por lo tanto, dentro de la generación t , un individuo i siempre será mejor que uno j si se verifica que $s(i,t) < s(j,t)$.

- **Fitness Ajustado:** $a(i,t)$

El fitness ajustado se obtiene aplicando la transformación reflejada en la ecuación (Eq. 6) al fitness estandarizado. De esta manera, el fitness ajustado tomará siempre valores del intervalo $[0,1]$. Cuando más se aproxime el fitness ajustado de un individuo a 1 mayor será su bondad.

$$a(i,t) = \frac{1}{1 + s(i,t)} \quad (\text{Eq. 6})$$

- **Fitness Normalizado:** $n(i,t)$

Los diferentes tipos de fitness vistos hasta ahora hacen referencia únicamente a la bondad del individuo en cuestión. El fitness normalizado introduce un nuevo aspecto: indica la bondad de una solución con respecto al resto de soluciones representadas en la población. Se obtiene de la siguiente forma (considerando una población de tamaño M):

$$n(i,t) = \frac{a(i,t)}{\sum_{k=1}^M a(k,t)} \quad (\text{Eq. 7})$$

Al igual que el fitness ajustado, siempre tomará valores del intervalo $[0,1]$, con mejores individuos cuanto más próximo esté a la unidad. Pero a diferencia de antes, un valor cercano a 1 no sólo indica que ese individuo represente una buena solución al problema, sino que además es una solución destacadamente mejor que las proporcionadas por el resto de la población.

Tal y como puede derivarse de su definición, la suma de los valores de fitness normalizado de una población será siempre 1.

Este tipo de ajuste es empleado en la mayoría de los métodos de selección proporcionales al fitness.

2.4 Redes de Neuronas Artificiales

Desde la primera mitad del siglo XX se han empezado a desarrollar modelos computacionales que han intentado emular el comportamiento del cerebro humano (McCulloch & Pitts, 1943). Aunque se han propuesto una gran cantidad de ellos, todos usan una estructura en red en la cual los nodos o neuronas son procesos numéricos que involucran estados de otros nodos según sus uniones. Una clase de estos modelos computacionales son las Redes de Neuronas Artificiales (Haykin, 1998).

Las Redes de Neuronas Artificiales (RNA) se han hecho muy populares debido a la facilidad en su uso (ver *Figura 2.15*) e implementación y la habilidad para aproximar cualquier función matemática. Las Redes de Neuronas Artificiales, con su marcada habilidad para obtener resultados de datos complicados e imprecisos, pueden utilizarse para extraer patrones y detectar tramas que son muy difíciles de apreciar por humanos u otras técnicas computacionales.

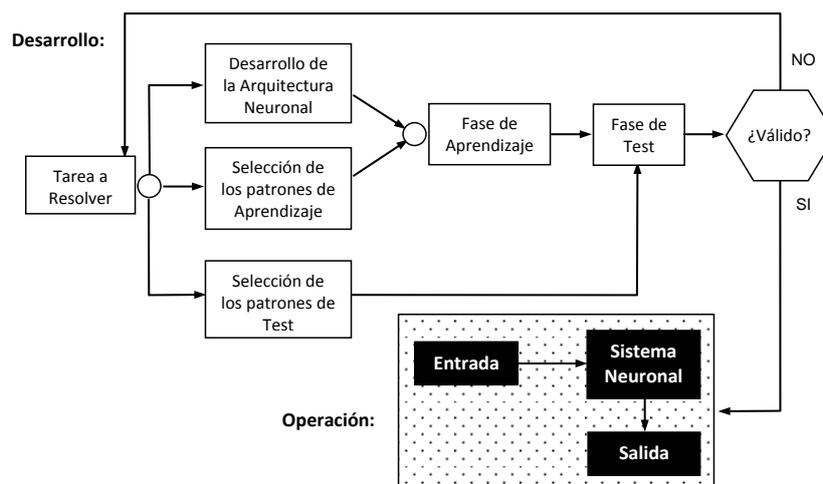


Figura 2.15: Esquema básico del trabajo con Redes de Neuronas Artificiales

El primer modelo de neurona artificial fue propuesto por McCulloch y Pitts (McCulloch & Pitts, 1943) donde se modelizaba una estructura y un funcionamiento simplificado de las neuronas del cerebro, considerándolas como dispositivos con n entradas, una única salida y sólo dos estados posibles: activa o inactiva.

Una red era, en ese planteamiento inicial, una colección de neuronas de McCulloch y Pitts, todas sincronizadas, donde las salidas de unas neuronas estaban conectadas a las entradas de otras. Algunos de los planteamientos de McCulloch y Pitts se han mantenido desde 1943 sin modificaciones, otros por el contrario han ido evolucionando, pero todas las formalizaciones matemáticas que se han realizado desde entonces, sobre las Redes de Neuronas Artificiales, aún sin pretender ser una modelización exacta de las redes de neuronas biológicas, sí han resultado un punto de partida útil para el estudio de las mismas.

Una de las definiciones que se estima más certera de Red de Neuronas Artificiales es la siguiente: “Las redes neuronales son conjuntos de elementos de cálculo simples, usualmente adaptativos, interconectados masivamente en

paralelo y con una organización jerárquica que le permite interactuar con algún sistema del mismo modo que lo hace el sistema nervioso biológico” (Kohonen, 1989). Su aprendizaje adaptativo, auto-organización, tolerancia a fallos, operación en tiempo real y fácil inserción dentro de la tecnología existente, han hecho que su utilización se haya extendido en áreas como la biológica, financiera, industrial, medio ambiental, militar, salud, etc. (Hilera & Martínez, 1995). Están funcionando en aplicaciones que incluyen identificación de procesos (González-García, Rico-Martínez, & Kevrekidis, 1998), detección de fallos en sistemas de control (Aldrich & van Deventer, 1995), modelación de dinámicas no lineales (Meert & Rijckaert, 1998; Wang, Oh, & Yoon, 1998), control de sistemas no lineales (Bloch, Sirou, Eustache, & Fatrez, 1997; Narendra, Feiler, & Tian, 2008; Rivals & Personnaz, 2000) y optimización de procesos (Kalogirou, 2004; Nascimento, Giudici, & Guardani, 2000).

En general, se puede encontrar que una Red de Neuronas Artificiales se suele caracterizar por tres partes fundamentales: la topología de la red, la regla de aprendizaje y el tipo de entrenamiento.

En este afán de emular el cerebro, esto es simular tanto su estructura como su funcionamiento, se han desarrollado numerosos modelos de Redes de Neuronas Artificiales (Freeman & Skapura, 1991), entre los que se pueden mencionar: Perceptron (1957), Adeline y Madeline (1960), Avalancha (1967), Retropropagación (1974), Hopfield y SOM (1980), ART (1986), etc. De los modelos anteriores se puede apreciar que esta idea tiene más de 40 años, sin embargo, sólo en las últimas décadas se ha desarrollado la tecnología que permita su aplicación de manera eficiente.

Cabe destacar, para concluir esta breve introducción, que las Redes de Neuronas Artificiales, gracias al masivo paralelismo de su estructura, gozan de una serie de ventajas:

1. **Aprendizaje adaptativo.** Capacidad de aprender a realizar tareas basadas en un entrenamiento o una experiencia inicial.
2. **Autoorganización.** Una red neuronal puede crear su propia organización o representación de la información que recibe durante la etapa de aprendizaje.
3. **Tolerancia a fallos gracias a poseer la información distribuída o vía información redundante.** La destrucción parcial de una red puede conducir a una degradación de su estructura; sin embargo, algunas capacidades de la red se pueden retener, incluso sufriendo daños considerables.
4. **Capacidad de generalización.** Ante la entrada de datos nuevos es capaz de producir resultados coherentes de acuerdo con la naturaleza del problema para el cual han sido entrenadas.
5. **Operación en tiempo real.** El cómputo neuronal puede realizarse en paralelo, bien vía software o mediante máquinas especiales para obtener esta ventaja (hardware conexionista o masivamente paralelo).

Pero los sistemas neuronales no están exentos de ciertos inconvenientes. Uno importante es que habitualmente realizan un complejo procesamiento que supone millones de operaciones, por lo que no es posible seguir paso a paso el razonamiento que les ha llevado a extraer sus

conclusiones. Sin embargo, en redes pequeñas, mediante simulación o por el estudio de los pesos sinápticos sí es posible saber, al menos, qué variables de las introducidas han sido relevantes para tomar la decisión.

2.4.1 Fundamentos y Conceptos Básicos

Warren McCulloch y Walter Pitts pueden ser considerados como los padres de las Redes de Neuronas Artificiales, ya que fueron los primeros en diseñar una neurona artificial. En los años 40 suponían que las neuronas biológicas eran de carácter binario, lo cual resulta ser bastante inexacto, pero sirvió de base para posteriores estudios sobre el sistema nervioso.

Así, McCulloch y Pitts proponen el modelo de neurona que lleva su nombre, el cual es un dispositivo binario con un umbral fijo que hay que superar para que cambie de estado. Recibe sinapsis excitadoras de otros elementos, los cuales tienen la característica de ser del mismo valor. Puede recibir sinapsis inhibitoras que son de acción total, lo cual supone que la recepción de una impide el cambio de estado del elemento, sin importar la cantidad de sinapsis excitadoras que hubiese.

La neurona artificial o elemento formal está conceptualmente inspirada en la neurona biológica. Esto es, los investigadores están en su inmensa mayoría pensando en la organización cerebral cuando consideran configuraciones y algoritmos de Redes de Neuronas.

Se va a considerar una neurona como un elemento formal o módulo o unidad básica de la red que recibe información de otros módulos o del entorno; la integra, la computa y emite una única salida que se va a transmitir idéntica a múltiples neuronas posteriores (Wasserman, 1989).

Considérese a los pesos sinápticos como referencia a la noción biológica de la fuerza de unión entre los elementos; es decir, a la fuerza de la sinapsis. Se considera que una sinapsis es fuerte; es decir, tiene un alto grado de conexión, cuando la información que transmite contribuye, en gran medida, a un nuevo estado o a la alteración que se produzca en la neurona receptora y, por tanto, en la respuesta que ésta elabora.

En una red de neuronas existe un peso o fuerza sináptica que va a ser un valor numérico que pondera las señales que se reciben por sus entradas. Este peso será un valor que determina la fuerza de conexión entre 2 neuronas. Cuando se evalúa una neurona se debe calcular el conjunto de todas las fuerzas o valores (denominado NET) que se reciben por sus entradas. Una vez calculado el valor conjunto de todas las entradas se aplica una función de activación (FA) que determinará el valor del estado interno de la neurona y que será lo que se transmita a su salida.

La combinación de las señales que recibe una neurona se puede calcular como sigue:

$$NET_i(t) = \sum_{j=0}^{N-1} [w_{ij} \cdot o_j \cdot (t - 1)] \quad (\text{Eq. 8})$$

w_{ij} = Peso de la conexión entre la neurona j (que emite) y la neurona i (que recibe).

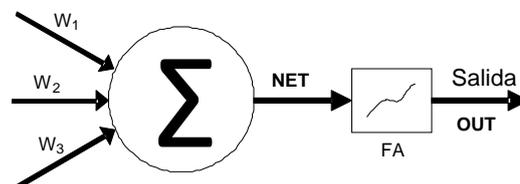


Figura 2.16: Neurona Artificial

La función de activación y la función de transferencia son las encargadas de definir el nuevo estado de activación A_i y la respuesta O_i de la neurona. Tanto el estado de activación como la salida de la neurona van a

variar en función de las entradas que recibe en un determinado momento y del estado de activación previo que tenga esa neurona (ver Figura 2.16).

2.4.2 Función de activación

La función de activación de una neurona es la encargada de relacionar la información de entrada de la neurona con el siguiente estado de activación que tenga esa neurona.

Existen dos modelos de función de activación:

- **Modelos acotados:** El valor de la activación de la neurona puede ser cualquiera dentro de un rango continuo de valores.
- **Modelos No acotados:** No existe ningún límite para los valores de activación.

Cuando se diseña una red debe establecerse cómo van a ser los valores de activación de cada neurona y se debe decidir la función de activación (FA) con la que cada neurona procesará las entradas (ver Eq. 9). Por lo tanto, la función de activación va a actuar sobre las señales de entrada, sobre los pesos sinápticos asociados con cada entrada y sobre el valor de activación que tenía la neurona en el momento de recibir las señales.

$$A_i(t) = FA(A_i(t-1), NET_i(t-1)) \quad (Eq. 9)$$

FA = Función de activación.

2.4.3 Tipos de neuronas

La linealidad de las funciones que definen a los elementos de la red es quizás lo que va a proporcionar la característica más definitoria. Así, se pueden clasificar las neuronas en lineales y no lineales.

Neuronas lineales

Una neurona es lineal cuando su salida es linealmente dependiente de sus entradas, es decir, proporcional a las funciones de transferencia y de activación.

Esto conlleva ciertos problemas como la falta de persistencia en las respuestas, de modo que cambios muy pequeños en las entradas pueden producir fluctuaciones bastante grandes en las respuestas, o la falta de adecuación simultánea, pues es imposible que con neuronas lineales la respuesta de una neurona se adapte tanto a señales grandes como a pequeñas.

Neuronas no lineales

En estas neuronas, o bien la función de activación, o bien la función de transferencia (o ambas) son funciones no lineales, dando lugar a que la respuesta de la neurona no sea función lineal de sus entradas.

Este tipo de neuronas va a producir respuestas acotadas, desapareciendo los problemas de fluctuación y la falta de adecuación a señales pequeñas y grandes.

Como ejemplo de funciones no lineales se pueden destacar la función umbral (Figura 2.17), la función sigmoide (Figura 2.18) o la función hiperbólica tangente (Figura 2.19).

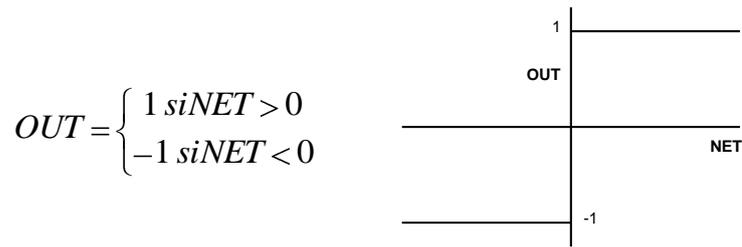


Figura 2.17: Función de transferencia umbral

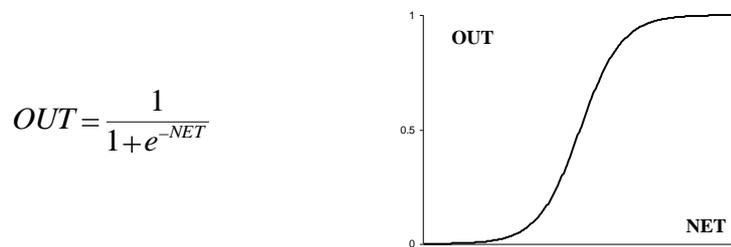


Figura 2.18: Función de transferencia sigmoide

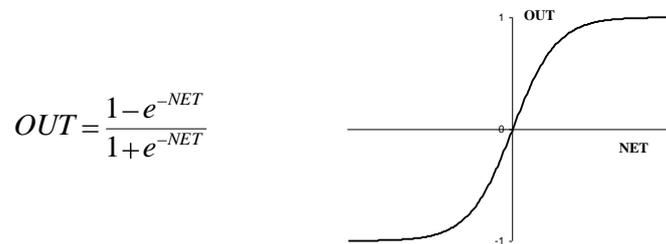


Figura 2.19: Función de transferencia hiperbólica-tangente

2.4.4 Flujo de datos en las Redes de Neuronas Artificiales

2.4.4.1 Redes alimentadas hacia adelante

Las redes alimentadas hacia adelante – generalmente conocidas como redes feedforward – son aquellas en las que, como su nombre indica, la información se mueve en un único sentido, desde la entrada hacia la salida (Figura 2.20).

Estas redes están clásicamente organizadas en “capas”. Cada capa agrupa a un conjunto de neuronas que reciben sinapsis de las neuronas de la capa anterior y emiten salidas hacia las neuronas de la capa siguiente. Entre las neuronas de una misma capa no hay sinapsis.

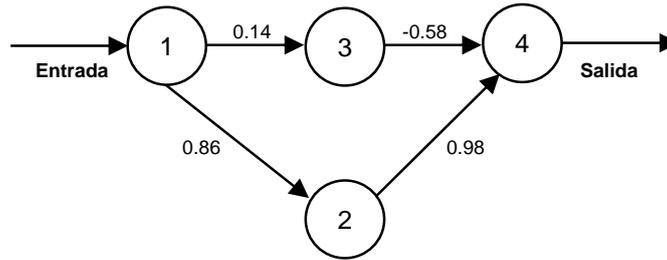


Figura 2.20: RNA alimentada hacia delante

En este tipo de redes existe al menos una capa de entrada, formada por las neuronas que reciben las señales de entrada a la red y una capa de salida, formada por una o más neuronas que emiten la respuesta de la red al exterior. Entre la capa de entrada y la de salida existen una o más capas intermedias.

En redes así construidas es evidente que la información sólo puede moverse en un sentido: desde la capa de entrada hasta la capa de salida, atravesando todas y cada una de las capas intermedias una sólo vez.

El hecho de que no haya conexión entre las neuronas de una misma capa hace que no haya tiempos de espera en los que las neuronas estén interactuando unas sobre otras hasta que toda la capa adquiera un estado estable. Se trata por tanto de redes rápidas en sus cálculos.

2.4.4.2 Redes con Retroalimentación total o parcial

En este tipo de redes los elementos pueden enviar estímulos a neuronas de capas anteriores, de su propia capa o a ellos mismos, por lo que desaparece el concepto de agrupamiento de las neuronas en capas (Figura 2.21). Cada neurona puede estar conectada a todas las demás; de este modo, cuando se recibe información de entrada a la red, cada neurona tendrá que calcular y recalcular su estado varias veces, hasta que todas las neuronas de la red alcancen un estado estable. Un estado estable es aquel en el que no ocurren cambios en la salida de ninguna neurona. No habiendo cambios en las salidas, las entradas de todas las neuronas serán también constantes, por lo que no tendrán que modificar su estado de activación ni su respuesta, manteniéndose así un estado global estable (Pearlmutter, 1990).

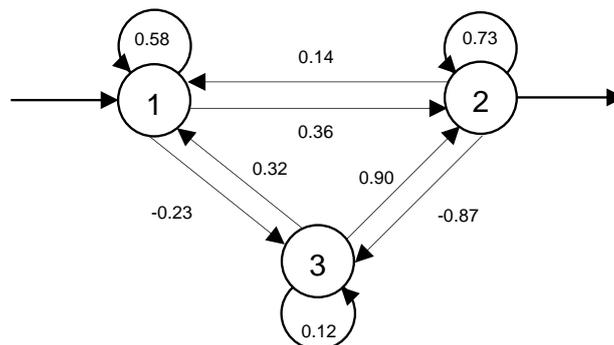


Figura 2.21: RNA con retroalimentación

Las redes retroalimentadas emulan más fielmente la estructura del cerebro humano, en donde los fenómenos de retroalimentación son fundamentales.

2.4.5 Información en las Redes de Neuronas Artificiales

En el cerebro todas las neuronas actúan del mismo modo. Una neurona concreta no puede distinguir características como “rojo” o “frío”. Lo único

que una neurona biológica distingue es el potencial de membrana, su estado de activación y sus secuencias de disparo. Estos tres últimos elementos pueden ser expresados numéricamente.

La información de entrada en números tiene que ser adecuada para las funciones de activación y transferencia de la red y se deben distribuir esas informaciones O_j entre las neuronas de la capa de entrada de forma adecuada.

Se observa, por tanto, que a la hora de diseñar una red neuronal es necesario realizar un análisis tanto de la información de entrada que hay que suministrarle para presentarle el problema, como de la información de salida que la red proporcionará como solución a dicho problema.

Si los datos de entrada, es decir, la información del problema que se desea que la red de neuronas resuelva, no se pueden representar mediante valores dicotómicos, se tendrán que modificar las funciones de la red para hacerla así capaz de trabajar con valores O_j con los que se puedan representar los datos.

Si, por ejemplo, se trabaja con información incierta, en donde sólo se pueda dar un valor de la probabilidad de que un determinado elemento pertenezca a un conjunto, se necesitará que la función sea continua para poder expresar valores de probabilidad. De esta manera el rango de los O_j variará de 0 a 1.

Si los valores de O_j son continuos y los datos también, se tendrá que volver a normalizar los datos de entrada para hacerlos corresponder con el rango de valores que los O_j pueden tomar. Es decir, es necesario reajustar la escala de los datos para que se adapte a la escala de los O_j . Así, si por ejemplo un dato de entrada es una longitud en metros (desde 0 a 1.000 m.) y los O_j se

mueven desde -1 a 1 se tendrá que establecer una correspondencia en la que a la longitud de 0 m. le corresponda -1, a 1.000 m. le corresponda +1, a 500 m. le corresponda 0, etc.

En cuanto al proceso de normalización se pueden realizar tres clases de transformación de los datos (Shi, 2000): transformación lineal, estandarización estadística y funciones matemáticas. La transformación lineal es la técnica más empleada, en este caso, se escalan los valores al rango $[0, 1]$ o $[-1, 1]$ de forma lineal. El objetivo es asegurar que todas las variables (entradas de la RNA) reciben igual atención en el proceso de entrenamiento y son escaladas de acuerdo a los límites de las funciones de activación utilizadas en las neuronas de la capa de salida de la red.

En la estandarización estadística, el escalado involucra la resta de una cierta medida, como por ejemplo la media aritmética, y dividir por un cierto valor, como por ejemplo la desviación estándar. Cualquier escalado que establezca una tendencia central próxima a cero suele ser beneficioso en el proceso de inicialización de los pesos de la red.

La última transformación se basa en funciones matemáticas como por ejemplo aplicar la función logaritmo sobre los datos, con el objetivo de estabilizar la variancia (Faraway & Chatfield, 1998). El aplicar el logaritmo o la raíz cuadrada, se usa principalmente para aproximar los datos a distribuciones Gaussianas con el objetivo de minimizar el efecto de los valores extremos del rango.

Se ha demostrado (Bowden, Dandy, & Maier, 2003; Shi, 2000) que transformar las entradas de la red para que sigan una distribución uniforme puede ayudar a un mejor proceso de conversión y transformación en los valores de salida por la red de neuronas y, en consecuencia, un mejor proceso

de entrenamiento de la red. Una forma de conversión a una distribución uniforme cuando los datos no siguen ninguna distribución probabilística conocida es la ecualización del histograma (Looney, 1997).

2.4.6 Tipos de problemas

Una vez decidido como representar la información, el siguiente paso es considerar qué tipo de problema va a resolver la red. La mayoría de los problemas se dividen en dos grandes grupos y, consecuentemente, existen dos grandes tipos de redes de neuronas: redes de clasificación y redes de evaluación. Las redes del primer tipo, asocian una determinada configuración de entrada, o patrón de entrada, con una salida. Se denomina a estas redes, redes asociativas, clasificadoras o reconocedoras de patrones. El segundo tipo de redes, maneja problemas en los que lo que se pide es un “juicio”, por lo que se denominan redes evaluadoras.

Otro grupo de Redes de Neuronas Artificiales lo constituyen modelos específicos para tareas especiales como: restauración de patrones, predicción, etc.

2.4.7 Fase de Entrenamiento

Norbert Wiener en su libro “Dios y el Golem” (Wiener, 1964) da tal vez una de las definiciones más neutrales de aprendizaje de todas las conocidas. Allí definió un sistema que aprende como: “Un sistema organizado puede definirse como aquel que transforma un cierto mensaje de entrada en uno de salida, de acuerdo con algún principio de transformación. Si tal principio está sujeto a cierto criterio de validez de funcionamiento, y si el método de transformación se ajusta a fin de que tienda a mejorar el funcionamiento del sistema de acuerdo con ese criterio, se dice que el sistema aprende”.

Esta definición es válida para definir el aprendizaje bajo los dos puntos de vista en que éste puede entenderse: el ontogenético, que es el aprendizaje de un individuo humano o no, y el filogenético, aprendizaje que afecta a la especie.

Uno de los principales objetivos de los sistemas autónomos es emular la habilidad que posee el hombre para interactuar con el ambiente y aprender de dichas interacciones. Es así como necesitan de una estructura flexible, capaz de desempeñarse en ambientes de operación dinámicos sujetos a diversas incertezas y perturbaciones. Dentro de las aproximaciones existentes en la teoría de Inteligencia Artificial, surgen las Redes de Neuronas Artificiales como elementos capaces de proveer de dicha estructura flexible, mediante la integración con diversos sistemas de aprendizaje. Tales sistemas están orientados hacia diferentes operaciones y pueden ser clasificados en dos tipos (Brown & Harris, 1995; C. T. Lin & Lee, 1996): aprendizaje supervisado y no supervisado, y dentro de este, aprendizaje auto-organizativo y aprendizaje por reforzamiento (Hoskins & Himmelblau, 1992).

Una vez diseñada la arquitectura de la red (capas y número de neuronas por capa) y las funciones que la regirán, se tiene que proceder a entrenar a la red para que “aprenda” el comportamiento que debe tener; es decir, para que aprenda a dar la respuesta adecuada a la configuración de estímulos o patrones de entrada que se le presenten (Chauvin & Rumelhart, 1995; Pazos, 1996).

Una excepción a esta regla general la constituyen las redes de Hopfield, que no son entrenadas sino construidas, de modo que tengan ya inicialmente el comportamiento deseado. Por este motivo, se ha dicho que las redes de Hopfield simulan el comportamiento “instintivo” mientras que las demás redes simulan el comportamiento “aprendido”.

En la vida de las redes con comportamiento aprendido se distinguen dos periodos o fases claramente diferenciados. Durante la fase de aprendizaje se entrena a la red para que vaya modificando sus pesos sinápticos, adaptándolos paulatinamente para que la respuesta de la red sea la correcta. Después, viene la fase de funcionamiento real o fase de ejecución, durante la cual la red ya es operativa y sus pesos sinápticos no volverán a ser modificados. Durante esta fase se usa la red como si se tratara de cualquier otro programa informático convencional.

2.4.7.1 Aprendizaje supervisado

Con esta técnica de aprendizaje el entrenamiento consiste en presentarle a la red repetitivamente patrones de estímulos de entrada pertenecientes a un juego de ensayo.

El juego de ensayo está formado por parejas “patrón de estímulos - respuesta correcta” y debe de ser elegido cuidadosamente. Cada pareja se denomina hecho. En el juego de ensayo debe estar representada equilibradamente toda la información que la red necesite aprender.

Al realizar el entrenamiento la respuesta que da la red a cada patrón se compara con la respuesta correcta ante dicho patrón y, en virtud de esa comparación, se reajustan los pesos sinápticos. El reajuste de los pesos sinápticos está orientado a que, ante el patrón de entrada, la red se acerque cada vez más a la respuesta correcta.

Cuando ante un patrón de entrada la red de neuronas ya responde correctamente, se pasa al siguiente patrón del juego de ensayo y se procede de la misma manera.

Cuando se termina con el último patrón del juego de ensayo, se tiene que volver a empezar con el primero, ya que los pesos se han seguido modificando.

En casos sencillos, al cabo de unos pocos pasos de entrenamiento completos, con todos los elementos del juego de ensayo, los pesos sinápticos de todas las neuronas se estabilizan en torno a unos valores óptimos. Se dice entonces que el algoritmo de aprendizaje converge. Es decir, después de sucesivas presentaciones de todos los patrones estimulares del juego de ensayo, la red, responderá correctamente a todos ellos y se puede considerar entrenada y dar por terminada la fase de aprendizaje.

2.4.7.2 Aprendizaje no supervisado

En este tipo de aprendizaje, no se especifica a la red cuál debe ser la respuesta correcta; es decir, no hay una comparación entre la respuesta de la red y la respuesta deseada. Además, en este modelo de aprendizaje no existe ninguna influencia externa a la red, puesto que no se le informa de si un resultado fue correcto o no; tan sólo se le suministran grandes cantidades de datos con los que la red pueda construir sus propias asociaciones. Se necesita, por tanto, una cantidad mucho mayor de patrones de entrada durante el entrenamiento para que la red pueda ajustar correctamente sus pesos sinápticos. Por supuesto, los procedimientos de aprendizaje son diferentes a los utilizados con el modelo de entrenamiento supervisado.

En este tipo de aprendizaje, lo que de hecho se está haciendo es exigirle a la red que capte por sí misma alguna de las características presentes en los datos de entrada.

Evidentemente, muchos de los aprendizajes básicos que realizan los sistemas biológicos son de este tipo. Los recién nacidos (al igual que los ciegos de nacimiento que recuperan la visión en edad adulta) aprenden a organizar los datos visuales sin ayuda de “profesor” que les indique para cada patrón de estímulos de entrada, cual es la organización - interpretación correcta de dichos estímulos; es decir, la respuesta del subsistema neurológico de visión que servirá a su vez de entrada a otros subsistemas.

De hecho, en el aprendizaje no supervisado se pretende que las neuronas se auto-organicen aprendiendo a captar las “regularidades” de los datos de entrada sin suministrarles ningún tipo de criterio o ayuda externa que dirija dicha autoorganización.

2.4.8 Selección del juego de ensayo

En vista de la gran cantidad de información que se le puede proporcionar a una red, se ha de buscar un criterio de selección para crear el juego de ensayo.

En el juego de ensayo debe de haber suficientes hechos, es decir, parejas “patrones de estímulos - respuesta correcta”. Además, los hechos del juego de ensayo deberán cubrir ampliamente la totalidad de las características a los que la red debe de enfrentarse. El problema de decidir cuántos hechos se han de incluir en el juego de ensayo es un problema de “Teoría de Muestras” al que se está buscando solución. Un problema muy similar se plantea al establecer, en base a casos de prueba, el grado de experiencia de un experto.

Por otra parte, a una red evaluadora es importante mostrarle tanto los patrones de entrada que llevan a evaluaciones positivas, como los patrones de entrada que llevan a evaluaciones negativas. Es decir, el entrenamiento de la red debe incluir situaciones que se evalúen negativamente, pues de lo

contrario la red simplemente aprenderá que todo está correcto siempre. También se debe incluir en el juego de ensayo cada uno de los casos en los cuales el valor de una entrada es causa de un “mal” resultado.

Por otra parte, no se puede incluir en el juego de ensayo una colección exageradamente grande de hechos. Es necesario seleccionar aquellos hechos que reflejen claramente cada uno de los patrones a reconocer y las situaciones extremas de evaluación en una red evaluadora.

Lo ideal es preparar una colección amplia de hechos de entrenamiento que cubran todos los problemas a los que se pueda tener que enfrentar la red. A continuación, se seleccionarán algunos de ellos para el juego de ensayo, procurando que todos los problemas queden bien representados.

2.4.9 Fase de ejecución

Tras la fase de entrenamiento viene la fase de ejecución, durante la que se le pedirá a la red que responda a estímulos diferentes a los presentados durante la fase de entrenamiento. Gracias a los ejemplos aprendidos del juego de ensayo, la red deberá ser capaz de generalizar y dar respuestas correctas ante patrones de estímulos nuevos.

En otras palabras, una vez terminado el aprendizaje, una red puede generalizar; es decir, ante entradas similares a las de su juego de ensayo, producirá salidas correctas. Hay que tener en cuenta que es muy difícil conseguir la capacidad de generalización de una red sin utilizar grandes cantidades de datos y que estos sean muy variados.

