



DEPARTAMENTO DE LENGUAJES Y CIENCIAS DE LA
COMPUTACIÓN
Universidad de Málaga

ASIGNACIÓN DE CRÉDITO MULTIOBJETIVO
EN COEVOLUCIÓN COOPERATIVA.
MODELO GENERAL Y APLICACIONES

TESIS DOCTORAL

ELOY RAFAEL SANZ TAPIA
Córdoba, septiembre de 2003

Memoria de tesis doctoral

Asignación de Crédito Multiobjetivo
en Coevolución Cooperativa.
Modelo General y Aplicaciones

Autor: Eloy Rafael Sanz Tapia

Directores

José Muñoz Pérez

César Hervás Martínez

Departamento de Lenguajes y Ciencias de la Computación
Universidad de Málaga

El Prof. Dr. D. José Muñoz Pérez, Catedrático de Universidad del Área de Ciencias de la Computación e Inteligencia Artificial de la Facultad de Informática de la Universidad de Málaga

Certifica

que D. Eloy Rafael Sanz Tapia, Licenciado en Informática, ha realizado en el Departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga bajo su dirección el trabajo de investigación correspondiente a sus tesis doctoral titulado

Asignación de Crédito Multiobjetivo en Coevolución Cooperativa. Modelo General y Aplicaciones

Revisado el mencionado trabajo, estima que puede ser presentado al tribunal que ha de juzgarlo y autoriza la defensa de esta Tesis Doctoral en la Universidad de Málaga.

Málaga, 2 de septiembre de 2003

Fdo.: Prof. Dr. D. José Muñoz Pérez
Catedrático de Universidad

El Prof. Dr. D. César Hervás Martínez, Catedrático de Escuela Universitaria del Área de Ciencias de la Computación e Inteligencia Artificial de la Escuela Politécnica Superior de la Universidad de Córdoba

Certifica

que D. Eloy Rafael Sanz Tapia, Licenciado en Informática, ha realizado en el Departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga bajo su dirección el trabajo de investigación correspondiente a sus tesis doctoral titulado

Asignación de Crédito Multiobjetivo en Coevolución Cooperativa. Modelo General y Aplicaciones

Revisado el mencionado trabajo, estima que puede ser presentado al tribunal que ha de juzgarlo y autoriza la defensa de esta Tesis Doctoral en la Universidad de Málaga.

Córdoba, 2 de septiembre de 2003

Fdo.: Prof. Dr. D. César Hervás Martínez
Catedrático de Escuela Universitaria

Este trabajo ha sido financiado en parte por la CICYT y Fondos Feder. Proyectos TIC-2002-04036-C05-02 (Department of Computer Science, University of Córdoba), TIC-2001-2577 y ALI98-0676.

A María Pilar y a Mai. Por María Pilar y por Mai.

Agradecimientos

Esto ha sido difícil. Compaginar el instituto con la Universidad y con la tesis ha requerido tiempo. Tiempo que no he dedicado a otros asuntos, y sobre todo a otras personas. Esas personas saben comprenderlo y eso me ayuda.

Y el tiempo y la paciencia que me han dedicado mis directores, mis compañeros y amigos del grupo AYRNA y tanta otra gente de la Universidad de Córdoba, de los institutos Trassierra, Vélez de Guevara y Gran Capitán... eso también ha ayudado mucho.

Gracias a todos.

Índice general

1. Introducción	1
1.1. Motivación	2
1.2. Objetivos	2
1.3. Metodología	3
1.4. Organización de la memoria	3
2. Estado actual	4
2.1. Coevolución	6
2.1.1. Coevolución competitiva	6
2.1.2. Coevolución cooperativa	8
Terminología y notación	10
División del problema	11
Evolución de los subcomponentes	12
Mezcla de los subcomponentes	13
Evaluación de los individuos	15
Evaluación de los subcomponentes	16
Modelo general	16
2.2. Optimización Multiobjetivo	21
2.2.1. Definición del problema	21
2.2.2. Conceptos	22
Dominancia o dominación	22
Optimalidad Pareto	23
2.2.3. Optimización Evolutiva Multiobjetivo	23

2.2.4.	Métodos no-Pareto	24
	Suma ponderada de Aptitudes	24
	VEGA	24
2.2.5.	Métodos Pareto	25
	MOGA	25
	NSGA	25
	NPGA	26
	PAES	27
	SPEA	28
2.2.6.	Técnicas de segunda generación	29
	NSGA-II	29
3.	Modelo Propuesto	33
3.1.	Introducción	33
3.2.	Descripción general	33
3.3.	Entidades	34
3.3.1.	Población de individuos	35
3.3.2.	Individuos	35
3.3.3.	Poblaciones de subcomponentes	37
3.3.4.	Subcomponentes	37
3.4.	Métodos	38
3.4.1.	Evolución de individuos	38
3.4.2.	Evolución de subcomponentes	39
3.4.3.	Creación de poblaciones iniciales: individuos	41
3.4.4.	Creación de poblaciones iniciales: subcomponentes	41
3.4.5.	Operadores genéticos: individuos	41
3.4.6.	Operadores genéticos: subcomponentes	42
3.4.7.	Operadores de selección	43
3.4.8.	Evaluación de individuos	43
3.4.9.	Evaluación de subcomponentes	43
	Tipos de crédito	44
	Métodos de asignación	46
3.5.	Ejecución del modelo	46

4. Implementación	48
4.1. Introducción	48
4.2. Esquema general	48
4.3. Estructuras de datos	49
4.4. Algoritmos	51
4.5. Adaptación al problema	59
4.6. Configuración	61
5. Problema: cobertura de vectores	64
5.1. Introducción	64
5.2. Problema de cobertura de vectores	64
5.2.1. Descripción	64
5.2.2. Generación coevolutiva de coberturas	66
5.2.3. Diseño experimental	68
5.2.4. Resultados	71
6. Problema: evolución de superficies de respuesta	79
6.1. Introducción a las superficies de respuesta	79
6.1.1. Concepto	79
6.1.2. Aproximación de funciones de respuesta	80
6.2. Generación coevolutiva de superficies de respuesta	84
6.3. Modelado de crecimiento microbiano: Leuconostoc	87
6.3.1. Descripción	87
6.3.2. Diseño experimental	88
6.3.3. Resultados	91
7. Conclusiones	99
7.1. Resultados de la aplicación del modelo	99
7.2. Contribuciones	100
7.3. Futuras mejoras	101
A. Resultados del problema de cobertura de vectores. Subproblema “medio”.	109

- B. Resultados del problema de cobertura de vectores. Subproblema “cuarto”.** 117
- C. Resultados del problema de cobertura de vectores. Subproblema “octavo”.** 125
- D. Resultados del problema de evolución de superficies de respuesta. Subproblema “Leuconostoc”.** 133

Índice de cuadros

2.1. Posibles relaciones entre especies en un ecosistema. El signo '+' implica un beneficio para la especie; el signo '-' un perjuicio; '0' indica que la especie no se ve afectada por la relación	7
3.1. Entidades y métodos.	35
3.2. Tipos de asignación de crédito proporcionados por el modelo	45
4.1. Entidades y métodos del problema.	60
5.1. Combinaciones de criterios para el diseño experimental. \checkmark significa "presente", - significa "no-presente".	70
5.2. Parámetros del proceso evolutivo. Cobertura de vectores.	70
5.3. Resultados para problema "half".	72
5.4. Resultados para problema "quarter".	73
5.5. Resultados para problema "eighth".	74
5.6. Resultados medios considerando/sin considerar cada criterio. Problema "medio".	75
5.7. Resultados medios considerando/sin considerar cada criterio. Problema "cuarto".	75
5.8. Resultados medios considerando/sin considerar cada criterio. Problema "octavo".	75
5.9. Pruebas de efectos inter-sujetos. Problema "medio". Criterios: Mejor (M), Media (P, promedio), Diferencia (D), Sustitución (S), Unión (U).	76
5.10. Pruebas de efectos inter-sujetos. Problema "cuarto". Criterios: Mejor (M), Media (P, promedio), Diferencia (D), Sustitución (S), Unión (U).	77
5.11. Pruebas de efectos inter-sujetos. Problema "octavo". Criterios: Mejor (M), Media (P, promedio), Diferencia (D), Sustitución (S), Unión (U).	78

6.1. Combinaciones de valores para el diseño del experimento <i>Leuconostoc</i> . Marcadas con * las condiciones centrales del DCC	89
6.2. Parámetros del proceso evolutivo. Crecimiento microbiano. <i>Leuconostoc</i>	91
6.3. Resultados para problema “grate”.	92
6.4. Resultados para problema “lnlag”.	93
6.5. Resultados para problema “yend”.	94
6.6. Resultados medios considerando/sin considerar cada criterio. <i>Leuconostoc</i> . Problema “grate”.	95
6.7. Resultados medios considerando/sin considerar cada criterio. <i>Leuconostoc</i> . Problema “lnlag”.	95
6.8. Resultados medios considerando/sin considerar cada criterio. <i>Leuconostoc</i> . Problema “yend”.	95
6.9. Pruebas de efectos inter-sujetos. Problema “grate”. Criterios: Mejor (M), Media (P, promedio), Diferencia (D), Sustitución (S), Unión (U).	96
6.10. Pruebas de efectos inter-sujetos. Problema “lnlag”. Criterios: Mejor (M), Media (P, promedio), Diferencia (D), Sustitución (S), Unión (U).	97
6.11. Pruebas de efectos inter-sujetos. Problema “yend”. Criterios: Mejor (M), Media (P, promedio), Diferencia (D), Sustitución (S), Unión (U).	98
A.1. Resultados completos. Cobertura de Vectores. Problema “medio”.	116
B.1. Resultados completos. Cobertura de Vectores. Problema “cuarto”.	124
C.1. Resultados completos. Cobertura de Vectores. Problema “octavo”.	132
D.1. Resultados completos. Superficies de respuesta. Problema “lnlag”.	140
D.2. Resultados completos. Superficies de respuesta. Problema “grate”.	147
D.3. Resultados completos. Superficies de respuesta. Problema “grate”.	154

Índice de figuras

2.1. Modelo general propuesto por Zhao para la coevolución cooperativa . . .	17
2.2. Notación que se usará en el modelo propuesto	20
3.1. Esquema general del modelo	34
3.2. Representación detallada de un individuo y sus relaciones con las pobla- ciones de subcomponentes.	36
3.3. Esquema de evolución de individuos (algoritmo AE_I)	39
3.4. Esquema de evolución de subcomponentes (algoritmo AE_S)	40
3.5. Cruce de dos puntos sobre individuos	42
3.6. Mutación de individuos	42
5.1. Ejemplo de cobertura de vectores. $N_T = 4, N_C = 2, L = 16$	66
6.1. (a) superficie de respuesta teórica para un proceso químico; (b) gráfica de contorno de la superficie de respuesta teórica.	80
6.2. (a) superficie de respuesta de primer orden; (b) gráfica de contorno . . .	81
6.3. (a) superficie de respuesta de segundo orden; (b) gráfica de contorno . .	82
6.4. Esquema general de una red neuronal artificial.	84
6.5. Genotipo para la evolución de superficies de respuesta.	85
6.6. División en subcomponentes de una superficie de respuesta	85
6.7. Cálculo de la tasa de crecimiento, punto de despegue y densidad máxima.	90

Índice de Algoritmos

1.	Algoritmo evolutivo clásico	5
2.	Ordenación No-dominada rápida	30
3.	Asignación de distancia de <i>crowding</i>	31
4.	Algoritmo genético de ordenación no-dominada II (NSGA-II)	32
5.	Ejecución del modelo	47
6.	Proceso Evolutivo General	52
7.	Salto generacional de individuos	53
8.	Mutación de individuos	53
9.	Cruce de individuos	53
10.	Salto generacional de subcomponentes	55
11.	Evaluación de subcomponentes	55
12.	Asignación de crédito	56
13.	Asignación de crédito: Mejor	56
14.	Asignación de crédito: Media	56
15.	Asignación de crédito: Sustitución	57
16.	Asignación de crédito: Diferencia	58
17.	Asignación de crédito: Unión	59
18.	Conversión multiobjetivo	59

Notación

A lo largo de este trabajo se usará una notación homogénea para referirse a las distintas entidades que forman parte del modelo propuesto. Se describe a continuación, a modo de referencia, dicha notación.

El modelo (descrito en el capítulo 3) consta de K poblaciones de subcomponentes denominadas PS_1, PS_2, \dots, PS_K y de una población de individuos llamada PI . Cada población de subcomponentes puede tener un tamaño (número de subcomponentes que alberga) diferente; estos tamaños se representarán mediante las constantes N_1, N_2, \dots, N_K . La población PI tiene un tamaño M .

$$\begin{aligned} |PI| &= M \\ |PS_1| &= N_1 \\ |PS_2| &= N_2 \\ &\vdots \\ |PS_K| &= N_K \end{aligned}$$

Los integrantes de cada población (subcomponentes e individuos) se representarán de la siguiente forma: los M individuos pertenecientes a la población PI se representarán como i_1, i_2, \dots, i_M ; los N_1 subcomponentes de PS_1 serán $sc_{1,1}, sc_{1,2}, \dots, sc_{1,N_1}$; los N_2 subcomponentes de PS_2 serán $sc_{2,1}, sc_{2,2}, \dots, sc_{2,N_2}$ y así sucesivamente hasta la población PS_K . En general, pues, los subcomponentes pertenecientes a la población p serán $sc_{p,1}, sc_{p,2}, \dots, sc_{p,N_p}$.

$$\begin{aligned} PI &= \{i_1, i_2, \dots, i_M\} \\ PS_1 &= \{sc_{1,1}, sc_{1,2}, \dots, sc_{1,N_1}\} \\ PS_2 &= \{sc_{2,1}, sc_{2,2}, \dots, sc_{2,N_2}\} \\ &\vdots \\ PS_K &= \{sc_{K,1}, sc_{K,2}, \dots, sc_{K,N_K}\} \end{aligned}$$

Cuando haga falta referirse a un subcomponente genérico, se usarán los subíndices p y q : $sc_{p,q}$. Una población de subcomponentes genérica se denotará por PS_p . Un individuo genérico se representará como i_j .

El operador \cup , cuando se aplique a subcomponentes, representará la *agregación* de los mismos para formar un individuo. Por ejemplo:

$$i_j = sc_{1,6} \cup sc_{2,13} \cup sc_{3,9} \cup sc_{4,10}$$

Introducción

La computación evolutiva trata de resolver problemas aplicando métodos de la genética de poblaciones. La evolución de una población de posibles soluciones al problema y la aplicación sobre las mismas de operadores de reproducción, cruce y mutación intentan aproximar dichas soluciones candidatas a una solución óptima.

Estas soluciones candidatas son tratadas como entidades monolíticas, y su modificación por los operadores genéticos se realiza sobre la solución completa. No hay en la computación evolutiva estándar una atención específica a la posibilidad de que existan subsoluciones: módulos de la solución completa que resuelvan subproblemas del problema completo. Holland propone la teoría de los esquemas [Holland, 1975] y la hipótesis de los bloques constructivos, y muchos autores posteriores la abrazan y la amplían.

Sin embargo la división de un problema en partes modulares (tratada en [Watson, 2002]) y la evolución por separado de soluciones específicas para dichas partes o subproblemas es un problema tratado específicamente por la coevolución cooperativa.

En este trabajo se propondrá un modelo general de coevolución cooperativa en el que la asignación de crédito a las subsoluciones que forman la solución candidata se realiza atendiendo a diferentes criterios. Estos criterios serán agrupados mediante técnicas de evolución multiobjetivo.

Se propondrá igualmente, durante la introducción de los criterios propuestos, un modelo general para la asignación de crédito en coevolución cooperativa.

El modelo propuesto, que es flexible y general, será adaptado para su uso en dos problemas principales: el problema de la cobertura de vectores y la evolución de superficies de respuesta para el modelado de sistemas. El primer problema usará tres diseños experimentales propuestos por Forrest [Forrest et al., 1993] y posteriormente por Potter [Potter y de Jong, 2000]. El segundo empleará datos experimentales sobre crecimiento de microorganismos.

1.1. Motivación

Los motivos principales para la orientación de este trabajo han sido varios. Entre ellos cabe destacar:

- El interés del autor por la arquitectura de la coevolución cooperativa, que presenta estructuras e interacciones cuya complejidad hace interesante su estudio.
- El escaso desarrollo del campo de la coevolución cooperativa, en parte debido a la escasez de problemas abordados por este tipo de métodos.
- La falta de marcos genéricos suficientemente detallados e implementados para el uso de los métodos coevolutivos cooperativos.
- El interés que presentan los problemas de la asignación de crédito y de la optimización multiobjetivo, así como el posible beneficio de la mezcla de ambos.
- La disponibilidad en el grupo de investigación del autor de datos utilizables para la experimentación en este campo.

1.2. Objetivos

Teniendo en cuenta los motivos enumerados arriba, el presente trabajo tiene como principales objetivos:

- Estudiar el estado actual del campo de la coevolución cooperativa. Abstractar implementaciones concretas y modelos cerrados, tratando de extraer puntos en común.
- Estudiar los problemas abordados por métodos de coevolución cooperativa.
- En función de los resultados de los dos puntos anteriores, diseñar y proponer un modelo general de coevolución cooperativa, que sea suficientemente flexible como para particularizarlo a métodos ya existentes.
- Implementar dicho modelo mediante un software configurable, flexible, eficiente y bien documentado.
- Integrar dentro del modelo propuesto un método de asignación de crédito que permita evaluar distintos objetivos en paralelo.
- Aplicar la implementación del modelo a problemas conocidos, usando los datos de que dispone el grupo de investigación.

1.3. Metodología

La metodología seguida en este trabajo es básicamente experimental. Cualquier intento de análisis teórico con las herramientas actuales y el estado en que se encuentra la coevolución es imposible.

Las pruebas realizadas han sido lo más rigurosas posible. Los problemas que se discuten son tanto problemas reales como problemas simulados. Las comparaciones entre diferentes resultados se refrendan siempre con contrastes estadísticos que nos informen de la significación de las diferencias entre los experimentos. Además, se plantean también contrastes de hipótesis que verifiquen que las hipótesis de partida de los contrastes de comparación de resultados utilizados se cumplen.

1.4. Organización de la memoria

Esta memoria detallará el trabajo realizado para la creación del modelo coevolutivo cooperativo con asignación de crédito multiobjetivo. En el capítulo 2 se hará un repaso del estado actual de los campos relacionados con el trabajo: la coevolución y la optimización evolutiva multiobjetivo. En el capítulo 3 se describirá el modelo general propuesto, así como el modelo de asignación de crédito y los criterios de asignación de crédito propuestos. El capítulo 4 tratará sobre la implementación que se ha realizado del modelo propuesto, detallando las estructuras de datos y algoritmos creados durante dicha fase de implementación. Esta implementación del modelo será adaptada a problemas reales para valorar su utilidad: en el capítulo 5 se aplicará a un problema de cobertura de vectores, y en el capítulo 6 se probará sobre la coevolución de superficies de respuesta para modelado de sistemas. El capítulo 7 extraerá conclusiones de los resultados obtenidos y marcará futuras líneas de investigación.

2

Estado actual

En esta sección se realizará un estudio previo de las soluciones existentes y del estado del arte en las áreas que conciernen al desarrollo de este trabajo. El objetivo es obtener una visión general sobre la que basar el desarrollo del modelo propuesto, así como disponer de información para el contraste de los resultados obtenidos en la experimentación.

Las áreas fundamentales sobre las que se ha realizado este estudio de estado actual son dos: la coevolución cooperativa y la optimización evolutiva multiobjetivo. Existen otras áreas sobre las que se basará el desarrollo de este trabajo, que serán tratadas en su momento de forma más escueta (por ejemplo, durante la presentación de los problemas a resolver usando el modelo).

El campo general de la computación evolutiva engloba una serie de paradigmas cuyas características comunes, inspiradas en los principios de la evolución natural, son

- el uso de un conjunto (población) de posibles soluciones (individuos) al problema que se trata de resolver
- la evolución de esto individuos utilizando operadores basados en la genética de poblaciones (principalmente la mutación y el cruce)
- la asignación a dichos individuos de un valor de aptitud (función de su bondad en la resolución del problema)
- la selección de los individuos de una generación que pasarán a la siguiente, realizada en base a sus aptitudes

Los métodos de computación evolutiva, en virtud del uso de una población de individuos, realizan una búsqueda en paralelo sobre el espacio de soluciones, lo que es especialmente interesante en problemas multimodales (se evita caer en mínimos locales) y en aquellos cuyo espacio de búsqueda es muy amplio.

Existen diferentes paradigmas de computación evolutiva que, aun compartiendo los puntos básicos anteriormente enumerados, difieren en aspectos como la representación de los individuos, el tipo de operadores genéticos aplicados. . . Así, tenemos

Algoritmo 1 Algoritmo evolutivo clásico

- 1: $t = 0$
 - 2: Inicializar P_t con individuos aleatorios
 - 3: Evaluar aptitudes de los individuos de P_t
 - 4: **mientras** condición de terminación no se cumpla
 - 5: Seleccionar individuos de P_t para reproducción en función de su aptitud
 - 6: Aplicar operadores genéticos a dichos individuos para obtener descendencia
 - 7: Evaluar aptitudes de los individuos de la descendencia
 - 8: Reemplazar miembros de P_t por miembros de la descendencia, obteniendo P_{t+1}
 - 9: $t = t + 1$
 - 10: **finmientras**
-

- los *algoritmos genéticos* clásicos[Holland, 1975][Goldberg, 1989], que codifican las soluciones mediante cadenas de bits
- las *estrategias evolutivas*[Bäck, 1996], que usan codificación real y estructuras de representación algo más complejas
- la *programación genética*, cuyos individuos son programas representados a menudo mediante árboles[Koza, 1994]
- la *programación evolutiva* [Fogel et al., 1966] y su codificación natural (no restringida) de las soluciones

Este último paradigma es de particular interés cuando se trata con problemas que presentan una estructura compleja, y se requiere el desarrollo de operadores de cruce y mutación específicos y adaptados a la codificación.

Tal como se ha señalado, la selección de los individuos que pasan de una generación a la siguiente se realiza en función de su aptitud. El valor de aptitud de un individuo mide su capacidad para resolver el problema objetivo, y habitualmente se representa mediante un único valor numérico.

Un esquema clásico y razonablemente general de algoritmo evolutivo se muestra en el algoritmo 1. Como puede verse, una población inicial P_0 aleatoria sufre una serie de transformaciones que van convirtiéndola en $P_1, P_2 \dots$ y que van modificando individuos seleccionados de la misma. Las modificaciones de estos individuos (los operadores genéticos) varían: simple duplicación (reproducción), modificación del genotipo del individuo (mutación), mezcla de los genotipos de dos o más individuos (cruce)...

Sin embargo, en lo que respecta a este estudio, interesan dos características del algoritmo evolutivo clásico. La primera es que las aptitudes de los individuos se miden con respecto a un patrón externo a la población, y esta medida tiene como único parámetro el genotipo del individuo. No se contempla ningún tipo de relación explícita entre los individuos de la población, ya sea esta de colaboración o de competición. Sí se puede argumentar que existe una competencia entre individuos durante la selección y en función de su aptitud, de forma indirecta (en los casos de selección por torneo) o directa

(en los casos de selección por ruleta, por ejemplo). Pero esta competencia la impone el algoritmo evolutivo en sí, sin que los individuos tengan relación entre ellos. Podría decirse que cada individuo en un algoritmo evolutivo clásico sólo tiene “conciencia” del entorno (el medio externo al cual debe adaptarse, y que proporciona una evaluación en forma de aptitud) y nunca entra en contacto directo con el resto de los individuos de la población.

La segunda característica que nos concierne es que la aptitud de un individuo, siempre en un algoritmo evolutivo clásico, es un único valor. Por contra, en muchos problemas del mundo real la bondad de una solución no se mide con un único parámetro, sino como un conjunto de aptitudes, cada una de las cuales corresponde a un objetivo. El concepto de individuo “bueno” se difumina, ya que un individuo puede superar a otro en un objetivo, y el segundo puede ser mejor que el primero en otro de los objetivos.

Estos dos aspectos (relación directa entre individuos y optimización multiobjetivo), no contemplados en los algoritmos evolutivos estándar, se tratarán en profundidad en las dos próximas secciones.

2.1. Coevolución

El término *coevolución* fue acuñado por Ehrlich y Raven en 1964 en su trabajo “Butterflies and plants, a study in coevolution” [Ehrlich y Raven, 1964] sobre las influencias evolutivas recíprocas que tienen las plantas y los insectos que se alimentan de ellas. La idea de la coevolución, sin embargo, puede remontarse incluso a los escritos de Darwin.

En la naturaleza la coevolución se produce cuando las características de dos especies evolucionan de forma conjunta, ya sea de forma simbiótica o de forma “depredador/presa”. Estas dos interacciones son sólo algunas de las posibles relaciones entre especies en un ecosistema (ver tabla 2.1).

En el campo de la computación evolutiva se han adoptado algunas de estas ideas, en particular el parasitismo (o depredación) y el mutualismo. Con ellas se han construido los paradigmas de la *coevolución competitiva* y la *coevolución cooperativa*, que pasamos a estudiar, centrándonos sobre todo en el segundo.

2.1.1. Coevolución competitiva

El paradigma de la coevolución competitiva se basa en la asignación de aptitud a los individuos en función de una relación de competencia entre los mismos.

El algoritmo evolutivo estándar mide la bondad de los individuos con referencia a un marco “externo” a la población (un marco o patrón de medida dado en gran parte por el problema que se está resolviendo). Esto puede plantear algunos problemas:

- Se deben especificar de antemano los parámetros y el método según los cuales se juzgará el progreso de los individuos. En situaciones reales puede ser difícil

Relación	Especie A	Especie B	Descripción
Neutralismo	0	0	Especies independientes. Sin interacción
Mutualismo	+	+	Ambas especies se benefician de la relación
Comensalismo	+	0	Una especie se beneficia. La otra no se ve influida
Competición	-	-	Ambas especies tienen efecto negativo sobre la otra, dado que compiten por los mismos recursos
Depredación	+	-	El depredador (A) se beneficia. La presa (B) se ve afectada negativamente (de hecho, muere)
Parasitismo	+	-	El parásito (A) se beneficia, en tanto que el huésped (B) se ve perjudicado, aunque no desaparece

Cuadro 2.1: Posibles relaciones entre especies en un ecosistema. El signo '+' implica un beneficio para la especie; el signo '-' un perjuicio; '0' indica que la especie no se ve afectada por la relación

identificar estos elementos, de forma que evalúen a los individuos de forma correcta frente a todas las posibles situaciones.

- Los miembros de la población pueden quedar atrapados en mínimos locales de aptitud cuando la función de evaluación es estática.
- La dificultad de la prueba de evaluación no cambia de acuerdo con la aptitud del individuo. Si ésta es demasiado fácil o demasiado difícil para un individuo medio no se puede discriminar de forma correcta el nivel de aptitud.

El método coevolutivo competitivo, en cambio, mide la bondad de un individuo comparándolo con otros individuos. Esto soluciona en gran parte los problemas arriba descritos, a la vez que proporciona un entorno de evaluación más exigente y adaptable. Así, por ejemplo, se puede aplicar a la evolución de jugadores de ajedrez (un campo en el que evaluar la aptitud de un individuo requiere un profundo conocimiento del problema). Se emplearía como medida de la bondad de un jugador el resultado de varios enfrentamientos con otros jugadores.