Para operar con una red entrenada, el proceso es el mismo que cuando se realizaba el entrenamiento. Por lo tanto, se le sigue suministrando información de entrada a la red, sólo que ahora no se realizará ningún ajuste

en los pesos sinápticos. La red reconocerá o evaluará y dará una respuesta a partir de los pesos establecidos durante el entrenamiento.

Para conseguir el mejor rendimiento de generalización, los datos usados para el entrenamiento deben cubrir un rango de hechos suficientemente amplio. En general, cuando aumenta el tamaño y variedad del juego de ensayo disminuye la necesidad de que los datos de entrada durante la fase de trabajo normal se parezcan mucho a los patrones del juego de ensayo; es decir, la red generalizará mejor. Si los datos de un problema se diferencian demasiado de todos los patrones del juego de ensayo, la red tendrá dificultades para encontrar la respuesta correcta.

Capítulo 3

Estado de la Cuestión

Invariablemente todo método de selección de variables debería conllevar la búsqueda de la solución a lo largo de un espacio de búsqueda multimodal. Ello es debido a que, aunque un problema presente únicamente una solución global, serán múltiples las combinaciones de variables que ofrezcan unos resultados, sino iguales, al menos sí similares.

En la inmensa mayoría de aplicaciones reales interesará conocer no sólo la solución global al problema, sino la mayoría de soluciones locales óptimas que se encuentren durante el proceso de resolución de dicho

problema. Tal y como se ha comentado anteriormente, esto permitirá no sólo extraer de un conjunto de variables inicial varios conjuntos de variables que aportan individualmente una mayor cantidad de información, sino también extraer conocimiento adicional del dominio del problema en base a qué variables forman parte de dichos conjuntos.

En los siguientes apartados se hace un breve recorrido por las técnicas más representativas empleadas en ambos problemas, tanto en la selección de variables, como en el tratamiento de la multimodalidad.

3.1 Métodos Matemáticos de Selección de Variables

A continuación se realiza una breve descripción de los métodos matemáticos más frecuentemente empleados para la realización de procesos de selección de variables.

3.1.1 Análisis de Componentes Principales

El Análisis de Componentes Principales es una técnica estadística de síntesis de la información, o reducción de la dimensión (número de variables) (Jolliffe, 2002). Es decir, ante un conjunto de datos multivariable, el objetivo será reducir en lo posible el número de variables perdiendo la menor cantidad de información posible.

Los componentes principales extraídos, denominados factores, serán una combinación lineal de las variables originales. Además serán independientes entre sí.

Un aspecto clave en componentes principales es la interpretación de los factores. Ésta no viene dada a priori, sino que será deducida tras observar

la relación de los factores con las variables iniciales (habrá, pues, que estudiar tanto el signo como la magnitud de las correlaciones). Esta tarea no es en absoluto trivial y será de vital importancia el conocimiento que el experto tenga sobre la materia de investigación.

A continuación se detallan cada una de las fases de un proceso de análisis de componentes principales clásico (Shlens, 2005):

- **Análisis de la matriz de correlaciones:** Un análisis de componentes principales tiene sentido si previamente se constata la existencia de altas correlaciones entre las variables. Este hecho será indicativo de que existe información redundante y, por tanto, mediante un número reducido de factores se podrá explicar gran parte de la variabilidad total.
- **Selección de los factores:** La elección de los factores en análisis de componentes principales se realiza de tal forma que el primero recoja la mayor proporción posible de la variabilidad original; el segundo factor debe recoger la máxima variabilidad posible no recogida por el primero, y así sucesivamente. Del total de factores se elegirán aquellos que recojan el porcentaje de variabilidad que se considere suficiente. Al conjunto de factores que aporten el total de variabilidad explicada se les denomina componentes principales.
- **Análisis de la matriz factorial:** Una vez seleccionados los componentes principales, éstos se representan de forma matricial. Cada elemento de dicha matriz representa los coeficientes factoriales de las variables, es decir, las correlaciones entre las variables y los componentes principales. La matriz tendrá tantas columnas como componentes principales y tantas filas como variables.

- **Interpretación de los factores:** Para que un factor sea fácilmente interpretable debe tener las siguientes características, que por regla general, no son fácilmente alcanzables:
 - Los coeficientes factoriales deben ser próximos a 1.
 - Una variable tendrá coeficientes elevados sólo con un factor.
 - No deben existir factores con coeficientes similares

- **Cálculo de las puntuaciones factoriales:** Las puntuaciones que tienen en cada caso los componentes principales son las que permitirán la representación gráfica. Éstas se obtienen mediante la siguiente expresión

$$X_{ij} = a_{i1} \cdot Z_{1j} + \dots + a_{ik} \cdot Z_{kj} = \sum_{s=1}^k a_{is} \cdot Z_{sk} \quad (\text{Eq. 10})$$

dónde a representa a los coeficientes y Z representa los valores estandarizados que tienen las variables para cada uno de los sujetos de la muestra.

Uno de los mayores inconvenientes de PCA reside en que extrae de un conjunto de datos aquellas variables que permiten explicar la mayor parte de las relaciones entre ellos, pero esas variables no pertenecen al conjunto original de variables, sino que son una combinación lineal de ellas.

3.1.2 PLS

Uno de los métodos más usados actualmente para calibración multivariada es el de regresión parcial mediante mínimos cuadrados (PLS ó PLSR) (Abdi, 2003; Barker & Rayens, 2003; Rosipal & Kramer, 2006), el cual ha ganado importancia en numerosos campos, principalmente de la Química (Analítica,

Física, Clínica y control de procesos industriales) (Beebe, Pell, & Seasholtz, 1998; Naes, Isaksson, Fearn, & Davies, 2002).

El punto de partida conceptual sería buscar no sólo las direcciones de mayor información en el conjunto de las variables experimentales (o predictoras), matriz X , sino también seleccionar aquellas que guarden mayor relación con la(s) variable(s) a predecir, matriz Y . Ésta es, de hecho, la idea que subyace en la técnica de PLS, desarrollada por Herman Wold para la econometría a mediados de los años 70 (Martens & Martens, 2001; Wold, 1975).

PLS es, pues, una metodología matemática usada para establecer un modelo que relacione la información de dos conjuntos de datos diferentes (pero ligados de alguna forma desconocida). El modelo de PLS intenta extraer la información importante de ambas fuentes con la única condición de que tales informaciones sean relevantes para establecer la relación entre ambos conjuntos de datos. Esto se consigue desarrollando un modelo de regresión sobre “variables observadas indirectamente”. Cada una de estas nuevas “variables” (llamadas Variables Latentes -VL- o factores) se calcula como combinación lineal de las variables originales bajo la condición de que estén correlacionadas con la variable (o variables) a predecir. De esta forma, los modelos de PLS se rigen por un criterio de capacidad predictiva más que por el ajuste del modelo a los datos.

Esto se logra gracias a que los algoritmos empleados relacionan iterativamente la información relevante en la matriz X con la de Y , siempre y cuando el modelo mejore su capacidad predictiva. Lo cierto es que la explicación del algoritmo de PLS no es intuitiva y debe hacerse mediante el análisis matemático (aquí, necesariamente, breve) del mismo.

Entre las ventajas aportadas por PLS cabe resaltar que las variables latentes (VL) se extraen de forma que sucesivamente expliquen menos información, por lo cual llegará un momento en que se pueda dejar de introducir variables latentes en el modelo sin perjudicar los resultados finales. Una de sus ventajas es que son ortogonales, lo que puede simplificar la interpretación final del modelo.

El hecho de trabajar con variables latentes permite "visualizar" las muestras en este nuevo subespacio. Por similitud con PCA, a las proyecciones de las muestras en las VL se les llamará scores. También las variables se pueden analizar con una cierta facilidad ya que cada VL recoge las relaciones existentes entre ellas y cada variable (original) tiene un peso que describe su contribución a la VL en cuestión. Tanto los scores como los pesos son muy útiles para estudiar los modelos de PLS.

Una característica esencial de PLS estriba en que puede reducir la influencia de factores dominantes pero irrelevantes para el modelo y, en algunos casos, se reduce la dimensionalidad, lo que hace que la interpretación sea más fácil (Hubert & Branden, 2003). Una de las mayores ventajas de PLS es que puede desarrollar modelos que predigan simultáneamente varios parámetros (a diferencia de los no basados en este método que sólo pueden predecir un parámetro por modelo). Mediante PLS sólo hay que establecer también las VL en el bloque de las Y (como se indicó en el algoritmo). Al procedimiento se le denomina PLS de 2 bloques o, simplemente, PLS2, mientras que si se predice una sólo Y, se denomina PLS1 (Brereton, 2000).

También existen inconvenientes al aplicar la técnica de PLS. El primero es que la interpretación del modelo puede ser difícil, aunque es cierto que las

características de ruido se encuadran en las últimas variables latentes y no deben usarse en las modelizaciones.

Otro inconveniente es que, a pesar de la sofisticación de los algoritmos, la base está en ajustes por mínimos cuadrados. Por lo que en casos donde exista una excesiva colinealidad entre las variables, los resultados pueden no ser buenos.

3.1.3 *K-means*

Bajo el nombre de K-Means se hace referencia a una técnica de clústering propiamente dicha, íntimamente ligada al proceso de selección de variables.

En líneas generales el método de K-means tratará de agrupar una colección de muestras en K grupos (denominados *clústers*) diferentes basándose en diversas medidas de distancia (Hartigan & Wong, 1979).

Para realizar dicha agrupación, los pasos generales que realiza el algoritmo son los siguientes:

- 1) El algoritmo se inicia creando k clústers diferentes. El conjunto inicial de muestras es inicialmente distribuido de manera aleatoria entre los clústers creados.
- 2) Se calcula una medida de la distancia existente entre cada una de las muestras de los diferentes clústers y el centroide del clúster correspondiente.
- 3) Las muestras se mueven al clúster K_c que minimiza la distancia entre dicha muestra y el centroide del clúster.

Generalmente, la ejecución del algoritmo de K-means se repite un cierto número de veces hasta obtener una solución óptima. Cada ejecución se inicia con un conjunto aleatorio de clústers diferente.

Las medidas más ampliamente utilizadas de distancia son las que a continuación se detallan:

- **Distancia Euclídea:** es la más usual, natural e intuitiva manera de establecer la distancia entre dos muestras. La distancia euclídea entre dos muestras es simplemente la magnitud del segmento más corto que une ambas muestras. Este tipo de distancia suele emplearse en conjuntos de datos normalizados o que no presentan ningún problema en cuanto a la distribución de las muestras.

$$D_e = \sqrt{\sum_{i=1}^M (x_i - y_i)^2} \quad (\text{Eq. 11})$$

- **Distancia Manhattan:** especialmente relevante cuando se trabaja con conjuntos discretos de datos. Mientras la distancia euclídea determina la longitud del camino más corto entre dos muestras, la distancia de Manhattan determina la suma de distancias a lo largo de cada una de las dimensiones.

$$D_M = \sum_{i=1}^M |x_i - y_i| \quad (\text{Eq. 12})$$

- **Distancia Hamming:** generalmente empleada cuando la codificación de las muestras se realiza en binario, mide el número de valores diferentes entre las dos muestras.

3.1.4 Limitaciones

La selección de variables mediante métodos matemáticos presenta principalmente dos limitaciones.

Por un lado, la aplicación de este tipo de métodos requiere un amplio conocimiento del dominio del problema. Por ello, en función de determinadas características de los datos con los que se esté trabajando (linealidad, interdependencia, etc.), únicamente será posible aplicar unos u otros métodos.

Por otra parte, este tipo de métodos tiende a aportar una única solución válida, no un conjunto de soluciones. Por lo tanto, no serían eficientes en problemas en los que existen variables soluciones globales, o una solución global y varias locales.

3.2 Métodos Evolutivos de Selección de Variables

Desde el ámbito de la computación evolutiva han sido múltiples las aproximaciones al problema de selección de variables (Gestal, Vázquez-Naya, & Ezquerro, 2008). La mayoría de las veces se trata de aproximaciones no genéricas, es decir, desarrolladas específicamente para el problema que en cada caso concreto que se trata de resolver. Aunque desde prácticamente todas las vertientes de la Computación Evolutiva se han realizado aproximaciones al presente problema, de entre ellas cabe destacar las siguientes.

3.2.1 Algoritmos Genéticos Clásicos

Se trata quizá del campo en el que más trabajo existe con respecto a la extracción de características, por lo que han sido ampliamente descritos en el apartado 2 de la presente Tesis.

Se han aplicado con éxito en los más diversos campos, entre los que cabe citar la bioinformática (Saeys et al., 2007), la clasificación de patrones (Peterson, Raymer, & Lamont, 2005) o reconocimiento de voz (Casale, Russo, & Serrano, 2007) entre otros.

La mayor parte de las aproximaciones se basan en la adecuación de la función fitness al clasificador empleado, es decir, se trata de aproximaciones *filter*. Sin embargo, están cobrando especial importancia los métodos híbridos que emplean por un lado Algoritmos Genéticos y por otro mecanismos de clasificación independientes (aproximaciones *wrapper*) entre los que cabe destacar SVM (siglas de Support Vector Machine) (Cristianini & Shawe-Taylor, 1999), (Talbi, Jourdan, Garcia-Nieto, & Alba, 2008; Tan et al., 2008; Yu & Cho, 2006).

Otra aproximación consiste, en vez de adaptar la función fitness para realizar un proceso de selección de variables, modificar la propia estructura de funcionamiento del Algoritmo Genético. Así, en las siguientes subsecciones se muestra una visión global de las modificaciones más comúnmente empleadas en selección de variables.

A la hora de codificar los problemas en forma de genotipo suele emplearse principalmente una aproximación no guiada, aplicable no sólo a los Algoritmos Genéticos sino también a cualquier técnica basada en Computación Evolutiva tras las ligeras modificaciones necesarias para adecuar el formato de representación de las soluciones.

Utilizando una estrategia de búsqueda no guiada, la longitud del genotipo vendrá determinada por la cardinalidad del conjunto inicial de variables. En dicho genotipo, generalmente codificado de forma binaria, cada variable tendrá asignado por lo tanto un gen. El valor de dicho gen

determinará si dicha variable se considera significativa ($gen=1$) o si por el contrario no se considera relevante ($gen=0$). A la hora de determinar el valor de ajuste de cada individuo genético será posible tener en cuenta la fiabilidad del sistema clasificador empleando únicamente las variables señaladas por el Algoritmo Genético como significativas, pero también se podrá considerar el número total de variables empleadas. De esta manera se podrá favorecer a aquellos individuos que, aún presentando unos resultados de clasificación ligeramente inferiores, los alcanzan empleando un número de variables significativamente inferior.

Por ejemplo dado un problema de selección de variables en el que se parte de un conjunto de 6 variables iniciales, un posible individuo genético válido sería 100110. Este individuo estaría indicando que las variables 1, 4 y 5 deberían considerarse significativas, mientras que el resto deberían de ser omitidas en el modelo final.

3.2.2 *Messy Genetic Algorithms*

Los Algoritmos Genéticos se han aplicado con gran éxito en los más diversos campos de aplicación. No obstante, para asegurar la convergencia de la ejecución en un óptimo global, es necesario que el problema pueda ser codificado de tal manera que los diferentes *bloques de construcción* – conjunto de parámetros, de tamaño reducido, que generan individuos altamente capacitados – se puedan combinar para formar un individuo lo más cerca del óptimo posible. Si la relación entre los bloques de construcción necesarios para obtener la solución global es débil, en cierto tipo de problemas denominados problemas deceptivos (Goldberg, 1989a; Whitley, 1991), los Algoritmos Genéticos convergen a soluciones parciales o subóptimas. Esta debilidad de los bloques de construcción puede deberse, por ejemplo, a que el

número de parámetros que forma parte de él es demasiado elevado, con lo cual es complejo que se transmitan de manera conjunta a la descendencia sin ser afectados por las operaciones de cruce o mutación. Con la intención de resolver el citado problema surgen los Algoritmos Genéticos Desordenados, o Messy Genetic Algorithms (Goldberg, Deb, & Korb, 1990).

Los Messy Genetic Algorithms combinan el uso de individuos de longitud variable, un esquema de selección en dos fases y operadores genéticos *messy*.

La principal diferencia con los Algoritmos Genéticos clásicos reside en el esquema de codificación de soluciones. Por ejemplo, en un Algoritmo Genético clásico con codificación binaria, la cadena 111 representaría una posible solución (suponiendo genotipos de 3 bits). Bajo un enfoque *messy* los cromosomas están formados por pares (NúmeroGen, ValorGen). Así serían posibles las representaciones del siguiente tipo ((1 1) (2 1) (3 1)), de tal manera que cada gen se identifica por su nombre y su valor. Esto posibilita desordenar los bits dentro de un cromosoma de manera que sea más sencillo mantener juntos aquellos que formen parte de un mismo bloque constructivo, aún estando en posiciones alejadas dentro del cromosoma.

Adicionalmente, será posible la definición de cromosomas que definan un número mayor o menor de valores para los genes por lo que el cromosoma está formado. Es decir, siguiendo con el ejemplo anterior, podrían considerarse válidos el individuo ((1 1) (3 1) (2 1) (1 0)) y el individuo ((1 1) (2 1)), a pesar de que en el primer individuo el gen 1 aparezca duplicado o de que el segundo individuo no proporcione un valor para el tercer gen. En el primer caso (sobre-especificación), se opta como valor del gen el primer valor que se lee cuando se recorre el cromosoma de izquierda a derecha. La interpretación del individuos similares al segundo ejemplo expuesto (sub-

especificación) requiera la definición del uso de plantillas competitivas (Goldberg, Deb, Kargupta, Harik, & Forrest, 1993).

Otra diferencia significativa es la relativa a la definición del proceso evolutivo en sí. Por lo general en un Algoritmo Genético clásico se genera una población inicial de manera aleatoria y, las siguientes generaciones – obtenidas tras procesos de cruce, mutación, copia, etc. – mantienen el mismo tamaño. Los Algoritmos Genéticos *messy* dividen el proceso evolutivo en dos fases bien diferenciadas: una fase primordial y una fase yuxtaposicional. Durante la primera fase la población se llena progresivamente con la mayor cantidad posible de bloques de construcción óptimos. A continuación durante la fase yuxtaposicional se aplican los operadores de selección, corte (divide en dos mitades un cromosoma) y unión (une dos cromosomas en un único descendiente).

La aplicación de Algoritmos Genéticos Desordenados ha obtenido brillantes resultados en diversos y diferentes campos directamente relacionados con la selección de variables, como son la predicción de estructuras (Lamont & Merkle, 2003; Sun, Bebis, & Miller, 2004), la caracterización de textos (Xu, Da-You, & Hu, 2004), el establecimiento de modelos de calibración para determinar el nivel de calidad del agua (Wu, 2005) o el reconocimiento de patrones (Sun et al., 2004).

3.2.3 Algoritmos de Estimación de Distribuciones

La diferencia más significativa entre los Algoritmos Genéticos y los Algoritmos de Estimación de Distribuciones (EDAs) es que la evolución desde una generación a la siguiente se realiza en el caso de los Algoritmos Genéticos mediante operaciones de mutación y cruce, mientras que en los EDAs esta evolución se realiza en base técnicas basadas en la teoría de la

probabilidad y más concretamente mediante el aprendizaje y la simulación de redes Bayesianas o redes Gausianas.

Adicionalmente, mientras que en el resto de algoritmos evolutivos las interrelaciones entre los genes se mantienen implícitas, en los EDAs dichas interrelaciones se expresan de manera explícita a través de la distribución de probabilidad conjunta asociada a los individuos seleccionados en cada generación. De hecho, dicha estimación de la distribución de probabilidad conjunta de los individuos seleccionados es el verdadero cuello de botella de esta aproximación. (Larrañaga & Lozano, 2002; Lozano, Larrañaga, Inza I., & Bengoetxea, 2006)

El proceso de ejecución de un EDA sigue esencialmente pasos especificados a continuación:

- 1) Primeramente, se genera la población inicial D_0 formada por R individuos. La creación de estos R individuos se realiza a menudo asumiendo una distribución uniforme en cada variable o gen.
- 2) Para evolucionar la $(n-1)$ -ésima población D_{n-1} hacia la siguiente D_n , se seleccionan N individuos ($N < R$) de D_{n-1} siguiendo un criterio. Se denomina D_{n-1}^N al conjunto de los N individuos seleccionados de la generación número $n - 1$.
- 3) Se induce el modelo gráfico probabilístico n -dimensional que mejor representa las interdependencias entre las n variables. Este paso es conocido como el del *aprendizaje*, y es el más crucial de los EDA debido a la importancia de tener en cuenta todas las dependencias entre variables para asegurar una evolución satisfactoria hacia individuos más válidos.

- 4) Finalmente, la nueva población D_n se constituye con R nuevos individuos obtenidos tras *simular* la distribución de probabilidad aprendida en el paso previo. A menudo se utiliza una aproximación elitista, manteniendo así el mejor individuo de la población D_{n-1}^N en la nueva población D_n .

Los pasos 2, 3 y 4 se repiten hasta satisfacer un criterio de parada concreto, como pueda ser llegar a un número de generación máximo, alcanzar un número máximo de individuos analizados, uniformidad en la población recién generada o no obtener un individuo con un valor de función objetivo mejor tras un cierto número de generaciones.

Se han propuesto una gran variedad de algoritmos que son parte de los EDA. Éstos pueden clasificarse en tres grandes grupos dependiendo de la complejidad del tipo de dependencias entre variables que tienen en cuenta:

- **Sin interdependencias entre variables:** estos EDAs se basan únicamente en distribuciones univariantes $\rho(x_i)$. Esto significa que la estructura en forma de red Bayesiana (o Gausiana si se trabaja en el dominio continuo) es fija y no contiene arcos. Es decir, que todas las variables del individuo se consideran independientes entre sí.

Ejemplos significativos de algoritmos pertenecientes a este grupo son UMDA (Univariate Marginal Distribution Algorithm) (Mühlenbein, 1997), donde la estimación de la distribución de probabilidad se realiza de la siguiente manera:

$$p(x) = \prod_{i=1}^n p(x_i) \quad (\text{Eq. 13})$$

Otro ejemplo es el conocido como UMDAc (Univariate Marginal Distribution Algorithm - continuous) (Larrañaga, Etxeberría, Lozano,

& Pena, 2000) equivalente a UMDA aunque en este caso corresponde al dominio continuo.

- **Con dependencias a pares entre variables:** los EDAs pertenecientes a este grupo están basados en distribuciones univariantes $\rho(x_i)$ y también en condicionales de segundo orden $\rho(x_j|x_i)$. La diferencia más significativa con respecto al grupo anterior es que la estructura de la red Bayesiana (o Gaussiana) puede ser diferente, aunque cada una de las variables puede tener como mucho un padre.

Un ejemplo de EDAs discretos pertenecientes a este grupo es MIMIC (Mutual Information Maximization for Input Clustering) (De Bonet, Isbell, & Viola, 1997) que propone realizar la siguiente factorización de la probabilidad:

$$p(x) = p(x_{i_1} | x_{i_2}) \cdot p(x_{i_{12}} | x_{i_3}) \cdots p(x_{i_{n-1}} | x_{i_n}) \cdot p(x_{i_n}) \quad (\text{Eq. 14})$$

De nuevo, existe una versión continua de MIMIC llamada MIMICc (Mutual Information Maximization for Input Clustering - continuous) (Larrañaga et al., 2000).

- **Dependencias múltiples entre variables:** los EDAs pertenecientes a este grupo consideran tanto distribuciones univariantes como condicionales de orden dos o superior, y por lo tanto las estructuras de redes Bayesianas o Gaussianas no tienen ninguna restricción en el número de arcos que pueden contener. Ello requiere una búsqueda exhaustiva de la mejor estructura del gráfico de probabilidades entre todas las posibles, y por lo tanto estos algoritmos son más costosos en tiempo de ejecución que los de los grupos anteriores.

En el dominio discreto, como un ejemplo de EDAs pertenecientes a este grupo puede citarse EBNA (Estimation of Bayesian Networks Algorithm) (Etxeberría & Larrañaga, 1999). Un ejemplo del dominio continuo es EGNA (Estimation of Gaussian Networks Algorithm) (Larrañaga, Lozano, & Bengoetxea, 2001).

3.2.4 Limitaciones

La selección de variables mediante métodos evolutivos soluciona uno de los problemas, anteriormente mencionados, que presentan los métodos matemáticos. La aplicación de métodos evolutivos, minimiza el conocimiento del dominio del problema para su ejecución. Únicamente se requiere conocer una forma válida de codificar las potenciales soluciones, y una forma de validar la bondad de dichas soluciones.

Sin embargo, el funcionamiento tradicional de este tipo de técnicas, sigue adoleciendo de focalizar la búsqueda entorno a la región en la que se sitúa la solución global, descartando otras áreas del espacio de búsqueda, en las que podrían estar localizadas otras soluciones globales o parciales.

3.3 Selección Multimodal de Variables: Tratamiento de la Multimodalidad con Algoritmos Genéticos

Tradicionalmente los Algoritmos Genéticos (AG) se han mostrado como una herramienta válida para la localización de óptimos en funciones unimodales (Harik, 1994; Harik & Eshelman, 1995), puesto que por su propia naturaleza su ejecución converge hacia una única solución dentro del espacio de búsqueda. No obstante, la mayoría de los problemas de optimización reales

están ligados con dominios multimodales en los que la solución requiere encontrar múltiples óptimos, bien locales o globales.

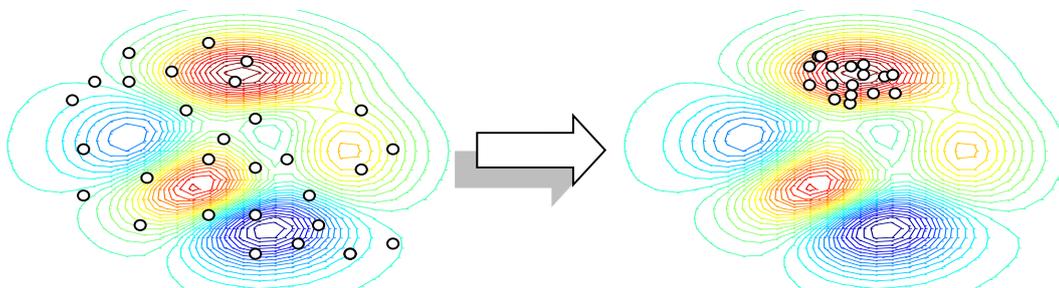


Figura 3.1: Homogeneización de la población genética

Cuando este tipo de problemas se abordan desde una perspectiva basada en Computación Evolutiva, el proceso evolutivo suele provocar la convergencia de toda la población a una zona restringida del espacio de búsqueda abandonando la exploración del resto de picos (óptimos locales). Es lo que se conoce como homogeneización de la población (Lande, 1976; Mahfoud, 1994). El efecto puede verse de manera gráfica en la Figura 3.1. Inicialmente los individuos genéticos (representados por círculos) aparecen uniformemente distribuidos a lo largo del espacio de búsqueda pero, a medida que avanzan las generaciones tienen a concentrarse en el entorno de la mejor solución.

Esta situación puede verse también en la evolución, no de la situación de los individuos sobre el espacio de búsqueda, sino del valor de ajuste de cada uno de ellos. Cuando se produce una situación de homogeneización, cada vez más individuos en la población incluyen la misma información genética. Esto implica que a medida que avanzan las generaciones la aptitud media de la población (la media de los valores de ajuste de todos los individuos) se aproxime cada vez más al valor de ajuste del mejor individuo, tal y como se refleja en la Figura 3.2. Esto ocasiona que cada vez sea más

compleja la obtención de diferentes soluciones que potencialmente exploren nuevas regiones del espacio de búsqueda.

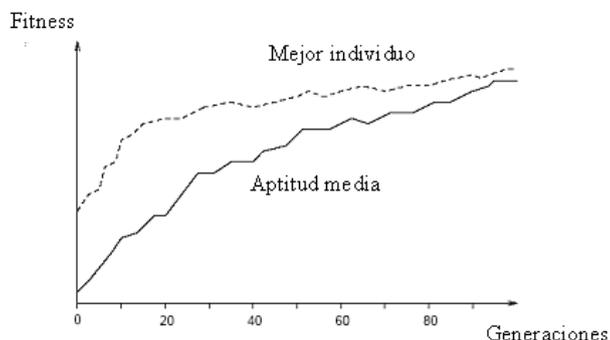


Figura 3.2: Evolución del ajuste ante una situación de homogeneización

Evitar este tipo de situaciones implica mantener la diversidad de la población genética a lo largo de las diferentes generaciones. Para conseguir que esta diversidad se mantenga estable existen diferentes opciones, que pueden englobarse en dos grandes grupos: aproximaciones implícitas y aproximaciones explícitas.

Ejemplos típicos de aproximaciones implícitas son aquellos que propugnan una separación geográfica de los individuos a lo largo del espacio de búsqueda, por ejemplo mediante la aplicación en paralelo del algoritmo sobre poblaciones de individuos aisladas que, cada cierto número de generaciones intercambian individuos entre ellas (Spears, 1994; Storn & Price, 1997). Otro ejemplo sería la constitución de especies dentro de una población. Dichas especies estarían formadas por individuos muy similares entre sí y serviría como ámbito espacial reducido para la aplicación de las diferentes operaciones genéticas (Li, Balazs, Parks, & Clarkson, 2002).

Por otra parte, las aproximaciones explícitas buscan el mantenimiento de la diversidad modificando en menor medida el funcionamiento el Algoritmo Genético clásico. Así, por lo general suelen hacer que los diferentes

individuos de la población genética compitan entre sí por los mismos recursos, por ejemplo, el fitness. En la siguientes subapartados pueden verse más en detalle algunas de estas técnicas de mantenimiento de diversidad.

Con el objetivo de evitar esta situación ante la que los individuos se focalizan en el entorno de la mejor solución, los métodos de niching extienden a los Algoritmos Genéticos favoreciendo la formación de subpoblaciones estables en las cercanías de las soluciones óptimas (Mahfoud, 2000).

Los métodos de niching han sido desarrollados con la intención de reducir el efecto de la homogeneización de la población genética resultante de la aplicación del operador de selección y del uso de elitismo en los Algoritmos Genéticos estándar o canónicos. Su objetivo primordial se centra en mantener la diversidad de la población y permitir al Algoritmo Genético la búsqueda simultánea de manera continuada en múltiples zonas del espacio de búsqueda. Por otro lado, reducen la posibilidad de que la búsqueda no alcance la solución global al problema debido a que se concentre en torno a una única solución local.

La idea subyacente de este tipo de técnicas se basa en la modelización de los mecanismos que tienen lugar en los ecosistemas naturales. En ellos, los animales compiten para sobrevivir a través de la caza, la recolección, ocultándose de posibles presas, etc. especializándose diferentes especies en diferentes aspectos o roles. Por lo tanto, un nicho puede ser visto como un subespacio dentro del entorno que puede soportar diferentes tipos de vida. Una especie podría ser definida como un conjunto de individuos de similares características capaz de relacionarse (o reproducirse) con individuos pertenecientes a la misma especie. Evidentemente, para cada nicho, los

recursos físicos (comida, agua, etc.) son finitos y deben ser compartidos entre toda la población de ese nicho.

Por lo tanto, las técnicas de niching tratarán de conseguir que dentro del entorno, es decir, en el espacio de búsqueda, emerjan diferentes nichos y especies que compitan por los diferentes recursos, es decir por localizar las diferentes soluciones.