Las aproximaciones principales a la coevolución son dos:

Población homogénea La población está compuesta por un solo tipo de individuos. La aptitud de éstos se mide a través de enfrentamientos entre ellos. Hay varios métodos para enfrentar individuos. En [Axelrod, 1987] se evolucionan estrategias

para resolver el Dilema del Prisionero enfrentando a cada estrategia (es decir, a cada individuo) con todas las demás de la población (con un coste computacional alto, por supuesto). Otros autores [Angeline y Pollack, 1993] usan un torneo para enfrentar a los individuos dos a dos, de modo que los vencedores se van emparejando entre sí, aunque los perdedores no se extinguen, sino que se reproducen con menor probabilidad.

Población dividida La población se divide en dos especies: una de *huéspedes* y otra de *parásitos*. Los segundos actúan como suministradores de casos de prueba para los primeros. La efectividad de los huéspedes se mide por lo bien que solucionan los problemas “propuestos” por los parásitos, y la de los parásitos se mide por lo bien que detectan debilidades en los huéspedes. Se puede asimilar este proceso tanto a la pedagogía (alumnos/profesores) como al sistema inmune, con oleadas de “epidemias” e “inmunizaciones” que crean una carrera de armamentos (*arms race*, [Juille y Pollack, 1998][Ficici y Pollack, 1998]) cuyo resultado es un constante estado de flujo. Las aplicaciones de este modelo incluyen la ordenación de números [Hillis, 1990], la generación de estrategias para jugar al Nim y al Tres en Raya [Rosin y Belew, 1997] y la coevolución de “profesores” y “alumnos” para resolver problemas de clasificación [Ficici y Pollack, 2001].

2.1.2. Coevolución cooperativa

El otro paradigma relacionado con la coevolución es el de la coevolución cooperativa. Éste, al igual que el de la coevolución competitiva, se caracteriza porque el cálculo de las aptitudes de los individuos se hace en función de la relación que se establece entre ellos. Sin embargo, a diferencia del paradigma anterior, las relaciones que se buscan en la coevolución cooperativa son de colaboración (o simbiosis) entre individuos.

De hecho, la coevolución cooperativa enfoca el esfuerzo a la obtención de poblaciones de individuos que deben cooperar para obtener una solución válida. Esto es, cada individuo es sólo una solución parcial al problema. Es por tanto necesaria la cooperación entre varios individuos para obtener una solución global. Una excelente introducción al modelo coevolutivo cooperativo se puede encontrar en [Potter y de Jong, 2000].

En un algoritmo coevolutivo cooperativo el objetivo de un individuo es optimizar una parte de la solución y cooperar con los demás para obtener una solución global. La hipótesis (y la ventaja de este paradigma) es que varias búsquedas paralelas de soluciones parciales son más eficientes que una única búsqueda de una solución global.

Otra ventaja adicional es que en caso de que se mantengan las diferentes partes de la solución en una única población, dicha población no converge a un único individuo. Si así ocurriera, la aptitud de este individuo sería necesariamente muy baja, ya que no podría cooperar con otros miembros de la población distintos a sí mismo. La diversidad se mantiene, forzada por la necesidad de cooperar, y el algoritmo evolutivo puede utilizar sus operadores de recombinación a lo largo de la evolución.

La diversidad es uno de los elementos claves en la evolución. El proceso continuo de seleccionar y reproducir los mejores individuos de la población provoca que las poblaciones pierdan diversidad y acaben convergiendo alrededor de un único tipo de individuo [Goldberg, 1989]. Esta convergencia no es deseable por dos motivos:

1. Las poblaciones suelen converger a soluciones subóptimas.
2. Las poblaciones convergidas no se pueden adaptar a cambios en el entorno de trabajo.

Un algoritmo evolutivo, cuando su población ha convergido, sólo puede continuar introduciendo una mutación aleatoria en la muestra. Esto produce una búsqueda lenta e ineficiente. El mantenimiento de poblaciones con diversidad es muy difícil y es uno de los campos de investigación abiertos en la comunidad científica. Los métodos aplicados hasta el momento para mantener la diversidad en la población son los siguientes:

Estrategias de selección poco agresivas. Es el método más común. Sin embargo, aunque estas estrategias no aseguran la diversidad, si producen una evolución más lenta. Este método, al menos, evita convergencias prematuras de la población a subóptimos locales muy alejados del valor óptimo.

Incremento del ratio de mutación. Este método inyecta artificialmente diversidad en la muestra mediante la adición de ruido. Como el anterior, suele funcionar aceptablemente en casos reales.

Compartición de la aptitud. Propuesto por D. Goldberg y J. Richardson como uno de los primeros mecanismos para mantener la diversidad en la población de individuos [Goldberg y Richardson, 1987]. Los individuos similares comparten parte de su aptitud, decrementando el valor de la aptitud de cada uno de ellos. De esta forma la selección actúa contra los nichos superpoblados. El problema radica en que el cálculo de la compartición implica $O(n^2)$ operaciones de comparación, donde n es el tamaño de la población. En poblaciones grandes con cromosomas grandes los métodos basados en la comparación no son operativos [Smith et al., 1993]. Además, este método no es aplicable a la programación evolutiva, en la cual es muy difícil la definición de criterios de similaridad.

Crowding. Propuesto en [de Jong, 1975]. Este método sigue la definición del algoritmo genético estándar excepto en que sólo una fracción de la población se reproduce y muere en cada generación [Mahfoud, 1992]. Se elige un porcentaje de la población, representado por la *distancia generacional*, mediante un método proporcional a la aptitud, para sufrir el proceso de cruce y mutación. Siendo n el número de individuos de la población, se seleccionan Gn individuos para ser reemplazados por los elegidos para reproducción. Para insertar un nuevo individuo en la población se toman aleatoriamente CF^1 individuos de la población. El nuevo individuo reemplaza al individuo que es más similar a él de los CF elegidos.

¹ *Crowding factor*.

Apareamiento local. Propuesto en [Collins y Jefferson, 1991]. Este algoritmo consiste en considerar que los individuos se encuentran en un espacio euclídeo y sólo se combinan con sus vecinos cercanos. En su formulación original se sitúan los individuos en una rejilla toroidal. Los individuos sólo se cruzan con aquellos otros individuos que se encuentran en su vecindad en la rejilla. Otras muchas ideas similares han sido definidas, todas basadas en la idea de sólo realizar apareamientos locales y no globales.

Compartición implícita de la aptitud. Inicialmente propuesto en [Horn et al., 1994] y en [Smith et al., 1993]. En este caso no se hacen comparaciones entre los individuos. En su lugar, la diversidad se asegura mediante el *comportamiento cooperativo* de los individuos de la población en la formación de la solución. Cada individuo representa una solución parcial y necesita combinarse con los demás para resolver el problema completo. Reduciendo la capacidad de los individuos y coevolucionándolos juntos la evolución busca diferentes tipos de individuos que juntos resuelvan el problema. Además, aparece un efecto secundario muy útil, durante la búsqueda de individuos que optimicen aspectos parciales de la solución se realiza una búsqueda paralela en descomposiciones parciales del espacio solución lo que provoca una búsqueda mucho más rápida.

Por tanto, la coevolución cooperativa tiene dos ventajas importantes. Por un lado, la especialización de las unidades asegura la diversidad y evita la convergencia de la población. Una única unidad no puede acaparar toda la población dado que para alcanzar un valor de aptitud alto es necesaria la presencia de otras especializaciones. Si una especialización se hace muy preeminente, sus miembros no se combinarán siempre con otras especializaciones de la población. Así, soluciones parciales redundantes no reciben siempre el beneficio de otras especializaciones e incurrirán en valores de aptitud bajos. La presión evolutiva está contra los miembros que tienden a monopolizar la población. Por otro lado, la búsqueda se ve acelerada por la paralelización implícita que supone descomponer el problema en subproblemas.

Terminología y notación

En este paradigma, por tanto, se tienen en cuenta dos entidades diferentes: aquellas que representan soluciones parciales al problema y aquellas otras que, al formarse por unión de las anteriores, representan soluciones completas al mismo. Diferentes autores denominan a estas entidades de varias formas. Así, Potter [Potter, 1997] habla de *individuos* que se unen para formar *colaboraciones*; Zhao [Zhao, 1998] prefiere los términos *módulo* (para los subproblemas) y *organización* (para las uniones de módulos); Moriarty evoluciona redes neuronales usando una población de neuronas (soluciones parciales) que combina de forma temporal [Moriarty y Miikkulainen, 1996] o mediante otra población de combinaciones de neuronas (a las que llama *blueprints*) [Moriarty y Miikkulainen, 1998b]; García-Pedrajas [García-Pedrajas et al., 2003] usa una estructura similar, evolucionando grupos de neuronas en lugar de neuronas individuales, y llamando *nódulos* a los grupos de neuronas que se unen para formar *redes*.

En nuestro estudio, y por homogeneidad con el método propuesto en el capítulo 3, usaremos los términos *subcomponente* e *individuo* para designar respectivamente a las soluciones parciales y a las uniones de éstas (soluciones completas). En alguna ocasión, durante la discusión que sigue, usaremos alguno de los términos alternativos descritos en el párrafo anterior, en caso de que nos estemos refiriendo al trabajo de algún autor en particular.

La notación que se empleará considerará que existen varias especies ², una o varias de subcomponentes a las que llamaremos PS_i ($1 \leq i \leq K$) y una de individuos, llamada PI . El tamaño de cada población podrá indicarse bien mediante una constante (N_p ($1 \leq p \leq K$) y M , respectivamente) o bien mediante el operador estándar ($|PS_i|$ y $|PI|$). Cuando nos refiramos a los componentes genéricos de dichas poblaciones usaremos la notación $sc_{p,q} \in PS_p$ ($1 \leq p \leq K, 1 \leq q \leq N_p$) e $i_j \in PI$ ($1 \leq j \leq M$).

Trataremos a continuación, previamente a estudiar un modelo genérico, los aspectos a tener en cuenta cuando se estudia la coevolución cooperativa.

División del problema

Una de las premisas fundamentales de la coevolución cooperativa es la división del problema que se trata de resolver en subproblemas, de modo que éstos puedan ser resueltos por los subcomponentes.

Si un problema puede dividirse en subcomponentes y éstos son independientes entre sí, cada uno puede ser optimizado o resuelto de forma independiente. Cada subcomponente evolucionaría en su propia superficie de aptitud (*fitness landscape*, la distribución de la aptitud sobre el espacio genotípico o fenotípico). Muchos problemas sólo pueden ser descompuestos en subcomponentes con interrelación entre ellos. Estas interrelaciones hacen que al modificar uno de los subcomponentes, las superficies de aptitud de todos los demás se vean deformadas [Kauffman y Johnsen, 1992].

Por tanto, sobre la división del problema influyen principalmente dos factores: la descomponibilidad del problema y las dependencias que puedan existir entre los subcomponentes del problema. Estas dependencias se suelen conocer en el campo de la genética como *interacciones epistáticas* y se producen cuando el valor de un gen influye sobre la aptitud del individuo de forma diferente según qué valor tenga otro gen. La epístasis puede producirse entre genes de un mismo individuo o, en el caso que nos ocupa, entre genes de distintas especies (los subcomponentes, en nuestro estudio).

La división implica, a bajo nivel, una fragmentación del genoma del individuo y una distribución de esos fragmentos entre los subcomponentes para que se coevolucionen por separado.

Una vez escogida la representación que se va a hacer del dominio del problema, es importante [Wiegand et al., 2002] que al dividir el problema se intente que el número de

²No todos los autores consideran que la unión de los subcomponentes sea en sí una especie. Para algunos es una unión temporal con el fin de crear una entidad evaluable. Consideraremos, al igual que Zhao (cuyo modelo se describe más adelante, en la página 16) que sí existen dos especies, y esta suposición se puede particularizar fácilmente.

subcomponentes (fragmentos del genotipo) sea el correcto ³ y que a cada subcomponente se le asigne el fragmento correcto del genotipo ⁴ para evitar en lo posible la epístasis interpoblacional.

El número de subcomponentes puede ser fijo (decidido previamente a la ejecución del algoritmo evolutivo) o variable. En este segundo caso (llamado por Potter [Potter, 1997] *descomposición emergente*) se puede monitorizar la colaboración entre subcomponentes y añadir o quitar especies (poblaciones de subcomponentes) según sea necesario. Obviamente, este método de división en número variable de subproblemas presenta dos inconvenientes: su complejidad y su adecuación sólo a determinados tipos de problemas.

Los subcomponentes del problema pueden ser homogéneos (todos ellos tienen un fenotipo igual o muy parecido) o heterogéneos. Esto concuerda con los tipos de división entre especies que se pueden dar en la coevolución competitiva (tal como se trató en la página 7. Un ejemplo de subcomponentes homogéneos son las neuronas del método SANE [Moriarty y Miikkulainen, 1998a]. No hay muchos casos de subcomponentes heterogéneos. En este trabajo se propone un problema de coevolución de superficies de respuesta (capítulo 6) que usa subcomponentes distintos. Otras aplicaciones están en el campo de los sistemas multiagente, donde cada agente tiene un objetivo distinto, y posiblemente una representación distinta, y todos deben cooperar [Wooldridge, 1999][Matsubara et al., 1996].

Los problemas que abordaremos en este trabajo serán divididos en un número fijo de subcomponentes, que en cada problema será representado por la constante K (ver apartado de notación).

Evolución de los subcomponentes

Una vez decidida la subdivisión del problema, se debe evolucionar cada subcomponente por separado, buscando especializarlo de modo que resuelva lo mejor posible su subproblema.

Para evolucionar cada subcomponente se usa un algoritmo evolutivo, lo que implica, por supuesto, la creación de una población inicial y la aplicación de operadores genéticos. Pero la diferencia con un algoritmo evolutivo estándar es que en coevolución cooperativa se crean implícitamente varias especies, una para cada componente, y se deben mantener separadas una de otra⁵.

Otro requisito, además de la separación entre las especies, es el mantenimiento de la diversidad de las mismas, puesto que la convergencia de subcomponentes de diferentes especies hacia un mismo genotipo empobrecería la mezcla de los mismos, y rompería el objetivo de que cada subcomponente se especialice en una parte del problema.

Las posibilidades para cumplir estos dos requisitos son varias:

³*decompositional bias* para Wiegand

⁴*linkage bias*

⁵las especies se mantienen separadas en lo que respecta a los operadores genéticos de reproducción, cruce y mutación; para la evaluación de los subcomponentes se unirán ejemplares de cada especie (subcomponentes) para formar un individuo, tal como se verá en los apartados siguientes

- Se puede usar una única población de subcomponentes, en la cual se almacenen las distintas especies. Debe implementarse un sistema que evite el cruce entre especies o, alternativamente, se debe potenciar la diversidad, de modo que se mantenga la especialización. Se pueden, por ejemplo, usar técnicas de apareamiento restringido (*restricted mating*) y compartición de aptitud (*sharing*)⁶. Una posibilidad adicional es usar etiquetas (*tags*) para clasificar a los individuos en una u otra especie dentro de la misma población [Spears, 1994] e incluso “migrarlos” entre ellas usando una técnica llamada *difusión* [Wiegand, 1998]. Este método dificulta mucho el uso de subcomponentes heterogéneos, que no pueden coexistir en la misma población.
- Como alternativa, se pueden evolucionar los subcomponentes en múltiples poblaciones, una para cada especie. Con esta solución se resuelve el problema de la separación de especies⁷ y el requisito de mantenimiento de la diversidad se reduce a una necesidad de evitar la convergencia prematura a una solución subóptima (algo siempre deseable en cualquier tipo de algoritmo evolutivo). El uso de varias poblaciones permite usar subcomponentes heterogéneos fácilmente, e incluso evolucionarlos con diferentes paradigmas de algoritmos evolutivos. Por último, la paralelización de un algoritmo que use múltiples poblaciones es más sencilla y directa.

En este trabajo nos orientaremos al uso de múltiples poblaciones, lo que facilita la segregación de las especies, y es especialmente útil para alguno de los problemas que se abordarán, compuestos por subcomponentes heterogéneos. Cada población de subcomponentes se identificará como PS_i , siendo $1 \leq i \leq K$. El número de miembros de cada población puede ser igual para todas o diferente para cada una de ellas. En nuestros experimentos usaremos poblaciones de subcomponentes de igual tamaño:

$$|PS_i| = |PS_j|, \forall i, j \in [1, K]$$

Mezcla de los subcomponentes

La evolución de los subcomponentes, tanto en una población como en múltiples poblaciones, trata de lograr una optimización independiente de cada una de las partes en que se ha subdividido el problema. Pero para resolver el problema, está claro, se necesita una unión de subcomponentes. Durante la evolución, los subcomponentes se agrupan para formar individuos que serán evaluados y posteriormente esa evaluación será transferida en forma de crédito a los subcomponentes que han participado en el individuo. Estos dos aspectos se tratarán en los siguientes puntos.

El objetivo de este paso es, pues, unir varios subcomponentes para formar un individuo que pueda ser evaluado, proporcionando además un “contexto” para que los propios subcomponentes sean también evaluados.

⁶De hecho, Goldberg [Goldberg y Richardson, 1987] ya las usó para crear varias especies en una misma población, sobre un problema de optimización

⁷Realmente, las poblaciones separadas, en ese aspecto, son un método de apareamiento restringido llevado al extremo.

Recordemos que las mezclas de subcomponentes reciben distintos nombres en función de los autores (organizaciones para Zhao, “blueprints” para Moriarty, colaboraciones para Potter. . .) y que en este trabajo agruparemos esos términos bajo la denominación “individuos”.

En este aspecto se plantea una pregunta: ¿cómo elegir qué subcomponentes se mezclarán para formar un individuo?. Wiegand [Wiegand et al., 2001] estudia los métodos de selección de subcomponentes. Los considera como selecciones de colaboradores para asociar a un subcomponente fijo, que es el que se desea evaluar. Propone tres atributos para definir estos métodos:

Presión selectiva (*greediness*). Determina el tipo de subcomponentes que seleccionará el método. Para cada especie de subcomponentes, se puede elegir el mejor representante de su población⁸, el mejor y otro al azar, el mejor y el peor, varios al azar, uno o varios seleccionados por torneo. . . Como puede verse, el número de representantes que se toman de cada población de subcomponentes puede variar.

Número de colaboradores (*pool size*). Este atributo indica el número de representantes que se tomarán de cada población de subcomponentes. Cuando es superior a uno, la complejidad computacional del método se incrementa, pero a cambio la evaluación de los subcomponentes es más completa y fiable. Wiegand confirma que se mejoran los resultados si se incrementa este parámetro, y usa valores entre 1 y 5. Algunos autores [Hillis, 1990] llevan este parámetro al extremo ($|PS_i|$) y evalúan todas las posibles colaboraciones entre subcomponentes.

Asignación de crédito. Cuando el número de colaboradores es mayor que uno se generarán varias colaboraciones y cada una de ellas representará a un individuo completo que obtendrá su aptitud (como se verá en el apartado siguiente). Estas aptitudes deben ser repartidas entre los subcomponentes que forman los individuos que las han obtenido. En este reparto se deben tener en cuenta dos importantes cuestiones: la proporción de la aptitud del individuo que se asigna a cada subcomponente (el total, el total dividido entre el número de subcomponentes, una parte proporcional a la importancia del subcomponente⁹. . .) y la forma en que se asignan a un subcomponente los créditos de varios individuos en los que ha participado (crédito medio, mejor crédito, peor crédito. . .). El problema de la asignación de crédito es central en el campo de la coevolución cooperativa y se estudiará en un próximo apartado.

Los métodos de selección que se están tratando, por tanto, tienen como misión elegir uno o varios subcomponentes de cada población PS_i para formar una o varias colaboraciones temporales (individuos) que puedan ser evaluados. Estos individuos se disgregarán inmediatamente después de la evaluación, y no se mantendrá memoria de su composición. Su única misión es proporcionar un entorno de evaluación a los subcomponentes.

⁸El mejor según las evaluaciones realizadas en la anterior generación. En caso de que se trate de la primera selección, se deberá elegir otro método

⁹Queda, obviamente, el problema de medir la importancia del subcomponente

Al final de la evolución habrá de que seleccionar de entre las últimas poblaciones de subcomponentes a los que formarán el individuo resultante del proceso.

Lo que acabamos de describir es una coevolución cooperativa con múltiples poblaciones de subcomponentes y *un único nivel de evolución*. Algunos autores [Moriarty y Miikkulainen, 1998b] [García-Pedrajas et al., 2002] usan un esquema de evolución en dos niveles: las colaboraciones de subcomponentes no son efímeras, sino que están determinadas por una *población de individuos (PI)* que se evoluciona a la vez que las poblaciones de subcomponentes (PS_i). Cada integrante de la población de individuos es una agrupación de K subcomponentes, uno de cada población PS_i . La población de individuos se inicia con combinaciones aleatorias que luego van evolucionando según un algoritmo evolutivo estándar.

Este esquema de coevolución en dos niveles ¹⁰ permite que un mismo subcomponente participe en varios individuos (el equivalente a un número de colaboradores variable, según los parámetros de Wiegand vistos arriba) y por tanto sea evaluado de forma más completa. Asimismo la evolución de la población de individuos puede muestrear con más probabilidad a los subcomponentes que muestren mejor comportamiento, potenciándolos.

Este trabajo usará una evolución en dos niveles, con un conjunto de K poblaciones de subcomponentes ($PS_1 \dots PS_K$) y una población de individuos PI .

Evaluación de los individuos

Como ya se ha apuntado previamente, en coevolución cooperativa la aptitud de un individuo es función de la interacción entre varios subcomponentes. En los apartados anteriores hemos estudiado la división de un problema en subproblemas y la asignación de cada subproblema a una población de subcomponentes, de la evolución por separado de los subcomponentes y de la unión de los mismos para formar un individuo.

Una vez unidos varios subcomponentes para formar el individuo, éste debe ser evaluado como paso previo a la asignación de crédito a los subcomponentes que lo forman. Esta evaluación de los individuos se hará normalmente mediante una aplicación del individuo (solución) al problema y una valoración de su aptitud. Sin embargo, nada impide que la valoración de los individuos se haga de forma competitiva, usando los métodos descritos cuando se trató la coevolución competitiva. Otra posibilidad es que el individuo deba optimizar una serie de objetivos simultáneamente, con lo que la evaluación del individuo en el ámbito del algoritmo evolutivo se realizaría según uno de los métodos que se verán en la sección 2.2.

En algunos casos podría ser necesario evaluar un individuo incompleto, por ejemplo para comprobar la importancia de un subcomponente (tal como se verá en el próximo apartado). Esto no siempre es posible, dependiendo del problema que se esté resolviendo. En algunos de los problemas que se han abordado en este trabajo se usa esta evaluación parcial.

¹⁰Que podría generalizarse a más niveles, como propone Zhao en su “hierarchical society model” [Zhao, 1998].

Evaluación de los subcomponentes

En el apartado anterior se obtuvo la aptitud de un individuo. Esta aptitud se debe a la mezcla de los subcomponentes que lo forman. Pero es lógico pensar que no todos los subcomponentes que han participado han aportado lo mismo a esa aptitud. Algunos pueden haber resuelto su subproblema mejor que otros. Un individuo, por ejemplo, puede constar de ocho subcomponentes de los cuales uno es bastante bueno (resuelve su subproblema muy bien), cinco son aceptables y dos son malos. Otro individuo podría constar de ocho subcomponentes aceptables. Las aptitudes de ambos individuos podrían ser iguales o muy parecidas, pero está claro que los subcomponentes del primer individuo deberían obtener diferentes valoraciones, y que los del segundo deberían obtener valoraciones iguales.

No se puede, por tanto, asignar directamente la aptitud del individuo (ya sea completa o dividida en partes iguales) a cada uno de los subcomponentes que participan en él. Aunque se hiciera así, también debe considerarse que un subcomponente puede participar en más de un individuo (dependiendo del número de colaboradores, tenemos un único nivel de evolución, o de la población de individuos si se evoluciona en dos niveles, tal como se vio en el apartado anterior).

Cuando se habla de la aptitud de un subcomponente se suele usar el término *crédito*. Se puede considerar, por tanto, que un subcomponente recibe como crédito *parte* de la aptitud de *cada* individuo en el que participe. Se debe determinar qué parte de la aptitud recibe el subcomponente y cómo se mezclan varios créditos procedentes de varias colaboraciones. En el capítulo 3 se propondrá un margo general para describir los distintos tipos de asignación de crédito.

Este problema de la asignación de crédito (*credit assignment problem*) es recurrente en el campo de la coevolución cooperativa [Zhao et al., 2000] y se han probado métodos que asignan la aptitud completa de la mejor colaboración [Potter, 1997], otros que usan la media de las colaboraciones [Wiegand et al., 2001] y otros que utilizan una asignación crédito multiobjetivo [García-Pedrajas et al., 2001]: un subcomponente recibe varios créditos, cada uno de los cuales es función de su papel en una o varias colaboraciones, y dichos créditos se consideran objetivos a optimizar de forma conjunta. Esta última aproximación permite mantener una diversidad interesante en las poblaciones de subcomponentes y se empleará en este trabajo.

Modelo general

Zhao [Zhao, 1998] propone un modelo general de coevolución cooperativa bastante adaptable a cualquier sistema y que recoge gran parte de las posibilidades que se han discutido en los apartados anteriores: el modelo de sociedad (*society model*).

El modelo (ver figura 2.1) consta de los siguientes elementos:

- Un conjunto P de poblaciones denominadas P_1, P_2, \dots, P_n que se encargan de evolucionar subcomponentes. Cada población $P_i \in P$ contiene

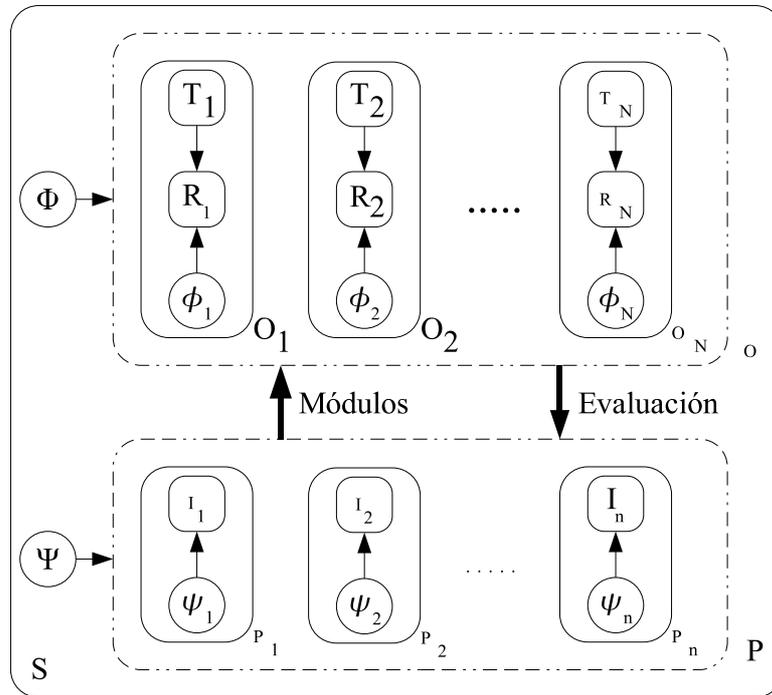


Figura 2.1: Modelo general propuesto por Zhao para la coevolución cooperativa

- Un conjunto de individuos (en la terminología de Zhao, subcomponentes en la terminología que se está usando en esta memoria) I_i . Cada uno de ellos es un candidato a ser representante de esta población en una organización (concepto que se describe a continuación)
- Un algoritmo evolutivo ψ_i (Zhao lo denomina “función”) que evoluciona el conjunto I_i .
- Un algoritmo Ψ para la evolución de P que puede añadir, modificar o eliminar poblaciones del conjunto P . No confundir con los algoritmos ψ_i que evolucionan las poblaciones P_i .
- Un conjunto O de organizaciones (O_1, O_2, \dots, O_N) que agrupan elementos de las poblaciones P para formar soluciones completas al problema. Cada organización $O_j \in O$ está a su vez compuesta por
 - Un conjunto T_j de tareas a realizar, que representan los datos del problema que se está resolviendo (datos de entrenamiento y salidas esperadas si se trata de un problema de aproximación de funciones, especificaciones si se trata de un problema de diseño...).
 - Un conjunto R_j de representantes escogidos de las poblaciones P . Aunque Zhao no lo especifica (y eso hace el modelo más genérico), lo normal es que

en R_j haya un único representante de cada una de las n poblaciones de subcomponentes: $R_j = \{r_1, r_2, \dots, r_n\} | r_k \in I_k$.

- Una función ϕ_j de construcción, que une los representantes de las poblaciones P para formar una organización (individuo en nuestra terminología). La función ϕ_j incluye la estructura general de la organización (el “esqueleto”), una serie de punteros a individuos de las poblaciones P y una regla de aprendizaje ζ_j que se aplica a la organización una vez construida y que puede afinar su aptitud.
- Un algoritmo Φ para la evolución de O que permite modificar las organizaciones existentes, crear nuevas organizaciones...

La interacción entre los componentes del modelo es la siguiente:

Las poblaciones $P_i (1 \leq i \leq n)$ evolucionan por separado, cada una según su propio algoritmo ψ_i que actúa sobre los individuos del conjunto I_i . Los algoritmos ψ_i pueden ser iguales o diferentes (diferentes parámetros, diferentes operadores genéticos, diferente tipo de evolución...). Los conjuntos de individuos I_i pueden ser también homogéneos o heterogéneos (ver discusión en página 2.1.2) y su tamaño ($|I_i|$) puede variar de uno a otro.

El algoritmo Ψ es un algoritmo evolutivo de nivel superior que actúa sobre el conjunto P creando nuevas poblaciones P_i cuando la evolución se estanque, por ejemplo [Potter, 1997], o modificándolas o eliminándolas cuando no contribuyan lo suficiente. Por lo tanto, el número n de poblaciones contenidas en P puede variar.

Cada organización $O_j (1 \leq j \leq N)$ tiene como misión crear un individuo completo a partir de una serie de representantes tomados de las poblaciones de P . Si $N = 1$ se tiene un único individuo y por tanto la evolución se realiza en un único nivel. Si $N > 1$ se realiza una coevolución en dos niveles (ver discusión en página 12).

El funcionamiento de las organizaciones O_j es el siguiente: la función de construcción ϕ_j selecciona una serie de elementos de las poblaciones P_i . Con ellos construye el conjunto de representantes R_j , que formarán el individuo. A continuación somete ese individuo a un proceso de aprendizaje ζ_j (que puede ser un enfriamiento simulado, un entrenamiento por retropropagación o cualquier otro método de optimización local o ajuste fino que se adapte al tipo de individuo que ha sido creado). Por último, evalúa el individuo resultante aplicándolo al problema descrito por el conjunto de tareas T_j .

Según se trató en la página 13, cuando se estudiaron los métodos de mezcla de subcomponentes, si se tiene un único nivel de evolución ($N = 1$ en el modelo de Zhao) la selección realizada por ϕ_j será efímera y únicamente a efectos de evaluación de los subcomponentes. Si en cambio se tiene $N > 1$ (una coevolución en dos niveles), cada organización representará un individuo, y el algoritmo Φ permitirá la evolución de éstos.

La información que se transfiere en el modelo de sociedad entre O y P está compuesta por los módulos que las poblaciones P_i envían como representantes y las evaluaciones de dichos módulos que se envían de vuelta por parte de las organizaciones O_j . En el

modelo no se proporciona un marco genérico para la evaluación de los módulos, sino que en las aplicaciones (sobre todo evolución de redes neuronales) Zhao usa una asignación de crédito que otorga a cada módulo la suma de las aptitudes de los individuos en los que participa.

El modelo que hemos descrito es un modelo general que se puede particularizar a los modelos propuestos en la bibliografía, que ya se han citado durante el estudio de los apartados anteriores:

- El modelo de Potter [Potter, 1997] usa múltiples poblaciones de módulos ($n > 1$) y un único organizador ($N = 1$). El organizador ϕ_1 se encarga de evaluar cada módulo $a \in I_i$ mezclándolo con los mejores representantes del resto de las poblaciones ($b \in I_k | k \neq i$).
- Potter hace también un estudio de descomposición emergente en el que el número de módulos varía: se crean nuevas poblaciones de módulos cuando se estanca la evolución y se destruyen cuando no contribuyen lo suficiente a la solución del problema global. En este método toma parte el algoritmo Ψ , que evoluciona el conjunto de poblaciones P .
- Moriarty [Moriarty y Miikkulainen, 1996] en su modelo SANE básico usa una única población de módulos ($n = 1$) y los agrupa de forma temporal para evaluarlos. Una vez más el único organizador ($N = 1$) evalúa múltiples combinaciones de módulos.
- En trabajos posteriores [Moriarty y Miikkulainen, 1998b] Moriarty introduce el modelo SANE jerárquico, que aún utiliza una población de módulos, pero añade un segundo nivel de evolución en el las $N > 1$ organizaciones (*blueprints*) evolucionan (a través del algoritmo evolutivo Φ).
- García-Pedrajas [García-Pedrajas et al., 2003] usa una estructura más completa, mezcla de las anteriores, con múltiples poblaciones de módulos ($n > 1$) y dos niveles de evolución ($N > 1$). La población de organizaciones evoluciona (algoritmo Φ) pero el número y tipo de módulos permanece fijo (no se usa el algoritmo Ψ). Obviamente, como en todos los métodos anteriores, sí se usan los algoritmos ψ_i .

A lo largo de esta discusión se ha usado la notación propuesta por Zhao en su modelo. En el capítulo siguiente se propondrá un nuevo modelo general de coevolución cooperativa, en el que se usará una notación propia de este trabajo. En la figura 2.2 podemos ver una adaptación del modelo de Zhao al propuesto, usando la nueva notación. Una figura más explicativa y completa se mostrará en el capítulo 3 (figura 3.1).

Puede verse en la figura que la población de organizaciones O (llamada población de individuos PI en el modelo propuesto) consta de una serie de individuos i_1, i_2, \dots, i_M que están compuestos por grupos de referencias a subcomponentes ($R_j, 1 \leq j \leq M$). Estos subcomponentes son agrupados para formar el individuo mediante un método equivalente a la función ϕ_j de Zhao, pero que en la figura denominamos ϕ , al ser

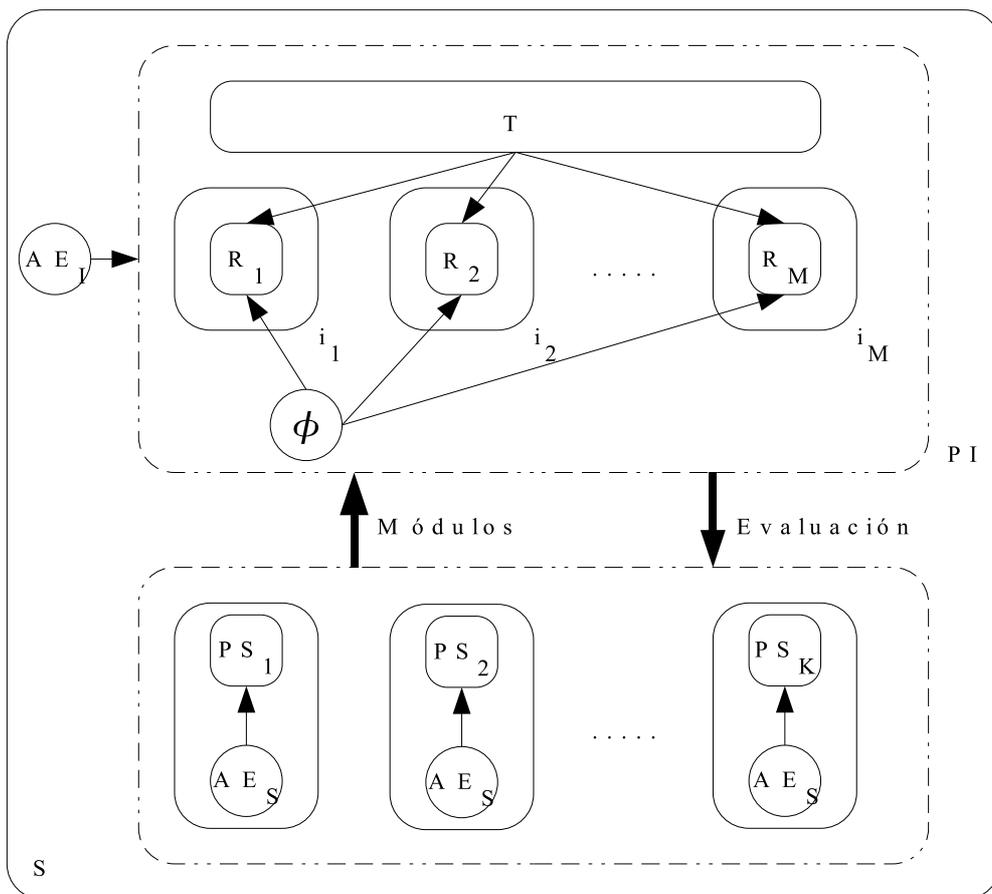


Figura 2.2: Notación que se usará en el modelo propuesto

idéntico para todos los individuos. Los individuos, para ser evaluados, serán aplicados a los datos del problema (T), compartidos entre todos ellos.

La población de individuos PI es evolucionada por un algoritmo evolutivo propio denominado AE_I , equivalente a la función Φ de Zhao.

En cuanto a las poblaciones de subcomponentes (PS_1, PS_2, \dots, PS_K), cada una de ellas está formada por un conjunto de subcomponentes ($sc_{p,q}, 1 \leq p \leq K, 1 \leq q \leq N_K$) y es evolucionada por un algoritmo evolutivo AE_S (semejante a la función ψ_i del modelo de Zhao). No se contempla la posibilidad de añadir o eliminar poblaciones de subcomponentes, por lo que no existe en el modelo propuesto un equivalente de la función Ψ .

El modelo propuesto será objeto de estudio en el capítulo 3, donde se detallará su funcionamiento y estructura, y en el capítulo 4, donde se describirá la implementación que se ha hecho del mismo.

2.2. Optimización Multiobjetivo

2.2.1. Definición del problema

Un problema de optimización multiobjetivo busca optimizar (minimizar o maximizar) de forma simultánea una serie de funciones objetivo. Cada una de estas funciones ($f_i(x), i = 1, 2, \dots, n$) toma como argumento un vector perteneciente al llamado *espacio de parámetros*, X , y genera como resultado un valor y_i . El vector formado por los valores generados por las funciones $f_i(x)$, es decir, $y = \{y_1, y_2, \dots, y_n\}$, es un vector perteneciente al *espacio de objetivos* Y . El vector x recibe comúnmente el nombre de *vector de decisión* y el vector y el de *vector objetivo*.

El espacio de parámetros puede estar delimitado por una serie de restricciones, que de forma general se pueden representar como un grupo de k funciones $e_i(x), i \in \{1, 2, \dots, k\}$ que deben ser menores o iguales que cero en los puntos del espacio de parámetros reducido. Este espacio de parámetros reducido suele denominarse *espacio factible* (*feasible set* en la bibliografía en Inglés).

Formalmente, pues, y sin pérdida de generalidad, un problema de optimización multiobjetivo se puede escribir de la siguiente forma:

$$\begin{aligned} \mathbf{min.} \quad & \mathbf{y} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_n(\mathbf{x})) \\ \mathbf{sujeto a} \quad & \mathbf{e}(\mathbf{x}) = (e_1(\mathbf{x}), e_2(\mathbf{x}), \dots, e_k(\mathbf{x})) \leq 0 \end{aligned}$$

donde

$$\begin{aligned} \mathbf{x} &= (x_1, x_2, \dots, x_m) \in X \\ \mathbf{y} &= (y_1, y_2, \dots, y_n) \in Y \end{aligned}$$

En un problema de optimización normal (con un único objetivo), se busca encontrar un valor \mathbf{x}^* de la variable de decisión que proporcione un resultado mínimo de la función

objetivo. El valor \mathbf{x}^* puede ser único o, en algunos casos, puede que existan varias soluciones al problema (si existen varios mínimos globales de igual profundidad). En este segundo supuesto todas las soluciones generan el mismo valor de la función objetivo. Al responsable de la resolución del problema (a veces llamado *decision maker* en la bibliografía) solamente le interesa un solución, y si hay varias no tiene ninguna preferencia por una u otra.

Sin embargo, en un problema de optimización multiobjetivo se busca una solución de compromiso que, aunque no proporcione valores óptimos para todas las funciones objetivo, sí los proporcione aceptables de forma general, es decir, considerando todas las componentes del vector objetivo de forma global.

Lo que se busca en el caso de la optimización multiobjetivo, pues, no es una única solución sino un conjunto de soluciones que generen resultados aceptables al aplicarles las funciones objetivo. Será el responsable de la resolución el que deba elegir la solución más apropiada, aplicando su conocimiento del problema. Esta toma de decisiones puede tener lugar no solo al finalizar el proceso de optimización, sino también durante el mismo, seleccionando y eliminando de forma interactiva las soluciones propuestas por el algoritmo de optimización, y guiando por tanto la búsqueda. También puede llevarse a cabo la toma de decisiones antes del inicio del algoritmo, ajustando parámetros de la búsqueda que la orienten hacia uno u otro tipo de soluciones.

En los párrafos anteriores se ha mencionado que las soluciones que se buscan en una optimización multiobjetivo no son óptimas en todas las componentes, sino “aceptables consideradas globalmente”. Esta definición bastante difusa se suele formalizar utilizando los conceptos introducidos por Vilfredo Pareto en el Siglo XIX, que se estudiarán a continuación.

2.2.2. Conceptos

Dominancia o dominación

Suponiendo un problema de minimización, y dados dos vectores \mathbf{a} y \mathbf{b} pertenecientes al espacio de decisión X , se dice que \mathbf{a} domina a \mathbf{b} ($\mathbf{a} \succ \mathbf{b}$), o que es parcialmente menor ($\mathbf{a} p < \mathbf{b}$) si y sólo si todas las componentes del vector objetivo asociado a \mathbf{a} ($\mathbf{f}(\mathbf{a})$) son menores o iguales que las correspondientes componentes del vector objetivo asociado a \mathbf{b} ($\mathbf{f}(\mathbf{b})$) y al menos una de las primeras es estrictamente menor que su correspondiente segunda:

$$\mathbf{a} \succ \mathbf{b} \Leftrightarrow \begin{cases} \forall i \in \{1, 2, \dots, n\}, f_i(\mathbf{a}) \leq f_i(\mathbf{b}) \\ \wedge \\ \exists j \in \{1, 2, \dots, n\} \mid f_j(\mathbf{a}) < f_j(\mathbf{b}) \end{cases}$$

Así, por ejemplo, dados los vectores \mathbf{a} , \mathbf{b} , \mathbf{c} y \mathbf{d} en el espacio de decisión, si sus correspondientes vectores en el espacio de objetivos son $\mathbf{t} = \mathbf{f}(\mathbf{a}) = (3, 3, 0)$, $\mathbf{u} = \mathbf{f}(\mathbf{b}) = (3, 5, 0)$, $\mathbf{v} = \mathbf{f}(\mathbf{c}) = (0, 2, 1)$ y $\mathbf{w} = \mathbf{f}(\mathbf{d}) = (3, 6, 1)$, se cumple $\mathbf{a} \succ \mathbf{b}$, $\mathbf{a} \succ \mathbf{d}$, $\mathbf{b} \succ \mathbf{d}$ y $\mathbf{c} \succ \mathbf{d}$.

Si consideramos el conjunto $X = \{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}\}$, entonces se puede afirmar que \mathbf{a} y \mathbf{c} no están dominados en X .

La relación de dominación entre vectores crea una ordenación parcial del espacio de decisión, ya que dos puntos cualesquiera \mathbf{a} y $\mathbf{b} \in X$ cumplen

$$\mathbf{a} \succ \mathbf{b} \vee \mathbf{b} \succ \mathbf{a} \vee \mathbf{a} \sim \mathbf{b}$$

Siendo \sim la relación de “indiferencia”, que indica que \mathbf{a} no domina a \mathbf{b} y \mathbf{b} no domina a \mathbf{a} .

Optimalidad Pareto

Dados dos espacios vectoriales X e Y y una función vectorial $\mathbf{f} : X \rightarrow Y$, se dice que un punto $\mathbf{a} \in X$ es Pareto-óptimo si $\mathbf{f}(\mathbf{a})$ no está dominado en X , es decir,

$$\nexists \mathbf{b} \in X, \mathbf{f}(\mathbf{b}) \succ \mathbf{f}(\mathbf{a})$$

O, lo que es igual, $\mathbf{a} \in X$ es Pareto-óptimo si

$$\forall \mathbf{c} \in X, \mathbf{f}(\mathbf{a}) = \mathbf{f}(\mathbf{c}) \vee \exists i \in \{1, 2, \dots, n\}, f_i(\mathbf{a}) > f_i(\mathbf{c})$$

Todos los puntos pertenecientes al espacio de decisión que sean Pareto-óptimos forman el *frente de Pareto*, *superficie de Pareto* o *conjunto Pareto-óptimo*. En el ejemplo presentado anteriormente, el frente de Pareto está compuesto por los puntos \mathbf{a} y \mathbf{c} .

2.2.3. Optimización Evolutiva Multiobjetivo

En su aplicación a la resolución de problemas multiobjetivo, los algoritmos evolutivos presentan una gran ventaja sobre otros algoritmos de búsqueda: al realizar la exploración del espacio factible en paralelo (puesto que podemos considerar a cada individuo como la búsqueda de una solución), un algoritmo evolutivo puede obtener varios puntos del frente de Pareto (es decir, varias soluciones a un problema multiobjetivo) en una sola ejecución.

La aplicación de los algoritmos evolutivos a los problemas de optimización multiobjetivo está fundamentada principalmente en la posibilidad de obtener varias soluciones no dominadas (Pareto-óptimas) en una sola ejecución del algoritmo. Esto se debe a la característica de búsqueda en paralelo sobre la población que presentan este tipo de algoritmos.

Se han propuesto bastantes métodos de optimización multiobjetivo basados en técnicas evolutivas. Algunos de estos métodos se basan en unificar los objetivos para constituir uno sólo, y usar un algoritmo evolutivo estándar para optimizar este único objetivo. Otros métodos hacen uso del concepto de optimalidad Pareto en los operadores genéticos, generalmente buscando un doble objetivo: guiar la evolución de los individuos hacia el frente de Pareto y aplicar una presión selectiva que los distribuya por el mismo.

A continuación se revisarán los más métodos de optimización multiobjetivo más conocidos. Se dividirán en tres tipos: los métodos no basados en Pareto, los métodos basados en Pareto (estos dos tipos pertenecen a las llamadas “técnicas de primera generación”) y los métodos de segunda generación

2.2.4. Métodos no-Pareto

Los métodos de primera generación no basados en Pareto son eficientes y sencillos de implementar, pero sólo son apropiados para pocos objetivos y debido a que no emplean el concepto de optimalidad de Pareto no pueden hallar determinadas zonas de los Frentes de Pareto.

Suma ponderada de Aptitudes

Uno de los métodos más simples para combinar un algoritmo evolutivo con una búsqueda multiobjetivo consiste en acercar la segunda al primero. Un algoritmo evolutivo basa su búsqueda en la función de aptitud de los individuos de su población, y esa función es única. Este método combina las n funciones objetivo, mediante una suma ponderada, en una única función, y ésta se usa como aptitud de los individuos.

Cada vez que se evalúe un individuo \mathbf{a} , se calcula el valor de cada una de las funciones objetivo ($y_i = f_i(\mathbf{a}), i \in \{1, 2, \dots, n\}$) y se agrupan, ponderándolas mediante pesos w_i , en un valor F que constituye la aptitud del individuo \mathbf{a} :

$$F = \sum_{i=1}^n w_i f_i(\mathbf{a})$$

Se suele cumplir que la suma de los pesos w_i es igual a la unidad.

VEGA

Schaffer, para construir su *Vector Evaluated Genetic Algorithm* [Schaffer, 1985], se basa en el algoritmo genético tradicional y modifica la selección de individuos de modo que se tomen los mejores individuos atendiendo a cada criterio de optimización.

Así, suponiendo k criterios y una población con tamaño N , se crearían k grupos de N/k individuos cada uno, seleccionando para cada uno en función de $f_i (i \in \{1, 2, \dots, k\})$. A continuación se unirían estos k grupos y se barajarían para obtener una población intermedia (*mating pool*, para algunos autores), sobre la cual se aplicarían los operadores de mutación y cruce para obtener la nueva población.

La selección de individuos de cada grupo atendiendo a una única función de aptitud hace que el algoritmo tienda a elegir los individuos en los extremos del frente Pareto-óptimo, produciendo por ello una especialización (o especiación) en la población, es decir, propiciando la aparición de individuos que son muy aptos según una de las funciones objetivo, pero no según el resto de las funciones. Aunque la aplicación de los

operadores de cruce y mutación pudieran crear soluciones (individuos) de compromiso, aceptables en todos los objetivos pero no destacadas en ninguno, estos individuos no serían seleccionados.

El propio Schaffer reconoce esta desventaja de su propuesta, y ofrece dos heurísticas para paliar el problema. La primera trata de proteger los individuos intermedios (*mid-dling*) favoreciendo la selección de soluciones no dominadas. La segunda busca generar individuos intermedios sustituyendo la selección para cruce (que en un algoritmo genético tradicional es aleatoria) por otra que favorezca la mezcla entre especies.

2.2.5. Métodos Pareto

Las técnicas basadas en Pareto fueron propuestas por Goldberg para solucionar los problemas presentados por el algoritmo VEGA. Estas técnicas se basan en el uso de la ordenación y selección por dominación para acercar la población al frente de Pareto.

MOGA

Fonseca y Fleming [Fonseca y Fleming, 1993] proponen una clasificación de individuos en función de su rango. El rango de cada individuo de la población es el número de congéneres que lo dominan más uno. Si un individuo \mathbf{x} es dominado por l individuos de la población, entonces su rango $r(\mathbf{x})$ será $l + 1$. Aquellos individuos de la población que no estén dominados (pertenecientes, por tanto, al frente de Pareto) tendrán rango 1. El peor individuo posible tendrá rango N (siendo N el tamaño de la población), ya que estará dominado por el resto de los individuos de la población, es decir, por $N - 1$ individuos.

Una vez calculado el rango de cada individuo, se ordena la población de acuerdo con los rangos y se asignan las aptitudes a los individuos usando una función de interpolación que mapeará rangos (en el intervalo 1 a N) a aptitudes (en el intervalo 0 a 1, por ejemplo, en el caso de aptitud normalizada). Fonseca y Fleming toman una función lineal para este mapeado, pero podría utilizarse otro tipo.

La aptitud así calculada no es la definitiva. Como último paso del algoritmo, se distribuye el valor de aptitud anteriormente calculado (como función del rango) entre todos los individuos que tengan ese rango.

NSGA

Srinivas y Deb [Srinivas y Deb, 1995] usan una clasificación en varias capas para su Algoritmo Genético de Ordenación por No-dominación (*Non-dominated Sorting GA*).

En primer lugar se toman todos los individuos de la población actual que no estén dominados. A todos ellos se les asigna un valor de aptitud predeterminado. A continuación, para obtener la aptitud final, se realiza una compartición (*sharing*) de esos valores, reduciendo la aptitud de cada individuo de forma proporcional al número de individuos en su mismo nicho, es decir, próximos a él (ver más adelante).

Una vez asignada la aptitud definitiva a todos los individuos no dominados, se repite el mismo procedimiento considerando el resto de la población y tomando como aptitud “falsa” (sujeta a posterior compartición) un valor menor que la menor aptitud final obtenida en el paso anterior. Se siguen realizando asignaciones y comparticiones de aptitud a grupos no dominados hasta que no quedan más individuos por clasificar.

Una vez obtenidos los valores de aptitud para todos los individuos de la población se pasa a las fases siguientes del algoritmo genético, en las cuales se usa como método de selección el de resto proporcional estocástico (selección por ruleta), que potencia la elección de los individuos más aptos, pero hace posible la selección de otros peores.

Srinivas y Deb realizan la compartición de aptitud anteriormente mencionada en el espacio de parámetros. Para ello definen una función $Sh(d_{ij})$ que basa su cálculo en la distancia entre los individuos i y j (d_{ij}).

$$Sh(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma_{share}}\right)^2 & \text{si } d_{ij} < \sigma_{share} \\ 0 & \text{en otro caso} \end{cases}$$

Para cada individuo i de cada frente Pareto F_n se calcula la cuenta de nicho *niche count*, que refleja el número de individuos en el mismo frente que comparten nicho con i .

$$\sum_{j \in F_n, j \neq i} Sh(d_{ij})$$

El valor de aptitud final de cada individuo será el valor de aptitud “falso” (o *dummy*) calculado de forma arbitraria dividido entre la cuenta de nicho.

NPGA

En su *Niched Pareto Genetic Algorithm*, Horn y Nafpliotis [Horn y Nafpliotis, 1993] proponen varias modificaciones al clásico operador de selección por torneo. Básicamente modifican la selección para que la competencia entre los individuos se haga en base a las relaciones de dominación Pareto.

Inicialmente Horn y Nafpliotis implementan el torneo por dominación de la siguiente forma: se seleccionan al azar dos individuos de la población y se comparan. Se toma el que domine al otro. Si son indiferentes, no se especifica el comportamiento del método.

Sin embargo, este *criterio de dominación local* crea poca presión selectiva. Por ello se desarrolla un segundo método en el que, sin incrementar el tamaño de torneo (se siguen enfrentando dos individuos), sí se añade un conjunto de individuos para comparación. Ahora se toman dos individuos a y b y un conjunto de comparación C constituido por t_{dom} individuos elegidos también al azar. Si a no está dominado por ningún elemento de C y b sí lo está, se toma a . Si b no está dominado por ningún individuo de C y a sí, se toma b . El problema surge cuando ambos están dominados o ninguno lo está. En ese caso se aplica la técnica de compartición de aptitud. Horn y Nafpliotis discuten en su artículo [Horn y Nafpliotis, 1993] sobre si la compartición de aptitud debe realizarse

sobre el genotipo o sobre el fenotipo. Tras inclinarse por la distribución sobre el fenotipo, se plantean si realizarla en el espacio de parámetros o en el de objetivos. Concluyen argumentando que dado que el frente de Pareto existe sólo en el espacio de objetivos, lo correcto sería aplicar distribución en dicho espacio.

El método de compartición usado en NPGA no reduce la aptitud de los individuos. Horn y Nafpliotis lo llaman Compartición por Clases de Equivalencia (*Equivalence Class Sharing*). Cuando se produce un empate en el torneo entre los dos individuos contendientes, se toma en consideración la cuenta los nichos respectivos y se selecciona aquel con menor cuenta de nicho. Esta forma de compartición es equivalente a la compartición estándar (se divide la aptitud entre la cuenta de nicho) en el caso de que los dos individuos tengan igual aptitud.

Al igual que con otros métodos de compartición, el radio de nicho σ_{share} es importante. Horn y Nafpliotis proponen métodos para calcular los límites inferior y superior, y también aconsejan el escalado de los objetivos, para que el cálculo de distancias en la formación de nichos no favorezca a uno frente a otros.

PAES

La Estrategia Evolutiva de Frente Pareto Almacenado (*Pareto Archived Evolution Strategy*, [Knowles y Corne, 2000]), en su forma básica, es una estrategia evolutiva de tipo (1+1), es decir, con una población de un sólo individuo que se renueva generando un hijo y eligiendo el mejor de los dos (padre o hijo). Existen extensiones del algoritmo que usan estrategias $(1 + \lambda)$ y $(\mu + \lambda)$.