El método de fitness sharing es probablemente el mejor conocido y más empleado de entre todas las técnicas de división del espacio de búsqueda en subespecies para la búsqueda paralela en espacios de búsqueda multimodal (Sareni & Krahenbuhl, 1998). Dichas técnicas, tal y como se ha comentado, suelen recibir el nombre de técnicas de niching. Esta técnica fue originalmente ideada por Holland (Holland, 1975) y posteriormente mejorada por Golberg y Richardson (Goldberg & Richardson, 1987). No obstante, existen otras aproximaciones que, una vez vistos los conceptos básicos de fitness sharing, también pasarán a detallarse.

3.3.1 Fitness Sharing

La técnica de fitness sharing modifica la estructura del espacio de búsqueda por medio de la reducción de la aptitud de aquellos individuos pertenecientes a poblaciones con una alta densidad de individuos. Esto se realiza reduciendo el fitness de cada individuo de la población en una cantidad proporcional al número de individuos similares en la población. Típicamente, el fitness shared, denotado f'_i , de un individuo i con fitness f_i , se obtiene aplicando la fórmula especificada por la Eq. 15.

$$f'_i = \frac{f_i}{m_i} \quad (\text{Eq. 15})$$

dónde el denominador m_i representa el número aproximado de individuos con los que el fitness f_i se comparte. Este término se obtiene por medio de la suma de la función compartida sobre todos los miembros de la población, tal y como se muestra en la Eq. 16.

$$m_i = \sum_{j=1}^N f_{Compartición}(d_{ij}) \quad (\text{Eq. 16})$$

siendo N el tamaño de la población y d_{ij} la distancia entre el individuo i y el individuo j . Así, la función de compartición evalúa el nivel de similitud entre dos elementos de una misma población. Dicha función devolverá 1 si los individuos son idénticos, 0 si la distancia entre ellos es superior a un umbral prefijado de desigualdad, o un valor intermedio en función de cuan similares sean los individuos.

La forma genérica, y más usualmente empleada, de dicha función de compartición es la mostrada en la Eq. 17, siendo σ_s el umbral de desigualdad (que puede ser visto como el radio límite de los nichos) y α un parámetro constante que regula la forma de la función de compartición (Goldberg, 1989b).

$$sh(d_{ij}) = \begin{cases} 1 - (d_{ij}/\sigma_s)^\alpha, & d < \sigma_s \\ 0, & \text{sino} \end{cases} \quad (\text{Eq. 17})$$

La distancia d_{ij} entre dos individuos i y j se establece en base a su similitud genotípica o fenotípica. En el primero de los casos lo más común es el empleo de la función de Hamming para la comparación de ambos genotipos. En el segundo caso, la distancia está directamente relacionada con parámetros reales del espacio de búsqueda. En estos casos, suele ser común el empleo de la distancia euclídea. Existen estudios (Deb & Goldberg, 1989) que

indican una ligera mejora en los resultados obtenidos cuando se emplea esta última opción.

Las técnicas de fitness sharing tienden a favorecer la búsqueda en regiones inexploradas del espacio de búsqueda, favoreciendo de esta manera la formación de subpoblaciones estables. No obstante, no está exento de limitaciones:

- La inicialización del parámetro σ_s requiere un conocimiento *a priori* acerca de la distancia existente entre los diferentes óptimos. Sin embargo, en problemas reales de optimización no suele conocerse dicha información. Por otro lado, σ_s es el mismo para todos los individuos, lo que supone que todos los óptimos locales deberían ser equidistantes. Por estas razones, pueden existir errores a la hora de retener todos los óptimos locales del espacio de búsqueda (Deb & Goldberg, 1989; Mahfoud, 1995).
- El método de *fitness sharing* es computacionalmente muy complejo, siendo necesarias del orden de $O(N^2)$ operaciones por generación. Técnicas de análisis de clúster y niching dinámico han sido desarrolladas con el objetivo de reducir dicha complejidad computacional y aumentar la eficiencia (Miller & Shaw, 1996; Yin & Gernay, 1993). No obstante, en muchos dominios el tiempo computacional requerido está fuertemente condicionado por la obtención del valor de bondad de cada individuo, por lo que el *overhead* introducido por las comparaciones entre individuos puede considerarse prácticamente residual.

3.3.2 *Fitness Scaling*

Una posibilidad para mejorar la eficiencia de los métodos basados en *fitness sharing* es el empleo de *fitness scaling* (Goldberg, 1989b). Una función de escalado incrementa la diferenciación entre los óptimos a la par que reduce la decepción (Dasgupta, 1994) es decir, reduce las probabilidades de que un conjunto de genes, que en un individuo originan una elevada aptitud, al combinarse originen descendencia con un menor nivel de bondad. Este comportamiento posibilita que los individuos de una especie puedan concentrarse en el óptimo, en vez de permanecer dispersos por sus cercanías.

Una técnica común de función de *fitness scaling* es el empleo de escalado por potencia. En este caso, la ecuación (Eq. 15) puede ser modificada de la siguiente manera:

$$f'_i = \frac{f_i^\beta}{m_i} \quad (\text{Eq. 18})$$

Esto ocasiona que siga existiendo el problema de escoger el valor adecuado del parámetro β para una función objetivo dada. Si se establece un valor demasiado elevado, la predominancia del fitness escalado puede impedir la reducción de la homogeneización introducida por el método de compartición. Así, la preponderancia de los conocidos como *superindividuos* en la población genética puede ocasionar a una convergencia prematura. Por otro lado, si el valor de β es demasiado reducido, la diferenciación entre los diferentes óptimos del espacio de búsqueda puede ser insuficiente, lo que puede conllevar que el Algoritmo Genético no sea capaz de detectar todos ellos.

El compromiso en la elección del factor de escalado estará directamente relacionado con el correcto balanceo entre los procesos de

exploración y explotación intrínsecos a todos los métodos de búsqueda estocástica.

Debido a la complejidad de obtener un factor β óptimo, algunas aproximaciones emplean técnicas de enfriamiento simulado a lo largo del proceso de búsqueda, con el objetivo de prevenir la convergencia prematura e incrementar la eficiencia del método de compartición (Darwen & Yao, 1995). En otras ocasiones se opta por implementaciones más sencillas basadas en escalar los individuos en base a la posición que ocupan teniendo en cuenta una clasificación de la población basada en el valor de ajuste (*fitness rank*), o en asignar a los n mejores individuos de una población un valor escalado idéntico (*fitness top*), etc. (Kreinovich, Quintana, & Fuentes, 1993).

3.3.3 *Crowding*

Los métodos de *crowding* (que podría traducirse como muchedumbre o manada, aunque es más usual el empleo del término original) se basan en el establecimiento de subespecies de manera no explícita dentro de una población genética a través de la inserción de nuevos individuos en dicha población. Dicha inserción vendrá precedida de la aplicación de un algoritmo de remplazo que será el que determine el tipo de *crowding* empleado.

Así, en función de la estrategia de remplazo escogida, podrá diferenciarse entre *crowding* estándar, *crowding* determinístico y no determinístico o selección por torneo restringido.

- **Crowding estándar:** originalmente desarrollado por DeJong (De Jong, 1975). Establece que únicamente una fracción de la población genética, especificado por un porcentaje G , se reproduce y muere en cada generación. Bajo este esquema de *crowding* la descendencia reemplaza

los individuos más similares a ella en términos genotípicos de entre una subpoblación de tamaño F (factor de *crowding*) de la población global. Debido al gran número de errores de reemplazo, entendiendo por tales reemplazos que originan un descenso en la aptitud media de la población, el *crowding* inicialmente propuesto por DeJong se considera muy limitado en cuanto a su aplicación en espacios de búsqueda multidimensionales (Deb & Goldberg, 1989; Mahfoud, 1995).

- **Crowding determinístico:** Mahfoud mejoró el esquema anterior introduciendo una competición entre descendientes y progenitores de idénticos nichos (Mahfoud, 1995). Una vez realizado el cruce, y eventualmente aplicado el operador de mutación, cada descendiente reemplaza a su progenitor más cercano en términos de fitness. De esta manera, se establece un torneo entre cada uno de los hijos y cada uno de los progenitores. Al contrario que en el *crowding* de DeJong la similitud entre individuos suele establecerse en base a términos fenotípicos. Teniendo en cuenta que serán necesarias 2 comparaciones por cada uno de los torneos y que deberán realizarse $N/2$ torneos en cada generación, el orden resultante de complejidad es de $O(N)$.
- **Crowding no determinístico:** su funcionamiento es prácticamente idéntico al *crowding* determinístico. La peculiaridad se establece a la hora de determinar cuál es el individuo ganador de un torneo. Mientras que en *crowding* determinístico siempre se escogía como ganador al más apto de los individuos participantes en el torneo, en este caso dicha elección se realizara en base a una probabilidad P_c . Así, para cada torneo se generará un número aleatorio en el intervalo $[0..1]$ y, si este es inferior a P_c (que generalmente tomará valores a 0.5) se escogerá como ganador al individuo más apto. En caso contrario, si el

valor generado es superior a P_c , se considerará como ganador al individuo menos adaptado. Ajustando el valor P_c se puede actuar sobre la presión de selección del método, otorgando mayores o menores oportunidades de supervivencia a los individuos menos adaptados.

- **Selección por torneo restringido:** consiste en una adaptación de la selección por torneo estándar para hacerla más adecuada a espacios de búsqueda multimodal (Harik & Eshelman, 1995). Dicha adaptación selecciona inicialmente dos elementos de la población genética y, a continuación, aplica las operaciones de cruce y mutación. Una vez producida la recombinación genética, se escoge aleatoriamente una muestra de N individuos (factor de crowding) al igual que en el *crowding* estándar. Cada uno de los individuos que forma parte de la descendencia participará en un torneo junto con el individuo más cercano o similar de la muestra anteriormente escogida. Los individuos que salgan vencedores de cada uno de los torneos se insertarán en la población. Dicho proceso que se repetirá $N/2$ veces, por lo que la complejidad final del método será de $O(F \cdot N)$, por lo que podrá variar desde $O(N)$ hasta $O(N^2)$ en función del factor de *crowding* escogido.

3.3.4 *Clearing*

El método de *clearing* es muy similar al anteriormente visto de *fitness sharing*, pero basado en el concepto de limitación de recursos existente en cualquier tipo de entorno (C. Y. Lin & Wu, 2002; Petrowski, 1996). En este caso, en lugar de realizar la compartición de recursos entre todos los individuos de una única subpoblación como ocurre en la técnica de *fitness sharing*, el método de

clearing restringe dicha compartición únicamente al subconjunto de los individuos mejor adaptados de dicha subpoblación.

A efectos prácticos, la capacidad k de un nicho especifica el número máximo de individuos que ese nicho puede albergar. De esta manera se preserva el fitness de esos k individuos mejor adaptados, denominados individuos dominantes, y reinicializa el valor de bondad del resto de individuos pertenecientes a la misma subpoblación, los individuos dominados (C. Y. Lin & Wu, 2002).

Al igual que en *fitness sharing*, los individuos pertenecerán a un mismo nicho o subpoblación si su distancia en el espacio de búsqueda es inferior a un umbral de desigualdad σ_s (radio de *clearing*).

El orden de complejidad del procedimiento básico de *clearing* es $O(qN)$, donde q es el número de subpoblaciones que se mantienen durante la búsqueda.

3.4 Motivación

En los apartados anteriores se han presentado diversas técnicas aplicadas hasta el momento con éxito en la tarea de selección de variables. En primer lugar se han mostrado métodos puramente matemáticos. La gran ventaja de estos métodos reside en que, aplicados sobre un mismo conjunto de datos, seleccionarán siempre el mismo conjunto de variables como variables más significativas. Otro de sus puntos fuertes es la fuerte base teórica en la que se sustentan. Sin embargo, no están exentos de inconvenientes. Cabría destacar entre dichos inconvenientes, las restricciones *a priori* que algunos de ellos imponen a los datos (independencia de muestras, grado de solapamiento, etc.), el fuerte conocimiento teórico que por lo general requieren para poder

establecer una relación causa-efecto entre los datos suministrados y las soluciones proporcionadas, la en ocasiones compleja configuración de los parámetros propios de cada método o la alta sensibilidad a los valores de dichos parámetros a la hora de obtener mejores o peores soluciones.

Posteriormente se han descrito diversas aproximaciones basadas en métodos evolutivos. Estos métodos basan su funcionamiento en la imitación de los principios naturales de supervivencia de los individuos más adaptados. Cabe destacar su fácil adaptación a prácticamente cualquier posible problema. Empleando este tipo de técnicas, no es estrictamente necesario saber cómo resolver un problema, basta con tener una forma de determinar cómo de buena es cada una de las soluciones proporcionadas por el algoritmo evolutivo. Es aquí donde reside gran parte de su fortaleza y capacidad de éxito.

No obstante, uno de sus puntos débiles reside precisamente a la hora de proporcionar no únicamente la solución global a un problema, sino también el conjunto de soluciones sub-óptimas del mismo. Por ello, dado que la temática de la presente Tesis se enmarca en la selección de variables considerando espacios de búsqueda multimodal, se ha prestado especial atención a aquellas técnicas evolutivas que precisamente abordan el problema de la multimodalidad.

Tal y como se ha mostrado existen diversas alternativas a la hora de afrontar el problema introducido por la multimodalidad. Sin embargo, ninguna de estas técnicas está especialmente diseñada o enfocada a su uso en problemas relativos a la selección de variables.

Capítulo 4

Hipótesis de Trabajo

En la sección inmediatamente anterior se han presentado una serie de técnicas y métodos destinadas a la realización de procesos de selección de variables. Tal y como se ha mostrado, aunque los resultados obtenidos por dichas técnicas no pueden calificarse de malos, sí puede llegarse a la conclusión de que no alcanzan el nivel que podría llegar a ser considerado como óptimo. Por lo tanto, puede considerarse esta tarea de selección de variables en ambientes multimodales como un campo abierto a la incorporación de nuevas técnicas o métodos de resolución.

De entre las técnicas comentadas, el empleo de Algoritmos Genéticos se ha mostrado como una de las más versátiles, debido primordialmente a sus características intrínsecas de paralelismo y auto-adaptación.

Es por ello que esta Tesis plantea como hipótesis de trabajo el empleo de técnicas de Computación Evolutiva para la resolución del citado problema. Se pretende de esta manera aprovechar precisamente las características mencionadas de paralelismo y auto-adaptación con el objetivo de ofrecer el mayor número de soluciones diferentes posibles a la hora de realizar un proceso de selección de variables en un problema multimodal. Puesto que dichas técnicas tienden a focalizar la búsqueda en el entorno de la mejor solución a medida que avanzan las generaciones, deberá prestarse especial atención a este hecho e intentar minimizar su incidencia para alcanzar los mejores resultados.

Asimismo, también forma parte de la hipótesis de trabajo, el hecho de que los resultados obtenidos ganarán en versatilidad si el método propuesto realiza la evaluación de las soluciones de la manera más independiente posible de cada problema concreto que se pretenda resolver. Se plantea por ello el uso de Redes de Neuronas Artificiales para la realización del citado proceso de evaluación.

Se espera que de esta manera, empleando técnicas de Computación Evolutiva para minimizar el impacto de la dispersión de soluciones a lo largo de un espacio de búsqueda multimodal y realizando la evaluación de soluciones mediante redes de neuronas, los resultados alcanzados mejoren sensiblemente los obtenidos hasta el momento.

Capítulo 5

Selección de Variables en Espacios de Búsqueda Multimodal Mediante Técnicas de Computación Evolutiva

5.1 Metodología Propuesta

Tal y como se comentó en la sección anterior, las técnicas de Computación Evolutiva son especialmente eficaces a la hora de explorar un espacio de búsqueda multimodal, pero presentan la tendencia de converger hacia el

entorno de la mejor solución encontrada a medida que avanzan las generaciones. Esta tendencia supone un claro inconveniente a la hora de ofrecer múltiples soluciones válidas ante un problema multimodal, que puede presentar un conjunto formado por varias soluciones globales y/o parciales.

Por lo tanto, este aspecto deberá ser tenido en cuenta a la hora de desarrollar un sistema de selección de variables – o cualquier otro – que implique la exploración de un espacio de búsqueda multimodal. Deberá conseguirse mantener la diversidad de la población genética con el objetivo de poder generar en cualquier momento soluciones en cualquier parte del espacio de búsqueda, evitándose de este modo la focalización de la búsqueda que inevitablemente conlleva el proceso de convergencia de la población genética. Este aspecto será clave en el desarrollo de cualquier sistema que deba ofrecer soluciones ante un problema multimodal.

Es por ello que en los siguientes puntos se plantean dos técnicas que, aunque presentando puntos de partida totalmente diferentes, persiguen el mismo objetivo de mantener la heterogeneidad de la población genética a medida que avancen las generaciones.

En primer lugar se plantea una modificación de un Algoritmo Genético tradicional, basada en la inclusión de una nueva población genética. Esta nueva población dividirá el espacio de búsqueda en una serie de subregiones, actuando a modo de *repositorio*. Gracias a la existencia de esas subregiones será posible generar soluciones en cualquier zona del espacio de búsqueda, sea cual sea la generación.

La segunda aproximación propuesta es totalmente diferente. En vez de modificar un Algoritmo Genético clásico para poder obtener múltiples

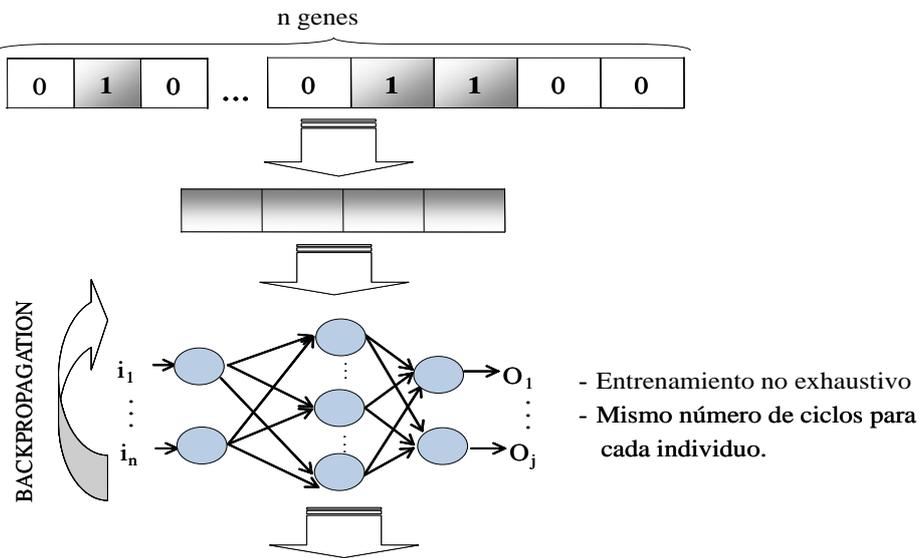
soluciones a partir de su población genética, se parte del empleo de especies de individuos. Cada especie, formada por individuos semejantes entre sí, será la encargada de explorar una zona determinada del espacio de búsqueda y de ofrecer la mejor solución de dicha zona. Asimismo cada especie se comportará de manera similar a cómo se comportan las especies animales en un entorno natural. Por una parte, explorará las inmediaciones de su región en busca de la mejor solución. Por otra parte, evitará que otras especies ocupen la región del espacio de búsqueda que dicha especie ya está explorando.

Adicionalmente, se ha formulado como hipótesis de trabajo la conveniencia de evaluar las soluciones ofrecidas por los métodos evolutivos de la manera más independiente, para incrementar de esta manera la versatilidad del sistema. Con este objetivo se opta por el empleo de Redes de Neuronas Artificiales para realizar dicha tarea. Podría hablarse por lo tanto, de una aproximación *wrapper*, dónde cada posible subconjunto de variables candidato es evaluado por medio del modelo clasificatorio inducido, en este caso la red de de neuronas, a partir del subconjunto que contiene exclusivamente las variables seleccionadas junto con la variable que determina la clase a la que pertenece dicha muestra.

La arquitectura de la red vendrá determinada en parte por la estrategia de selección de variables escogida (ver sección 3.2.1).

Si se escoge una estrategia no guiada, el número de elementos de procesado de la capa de entrada vendrá determinado por el número de variables significativas, es decir con valores no nulos, incluidas en el genotipo del individuo genético (ver Figura 5.1). En el caso de haberse escogido una estrategia guiada, y por lo tanto fijado de antemano el número de variables

que deben considerarse como significativas, el número de genes de cada individuo será el que determine de manera directa el número de elementos de procesado en la capa de entrada.



$$\text{Ajuste individuo}_1 = (\text{error entrenamiento})$$

$$\text{Ajuste individuo}_2 = (\text{error entrenamiento}) * (\text{entradas significativas}) / n$$

Figura 5.1: Evaluación mediante redes de neuronas artificiales

El número de elementos en la capa de salida vendrá determinado por el tipo de problema con el que se esté a tratar. Si se trata de un problema de clasificación, lo más común cuando se requiere un proceso previo de selección de variables, el número de categorías en las que pueden clasificarse los datos de entrada determinará el número de neuronas en la capa de salida. Ante otro tipo de problemas, habrá que adecuar este punto. Una opción sería la de establecer un único elemento de procesado a la salida de la red, que se encargase de proporcionar una medida de bondad de la solución codificada en el individuo genético.

En cuanto al número de capas ocultas y elementos de procesado en cada una de ellas, nuevamente dependerá del problema con el que se esté trabajando. No obstante, no será un aspecto determinante, puesto que tal y como se verá a continuación, no se tratará de realizar un proceso completo de entrenamiento de la red de neuronas.

Una vez configurada la arquitectura de la red, podrá comenzar su proceso de entrenamiento. Para ello se hará uso únicamente de las variables escogidas como significativas por el individuo genético que está siendo evaluado.

El valor de ajuste del individuo podrá determinarse de diferentes maneras:

- Puede asignarse directamente el valor de error cometido durante el proceso de aprendizaje (primera función de ajuste en la Figura 5.1)
- En el caso de poseer un conjunto de datos de validación, podría optarse por asignar como valor de bondad el error cometido por la red con dichos datos de validación
- Si se desea favorecer a aquellos individuos, que aún cometiendo un error de clasificación similar, emplean un menor número de variables declaradas como significativas, podrá modificarse la función de *fitness* para adecuarla a este objetivo (segunda función de ajuste en la Figura 5.1)
- Etc.

Independientemente de la estrategia de selección de variables escogida hay una serie de aspectos que no pueden dejar de tenerse en cuenta.

Por una parte el entrenamiento al que se somete la red no tiene porqué ser exhaustivo. De ser así, los requerimientos computacionales crecerían exponencialmente. En su lugar bastará con alcanzar el inicio de convergencia, con el objeto de determinar si el individuo genético proporciona una buena aproximación o punto de partida para resolver el citado problema.

En segundo lugar, en el caso de que las arquitecturas generadas a partir de dos individuos sean equivalentes (mismo número de capas, mismo número de elementos de procesado por capa) el número de ciclos del citado entrenamiento debería ser idéntico el mismo a la hora de evaluarlos. De esta forma se consigue que los resultados obtenidos para individuos equivalentes sean comparables entre sí.

5.2 Tratamiento de la multimodalidad

A continuación se presentan dos propuestas para la búsqueda de soluciones en un espacio multimodal, punto dónde reside uno de los caballos de batalla de cualquier método de selección de variables. Por lo tanto, tal y como se ha comentado anteriormente, será requisito indispensable el poseer métodos capaces de mantener la heterogeneidad durante el proceso de búsqueda. Es decir, que durante la búsqueda de la solución global no descarten las soluciones parciales que puedan ir apareciendo durante la exploración del espacio de soluciones.

La primera de estas propuestas se basa en una modificación de un Algoritmo Genético clásico, mediante la adición de una nueva población

genética, encargada de mantener la diversidad a lo largo del espacio de búsqueda. La segunda de ellas, parte de un enfoque distinto, como es la creación de especies, cada una encargada de buscar por separado una solución al problema planteado.

5.2.1 Algoritmo Genético de Dos Poblaciones

5.2.1.1 Introducción

Tal y como se ha comentado anteriormente, la evolución de una población de soluciones a lo largo de sucesivas generaciones puede implicar una pérdida de diversidad en la población. Esta situación es especialmente crítica cuando se trabaja con problemas de selección de variables. En este escenario de homogeneización, al ser los individuos similares, sólo la mutación explora regiones nuevas, puesto que el cruce entre ellos aporta información ya conocida y no se generarán nuevas soluciones. Debido a esta similitud genética, un Algoritmo Genético tradicional centrará su búsqueda en determinadas zonas reduciendo la probabilidad de hallar más de una solución e incluso la de encontrar la solución global.

Se debe pensar, por lo tanto, en una forma de poder aplicar el Algoritmo Genético de un modo ordenado y eficiente para explorar la mayor parte del espacio de búsqueda con el objetivo de localizar el mayor número posible de soluciones. Una posible opción consistiría en aplicar varias veces un Algoritmo Genético tradicional a la población para localizar las diferentes soluciones, pero eso dejaría la búsqueda de soluciones en manos del azar y del tiempo.

Por otro lado, se pretende que la búsqueda de soluciones no se estanque en algunas zonas determinadas, por lo que se intenta de alguna forma que el sistema explore nuevos lugares. Una opción sería tomar individuos de la población cuando esta empiece a mostrar signos de homogeneización, y aplicarles el operador de mutación. Sin embargo, únicamente con esto no es suficiente. Por un lado, puede ser que el individuo resultante pertenezca a una zona que ya está siendo explorada por otro conjunto de individuos, con lo que no aportaría nada nuevo a la búsqueda; por otro lado, aunque se genere sobre una zona no explorada, se carece de la certeza de que dicho individuo pueda estar cercano a una solución del problema.

Para evitar esta situación se plantea una opción para mantener la diversidad basada en la existencia de una población externa que actúe a modo de repositorio de información, es decir, una segunda población donde estuviesen representados todos los individuos del espacio de búsqueda, y que obligase a explorar dicho espacio en su totalidad a la hora de resolver el problema.

Por lo tanto, a continuación se expone una variante de los Algoritmos Genéticos que, a grandes rasgos, consistirá en utilizar una segunda población cuya finalidad principal será forzar al Algoritmo Genético a la exploración total del espacio de búsqueda intentando solventar en lo posible los problemas originados por la pérdida de diversidad.

5.2.1.2 Esquema General de Funcionamiento

Tal y como se ha comentado, este sistema consiste en añadir al Algoritmo Genético tradicional una nueva población genética, denominada piscina

genética, con el objetivo de explorar de forma homogénea todo el espacio de búsqueda.

Esta piscina genética divide el espacio de exploración en varias zonas acotadas o subregiones en las que se podrá llevar a cabo la búsqueda de manera independiente. Dicha piscina genética está compuesta por una serie de individuos, dónde cada uno de los cuales representa una subregión del espacio de búsqueda global.

Los individuos de la piscina genética representan subregiones del espacio de búsqueda y, por lo tanto, soluciones potenciales (aunque parciales en el sentido de que estarán restringidas a dicha subregión) del problema. Consecuentemente, el genotipo de dichos individuos deberá poseer una estructura similar a los individuos genéticos de una población tradicional, al menos en cuanto al número de genes. Pero, y aquí radica la diferencia con una población genética tradicional, cada individuo de la piscina tendrá acotado el rango en el que podrán variar sus genes. De esta manera proporcionará soluciones únicamente dentro de una subregión del espacio de búsqueda. Asimismo, el conjunto de los rangos individuales en los que cada gen puede tomar valores, deberá comprender la totalidad de los valores posibles que dicho gen puede tener. Es decir, no habrá ningún posible valor que un gen pueda tomar que no esté representado por algún individuo en la piscina genética. Por consiguiente, el conjunto de individuos que forma la piscina genética explorará el espacio de búsqueda en su totalidad.

Dependiendo de la complejidad del problema que se desee resolver y del tipo del mismo, el número de subregiones en las que se ha de dividir el espacio de búsqueda podrá variar.

Además de la piscina genética, se empleará otra población, denominada población secundaria. En esta población un Algoritmo Genético tradicional irá evolucionando a sus individuos a lo largo de sucesivas generaciones, en interacción con los individuos de la piscina genética, tal y como se comentará más adelante. Esta población secundaria será la que aporte las soluciones al problema, mientras que la piscina genética contribuirá a que la búsqueda de dichas soluciones se realice explorando de forma homogénea el espacio de búsqueda.

Por lo tanto, la piscina genética facilita que el proceso de búsqueda se realice teniendo en cuenta varias subregiones, mientras que un Algoritmo Genético tradicional sólo tiene en cuenta una subregión, que es el espacio de búsqueda en sí mismo.

5.2.1.3 Descripción del Sistema

5.2.1.3.1 Piscina Genética

Como ya se ha explicado en el apartado anterior, cada uno de los individuos constituyentes de la piscina genética representa una subregión del espacio total de búsqueda. Por lo tanto, la estructura de estos individuos tiene que ser la misma que si se emplease un Algoritmo Genético tradicional. Es decir, la misma que las soluciones potenciales del problema, pero con una pequeña diferencia. Los valores que puedan tomar los genes de un individuo de la piscina genética no son los mismos que pueden tomar en el caso de que se emplease un Algoritmo Genético tradicional. En este último caso, los genes pueden tomar cualquier valor del rango total permitido, mientras que los valores de los genes de los individuos de la piscina genética estarán acotados. Para formar la piscina, el rango total de valores se divide entre tantos

subrangos como individuos como se deseen, asignándose así un subrango de valores a cada individuo. Dicho subrango permanece invariable a lo largo de toda la ejecución del Algoritmo Genético.

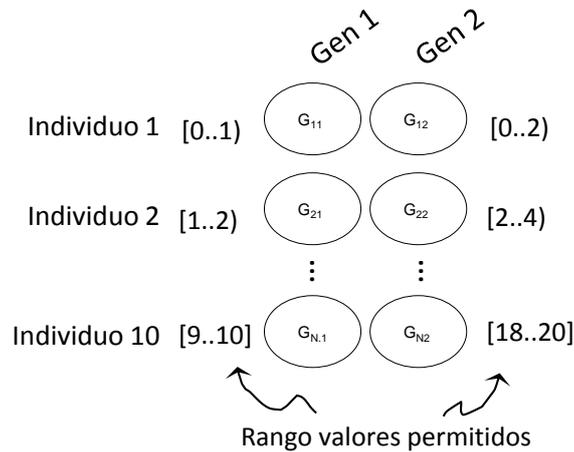


Figura 5.2: Ejemplo de configuración de la piscina genética

En la Figura 5.2 se ilustra un ejemplo de piscina genética en un determinado problema. En el ejemplo concreto, cada posible solución se caracteriza por tener dos genes que comprenden los rangos $[0..10]$ y $[0..20]$ respectivamente. Si se deseara configurar la piscina genética con 10 individuos (10 subregiones del espacio de búsqueda), los subrangos de los genes de los individuos de la piscina quedarían de la manera que muestra la Figura 5.2. Se puede observar que la piscina genética divide el espacio de búsqueda en tantas partes de igual tamaño como individuos haya en la piscina genética. Así, el espacio de búsqueda estará uniformemente distribuido entre todos los individuos.

Además de los valores de sus genes, cada individuo de la piscina genética incluye información adicional. Por ejemplo lleva control acerca de si sus genes pertenecen o no a la mejor solución hallada hasta ese momento, es

decir, al mejor individuo de la población secundaria. Dicho control, se emplea para no modificar aquellos genes que, en un momento determinado de la ejecución, pertenecen a la mejor solución. Además, cada uno de los genes tiene asociado un valor I_i que permite configurar la aplicación del operador de mutación, tal y como se mostrará en un punto posterior.