Tres componentes: generador de individuos, función de aceptación y lista de soluciones no dominadas (LSND).

Dado el individuo \mathbf{c} , se le muta y se genera un hijo \mathbf{m} . Para decidir cuál de los dos es la nueva solución, se sigue el siguiente proceso:

- Si $\mathbf{c} \succ \mathbf{m}$, se descarta \mathbf{m} .
- Si $\mathbf{m} \succ \mathbf{c}$, se toma \mathbf{m} como nueva solución y se añade a la LSND.
- Si son indiferentes (ninguno domina al otro):
 - Si algún miembro de la LSND domina a \mathbf{m} , se descarta \mathbf{m} .
 - Si no, se decide cuál de los dos individuos es la nueva solución basándose en si la LSND está llena o no, y en el estado de saturación de las regiones del espacio de búsqueda.

Knowles y Corne desarrollan, para implementar el último punto, un método de control de la lista de soluciones no dominadas que trata de favorecer la distribución de las soluciones a lo largo del frente de Pareto. Lo hacen usando un algoritmo de partición adaptativa que tiene un coste computacional menor que los algoritmos clásicos de creación de nichos y, además, no requiere el parámetro de radio del nicho, σ_{share} .

SPEA

El *Strength Pareto Genetic Algorithm*, de Eckart Zitzler [Zitzler, 1999][Zitzler y Thiele, 1999] usa, al igual que el PAES, una segunda población en la que almacena soluciones no dominadas. Esta segunda población se usa para calcular la aptitud de los individuos, que depende del número de puntos de dicha población que dominan al individuo. No se tienen en cuenta las relaciones de dominación entre los miembros de la población principal. La población externa se actualiza continuamente, y se intenta mantener la diversidad usando la relación de dominancia Pareto. Como última característica del método, se intenta reducir el tamaño del frente de Pareto mediante un procedimiento de agrupación (*clustering*).

El algoritmo mantiene una población real (P) y una población externa P' . Al inicio de cada generación copia los individuos no dominados de P en P' . Si el tamaño de P' crece por encima de un umbral, la población externa es podada usando un procedimiento de agrupación. La selección de individuos previa a la aplicación del cruce y la mutación se realiza por torneo sobre el conjunto $P + P'$.

La asignación de aptitud a los individuos se realiza en dos fases:

- En primer lugar, se calcula la aptitud de cada individuo $i \in P'$, que es igual a su fuerza (s_i). La fuerza del individuo i es proporcional al número de individuos de la población real a los que cubre (domina o es igual):

$$f_i = s_i = \frac{|\{j \in P \mid i \succeq j\}|}{N + 1}$$

donde N es el tamaño de la población P y $|\dots|$ representa la cardinalidad del conjunto.

- En segundo lugar, la aptitud de cada individuo $j \in P$ se calcula como uno más la suma de las fuerzas de todos los individuos $j \in P'$ que lo cubren:

$$f_j = 1 + \sum_{i \mid i \succeq j} s_i$$

Como se ve, Zitzler considera la aptitud como un valor a minimizar.

Este método de asignación de aptitud asegura que los individuos estén cerca del frente de Pareto y que la población se distribuya a lo largo del mismo. Este segundo objetivo se podría conseguir mediante métodos de compartición de aptitud que requerirían encontrar valores para el parámetro radio del nicho (σ_{share}). En el método usado por Zitzler no se necesita dicho parámetro, aunque depende del tamaño máximo de la población externa P' (N').

El hecho de que el tamaño de la población no dominada externa P' influya sobre el método de asignación de aptitud hace necesario mantener un control sobre dicho tamaño. También es recomendable que la población P' esté distribuida uniformemente

sobre el espacio de objetivos. De estas dos premisas se ocupa un método de reducción por agrupamiento (*clustering*) basado en ideas de Morse [Morse, 1980] y Rosenman y Gero [Rosenman y Gero, 1985].

El método de reducción aglutina a los individuos de P' en grupos *clusters*. Inicialmente, cada punto constituye un grupo por sí mismo. Si el número de grupos es mayor que el límite N' , se unifican los dos grupos cuya distancia sea mínima (dicha distancia entre grupos se define como la distancia media entre pares de puntos de ambos). Una vez que se tienen N' grupos, se toma un elemento de cada grupo (el centroide o punto con mínima distancia promedio al resto de los puntos del grupo) como representante del mismo.

2.2.6. Técnicas de segunda generación

Lás técnicas de evolución multiobjetivo de segunda generación también se basan en los conceptos de Pareto, pero buscan una mayor eficiencia. Se utilizan a menudo poblaciones externas de soluciones no dominadas. Asimismo se emplean técnicas específicas para asegurar la distribución uniforme de las soluciones a lo largo del frente de Pareto.

NSGA-II

Entre las técnicas de segunda generación, una de las más usadas es la segunda versión del Algoritmo Genético de Ordenación No dominada o NSGA-II [Deb et al., 2002]. Este método surge a partir del NSGA original [Srinivas y Deb, 1995] añadiendo elitismo y eliminando los parámetros que constituían el principal inconveniente de NSGA.

La ordenación de la población se realiza al igual que en NSGA según un criterio de no-dominación. Sin embargo se usa un algoritmo más rápido para ordenar la población, con lo que la complejidad original de $O(mN^3)$ se ve reducida a $O(mN^2)$, siendo m el número de objetivos y N el tamaño de la población. Esta ordenación no-dominada rápida (algoritmo 2) toma como parámetro un conjunto de individuos (soluciones) a ordenar y genera una serie de frentes no dominados F_1, F_2, \dots, F_k , con las mismas características que los frentes del anterior algoritmo NSGA. A cada solución se le asigna también su rango (que no es otra cosa que el número del frente al que pertenece y, por tanto, el número de individuos que la dominan más uno).

Para mantener la diversidad en los frentes no dominados se emplea un método que evalúa el espacio libre que hay alrededor de cada solución para asignarle un factor denominado *crowding distance*, que se calcula como el tamaño del mayor rectángulo libre alrededor de la solución. El algoritmo 3 detalla el proceso de cálculo de dicha distancia.

Dentro de un mismo frente, los individuos con mayor distancia *crowding* tienen una mejor evaluación. La comparación entre individuos se realiza dentro del algoritmo evolutivo

Algoritmo 2 Ordenación No-dominada rápida

Parámetros: P: conjunto a ordenar

para todo $p \in P$

$S_p \leftarrow \emptyset$ // S_p : soluciones dominadas por p

$n_p \leftarrow 0$ // n_p número de soluciones que dominan a p

para todo $q \in P$

si $p \prec q$

$S_p \leftarrow S_p \cup \{q\}$ // añadir q al conjunto de soluciones dominadas por p

si no si $q \prec p$

$n_p \leftarrow n_p + 1$

finsi

finpara

si $n_p = 0$

// p pertenece al primer frente (nadie lo domina)

$p.rango \leftarrow 1$

$F_1 \leftarrow F_1 \cup \{p\}$

finsi

finpara

$i \leftarrow 1$ // calculado F_1 . Ahora, el resto de los frentes

mientras $F_i \neq \emptyset$

$Q \leftarrow \emptyset$

para todo $p \in F_i$

para todo $q \in S_p$

// q recorre los individuos dominados por p

$n_q \leftarrow n_q - 1$

si $n_q = 0$

$q.rango \leftarrow i + 1$

$Q \leftarrow Q \cup \{q\}$

finsi

finpara

finpara

$i \leftarrow i + 1$

$F_i \leftarrow Q$

finmientras

Algoritmo 3 Asignación de distancia de *crowding*

Parámetros: **I**: conjunto a asignar

$l \leftarrow |I|$

para $i = 1$ a l

$I[i].distancia \leftarrow 0$

finpara

para todo objetivo m

 ordenar I en función del objetivo m

$I[1].distancia \leftarrow \infty$

$I[l].distancia \leftarrow \infty$

para $i = 2$ a $l - 1$

$I[i].distancia \leftarrow I[i].distancia + (I[i + 1].m - I[i - 1].m) / (f_m^{max} - f_m^{min})$

finpara

finpara

mediante el operador \prec_n , que se define de la siguiente forma:

$$i \prec_n j \Leftrightarrow \begin{cases} i.rango < j.rango \\ \vee \\ i.rango = j.rango \wedge i.distancia > j.distancia \end{cases}$$

Los dos algoritmos anteriores, junto con el operador \prec_n , se aplican en el algoritmo 4, que describe el comportamiento de una generación del método NSGA-II. Como puede verse, la ordenación no-dominada rápida se aplica a la unión de la población de progenitores y la de descendientes. La población de la siguiente generación se crea tomando los frentes no dominados en orden, y dentro de ellos los individuos en orden de distancia *crowding*, hasta rellenar dicha población. Sobre esta población se aplican operadores de cruce y mutación para obtener un nuevo conjunto de descendientes antes de reiniciar el proceso.

Algoritmo 4 Algoritmo genético de ordenación no-dominada II (NSGA-II)

Parámetros: \mathbf{P}_t : población de N progenitores, \mathbf{Q}_t : población de descendientes
 $R_t \leftarrow P_t \cup Q_t$ // combinar ambas poblaciones
 $F \leftarrow$ Ordenación No-dominada rápida (R_t) // $F = (F_1, F_2, \dots)$, frentes no dominados de R_t
 $P_{t+1} \leftarrow \emptyset$
 $i \leftarrow 1$
mientras $|P_{t+1}| + |F_i| > N$
 Asignación de distancia de *crowding* (F_i)
 $P_{t+1} \leftarrow P_{t+1} \cup F_i$
 $i \leftarrow i + 1$
finmientras
 Ordenar F_i según \prec_n
 $P_{t+1} \leftarrow P_{t+1} \cup F_i[1 : (N - |P_{t+1}|)]$
 $Q_{t+1} \leftarrow$ nueva población (P_{t+1}) // aplicar cruce y mutación
 $t \leftarrow t + 1$

3

Modelo Propuesto

3.1. Introducción

Basándose en los apartados estudiados en el capítulo 2 se propone a continuación un modelo de coevolución cooperativa general que usa una asignación de crédito multiobjetivo.

El modelo usa varias poblaciones de subcomponentes (número fijo no evolucionable) y una población de individuos que almacena combinaciones de subcomponentes. Las poblaciones de subcomponentes son gestionadas por algoritmos evolutivos idénticos, y la de individuos es manejada por su propio algoritmo evolutivo de estado estacionario, con operadores y parámetros diferentes. Con esta evolución en dos niveles se consigue una evaluación más eficiente y verosímil de los subcomponentes.

La asignación de crédito a los subcomponentes se realiza en función de varios criterios, y los subcomponentes se clasifican mediante un método multiobjetivo. Esto permite mantener la diversidad de los subcomponentes a lo largo de un frente de Pareto, ofreciendo mayores posibilidades a los individuos para obtener buenos subcomponentes.

3.2. Descripción general

El modelo propuesto es general y por tanto aplicable a cualquier problema, siempre y cuando el problema sea divisible en un número fijo de subcomponentes ¹.

El modelo, al ser genérico, proporciona una serie de entidades y métodos propios y deja en función del problema la implementación de entidades y métodos específicos. Hablaremos por tanto de entidades y métodos *del modelo* y entidades y métodos *del problema*.

En la figura 3.1 puede verse un esquema general del modelo propuesto: K poblaciones de subcomponentes (PS_1, \dots, PS_K) evolucionan por separado (aunque usan un esquema

¹Realmente, como se verá luego, se trata de un número máximo, ya que el modelo permite la creación y destrucción de poblaciones de subcomponentes.

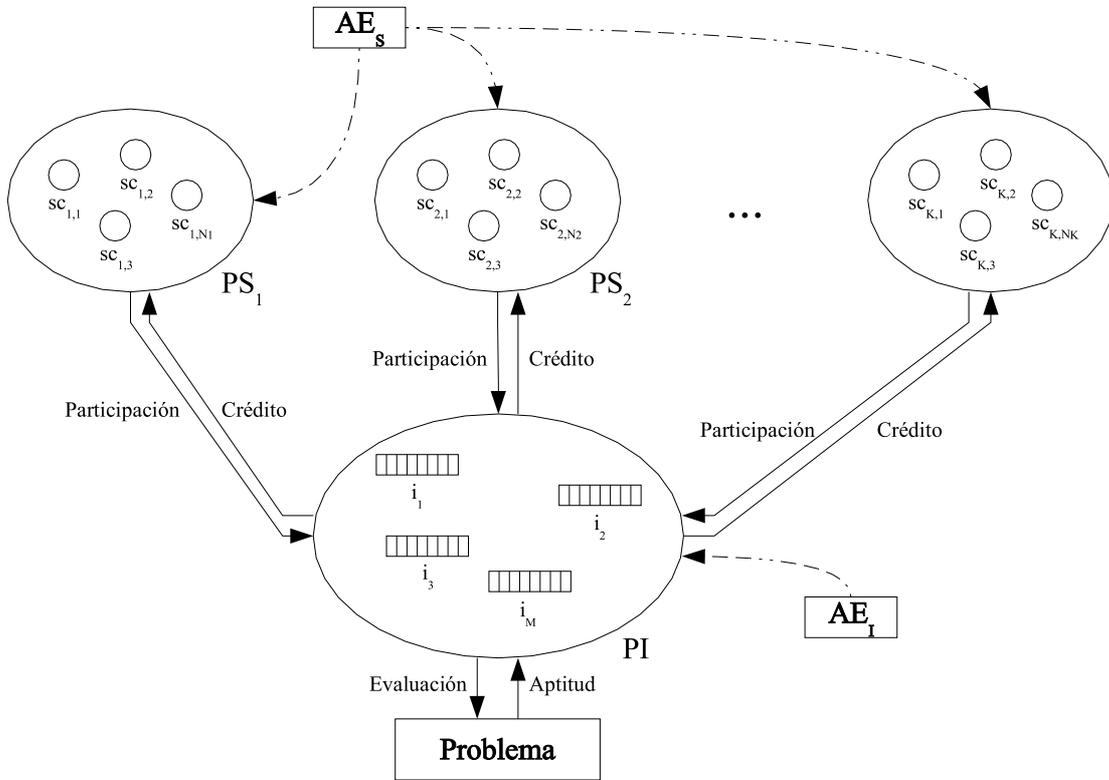


Figura 3.1: Esquema general del modelo

común de algoritmo evolutivo AE_s) para buscar soluciones a subproblemas. En paralelo, una población de individuos (PI) evoluciona mezclas de subcomponentes, guiada por el algoritmo evolutivo AE_I . La interacción entre la población de individuos y el problema se concreta en la evaluación de los individuos (i_j) frente al mismo y la asignación de aptitud resultante de dicha evaluación. La interacción entre las poblaciones de subcomponentes y la de individuos se plasma en la participación de elementos de las primeras en elementos de la segunda y en la asignación del crédito asociado a dicha participación.

La tabla 3.1 muestra todos los componentes del modelo, tanto entidades como métodos, indicando cuáles son del modelo y cuáles deben ser especificados por el problema.

3.3. Entidades

A continuación se describirán las entidades que intervienen en el modelo. Algunas son entidades del modelo y son proporcionadas por el mismo; otras son entidades del problema, varían en función de la tarea a la que se esté aplicando el modelo y por tanto deben ser especificadas en el contexto de dicha tarea.

Nombre	Tipo	Pertenencia
Población de individuos	entidad	modelo
Individuos	entidad	modelo
Población de subcomponentes	entidad	modelo
Subcomponentes	entidad	problema
Evolución de individuos	método	modelo
Evolución de subcomponentes	método	modelo
Creación de poblaciones iniciales: individuos	método	modelo
Creación de poblaciones iniciales: subcomponentes	método	problema
Operadores genéticos sobre individuos	método	modelo
Operadores genéticos sobre subcomponentes	método	problema
Operadores de selección	método	modelo
Evaluación de individuos	método	problema
Evaluación de subcomponentes	método	modelo

Cuadro 3.1: Entidades y métodos.

3.3.1. Población de individuos

La población de individuos (PI) es una entidad del modelo que agrupa M individuos. Estos son evolucionados por el algoritmo evolutivo AE_I , cuyo funcionamiento se verá posteriormente. El modelo plantea una coevolución en dos niveles, del tipo que se vio en la página 13, en la que las mezclas de subcomponentes no son efímeras sino que se mantienen en una población que evoluciona en paralelo.

Esta evolución en dos niveles tiene varias ventajas: permite al modelo evaluar de forma más realista y precisa a los subcomponentes; mantiene una continuidad temporal de las buenas agrupaciones de los mismos y en lugar de centrarse sólo en una búsqueda de buenas soluciones para los subproblemas también busca buenas combinaciones de dichas soluciones.

3.3.2. Individuos

Los individuos son entidades del modelo que tienen el objetivo de agrupar subcomponentes con el fin de formar una solución completa al problema. Los individuos también actúan como medio para la evaluación de los subcomponentes, ya que una parte de la aptitud obtenida por el individuo será pasada en forma de crédito a los subcomponentes que han participado en él.

Cada individuo (ver figura 3.2) consta de K referencias o punteros a otros tantos subcomponentes. Cada una de estas referencias consta de dos partes: un indicador p de la población y un indicador s del subcomponente dentro de esa población. Por tanto el genotipo de un individuo i_j es un conjunto de K pares (p, s) que apuntan a los

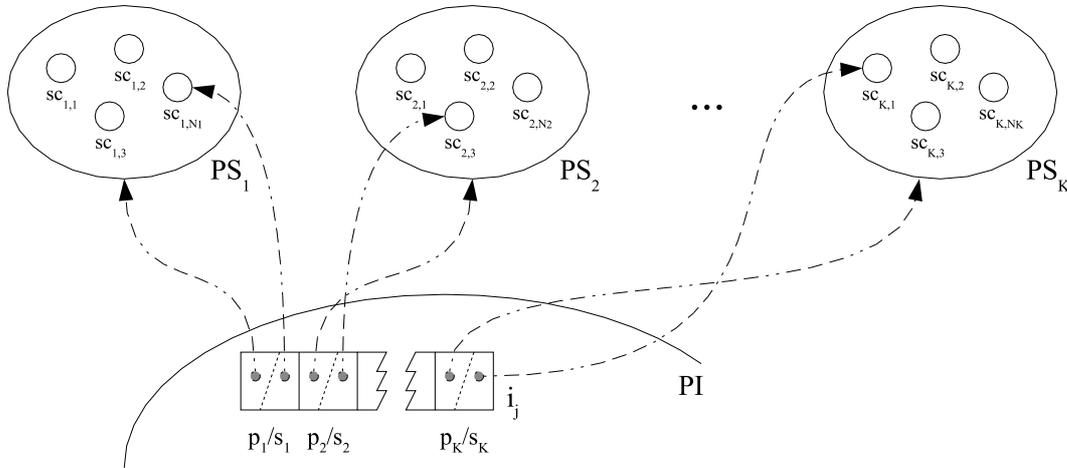


Figura 3.2: Representación detallada de un individuo y sus relaciones con las poblaciones de subcomponentes.

subcomponentes que lo forman.

$$\text{genotipo}(i_j) = \{(p_1, s_1), (p_2, s_2), \dots, (p_K, s_K)\}$$

Cuando se necesita construir el individuo a partir de sus subcomponentes se obtiene el fenotipo seleccionando de cada población el subcomponente correspondiente y combinándolos todos. Así, si se tiene (suponiendo, por ejemplo, $K = 4$)

$$\text{genotipo}(i_j) = \{(1, 6), (2, 13), (3, 9), (4, 10)\}$$

entonces el fenotipo se obtendrá como

$$\text{fenotipo}(i_j) = sc_{1,6} \cup sc_{2,13} \cup sc_{3,9} \cup sc_{4,10}$$

donde el operador \cup , como ya se ha dicho anteriormente, debe ser interpretado como agregación de subcomponentes.

Como se ve en el ejemplo, se toman cuatro subcomponentes, uno de cada población. Esto es lo normal, aunque el hecho de que los individuos almacenen las referencias incluyendo la población permitiría usar varios subcomponentes de la misma población para formar un individuo. Esto sería poco aplicable, por supuesto, ya que equivaldría a incluir dos o más soluciones distintas al mismo subproblema dentro de la solución completa. Sin embargo es concebible encontrar problemas que requieran este tipo de comportamiento.

Por otra parte, aunque normalmente el número de subcomponentes (K) es fijo, puede haber casos de individuos compuestos por un número menor de subcomponentes:

- En experimentos que traten de obtener una descomposición emergente del problema. En este caso K se consideraría el número *máximo* de subcomponentes y un método externo (por ejemplo, el algoritmo evolutivo AE_I encargado de la población de individuos) decidiría cuántos subcomponentes son realmente usados por los individuos. Este mismo método tendría la capacidad de ampliar o reducir el número de componentes según las necesidades de la evolución.
- En asignaciones de crédito que requieran la evaluación de individuos incompletos (como la *asignación por diferencia* que se verá más adelante).

En ambos casos los subcomponentes que no fueran utilizados en un individuo se marcarían de forma especial (por ejemplo, indicando un valor de p negativo):

$$\begin{aligned} \text{genotipo}(i_j) &= \{(1, 6), (-1, \mathbf{0}), (3, 9), (4, 10)\} \\ \text{fenotipo}(i_j) &= sc_{1,6} \cup sc_{3,9} \cup sc_{4,10} \end{aligned}$$

Hay que puntualizar que, aunque el individuo contenga las referencias a los subcomponentes que lo componen, la agregación de los mismos para componer un individuo real que pueda ser aplicado al problema está a cargo del método de evaluación, que se describirá en la sección 3.4.8.

3.3.3. Poblaciones de subcomponentes

Las poblaciones de subcomponentes son entidades del modelo y no dependen del problema. Su cometido es albergar los subcomponentes que serán evolucionados para resolver subproblemas.

El modelo evoluciona los subcomponentes en poblaciones separadas lo que, tal como se discutió ya en la página 12, separa claramente las distintas especies, reduce el problema del mantenimiento de la diversidad, facilita la paralelización del algoritmo y permite la evolución de subcomponentes heterogéneos.

Hay K poblaciones de subcomponentes denominadas PS_1, PS_2, \dots, PS_K y el número de elementos en cada una de ellas es potencialmente distinto (N_1 elementos en la primera, N_2 en la segunda $\dots N_K$ en la última). Todas las poblaciones son evolucionadas de forma independiente por el algoritmo evolutivo AE_S , que sigue el mismo esquema de evolución pero permite diferentes parámetros en cada población.

3.3.4. Subcomponentes

Los subcomponentes son entidades del problema, es decir, deben ser descritas cuando se aplique el modelo general a un problema específico. Puede haber diferentes tipos de subcomponentes (subcomponentes heterogéneos) con distintos genotipos adaptados al tipo de subproblema que deban resolver.

Los subcomponentes son evolucionados de forma independiente dentro de sus propias poblaciones. El algoritmo evolutivo AE_S actúa aplicando sobre ellos operadores de mutación y cruce (métodos del problema que dependen de la codificación del subcomponente) y seleccionándolos en función de su aptitud.

La aptitud de los subcomponentes, en tanto que miembros de una población evolucionable, es el crédito que obtengan de su participación en uno o varios individuos. La asignación del crédito a un subcomponente se realiza mediante un método del modelo que será detallado en la sección 3.4.9. Este método permite evaluar un subcomponente en función de varios criterios de crédito, y ordenar la población de subcomponentes según criterios de optimización multiobjetivo previamente a la selección.

3.4. Métodos

Una vez presentadas las entidades del modelo, pasaremos a describir los métodos usados sobre las mismas. Una vez más, se discutirán métodos del modelo y métodos del problema. Los primeros serán explicados en detalle. Los segundos serán descritos en general, ya que su implementación dependerá del problema al que se esté aplicando el modelo. En los capítulos 6 y 5 se aplicará este método a problemas reales y se especificarán las entidades y métodos usados para dichos problemas.

Los métodos que se describen a continuación tienen que ver básicamente con la evolución de las poblaciones de subcomponentes ($PS_1 \dots PS_K$) y de individuos (PI). Los dos primeros métodos que se tratarán (evolución de individuos y de subcomponentes) constituyen el esquema general de los algoritmos evolutivos AE_I y AE_S respectivamente. El resto de los métodos son subordinados de estos dos primeros.

3.4.1. Evolución de individuos

La población de individuos PI es evolucionada por el algoritmo evolutivo AE_I , que tiene como objetivo dirigir a los individuos hacia la selección de combinaciones de subcomponentes que resuelvan el problema de forma óptima. Este algoritmo es un método del modelo.

El algoritmo AE_I sigue un esquema de evolución de estado estacionario (*steady-state* [Syswerda, 1989][Whitley y Kauth, 1988]). Como puede verse en la figura 3.3, la población pasa de una generación a la siguiente ² prácticamente sin cambios. Los dos peores individuos son substituidos por un cruce entre otros dos individuos seleccionados por torneo. El diez por ciento de la población (seleccionado al azar) sufre mutación. El resto de la población no varía. Posteriormente (sección 3.4.5) se verán los operadores de cruce y mutación sobre individuos.

²Aunque en el contexto de un algoritmo evolutivo de estado estacionario no tiene sentido hablar de generaciones, usaremos ese término, por homogeneidad, para referirnos a los intervalos de tiempo que organizan la evolución

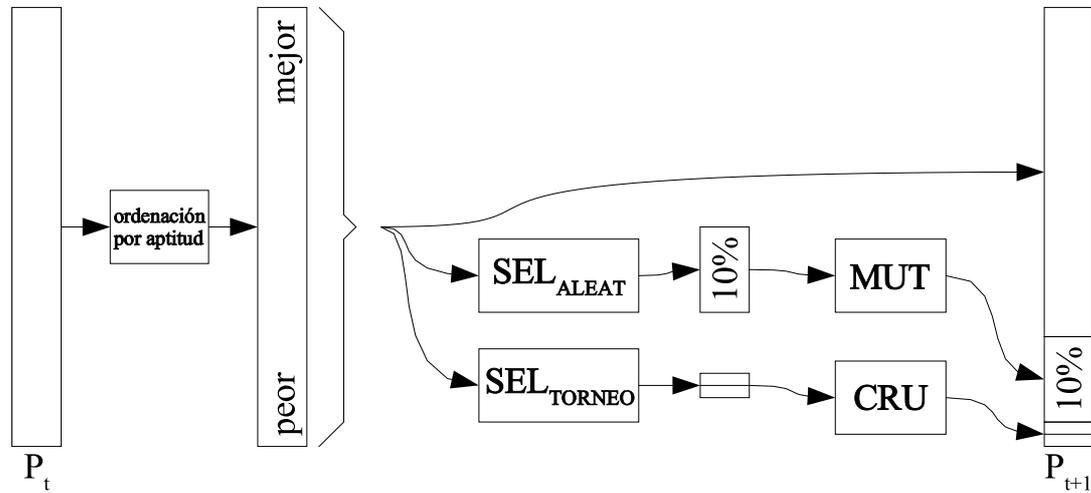


Figura 3.3: Esquema de evolución de individuos (algoritmo AE_I)

Originalmente el método propuesto usaba para la evolución de la población un esquema generacional similar al que se describirá para la evolución de subcomponentes, pero el algoritmo de estado estacionario se ha demostrado mejor para asegurar la convergencia de la población. Esto es debido probablemente a la sensibilidad de la población de individuos a los cambios, por lo que la evolución lenta y sin cambios bruscos que proporciona un algoritmo de estado estacionario es la más indicada.