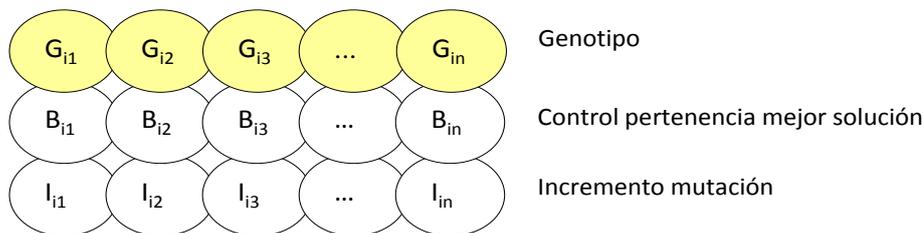


Figura 5.3: Estructura de un individuo de la piscina genética

En la Figura 5.3 puede apreciarse de manera gráfica la estructura de cada uno de los individuos que forma la piscina genética.

Otro punto a tener en cuenta con respecto a la piscina genética es que, al actuar de repositorio de información, no es necesario conocer el valor de ajuste de sus individuos, favoreciéndose de esta manera el rendimiento global del sistema.

5.2.1.3.2 Población Secundaria

A diferencia de los individuos de la piscina genética, que tienen sus genes acotados, los genes de los individuos que forman la población secundaria pueden tomar cualquier valor perteneciente al rango del área de soluciones. Por lo tanto, será esta población secundaria la que aporte las soluciones globales al problema que se desea resolver y no la piscina genética al tener los genes limitados a un determinado rango.

En esta población los individuos evolucionan siguiendo las reglas de un Algoritmo Genético tradicional, pero con una variación: la operación de cruce se realiza entre un individuo de la población secundaria y un representante de la piscina genética, y no entre dos individuos de una misma población como en el Algoritmo Genético clásico. Este proceso se detalla a continuación.

5.2.1.3.3 Operadores Genéticos

Como sucede con los Algoritmos Genéticos tradicionales, la evolución de las soluciones de la población se realiza mediante la aplicación de operadores genéticos. Al igual que en ellos, en el método propuesto también se emplean los operadores genéticos de cruce y la mutación. La diferencia radica en quiénes son los participantes de cada proceso.

- **Operador genético de cruce**

El objetivo del operador de cruce será introducir en la población secundaria la información almacenada en la piscina genética. De esta manera, se mantiene en la primera la diversidad entre los diferentes individuos.

El cruce se realiza mediante la recombinación de la información genética de un individuo de la población secundaria, seleccionado al azar, o a través de cualquiera de los operadores de selección previamente comentados en la sección 2.3.5.1, con la de un representante de la piscina genética. Para formar el representante de la piscina genética se elige al azar para cada uno de sus genes un individuo de la piscina y se asigna el valor correspondiente a dicho gen al genotipo del representante. De esta forma, se logra que el individuo resultante represente una solución global al problema y no parcial como cada uno de los individuos por separado. Es decir, el representante de

la piscina corresponde a una solución válida al problema, aunque esté formado por valores de genes provenientes de la piscina.

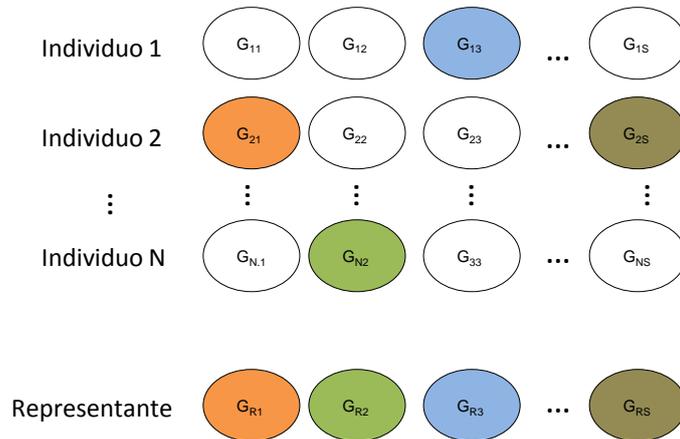


Figura 5.4: Construcción del representante de la piscina genética

En la Figura 5.4 se muestra un ejemplo de la construcción del representante de la piscina genética. En este ejemplo concreto, los individuos de la piscina tienen cuatro genes, al igual que los de la población secundaria. Para el valor del Gen 0 del representante se ha seleccionado al azar el individuo 2, por lo tanto, el valor del gen del representante pertenecerá al mismo subrango que ese individuo de la piscina; para el valor del Gen 1, se ha elegido el individuo N, y así sucesivamente hasta completar el representante completo. Tal y como se aprecia en la figura, un mismo individuo de la piscina genética, en este caso el individuo 2, puede aportar más de un gen al representante de la misma.

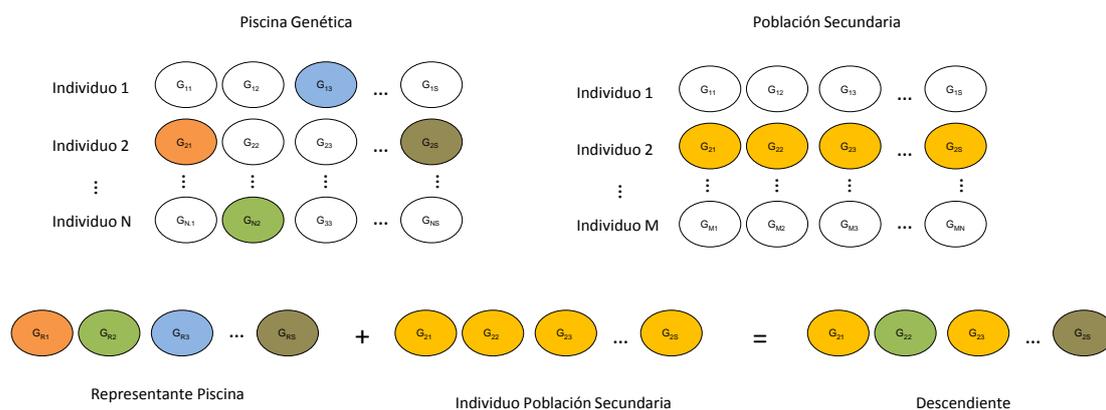


Figura 5.5: Operador de Cruce

Una vez seleccionados los participantes en el cruce, es decir, el representante de la piscina y el individuo de la población secundaria, se procede al cruce o reproducción. En la Figura 5.5 se muestra un ejemplo con cruce uniforme, pero cualquier otro cruce podrá ser empleado para recombinar la información genética entre el individuo de la población secundaria y el representante de la piscina genética. Puesto que únicamente uno de los progenitores procede de la población secundaria, el operador de cruce genera un único descendiente. Una vez generado el descendiente, éste deberá ser insertado en la población secundaria. Para ello, podrá emplearse cualquiera de los algoritmos de reemplazo comentados anteriormente. La estrategia de reemplazo podrá ser destructiva o no destructiva. En el primer caso, el reemplazo tendrá lugar siempre; en el segundo, después de generada la descendencia, ésta pasará a formar parte de la población secundaria únicamente en el caso de que su función de ajuste sea superior a la del progenitor seleccionado de dicha población para el cruce.

Si además el descendiente tiene una bondad o nivel de ajuste superior a la mejor solución hallada hasta ese momento, se activarán los valores B_{ij} de

los genes correspondientes aportados por el representante de la piscina genética. Esto se hace con el objetivo de evitar modificarlos durante las posteriores aplicaciones del operador de mutación.

- Operador genético de mutación

Para aplicar el operador de mutación, en primer lugar se debe seleccionar al azar un representante de la piscina genética de igual modo que se selecciona para el cruce. Es decir, para cada uno de los genes de la población se elige de forma aleatoria un individuo de la piscina genética, pero con la restricción de que dicho gen no debe pertenecer a la mejor solución hallada hasta ese momento, o lo que es lo mismo, no debe tener el valor B_{ij} activado. Una vez elegido el representante de la piscina genética, se procede a la mutación. Ésta consiste en aplicar a cada uno de sus genes los incrementos especificados por el valor I a dichos genes (ver pseudocódigo en la Figura 5.6).

```
IF (not Bij)
  Gij = Gij + Iij
  IF (Gij > LIM_SUP_IND)
    Gij = LIM_INF_IND
    Iij = Iij /2
  END
END
```

Figura 5.6: Pseudocódigo de la operación de mutación

Cuando la suma del valor actual del gen y del incremento excede el límite superior del subrango en el que puede variar dicho gen, LIM_SUP_IND, éste vuelve a tomar el valor inferior del subrango, LIM_INF_IND. También se reduce el valor del incremento que se aplicará en

posteriores mutaciones de dicho gen. El objetivo de esta reducción es lograr un recorrido más exhaustivo sobre todos los valores que un gen puede tomar a lo largo de las sucesivas generaciones. En el algoritmo propuesto, se ha optado por dividir el valor del incremento por dos cada vez que se alcance el límite superior, de tal manera que la precisión para la obtención de valores para un gen se dobla tras cada nueva generación.

La operación de mutación afecta también, aunque en menor medida, a la población secundaria. Para ello se selecciona al azar un individuo de dicha población y se le aplica a uno de sus genes (elegido también de forma aleatoria) el operador de mutación. En este caso, dicha operación consistirá en incrementar el valor de ese gen con un valor aleatorio, siempre que el valor resultante pertenezca al rango de valores válidos. En el caso de la mutación en la población secundaria, se opta por una estrategia elitista. Si el individuo modificado tiene un nivel de ajuste inferior al anterior, se restaura el valor del gen previo a la mutación; si por el contrario el nivel de ajuste se ha mejorado gracias a la mutación, dicha mutación se conserva.

Además, si el nuevo descendiente generado tiene una bondad mayor que la de la mejor solución hallada hasta ese momento, al igual que sucedía con el cruce, se activan los valores B_{ij} de los genes correspondientes (y se actualizan los valores de dichos genes en el resto de la piscina genética en caso de ser necesario) en los individuos correspondientes en la piscina genética, para evitar modificaciones en sus valores durante la aplicación de posteriores operaciones de mutación.

5.2.2 Algoritmo Genético de Dos Poblaciones con Regiones Variables

5.2.2.1 Introducción

Tal y como se ha comentado, aplicar un Algoritmo Genético a un problema multimodal presenta el inconveniente de que tiende a alcanzarse una única solución de todas las posibles que se podrían encontrar. Para lograr alcanzar más soluciones es fundamental explorar el espacio de búsqueda en su totalidad. La variante planteada en el punto anterior ha consistido en incorporar una nueva población genética con el objetivo de mantener la diversidad durante el proceso de búsqueda. La inclusión de una población adicional implica una exploración más en profundidad del espacio de búsqueda, puesto que requiere que el sistema divida el problema en subregiones, en cada una de las cuales se centrará cada individuo en busca de soluciones. Así se evita que la búsqueda se centre en una sola región y, por lo tanto, que se produzca la homogeneización de la población.

No obstante, este método también tiene una serie de limitaciones. Una de ellas es que el número de subregiones en las que se divide el espacio de búsqueda es estático. Teniendo en cuenta que cada individuo de la piscina genética busca una solución en esa subregión, podría ocurrir que dicha subregión incluyese más de una solución. Asignando un único individuo a cada subregión sería posible detectar una única solución por subregión. La solución podría pasar por incrementar el número de subregiones. Sin embargo, de esta manera puede estar sobreexplotándose otras zonas del espacio de búsqueda.

Por lo tanto, lo más interesante sería que durante la ejecución del problema se pudiese dividir el espacio de búsqueda en más subregiones si se desea realizar una exploración más exhaustiva en una determinada zona, o agrupar subregiones si, por ejemplo, no se desea buscar más soluciones en una determinada zona del espacio. Ninguna de estas dos opciones, si el número de subregiones es fijo, se puede llevar a cabo.

Asimismo, el número de subregiones en las que se divide cada gen es idéntico para todos los genes de los individuos y permanece invariable durante toda la ejecución del sistema. Dependiendo de cómo se desarrolle la búsqueda, puede interesar disminuir el número de subregiones únicamente para un determinado gen para que cada una de ellas tenga menos posibilidades de ser seleccionada para la reproducción o, por el contrario, aumentar dicho número si se desea incrementar el porcentaje de ser elegido, o lo que es lo mismo, realizar una mayor exploración sobre el área del espacio de búsqueda dependiente de dicho gen.

Para evitar este inconveniente se propone a continuación una modificación de ese sistema que, además de incluir una piscina genética para una exploración total del espacio de búsqueda, permite automatizar el tratamiento de las subregiones definidas por dicha piscina explorando con la intensidad que se considere necesaria cada zona, solventando de este modo las limitaciones del sistema anterior.

Este sistema permitirá la posibilidad de agrupar y/o dividir el número de subregiones en las que se divide el espacio de búsqueda durante la ejecución de la aplicación, es decir, el número de subregiones en las que se descompone cada gen. De este modo, según los resultados que se vayan obteniendo durante la ejecución del Algoritmo Genético, se podrá explorar,

en mayor o menor medida, el espacio de búsqueda representado por un determinado gen.

Para una generación individual, el funcionamiento del sistema propuesto es idéntico al comentado en el apartado anterior, es decir, un algoritmo que además de incluir una población de potenciales soluciones tiene una piscina genética para preservar la heterogeneidad de la distribución de los individuos a lo largo del espacio de búsqueda. La diferencia radica en que al final de cada generación, se procede a la agrupación y/o división de genes si se han cumplido unas condiciones establecidas a priori.

5.2.2.2 Genes Activos y no Activos

Tal y como se ha comentado anteriormente, cada vez que durante el proceso evolutivo se encuentre una solución con una bondad superior a la existente en la generación actual, se busca el representante de la piscina genética que simbolice dicha solución. Una vez encontrado dicho representante, se modifica el valor de sus genes (allí dónde no coincidan con los de la mejor solución) y se activan los valores B_{ij} con el objetivo de no modificarlos durante posteriores operaciones de mutación. Dichos genes con los valores B_{ij} activados se denominarán genes *Activos*. Al contrario, aquellos genes que transcurrida una generación no pertenezcan a la mejor solución, se denominarán genes *No Activos*. Obviamente, un gen puede no activarse en una generación y sí hacerlo en la siguiente o viceversa.

Para llevar control acerca del estado de cada uno de los genes, se añaden dos nuevos contadores a cada gen de cada individuo de la piscina genética: uno que indica el número de veces que dicho gen ha pertenecido a la mejor solución, A_{ij} , y otro que indica el número de generaciones

consecutivas que lleva sin formar parte de la mejor solución, D_{ij} , es decir, el número de generaciones que lleva sin activarse.

En la Figura 5.7 puede observarse cómo es la estructura de un individuo de la piscina genética. Es similar a la de los individuos de la piscina del sistema anterior, salvo que además incluye los dos contadores anteriormente mencionados.

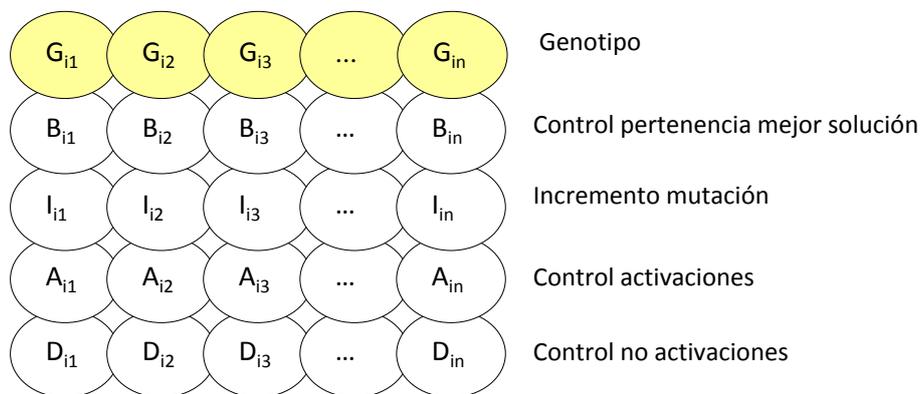


Figura 5.7: Estructura modificada de un individuo de la piscina genética

En la Figura 5.8 y la Figura 5.9 se muestra un ejemplo correspondiente a un problema de maximización. En la Figura 5.8 puede observarse el estado de la piscina genética, dentro de una determinada generación. En este caso, la piscina genética está formada por un total de tres individuos, para los que se muestran los rangos entre los que pueden variar los valores de cada uno de sus genes. Se muestran también los valores actuales de los genes, así como sus valores B_{ij} , sus contadores de activaciones y desactivaciones. También se muestran los valores de los genes que componen la mejor solución hallada hasta esa generación. Aquellos genes marcados en color verde se corresponden con los genes Activos (tienen los valores B_{ij} activados),

mientras que los demás son genes No Activos en esa generación (tienen los valores B_{ij} desactivados).

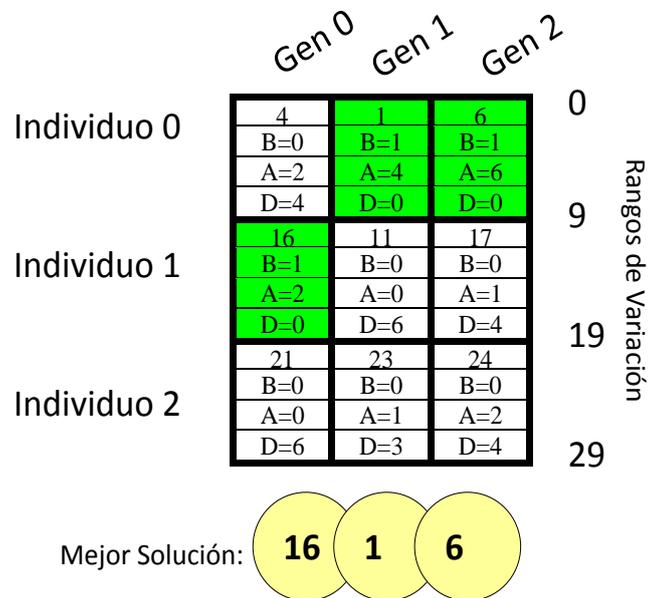


Figura 5.8: Piscina genética y mejor solución

Si en la siguiente generación, el descendiente obtenido tras una operación de cruce (o el nuevo individuo obtenido tras una mutación) tiene una aptitud mayor que la mejor solución actual, además de insertarlo en la población secundaria mediante el algoritmo de reemplazo adecuado, se deberán actualizar los valores de los individuos de la piscina genética y del mejor individuo.

Siguiendo el ejemplo anterior, en la Figura 5.9 se muestra cómo quedarían dichos elementos tras el paso de la generación.

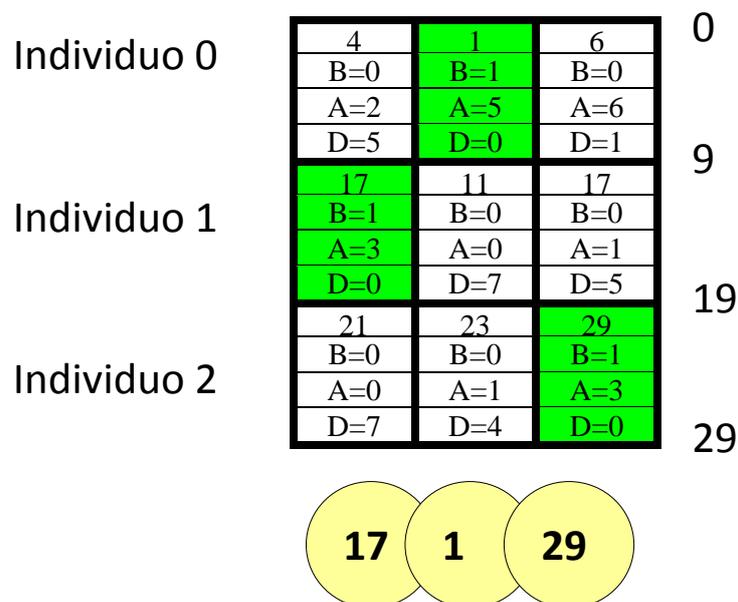


Figura 5.9: Modificaciones en la piscina genética tras mejorar solución actual

En la Figura 5.9 puede apreciarse cómo se modifican los contadores de cada gen dependiendo de si han sido o no activados: si forman parte de la mejor solución, el contador A_{ij} se incrementa y D_{ij} se inicializa a cero, puesto que este contador indica el número de generaciones consecutivas que permanece sin activarse; si no forma parte del mejor individuo, simplemente se incrementa el valor D_{ij} y el contador A_{ij} no se modifica. Puede observarse, por ejemplo, como los valores que abarca el Gen 1 del individuo número 1 de la piscina (rango [10..19]) llevan siete generaciones consecutivas sin activarse, es decir, sin formar parte de la mejor solución. Puede observarse también cómo cambian los controles de pertenencia a la mejor solución encontrada y, lógicamente, el valor numérico del gen.

Antes de pasar a la siguiente generación, se procederá a la agrupación y/o división de los genes Activos y/o No Activos, si se producen unas determinadas condiciones establecidas a priori. Al inicio de la ejecución

deberán establecerse una serie de parámetros, entre ellos, la decisión de qué hacer con los genes activos y los genes no activos. Podría optarse por dividirlos o agruparlos, o incluso, no realizar nada con ellos (ante esta situación, se estaría ejecutando el sistema del apartado anterior, con una distribución estática de las regiones de búsqueda). Para ello, se han de establecer los ratios de agrupación y/o división adecuados para ambos tipos de genes. Dichos ratios se basarán en los controles de activaciones y no activaciones que forman parte del genotipo de los individuos de la piscina genética (ver Figura 5.7), tal y como se ha mostrado anteriormente. Es decir, se han de especificar los valores que los controles de activaciones y no activaciones han de alcanzar para que se proceda a la agrupación o división del tipo de gen (activo/no activo) correspondiente.

Por ejemplo, si se establece un ratio de agrupación para los genes activos de 75 generaciones significa que si algún gen de la piscina genética se ha activado durante 75 generaciones (no tienen por qué ser consecutivas) se agrupará. Es decir, se producirá la operación de agrupamiento cuando el control de activaciones de un gen alcance el valor de 75 activaciones. Del mismo modo, si se especifica un ratio de división para los genes no activos de 50, se procederá a la división cuando algún gen lleve 50 generaciones consecutivas sin pertenecer a la mejor solución (es decir, cuando el control de no activaciones alcance el valor de 50).

A continuación se detallarán cada una de estas operaciones.

5.2.2.3 Agrupación de Subregiones de Búsqueda

La agrupación de subregiones se llevará a cabo siempre que existan individuos de la piscina genética con algún gen que alcance el ratio de

agrupación especificado previamente. Es decir, cuando se haya especificado una condición de agrupamiento y el controlador correspondiente (el de activaciones o no activaciones) haya alcanzado el valor indicado. Así pues, se podrán realizar múltiples agrupaciones en una generación.

Básicamente la agrupación de subregiones consiste en transformar los rangos de los valores permitidos para diferentes individuos de la piscina genética en un único rango que definirá los valores permitidos para la subregión agrupada (ver Figura 5.10).

Al agrupar diferentes subregiones, se disminuye la exploración en esa zona del espacio de búsqueda puesto que, al agrupar dos individuos, las posibilidades que tiene la subregión resultante de ser seleccionada para la reproducción son menores que la suma de las posibilidades que tenían ambas subregiones por separado. Visto de otra manera, se pasa de explorar una región por al menos dos individuos de la piscina genética (uno por cada subregión original) a hacerlo con un único individuo (el correspondiente a la subregión resultante tras la agrupación).

Por ejemplo, considérese el siguiente caso. Sea un gen que está representado en la piscina mediante cinco individuos. Por lo tanto será un gen sobre el que se definen cinco regiones de búsqueda. Todos los individuos tienen una probabilidad de ser seleccionados para formar parte del representante de la piscina genética de $1/5$ cada uno. Si se consideran por separado los genes de dos individuos, la zona especificada por ellos tendrá una probabilidad de $2/5$ de ser elegida. Si se agrupan los genes de los individuos, pasan a tenerse únicamente 4 subregiones (en este caso de diferente tamaño debido a la agrupación), cada una de ellas con probabilidad de $1/4$ de formar parte del representante de la piscina, y por tanto de pasar a

la población secundaria. Por lo tanto, la probabilidad de seleccionar un valor de la zona agrupada como parte de representante de la piscina genética es bastante inferior a la que existe previamente a la agrupación. Por consiguiente, la agrupación puede ser una buena opción para aquellas zonas que no han aportado buenas soluciones y en las que no se desea incidir, mientras que si se desea buscar con intensidad en una zona concreta es totalmente contraproducente.

Cuando algún gen alcanza el ratio de agrupación, se busca a aquel individuo con un valor de dicho gen más cercano para unirlos. Si uno de los genes del Individuo X de la piscina genética alcanza el ratio de agrupación, en función de su valor podrán darse dos situaciones:

- Si el valor de dicho gen está más cercano al límite superior de los valores que puede tomar el gen, la agrupación se llevará a cabo con el gen del individuo $X+1$
- Si por el contrario, el valor del gen es más cercano al límite inferior, se agrupará con el gen del individuo $X-1$

Una vez que se establece entre qué genes de la piscina genética se realiza la agrupación, se produce la unión de los rangos permitidos para cada uno de los genes de manera individual. Se obtiene de esta manera un *nuevo* gen en la piscina genética. El valor de este nuevo gen será el valor del gen que haya alcanzado el ratio de agrupación, mientras que su valor B_{ij} estará desactivado. Asimismo, los contadores A_{ij} y D_{ij} , que indican el número de generaciones que se activa un determinado gen de un individuo y las generaciones consecutivas que no se ha activado respectivamente, se inicializan a cero.

En la Figura 5.10 se muestra una situación en la que se va a proceder a una agrupación en la piscina genética, habiéndose decidido agrupar los genes no activos con un ratio de 10 generaciones. Aparecen representados en azul aquellos genes que han alcanzado el ratio de agrupación y en amarillo aquellos genes con los que se realizará la agrupación.

	Gen 0	Gen 1	Gen 2	
Individuo 0	4	8	6	0
	B=0	B=0	B=1	
	A=2	A=5	A=6	
Individuo 1	D=5	D=10	D=0	9
	10	11	12	
	B=1	B=0	B=0	
Individuo 2	A=3	A=5	A=1	19
	D=0	D=7	D=10	
	21	23	28	
Individuo 3	B=0	B=0	B=0	29
	A=0	A=1	A=0	
	D=10	D=4	D=10	
Individuo 3	35	31	36	39
	B=0	B=1	B=0	
	A=2	A=5	A=6	
	D=5	D=0	D=1	

Figura 5.10: Agrupación de subregiones: Paso 0

En la Figura 5.11, se muestra cómo quedaría la piscina genética después de realizar una agrupación. Se observa que los genes no están representados en la piscina por el mismo número de individuos: mientras que el Gen 0 está representado por tres individuos genéticos al igual que el Gen 1, el Gen 2 sólo lo está por dos individuos en la piscina. Es decir, el rango total del Gen 2 sólo se reparte entre dos individuos. Asimismo, puede apreciarse cómo cambian los contadores A_{ij} , D_{ij} y los valores B_{ij} de pertenencia al mejor individuo de la población para el nuevo individuo que se ha generado.

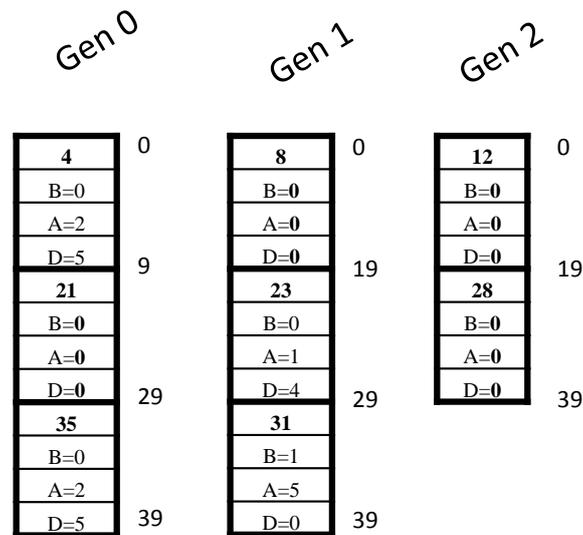


Figura 5.11: Agrupación de subregiones: Paso 1

5.2.2.4 División de Subregiones de Búsqueda

Al contrario de lo que sucede con la agrupación, al dividir un determinado gen o subregión, se otorgan más oportunidades de ser elegida a cada una de las partes por separado que estando juntas. Por lo tanto, se hará una exploración mayor del espacio de búsqueda. Esto en ocasiones puede interesar si se desea buscar exhaustivamente en alguna subregión determinada, o por el contrario, puede que no sea conveniente si no se desea buscar más soluciones en alguna zona.

Cuando algún gen en piscina genética alcanza el ratio de división, se procede a la división de dicho gen en dos regiones de búsqueda. Es decir, cuando se haya especificado una condición de división y el controlador correspondiente (el de activaciones o no activaciones) haya alcanzado el valor indicado. El rango de posibles valores para el gen que ha alcanzado el ratio de división, se reparte entre las dos nuevas regiones. El valor del gen que ha

provocado la división se establecerá como valor de la región a cuyo rango le corresponda, mientras que el valor de la segunda región se inicializará aleatoriamente dentro de los valores que delimiten su rango. Los controles de pertenencia, B_{ij} , se inicializan desactivados. Al igual que sucedía con la agrupación, los contadores A_{ij} y D_{ij} se resetean.

	Gen 0	Gen 1	Gen 2
Individuo 0	4	1	6
	B=0	B=1	B=0
	A=2	A=4	A=3
	D=5	D=0	D=1
Individuo 1	17	11	17
	B=1	B=0	B=0
	A=5	A=0	A=1
	D=0	D=7	D=5
Individuo 2	21	23	29
	B=0	B=0	B=1
	A=0	A=1	A=5
	D=7	D=4	D=0

Figura 5.12: División de subregiones: Paso 0

En la Figura 5.12 y Figura 5.13 se muestra un ejemplo de división, en este caso de los genes activos, tomando un valor de 5 como ratio de división. En la piscina genética se encuentran 2 genes que alcanzan este ratio de división (señalado en negrita), por lo tanto serán los genes que se dividirán.

En la Figura 5.13 se muestra el resultado tras la división. Al igual que sucedía con la agrupación, se puede observar cómo los genes no están todos representados por el mismo número de individuos en la piscina genética.

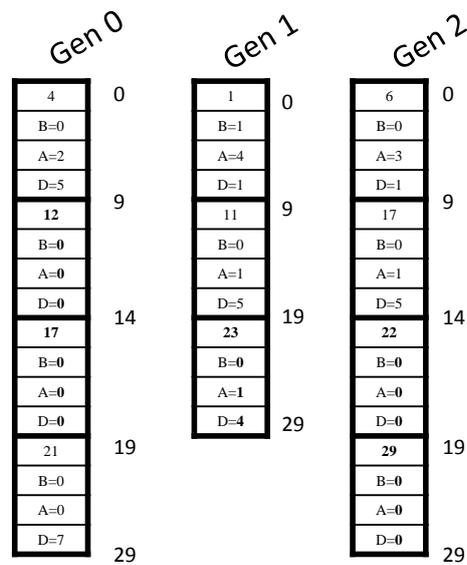


Figura 5.13: División de subregiones: Paso 1

5.2.2.5 Estabilidad Subregiones

Siguiendo el esquema de funcionamiento anteriormente propuesto se incurre en una potencial situación peligrosa. Por una parte, si la solución global está muy definida dentro del espacio puede ocurrir que ésta se localice en una fase temprana del proceso de búsqueda. En el caso de estar activa la división para los genes activos, la región que contenga los genes pertenecientes a dicha solución se vería involucrada en un proceso continuo de divisiones. Esta situación, lejos de mejorar el rendimiento del proceso de búsqueda, desembocará en focalizar la generación de nuevos individuos en el entorno de la solución global.

Siguiendo con el escenario anterior relativo a la localización de la solución global en fases tempranas del proceso de búsqueda, pero considerando en este caso una agrupación de genes no activos, este contexto también desemboca en una situación no deseable. Bajo estas circunstancias las diferentes subregiones se irían agrupando, puesto que el mejor individuo

pertenecería siempre a la misma región. Esta agrupación progresiva podría llegar a originar una piscina genética únicamente con dos regiones, una de ellas la correspondiente al mejor individuo y la otra correspondiente al restante espacio de búsqueda.

Para evitar estas situaciones bastará con definir un límite superior al número de divisiones que se pueden realizar y un límite inferior al número de subregiones resultante tras la aplicación de las operaciones de agrupamiento. Dichos límites deberían estar marcados por el tamaño o número inicial de subregiones. Una posibilidad sería restringir el número máximo de las subregiones creadas como consecuencia de un proceso de división a un 50% del número de subregiones iniciales; en cuanto a las agrupaciones, una opción sería limitar el número mínimo de subregiones de la piscina genética a un 50% del número inicial de las mismas.

5.2.3 Agrupación en Especies con Regiones de Rechazo

Tal y como se ha comentado, los Algoritmos Genéticos tradicionales tienden a proporcionar una única solución para los problemas que tratan, ocasionando un gran inconveniente para la resolución de problemas multimodales, ya que interesaría conocer el mayor número posible de soluciones.

A continuación se describe un método que trata de maximizar el número de soluciones encontradas, imitando para ello el funcionamiento de la Naturaleza, en concreto la forma en la que los individuos tienden a agruparse en especies. Se modificará la estructura de un Algoritmo Genético clásico para incorporar este comportamiento con el objetivo de alcanzar la máxima cantidad de soluciones globales y parciales dentro del espacio de búsqueda.

5.2.3.1 Exploración del espacio de búsqueda mediante especies

La Naturaleza lleva millones de años utilizando los mismos métodos para alcanzar su objetivo: el mantenimiento de la vida. Este objetivo se logra, con resultados plenamente satisfactorios, principalmente mediante la evolución continua de especies.

Esta evolución de las especies se origina debido a que los individuos de una población deben competir constantemente con otros por recursos tales como comida, agua o refugio. Los individuos que tienen más éxito en la lucha por los recursos tienen mayores probabilidades de sobrevivir y generalmente una descendencia mayor. Al contrario, los individuos peor adaptados tienen un menor número de descendientes, o incluso ninguno. Esto implica que los genes de los individuos mejor adaptados se propagarán a un número cada vez mayor de individuos de las sucesivas generaciones. La combinación de características buenas de diferentes ancestros puede originar en ocasiones que la descendencia esté incluso mejor adaptada al medio que los propios progenitores. De esta manera, las especies evolucionan adaptándose más y más al medio a medida que transcurren las generaciones.

En la Naturaleza puede observarse como los individuos se reparten en poblaciones, buscando un mayor número de oportunidades de supervivencia. Esto es, no conviven en un caos repartiéndose aleatoriamente por el espacio disponible. Así, en cada población, siempre habrá individuos mejor y peor adaptados al medio, lo que redundará en que los primeros tendrán muchas más opciones de reproducirse, y los segundos pocas. Esto implicará que, dentro de cada especie, los individuos lucharán por alcanzar la mejor posición, lo que trasladado a términos de Algoritmos Genéticos corresponderá a una potencial solución.

Pero no sólo se producirá una competición por alcanzar la mejor posición dentro de cada especie, sino que también las especies lucharán entre sí por alcanzar la mejor situación, defendiéndola una vez lograda. Así, en ocasiones las poblaciones podrán desaparecer, si sus individuos están muy mal adaptados al medio donde viven. Del mismo modo, otras nuevas especies podrán crearse, gracias a individuos que han salido de sus poblaciones originales y se reproducen en otro sitio, generando una nueva población.

Por lo tanto, si este esquema de funcionamiento ofrece resultados satisfactorios en entornos naturales, ¿por qué no ha de funcionar también a la hora de determinar el comportamiento de un sistema de resolución de problemas? Bajo esta premisa se desarrolla el sistema descrito en los siguientes apartados.

Para trasladar el esquema anteriormente referido a un sistema de computación evolutiva, debe comenzarse por repartir a todas las posibles soluciones del problema (los individuos) en varias poblaciones (o especies) iniciales de forma aleatoria. A partir de ahí, y generación tras generación, la aplicación de los operadores genéticos de selección, cruce y mutación irán modificando el aspecto de esas poblaciones (y de otras, si llegan a crear alguna nueva) de tal manera que cada una de ellas se centrará en una región concreta del espacio de búsqueda (una solución global, o parcial) que *defenderá* ante el resto de poblaciones. Así, estas poblaciones se verán obligadas a migrar hacia otras regiones del espacio de búsqueda todavía no explotadas.

Siguiendo este esquema de funcionamiento, se garantiza el encontrar no una única solución, sino un conjunto de ellas.

5.2.3.2 Regiones de rechazo

Continuando con el paralelismo con la Naturaleza, se puede observar cómo los individuos suelen desarrollar toda su actividad en la población en la que han nacido. Se cruzan con otros individuos vecinos, mutan,... pero todo ello en la misma población a la que pertenecen desde su nacimiento. Sólo en algunas ocasiones un individuo sale de una población para cruzarse con otro de alguna población vecina.

Para imitar este comportamiento, se propone la utilización de un mecanismo de protección para que las poblaciones no se vean afectadas por individuos ajenos: las regiones de rechazo. Estas regiones de rechazo actuarán a modo de *barreras* que rodean a cada población, impidiendo que nuevos individuos generados a partir de un cruce, se sitúen en poblaciones ya existentes, y por lo tanto exploren regiones del espacio de búsqueda que ya estaban siendo examinadas.

Estas regiones de rechazo incluirán a todos los individuos de una especie o población. Asimismo presentarán la suficiente *holgura* para que los descendientes de cada especie puedan ser insertados en la misma, sin que todos ellos estén demasiado próximos entre sí. Es decir, permitirán que la heterogeneidad dentro de cada especie se mantenga, con el fin de evitar la homogenización de la población genética.

Cada región de rechazo tendrá un radio determinado, tomándose como centro el centroide de la población. Este centroide, tal y como puede desprenderse de su nombre, será el individuo de la especie que esté más próximo a todo el resto de individuos.

Tras la generación de nueva descendencia, si ésta se origina en el interior de la región de rechazo de la propia especie, pasará a formar parte de dicha especie (eventualmente reemplazando a alguno de los individuos previos). Por el contrario, si se genera en el exterior de la región de rechazo de la propia especie, existen dos opciones: que se cree dentro de la región de rechazo de otra especie o bien que se cree en una región del espacio de búsqueda todavía no explorada. En el primer caso, la descendencia será descartada automáticamente, puesto que esa zona del espacio de búsqueda ya estaba siendo examinada. En el segundo caso, la descendencia podrá ser el germen de una nueva especie que busque nuevas soluciones en una región hasta el momento inexplorada.

De esta manera se consigue que las poblaciones se repartan a lo largo del espacio de poblaciones y ocupen la mayor parte de soluciones posibles. Evidentemente, si el número de soluciones encontradas aumenta, también lo harán las probabilidades de generar especies que se acerquen más a la solución óptima del problema.

5.2.3.3 Algoritmo Genético basado en especies con regiones de rechazo

A continuación se detallará la propuesta del sistema basada en las consideraciones previas, centrandó la explicación sobre todo en las características añadidas que diferencian a este algoritmo de los tradicionales y que, entre otras, permiten la búsqueda simultánea de máximos y mínimos locales y globales (a diferencia de los Algoritmos Genéticos tradicionales, que sólo realizan procesos de maximización o de minimización, no ambos de manera conjunta).

Una vez iniciado el proceso de búsqueda, los mejores individuos de cada especie proporcionarán cada una de las soluciones (globales o locales) mantenidas en el sistema.

5.2.3.3.1 Creación de especies

Para que un Algoritmo Genético pueda trabajar con individuos pertenecientes a poblaciones o especies diferentes, es obvio que tanto los individuos como las especies deben existir previamente.

Así, este proceso de inicialización es lo primero que debe acometerse. Para ello, se crea un número determinado especies, es decir, una serie de poblaciones iniciales de individuos. Dicho número deberá ser especificado previamente, en función de la naturaleza del problema a tratar, por ejemplo. Dentro de cada especie se generarán una serie de individuos cercanos entre sí (tal y como sucede en la Naturaleza en dónde dentro de una población los individuos están siempre relativamente próximos). Para ello se parte de un individuo *semilla* y se aplican una secuencia de pequeñas mutaciones o variaciones en sus genes de tal forma que los individuos generados pertenezcan al mismo entorno que el individuo original.

Para asegurar la corrección de dicha inicialización, a continuación deberá verificarse si los todos los individuos son válidos. Esto es necesario, puesto que, si la *semilla* original está muy cercana a otra especie, podría ocurrir que los individuos generados estuviesen incluidos dentro de la región de rechazo de una especie diferente a la que realmente deberían pertenecer. Si eso sucede, se elimina la especie creada, regenerándose nuevamente de manera aleatoria.

5.2.3.3.2 Obtención de los centroides

Una vez creadas las poblaciones el siguiente paso consistirá en determinar su centroide, es decir, el individuo que, tal y como indica su nombre, está situado en el centro de la población. Por lo tanto, éste será el individuo que más próximo esté al resto de sus vecinos (siendo el que minimice el sumatorio de distancias entre él y el resto de individuos). En la Figura 5.14, se muestra un ejemplo gráfico, resaltándose en negrita el centroide de la población.

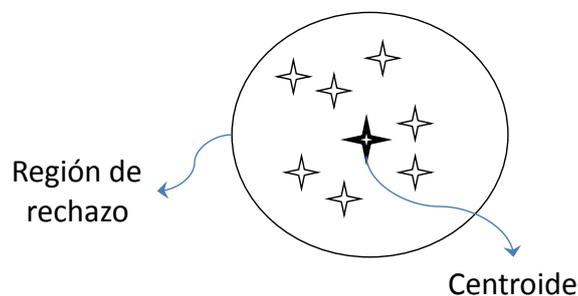


Figura 5.14: Centroide de una especie

Este individuo será básico en el funcionamiento del algoritmo ya que en base a él se definirá la región de rechazo de la especie a la que pertenece. Asimismo será crucial para determinar si la citada especie será encargada de localizar un mínimo o un máximo en el espacio de búsqueda, tal y como se mostrará más adelante.

Para determinar el centroide de una especie, se obtiene la distancia euclídea del fenotipo de cada individuo con el de todos sus vecinos y se suman las distancias con cada uno del resto de individuos (por ejemplo, en el caso del individuo 1, se suman sus distancias con el individuo 2, 3, 4, etc.). El individuo que presente un menor valor en dicho sumatorio de distancias, será elegido como centroide de la población.

Evidentemente, este proceso deberá repetirse cada vez que se produzca alguna alteración en la especie, bien sea por la inserción de la

descendencia generada por una operación de cruce, por la aplicación del operador de mutación, etc. Esto es debido a que estas operaciones han podido generar cambios en las especies que afecten lo suficiente como para que un individuo deje de ser centroide en un momento dado; o al revés, que un individuo pase a ser el centroide de la especie al variar su situación o la de sus vecinos en la especie.

5.2.3.3.3 Creación de las regiones de rechazo

Tal y como se ha comentado anteriormente, una región de rechazo es una barrera que se crea alrededor de cada especie, con un radio variable, que siempre rodeará a todos los individuos de la población. Estas regiones sirven para evitar que nuevos individuos generados en otras poblaciones se introduzcan en una especie ya existente, a imitación de lo que sucede en la Naturaleza (una cría no se traslada a otra población al nacer, se mantiene al lado de sus padres), y así modificar la tendencia de la población.

Para crear estas regiones, se halla la distancia del centroide al individuo más alejado, lo que determinará la distancia mínima de la región. De esta manera quedarán incluidos dentro de la región de rechazo todos los individuos de la especie. Además, adicionalmente se añade un porcentaje al radio de la región de rechazo, para hacerla más grande y permitir la inserción de individuos tras las operaciones de cruce y mutación, así como evitar la generación de nuevas especies en sus inmediaciones. Este porcentaje añadido será dependiente del número de individuos de la especie, que tal y como se verá más adelante variará a lo largo de las generaciones. De esta manera se imita el comportamiento natural, puesto que las especies con mayor cantidad de individuos será capaces de *defender* una mayor cantidad de terreno. En la Figura 5.15, puede verse este aspecto gráficamente. En dicha figura, se

representa con una línea punteada la región que engloba a todos los individuos de la especie, mientras que el espacio adicional entre dicha línea y la línea continua externa vendrá determinado por la cantidad de individuos de dicha especie.

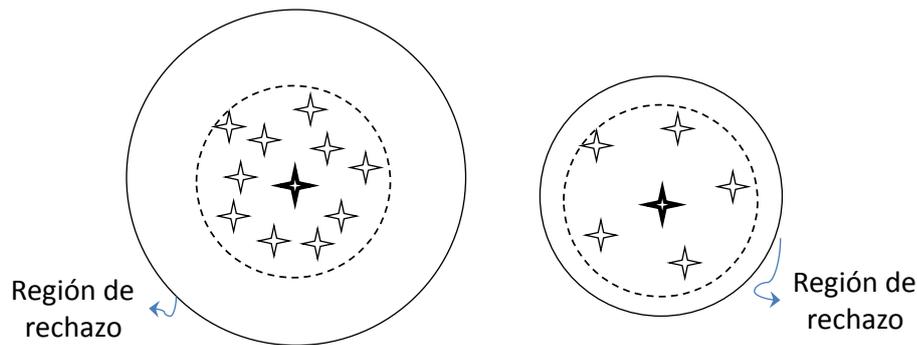


Figura 5.15: Tamaño de la región de rechazo

5.2.3.3.4 Maximización vs. Minimización

En un espacio de búsqueda no se encontrarán únicamente máximos o mínimos, sino una combinación de ambos.

Por lo general los métodos de búsqueda suelen centrarse en localizar exclusivamente un tipo de soluciones basándose en procesos de maximización o minimización. Sin embargo, podría ser interesante, a medida que se explore el espacio de búsqueda, proporcionar de manera simultánea tanto unos como otros. Ahora bien, la función de evaluación de los métodos evolutivos es ligeramente distinta si se está ante un problema de maximización o minimización, por lo que será necesario conocer en cada momento para cada especie qué tipo de búsqueda está realizando.

Sin embargo los propios individuos de una población aportan información suficiente para realizar esta distinción. Para ello podrá hacerse uso del centroide de cada una de las especies. Así, para determinar si una

especie está más cercana a un mínimo o a un máximo, puede calcularse la distancia entre el centroide y los individuos que presenten el mayor y el menor fitness dentro de dicha especie. Cuando una especie busca un máximo todos los individuos de dicha especie tienden a aproximarse paulatinamente a dicho máximo. Por lo tanto, si el centroide está más cercano al individuo que presenta el mayor fitness puro de la especie, podrá suponerse la población que forma la especie está buscando un máximo; en caso contrario, si el centroide está más cercano al individuo con el menor fitness puro de la población está buscando un mínimo. Es decir, comparando el centroide con el mejor y el peor individuo (en términos de valor de ajuste) se puede determinar la tendencia que presenta cada especie en cada momento.

En la Figura 5.16, puede observarse gráficamente esta cuestión. En este caso concreto, puede considerarse que la especie está escrutando el espacio de soluciones en busca de máximo, puesto que el centroide (representado por un punto negro) está más cercano al individuo de mayor fitness (punto rojo) que al individuo de menor fitness (punto gris).

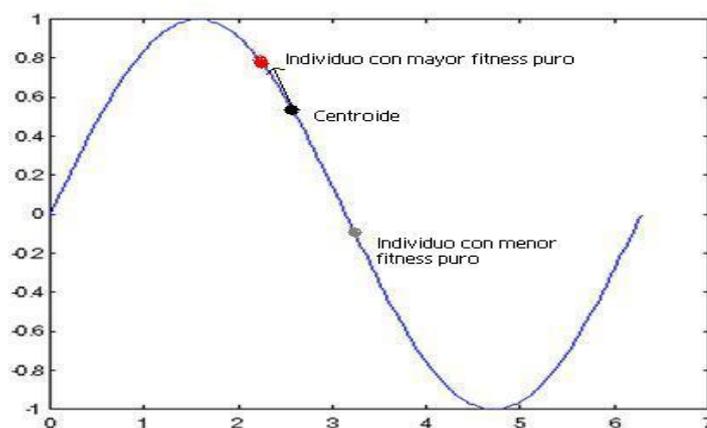


Figura 5.16: Maximización vs. Minimización

Una vez se establezca lo anterior, podrá obtenerse el valor de ajuste de cada individuo, bien sea en términos de fitness ajustado o normalizado. Será preferible el empleo de este último, puesto que es el único tipo de ajuste que tiene en cuenta al resto de individuos de la población, valorando así con mejor criterio al individuo.

Evidentemente, los mejores y peores individuos de cada especie van variando a lo largo de la evolución del algoritmo, puesto que las poblaciones se van moviendo por el espacio de soluciones, acercándose y alejándose de los mínimos y máximos. Adicionalmente, el operador de reemplazo podrá eliminar a alguno de ellos en alguna de las generaciones. Por lo tanto será necesario tener en cuenta este aspecto.

5.2.3.3.5 Aplicación de los operadores genéticos

Una vez creadas cada una de las especies, establecidas sus regiones de rechazo y definido el tipo de objetivo (maximización o minimización) que busca cada una de ellas, es el momento de aplicar los operadores genéticos.

El esquema de aplicación, que se sigue dentro de cada especie de manera independiente, es idéntico al seguido en un Algoritmo Genético clásico. Es decir, en primer lugar se seleccionan los individuos mediante alguno de los algoritmos de selección existentes, a continuación se aplica un operador de cruce con una probabilidad P_c determinada y finalmente parte de la descendencia se muta en base a una tasa P_m .

5.2.3.3.6 Ubicación de la descendencia

Una vez generada la descendencia, es el momento de incorporarla a la población genética. En función del lugar en el que se genere la descendencia,

dentro de la propia especie, dentro de una especie ya existente o bien en una región del espacio de búsqueda todavía no explorada, se procederá de uno u otro modo.

En primer lugar puede ocurrir que la descendencia se genere dentro del espacio delimitado por la región de rechazo de su propia especie. En este caso el nuevo individuo se añadirá a dicha población. Tal y como se verá en el siguiente punto, el tamaño de la poblaciones no permanece constante a lo largo de las evoluciones. Por lo tanto, si en el momento de la inserción del nuevo individuo, la población no ha alcanzado su límite máximo de individuos, éste se incorporará directamente a la población. En caso contrario, pasará a formar parte de la población tras aplicar alguno de los algoritmos de reemplazo mencionados en la sección de Fundamentos.

Si la descendencia se genera dentro de la región de rechazo de otra especie, simplemente se descarta. De esta manera se evita explorar una zona del espacio de búsqueda que ya estaba siendo explorada.

Una última opción es que la nueva descendencia ocupe una zona que en la generación actual no esté siendo explorada por ninguna especie. En este caso, a partir de esta descendencia se generará una nueva especie. Puesto que para que una especie sea viable es necesario que puedan realizarse las operaciones de cruce y mutación, no será suficiente con crear una nueva especie a partir únicamente de un individuo. Por lo tanto, deberán generarse una serie de nuevos individuos. Así, el individuo a partir del que se crea esta nueva población actuará de semilla. Modificándose ligeramente los valores de sus genes de manera aleatoria se conseguirán los nuevos individuos que pasarán a formar parte de dicha población.

Una vez creada la nueva especie, deberán obtenerse el centroide, el mejor y peor individuo y la región de rechazo, para que dicha especie pueda incorporarse al conjunto global de poblaciones del sistema.

5.2.3.3.7 Eliminación de especies

Para evitar el crecimiento exponencial del número de especies sobre el espacio de búsqueda, se hace necesario eliminar alguna de ellas a medida que avanzan las generaciones.

Para ello vuelve a hacerse uso del centroide de cada especie. Así, para cada especie se calcula la distancia del centroide al mejor individuo de cada especie. Cuando esta distancia es máxima puede inferirse que los individuos de la citada población no están en el entorno de una solución (global o local), sino que se encuentran separados entre sí y dispersos en una determinada zona del espacio de búsqueda. Por el contrario, cuando una especie se acerca a una solución, la distancia entre el centroide y el mejor individuo tiende a reducirse, puesto que todos los individuos tienden a dicha solución.

Así, la población que presente una mayor distancia entre el centroide y el mejor individuo se marca como peor especie de la presente generación. Cuando una especie sea definida como la peor durante una serie de generaciones predefinidas (dependerá del problema) podrá ser eliminada, puesto que puede considerarse que no aporta – ni posiblemente tienda a aportar, puesto que han transcurrido un cierto número de generaciones en la misma situación – ninguna solución válida al problema.

5.2.3.3.8 Modificación del tamaño de las especies

Tal y como se ha comentado anteriormente, el número de individuos que una especie puede albergar no se mantiene estático a lo largo de todo el proceso de búsqueda, sino que puede variar. Por lo tanto puede considerarse la reducción o el aumento del número de individuos de cada especie al terminar cada una de las generaciones. Esta reducción o aumento será proporcional al número de individuos que en el momento de la modificación presente la especie.

El criterio que se sigue para determinar el tamaño de cada una de las especies es el siguiente:

1. Si el mejor individuo de una especie presenta un error mayor que el que presentaba en la generación anterior, se reduce o se limita el tamaño de la especie. Esto impide que se sigan añadiendo más individuos a una población que se va alejando del objetivo. De esta forma trata de impedirse que la especie se aleje más del citado objetivo, y quizá pueda *enderezar* la búsqueda en posteriores generaciones.
2. Si el mejor individuo de una especie mejora su error con respecto a las generaciones anteriores, se permite ampliar el tamaño de la especie. Así, se permite que se puedan incorporar más soluciones a la población, puesto que se va acercando al objetivo, y así poder obtener mejores soluciones generación a generación, al disponerse de una mayor cantidad de información genética.

5.2.3.3.9 Esquema general de funcionamiento

A continuación se muestra el funcionamiento del Algoritmo Genético propuesto, en forma de diagrama general de funcionamiento (Figura 5.17) y de pseudocódigo detallado (Figura 5.18).

El funcionamiento general del sistema, con sus regiones de rechazo, se asemeja en cierta manera a la idea general de un sistema basado en *fitness sharing*, intentando evitar la masificación de individuos entorno a una solución. Sin embargo, aporta la ventaja de una gestión automática del número de especies en busca de una solución. Adicionalmente, se incluye la posibilidad de discriminar si una especie está en las cercanías de un mínimo o un máximo, con lo que puede centrarse la búsqueda de mínimos, máximos o incluso ambos de manera simultánea, según sea de interés en cada caso.

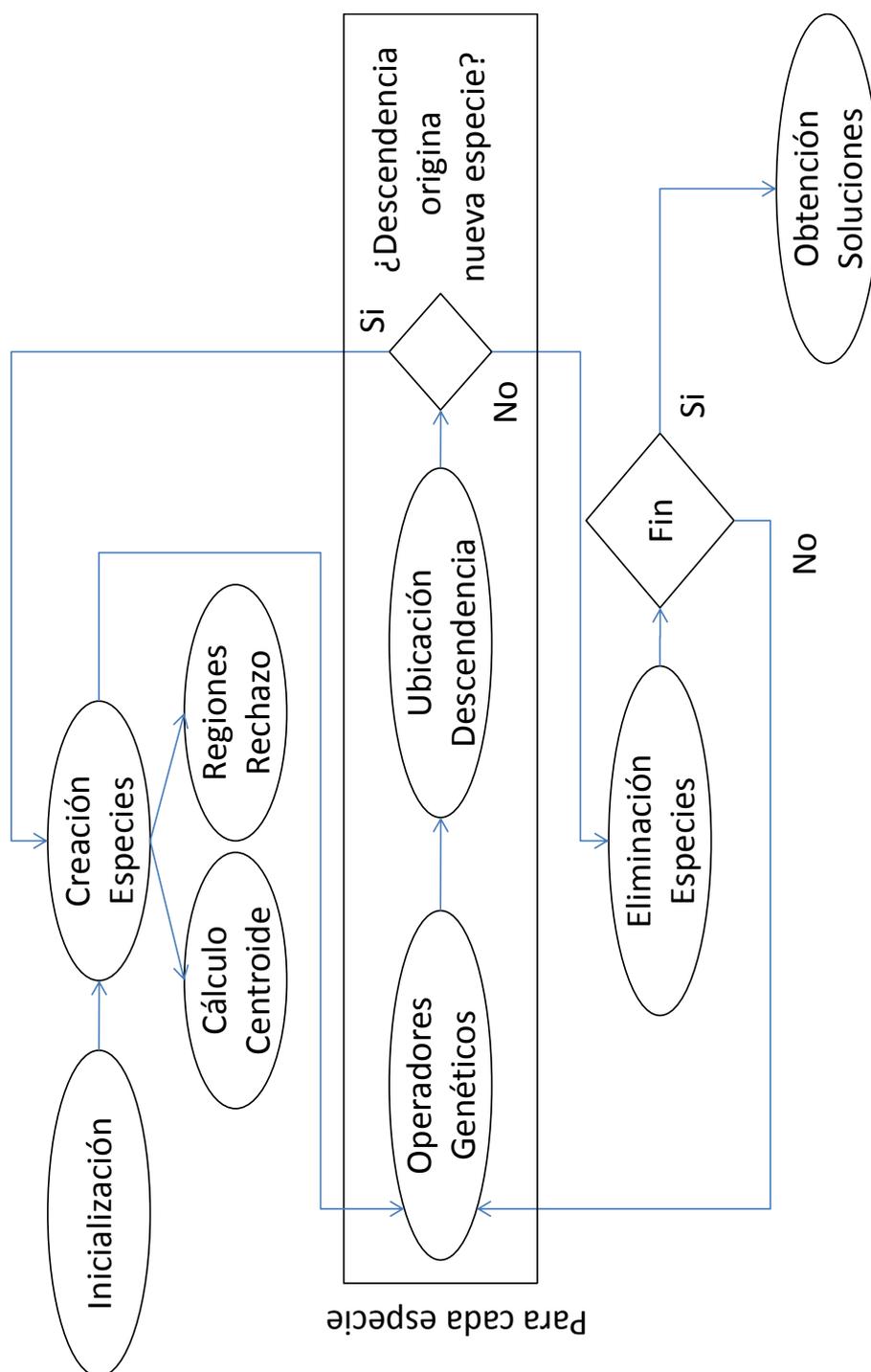


Figura 5.17: Diagrama general de funcionamiento AG basado en especies

MIENTRAS no se alcance el número de poblaciones deseadas

- Crear aleatoriamente una población
- Añadir la población a la lista de poblaciones
- Transformar fenotipo a genotipo
- Comprobación de que no interfiere con las regiones de rechazo existentes
 - SI la población no es válida se elimina
 - SI NO
 - Cálculo de centroides
 - Evaluación
 - Cálculo del mejor y peor individuo
 - Creación de la región rechazo

FIN MIENTRAS

MIENTRAS no se alcance la condición de terminación

- Selección
- Cruce
- Mutación
- Cálculo de fenotipos
- SI nuevos descendientes válidos
 - Se añaden a la población de los padres
- SI NO
 - Se crea una nueva población con cada nuevo individuo
 - Se crean nuevos individuos aleatorios para la nueva población
 - Se calculan los centroides de la nueva población
 - Evaluación de la nueva población
 - Cálculo del mejor y peor individuo de la nueva población
 - Creación de región rechazo de la nueva población
 - Actualización de contadores de la nueva población

- Evaluación
- Cálculo de los mejores y peores individuos
- Cálculo de peor población y eliminación si es el caso
- Cálculo de centroides
- Actualización del tamaño de las poblaciones
- Actualización de los contadores de las poblaciones

FIN MIENTRAS

Identificación óptimo global de cada población

Figura 5.18: Pseudocódigo detallado AG basado en especies

Capítulo 6

Resultados

6.1 Introducción

En el presente apartado se presenta una batería de pruebas realizada con el objetivo de verificar el correcto funcionamiento de la propuesta planteada en la presente Tesis.

En primer lugar, se comprobará la validez de dichas propuestas para la el mantenimiento de la heterogeneidad durante la búsqueda de soluciones en un espacio multimodal. Para ello se emplearán los problemas clásicos

expuestos previamente en la sección de fundamentos. Dichos problemas, además de la alta representatividad de lo que un espacio de búsqueda multimodal significa, permitirán una representación tridimensional de los datos, lo que facilitará significativamente su análisis.

Esta fase de las pruebas servirá también para ajustar tanto los valores óptimos de los parámetros en cada una de las propuestas, como las tasas de cruce y mutación, algoritmos de cruce, reemplazo, etc. Para minimizar la variabilidad de los resultados obtenidos, para cada tipo de prueba se repetirán 10 ejecuciones con los mismos parámetros, obteniéndose la media de los resultados (número de soluciones encontradas, error mejor individuo, etc.) cada 10 iteraciones, hasta llegar a las 100 finales.

El primer tipo de prueba consistirá en determinar la capacidad de fijar múltiples soluciones. Es decir, en verificar que la búsqueda no se focaliza en el entorno de la solución global que presenta cada uno de los ejemplos. Para ello se irá anotando, a medida que avancen las generaciones, el número total de soluciones localizadas. Se considerará que una solución se ha localizado cuando la diferencia del valor de ajuste de alguno de los individuos localizados en su entorno es menor al 1% del valor real de la solución.

Una vez realizadas estas pruebas, se procederá a aplicar las soluciones propuestas a un problema real, que incluye no sólo el aspecto de búsqueda multimodal, sino también de selección de variables. Concretamente se procederá a buscar las variables que aportan mayor información a la hora de clasificar una muestra de zumo en base a su contenido real en zumo.

6.2 *Funciones Clásicas*

6.2.1 *Configuraciones iniciales*

En las siguientes tablas se muestran los parámetros iniciales de cada uno de los algoritmos a la hora de comenzar la resolución de los problemas. Estos parámetros no variarán, evidentemente, entre las diferentes resoluciones de un mismo problema, pero tampoco entre los diferentes problemas. Se busca de esta manera minimizar el impacto que las citadas configuraciones puedan ocasionar en los resultados finales. Por esta misma razón, se intenta conseguir una configuración inicial lo más homogénea posible entre las diferentes técnicas de búsqueda.