3.4.2. Evolución de subcomponentes

El algoritmo evolutivo AE_S se ocupa de evolucionar las K subpoblaciones $PS_1 \dots PS_K$ de subcomponentes. En realidad, puesto que cada población de subcomponentes evoluciona por separado y los parámetros de la evolución son independientes para cada una, se podría considerar que existen K algoritmos de evolución de subcomponentes. Sin embargo, todos ellos siguen el mismo esquema de evolución. Este método es proporcionado por el modelo.

La evolución de los subcomponentes (ver figura 3.4) se hace por un esquema generacional, en el que la población P_{t+1} ³ se obtiene a partir de la población P_t uniendo tres partes: una parte consta de un porcentaje de subcomponentes intactos; otra parte está formada por un número de subcomponentes de P_t mutados; la tercera parte aporta una cantidad de subcomponentes cruzados.

El modelo permite que parte de los subcomponentes que se transferirán intactos por reproducción sean escogidos de forma elitista con respecto a los individuos: si el número de subcomponentes elitistas es n , no se transfieren los n mejores subcomponentes de

³La variable t indica el número de generación.

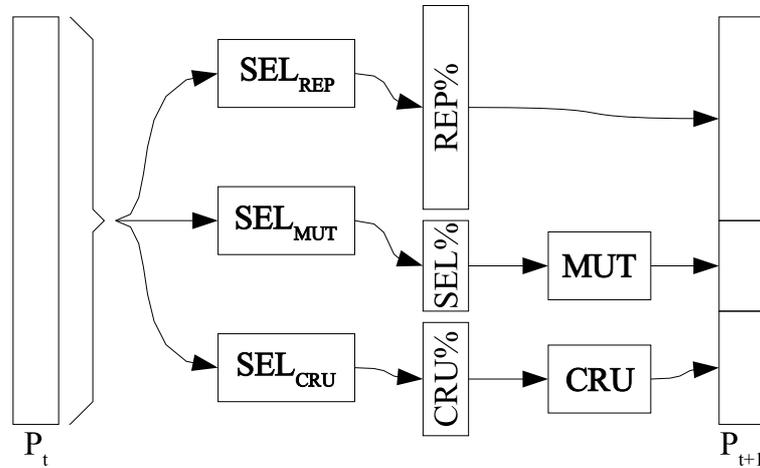


Figura 3.4: Esquema de evolución de subcomponentes (algoritmo AE_S)

una población a la siguiente, sino los subcomponentes pertenecientes a los n mejores individuos. Por supuesto, el número real de subcomponentes elitistas puede ser menor que n , ya que un subcomponente puede participar en más de un individuo. .

Si se quisiera un elitismo relativo a la aptitud de los subcomponentes (su crédito), es decir, un elitismo en el sentido estándar del término, podría reforzarse la selección para reproducción (incrementando el tamaño de torneo o usando selección de los mejores o ruleta).

La selección para reproducción (la parte no elitista de la misma) es un torneo de tamaño 2. Para cruce y mutación se usan selecciones aleatorias. El número de subcomponentes que forman cada una de las secciones (reproducción, cruce y mutación) puede ser especificado en cada experimento, por supuesto (al igual que el tipo de selección y muchos otros parámetros, como se verá en el capítulo 4). Sin embargo, en la mayoría de los experimentos se reproducirá el 50 % de la población, se cruzará el 25 % y se mutará otro 25 %. Estas proporciones han sido sugeridas por resultados experimentales, a través de un diseño de experimentos que permitió llegar a estos valores.

Dadas las características especiales del modelo, durante la creación de las nuevas poblaciones de subcomponentes se debe tener en cuenta que las posiciones de los subcomponentes dentro de la población no deben modificarse, ya que los individuos mantienen referencias a ellos. Este aspecto será tratado más a fondo en el capítulo 4, cuando se trate acerca de la implementación del modelo.

3.4.3. Creación de poblaciones iniciales: individuos

La población PI se inicia con M individuos aleatorios. Cada individuo i_j creado contiene referencias a subcomponentes aleatorios, uno de cada población PS_p :

$$i_j = \left\{ (1, U[1, N_1]), (2, U[1, N_2]), \dots, (K, U[1, N_K]) \right\}, 1 \leq j \leq M$$

donde $U[a, b]$ representa un valor entero aleatorio uniforme en el intervalo $[a, b]$.

El método de creación de individuos es un método del modelo.

3.4.4. Creación de poblaciones iniciales: subcomponentes

El método usa K poblaciones de subcomponentes ($PS_1 \dots PS_K$) que contienen $N_1 \dots N_K$ elementos respectivamente. Esos subcomponentes se crean al inicio de la evolución de forma aleatoria. Dado que las entidades (los subcomponentes) pertenecen al problema y no al modelo, la creación de las mismas es un método del problema. En los capítulos 5 y 6 se especificará la forma de creación de subcomponentes específicos de cada problema.

3.4.5. Operadores genéticos: individuos

El algoritmo evolutivo AE_I usa para modificar la población PI tres operadores genéticos: la reproducción, el cruce y la mutación. Todos ellos están proporcionados por el modelo.

El operador de reproducción transfiere un individuo intacto de una población a otra. El operador de cruce sobre individuos es un operador estándar de cruce de dos puntos (ver figura 3.5) que a partir de dos individuos i_j e i_k genera dos descendientes i_m e i_n seleccionando dos valores (a y b) enteros aleatorios uniformes en el intervalo $[1, K]$ e intercambiando el segmento del vector de referencias a subcomponentes del individuo i_j con el correspondiente segmento de i_k .

El operador de mutación de individuos sustituye uno de los subcomponentes que forman el individuo (tomado al azar) por otro subcomponente de la misma población. Así, por ejemplo, si se va a mutar el individuo i_j

$$\text{genotipo}(i_j) = \{(1, 6), (2, 3), (3, 9), (4, 10)\}$$

y la mutación se produce en el segundo subcomponente, el individuo resultante i_k podría tener el siguiente genotipo, por ejemplo:

$$\text{genotipo}(i_k) = \{(1, 6), (2, 5), (3, 9), (4, 10)\}$$

Como se ve, se ha sustituido el tercer subcomponente de la segunda población (PS_2) por el quinto subcomponente de la misma.

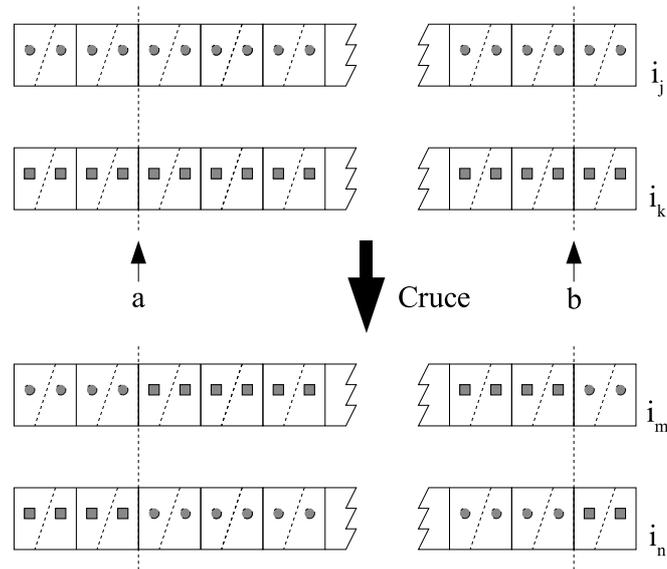


Figura 3.5: Cruce de dos puntos sobre individuos

La elección del subcomponente que sustituye al actual puede hacerse al azar o por un método de que aplique una presión selectiva (ruleta, por ejemplo). Sin embargo, los resultados experimentales con un método y otro son equivalentes. Se ha decidido usar selección al azar.

Puede verse gráficamente otro ejemplo de mutación en la figura 3.6.

3.4.6. Operadores genéticos: subcomponentes

Los operadores genéticos sobre subcomponentes son empleados por el algoritmo evolutivo AE_S durante la evolución de las poblaciones PS_p . Todos ellos son operadores del

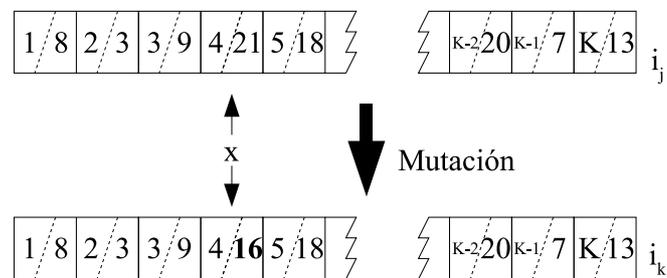


Figura 3.6: Mutación de individuos

problema: las posibilidades de cruce y mutación de subcomponentes dependen enteramente del genotipo que se use para los mismos. En cuanto al operador de reproducción, como se verá en el capítulo 4, la implementación del modelo lo sitúa como método del problema, ya que requiere conocimiento del genotipo de los subcomponentes para duplicarlo.

Por otra parte, estos operadores pueden ser diferentes para cada población de subcomponentes, ya que el modelo permite subcomponentes heterogéneos, con distinta representación.

3.4.7. Operadores de selección

Tal como se ha apuntado cuando se describieron los algoritmos AE_I y AE_S , el modelo usa varios métodos estándar de selección basados en aptitudes. Todos ellos son métodos del modelo. Aunque la implementación permite (ver capítulo 4) elegir el método de selección particular para un experimento determinado, se han usado en los experimentos los métodos que se han indicado en las secciones anteriores.

3.4.8. Evaluación de individuos

Cada individuo representa una solución tentativa al problema que se trata de resolver. Esta solución, por la naturaleza del modelo, está formada por una serie de soluciones parciales (subcomponentes) que tratan de especializarse en resolver partes del problema principal. La aplicación de un individuo al problema para evaluar la bondad de la solución y la medición de esa bondad en forma de valor de aptitud es un proceso específico del problema.

Sin embargo, sí se requiere que el método de evaluación de individuos genere un valor de aptitud de tipo real, tal que un valor mayor implique una mejor aptitud. Esto deja abierto el intervalo de aptitudes, algo muy útil para problemas en los que la aptitud óptima no es conocida.

3.4.9. Evaluación de subcomponentes

La aptitud de un individuo indica lo bien que éste resuelve el problema. Esa aptitud, como se indicó en el capítulo 2, debe ser distribuida entre los subcomponentes que forman el individuo y esta división se realiza haciendo que el subcomponente reciba una parte de la aptitud de cada uno de los individuos en los que participe. Realmente algunos criterios de asignación no consideran que el subcomponente participe en un individuo, sino que crean un individuo (uniendo una serie de subcomponentes) específicamente para la asignación de crédito.

Propondremos a continuación un modelo general para la asignación de crédito que será posteriormente aplicado a la descripción de los criterios de asignación usados en este trabajo. El modelo se basa en tres pasos: la creación de un conjunto de individuos

de los cuales se extraerá información ⁴, la definición de un método de extracción de esa información y la definición de un método de agregación de esas informaciones extraídas.

La asignación de crédito a un subcomponente $sc_{p,q}$ se realizará mediante la siguiente fórmula:

$$credito(sc_{p,q}) = \mathbb{F}(\{assign(i, sc_{p,q}), \forall i \in CI\})$$

donde $CI \subset PI$, \mathbb{F} es una función de agregación (suma, máximo, media. . .) y $assign(i, sc_{p,q})$ es una función que evalúa la importancia del subcomponente $sc_{p,q}$ en el individuo i .

Se debe determinar, pues, el conjunto de colaboraciones CI , la parte de la aptitud que recibe el subcomponente (función $assign()$) y el método de mezcla de varios créditos procedentes de varias colaboraciones (función \mathbb{F}).

Este sistema construye en primer lugar un conjunto CI de individuos, ya sean individuos existentes (pertenecientes a PI) o individuos virtuales (creados al efecto). Una vez creado el conjunto CI se evalúa, para cada elemento i del conjunto, la asignación de crédito que el individuo i otorga al subcomponente $sc_{p,q}$ (función $assign(i, sc_{p,q})$). Se obtiene así un conjunto de una o más asignaciones, que se agregan mediante la función \mathbb{F} . Ésta puede hallar la asignación máxima, o la media, o cualquier otro tipo de relación.

Existe la posibilidad de que el conjunto CI esté vacío. Por ejemplo: si $CI = \{i_j \in PI | sc_{p,q} \in i_j\}$ y el subcomponente $sc_{p,q}$ no participa en ningún individuo, se tiene $CI = \emptyset$. En estos casos se debe asignar a $sc_{p,q}$ un crédito menor que el de todos los demás subcomponentes. Idealmente, se debería usar un valor especial de crédito, indicando “no evaluable”, pero no se contempla esta posibilidad.

Tipos de crédito

El modelo proporciona cinco tipos de crédito que se pueden asignar a un subcomponente. Se describen de forma resumida en la tabla 3.2. Cada uno intenta capturar una característica deseable de los subcomponentes. Este hecho, añadido al uso (que se tratará a continuación) de un algoritmo multiobjetivo, permite mantener un tipo especial de diversidad en las poblaciones de subcomponentes: se evolucionan subcomponentes que resulten efectivos sobre una serie de criterios, y se mantiene un frente de Pareto sobre estos criterios.

Los criterios que se proporcionan como métodos del modelo son los siguientes:

Mejor El subcomponente obtiene como crédito la aptitud del mejor individuo en el que participe. El conjunto CI está formado por aquellos individuos que contienen al subcomponente evaluado. La función de agregación \mathbb{F} es el máximo y cada individuo aporta su aptitud al subcomponente.

⁴Normalmente se tratará de aquellos individuos de PI en los que participe el subcomponente a evaluar, pero el modelo permite otras variantes, como se pondrá de manifiesto en el criterio de asignación “Unión”, que se verá más tarde.

Criterio	CI	\mathbb{F}	$asign(i, sc_{p,q})$
Mejor	$\{i_j \in PI sc_{p,q} \in i_j\}$	<i>máximo</i>	$aptitud(i)$
Media	$\{i_j \in PI sc_{p,q} \in i_j\}$	<i>media</i>	$aptitud(i)$
Sustitución	$\{i_j \in PI sc_{p,q} \notin i_j\}$ (N_S mejores)	<i>media</i>	$aptitud(i - sc_{p,q} \cup \{sc_{p,q}\}) - aptitud(i)$
Diferencia	$\{i_j \in PI sc_{p,q} \in i_j\}$	<i>media</i>	$aptitud(i - sc_{p,q}) - aptitud(i)$
Unión	$\{sc_{p,q} \cup \mathbb{M}\}$	—	$aptitud(i)$

Donde $\mathbb{M} = \bigcup_{r=1, r \neq p}^K \{sc_{r,x} | credito(sc_{r,x}) > credito(sc_{r,y}), \forall y \neq x\}$

Cuadro 3.2: Tipos de asignación de crédito proporcionados por el modelo

Media Se asigna al subcomponente la media de las aptitudes de aquellos individuos en los que participe. En este criterio el conjunto CI está formado, al igual que en el anterior, por los individuos de PI que usan el subcomponente evaluado. Se usa la media como función de agregación. Cada individuo asigna al subcomponente su aptitud como crédito.

Sustitución Se seleccionan los N_S mejores individuos de PI que no usen el subcomponente a evaluar. En cada uno de ellos se inserta el subcomponente que está siendo evaluado, reemplazando al subcomponente de su misma población. Se recalcula la aptitud del nuevo individuo. Se calcula el incremento de aptitud producido ($aptitud(nuevo) - aptitud(original)$), que puede ser negativo. Se asigna al subcomponente como crédito la media de estas diferencias de aptitud. Se recordará que la aptitud usada en el modelo considera mejores los valores mayores. Este método de asignación de aptitud requiere evaluaciones adicionales de individuos.

La formalización del criterio es la siguiente: el conjunto CI está formado por los N_S mejores individuos de PI que no usen el subcomponente a evaluar; la función \mathbb{F} es la media; la asignación de crédito de cada individuo es la diferencia entre la aptitud del individuo modificado (con el subcomponente reemplazado) y la aptitud original del individuo.

Diferencia En cada uno de los individuos que utilizan el subcomponente a evaluar se elimina éste. Se evalúa el individuo y se calcula la diferencia entre la aptitud con el subcomponente y la aptitud sin él ($aptitud(con) - aptitud(sin)$). Esta diferencia puede, de nuevo, ser negativa, indicando que el subcomponente “entorpece” al individuo y no aporta nada a la resolución del problema. La media de estas diferencias se asigna como crédito.

Este criterio requiere también evaluaciones adicionales de individuos. Por otra parte, genera individuos incompletos, por lo que sólo puede ser aplicado a determinados problemas que permiten evaluaciones de individuos parciales.

El conjunto CI está formado por los individuos que usan el subcomponente. La función de agregación es la media y el crédito asignado por cada individuo es la diferencia entre la aptitud del individuo original y la aptitud del individuo sin el subcomponente a evaluar.

Unión Este criterio de asignación de crédito es diferente a los anteriores: no utiliza individuos existentes en la población PI , sino que genera un único individuo constituido a partir del subcomponente que está siendo evaluado y un conjunto (llamado \mathbb{M} en la tabla 3.2) de los mejores subcomponentes del resto de las poblaciones ⁵. Este criterio de asignación de crédito está basado en el método usado por Potter [Potter y de Jong, 2000].

El conjunto CI consta de un individuo solamente. La función de agregación \mathbb{F} no tiene aplicación, ya que $|CI| = 1$. La asignación al subcomponente es la aptitud del individuo sintetizado.

Cada uno de estos criterios trata de buscar que las poblaciones PS_p evolucionen subcomponentes eficientes en múltiples aspectos: los criterios *mejor* y *media* evitan que se pierdan subcomponentes buenos; el criterio *sustitución* intenta fomentar la competición entre módulos de la misma población; *diferencia*, en cambio, refuerza la competición entre poblaciones de subcomponentes y evita que dos poblaciones generen subcomponentes que resuelvan el mismo subproblema; por último, el criterio *unión* evalúa al subcomponente de forma ajena a la población de individuos, y lo coloca en un entorno complejo y exigente (los mejores subcomponentes del resto de las poblaciones).

Métodos de asignación

El modelo puede utilizarse tanto con asignación de crédito estándar (monoobjetivo) como con múltiples asignaciones de crédito agrupadas mediante un método multiobjetivo.

Se proporcionan dos métodos de asignación de aptitud multiobjetivo: el algoritmo genético de ordenación no-dominada (*nondominated sorting genetic algorithm*, NSGA [Srinivas y Deb, 1995]) y su versión mejorada, NSGA-II [Deb et al., 2002], que elimina la necesidad de parámetros. Se han escogido estos dos métodos debido a su popularidad y a que no utilizan poblaciones externas de soluciones no dominadas.

Tanto uno como otro se utilizan en el modelo únicamente para la conversión de múltiples objetivos (créditos) en un único valor de aptitud, que será usado para la selección en el algoritmo evolutivo AE_S que controla las poblaciones de subcomponentes. NSGA y NSGA-II originalmente incluyen una serie de métodos evolutivos (elitismo, modelos de creación de nuevas poblaciones...) que han sido descartados en favor del esquema evolutivo descrito anteriormente.

3.5. Ejecución del modelo

En el algoritmo 5 puede verse el sistema que usa el modelo para gestionar la evolución e interacción de todas las poblaciones ($PI, PS_1, PS_2, \dots, PS_K$).

⁵Los mejores subcomponentes se hallan en función de la asignación de crédito realizada en la generación anterior. Si se está en la primera generación se considerarán subcomponentes aleatorios

Algoritmo 5 Ejecución del modelo

```
para todo  $i \in 1 \dots n$ 
  Crear subcomponentes aleatorios para rellenar  $PS_i$ 
finpara
  Crear individuos aleatorios para rellenar  $IP$ 
mientras no termine la evolución
  para todo  $i \in 1 \dots n$ 
    Evolucionar  $PS_i$  una generación
  finpara
  Evolucionar  $IP$  una generación
  Evaluar individuos de  $IP$ 
  para todo  $i \in 1 \dots n$ 
    Evaluar subcomponentes de  $PS_i$ 
  finpara
  Actualizar estadísticas
finmientras
  Mostrar resultados y estadísticas finales
```

En líneas generales, tras la creación de las poblaciones iniciales aleatorias, se entra en un bucle de evolución. Dentro del mismo se evolucionan incrementalmente las poblaciones de subcomponentes, se modifica la población de individuos (recordemos que PI se evoluciona mediante un modelo de estado estacionario) y se evalúan todos los individuos frente al problema y todos los subcomponentes en función de su participación en individuos. Por último se actualizan las estadísticas: aptitud del mejor individuo, aptitud media, desviación estándar de las aptitudes, aptitud del mejor individuo en toda la evolución, y sus correspondientes para cada población de subcomponentes.

El criterio de final de evolución se cumple al alcanzarse un número máximo de generaciones, indicado como parámetro en cada experimento. Es sencillo cambiar el criterio de parada del algoritmo y sustituirlo por uno basado en las aptitudes de los individuos (obtención de un valor límite, estancamiento de las aptitudes...), pero con un criterio basado en el número de generaciones se pueden realizar los experimentos de forma más homogénea. Cuando termina la evolución se muestran los resultados y las estadísticas del proceso evolutivo. Se obtiene como solución el mejor individuo de la población PI .

4

Implementación

4.1. Introducción

En anteriores capítulos se ha tratado el campo de la coevolución cooperativa de forma general y se ha propuesto un modelo genérico para la resolución de problemas mediante esa técnica evolutiva. En este capítulo se describirá la implementación que se ha hecho de dicho modelo general, y en los siguientes capítulos (5 y 6) se describirán las adaptaciones del modelo y de la implementación a problemas reales, así como las pruebas realizadas y los resultados obtenidos.

El modelo propuesto, descrito en el capítulo 3, ha sido implementado en lenguaje C [Schildt, 2000] [van der Linden, 1994] [Kernighan y Ritchie, 1988], usando el compilador `gcc`. Todo el desarrollo se ha realizado en el sistema operativo GNU/Linux [Welsh y Kaufman, 1996] [Siever et al., 2000], usando la distribución Debian [Murdock, 1994] [Camou et al., 2000]. El código fuente está disponible en <ftp://ftp.ayrna.org/pub/ERSanz>.

En la sección 4.2 se describirá el esquema general de la implementación. La sección 4.3 tratará las principales estructuras de datos. Los algoritmos más importantes se describen en la sección 4.4. La forma de adaptar un problema para su uso con el modelo se trata en la sección 4.5 y los ficheros de configuración se describen en la sección 4.6.

4.2. Esquema general

La implementación consta de una serie de funciones que codifican los métodos del modelo (que se describieron en el capítulo 3) y que a su vez utilizan determinadas funciones que implementan los métodos del problema. Estas últimas han de ser codificadas en función del problema que se trate de resolver. Veremos en los capítulos 5 y 6 la aplicación del modelo a dos problemas reales y entonces se describirán las funciones que se han codificado para dichos problemas.

Todas estas funciones (del modelo y del problema) trabajan sobre un conjunto de estructuras de datos que almacenan la información sobre el estado de las poblaciones, las estadísticas de la evolución y los parámetros de los algoritmos evolutivos.

Comenzaremos describiendo las estructuras de datos principales para pasar luego a detallar los algoritmos.

4.3. Estructuras de datos

Una de las estructuras de datos más usadas en la implementación es la estructura **configuración**. Hay una variable de este tipo, llamada *conf*, accesible a todas las funciones.

La variable *conf* contiene parámetros como el número de generaciones a ejecutar, el número de subcomponentes (K), el tamaño de la población de individuos (M), el tamaño de cada una de las K poblaciones de subcomponentes (N_p) en forma de vector de números enteros ($N[p]$).

También incluye información sobre el número de criterios de asignación de crédito a usar en el método multiobjetivo, y el tipo de éstos criterios. Se almacena el número de subcomponentes elitistas que se reproducirán y el porcentaje de la nueva población de subcomponentes que se creará por reproducción, cruce y mutación.

Otros valores almacenados en la variable de configuración determinan parámetros para los criterios de asignación de crédito, métodos alternativos de selección para los algoritmos evolutivos, parámetros propios del problema...

```

----- Configuración -----
ENUMERACION objetivo = { MEJOR, MEDIA, SUSTITUCION, DIFERENCIA, UNION }

ESTRUCTURA configuración:
    ENTERO n_generaciones

    ENTERO K          // Número de subcomponentes
    ENTERO M          // Número de individuos
    ENTERO N[K]       // Tamaño de cada población de subcomps.

    ENTERO nobjs      // Número de criterios a evaluar
    ENUMERACION objetivo objs[nobjs]
                    // Tipos de criterio

    ENTERO elitismo
    ENTERO pct_rep     // Porcentaje de subcomps. a reproducir
    ENTERO pct_cru     // íd. a cruzar
    ENTERO pct_mut     // íd. a mutar

    ENTERO numsustituciones
                    // parámetro para la asignación de crédito
                    // por sustitución

    { Otras opciones:

```

```

    depuración,
    descripción del problema,
    tipos de selección,
    parámetros del algoritmo multiobjetivo
}

```

ESTRUCTURA configuración conf

La estructura de datos **subcomponente** alberga información sobre el genotipo del subcomponente (información que será manejada únicamente por las funciones del problema), los valores de cada criterio de asignación de crédito y el valor único de crédito (obtenido mediante el algoritmo multiobjetivo).

———— Subcomponente ————

```

ESTRUCTURA subcomponente:
    DATOS usr          // datos propios, usados por el problema
    REAL objetivos[conf.nobjs]
    REAL crédito

```

Con la estructura **individuo** se describe una combinación de K subcomponentes. Esto se hace mediante un vector de K estructuras **puntero**. Cada una de ellas contiene el índice de la población y el índice del subcomponente dentro de esa población.

La estructura descriptiva del individuo también contiene el valor de la aptitud del mismo.

———— Individuo ————

```

ESTRUCTURA puntero:
    ENTERO pob
    ENTERO sc

ESTRUCTURA individuo:
    ESTRUCTURA puntero scs[conf.K]
    REAL aptitud

```

Los individuos están agrupados para su evolución en la estructura **pobinds**, que contiene un vector de M individuos y estadísticas sobre la evolución. La variable PI almacena esta información.

———— Población de individuos ————

```

ESTRUCTURA pobinds:
    ESTRUCTURA individuo pob[conf.M]

    { Estadísticas de la población actual }
    { Estadísticas históricas }

ESTRUCTURA pobinds PI

```

Por último, las K poblaciones de subcomponentes se almacenan en el vector PS , cuyos elementos son estructuras de tipo **pobscs** y contienen la población de subcomponentes y las estadísticas de la evolución de la misma. Cada población de subcomponentes tiene tamaño diferente, especificado mediante el valor $N[p]$ de la estructura de configuración (variable *conf*).

Población de subcomponentes
<pre> ESTRUCTURA pobscs: ESTRUCTURA subcomponente pob[conf.N[p]] // tamaño diferente para cada instancia { Estadísticas de la población } ESTRUCTURA pobscs PS[conf.K] </pre>

4.4. Algoritmos

El algoritmo principal de la implementación es el Proceso Evolutivo General (algoritmo 6). Tras cargar la configuración de un fichero y almacenarla en la variable *conf*, inicia la población de individuos de forma aleatoria y evalúa cada uno de los miembros de la misma. A continuación crea y evalúa las poblaciones de subcomponentes.

El bucle principal del algoritmo se repite tantas veces como generaciones haya que ejecutar. Cada generación implica la evolución de todas las poblaciones (la de individuos y las de subcomponentes) y la evaluación de sus miembros. El orden en que se evolucionan y evalúan las poblaciones es importante debido a las interacciones existentes entre ellas, y sobre todo a la necesidad de obtener las aptitudes de los individuos antes de asignar crédito a los subcomponentes que han participado en ellos.

Cuando termina el proceso evolutivo se muestra el resultado final y se proporciona información estadística sobre la ejecución del experimento.