En el caso del Algoritmo Genético clásico y del Algoritmo Genético basado en especies, el genotipo de los individuos es binario, representando las coordenadas (x,y) de la solución sobre el espacio de búsqueda. La obtención de los valores fenotípicos se obtiene considerando los bits del individuo genético como las representaciones binarias en formato signo-magnitud. El número de bits se establece de tal manera que el rango de los valores reales que sea posible generar permita cubrir la totalidad del espacio de búsqueda, tomándose 2 posiciones decimales en cada solución. Por ejemplo, supóngase un espacio de búsqueda restringido al rango $[-100, 100]$, con lo cual se necesitarán 7 bits ($2^6 < 100 < 2^7$) para representar la parte entera de las coordenadas y otros 7 para representar la parte decimal de las mismas. La codificación genotípica de un individuo que represente la solución dada por la coordenada $(5'7, 8'1)$, sería $(0|0000101|0000111|0|0001000|0000001)$, dónde:

- 0: bit de signo de la componente x. Como es positiva, es cero.
- 000 0101: representa la parte entera de la componente x (en este caso 5)

- 000 0111: representa la parte decimal de la componente x (en este caso 7)
- 0: bit de signo de la componente y. De nuevo, positiva y cero.
- 0001000: parte entera de la componente y (en este caso 8)
- 0000001: parte decimal de la componente y (en este caso 1)

En el caso del Algoritmo Genético de dos poblaciones se ha utilizado una codificación decimal para representar las coordenadas del punto seleccionado como solución por el sistema. Esta diferencia viene marcada por la necesidad de la piscina genética de determinar subregiones dentro del espacio de búsqueda de cada gen. Si los genes pueden tomar únicamente valores binarios, no será posible establecer dichas subregiones. Por el contrario, si un gen puede tomar valores pertenecientes a un rango continuo, dentro de dicho rango cada uno de los individuos de la piscina genética podrá representar un subconjunto de valores que originará una subregión en el espacio de búsqueda.

A modo de método de referencia se empleará un Algoritmo Genético clásico, configurado con los valores indicados en la Tabla I, y un Algoritmo Genético, con los mismos parámetros generales, al que se ha añadido la opción de *fitness scaling* con el objetivo de minimizar la homogeneización de la población genética. Se han realizado pruebas con diversos escalados, entre ellos, *fitness rank*, *fitness proportional* o *fitness top (quantity=0.4)* obteniéndose resultados muy similares con todos ellos.

Los parámetros de configuración del Algoritmo Genético de dos poblaciones, se recogen en la Tabla II. Por lo que respecta a la versión que no emplea la modificación dinámica del número de regiones, los parámetros son los mismos, salvo, claro está, los referentes precisamente a la gestión del tratamiento de las regiones.

Finalmente, la Tabla III recoge los parámetros con los que se ha configurado el Algoritmo Genético con Agrupación en Especies.

Se han escogido los presentes valores como óptimos tras la realización de diversas pruebas variando el valor de los mismos. Para mostrar la influencia de las diferentes configuraciones de parámetros, en la siguiente sección se muestra el comportamiento de cada una de las técnicas con las diversas configuraciones.

Tabla I: Configuración Algoritmo Genético clásico

- Selección: Ruleta + Aleatorio
- Tasa de Cruces: 90%
- Individuos: 200
- Reemplazo: Similares
- Tasa de Mutaciones: 1%
- Fitness Scaling: *rank*
- Cruce: Uniforme

Tabla II: Configuración Algoritmo Genético de dos poblaciones
(regiones variables)

- Selección: Ruleta + Aleatorio
- P_c : 95%
- **Ratio división activos: 10**
- Reemplazo: Peores (5 indiv.)
- P_m : 1%
- **Agrupar no activos: 75**
- Cruce: Uniforme
- **Indiv. Pob. Secundaria: 100**
- **Indiv. Piscina Genética: 40**

Tabla III: Configuración Algoritmo Genético con agrupación en especies

- Selección: Ruleta + Aleatorio
- P_c : 95%
- Amplitud región rechazo: 2%
- Reemplazo: Peores (5 indiv.)
- P_m : 1%
- Generación inicio creación subespecies: 30
- Cruce: Uniforme
- Poblaciones Iniciales: 30
- Máx. núm. individuos: 1000

La mayoría de los parámetros no han mostrado excesiva influencia en la calidad final de los resultados, aunque sí en el número de generaciones requerido para alcanzar dichos resultados. Tal y como es previsible, se ha observado una mayor influencia de los parámetros propios de cada uno de los métodos de búsqueda definidos: ratio de división/agrupación en el caso del Algoritmo Genético con 2 poblaciones, o tamaño de la región de rechazo en el caso del Algoritmo Genético basado en especies.

6.2.2 Localización de soluciones globales y parciales

Tal y como ya se ha comentado, la primera de las pruebas planteadas consiste en analizar el número de soluciones que, a medida que avanzan las generaciones, mantiene cada uno de los métodos expuestos. Así, las siguientes gráficas mostrarán el número de soluciones localizadas y mantenidas por la población genética en la generación especificada. Puesto que este tipo de técnicas estocásticas aportan aproximaciones a la solución a un problema (difícilmente, y más en el tipo de funciones empleadas en los siguientes ejemplos, se obtendrá una solución exacta), se contabilizará una solución como válida cuando en su entorno se sitúe algún individuo que presente una diferencia del valor de ajuste menor al 1% del valor real de dicha solución.

6.2.2.1 Función de Ackley

En la Figura 6.1 se muestra la evolución del número de soluciones que, a medida que avanzan las generaciones, se mantienen en la población genética.

Puede verse como el Algoritmo Genético clásico tiende a localizar una única solución (en este caso localiza la solución global al problema), presentando unos resultados netamente inferiores a las dos técnicas aquí

propuestas. Añadiendo la opción de *fitness scaling* se obtienen resultados ligeramente mejores durante las primeras generaciones, pero a medida que éstas avanzan la búsqueda tiende a focalizarse y el algoritmo proporciona una única solución. El comportamiento de las técnicas basadas en el Algoritmo Genético de 2 poblaciones y el basado en agrupación en especies se muestran netamente superiores.

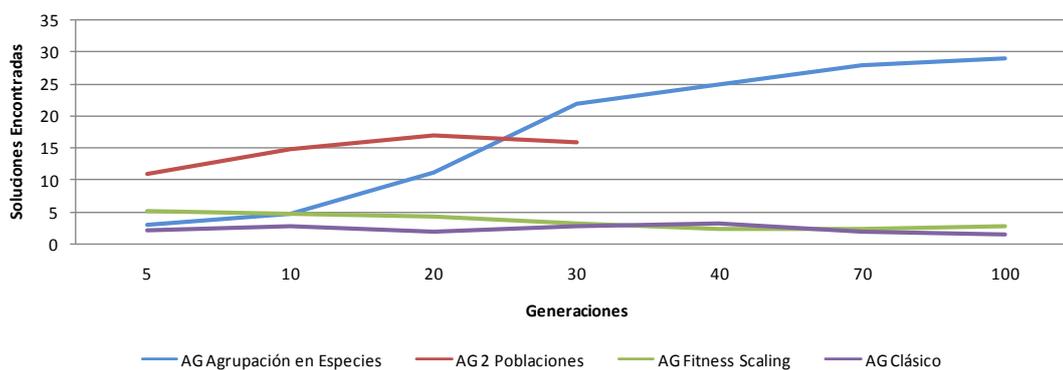


Figura 6.1: Ackley: Evolución del número de soluciones encontradas

Observando la Figura 6.1 resulta significativo el comportamiento del Algoritmo Genético de 2 poblaciones. Tal y como se aprecia en la gráfica, su ejecución no finaliza tras alcanzarse 100 iteraciones como el resto de algoritmos, sino tras llegar únicamente a las 30 generaciones. Como se verá posteriormente, este comportamiento se repetirá con el resto de ejemplos.

Esto ocurre así, puesto que es en este punto en dónde se alcanza el hecho de que, tras sucesivas divisiones de las regiones de búsquedas, el número de nuevas regiones creadas alcanza el 50% de las regiones originales. Esto ocasiona que, una vez alcanzada la solución global, la región en la que se sitúan los genes pertenecientes a dicha solución se ven involucrados en un continuo proceso de divisiones. Este hecho ocasiona que cada vez más individuos exploren la zona aledaña a la solución global, dejando cada vez

más despoblado el resto del espacio de búsqueda. Por lo tanto, lejos de mejorarse el número de soluciones encontradas, caso de continuarse con el proceso de búsqueda, el número de soluciones mantenidas por la población tiende a disminuir (Figura 6.2). Además, esta tendencia se hace más patente a medida que aumenta el número de generaciones. Por lo tanto, con el objetivo de mantener la estabilidad de las soluciones se detiene de manera automática la ejecución del algoritmo cuando se detecta la situación comentada.

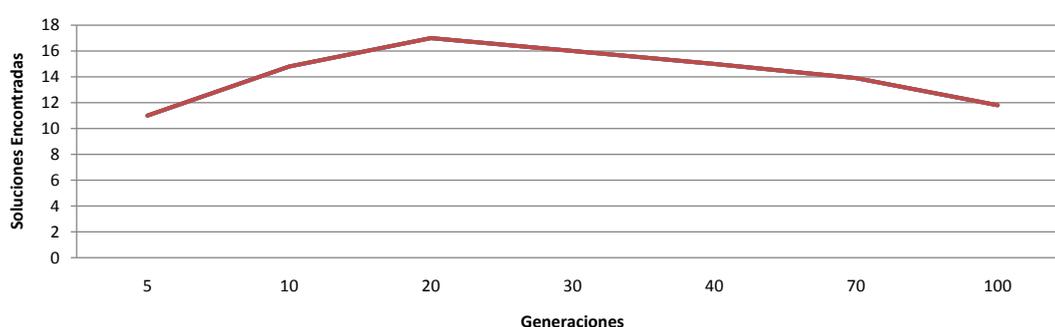


Figura 6.2: Algoritmo Genético de 2 Poblaciones: Número de soluciones encontradas sin el control de subregiones

En el caso de producirse una excesiva agrupación de subregiones, el resultado es prácticamente idéntico. Es decir, una vez alcanzado cierto punto de la ejecución, el número de soluciones mantenidas en la población se ve reducido. Esto es debido a que, mientras no se localiza la solución global, en este caso un mínimo, la región que contiene al mejor individuo va variando a medida que se localizan mínimos locales. Consecuentemente, a medida que se encuentran dichos mínimos se van actualizando los valores B_i para indicar las regiones que contienen los genes pertenecientes a la mejor solución. Sin embargo, una vez localizado el mínimo global, estos valores B_i ya no se modificarán. Esto ocasiona que el resto de regiones se vayan agrupando sucesivamente hasta formar una piscina genética con una única subregión, por lo cual su comportamiento tiende al de un Algoritmo Genético clásico.

En la Figura 6.3 se muestra la evolución de la distancia a la solución global que presenta el mejor individuo en cada una de las aproximaciones. Puede comprobarse como en todos los casos se alcanza el mínimo global, aunque la evolución del error en el caso del Algoritmo Genético con agrupación en especies sea menos pronunciada que en el resto de aproximaciones.

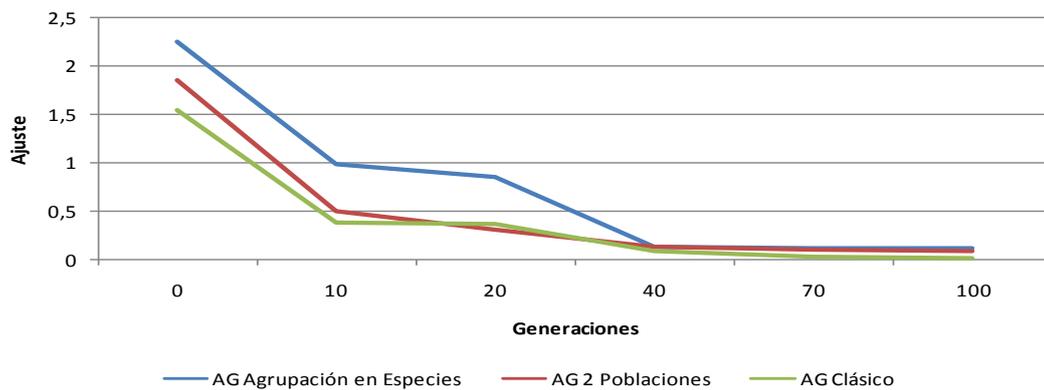


Figura 6.3: Ackley: Evolución del error

Con respecto al Algoritmo Genético de 2 poblaciones, esta gráfica permite corroborar la afirmación anterior relativa a la reducción del número de soluciones mantenidas en la población a medida que avanzan las generaciones. Puede verse como, a partir de la generación 20, justo en la que se produce el máximo número de soluciones localizadas, se produce también una reducción significativa en el error cometido por el mejor individuo. Sin embargo, a partir de la generación 30 (en la que bajo circunstancias normales se detendría la ejecución del algoritmo) los cambios en el error son mucho menos pronunciados, hecho debido, precisamente, a que la solución global al problema ha sido encontrada.

Teniendo en cuenta que, dentro del intervalo considerado para la realización de las pruebas, la función de Ackley cuenta con un total de 66

mínimos locales en la Figura 6.4 se representa el porcentaje total de estos mínimos localizados por cada una de las técnicas.

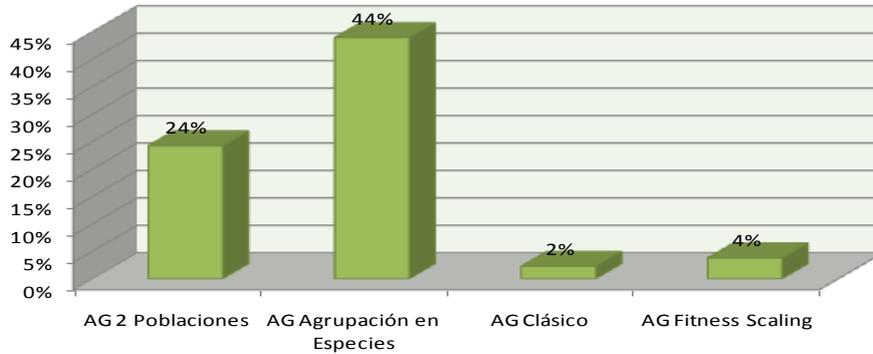


Figura 6.4: Ackley: Porcentaje de soluciones encontradas tras 100 generaciones

En el caso de emplear la técnica de *fitness scaling* el porcentaje total de soluciones encontrado es muy similar al mostrado para el Algoritmo Genético Clásico, puesto que al cabo de 100 generaciones, la búsqueda tiende a focalizarse en el entorno de la mejor solución. No obstante, sí se aprecia una ligera diferencia si, en lugar de considerar las soluciones mantenidas por la población al cabo de 100 generaciones, se considera el máximo número de soluciones mantenidas por la población a lo largo de la ejecución. En este caso, el porcentaje de soluciones encontradas por el Algoritmo Genético Clásico es de un 4%, mientras que asciende al 8% cuando se emplea el escalado.

Cabe destacar la gran diferencia en cuanto al porcentaje de soluciones encontradas de las dos técnicas propuestas con respecto al Algoritmo Genético Tradicional. En el caso concreto de la función de Ackley el porcentaje total de los mínimos locales no es elevado – en relación a los obtenidos en el resto de funciones, tal y como se mostrará a continuación – debido principalmente a dos circunstancias. Por un lado debido al elevado

número de mínimos locales en relación a la extensión del campo de búsqueda y por otro lado debido al tamaño de las poblaciones genéticas en todas las técnicas. No obstante, aunque estos resultados serían fácilmente mejorables ampliando la cantidad de individuos en las citadas poblaciones, se ha optado por mantener fijo el tamaño de las poblaciones con la intención de comprobar la adaptación de cada una de las técnicas – bajo unos mismos parámetros de funcionamiento – a los diferentes entornos proporcionados por las diferentes funciones de prueba y, al mismo tiempo, comparar el rendimiento de ambas técnicas bajo condiciones de carga computacional similares.

6.2.2.2 Función de Rastrigin

Tal y como puede verse en la Figura 6.5 y en la Figura 6.6, el comportamiento de todas las técnicas es similar en el caso de la función de Rastrigin a la hora de obtener una exploración sistemática del espacio de búsqueda.

En la Figura 6.5 puede observarse como, nuevamente, aunque en esta ocasión de una forma más notoria, el número de soluciones encontradas y mantenidas es superior en cualquiera de las técnicas propuestas en esta tesis que las obtenidas cuando se emplea un Algoritmo Genético clásico.

Nuevamente puede observarse en el comportamiento del Algoritmo Genético de 2 poblaciones, que una vez localizada la solución global, reduce paulatinamente el número de soluciones mantenidas en la población.

A partir de la Figura 6.5 puede constatarse como, hasta el momento en que comienza a producirse este descenso en la heterogeneidad de la piscina, la técnica de 2 poblaciones, obtiene resultados ligeramente superiores a los obtenidos con la técnica de agrupación en especies. Sin embargo, una vez

superado ese umbral la técnica de agrupación en especies supera el número de soluciones ofrecidas.

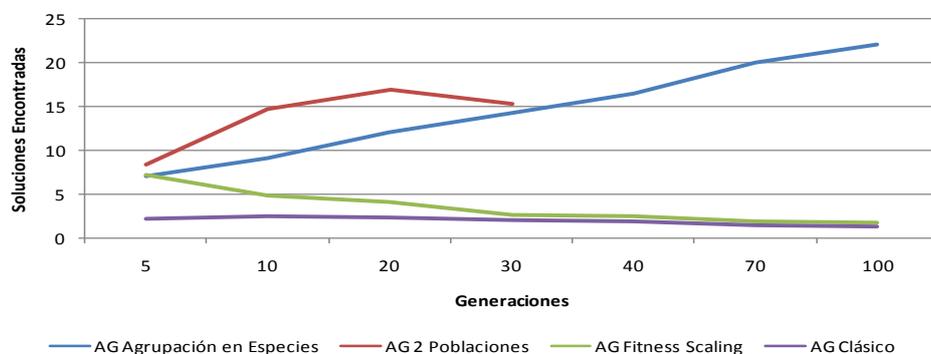


Figura 6.5: Rastrigin: Evolución del número de soluciones encontradas

En la Figura 6.6 se muestra la evolución del error a medida que avanzan las generaciones. En este caso, el Algoritmo Genético clásico (con o sin el empleo de *fitness scaling*) se comporta ligeramente mejor. La explicación reside en el hecho de que, desde las generaciones iniciales, ha centrado su proceso de búsqueda en una única región (la correspondiente a la solución global) descartando el resto del espacio de búsqueda, lo que ocasiona que el error pueda reducirse hasta valores más mínimos.

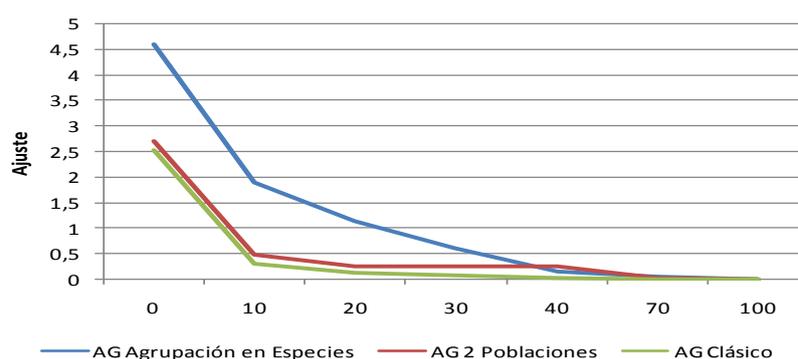


Figura 6.6: Rastrigin: Evolución del error

En la Figura 6.7 se representa el porcentaje de soluciones encontradas, teniendo en cuenta que, dentro del intervalo considerado, la función de Rastrigin posee un total de 23 mínimos locales.

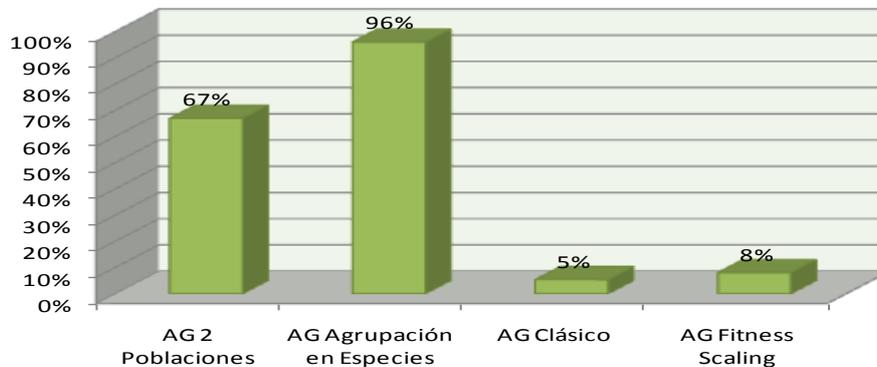


Figura 6.7: Rastrigin: *Porcentaje de soluciones encontradas tras 100 generaciones*

Nuevamente, si se considera para el Algoritmo Genético clásico el máximo número de soluciones encontradas a lo largo de la ejecución, el porcentaje sube al 11%, mientras que para la opción con el empleo de *fitness scaling* este porcentaje se eleva hasta el 31% de soluciones encontradas.

6.2.2.3 *Función de Schewefel*

En la Figura 6.8 y la Figura 6.9 se muestran las pruebas realizadas tomando como base la función de Schewefel. En ellas puede confirmarse la tendencia observada a raíz de la realización de las pruebas anteriores. Por una parte, el número de soluciones mantenidas por el Algoritmo Genético clásico es siempre menor al proporcionado bien por la técnica de 2 poblaciones, bien por la técnica de agrupación en especies. Por otro lado, el Algoritmo Genético clásico obtiene una evolución del error del mejor individuo ligeramente mejor que las dos técnicas restantes.

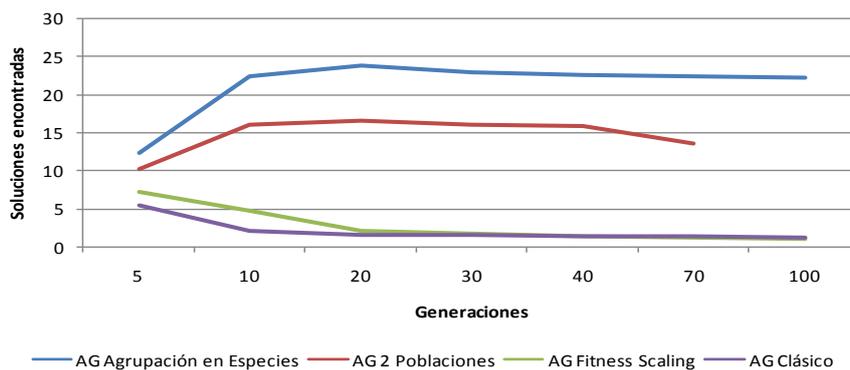


Figura 6.8: Schewefel: Evolución del número de soluciones encontradas

Al contrario de lo que ocurría con la función de Rastrigin y con la función de Ackley, en este caso las técnicas de 2 poblaciones y de agrupación en especies ofrecen resultados más similares, aunque la técnica de agrupación en especies continúa comportándose de manera más eficiente a la hora de escrutar el espacio de búsqueda en busca de múltiples soluciones, tal y como puede verse en la Figura 6.8.

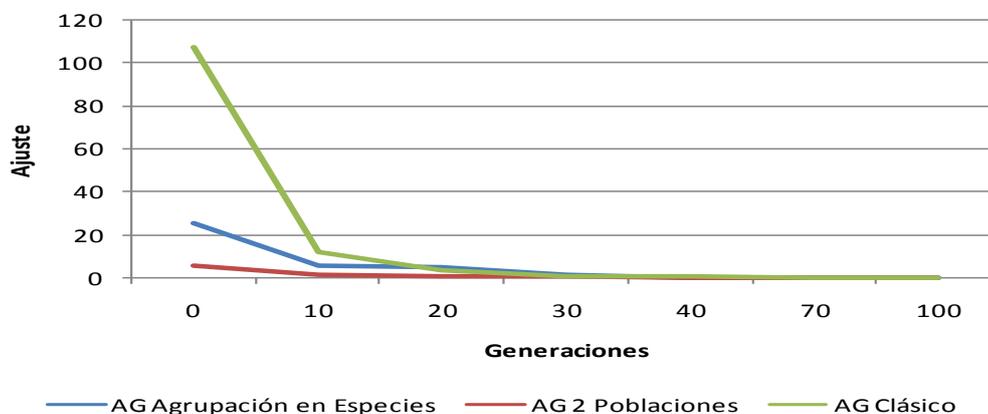


Figura 6.9: Schewefel: Evolución del error

Teniendo en cuenta que, dentro del espacio de búsqueda considerado para la función de Schewefel, se encuentran un total de 34 mínimos locales, en la Figura 6.10 se representa el porcentaje de soluciones encontradas por cada una de las técnicas.

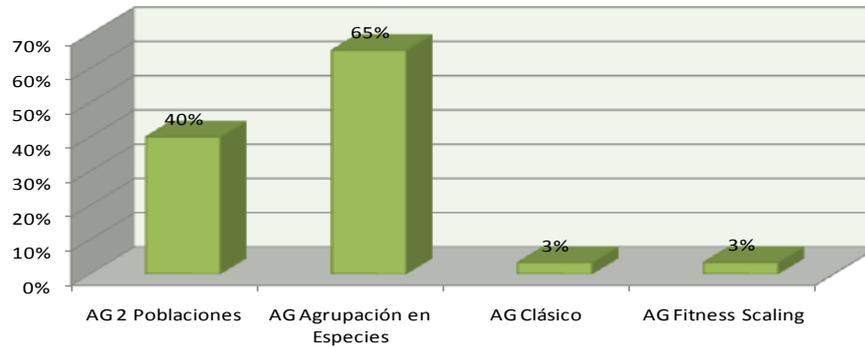


Figura 6.10: Schewefel: Porcentaje de soluciones encontradas tras 100 generaciones

El porcentaje de soluciones encontradas por el Algoritmo Genético clásico y por el Algoritmo Genético con la opción de *fitness scaling* es prácticamente idéntico tras 100 generaciones, debido, tal y como se comentó, a la tendencia de ambas opciones a focalizar la búsqueda en el entorno de la solución final. Considerando, en lugar de las soluciones encontradas al alcanzarse la generación número 100, el máximo número de soluciones encontradas a lo largo de la ejecución, el porcentaje sube al 16% para el Algoritmo Genético Clásico, y al 21% en el caso de emplearse la opción de *fitness scaling*.

6.3 Clasificación de Zumos Naturales

6.3.1 Introducción

Debido a las tendencias de la sociedad actual, cada día más concienciada de la necesidad de una alimentación saludable, la industria de fabricación de zumos de frutas ha experimentado en los últimos años un gran incremento en sus niveles de producción y ventas. Sin embargo, asociado a este incremento,

también se ha observado un aumento en el número de los casos de adulteración detectados. Podrían considerarse como causas principales de este incremento las dos razones a continuación expuestas:

- El beneficio económico obtenido, puesto que el precio de algunos zumos es realmente elevado.
- La dificultad de detección que presentan algunos edulcorantes. Algunos de ellos, por ejemplo la sacarosa, forman parte intrínseca de los propios zumos.

Al menos de momento, la Unión Europea no fuerza a declarar la cantidad de azúcares añadidos en las etiquetas de los productos. Por lo tanto la obtención de métodos que permitan determinar de manera precisa la cantidad de zumo puro empleado en la preparación de un producto se convierte en un aspecto cada vez más importante (Braakman, 2005; Heldman & Lund, 2007).

Para prevenir las adulteraciones, los productos han de someterse a una serie de controles de calidad, cada día más rigurosos. Los métodos de adulteración han progresado desde simples diluciones con agua y adición de azúcares, ácidos y colorantes hasta métodos mucho más sofisticados, lo que hace muy difícil su detección por las técnicas de análisis habituales. Pero también se han realizado significativos avances en el campo de la analítica, perfeccionándose los análisis existentes y desarrollándose otros nuevos (Hofsommer & Koswig, 2000). Entre los controles de calidad que se pueden aplicar se incluyen, por ejemplo HPLC (siglas de High Performance Liquid Chromatography) (Yuan & Chen, 1999), cromatografía de gases (Stöber, Martin, & Peppard, 1998) o métodos isotópicos (Jamin, Gonzalez, Remaud,

Naulet, & Martin, 1997). Debido a que todos ellos son lentos y relativamente caros no son muy convenientes para la realización de análisis rutinarios.

La aplicación de medidas químicas a la detección de adulterantes requiere una gran cantidad de datos reales acerca de la composición de zumos auténticos. Pero, además de la complejidad y variables naturales, debe tenerse presente que ciertas variables industriales inevitables, como el procesado y el almacenamiento de la fruta, pueden impactar en la composición del zumo (Robards & Antolovich, 1995).

Al contrario que el tipo de técnicas anteriormente mencionadas, la espectroscopía IR (Rayos Infrarrojos) (Stuart & Ando, 1998) presenta una alternativa rápida y barata que, además, aporta gran información acerca de los componentes mayoritarios de los zumos. Estos componentes mayoritarios son, a priori, los adulterantes típicos.

Un espectroscopio tiene la capacidad de proyectar un haz de luz monocromática o de infrarrojos (de una longitud de onda particular) a través de una muestra y medir la cantidad de luz que es absorbida por dicha muestra. Esto permite obtener información sobre la naturaleza de las sustancias incluidas en la muestra. Para ellos es necesario medir la absorbancia a distintas longitudes de onda (λ) y obtener una representación gráfica de dichos valores en función de la longitud de onda, formando un espectrograma (ver Figura 6.11).

Más formalmente, la absorbancia puede ser definida tal y como se refleja en la ecuación (Eq. 19)

$$A_{\lambda} = -\log_{10} \left(\frac{I}{I_0} \right) \quad (\text{Eq. 19})$$

dónde el valor I denota una intensidad de luz con longitud de onda λ después de que se haya producido la absorción de luz por parte de la muestra dada e I_0 determina la intensidad de luz que inicialmente se aplica a la muestra.

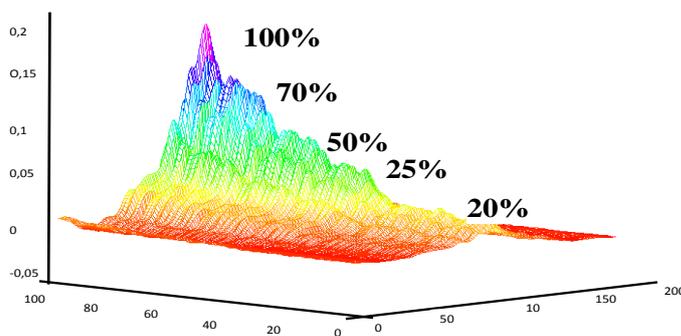


Figura 6.11: Espectros de zumos diluidos

Con este tipo de análisis de espectroscopía IR pueden detectarse los valores correspondientes a la concentración de cada tipo de azúcares: sacarosa, fructosa, glucosa, etc. Concretamente, esta detección se realizará en el ejemplo a continuación mostrado a partir de los valores de absorbancia de una muestra dada, tal y como se muestra en la Figura 6.11, para un conjunto de muestras con diverso porcentaje de zumo natural diluido (Gómez-Carracedo, Gestal, Dorado, & Andrade, 2007).

6.3.2 Clasificación de muestras

Para prevenir la citada adulteración es necesario poseer algún mecanismo que permita determinar el contenido real de zumo de cada una de las muestras introducidas en el espectrofotómetro. Dicho método podría consistir en considerar los datos obtenidos de la espectroscopía como las entradas de una red de neuronas artificiales. Dicha red neuronal podrá actuar

como un clasificador que permita categorizar cada muestra en función de su contenido real de zumo, basándose para ello en la información aportada por las absorbancias obtenidas por espectroscopio (Gestal et al., 2004). En la capa de salida se asignará un elemento de procesado a cada una de las categorías en las que se deseen clasificar los zumos, determinadas en función de las diferentes concentraciones de zumo natural que presenten las muestras. Bajo esta óptica, cada muestra o patrón de entrada se deberá asignar a la categoría indicada por el elemento de procesado que ofrezca la mayor salida.

Para poder realizar la clasificación de una muestra de esta manera, previamente deberá realizarse el entrenamiento de la red de neuronas. Por lo tanto se ha de disponer de un conjunto de entrenamiento y otro de validación, tal y como se describirá en la sección 6.3.3.

Con el conjunto de entrenamiento se proporcionará a la red de neuronas una serie de ejemplos a través de un conjunto de pares entrada-salida para que *aprenda* a partir de ellos. Es decir, para que extraiga de dicho conjunto de pares la información subyacente que permita posteriormente clasificar – en la terminología de redes de neuronas artificiales, generalizar – de manera correcta muestras nuevas no vistas durante esta etapa de entrenamiento.

Posteriormente se empleará el conjunto de validación para asegurar este hecho, comprobando que la red no ha memorizado los patrones de entrenamiento y que es capaz de generalizar de la manera adecuada. Esto es, debe comprobarse que es capaz de proporcionar una salida correcta para patrones no vistos en la fase de entrenamiento.

6.3.3 Descripción de los datos

Tal y como se ha comentado anteriormente, si se desea clasificar los datos obtenidos a través de la espectroscopía IR en base a su contenido en zumo, previamente será necesaria la definición de un conjunto de datos sobre los que la red de neuronas artificiales pueda emplearse posteriormente para extraer la información necesaria.

Para la obtención del conjunto de datos se ha contado con la colaboración del grupo de Química Analítica de la Universidade da Coruña. Así, a partir de zumos 100% puros – obtenidos a partir del licuado, centrifugado y filtrado de diferentes tipos de manzanas – se obtienen una serie de muestras de zumos puros diluidos. Estos zumos diluidos serán los empleados para la obtención de los diferentes conjuntos de entrenamiento y validación.

Tabla IV: Muestras concentraciones bajas (2%-20%)

Aprendizaje		Validación		Comerciales	
%	Muestras	%	Muestras	%	Muestras
2%	19	2%	1		
4%	17	4%	1		
6%	16	6%	13		
8%	22	8%	6		
10%	21	10%	6	10%	1
16%	19	16%	6	16%	1
20%	20	20%	6		
Total: 134		Total: 39		Total:2	

Se considerarán dos conjuntos de muestras diferenciados: uno para muestras con baja concentración en zumo (ver Tabla IV) y otro para muestras con alta concentración en zumo (ver Tabla V). Esta diferenciación en base al porcentaje de zumo se debe principalmente a la diferente nomenclatura

establecida por las normativas de la Unión Europea. Así, las bebidas con una baja concentración en zumo se encuadran dentro de lo que se denomina bebidas energéticas (dentro de las que se incluyen los refrescos), mientras que las bebidas con una mayor concentración reciben el nombre genérico de zumos.

Tabla V: Muestras concentraciones altas (20%-100%)

Aprendizaje		Validación		Comerciales	
%	Muestras	%	Muestras	%	Muestras
20%	20	20%	6	25%	2
25%	19	25%	18		
50%	16	50%	13		
70%	14	70%	1		
100%	17	100%	6	100%	19
Total: 86		Total: 44		Total:21	

En el caso de concentraciones de zumo difícilmente localizables en situaciones reales, se han obtenido un menor número de muestras de laboratorio. No se han descartado para conseguir de esta manera un conjunto final de resultados más completo. Adicionalmente, las muestras formadas por un 20% de zumo diluido (el límite entre lo que se consideran concentraciones *bajas* y lo que se consideran concentraciones *altas*) se han localizado en ambos conjuntos.

Todas las muestras fueron caracterizadas inmediatamente (empleando un espectroscopio Perkin Elmer 16PC-FTMIR con un dispositivo ATR) siguiendo para todas ellas exactamente el mismo procedimiento. Como resultado de esta caracterización se obtiene para cada una de las muestras un espectro como el mostrado en la Figura 6.12, en la que se digitaliza, mediante 176 números de onda o variables, la región en la que los tres principales

azúcares encontrados en los zumos naturales (glucosa, fructosa y sacarosa) absorben las radiaciones IR (Sivakesava & Irudayaraj, 2000).

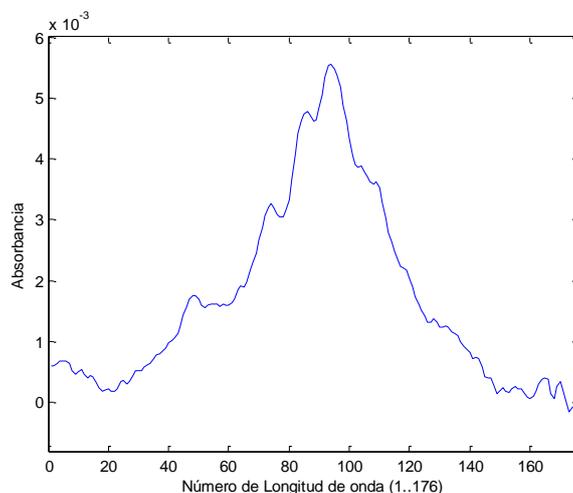


Figura 6.12: Espectro característico de una muestra

6.3.4 Selección de Variables

Uno de los puntos débiles de la espectroscopía IR reside en la cantidad de información generada. Trabajando con un rango de longitudes de onda típico de entre 1250cm^{-1} y 900cm^{-1} , una resolución de 4cm^{-1} y tomando 50 interferogramas/espectro se obtienen 176 valores espectrales diferentes para cada muestra. Además debe señalarse que no todos los valores espectrales obtenidos tendrán la misma influencia a la hora de determinar el contenido real en zumo de una muestra. Esto implica que con cada patrón se están introduciendo en la red de neuronas valores que no aportan demasiada información para la clasificación de una muestra.

Es por ello conveniente filtrar la información incluida en el espectro IR para quedarse únicamente con aquella que aporta una mayor cantidad de información a la hora de realizar la tarea de clasificación de una muestra (Gestal, Cancela, Andrade, & Gomez-Carracedo, 2007). Pero para poder

realizar de manera correcta ese proceso de filtrado, en primer lugar es necesario resolver dos cuestiones previas. En primer lugar debe establecerse un criterio que permita determinar cuál es el número óptimo de variables que desean mantenerse tras el proceso de filtrado o de selección de variables. En segundo lugar, debe conseguirse un modelo que sirva de referencia a los resultados obtenidos una vez seleccionadas las variables.

La primera de las tareas ha sido resuelta mediante un método derivado del Análisis de Componentes Principales, y ampliamente usado en el procesado de datos químicos, denominado Procrustes Rotation (Krzanowski, 2000). Básicamente, se trata de un criterio de tipo paramétrico que establece que el número mínimo de variables que deben ser retenidas equivale al número de componentes principales. Con el conjunto de datos existente, con únicamente 2 componentes principales puede explicarse el 99.76% de la varianza de los mismos. Por lo tanto, los métodos de selección de variables aquí propuestos tratarán de simplificar el conjunto inicial de variables de tal manera que puedan ser obtenidos resultados similares empleando sólo 2 de estas variables originales. Para los datos relativos a concentraciones bajas, con la técnica de Procrustes Rotation, las variables seleccionadas fueron la 84 y la 96 (correspondientes a las longitudes de onda 1084 cm^{-1} y 1000 cm^{-1} respectivamente), mientras que para las concentraciones altas se han seleccionado las variables 95 y 126 (correspondientes a las longitudes de onda 1064 cm^{-1} y 1000 cm^{-1} respectivamente). De entre estas longitudes de onda, la correspondiente a 1084 cm^{-1} está asociada al coeficiente de absorción de fructosa y glucosa, mientras que la correspondiente a 1064 cm^{-1} está asociada a la banda de la sacarosa (Cadet, 1999).

En cuanto a la tarea de establecer un modelo de referencia – detallado en la sección 6.3.4.1 – éste puede estar formado por los resultados de

clasificación empleando, bien la totalidad de las 176 variables originales del espectrograma, bien por las 2 variables seleccionadas por Procrustes.

6.3.4.1 Modelos de Referencia

Tal y como se ha comentado, se ha optado por constituir dos conjuntos de referencia. El primero de ellos formado por los resultados de clasificación obtenidos cuando se realiza la clasificación empleando la totalidad de los datos proporcionados por el espectrómetro (ver Tabla VI y Tabla VII). El segundo conjunto de referencia (ver Tabla VIII y Tabla IX) considera para la clasificación únicamente los datos aportados por las variables seleccionadas mediante Procrustes Rotation.

Para que el conjunto de referencia sea lo más completo posible, se ha realizado la clasificación a través de diversas técnicas, prestando especial énfasis a la clasificación mediante Redes de Neuronas Artificiales, puesto que, tal y como se observará posteriormente, será el método empleado por los diferentes Algoritmos Genéticos para establecer la bondad de las soluciones. De entre todas las redes disponibles, se ha optado para las pruebas mostradas en los siguientes apartados por una red *feedforward* totalmente conectada (la arquitectura dependerá de cada clasificación concreta) y entrenada mediante el algoritmo de retropropagación del error (*backpropagation*) debido a sus demostradas habilidades a la hora de abordar problemas relativos a clasificación de muestras (Bishop, 1995, 2006).

6.3.4.1.1 Clasificación con la totalidad de variables

El primero de los conjuntos de referencia, está constituido por los resultados obtenidos tras clasificar las diferentes muestras tomando como partida la totalidad de las 176 variables proporcionadas por el espectrómetro IR.

Tabla VI: Concentraciones Bajas: Errores de clasificación empleando 176 variables

	Entrenamiento (134 muestras)	Validación (39 muestras)	Comercial (2 muestras)
PLS	29	11	0
SIMCA	19	14	0
Potential Functions	4	9	0
R.N.A.	0	4	0

Por lo que respecta a la configuración de cada uno de los métodos de clasificación, cabe decir que el número de variables latentes empleado en el método de Partial Least Squares ha sido de 2 (LV=2), mientras que el número de factores por clase empleado en SIMCA ha sido el siguiente: 2%: 1 factor, 4%: 2 factores, 6%: 3 factores, 8%: 4 factores, 10%: 2 factores, 16%: 3 factores, 20%: 4 factores. El método de Potential Functions trabaja sobre el subespacio definido por los dos primeros componentes principales (PC1-PC2). En cuanto a la configuración de la red de neuronas artificiales, después de la realización de numerosas pruebas (Gómez-Carracedo et al., 2001), se ha obtenido como arquitectura óptima la formada por una capa de entrada con 176 elementos de procesado (1 por longitud de onda), 2 capas ocultas con 50 y 80 elementos de procesado respectivamente, y una capa de salida con 7 elementos de procesado (1 por cada categoría). La velocidad de aprendizaje, o *learning rate*, ha sido fijada a 0.0005 y se ha establecido como criterio de parada que el error cuadrático medio baje de 1 o bien que el proceso de entrenamiento alcance los 500.000 ciclos o *epochs*.

Tabla VII: Concentraciones Altas: Errores de clasificación empleando 176 variables

	Entrenamiento (86 muestras)	Validación (44 muestras)	Comercial (21 muestras)
PLS	11	5	1
SIMCA	15	12	1
Potential Functions	4	6	0
R.N.A.	0	1	0

En el caso de concentraciones altas, la configuración de cada uno de los métodos de clasificación es muy similar a la anteriormente comentada. Nuevamente, el número de variables latentes empleado en el método de Partial Least Squares ha sido de 2 (LV=2), mientras que el número de factores por clase empleado en SIMCA se ha establecido de la siguiente manera: 20%: 2 factores, 25%: 2 factores, 50%: 3 factores, 70%: 4 factores, 100%: 4 factores. Igual que en el caso anterior, el método de Potential Functions trabaja sobre el subespacio definido por los dos primeros componentes principales (PC1-PC2). En cuanto a la configuración de la red de neuronas artificiales, se ha obtenido como arquitectura óptima la formada por una capa de entrada con 176 elementos de procesado (1 por longitud de onda), 2 capas ocultas con 8 y 5 elementos de procesado respectivamente, y una capa de salida con 5 elementos de procesado (1 por cada categoría). Tal y como puede observarse, la configuración final de la red es significativamente menos compleja que la obtenida en el caso de concentraciones bajas. Este hecho es indicativo de la mayor complejidad existente en las muestras de concentración baja con respecto a concentraciones altas (debido a un mayor solapamiento entre las muestras). La velocidad de aprendizaje, o *learning rate*, ha sido fijada a 0.0005

y se ha establecido como criterio de parada que el error cuadrático medio baje de 0.5 o bien que el proceso de entrenamiento alcance los 500.000 *epochs*.

El error cuadrático medio empleado como criterio de parada en ambos casos (clasificación muestras concentración baja y alta) quizás pueda parecer demasiado elevado. Sin embargo, éste debe mantenerse en esos umbrales para evitar que exista una degradación en la capacidad de generalización de la red de neuronas debido a un sobreentrenamiento, que originaría que la red memorizase los patrones de entrenamiento en vez de extraer la información subyacente que le permita posteriormente ofrecer salidas válidas ante nuevos patrones.

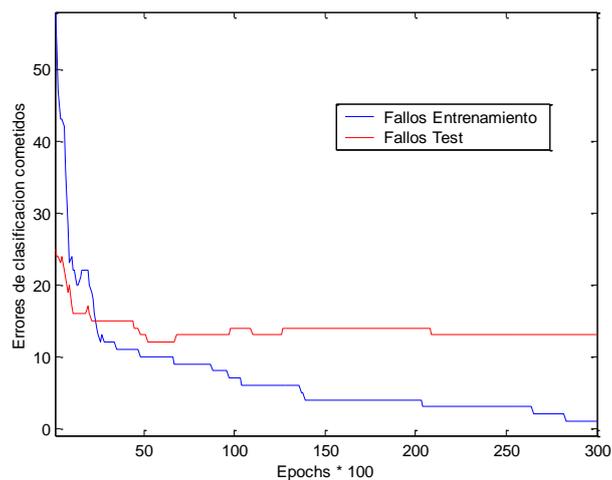


Figura 6.13: Evolución de los errores de clasificación en entrenamiento y validación

Esta situación puede verse reflejada en la Figura 6.13, correspondiente a un entrenamiento en el que se ha fijado a la red de neuronas un umbral de error cuadrático medio menor para detener dicho proceso.

Además, también ha de tenerse en cuenta que la salida que se busca es la indicación de la pertenencia de una muestra a una categoría dada (indicada por el elemento de procesado con una mayor salida), no un patrón específico

de salida. Es decir, si se busca que la red clasifique una muestra dentro de la primera categoría el patrón de salida sería [1; 0; 0; 0; 0]. Una posible salida de la red de neuronas podría ser [0.95; 0.23; -0.42; 0.12; -0.75], que tendría un elevado error cuadrático medio (puesto que está considerablemente alejado del patrón inicial), pero que sin embargo proporciona una clasificación correcta de la muestra (puesto que el elemento de procesado con la mayor salida es el correcto).

6.3.4.1.2 Clasificación con variables seleccionadas por Procrustes Rotation

El segundo de los modelos de referencia descritos en la sección 6.3.4.1 hacía referencia al formado por los resultados obtenidos tras clasificar las muestras existentes en base únicamente el conjunto de variables consideradas por el método de Procrustes que, tal y como se comentó anteriormente, era el formado por las variables 84 y 96 para concentraciones bajas y 95 y 126 para concentraciones altas.

Tabla VIII: Concentraciones Bajas: Errores de clasificación empleando las dos variables seleccionadas por Procrustes Rotation

	Entrenamiento (134 muestras)	Validación (39 muestras)	Comercial (2 muestras)
PLS	29	9	0
SIMCA	78	31	0
Potential Functions	28	11	0
R.N.A.	24	8	0

A continuación se detallan los parámetros de configuración de los diversos métodos de clasificación empleados. En primer lugar, el número de variables latentes en el método de Partial Least Squares ha sido de 2 (LV=2). Para el método SIMCA, se han definido 2 factores para cada una de las 7

categorías de clasificación. El método de Potential Functions trabaja sobre el subespacio definido por los dos primeros componentes principales (PC1-PC2). En cuanto a la configuración de la red de neuronas artificiales, el conjunto de parámetros que arrojó mejores resultados ha sido el formado por una red con 2 elementos de procesado en la capa de entrada (1 por cada una de las variables seleccionadas), una única capa oculta con 8 elementos de procesado y una capa de salida con 7 elementos de procesado identificando a cada una de las categorías de salida. El entrenamiento se realiza hasta que el error cuadrático medio baja de 0,25 o bien hasta que se alcanzan los 500.000 ciclos. Por último, el valor de 0.001 ha sido establecido como velocidad de aprendizaje.

Tabla IX: Concentraciones Altas: Errores de clasificación empleando las dos variables seleccionadas por Procrustes Rotation

	Entrenamiento (86 muestras)	Validación (44 muestras)	Comercial (21 muestras)
PLS	6	5	1
SIMCA	28	20	7
Potential Functions	4	5	1
R.N.A.	6	8	0

Para las clasificaciones relativas al conjunto de concentraciones bajas, al contrario que en los casos anteriores, el número de variables latentes empleado en el método de Partial Least Squares ha sido de (LV=1). Por otro lado, para SIMCA se mantienen los 2 factores por clase y para Potential Functions el marco de trabajo definido por los dos primeros componentes principales (PC1-PC2). En cuanto a la configuración óptima de la red de neuronas artificiales, se ha obtenido una arquitectura formada por una capa

de entrada con 2 elementos de procesado (1 por cada una de las variables seleccionadas), 1 capa oculta constituida por 5 elementos de procesado y una capa de salida también con 5 elementos de procesado (1 por cada categoría). Al igual que en el modelo de referencia formado a partir de las 176 variables originales, puede comprobarse como la configuración final de la red en este caso es ligeramente menos compleja que la obtenida en el caso de concentraciones bajas. Por lo tanto, puede corroborarse el hecho relativo a la mayor complejidad de los espectros relativos a las muestras con una menor concentración de zumo natural. En este caso, la velocidad de aprendizaje ha sido fijada nuevamente a 0.001 y se ha establecido como criterio de parada que el error cuadrático medio baje de 0.1 o bien que el proceso de entrenamiento alcance los 500.000 *epochs*.

6.3.4.2 Selección multimodal de variables

En trabajos previos (Gestal et al., 2004; Gestal et al., 2005) se han aplicado Algoritmos Genéticos para discernir de entre las variables proporcionadas por la espectroscopía IR cuáles de las variables aportaban una mayor cantidad de información a la hora de clasificar correctamente las muestras. En estos trabajos, basados en un Algoritmo Genético clásico, se ha empleado tanto una aproximación guiada como no guiada, como las comentadas en la sección 3.2.1.

Un denominador común de estas pruebas era la obtención de conjuntos de variables totalmente diferentes en las múltiples pruebas realizadas. No obstante, los métodos de clasificación realizados a partir de las variables seleccionadas proporcionaban niveles de ajuste prácticamente idénticos. Al obtenerse en cada prueba conjuntos de variables totalmente diferentes, se planteó la posibilidad de que cualquier conjunto de variables

contuviese la información suficiente para construir un buen sistema de clasificación. Sin embargo, la clasificación de muestras a partir de variables seleccionadas al azar arrojó resultados totalmente negativos.

Estas dos circunstancias apuntan de manera inequívoca a la naturaleza altamente multimodal del espacio de búsqueda del problema.

Por lo tanto, se plantea la aplicación de las dos técnicas propuestas en la presente Tesis al citado problema de clasificación. Ambas técnicas emplearán una misma función fitness, basada en el empleo de una red de neuronas artificiales para confirmar la validez del conjunto de variables seleccionadas. Esta función, basada en una aproximación *wrapper* (ver sección 2.1.3.2), puede verse de manera gráfica en la Figura 6.14.

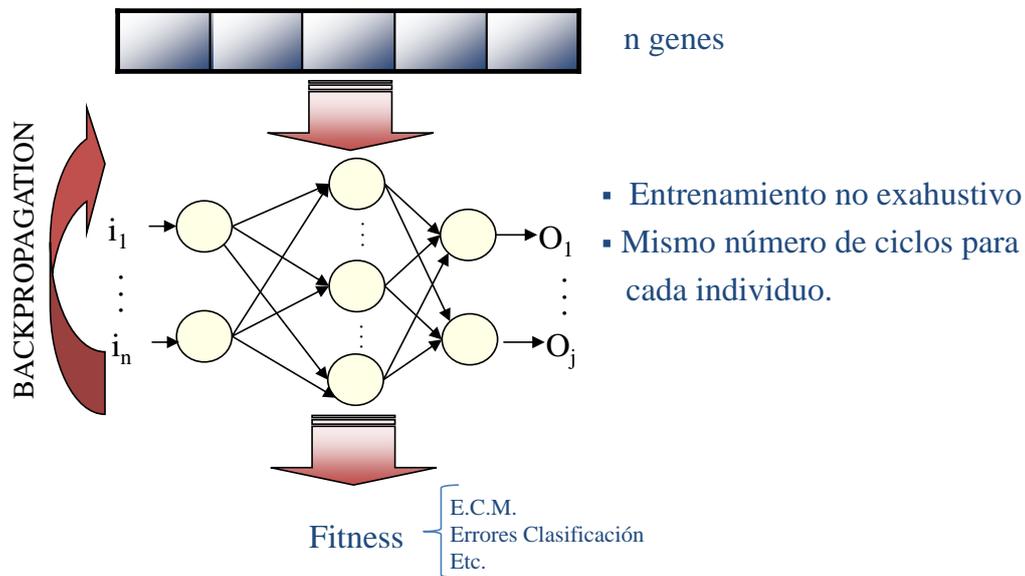


Figura 6.14: Clasificación muestras: función fitness

Tal y como se ha comentado, con ambas aproximaciones se pretende determinar cuáles de los valores obtenidos a partir del espectrofotómetro son realmente necesarios para establecer el porcentaje real de zumo de una

muestra. Filtrando los datos proporcionados por el espectroscopio se conseguirá, por una parte, acelerar el proceso de entrenamiento de las redes y, por otra parte, determinar qué partes de la señal son las que cuentan con una mayor información.

El genotipo de cada individuo genético determinará qué longitudes de onda de cada muestra se consideran descriptivas. Dichas longitudes de onda serán las empleadas en el posterior entrenamiento de la red de neuronas. Para determinar la bondad de una solución se construye una red neuronal y, tomando como entradas las longitudes de onda asociadas a los genes activos, se entrena la red durante varios ciclos. Es importante señalar que el objetivo de este proceso no consiste en entrenar totalmente la red para que esta clasifique con el menor error la totalidad de los conjuntos de entrenamiento y validación. El objetivo del entrenamiento será determinar si la alternativa formada por el conjunto de variables seleccionadas puede considerarse viable o, por el contrario, el error es tan elevado que debe descartarse. Igualmente deberá tenerse en cuenta que, para que los valores de ajuste de los diferentes individuos sean comparables, el número de ciclos de todos los entrenamientos deberá ser idéntico, al igual que la arquitectura de la red. En este caso, se ha observado que el error de la red deja de tener saltos abruptos y comienza a estabilizarse aproximadamente a partir de los 5000 ciclos, por lo que para asegurar que se alcanza dicho punto se ha optado por entrenar la red construida a partir de cada individuo genético durante 10000 ciclos.

A la hora de determinar la bondad de un individuo pueden escogerse diferentes métricas: basadas en la clasificación realizada, como porcentaje de muestras correctamente clasificadas, o bien basadas en parámetros de la red de neuronas empleada para el entrenamiento, como el error absoluto o error cuadrático medio, siendo este último la alternativa finalmente escogida.

Para minimizar en lo posible la variabilidad del proceso, para un conjunto de variables dado, se repite cinco veces el proceso de entrenamiento, tomándose para determinar bondad del individuo la media de los errores en todos los entrenamientos.

6.3.4.2.1 Selección multimodal de variables mediante Algoritmo Genético de dos Poblaciones

Tal y como se ha descrito en la sección 5.2.1, el empleo de un Algoritmo Genético de 2 poblaciones tiene como objetivo principal mantener constante la exploración sobre la mayor superficie posible del espacio de búsqueda. Para ello se añade una nueva población genética al Algoritmo Genético clásico, pero con la peculiaridad de que los valores de sus genes pueden variar únicamente dentro de un rango prefijado, justamente para evitar que la búsqueda se focalice entorno a una subregión del espacio de búsqueda.

Si el número total de individuos de la población genética es z (y por tanto de subregiones definidas) y el rango en el que los genes de los individuos podrán variar es p (en este caso el número total de longitudes de onda consideradas, es decir, 176), el ratio p/z determinará el rango en el que cada uno de los genes de la piscina genética podrá variar. Deberá tenerse en cuenta que no es conveniente que este ratio sea demasiado pequeño, lo que dificultaría enormemente el proceso de búsqueda debido a la imposibilidad de los genes de variar, ni demasiado grande, puesto que originaría subregiones excesivamente grandes en las que potencialmente se encuadrasen múltiples mínimos óptimos. En las pruebas anteriormente referidas se ha detectado que el valor del ratio escogido no es crítico, siempre y cuando incluya rangos razonables (entre 2 y 10), notándose una ligera ventaja en los modelos con menores valores. En este caso se ha optado por un

valor $p/z=4$, de tal manera que la piscina genética estará formada por un total de 44 individuos, cuyos genes podrán variar en intervalos de 4 unidades.

Tabla X: AG 2 Poblaciones: Errores Clasificación en Concentraciones Bajas

	Entrenamiento (134 muestras)	Validación (39 muestras)	Comercial (2 muestras)
[87 102]	4	11	1
[88 89]	14	10	2
[88 92]	7	9	2
[88 102]	4	10	2
[85 102]	0	11	1

En la Tabla X y en la Tabla XI se muestran los resultados de sendas ejecuciones del Algoritmo Genético de dos poblaciones, una empleando datos relativos a muestras con concentraciones bajas en zumo y otra con muestras con concentraciones altas. Cabe destacar que estos resultados no son ofrecidos por las redes de neuronas empleadas para validar la bondad de un individuo genético, sino que, una vez finalizado el proceso de selección, comienza un proceso de entrenamiento exhaustivo con los conjuntos de variables seleccionados.

Puede observarse como los resultados, aún siendo ligeramente inferiores a los obtenidos cuando se realizaba la clasificación con todas las longitudes de onda, sí son muy similares a los obtenidos a partir de las variables seleccionadas por Procrustes Rotation. De hecho, los resultados de entrenamiento de la red de neuronas son, en este caso, netamente superiores que los obtenidos en el caso anteriores, mientras que los resultados de validación son prácticamente idénticos.

Con respecto a los resultados es necesario resaltar dos aspectos. Por una parte, cabe citar que todos los conjuntos de variables empleados para categorizar las muestras de cada uno de los grupos de concentración han sido obtenidos tras una única ejecución del algoritmo. Es decir, que todas las soluciones han sido aportadas por una misma población genética. En el caso de la ejecución con el Algoritmo Genético clásico, su ejecución derivaba en la obtención de una solución válida (dos a lo sumo) mientras que el resto de individuos de la población no aportaban soluciones válidas en cuanto a calidad de la clasificación. Por otra parte, cabe destacar que la mayoría de los errores cometidos se corresponden a muestras clasificadas, en vez de en su categoría correcta, en una categoría adyacente. Es decir, si una muestra con un 8% de zumo se clasifica de manera errónea, lo más probable es que se encuadrese dentro de la categoría del 6% o del 10%.

Tabla XI: AG 2 Poblaciones: Errores Clasificación en Concentraciones Altas

	Entrenamiento (86 muestras)	Validación (44 muestras)	Comercial (21 muestras)
[52 141]	4	8	1
[102 129]	3	10	0
[145 175]	1	14	4
[18 149]	1	12	3

6.3.4.2.2 Selección multimodal de variables mediante Algoritmo Genético con Agrupación en Especies

En la Tabla XII y en la Tabla XIII se observan los conjuntos de variables seleccionados a partir de la técnica de Agrupación en Especies. En este caso los resultados también pueden considerarse comparables con los obtenidos cuando se clasifica con las variables seleccionadas mediante Procrustes. De

hecho alguno de los resultados obtenidos – por ejemplo los proporcionados por el conjunto de variables [9 171] – los mejoran notablemente.

Tabla XII: Agrupación en Especies: Errores Clasificación en Concentraciones Bajas

	Entrenamiento (134 muestras)	Validación (39 muestras)	Comercial (2 muestras)
[88 97]	3	7	1
[96 124]	2	9	0
[3 166]	4	11	2
[16 18]	5	20	2

Puede observarse también como los conjuntos de variables seleccionadas difieren notablemente de los obtenidos en el apartado anterior, lo que redundaría en la percepción acerca de la multimodalidad del espacio de búsqueda que muestra el presente problema.

Tabla XIII: Agrupación en Especies: Errores Clasificación en Concentraciones Altas

	Entrenamiento (86 muestras)	Validación (44 muestras)	Comercial (21 muestras)
[9 171]	0	4	3
[12 15]	3	10	3
[161 165]	3	11	4

Al igual que en el caso anterior, todos los conjuntos de variables empleados para la clasificación de cada uno de los dos grupos de concentraciones existentes, han sido obtenidos en una única ejecución del algoritmo.

6.3.4.3 Consideraciones acerca de las pruebas

Tal y cómo se ha destacado en los apartados anteriores, los resultados mostrados en las diferentes tablas, se corresponden a resultados obtenidos a partir de una única ejecución de los diferentes algoritmos. De esta manera se demuestra la capacidad de ambos de proporcionar, en una única ejecución múltiples soluciones válidas, tal y como del análisis de las citadas tablas se desprende. Es decir, se demuestra el correcto funcionamiento de las técnicas propuestas en un espacio de búsqueda multimodal.

Evidentemente se han realizado múltiples pruebas con ambas técnicas, obteniéndose resultados similares en todas ellas en cuanto a la calidad del sistema final de clasificación. Se ha observado una alta variabilidad en los subconjuntos de longitudes de onda seleccionados para realizar la clasificación, a pesar de lo que, según se ha comentado, las clasificaciones realizadas a partir de dichos subconjuntos son prácticamente idénticas. Esto no hace sino ahondar en la alta multimodalidad percibida en el problema a resolver.

Indudablemente, una manera de minimizar esta variabilidad de resultados entre las diferentes pruebas sería incrementar el número de individuos en la población tradicional, en el caso del Algoritmo Genético con 2 poblaciones, o el máximo número de especies permitido, en el Algoritmo de Agrupación en especies. De esta manera se facilitarían que una misma ejecución pudiese proporcionar un mayor número de soluciones. Sin embargo, debido a que la evaluación de cada individuo genético requiere el entrenamiento de una red de neuronas (realmente varios entrenamientos para minimizar la variabilidad del error obtenido), el tiempo total requerido sería difícilmente asumible.