En este algoritmo hay dos métodos que deben ser implementados en función del problema: la evaluación de individuos y la creación de subcomponentes aleatorios.

La evolución de la población de individuos se realiza mediante el algoritmo 7, que representa un paso de esta evolución. Tras seleccionar los dos peores individuos de la población actual, se cruzan y los dos individuos resultantes se insertan en la nueva población. El resto de la población nueva se construye por reproducción (probabilidad 90 %) o cruce (probabilidad 10 %) del resto de la población actual.

La mutación de individuos (algoritmo 8) elige un subcomponente al azar del individuo a mutar y lo sustituye por otro de la misma población, seleccionado también al azar.

El operador de cruce aplicado a individuos crea dos hijos a partir de dos padres y modifica el genotipo de los hijos mediante la elección de dos puntos de corte y el intercambio del segmento comprendido entre ambos puntos de corte (ambos incluidos).

Algoritmo 6 Proceso Evolutivo General

Parámetros: ninguno

```
conf ← Leer configuración
para i= 1 a conf.M
     $PI[i] \leftarrow$  nuevo individuo aleatorio
    Evaluar  $PI[i]$ 
finpara
para s= 1 a conf.K
    para sc= 1 a conf.N[s]
         $PS[s].pob[sc] \leftarrow$  nuevo subcomponente aleatorio
    finpara
    Evaluar  $PS[s]$ 
finpara

para gen= 1 a conf.n_generaciones
     $PI \leftarrow$  Salto generacional( $PI.pob$ )
    para s= 1 a conf.K
         $PS[s] \leftarrow$  Salto generacional( $PS[s].pob$ )
    finpara

    Evaluar  $PI$ 
    para s= 1 a conf.K
        Evaluar  $PS[s].pob$ 
    finpara

    Actualizar estadísticas
finpara

Mostrar resultado final y estadísticas globales
```

Algoritmo 7 Salto generacional de individuos

Parámetros: p: población de individuos

```

pnueva ← [ ]
peor1, peor2 ← índices de los dos peores individuos de PI
r1, r2 ← cruce(p[peor1], p[peor2])
pnueva[peor1] ← r1
pnueva[peor2] ← r2

para j = 1 a conf.M
  si j ≠ peor1 ∧ j ≠ peor2
    si U[0, 100] ≤ 10
      pnueva[j] ← mutacion(p[j])
    si no
      pnueva[j] ← p[j]
  finsi
finsi
finpara

devolver pnueva

```

Algoritmo 8 Mutación de individuos

Parámetros: i: individuo a mutar

```

componente ← U[1, conf.K]
nuevosc ← U[1, conf.N[componente]]
i.scs[componente].sc= nuevosc

```

Algoritmo 9 Cruce de individuos

Parámetros: i₁, i₂: individuos a cruzar

```

hijo1 ← duplicado(i1)
hijo2 ← duplicado(i2)
corte1 ← U[1, conf.K]
corte2 ← U[corte1, conf.K]

para s= corte1 a corte2
  aux ← hijo1.scs[s]
  hijo1.scs[s] ← hijo2.scs[s]
  hijo2.scs[s] ← aux
finpara

devolver(hijo1, hijo2)

```

Se intercambian los punteros a subcomponentes contenidos en el vector *scs* de cada individuo.

Cada población de subcomponentes sigue el mismo algoritmo evolutivo para avanzar una generación (algoritmo 10). La nueva población se crea en cuatro fases: primero se reproducen los subcomponentes que forman parte de los mejores individuos (elitismo); luego se reproducen más subcomponentes ¹ hasta completar el número requerido (según el parámetro *conf.pct_{rep}*); a continuación se cruzan subcomponentes de la población actual (un porcentaje dado por *conf.pct_{cru}*) y sus hijos se transfieren a la nueva población; por último, se mutan un porcentaje (dado por *conf.pct_{mut}*) de subcomponentes y los resultados se insertan en la nueva población.

No se muestra en el algoritmo, pero se intenta posicionar a cada subcomponente en la misma situación que tenía en la población original, dado que las referencias desde la población de individuos están basadas en la posición.

En este algoritmo se usan funciones de duplicación, cruce y mutación de subcomponentes. Estas funciones son métodos del problema, y su implementación deberá ser realizada cuando se aplique el método general a un problema concreto.

La evaluación de una población de subcomponentes (algoritmo 11) consta de dos pasos. En primer lugar se asigna a cada subcomponente de la población uno o más créditos en función de su participación en los individuos de *PI*. Una vez asignados estos créditos, y únicamente en el caso de que sean más de uno, se aplicará un método multiobjetivo que, teniendo en cuenta *toda la población* y no cada subcomponente por separado, asignará un valor único de crédito a cada subcomponente.

La asignación de uno o varios créditos se hace mediante una serie de algoritmos específicos, cada uno de los cuales evalúa la población de subcomponentes y rellena el elemento correspondiente del vector *objetivos* en cada estructura **subcomponente**. Estas funciones son llamadas desde el algoritmo 12.

La asignación de crédito “Mejor”, tal como se dijo al describir el modelo, obtiene la mejor aptitud de los individuos en los que participa el subcomponente evaluado. El algoritmo 13 muestra esta asignación aplicada a una población completa de subcomponentes.

Similar a la anterior, la asignación “Media” evalúa las aptitudes de los individuos que usan al subcomponente, y asigna como crédito la media de dichas aptitudes. El algoritmo 14 muestra el proceso. Debe observarse que en caso de que el subcomponente no participe en ningún individuo se le asigna una aptitud ficticia ($-\infty$).

Algo más compleja de implementar es la asignación de crédito “Sustitución” que, para cada componente a evaluar, busca una serie de individuos que no lo utilicen (con un máximo dado por el parámetro *conf.num_{sustituciones}*) y en cada uno de ellos inserta el subcomponente reemplazando al ya existente (de su misma población). Se asigna al subcomponente como crédito la media de las diferencias (*después – antes*) de aptitud

¹ Seleccionados mediante un método especificado en la configuración, aunque en los experimentos se ha usado un método de torneo. Para el cruce y la mutación se opera igualmente, y en los experimentos se ha determinado mejor el uso de una selección aleatoria.

Algoritmo 10 Salto generacional de subcomponentes

Parámetros: **ps**: población de subcomponentes

```

psnueva ← []
ips ← índice de la población ps

numrep ← conf.pct_rep * conf.N[ips]
nummut ← conf.pct_mut * conf.N[ips]
numcru ← conf.pct_cru * conf.N[ips]

mejores ← mejores conf.elitismo individuos
para i= 1 a conf.elitismo
  componente ← mejores[i].scs[ips].sc
  psnueva[componente] ← duplicado(ps[componente])
  numrep ← numrep - 1
finpara

para i= 1 a numrep
  componente ← selección para reproducción(ps)
  psnueva[componente] ← duplicado(ps[componente])
finpara

para i= 1 a numcru
  componente1, componente2 ← selección para cruce(ps)
  hijo1, hijo2 ← cruce(ps[componente1], ps[componente1])
  psnueva[componente1] ← hijo1
  psnueva[componente2] ← hijo2
finpara

para i= 1 a nummut
  componente ← selección para mutación(ps)
  hijo ← mutación(ps[componente])
  psnueva[componente] ← hijo
finpara

Devolver psnueva

```

Algoritmo 11 Evaluación de subcomponentes

Parámetros: **ps**: población de subcomponentes

```

asignar_crédito(ps)
conversión_multiobjetivo(ps)

```

Algoritmo 12 Asignación de crédito

Parámetros: **ps**: población de subcomponentes

```

para o = 1 a conf.nobjs
  tipo ← conf.creditos[o]
  Asignación_tipo(ps, o)
finpara

```

Algoritmo 13 Asignación de crédito: Mejor

Parámetros: **ps**: población de subcomponentes, **o**: índice del objetivo a asignar

```

ips ← índice de la población ps
para s= 1 a conf.N[ips]
  mejor ←  $-\infty$ 
  para i= 1 a conf.M
    si PI[i].scs[ips].sc = s
      si PI[i].aptitud > mejor
        mejor ← PI[i].aptitud
      finsi
    finsi
  finpara
  ps[s].objetivos[o] ← mejor
finpara

```

Algoritmo 14 Asignación de crédito: Media

Parámetros: **ps**: población de subcomponentes, **o**: índice del objetivo a asignar

```

ips ← índice de la población ps
para s= 1 a conf.N[ips]
  suma ← 0
  cont ← 0
  para i= 1 a conf.M
    si PI[i].scs[ips].sc = s
      suma ← suma + PI[i].aptitud
      cont ← cont + 1
    finsi
  finpara
  si cont ≥ 1
    ps[s].objetivos[o] ← suma/cont
  si no
    ps[s].objetivos[o] ←  $-\infty$ 
  finsi
finpara

```

originadas por los cambios. Se muestra la implementación de este método en el algoritmo 15.

Algoritmo 15 Asignación de crédito: Sustitución

Parámetros: **ps**: población de subcomponentes, **o**: índice del objetivo a asignar

```

ips ← índice de la población ps
para s = 1 a conf.N[ips]
  suma ← 0
  cont ← 0

  para i = 1 a conf.M
    si PI[i].scs[ips].sc ≠ s
      aptorig ← PI[i].aptitud
      PI[i].scs[ips].sc ← s
      Reevaluar PI[i]
      aptnueva ← PI[i].aptitud

      suma ← suma + (aptnueva - aptorig)

      cont ← cont + 1

      si cont = conf.numsubstituciones
        Salir del bucle
      fin si
    fin para
  si cont ≥ 1
    ps[s].objetivos[o] ← suma/cont
  si no
    ps[s].objetivos[o] ←  $-\infty$ 
  fin si
fin para

```

La asignación de crédito por “Diferencia”, que elimina el subcomponente evaluado de todos los individuos en los que participe y evalúa la media de las diferencias de aptitud (*antes* - *después*), se muestra en el algoritmo 16. Este método sólo se puede aplicar si el problema permite una evaluación de individuos parciales, es decir, aquellos a los que les falte algún subcomponente. Se indica que un subcomponente de la población *p* no está presente en un individuo *i* fijando un valor especial en la referencia *i.scs*[*p*].*sc*. En este caso se ha usado el valor $-\infty$.

El criterio de asignación de crédito por unión (algoritmo 17) no utiliza la población de individuos, sino que para cada subcomponente a evaluar crea un individuo desde cero, uniendo ese subcomponente a los mejores del resto de las poblaciones. La aptitud del individuo sintético así generado se asigna como crédito.

Algoritmo 16 Asignación de crédito: Diferencia

Parámetros: **ps**: población de subcomponentes, **o**: índice del objetivo a asignar

```

ips ← índice de la población ps
para s = 1 a conf.N[ips]
  suma ← 0
  cont ← 0

  para i = 1 a conf.M
    si PI[i].scs[ips].sc = s
      aptorig ← PI[i].aptitud
      PI[i].scs[ips].sc ←  $-\infty$ 
      Reevaluar PI[i]
      aptnueva ← PI[i].aptitud

      suma ← suma + (aptorig - aptnueva)

    cont ← cont + 1
  finsi
finpara
si cont ≥ 1
  ps[s].objetivos[o] ← suma/cont
si no
  ps[s].objetivos[o] ←  $-\infty$ 
finsi
finpara

```

Algoritmo 17 Asignación de crédito: Unión

Parámetros: **ps**: población de subcomponentes, **o**: índice del objetivo a asignar

```

iunion ← [ ]
ips ← índice de la población ps
para s = 1 a conf.N[ips]
  para p = 1 a conf.K
    iunion.scs[p].pob ← p
    si p ≠ ips
      m ← índice del mejor subcomponente de PS[p]
      iunion.scs[p].pob ← m
    si no
      iunion.scs[p].pob ← s
    fin si
  fin para
  Evaluar iunion
  ps[s].objetivos[o] ← iunion.aptitud
fin para

```

Cada subcomponente de cada población, por tanto, recibe uno o más créditos, que se almacenan en su vector *objetivos*. Para obtener un valor único de crédito se aplica un algoritmo multiobjetivo. En la implementación realizada se proporcionan los algoritmos NSGA y NSGA-II.

Algoritmo 18 Conversión multiobjetivo

Parámetros: **ps**: población de subcomponentes

```

si conf.nobjs = 1
  ips ← índice de la población ps
  para s = 1 a conf.N[ips]
    ps[s].crédito ← ps[s].objetivos[1]
  fin para
si no
  Aplicar NSGA-II a ps
fin si

```

4.5. Adaptación al problema

Esta implementación únicamente codifica las entidades y los métodos del modelo. Como se recordará del capítulo 3, el modelo propuesto consta de entidades y métodos, y tanto unas como otros pueden ser propias del modelo (y por tanto completamente especificadas en cuanto a su contenido y funcionalidad) o del problema. En este segundo

Nombre	Tipo	Pertenencia
Subcomponentes	entidad	problema
Creación de poblaciones iniciales: subcomponentes	método	problema
Operadores genéticos sobre subcomponentes	método	problema
Evaluación de individuos	método	problema

Cuadro 4.1: Entidades y métodos del problema.

caso, se trata de entidades y métodos que, por su especificidad, deben ser descritas e implementadas cuando se trate con un problema concreto.

Así, por ejemplo, la entidad “individuo” es una entidad del problema, puesto que un individuo consta de una serie de referencias a subcomponentes de distintas poblaciones y de una aptitud. El método “evolución de subcomponentes” también es del modelo, ya que las funciones a realizar están definidas de forma independiente del problema al que se aplique el modelo.

Por otro lado, la entidad “subcomponente” no está completamente definida, y es por tanto una entidad del problema. Se necesita información sobre el genotipo de un subcomponente para poder especificar completamente esta entidad. Si el problema, por ejemplo, es el de cobertura de vectores (que se tratará en el capítulo 5), el genotipo de un subcomponente será un vector binario de longitud fija. El método “cruce de subcomponentes” tampoco puede definirse hasta el momento en que el modelo se aplique a un problema. En el problema de cobertura de vectores se podrá definir dicho método como un cruce de dos puntos o de otro tipo, siempre en función del tipo de genotipo y de las características propias del problema a resolver.

En la tabla 3.1 (página 35) se indicó qué entidades y qué métodos pertenecían al modelo y cuáles al problema. Transferimos parcialmente dicha información a la tabla 4.1 para enumerar a continuación las estructuras de datos y las funciones que hay que implementar para adaptar el método a un problema.

Estructuras de datos de los subcomponentes. Al definir al inicio de este capítulo la estructura de datos de un subcomponente se vio que constaba de un vector de objetivos (indicando los valores obtenidos en cada criterio de asignación de crédito), un crédito (que se obtiene a partir de dicho vector por un algoritmo multiobjetivo) y una estructura de nombre *usr* y de tipo **DATOS**. Esa estructura debe ser definida por el problema, ya que contiene el genotipo del subcomponente. La estructura será utilizada por los métodos que se reseñarán a continuación.

Creación de subcomponentes aleatorios. Durante la iniciación del proceso evolutivo se crean las poblaciones iniciales. La creación de la población de individuos *PI* es un método del modelo, pero la creación de las *K* poblaciones de subcomponentes *PS*₁, *PS*₂, . . . *PS*_{*K*} se realiza mediante una repetida generación de subcomponentes aleatorios. Esta creación de subcomponentes implica almacenar valores

aleatorios, aunque válidos (pueden tener que cumplir algunas restricciones) en la estructura *usr* del subcomponente.

Duplicación, cruce y mutación de subcomponentes. Los operadores genéticos que se aplican a un subcomponente dependen claramente del genotipo del mismo y de las características específicas del problema.

Evaluación de individuos. Aunque la estructura de datos **individuo** haya sido definida completamente más arriba (y de hecho sea una entidad del modelo) su contenido no proporciona directamente el genotipo del individuo. Por el contrario, se deben unir los genotipos de los subcomponentes que forman el individuo para obtenerlo. El genotipo resultante tiene un contenido y un formato dependientes del problema, y sólo puede ser evaluado por una función específica que lo aplique al problema y calcule la aptitud del individuo.

Si el problema lo permite, la evaluación de individuos puede aceptar individuos incompletos (a los que les falte algún subcomponente). En ese caso se puede aplicar el criterio de asignación de crédito Diferencia.

Existen otras funciones que deben ser implementadas para cada problema. No se han incluido en la enumeración anterior porque se trata de funciones auxiliares que ni siquiera pertenecen al modelo: reserva y liberación de memoria, generación de informes por generación y finales, representación textual de individuos y subcomponentes para su examen. . .

4.6. Configuración

Como se puede ver en el algoritmo 6, al inicio del Proceso Evolutivo General se lee la configuración de un fichero. En ese fichero se especifican todos los parámetros que guiarán el experimento, tanto valores del proceso evolutivo (porcentaje de subcomponentes a mutar, cruzar y reproducir, número de generaciones. . .) como parámetros relativos a la depuración y traza del proceso, y por supuesto valores específicos para describir el problema a resolver.

De forma muy breve, y únicamente a modo ilustrativo, se reproducen a continuación los ficheros de configuración de uno de los experimentos realizados en el capítulo 6. Las directivas de configuración están en inglés, al igual que el código (variables y comentarios) de todos los programas desarrollados en este trabajo, para facilitar la distribución de la implementación y su uso por otros investigadores.

El fichero principal de configuración “gr-5000-500.conf”

```
gr-5000-500.conf
# Configuración de depuración y trazo
include = deb-trace.confh
```

```

# Poblaciones y parámetros del proceso evolutivo
include = pops5000-500.confh

# Parámetros del problema
include = prob-aureusgr.confh

# Criterios de asignación de crédito:
# media, sustitución y diferencia
obj = OBJ_IND_AVERAGE
obj = OBJ_SUBSTITUTION
obj = OBJ_DIFFERENCE

# Agregación multiobjetivo de criterios por NSGA-II
mo_evaluation = MOE_NSII

```

referencia a otro fichero de control de depuración “deb-trace.confh”,

```

----- deb-trace.confh -----
# Nivel de depuración ligero
deb_level = 1

# Volcado de poblaciones deshabilitado
trace_show_pops_interval = 0

# Volcado completo de poblaciones
trace_show_pops_full = 1

```

a un fichero de descripción de poblaciones y otros parámetros “pops5000-500.confh”

```

----- pops5000-500.confh -----
# Número de subcomponentes
n_scs = 4

# Tamaño de las poblaciones de subcomponentes y de individuos
scpop_size_1 = 150
scpop_size_2 = 150
scpop_size_3 = 150
scpop_size_4 = 150
indpop_size = 500

# Número de generaciones
max_generations = 5000

# Porcentajes de reproducción, cruce y
# mutación de subcomponentes

```

```
sub_rep_pct = 50
sub_mut_pct = 25
sub_xvr_pct = 25

# Tipo de cruce de subcomponentes
sc_xvr = XVR_BLX_ALPHA

# Selección para reproducción, cruce y
# mutación de subcomponentes
sub_rep_selection = SEL_TOURNAMENT
sub_mut_selection = SEL_RANDOM
sub_xvr_selection = SEL_RANDOM

# Tamaño de torneo para subcomponentes
sc_tournament_size= 2

# Población de individuos evoluciona
# en estado estacionario
ind_steady= 1

# Tipo de cruce de individuos
ind_xvr = XVR_TWO_POINT

# Tamaño de torneo para individuos
ind_tournament_size= 2

# Número de subcomponentes elitistas
elitism = 50
```

y, por último, a un fichero de descripción del problema “prob-aureusgr.confh”.

```
_____ prob-aureusgr.confh _____
# Número de variables y grado de la superficie de respuesta
l_nvars= 4
l_grade= 3

# Fichero de datos
s_datafile= data/aureus-an-grate-typ.txt

# Número de muestras para entrenamiento y para generalización
l_nsamples_train= 100
l_nsamples_gener= 50
```

5

Problema: cobertura de vectores

5.1. Introducción

El primer problema al que se ha aplicado el modelo coevolutivo cooperativo propuesto es el problema de cobertura de vectores. Este problema plantea la obtención de un conjunto pequeño de vectores (normalmente binarios) llamado “conjunto de cobertura” que debe “encajar” lo mejor posible con un conjunto de vectores grande, llamado “conjunto objetivo”. El significado del término “encaje” puede variar según los autores [Balthrop et al., 2002] [Smith et al., 1993], aunque en este trabajo se usará el más aplicado: una medida de similaridad entre vectores.

Este problema ha sido abordado desde la computación evolutiva estándar, generando una población de vectores y tratando de adaptarlos al conjunto objetivo, y desde la coevolución cooperativa, haciendo que cada población de subcomponentes evolucione un vector de cobertura y mejorándolos en cooperación. En este trabajo se coevolucionarán vectores de cobertura de forma cooperativa y también se evolucionarán en paralelo las posibles combinaciones de dichos vectores para formar un conjunto de cobertura. Todo esto se realizará aplicando al problema el modelo genérico descrito en el capítulo 3.

5.2. Problema de cobertura de vectores

5.2.1. Descripción

El problema de la cobertura de vectores consiste en encontrar un conjunto de vectores binarios que se ajuste de la mejor forma posible a otro conjunto mayor de vectores objetivo. Este problema ha sido abordado por varios autores [Potter y de Jong, 2000] [Forrest et al., 1993]. Forrest propone un modelo simplificado del sistema inmune, basándose en un modelo previo de Farmer [Farmer et al., 1986], en el cual una población de anticuerpos es evolucionada para ajustarse a un conjunto de antígenos.

Posteriormente Potter y de Jong [Potter y de Jong, 2000] aplican su método coevolutivo al problema de cobertura de vectores, estudiando la adaptabilidad de la población a

múltiples nichos en el conjunto objetivo, la especialización de subcomponentes y la descomposición emergente del problema.

La descripción del problema de cobertura de vectores es la siguiente: sea T un conjunto de N_T vectores binarios, y C otro conjunto de N_C vectores binarios, todos de igual longitud (L). Típicamente $N_T \gg N_C$. El conjunto T recibe el nombre de conjunto objetivo (*target set*) y el conjunto C se denomina conjunto de cobertura o de encaje (*match set*).

Se define una función $S : T \times C \rightarrow \mathbb{N}$ cuyo cometido es calcular la cobertura (*match strength*) de dos vectores:

$$S(x, y) = \sum_{i=1}^L m_i$$

donde $x \in T, y \in C$ y el valor m_i se define así:

$$m_i = \begin{cases} 1 & \text{si } x_i = y_i \\ 0 & \text{en otro caso} \end{cases}$$

La función S , como se ve, calcula el número de componentes iguales entre dos vectores binarios. Esta función se utiliza en la función S_T para calcular la cobertura del conjunto C sobre el conjunto T , que se define como la media de las mejores coberturas de C sobre cada vector de T :

$$S_T(C) = \frac{1}{N_T} \sum_{i=1}^{N_T} \max(S(m_1, t_i), \dots, S(m_{N_C}, t_i)) \quad (5.1)$$

donde $m_x (x = 1, \dots, N_C) \in C$ y $t_i \in T$.

El objetivo del problema de cobertura de vectores es encontrar, dado un conjunto objetivo T , un conjunto de cobertura C que maximice $S_T(C)$. En el ejemplo de la figura 5.1 se ve cómo los $N_C = 2$ vectores del conjunto C se emparejan con los $N_T = 4$ vectores del conjunto T y se calcula la cobertura de cada uno de los primeros sobre cada uno de los segundos ($S(m_j, t_i)$). El valor de cobertura del conjunto C se calcula como

$$S_T(C) = \frac{1}{4} (\max(S(m_1, t_1), S(m_1, t_2)) + \max(S(m_2, t_1), S(m_2, t_2)) + \max(S(m_3, t_1), S(m_3, t_2)) + \max(S(m_4, t_1), S(m_4, t_2)))$$

es decir,

$$\begin{aligned} S_T(C) &= \frac{1}{4} (\max(6, 8) + \max(9, 11) + \max(3, 9) + \max(10, 7)) \\ &= \frac{1}{4} (8 + 11 + 9 + 10) = 9,5 \end{aligned}$$

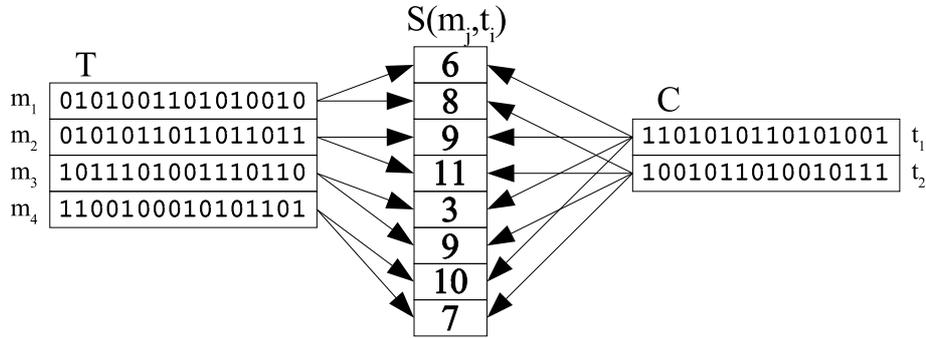


Figura 5.1: Ejemplo de cobertura de vectores. $N_T = 4, N_C = 2, L = 16$

5.2.2. Generación coevolutiva de coberturas

La aplicación de la computación evolutiva a este problema parte de la creación de un conjunto objetivo T aleatorio que puede ser accedido por todos los individuos de la población (este conjunto, de hecho, es el entorno al que deben adaptarse como meta del proceso evolutivo).

Cada individuo puede representar un anticuerpo o un conjunto de ellos (un conjunto C candidato). En el primer método (aplicado por Forrest) cada individuo de la población representa un anticuerpo, y su genotipo es un vector de bits. Para calcular la aptitud de un individuo i se elige una muestra de ρ antígenos del conjunto T :

$$\Gamma = \{\gamma_1, \gamma_2, \dots, \gamma_\rho\} \subset T$$

A continuación se evalúa la aptitud de i calculando la cobertura de i sobre cada antígeno del conjunto Γ y hallando la media:

$$aptitud(i) = \frac{1}{\rho} \sum_{p=1}^{\rho} S(\gamma_p, i)$$

También se puede considerar a cada individuo como un conjunto de N vectores y su aptitud se calcula aplicando la función S_T a dicho conjunto. Potter, por ejemplo, evoluciona una serie de poblaciones de vectores de cobertura, de forma independiente. Las poblaciones se unen únicamente para aportar representantes durante la fase de evaluación¹. El resultado final del algoritmo es una serie de poblaciones de subcomponentes (vectores de cobertura). De cada una de ellas se escogerá el mejor, y la unión de estos vectores será el conjunto C , la solución al problema.

¹La evaluación de un subcomponente en el método propuesto por Potter, como se explicó en el capítulo 2, consiste en la unión del subcomponente con representantes de las demás poblaciones para formar un individuo efímero, cuya única finalidad es proporcionar un valor de aptitud.

Aplicando el método propuesto en esta memoria, se abordará el problema de cobertura de vectores creando un algoritmo coevolutivo cooperativo con los siguientes características ²:

- $K = N_C$, es decir, habrá tantas poblaciones de subcomponentes como vectores deba tener el conjunto C
- Las poblaciones PS_1, PS_2, \dots, PS_K evolucionarán vectores binarios de longitud L . Cada una de ellas aportará un único vector a cada conjunto C candidato (es decir, a cada individuo).
- Los integrantes de las poblaciones PS_p , es decir, los subcomponentes, tendrán como genotipo un vector binario de longitud L .
- Los tamaños de las poblaciones de subcomponentes serán iguales: $N_1 = N_2 = \dots = N_K$.
- La población PI evolucionará combinaciones de vectores para tratar de encontrar conjuntos C que resuelvan el problema lo mejor posible. Constará de M individuos i_j ($1 \leq j \leq M$).
- Cada individuo $i_j \in PI$ estará compuesto de un vector tomado de cada una de las poblaciones PS_p .
- La aptitud del individuo i_j se calculará aplicando el conjunto C_{i_j} de los vectores que lo forman al conjunto objetivo T y midiendo la cobertura obtenida mediante la función $S_T(C_{i_j})$ (ver ecuación 5.1)

Tal como se describió en la sección 4.5, para adaptar el modelo genérico a este problema se implementarán estructuras de datos y funciones que realicen una serie de tareas. A continuación se enumeran unas y otras y se describe su aplicación al problema de cobertura de vectores.