Capítulo 7

Conclusiones

En aquellos problemas cuya resolución implica la búsqueda en espacios multimodales, los algoritmos evolutivos, en general, y los Algoritmos Genéticos, en particular, pueden quedar atrapados en un óptimo local, sin ser capaces de alcanzar la solución global del problema. Para que se produzca una exploración realmente eficaz del espacio de búsqueda multimodal, y la búsqueda converja al óptimo global – sin que ello implique que se descarten el resto de óptimo locales – es un requisito indispensable mantener e incrementar la diversidad de la población en evolución.

Es también altamente interesante mantener el máximo número de óptimos locales en la población a lo largo de las generaciones, puesto que ellos también representan soluciones potenciales al problema que se plantea resolver que, en ocasiones, y bajo ciertas circunstancias, quizás sean más factibles de implementar, bien por cuestiones de simplicidad, de eficiencia, de menor coste de implantación, o simplemente por ser más fáciles de interpretar.

Ambos aspectos, por un lado evitar que se produzca la caída del método de búsqueda empleado y por otro proporcionar no sólo una única solución global sino las múltiples existentes (bien sean globales o locales), son especialmente cruciales cuando los problemas son relativos a procesos de selección de variables.

Tal y como se planteaba en uno de los objetivos de la presente Tesis, se ha procedido al desarrollo de dos métodos para la selección de variables en problemas que impliquen la exploración de espacios de búsqueda multimodal. Ambos métodos están basados en técnicas de Computación Evolutiva.

El primero de estos métodos parte de un Algoritmo Genético clásico, modificándolo mediante la inclusión de una nueva población genética que actuará a modo de repositorio de información con el objetivo de mantener la diversidad genética a lo largo de toda la ejecución del algoritmo. Los operadores genéticos habituales de selección, cruce y mutación han sido adecuados para que trabajen con información proveniente de las dos poblaciones genéticas, la tradicional y la añadida, denominada piscina genética.

El segundo de los métodos intenta alcanzar la misma meta pero por un camino totalmente distinto. En este caso se parte de la premisa que la Naturaleza es uno de los sistemas más intensamente multimodales que puedan encontrarse. Múltiples organismos, generalmente agrupados en torno a una especie, han encontrado diferentes maneras de alcanzar el objetivo final, la supervivencia de dicha especie. Y lo que es más importante, las diferentes especies conviven y evolucionan en un mismo entorno, cada una defendiendo su territorio y luchando por mejorar la especie generación a generación. Este ha sido el esquema que se ha llevado a la práctica, de manera simplificada, de tal forma que diferentes especies de individuos genéticos representen las diferentes maneras de dar solución a un problema dado.

Ambos métodos, tal y como se planteaba como punto de partida, han demostrado su validez en los problemas planteados en la sección 6 de la presente Tesis, en primer lugar con una serie de ejemplos típicos cuando se habla de espacios de búsqueda multimodal como son las funciones de Rastrigin, Ackley y Schewefel. A continuación, se han probado en un ejemplo con una vertiente más cercana a aspectos prácticos y no meramente teóricos, que involucraba la selección de variables sobre un espacio de búsqueda netamente multimodal.

Tal y como se puede comprobar en base a los resultados obtenidos, ambos métodos propuestos ofrecen resultados altamente satisfactorios.

En este último ejemplo enfocado a la clasificación de muestras en base a su contenido en zumo natural, se ha empleado en los dos métodos desarrollados la misma métrica para definir la bondad de cada una de las soluciones aportadas. Concretamente se han empleado redes de neuronas artificiales para determinar si el conjunto de variables seleccionadas como significativas por el individuo genético eran o no una buena elección. Dichas

redes de neuronas tomaban como parámetros de entrada los valores especificados por el individuo genético y comenzaban el proceso de entrenamiento a partir de ellos. Aunque no es necesario finalizar dicho proceso de entrenamiento, los ciclos realizados permiten establecer si las variables seleccionadas configuran un punto de partida válido para realizar la clasificación.

A pesar de ocasionar unos tiempos de ejecución considerablemente elevados, este hecho aporta la considerable ventaja de ser trasladable a prácticamente cualquier otro problema, debido a la alta versatilidad de las redes de neuronas artificiales. Además, es aplicable en problemas de selección de variables en los que se emplea una técnica guiada o no guiada de búsqueda únicamente con unos mínimos cambios (en el primer caso la estructura de la capa de entrada de la red es fija, mientras que en el segundo caso la definirá cada uno de los individuos genéticos en el momento de su evaluación).

Capítulo 8

Futuros Desarrollos

Las líneas de trabajo futuro necesariamente han de ir encaminadas a la mejora de los métodos presentados, así como a incrementar las evidencias existentes en cuanto a su correcto funcionamiento.

En lo que respecta al Algoritmo Genético de dos poblaciones, una vez demostrada la eficacia de dividir el espacio de búsqueda en subregiones para evitar la homogeneización de la población genética, será interesante investigar en nuevas métricas que desemboquen en la división o agrupamiento de subregiones. Estas métricas deberían incorporar parámetros

más cercanos al estado de la población principal como puede ser la densidad de individuos en una u otra región. Esto sin duda redundaría en un control más eficiente del número de subregiones que la *simple* monitorización estadística actual acerca del número de veces que una región ha contenido a la mejor solución o el número de generaciones que lleva sin proporcionar un individuo que mejore al resto de los existentes.

Igualmente sería interesante definir y comprobar la utilidad de nuevos mecanismos de compartición de información entre la población principal y la piscina genética, actualmente en manos únicamente del operador de cruce. Aquí se abre la posibilidad de transferir bloques de información desde el *repositorio* de información a la población principal bien periódicamente bien en base a criterios, por ejemplo, de pérdida de diversidad. Este hecho incidiría especialmente en la variabilidad del número de regiones definidas por la nueva población y en el hecho de detener la ejecución prematuramente para evitar la homogeneización de las soluciones.

Una vez vista la buena respuesta ofrecida por el método de agrupación en especies, se abre una prometedora línea de investigación relativa a la introducción de nuevos aspectos observados en la Naturaleza y que tienen una relación o incidencia directa con la supervivencia. Especialmente interesante parece introducir el concepto de presa-predador en el modelo descrito, el concepto de envejecimiento de poblaciones o bien el relativo a la consideración de aspectos culturales o de memoria histórica referentes a las lecciones aprendidas por una especie acerca de actuaciones pasadas.

También resultaría de interés adentrarse en los aspectos relativos a la distribución de poblaciones. No sólo por lo que pueda aportar a la hora de reducir tiempos de ejecución, sino por las ventajas que podrían obtenerse de

la evolución aislada de las diferentes especies y su posterior agrupamiento e interacción cada cierto número de generaciones.

A pesar de los resultados altamente satisfactorios obtenidos con el empleo de redes de neuronas artificiales para el cálculo del valor de ajuste de los individuos, sería altamente sugestivo investigar otro tipo de métricas que, manteniendo la premisa de adaptabilidad a múltiples escenarios, permitiesen mitigar en lo posible los problemas de variabilidad inherentes a todo proceso de aprendizaje de una red de neuronas. En este aspecto parece especialmente prometedora la introducción de métricas basadas en SVM (siglas de Support Vector Machine).

Y por último, aunque no por ello menos importante, tal y como se comentó al inicio de esta sección, se considerará vital cualquier paso dado en la dirección de incrementar las evidencias existentes en cuanto al correcto funcionamiento de las técnicas propuestas. Para ello no hay mejor manera que comprobar el funcionamiento de dichas técnicas sobre diferentes conjuntos de datos, de entre los que destacan aquellos de origen biomédico. Este campo se adecúa perfectamente al ámbito de aplicación de las técnicas aquí mostradas debido generalmente a la cada día más elevada cantidad de datos descriptivos asociados a una muestra. Esta explosión de información debida a la aparición de nuevos métodos de diagnóstico o mejora de los ya existentes acarrea en ocasiones el hecho de que la gran cantidad de información disponible acerca de un hecho llega a dificultar en ocasiones su correcta interpretación.

Bibliografía

- Abdi, H. (2003). Partial least squares (PLS) regression. In *Encyclopedia of social sciences research methods* (ed. M. Lewis–Beck, A. Bryman and T. Futing) (pp. 1-7).
- Ackley, D. H. (1987). *A connectionist machine for genetic hillclimbing*: Kluwer Academic Publishers.
- Alba, E., & Tomassini, M. (2002). Parallelism and evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 6(5), 443-462.
- Aldrich, C., & van Deventer, J. S. J. (1995). Comparison of Different Artificial Neural Nets for the Detection and Location of Gross Errors in Process Systems. *Industrial & Engineering Chemistry Research*, 34(1), 216-224.
- Ariew, R. (1976). *Ockham's Razor: A Historical and Philosophical Analysis of Ockham's Principle of Parsimony*: Champaign-Urbana, University of Illinois.

Bibliografia

- Bäck, T. (1996). *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*: Oxford University Press, USA.
- Barker, M., & Rayens, W. (2003). Partial least squares for discrimination. *Journal of Chemometrics*, 17(3), 166-173.
- Bate, S. T., & Jones, B. (2008). A review of uniform cross-over designs. *Journal of Statistical Planning and Inference*, 138(2), 336-351.
- Beasley, D., Bull, D. R., & Martin, R. R. (1993). An Overview of Genetic Algorithms: Part 1, Fundamentals. *University Computing*, 15(2), 58-69.
- Beebe, K. R., Pell, R. J., & Seasholtz, M. B. (1998). *Chemometrics: a practical guide*: Wiley, New York.
- Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*: Oxford University Press, USA.
- Bishop, C. M. (2006). *Pattern recognition and machine learning*: Springer.
- Blickle, T., & Thiele, L. (1995). *A Mathematical Analysis of Tournament Selection*. Paper presented at the Sixth International Conference on Genetic Algorithms, Pittsburgh, PA, USA.
- Bloch, G., Sirou, F., Eustache, V., & Fatrez, P. (1997). Neural intelligent control for a steel plant. *IEEE Transactions on Neural Networks*, 8(4), 910-918.
- Booker, L. B., Fogel, D. B., Whitley, D. L., & Angeline, P. J. (1997). Recombination. In T. Bäck, D. B. Fogel & Z. Michalewicz (Eds.), *Handbook of Evolutionary Computation* (pp. 1--27). Bristol, New York: Institute of Physics Publishing and Oxford University Press.
- Bowden, G. J., Dandy, G. C., & Maier, H. R. (2003). Data transformation for neural network models in water resources applications. *Journal of Hydroinformatics*, 5(4), 245-258.
- Braakman, L. (2005). What to get ready for. *Food Engineering & Ingredients*, 27(6), 14-19.
- Brereton, R. G. (2000). Introduction to multivariate calibration in analytical chemistry. *The Analyst*, 125(11), 2125-2154.
- Brieman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees*: Wadsworth and Brooks, Inc.
- Brown, M., & Harris, C. (1995). *Neurofuzzy adaptive modelling and control*: Prentice Hall.

Bibliografía

- Cadet, F. (1999). Measurement of sugar content by multidimensional analysis and mid-infrared spectroscopy. *Talanta*, 48(4), 867-875.
- Cantú-Paz, E. (2000). *Selection Intensity in Genetic Algorithms with Generation Gaps*. Paper presented at the Genetic and Evolutionary Computation Conference, Las Vegas, Nevada, USA.
- Cantú-Paz, E., Newsam, S., & Kamath, C. (2004). *Feature selection in scientific applications*. Paper presented at the 2004 ACM international conference on Knowledge discovery and data mining (SIGKDD), Seattle, Washington, USA.
- Casale, S., Russo, A., & Serrano, S. (2007). Multistyle classification of speech under stress using feature subset selection based on genetic algorithms. *Speech Communication*, 49(10-11), 801-810.
- Cotta, C., Sloper, C., & Moscato, P. (2004). Evolutionary Search of Thresholds for Robust Feature Set Selection: Application to the Analysis of Microarray Data. *Applications of Evolutionary Computing, LNCS 3005*, 21-30.
- Cristianini, N., & Shawe-Taylor, J. (1999). *An introduction to support Vector Machines: and other kernel-based learning methods*: Cambridge University Press, New York, USA.
- Chauvin, Y., & Rumelhart, D. E. (1995). *Backpropagation: Theory, Architectures, and Applications*: Lawrence Erlbaum Associates.
- Darwen, P. J., & Yao, X. (1995). *A Dilemma for Fitness Sharing with a Scaling Function*. Paper presented at the Second IEEE International Conference on Evolutionary Computation, Piscataway, New York, USA.
- Darwin, C. (1859). *On the Origin of Species by Means of Natural Selection*: John Murray.
- Darwin, C. (2007). *Descent of Man*: Nuvision Publications.
- Dasgupta, D. (1994). *Handling deceptive problems using a different genetic search*.
- De Bonet, J. S., Isbell, C. L., & Viola, P. (1997). *MIMIC: Finding Optima by Estimating Probability Densities*. Paper presented at the Advances in Neural Information Processing Systems, Denver, Colorado, USA.
- De Jong, K. A. (1975). An analysis of the behavior of a class of genetic adaptative systems. *University of Michigan, Ann Arbor, MI, Ph.D. Thesis*.

Bibliografía

- De Jong, K. A., & Spears, W. M. (1992). A formal analysis of the role of multi-point crossover in genetic algorithms. *Annals of Mathematics and Artificial Intelligence*, 5(1), 1-26.
- Deb, K., & Goldberg, D. E. (1989). *An Investigation of Niche and Species Formation in Genetic Function Optimization*. Paper presented at the 3rd International Conference on Genetic Algorithms, San Mateo, California, USA.
- Deb, K., Sindhya, K., & Okabe, T. (2007). *Self-adaptive simulated binary crossover for real-parameter optimization*. Paper presented at the 9th annual conference on Genetic and Evolutionary Computation, University College London, London, UK.
- Etxeberria, R., & Larrañaga, P. (1999). *Global optimization using Bayesian networks*. Paper presented at the II Symposium on Artificial Intelligence (CIMAF-99), La Habana, Cuba.
- Faraway, J., & Chatfield, C. (1998). Time series forecasting with neural networks: a comparative study using the air line data. *Journal of the Royal Statistical Society: Series C: Applied Statistics*, 47(2), 231-250.
- Fogel, D. B. (2000). What is evolutionary computation? *Spectrum, IEEE*, 37(2), 26, 28-32.
- Fogel, L. J., Owens, A. J., & Walsh, M. J. (1966). *Artificial Intelligence through Simulated Evolution*: John Wiley.
- Freeman, J. A., & Skapura, D. M. (1991). *Neural Networks: Algorithms, Applications, and Programming Techniques*: Addison-Wesley.
- Gestal, M. (2002). Evaluación Distribuida de Algoritmos Genéticos con Ajuste Fino. A Coruña: Proyecto Fin de Carrera Ingeniería Informática, UDC.
- Gestal, M., Cancela, A., Andrade, J. M., & Gomez-Carracedo, M. P. (2007). Several Approaches to Variable Selection by Means of Genetic Algorithms. *Intelligent Information Technologies: Concepts, Methodologies, Tools and Applications*.
- Gestal, M., Gómez-Carracedo, M. P., Andrade, J. M., Dorado, J., Fernández, E., Prada, D., et al. (2004). Classification of apple beverages using artificial neural networks with previous variable selection. *Analytica Chimica Acta*, 524(1-2), 225-234.

Bibliografía

- Gestal, M., Gómez-Carracedo, M. P., Andrade, J. M., Dorado, J., Fernández, E., Prada, D., et al. (2005). Selection of Variables by Genetic Algorithms to classify Apple Beverages by Artificial Neural Networks. *Applied Artificial Intelligence*, 19(2), 181-198.
- Gestal, M., Vázquez-Naya, J. M., & Ezquerra, N. (2008). Genetic Algorithms and Multimodal Search. *Advancing Artificial Intelligence Through Biological Process Applications*, 231.
- Goldberg, D. E. (1989a). Genetic algorithms and Walsh functions: Part II, deception and its analysis. *Complex Systems*, 3(2), 153-171.
- Goldberg, D. E. (1989b). *Genetic algorithms in search, optimization, and machine learning*. Reading, Massachusetts: Addison-Wesley.
- Goldberg, D. E., & Deb, K. (1991). A Comparative Analysis of Selection Schemes Used in Genetic Algorithms. *Urbana*, 51, 61801-62996.
- Goldberg, D. E., Deb, K., Kargupta, H., Harik, G., & Forrest, S. (1993). *Rapid Accurate Optimization of Difficult Problems Using Fast Messy Genetic Algorithms*. Paper presented at the Fifth International Conference on Genetic Algorithms.
- Goldberg, D. E., Deb, K., & Korb, B. (1990). An investigation of messy genetic algorithms. *The Clearinghouse for Genetic Algorithms Group Report*, 90005.
- Goldberg, D. E., & Richardson, J. (1987). *Genetic algorithms with sharing for multimodal function optimization*. Paper presented at the Second International Conference on Genetic Algorithms and their application, Massachusetts Institute of Technology, Cambridge, MA, USA.
- Gómez-Carracedo, M. P., Andrade, J. M., Fernández, E., Dorado, J., Rabuñal, J. R., Prada, D., et al. (2001). *Clasificación de Refrescos y Zumos Comerciales de Manzana en Función de su Contenido Real en Zumo mediante Espectroscopia IR y Redes Neuronales*. Paper presented at the XV Encontro Galego Portugués de Química.
- Gómez-Carracedo, M. P., Gestal, M., Dorado, J., & Andrade, J. M. (2007). Linking chemical knowledge and genetic algorithms using two populations and focused multimodal search. *Chemometrics and Intelligent Laboratory Systems*, 87(2), 173-184.
- González-García, R., Rico-Martínez, R., & Kevrekidis, I. G. (1998). Identification of distributed parameter systems: A neural net based approach. *Computers and Chemical Engineering*, 22, 965-968.

Bibliografía

- Grefenstette, J. J. (1992). *Genetic algorithms for changing environments*. Paper presented at the Parallel Problem Solving from Nature, Bruselas, Bélgica.
- Guyon, I., Elisseeff, A., & Kaelbling, L. P. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, 3(7-8), 1157-1182.
- Harik, G. (1994). *Finding Multiple Solutions In Problems Of Bounded Difficulty*: IlliGAL Report No. 94002.
- Harik, G., & Eshelman, L. (1995). *Finding Multimodal Solutions Using Restricted Tournament Selection*. Paper presented at the Sixth International Conference on Genetic Algorithms.
- Hartigan, J. A., & Wong, M. A. (1979). A K-means clustering algorithm. *Applied Statistics*, 28(1), 100-108.
- Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation*. NJ, USA: Prentice Hall
- Hedar, A.-R. Test Functions for Unconstrained Global Optimization. Retrieved 29/01/2009, from http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO_files/Page364.htm
- Heldman, D. R., & Lund, D. B. (2007). *Handbook of Food Engineering*: CRC Press.
- Herrera, F., Lozano, M., & Sánchez, A. M. (2005). Hybrid crossover operators for real-coded genetic algorithms: an experimental study. *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, 9(4), 280-298.
- Hilera, J. R., & Martínez, V. J. (1995). *Redes neuronales artificiales. Fundamentos, modelos y aplicaciones*. Madrid: Addison-Wesley Iberoamericana S.A.
- Hofsommer, H. J., & Koswig, S. (2000). New analytical techniques for judging the authenticity of fruit juices. *Przem. Ferment. Owoc. Warz*, 44(3), 24-27.
- Holland, J. (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*: University of Michigan Press.
- Hoskins, J. C., & Himmelblau, D. M. (1992). Process control via artificial neural networks and reinforcement learning. *Computers & chemical engineering*, 16(4), 241-251.

- Hubert, M., & Branden, K. V. (2003). Robust methods for partial least squares regression. *Journal of Chemometrics*, 17(10), 537-549.
- Jamin, E., Gonzalez, J., Remaud, G., Naulet, N., & Martin, G. G. (1997). Detection of Exogenous Sugars or Organic Acids Addition in Pineapple Juices and Concentrates by ^{13}C IRMS Analysis. *Journal of Agricultural and Food Chemistry*, 45, 3961-3967.
- Jolliffe, I. T. (2002). *Principal Component Analysis* (2 ed.): Springer-Verlag, New York.
- Kalogirou, S. A. (2004). Optimization of solar systems using artificial neural-networks and genetic algorithms. *Applied Energy*, 77(4), 383-405.
- Kohavi, R. (1995). *Wrappers for performance enhancement and oblivious decision graphs*. Stanford university.
- Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2), 273-324.
- Kohonen, T. (1989). *Self-Organization and Associative Memory*. New York: Springer Verlag.
- Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*: MIT Press.
- Koza, J. R. (1999). *Genetic Programming III: Darwinian Invention and Problem Solving*: Morgan Kaufmann.
- Kreinovich, V., Quintana, C., & Fuentes, O. (1993). Genetic Algorithms: What fitness scaling is optimal? *Cybernetics and Systems*, 24(1), 9-26.
- Krzanowski, W. J. (2000). *Principles of Multivariate Analysis: A User's Perspective*: Oxford University Press.
- Lamont, G. B., & Merkle, L. D. (2003). Toward effective polypeptide structure prediction with parallel fast messy genetic algorithms. In *Evolutionary Computation in Bioinformatics* (pp. 137-162): Morgan Kaufmann.
- Land, A. H., & Doig, A. G. (1960). An automatic method for solving discrete programming problems. *Econometrica*, 28(3), 497-520.
- Lande, R. (1976). Natural selection and random genetic drift in phenotypic evolution. *Evolution*, 30(2), 314-334.
- Larrañaga, P., Etxeberria, R., Lozano, J. A., & Pena, J. M. (2000). *Optimization in continuous domains by learning and simulation of Gaussian networks*. Paper presented at the Genetic and Evolutionary Computation Conference (GECCO 2000), Las Vegas, Nevada, USA.

Bibliografía

- Larrañaga, P., & Lozano, J. A. (2002). *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*: Kluwer Academic Publishers.
- Larrañaga, P., Lozano, J. A., & Bengoetxea, E. (2001). *Estimation of distribution algorithms based on multivariate normal and Gaussian networks*. Vizcaya, España: Departamento de Informática e Inteligencia Artificial, Universidad del País Vasco.
- Li, J. P., Balazs, M. E., Parks, G. T., & Clarkson, P. J. (2002). A Species Conserving Genetic Algorithm for Multimodal Function Optimization. *Evolutionary Computation*, 10(3), 207-234.
- Lin, C. T., & Lee, C. S. G. (1996). *Neural fuzzy systems: a neuro-fuzzy synergism to intelligent systems*: Prentice-Hall, Inc. Upper Saddle River, NJ, USA.
- Lin, C. Y., & Wu, W. H. (2002). Niche identification techniques in multimodal genetic search with sharing scheme. *Advances in Engineering Software*, 33(11-12), 779-791.
- Liu, H., & Motoda, H. (2007). *Computational Methods of Feature Selection* Chapman & Hall / CRC.
- Looney, C. G. (1997). *Pattern recognition using neural networks: theory and algorithms for engineers and scientists*: Oxford University Press, Inc. New York, NY, USA.
- Lozano, J. A., Larrañaga, P., Inza I., & Bengoetxea, E. (2006). *Towards a New Evolutionary Computation: Advances in the Estimation of Distribution Algorithms*: Springer.
- Mahfoud, S. W. (1994). *Genetic drift in sharing methods*. Paper presented at the IEEE World Congress on Computational Intelligence, Orlando, USA.
- Mahfoud, S. W. (1995). *Niching Methods for Genetic Algorithms*. University of Illinois, Urbana-Champaign.
- Mahfoud, S. W. (2000). Niching methods. In T. Bäck, D. B. Fogel & Z. Michalewicz (Eds.), *Evolutionary Computation 1: Advanced Algorithms and Operators*.
- Mao, K. Z. (2004). Orthogonal forward selection and backward elimination algorithms for feature subset selection. *Systems, Man and Cybernetics, Part B, IEEE Transactions on*, 34(1), 629-634.
- Martens, H., & Martens, M. (2001). *Multivariate Analysis of Quality: An Introduction*: Wiley.

Bibliografía

- McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology*, 5(4), 115-133.
- Meert, K., & Rijckaert, M. (1998). Intelligent modelling in the chemical process industry with neural networks: a case study. *Computers and Chemical Engineering*, 22, 587-593.
- Michalewicz, Z., & Fogel, D. B. (2000). *How to Solve It: Modern Heuristics*: Springer Verlag.
- Miller, B. L., & Shaw, M. J. (1996). *Genetic algorithms with dynamic niche sharing for multimodal function optimization*. Paper presented at the IEEE International Conference on Evolutionary Computation, Nayoya University, Japon.
- Mühlenbein, H. (1997). The Equation for Response to Selection and Its Use for Prediction. *Evolutionary Computation*, 5, 303-346.
- Mühlenbein, H., Schomisch, M., & Born, J. (1991). The parallel genetic algorithm as function optimizer. *Parallel computing*, 17(6-7), 619-632.
- Naes, T., Isaksson, T., Fearn, T., & Davies, T. (2002). Multivariate Calibration and Classification. *NIR, Chichester, UK*.
- Narendra, K. S., Feiler, M. J., & Tian, Z. (2008). Control of Complex Systems Using Neural Networks. *Modeling and Control of Complex Systems*.
- Nascimento, C. A. O., Giudici, R., & Guardani, R. (2000). Neural network based approach for optimization of industrial chemical processes. *Computers and Chemical Engineering*, 24(9-10), 2303-2314.
- Pazos, A. (1996). *Redes de neuronas artificiales y algoritmos genéticos*: A Coruña: Servicio de Publicacións, Universidade da Coruña.
- Pearlmutter, B. A. (1990). *Dynamic Recurrent Neural Networks*: Technical Report CMU-CS. School of Computer Science, Carnegie Mellon University, 1990.
- Peterson, M. R., Raymer, M. L., & Lamont, G. B. (2005). *Balanced accuracy for feature subset selection with genetic algorithms*. Paper presented at the IEEE Congress on Evolutionary Computation, Edinburgh, UK.
- Petrowski, A. (1996). *A clearing procedure as a niching method for genetic algorithms*. Paper presented at the IEEE International Conference on Evolutionary Computation.
- Rechenberg, I. (1973). *Evolutionsstrategie -- Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Stuttgart: Frommann-Holzboog.

Bibliografía

- Rivals, I., & Personnaz, L. (2000). Nonlinear internal model control using neural networks: application to processes with delay and design issues. *IEEE Transactions on Neural Networks*, 11(1), 80-90.
- Robards, K., & Antolovich, M. (1995). Methods for assessing the authenticity of orange juice. A review. *The Analyst*, 120(1), 1-28.
- Rosipal, R., & Kramer, N. (2006). Overview and Recent Advances in Partial Least Squares. *Subspace, latent structure and feature selection techniques, Lecture Notes in Computer Science*, 34-51.
- Saeys, Y., Inza, I., & Larranaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19), 2507.
- Sareni, B., & Krahenbuhl, L. (1998). Fitness sharing and niching methods revisited. *IEEE Transactions on Evolutionary Computation*, 2(3), 97-106.
- Schwefel, H. P. (1981). *Numerical Optimization of Computer Models*: John Wiley & Sons, Inc. New York, NY, USA.
- Shi, J. J. (2000). Reducing Prediction Error by Transforming Input Data for Neural Networks. *Journal of Computing in Civil Engineering*, 14, 109.
- Shlens, J. (2005). A tutorial on principal component analysis. *Systems Neurobiology Laboratory, University of California at San Diego, December*.
- Sivakesava, S., & Irudayaraj, J. (2000). Determination of Sugars in Aqueous Mixtures Using Mid-infrared Spectroscopy. *Applied Engineering in Agriculture*, 16(5), 543-554.
- Spears, W. (1994). *Simple subpopulation schemes*. Paper presented at the Evolutionary Programming Conference, River Edge, NJ, USA.
- Spears, W., De Jong, K., & Rawlins, G. (1991). An Analysis of Multi-Point Crossover. In *Foundations of Genetic Algorithms* (pp. 301-315): San Francisco, CA: Morgan Kaufmann.
- Stöber, P., Martin, G. G., & Peppard, T. L. (1998). Quantitation of the undeclared addition of industrially produced sugar syrups to fruit by juices capillary gas chromatography. *Deutsche Lebensmittel-Rundschau*, 94(9), 309-316.
- Storn, R., & Price, K. (1997). Differential Evolution—A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11(4), 341-359.
- Stuart, B., & Ando, D. J. (1998). *Modern Infrared Spectroscopy: Analytical Chemistry by Open Learning*: Wiley, New York.

Bibliografia

- Sun, Z., Bebis, G., & Miller, R. (2004). Object detection using feature subset selection. *Pattern Recognition*, 37(11), 2165-2176.
- Talbi, E. G., Jourdan, L., Garcia-Nieto, J., & Alba, E. (2008). *Comparison of population based metaheuristics for feature selection: Application to microarray data classification*. Paper presented at the IEEE International Conference on Computer Systems and Applications (AICCSA 2008), Doha, Qatar.
- Tan, Q., Thomassen, M., Jochumsen, K. M., Zhao, J. H., Christensen, K., & Kruse, T. A. (2008). Evolutionary Algorithm for Feature Subset Selection in Predicting Tumor Outcomes Using Microarray Data. *Bioinformatics Research and Applications, LNCS 4983*, 426.
- Törn, A., & Zilinskas, A. (1989). Global optimization. *LNCS 350*.
- Wallace, A. R. (1855). On the law which has regulated the introduction of new species. *Annals and Magazine of Natural History*, 16(2nd Series), 184-196.
- Wang, H., Oh, Y., & Yoon, E. S. (1998). Strategies for modeling and control of nonlinear chemical processes using neural networks. *Computers and Chemical Engineering*, 22, 823-826.
- Wasserman, P. D. (1989). *Neural computing: theory and practice*: Van Nostrand Reinhold Co. New York, NY, USA.
- Whitley, L. D. (1991). Fundamental principles of deception in genetic search. In *Foundations of Genetic Algorithms* (pp. 221-241): Morgan-Kaufmann.
- Wiener, N. (1964). *God and Golem: a Comment on Certain Points where Cybernetics Impinges on Religion*: The MIT Press.
- Wold, H. (1975). Soft Modeling by Latent Variables. The Nonlinear Iterative Partial Least Square (NIPALS). *Perspectives in Probability and Statistics: Papers in Honour of MS Bartlett*. Applied Probability Trust. London: Academic, 117-142.
- Wright, A. H. (1991). *Genetic algorithms for real parameter optimization*. Paper presented at the Foundations of Genetic Algorithms.
- Wu, Z. Y. (2005). *Water Quality Model Calibration by Means of Fast Messy Genetic Algorithm*. Paper presented at the World Water and Environmental Resources Congress, Anchorage, Alaska, USA.
- Xu, J. C., Da-You, L., & Hu, M. (2004). *Feature selection and text classification for Chinese Web documents*. Paper presented at the Machine Learning and Cybernetics, Shanghai.

Bibliografia

- Yang, J., & Honavar, V. G. (1998). Feature Subset Selection Using a Genetic Algorithm. *IEEE Intelligent Systems*, 13(2), 44-49.
- Yin, X., & Gernay, N. (1993). *A fast genetic algorithm with sharing scheme using cluster analysis methods in multimodal function optimization*. Paper presented at the Artificial Neural Nets and Genetic Algorithms, Innsbruck, Austria.
- Yu, E., & Cho, S. (2006). Constructing response model using ensemble based on feature subset selection. *Expert Systems With Applications*, 30(2), 352-360.
- Yuan, J. P., & Chen, F. (1999). Simultaneous separation and determination of sugars, ascorbic acid and furanic compounds by HPLC—dual detection. *Food Chemistry*, 64(3), 423-427.