Estructura de datos de subcomponentes. Cada subcomponente tiene como genotipo un vector de valores binarios de longitud L .

Creación de subcomponentes aleatorios. Cuando se crea un subcomponente (durante la fase de iniciación del proceso evolutivo) se rellena su genotipo con L valores aleatorios enteros, cada uno con distribución uniforme en el intervalo $[0, 1]$.

Duplicación, cruce y mutación de subcomponentes. La duplicación de subcomponentes es trivial. Como operador de cruce se ha implementado un cruce de dos puntos. El operador de mutación elige al azar uno de los L valores del vector binario y aplica una negación lógica al mismo, invirtiendo su valor.

²Recordemos que M es el tamaño de la población de individuos PI , K es el número de subcomponentes y $N_1, N_2 \dots N_K$ son los respectivos tamaños de las poblaciones de subcomponentes PS_1, PS_2, \dots, PS_K .


```

*****1111111*****
*****1111111*****
*****1111111*****

```

El modelo propuesto, adaptado al problema, ha sido aplicado a tres instancias del mismo.

- *Medio*: el conjunto objetivo T está compuesto de 200 vectores de 64 bits, generados a partir de los dos esquemas mostrados anteriormente. Hay dos poblaciones de subcomponentes, cada una evolucionando un vector del conjunto C .
- *Cuarto*: 200 vectores de 64 bits forman el conjunto T . Cada uno de ellos sigue uno de los cuatro esquemas mostrados para este problema. El número de poblaciones de subcomponentes es 4.
- *Octavo*: 8 poblaciones de subcomponente evolucionan vectores para formar el conjunto C y adaptarlo a un conjunto T de 200 vectores de longitud 64, generados aleatoriamente en función de los esquemas del problema “octavo”.

El principal objetivo de las pruebas es mostrar que al introducir múltiples criterios de asignación de crédito se produce una mejora en los resultados finales. Para ello se ha realizado un diseño de experimentos estableciendo cinco factores (uno por cada criterio de asignación de crédito propuesto por el modelo). Para cada factor se han fijado dos niveles (presente y no-presente). Para un experimento determinado, el hecho de que el factor X tenga el nivel “presente” significa que durante el proceso evolutivo de dicho experimento se ha tenido en cuenta el criterio X para la asignación de crédito a los subcomponentes.

Los factores son los siguientes: Mejor, Media, Sustitución, Diferencia y Unión. Dados dos niveles por cada uno, el total de combinaciones es de 32. Todas ellas se muestran en la tabla 5.1.

Para cada combinación de criterios (es decir, para cada experimento) se lanzaron 30 réplicas de la ejecución. De cada réplica se tomó el mejor individuo de la última generación. Como resultado de cada experimento se considera la media de aptitud de esos 30 individuos.

Para la combinación número 32, en la que todos los factores tienen el nivel “no-presente”, y que se considera como grupo de control, se asignó el mismo crédito a todos los subcomponentes de todas las poblaciones en cada generación, con lo que el proceso evolutivo se convirtió en una búsqueda aleatoria.

Los parámetros del proceso evolutivo se muestran en la tabla 5.2. Los tamaños de la población de individuos y de las poblaciones de subcomponentes se escogieron experimentalmente, para albergar suficientes combinaciones de subcomponentes y proporcionar variedad en la evolución, respectivamente. Las proporciones de reproducción, cruce y mutación de las poblaciones de subcomponentes fueron escogidas tras la preparación y ejecución de un diseño de experimentos en el que se probaron diferentes combinaciones de valores.

Comb.	Mj.	Md.	S.	D.	U.	Comb.	Mj.	Md.	S.	D.	U.
1	✓	–	–	–	–	17	✓	✓	–	✓	–
2	–	✓	–	–	–	18	✓	✓	–	–	✓
3	–	–	✓	–	–	19	✓	–	✓	✓	–
4	–	–	–	✓	–	20	✓	–	✓	–	✓
5	–	–	–	–	✓	21	✓	–	–	✓	✓
6	✓	✓	–	–	–	22	–	✓	✓	✓	–
7	✓	–	✓	–	–	23	–	✓	✓	–	✓
8	✓	–	–	✓	–	24	–	✓	–	✓	✓
9	✓	–	–	–	✓	25	–	–	✓	✓	✓
10	–	✓	✓	–	–	26	✓	✓	✓	✓	–
11	–	✓	–	✓	–	27	✓	✓	✓	–	✓
12	–	✓	–	–	✓	28	✓	✓	–	✓	✓
13	–	–	✓	✓	–	29	✓	–	✓	✓	✓
14	–	–	✓	–	✓	30	–	✓	✓	✓	✓
15	–	–	–	✓	✓	31	✓	✓	✓	✓	✓
16	✓	✓	✓	–	–	32	–	–	–	–	–

Cuadro 5.1: Combinaciones de criterios para el diseño experimental. ✓ significa “presente”, – significa “no-presente”.

Parámetro		Valor
Número de generaciones		500
Poblaciones de subcomponentes	Tamaño (M)	50
	Reproducción	50 %, selección por torneo
	Mutación	25 %, selección aleatoria
	Cruce	dos puntos, 25 %, selección aleatoria
	Elitismo	subcomponentes de los 5 mejores individuos
Población de individuos	Tamaño (N_p)	200
	Evolución	estado estacionario
	Mutación	10 %
	Cruce	dos puntos. Dos individuos (selección por torneo) se cruzan y substituyen a los dos peores de la población.

Cuadro 5.2: Parámetros del proceso evolutivo. Cobertura de vectores.

5.2.4. Resultados

Forrest establece la máxima cobertura que puede tener un único vector sobre un conjunto de esquemas de longitud s , siendo ℓ el tamaño de los vectores. Este vector es lo que se llama un generalista: aporta una cobertura aceptable a todos los vectores de T , pero no tiene una aptitud destacada con ninguno de ellos.

Así, por ejemplo, para los esquemas del problema “medio” el vector de cobertura óptimo encaja la mitad de los valores con probabilidad 1 y la restante mitad con probabilidad $1/2$ (ya que estos bits restantes son elegidos al azar durante la creación del vector objetivo). Por lo tanto, su aptitud esperada (número de bits encajados) será $((1/2)(1) + (1/2)(1/2))64 = 48$. Para los esquemas “cuarto” la aptitud máxima será $((1/4)(1) + (3/4)(1/2))64 = 40$ y para los “octavo”, $((1/8)(1) + (7/8)(1/2))64 = 36$. En general, para una longitud (L o ℓ) determinada de los vectores y para unos esquemas con cadenas de unos de longitud s , la aptitud máxima del individuo generalista será:

$$[(s/\ell)(1) + (1 - \frac{s}{\ell})(1/2)]\ell = \frac{\ell + s}{2}$$

Estos valores (48, 40 y 36) pueden tomarse como referencia, pero siempre teniendo en cuenta que en nuestra aplicación se están generando N_C vectores en lugar de uno, con lo que algunos de ellos pueden derivar a vectores especialistas y cubrir mejor algunas partes del conjunto T , con lo cual la aptitud del conjunto (según la ecuación 5.1) sería superior a la de un único vector generalista.

En la tabla 5.3 se muestran los resultados promedio de las 30 ejecuciones para cada uno de los 32 experimentos sobre el problema “medio”. La tabla 5.4 muestra los resultados para el problema “cuarto” y en la tabla 5.5 pueden verse las aptitudes finales medias para el problema “octavo”.

Para poder hacer contrastes de diferencia de medias a partir de las aptitudes obtenidas se ha realizado un análisis de varianza (ANOVA [Kendall y Stuart, 1977]) de cinco factores, considerando interacciones.

Una vez terminado el análisis y en función del test de Levene de igualdad de varianzas se hicieron contrastes de comparación entre las medias con y sin cada criterio. Se usaron contrastes t de Student, suponiendo las varianzas poblacionales iguales.

Las tablas 5.6, 5.7 y 5.8 muestran los resultados medios de todos los experimentos con y sin cada criterio. Y en las tablas 5.9, 5.10 y 5.11 se muestran las pruebas de efectos inter-sujetos, con los niveles de significación del test T de diferencia de medias considerando o sin considerar los criterios y sus combinaciones dos a dos, tres a tres, cuatro a cuatro y totales.

Combinación	Mejor	Media	Sust.	Dif.	Unión	Aptitud	Desv. Est.
1	✓	—	—	—	—	49.27475	.125126
2	—	✓	—	—	—	49.30225	.130510
3	—	—	✓	—	—	49.30650	.138593
4	—	—	—	✓	—	49.26725	.166934
5	—	—	—	—	✓	49.33400	.131955
6	✓	✓	—	—	—	49.25350	.132507
7	✓	—	✓	—	—	49.36775	.186473
8	✓	—	—	✓	—	49.31300	.155059
9	✓	—	—	—	✓	49.34875	.127195
10	—	✓	✓	—	—	49.28250	.100884
11	—	✓	—	✓	—	49.24825	.125366
12	—	✓	—	—	✓	49.29000	.144213
13	—	—	✓	✓	—	49.32100	.138769
14	—	—	✓	—	✓	49.30850	.129280
15	—	—	—	✓	✓	49.26525	.159402
16	✓	✓	✓	—	—	49.29150	.121093
17	✓	✓	—	✓	—	49.27450	.117506
18	✓	✓	—	—	✓	49.25875	.088806
19	✓	—	✓	✓	—	49.29325	.134480
20	✓	—	✓	—	✓	49.28800	.150739
21	✓	—	—	✓	✓	49.27475	.098214
22	—	✓	✓	✓	—	49.30275	.160177
23	—	✓	✓	—	✓	49.28825	.147463
24	—	✓	—	✓	✓	49.28125	.139462
25	—	—	✓	✓	✓	49.31350	.152247
26	✓	✓	✓	✓	—	49.33200	.153369
27	✓	✓	✓	—	✓	49.31675	.177366
28	✓	✓	—	✓	✓	49.26100	.137061
29	✓	—	✓	✓	✓	49.32400	.168450
30	—	✓	✓	✓	✓	49.25775	.166903
31	✓	✓	✓	✓	✓	49.32775	.213877
32	—	—	—	—	—	49.25250	.090822

Cuadro 5.3: Resultados para problema “half”.

Combinación	Mejor	Media	Sust.	Dif.	Unión	Aptitud	Desv. Est.
1	✓	—	—	—	—	43.73825	.155895
2	—	✓	—	—	—	43.77625	.198593
3	—	—	✓	—	—	43.79875	.179992
4	—	—	—	✓	—	43.68200	.207595
5	—	—	—	—	✓	43.81525	.152923
6	✓	✓	—	—	—	43.76000	.132387
7	✓	—	✓	—	—	43.78625	.141643
8	✓	—	—	✓	—	43.76400	.221143
9	✓	—	—	—	✓	43.71975	.197114
10	—	✓	✓	—	—	43.86375	.126635
11	—	✓	—	✓	—	43.75800	.112734
12	—	✓	—	—	✓	43.77900	.198823
13	—	—	✓	✓	—	43.76875	.160385
14	—	—	✓	—	✓	43.76850	.155590
15	—	—	—	✓	✓	43.73350	.192942
16	✓	✓	✓	—	—	43.74375	.114855
17	✓	✓	—	✓	—	43.79000	.149701
18	✓	✓	—	—	✓	43.72025	.175420
19	✓	—	✓	✓	—	43.79050	.116550
20	✓	—	✓	—	✓	43.78875	.153879
21	✓	—	—	✓	✓	43.75225	.107526
22	—	✓	✓	✓	—	43.73875	.130766
23	—	✓	✓	—	✓	43.75875	.144029
24	—	✓	—	✓	✓	43.75200	.153592
25	—	—	✓	✓	✓	43.69325	.160470
26	✓	✓	✓	✓	—	43.78400	.146957
27	✓	✓	✓	—	✓	43.78175	.156905
28	✓	✓	—	✓	✓	43.86275	.152380
29	✓	—	✓	✓	✓	43.74750	.152880
30	—	✓	✓	✓	✓	43.75050	.148492
31	✓	✓	✓	✓	✓	43.78425	.151608
32	—	—	—	—	—	43.43625	.189465

Cuadro 5.4: Resultados para problema “quarter”.

Combinación	Mejor	Media	Sust.	Dif.	Unión	Aptitud	Desv. Est.
1	✓	—	—	—	—	42.17100	.147440
2	—	✓	—	—	—	42.22275	.133500
3	—	—	✓	—	—	42.40925	.155583
4	—	—	—	✓	—	42.40900	.145300
5	—	—	—	—	✓	42.45925	.120823
6	✓	✓	—	—	—	42.30000	.214709
7	✓	—	✓	—	—	42.41850	.175395
8	✓	—	—	✓	—	42.39975	.104119
9	✓	—	—	—	✓	42.39800	.133174
10	—	✓	✓	—	—	42.42500	.181898
11	—	✓	—	✓	—	42.37825	.149167
12	—	✓	—	—	✓	42.40325	.138177
13	—	—	✓	✓	—	42.41125	.170069
14	—	—	✓	—	✓	42.44025	.133658
15	—	—	—	✓	✓	42.42950	.108104
16	✓	✓	✓	—	—	42.40075	.129800
17	✓	✓	—	✓	—	42.37400	.114784
18	✓	✓	—	—	✓	42.45800	.138786
19	✓	—	✓	✓	—	42.46100	.154380
20	✓	—	✓	—	✓	42.51525	.069462
21	✓	—	—	✓	✓	42.39500	.110048
22	—	✓	✓	✓	—	42.44025	.127893
23	—	✓	✓	—	✓	42.41575	.106218
24	—	✓	—	✓	✓	42.38175	.188779
25	—	—	✓	✓	✓	42.47775	.184772
26	✓	✓	✓	✓	—	42.36900	.138323
27	✓	✓	✓	—	✓	42.45650	.152696
28	✓	✓	—	✓	✓	42.45800	.149186
29	✓	—	✓	✓	✓	42.49225	.152415
30	—	✓	✓	✓	✓	42.43825	.193454
31	✓	✓	✓	✓	✓	42.49650	.121103
32	—	—	—	—	—	41.37225	.147537

Cuadro 5.5: Resultados para problema “eighth”.

Criterio	Con criterio		Sin criterio	
	Aptitud	Desv. Est.	Aptitud	Desv. Est.
Mejor	49.30000	.146670	49.28884	.139356
Media	49.28555	.142850	49.30330	.142935
Sust.	49.30761	.153032	49.28123	.131251
Dif.	49.29108	.150219	49.29777	.135670
Unión	49.29614	.147806	49.29270	.138355

Cuadro 5.6: Resultados medios considerando/sin considerar cada criterio. Problema “medio”.

Criterio	Con criterio		Sin criterio	
	Aptitud	Desv. Est.	Aptitud	Desv. Est.
Mejor	43.76963	.154319	43.74208	.184775
Media	43.77523	.152794	43.73647	.185032
Sust.	43.77173	.148019	43.73997	.189528
Dif.	43.75950	.158528	43.75220	.182150
Unión	43.76300	.162135	43.74870	.178735

Cuadro 5.7: Resultados medios considerando/sin considerar cada criterio. Problema “cuarto”.

Criterio	Con criterio		Sin criterio	
	Aptitud	Desv. Est.	Aptitud	Desv. Est.
Mejor	42.41022	.160596	42.34461	.296753
Media	42.40112	.161568	42.35370	.297960
Sust.	42.44172	.151319	42.31311	.291237
Dif.	42.42572	.149490	42.32911	.298302
Unión	42.44470	.143042	42.31012	.294054

Cuadro 5.8: Resultados medios considerando/sin considerar cada criterio. Problema “octavo”.

Combinación	Significación test T
M	0.326
P	0.118
S	0.020
D	0.556
U	0.762
M * P	0.770
M * S	0.434
P * S	0.836
M * P * S	0.124
M * D	0.552
P * D	0.543
M * P * D	0.302
S * D	0.404
M * S * D	0.580
P * S * D	0.956
M * P * S * D	0.775
M * U	0.758
P * U	0.714
M * P * U	0.514
S * U	0.270
M * S * U	0.622
P * S * U	0.444
M * P * S * U	0.624
D * U	0.413
M * D * U	0.789
P * D * U	0.821
M * P * D * U	0.461
S * D * U	0.296
M * S * D * U	0.198
P * S * D * U	0.027
M * P * S * D * U	0.727

Cuadro 5.9: Pruebas de efectos inter-sujetos. Problema “medio”. Criterios: Mejor (M), Media (P, promedio), Diferencia (D), Sustitución (S), Unión (U).

Combinación	Significación test T
M	0.030
P	0.002
S	0.012
D	0.564
U	0.259
M * P	0.092
M * S	0.127
P * S	0.015
M * P * S	0.496
M * D	0.079
P * D	0.831
M * P * D	0.034
S * D	0.004
M * S * D	0.514
P * S * D	0.469
M * P * S * D	0.287
M * U	0.261
P * U	0.168
M * P * U	0.006
S * U	0.002
M * S * U	0.002
P * S * U	0.023
M * P * S * U	0.034
D * U	0.259
M * D * U	0.138
P * D * U	0.003
M * P * D * U	0.070
S * D * U	0.311
M * S * D * U	0.003
P * S * D * U	0.209
M * P * S * D * U	0.772

Cuadro 5.10: Pruebas de efectos inter-sujetos. Problema “cuarto”. Criterios: Mejor (M), Media (P, promedio), Diferencia (D), Sustitución (S), Unión (U).

Combinación	Significación test T
M	0.000
P	0.000
S	0.000
D	0.000
U	0.000
M * P	0.001
M * S	0.000
P * S	0.000
M * P * S	0.063
M * D	0.000
P * D	0.000
M * P * D	0.000
S * D	0.000
M * S * D	0.000
P * S * D	0.000
M * P * S * D	0.000
M * U	0.001
P * U	0.000
M * P * U	0.000
S * U	0.000
M * S * U	0.000
P * S * U	0.000
M * P * S * U	0.000
D * U	0.000
M * D * U	0.000
P * D * U	0.000
M * P * D * U	0.002
S * D * U	0.000
M * S * D * U	0.000
P * S * D * U	0.000
M * P * S * D * U	0.000

Cuadro 5.11: Pruebas de efectos inter-sujetos. Problema “octavo”. Criterios: Mejor (M), Media (P, promedio), Diferencia (D), Sustitución (S), Unión (U).

6

Problema: evolución de superficies de respuesta

6.1. Introducción a las superficies de respuesta

6.1.1. Concepto

La metodología basada en superficies de respuesta es una colección de técnicas estadísticas y matemáticas usadas para desarrollar, mejorar y optimizar procesos [Rawlings et al., 1998] [Myers y Montgomery, 2002].

En esta metodología la relación existente entre la respuesta de un fenómeno (variable dependiente) y un conjunto de medidas experimentales obtenidas directa o indirectamente (variables de predicción) es de tipo polinomial. Este polinomio, denominado superficie de respuesta, suele presentar grado 1 ó 2 en las variables de predicción, aunque puede presentar un grado superior [Myers y Montgomery, 2002].

Supongamos que y es la variable que mide la respuesta de un fenómeno, y que hay un conjunto de variables de predicción x_1, x_2, \dots, x_k . En algunos sistemas de la naturaleza la relación entre y y x_i puede ser conocida exactamente. Entonces se podría construir un modelo de la forma $y = g(x_1, x_2, \dots, x_k) + \epsilon$, donde ϵ representa el error en el sistema. Este tipo de modelos son llamados modelos mecanicistas. Interesa más considerar la situación más común donde la relación subyacente entre y (la variable dependiente o de respuesta) y un conjunto de variables independientes (de predicción) x_1, x_2, \dots, x_k es desconocida, siendo necesario encontrar una función f tal que $y = f(x_1, x_2, \dots, x_k) + \epsilon$ siendo f un modelo de superficie de respuesta [Myers y Montgomery, 2002].

Los modelos de superficie de respuesta son empleados en multitud de aplicaciones industriales y explican una gran variedad de fenómenos, particularmente en situaciones donde bastantes variables de entrada influyen potencialmente en la medida del rendimiento de un producto o proceso. Esta medida (o medidas) es llamada respuesta.

En la figura 6.1 se puede observar gráficamente la relación entre la variable de respuesta (y) en el rendimiento de un proceso químico y dos variables independientes: el tiempo

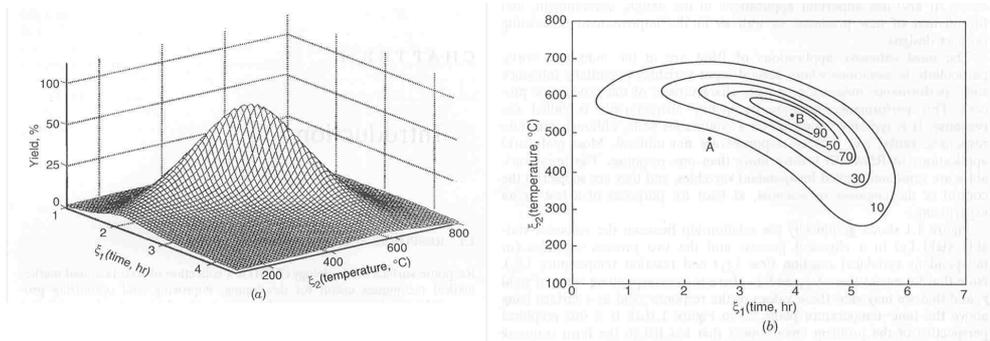


Figura 6.1: (a) superficie de respuesta teórica para un proceso químico; (b) gráfica de contorno de la superficie de respuesta teórica.

de reacción (ξ_1) y la temperatura de reacción (ξ_2). Para cada par de valores de ξ_1 y ξ_2 hay un valor correspondiente de y , de forma que se pueden ver estos valores de respuesta como una superficie en el plano tiempo-temperatura. Esta perspectiva gráfica del entorno del problema ha llevado al término de superficie de respuesta. Conviene ver también la superficie de respuesta en dos dimensiones conectando todos los puntos que tienen la misma respuesta con líneas de contorno de respuestas constantes. Este tipo de representación es llamada gráfica de contorno [Myers y Montgomery, 2002].

6.1.2. Aproximación de funciones de respuesta

En general se supone que el investigador está interesado en un producto, proceso, o sistema que implica una respuesta y que depende de unas variables de entrada manejables $\xi_1, \xi_2, \dots, \xi_k$. La relación entre éstas es una función

$$y = f(\xi_1, \xi_2, \dots, \xi_k) + \epsilon$$

donde la forma de la función de superficie de respuesta f es desconocida y posiblemente muy complicada, ϵ es un término que representa otras fuentes de variabilidad no tenidas en cuenta en la función f . En general ϵ es tratado como una variable aleatoria con una distribución normal $N(0, \sigma^2)$.

Las variables $\xi_1, \xi_2, \dots, \xi_k$ son llamadas usualmente variables naturales. En muchos problemas de superficies de respuesta conviene transformar las variables de entrada originales a variables tipificadas $x_1, x_2 \dots x_k$, es decir, con media cero y desviación típica 1.

Dado que la forma de la función de respuesta f es desconocida, debemos aproximarla. Normalmente se usan polinomios de primer o segundo orden, aunque en algunos casos se usan polinomios de mayor grado. Para el caso de dos variables independientes (x_1, x_2) una superficie de respuesta de primer orden sería:

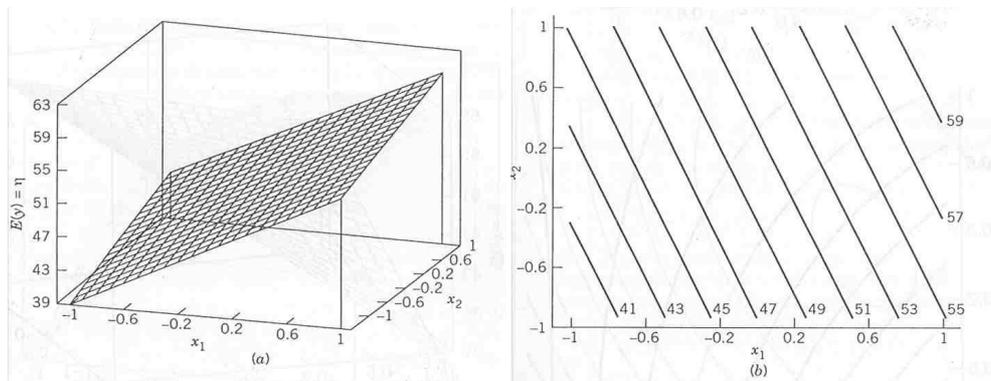


Figura 6.2: (a) superficie de respuesta de primer orden; (b) gráfica de contorno

$$f = c_0 + c_1x_1 + c_2x_2$$

El modelo de primer orden es apropiado cuando el investigador está interesado en aproximar la superficie de respuesta de una región relativamente pequeña del espacio de las variables independientes donde hay poca curvatura en f y poca interacción entre las variables independientes. En la figura 6.2 se puede observar un ejemplo de este tipo.

Cuando hay interacción entre las variables independientes, un modelo de primer orden resulta inadecuado, y puede ser apropiado uno de segundo orden. Para el caso de dos variables independientes éste sería

$$f = c_0 + c_1x_1 + c_2x_2 + c_{11}x_1^2 + c_{22}x_2^2 + c_{12}x_1x_2$$

En la figura 6.3 podemos un ejemplo de una superficie de respuesta de grado 2 y su gráfica de contorno.

En general un modelo de superficie de respuesta de primer orden con n variables independientes sería

$$f(x_1, x_2 \dots x_n) = c_0 + \sum_{i=1}^n c_i x_i$$

uno de segundo orden

$$f(x_1, x_2 \dots x_n) = c_0 + \sum_{i=1}^n c_i x_i + \sum_{i,j=1}^n c_{ij} x_i x_j$$

y en general, una superficie de respuesta de grado G respondería al siguiente modelo

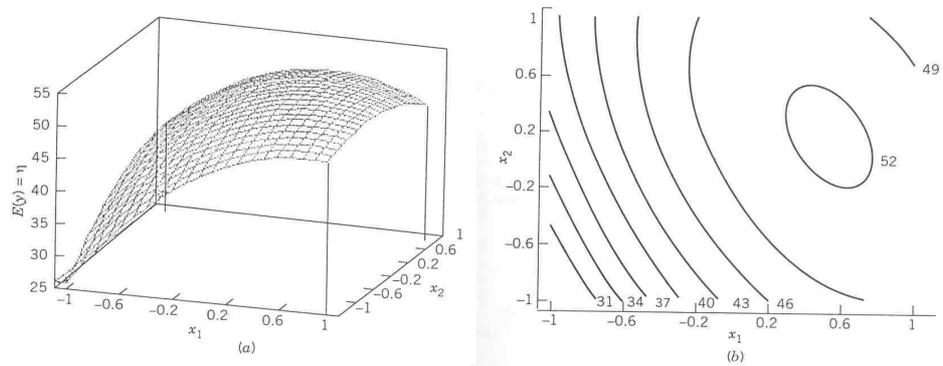


Figura 6.3: (a) superficie de respuesta de segundo orden; (b) gráfica de contorno

$$f(x_1, x_2 \dots x_n) = c_0 + \sum_{i=1}^n c_i x_i + \sum_{i,j=1}^n c_{ij} x_{ij} + \dots + \sum_{\substack{i_1, i_2, \dots, i_G=1 \\ i_k \leq i_{k+1}}}^n c_{i_1 i_2 \dots i_G} x_{i_1} x_{i_2} \dots x_{i_G}$$

El número de coeficientes de una superficie de respuesta de grado G con n variables es uno más la suma de los coeficientes de cada grado. Los coeficientes de cada grado i ($1 \leq i \leq G$) dependen de un número combinatorio $\binom{i+n-1}{i}$. Por tanto, el número total de coeficientes se puede calcular de la siguiente forma:

$$1 + \sum_{i=1}^G \binom{i+n-1}{i} = 1 + \sum_{i=1}^G \left(\frac{(i+n-1)!}{i!(n-1)!} \right)$$

Así, para una superficie de respuesta de grado 3 con 3 variables, el número de coeficientes es: $1 + \binom{3}{1} + \binom{4}{2} + \binom{5}{3} = 1 + 3 + 6 + 10 = 20$. Las combinaciones de variables de predicción a las que se aplicarán estos coeficientes serán las siguientes (hacen un total de 19 combinaciones, al que habría que sumar el coeficiente de intercepción o constante):

$$\begin{aligned} & x_0, x_1, x_2, && \text{(grado 1)} \\ & x_0^2, x_0 x_1, x_0 x_2, x_1^2, x_1 x_2, x_2^2, && \text{(grado 2)} \\ & x_0^3, x_0^2 x_1, x_0^2 x_2, x_0 x_1^2, x_0 x_1 x_2, x_0 x_2^2, x_1^3, x_1^2 x_2, x_1 x_2^2, x_2^3 && \text{(grado 3)} \end{aligned}$$

El establecimiento de un modelo de superficie de respuesta para un fenómeno puede abordarse mediante varias metodologías , pero todas tienen en común dos aspectos:

- La obtención de la estructura de la superficie de respuesta. Es decir, la detección de qué variables de predicción y combinaciones de las mismas tienen influencia sobre la variable de respuesta.
- Una vez definida la estructura de la superficie de respuesta, la obtención de valores para los coeficientes que la forman.

Algunos investigadores [Carpenter y Barthelemy, 1993] [Barton, 1999], sugieren que las redes neuronales artificiales pueden ser una alternativa a la metodología de superficie de respuesta. Una red neuronal sin realimentación con una capa oculta (ver figura 6.4) relaciona p variables de predicción x_1, x_2, \dots, x_p con q variables de respuesta y_1, y_2, \dots, y_q mediante una serie de transformaciones que se llevan a cabo en las k neuronas de la capa oculta (a_1, \dots, a_k) y en las de la capa de salida. La primera fase es el paso de los valores de entrada a las neuronas de la capa oculta, que se activarán de la siguiente forma:

$$salida(a_u) = g\left(\sum_{j=1}^p w_{1ju} x_j + \theta_u\right)$$

donde w_{1ju} son los pesos que relacionan la entrada x_j con la neurona a_u , θ_u es un parámetro equivalente a la constante (el punto de intercepción) en una superficie de respuesta y $g(\cdot)$ es la función de activación de las neuronas de la capa oculta (típicamente una función tangente hiperbólica). Una vez calculadas las salidas de las neuronas a_u se calculan las salidas de la red:

$$salida(y_v) = g_2\left[\sum_{u=1}^k w_{2kv} g\left(\sum_{j=1}^p w_{1ju} x_j + \theta_{1u}\right) + \theta_{2u}\right] \quad (6.1)$$

donde $g_2(\cdot)$ es la función de activación de las neuronas de la capa de salida (normalmente una función sigmoide o una función identidad, dependiendo del problema), θ_{2u} es el parámetro de intercepción y w_{2kv} es el peso que relaciona la neurona a_k de la capa oculta con la neurona y_v de la capa de salida.

La ecuación 6.1 describe de forma general la transformación aplicada por una red neuronal artificial. Dada su forma flexible y su número de parámetros, se la puede considerar como un aproximador universal. Sin embargo, la especificación de la estructura y los parámetros de una red neuronal hacen difícil esta tarea.

Por otra parte, normalmente las redes neuronales artificiales se usan para resolver problemas de predicción de un sistema, no para la construcción de modelos formales del funcionamiento del mismo. De hecho, comúnmente las redes neuronales se consideran “cajas negras”, en el sentido de que la expresión que relaciona las entradas con la salida (o las salidas) no suele ser fácil de analizar. Una superficie de respuesta sí puede presentar un modelo interpretable del sistema al que describa, algo que suele ser interesante para el investigador.

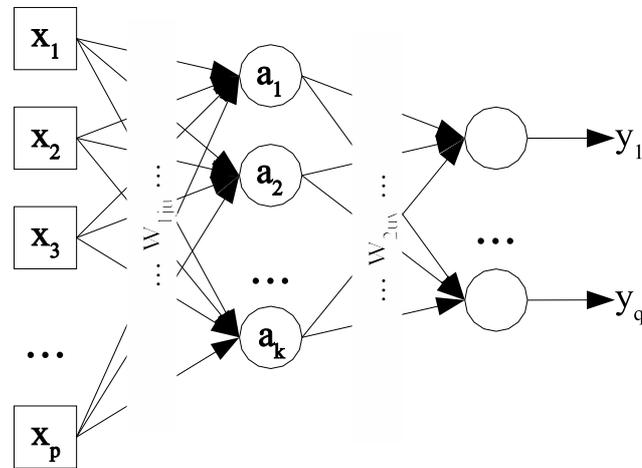


Figura 6.4: Esquema general de una red neuronal artificial.

6.2. Generación coevolutiva de superficies de respuesta

Se aplicará el modelo general descrito en el capítulo 3 a la coevolución cooperativa de superficies de respuesta. Se evolucionarán tanto la estructura como los coeficientes de las mismas. En la figura 6.5 se puede ver un ejemplo del tipo de genotipo que se empleará y del método de obtención del fenotipo a partir del mismo. En el ejemplo se considerarán superficies de respuesta de grado 2 con dos variables de predicción (x_1 y x_2), con lo que el número de términos (incluida la constante) será de 6.

Se evolucionará para cada superficie de respuesta un genotipo compuesto por dos vectores llamados **Presente** y **Valor**. El tamaño de ambos vectores será igual al número de coeficientes de la superficie de respuesta. El vector **Presente** contendrá valores binarios, cada uno de los cuales indicará si el coeficiente correspondiente está incluido (valor 1) o no (valor 0) en el fenotipo. Por su parte, el vector **Valor** contendrá valores reales que representarán los coeficientes de la superficie de respuesta.

Para obtener un fenotipo evaluable a partir de un genotipo como el anteriormente descrito se construye una expresión polinómica en la que los coeficientes serán los descritos por los vectores **Presente** y **Valor** y los términos serán generados en función del grado y número de variables de la superficie de respuesta. Sólo aparecerán en el fenotipo aquellos términos (con sus coeficientes) cuyos correspondientes elementos del vector **Presente** estén activos (con valor 1).

La utilización de este tipo de genotipo (dividido en dos vectores) permite independizar los dos problemas anteriormente mencionados en la generación de superficies de respuesta: la obtención de la estructura y la obtención de los coeficientes. Los operadores de cruce y mutación, que se explicarán más abajo, tendrán en cuenta este genotipo para aplicar por separado las modificaciones.

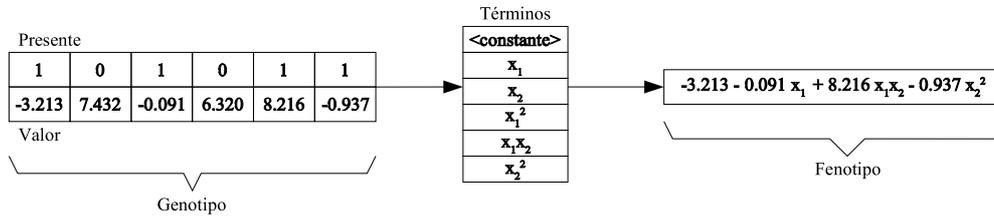


Figura 6.5: Genotipo para la evolución de superficies de respuesta.

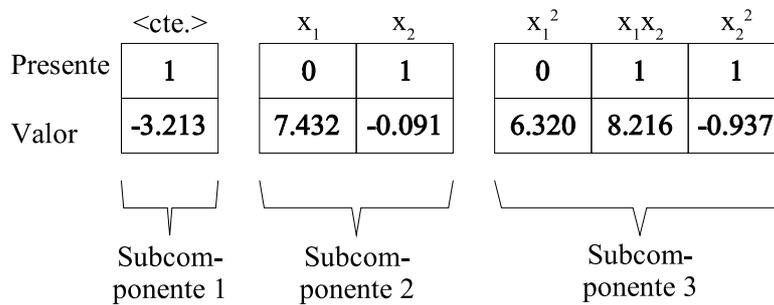


Figura 6.6: División en subcomponentes de una superficie de respuesta

Al evolucionar este tipo de codificación para una superficie de respuesta mediante un modelo coevolutivo como el propuesto en este trabajo se plantea una cuestión: la división del genotipo en subcomponentes. Existen varias posibilidades (división por variables, por grados, por interacciones entre variables...), pero se ha escogido la división por grados.

En la división que se ha planteado, el número K de subcomponentes es igual al grado de la superficie de respuesta más uno. En cada subcomponente p ($1 \leq p \leq K$) se evolucionan los $\binom{p+n-1}{p}$ coeficientes de ese grado (siendo n el número de variables de predicción del modelo). En la figura 6.6 se muestra un ejemplo basado en el anteriormente mostrado en la figura 6.5 (superficie de respuesta de grado 2 con 2 variables). Como se ve, el primer subcomponente evoluciona el término de intercepción, el segundo evoluciona los términos de grado 1 (x_1, x_2) y el tercer subcomponente se encarga de la evolución de los términos de grado 2 (x_1^2, x_1x_2, x_2^2). Esta división crea, al igual que las demás posibles, una epístasis interpoblacional, aunque los resultados experimentales muestran que el método elegido obtiene buenas prestaciones.

La adaptación al modelo propuesto, como se explicó en la sección 4.5, pasa por la implementación de las estructuras de datos y las funciones propias del problema. A continuación veremos cómo se ha realizado esta adaptación.

Estructura de datos de subcomponentes. Los subcomponentes tienen un genotipo formado por dos vectores. El tamaño de ambos vectores depende de la población p a la que pertenezca el subcomponente y se calcula como $\binom{p+n-1}{p}$ siendo n el número de variables del modelo. El primer vector, **Presente**, contiene valores numéricos enteros que pueden ser cero, si el coeficiente correspondiente a ese valor no participa en la superficie de respuesta, y uno si el coeficiente sí participa. El vector **Valor** contiene valores numéricos reales que expresan el valor de cada coeficiente de la superficie de respuesta descrito por el subcomponente.

Creación de subcomponentes aleatorios. Durante la iniciación del proceso evolutivo los subcomponentes se crean asignando a cada elemento del vector **Presente** un valor cero con probabilidad 0.5 o un valor uno con probabilidad 0.5. A cada elemento del vector **Valor** se le asigna un valor real aleatorio según una distribución uniforme en el intervalo $[-1, 1]$.

Duplicación de subcomponentes. La duplicación de subcomponentes únicamente crea un nuevo subcomponente y copia los vectores **Presente** y **Valor** al mismo.

Cruce de subcomponentes. Sobre los subcomponentes se aplica un cruce BLX- α de la siguiente forma:

- Se crean dos hijos a partir de los padres.
- Cada elemento del vector **Valor** de los hijos se construye mediante un cruce BLX- α entre los correspondientes elementos de los vectores **Valor** de los padres.
- Cada elemento del vector **Presente** de los hijos se construye mediante un cruce BLX- α entre los correspondientes elementos de los vectores **Presente** de los padres.

Mutación de subcomponentes. La mutación sobre un subcomponente puede ser de dos tipos: estructural o numérica. Previamente a la mutación se decide si el subcomponente la sufrirá o no. Este paso adicional permite modular la mutación de modo que los subcomponentes con menos coeficientes (los de las primeras poblaciones) sufran menos mutación que los subcomponentes con más coeficientes. La probabilidad de que un subcomponente seleccionado para mutación por el proceso evolutivo realmente sufra esa mutación es

$$prob = 0,5 + 0,5 \frac{p-1}{K-1}$$

donde p es el índice ($1 \leq p \leq K$) de la población a la que pertenece el subcomponente, y K es el número de subcomponentes usados en la coevolución. Según esta fórmula, por ejemplo, en una superficie de respuesta de grado 3, los subcomponentes que representan el coeficiente de intercepción o constante (los pertenecientes a la población PS_1) tendrían una probabilidad de ser mutados de $0,5 + 0,5 \cdot 0/2 = 0,5$, los de la población PS_2 tendrían una probabilidad de $0,5 + 0,5 \cdot 1/2 = 0,75$ y los

subcomponentes de PS_3 serían mutados (una vez seleccionados para ello) con probabilidad $0,5 + 0,5 \cdot 2/2 = 1$.

Una vez decidida la mutación del subcomponente, con probabilidad 0,5 se le aplicará una mutación estructural o una mutación numérica. La mutación estructural consiste en la selección de uno de los coeficientes representados por el subcomponente y su modificación: si el coeficiente está activo (valor 1 en el vector **Presente**), será desactivado y su valor será puesto a cero. Si el coeficiente está inactivo, será activado y su valor se fijará de forma aleatoria uniforme en el intervalo $[-1, 1]$.

La mutación numérica se basa en la perturbación de todos sus coeficientes (todos los valores del vector **Valor**) en una cantidad aleatoria perteneciente a una distribución normal de media cero y varianza $0,1 \cdot v_{actual}$, siendo v_{actual} el valor actual del coeficiente.

Evaluación de individuos. La evaluación de la aptitud de un individuo (es decir, de una superficie de respuesta) se realiza calculando su SEP de generalización. El SEP (*standard error of prediction*) es un error relativo estándar, expresado como porcentaje, que presenta la ventaja de ser adimensional. Se calcula en función de los valores experimentales que se tienen de la variable de respuesta (y_i) y de los valores predichos (\hat{y}_i) por el modelo de superficie de respuesta que se desea evaluar, y que se define como

$$SEP = \frac{100}{\bar{y}} \sqrt{\frac{\sum_{i=1}^m (y_i - \hat{y}_i)^2}{m}}$$

donde m es el número de observaciones que se tienen.

La evolución de las superficies de respuesta se hace para ajustarlas a un conjunto de observaciones que consta de una serie de tuplas que describen los valores de las variables de predicción y la variable de respuesta. De esas tuplas se toma el 75 % como conjunto de entrenamiento y el restante 25 % como conjunto de generalización. El primero será usado durante el proceso evolutivo. Cuando dicho proceso finalice se evaluará el mejor individuo obtenido frente al conjunto de generalización. Si no se especifica lo contrario, todos los resultados que se discutan en las siguientes secciones se referirán al SEP de generalización.

Si se obtienen individuos incompletos, por ejemplo como resultado de una aplicación del criterio de asignación de crédito Diferencia, la evaluación se realizará sin tener en cuenta los coeficientes que serían aportados por el subcomponente que falte.

6.3. Modelado de crecimiento microbiano: *Leuconostoc*

6.3.1. Descripción

Las bacterias ácido lácticas (BAL) son consideradas como los principales microorganismos responsables del deterioro de productos cárnicos cocidos envasados, debido a la

formación de ácido láctico, limo y CO_2 , que provocan la aparición de olores y sabores extraños; afectando al grado de aceptación del producto. Dentro de este grupo de BAL, *Leuconostoc mesenteroides subsp. mesenteroides* ha sido frecuentemente aislado como microorganismo responsable de alteración en diversos tipos de productos cárnicos.

Para evitar las grandes pérdidas económicas que se producen en las industrias cárnicas debido al deterioro de los productos, es muy importante la aplicación de la Microbiología Predictiva en este ámbito y de este modo desarrollar modelos matemáticos que estimen el crecimiento y evolución de los microorganismos alterantes, así como la vida media comercial de este tipo de productos cárnicos.

En esta sección del presente trabajo se ha elaborado un modelo predictivo de crecimiento del microorganismo alterante *Leuconostoc mesenteroides subsp. mesenteroides*, englobando los principales factores que afectan la estabilidad microbiana en los productos cárnicos: la temperatura, el pH, la concentración de ClNa, concentración de $NaNO_2$ y la atmósfera gaseosa (aerobiosis/anaerobiosis).

Se han realizado algunos trabajos usando redes neuronales para solucionar el mismo problema o problemas similares [García-Gimeno et al., 2003a] [Sanz-Tapia et al., 2003] [García-Gimeno et al., 2003b].

6.3.2. Diseño experimental

Para la obtención de datos de absorbancia (a partir de los cuales se obtendrán los parámetros de las curvas de crecimiento) se elaboró un Diseño Central Compuesto (DCC), el cual comprende las condiciones experimentales que aparecen en la tabla 6.1. Se seleccionaron como variables más importantes la temperatura ($^{\circ}C$), pH, concentración de cloruro sódico (%), concentración de nitrito sódico (ppm) y atmósfera aeróbica/anaeróbica, para estudiar el efecto combinado de las mismas en el crecimiento de *L. mesenteroides* en medio de cultivo Caldo Triptona Soja (TSB).

Los rangos de las variables estudiadas fueron escogidos teniendo en cuenta los procesos seguidos por la industria cárnica y comprendieron los siguientes valores:

- Temperatura ($^{\circ}C$): 10.5, 14, 17.5, 21, 24.5
- pH: 5.5, 6, 6.5, 7, 7.5
- Concentración de NaCl (%): 0.25, 1.75, 3.25, 4.75, 6.25
- Concentración de $NaNO_2$ (ppm): 0, 50, 100, 150, 200
- Condiciones de la atmósfera: aerobiosis y anaerobiosis

Las combinaciones de valores que se usaron, dictadas por el método de diseño central compuesto, se muestran en la tabla 6.1. Para cada combinación se realizaron siete réplicas en anaerobiosis y otras siete en aerobiosis.

Temp. (°C)	pH	NaCl (%)	NaNO2 (ppm)	Temp. (°C)	pH	NaCl (%)	NaNO2 (ppm)
10.5	6.5	3.25	100	17.5*	6.5	3.25	100
14	6	1.75	50	17.5*	6.5	3.25	100
14	6	1.75	150	17.5*	6.5	3.25	100
14	6	4.75	50	17.5	6.5	3.25	200
14	6	4.75	150	17.5	6.5	6.25	100
14	7	1.75	50	17.5	7.5	3.25	100
14	7	1.75	150	21	6	1.75	50
14	7	4.75	50	21	6	1.75	150
14	7	4.75	150	21	6	4.75	50
17.5	5.5	3.25	100	21	6	4.75	150
17.5	6.5	0.25	100	21	7	1.75	50
17.5	6.5	3.25	0	21	7	1.75	150
17.5*	6.5	3.25	100	21	7	4.75	50
17.5*	6.5	3.25	100	21	7	4.75	150
17.5*	6.5	3.25	100	24.5	6.5	3.25	100

Cuadro 6.1: Combinaciones de valores para el diseño del experimento *Leuconostoc*. Marcadas con * las condiciones centrales del DCC

Una vez diseñado el experimento se prepararon los cultivos y mediante un espectrofotómetro se realizaron lecturas de absorbancia, ya que el crecimiento del microorganismo se manifiesta con un incremento de la turbidez del medio.

Se obtienen, por tanto, 7 curvas de crecimiento (relaciones señal-tiempo) para cada una de las 30 combinaciones de parámetros, para cada modelo de crecimiento (aeróbico y anaeróbico). Un total de 420 curvas de crecimiento (210 para crecimiento anaerobio y 210 para crecimiento aerobio).

Los valores de densidad óptica obtenidos se transformaron a logaritmo neperiano para homogeneizar la variabilidad. A continuación estos valores de absorbancia resultantes, enfrentados al tiempo, fueron ajustados al modelo primario de Baranyi y Roberts [Baranyi y Roberts, 1994] con la ayuda del programa DMFit [Baranyi,], que permite modelar la variación del logaritmo de la concentración celular de cultivos bacterianos a lo largo del tiempo. Como resultado se obtienen los principales parámetros cinéticos de crecimiento (tasa o velocidad de crecimiento, punto de despegue, densidad máxima, etc.) del microorganismo para las condiciones experimentales estudiadas. En la figura 6.7 se muestran gráficamente estos parámetros. La tasa de crecimiento o *growth rate* se define como el valor máximo de la pendiente de la curva de crecimiento (en la figura se alcanza en el punto (t_m, y_m)). El punto de despegue o *lag* es el punto de intersección de la asíntota inferior de la curva (y_{min}) con la recta tangente a la misma en el punto de máxima pendiente. La densidad máxima (y_{end}) es el valor máximo asíntótico alcanzado por la curva de crecimiento.

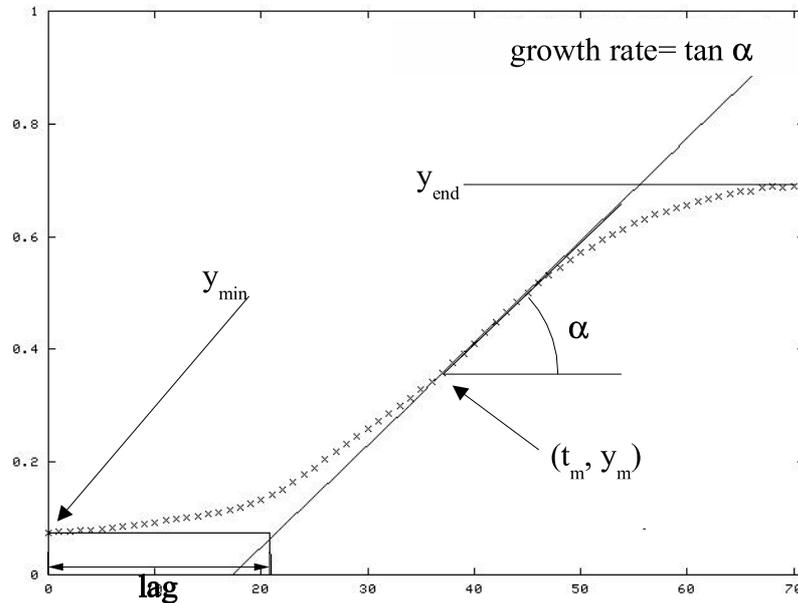


Figura 6.7: Cálculo de la tasa de crecimiento, punto de despegue y densidad máxima.

El resultado fueron 210 tuplas por cada tipo de crecimiento. Cada tupla consta de cuatro variables de predicción (temperatura, pH, NaCl y $NaNO_2$) y tres variables de respuesta (tasa de crecimiento, punto de despegue y densidad máxima). De las 210 tuplas, 150 fueron usadas como conjunto de entrenamiento durante la evolución de las poblaciones y las restantes 60 constituyeron el conjunto de generalización.

Se plantearon tres subproblemas: el problema “grate” trata de modelar la tasa de crecimiento en función de la temperatura, el nivel de pH, la concentración de cloruro sódico y la concentración de $NaNO_2$; el problema “lnlag” modelará el punto de despegue (pre-procesado mediante un logaritmo neperiano, puesto que se observó experimentalmente que los resultados mejoraban) en función de las mismas variables de predicción; y el problema “yend” intentará obtener una superficie de respuesta que describa la densidad máxima como respuesta a dichas variables.

Los parámetros del proceso evolutivo se muestran en la tabla 6.2. Son similares a los usados en el problema de cobertura de vectores (capítulo 5), aunque se han realizado cambios para adaptarlo al modelo, que tiene mayor complejidad.

Las pruebas realizadas tratan de discernir qué combinación o combinaciones de criterios de asignación de crédito proporcionan mejores resultados. El diseño de experimentos es idéntico al realizado en el problema de cobertura de vectores (sección 5.2): cinco factores (los criterios de asignación de crédito propuestos por el modelo) con dos niveles (presente y no-presente) cada uno. Los niveles indican si un factor se tiene en cuenta o no en la asignación de crédito a subcomponentes durante la evolución. La tabla 5.1, en la página 70, muestra todas las combinaciones de niveles para los factores.

Parámetro		Valor
Número de generaciones		5000
Poblaciones de subcomponentes	Tamaño (M)	150
	Reproducción	50 %, selección por torneo
	Mutación	25 %, selección aleatoria
	Cruce	BLX- α , 25 %, selección aleatoria
	Elitismo	subcomponentes de los 50 mejores individuos
Población de individuos	Tamaño (N_p)	500
	Evolución	estado estacionario
	Mutación	10 %
	Cruce	dos puntos. Dos individuos (selección por torneo) se cruzan y substituyen a los dos peores de la población.

Cuadro 6.2: Parámetros del proceso evolutivo. Crecimiento microbiano. *Leuconostoc*.

Se ejecutaron 30 réplicas de cada combinación y se obtuvo el mejor individuo de la generación final de cada una de ellas. La combinación 32, considerada el grupo de control, tiene todos los factores con valor “no-presente” y representa una búsqueda aleatoria.

6.3.3. Resultados

En la tabla 6.3 se muestran los resultados promedio de las 30 ejecuciones para cada uno de los 32 experimentos sobre el problema “grate”. La tabla 6.4 muestra los resultados para el problema “Inlag” y en la tabla 6.5 pueden verse las aptitudes finales medias para el problema “yend”.

Para poder hacer contrastes de diferencia de medias a partir de las aptitudes obtenidas se ha realizado un análisis de varianza (ANOVA [Kendall y Stuart, 1977]) de cinco factores, considerando interacciones.

Una vez terminado el análisis y en función del test de Levene de igualdad de varianzas se hicieron contrastes de comparación entre las medias con y sin cada criterio. Se usaron contrastes t de Student, suponiendo las varianzas poblacionales iguales.

Las tablas 6.6, 6.7 y 6.8 muestran los resultados medios de todos los experimentos con y sin cada criterio. Y en las tablas 6.9, 6.10 y 6.11 se muestran las pruebas de efectos inter-sujetos, con los niveles de significación del test T de diferencia de medias considerando o sin considerar los criterios y sus combinaciones dos a dos, tres a tres, cuatro a cuatro y totales.

Combinación	Mejor	Media	Sust.	Dif.	Unión	Aptitud	Desv. Est.
1	✓	—	—	—	—	10.0235861	.93750627
2	—	✓	—	—	—	9.7303497	.94518820
3	—	—	✓	—	—	10.3543580	.48970112
4	—	—	—	✓	—	9.8860130	.73705045
5	—	—	—	—	✓	10.0656645	.78688834
6	✓	✓	—	—	—	9.8837692	.79238848
7	✓	—	✓	—	—	10.0935295	.80350096
8	✓	—	—	✓	—	9.9116626	.81154689
9	✓	—	—	—	✓	10.2072342	.94780708
10	—	✓	✓	—	—	9.9958808	.93501854
11	—	✓	—	✓	—	10.1191002	.69907463
12	—	✓	—	—	✓	10.2825936	.84125217
13	—	—	✓	✓	—	9.7403148	.82063520
14	—	—	✓	—	✓	10.4529281	.55740489
15	—	—	—	✓	✓	10.3256759	.69290413
16	✓	✓	✓	—	—	9.8956342	.65468121
17	✓	✓	—	✓	—	10.0320456	.77725785
18	✓	✓	—	—	✓	9.8199029	.87148737
19	✓	—	✓	✓	—	9.5372406	.80004381
20	✓	—	✓	—	✓	10.0530673	.90273068
21	✓	—	—	✓	✓	9.5970226	.59215054
22	—	✓	✓	✓	—	10.3659880	.59664560
23	—	✓	✓	—	✓	10.2785920	.89589936
24	—	✓	—	✓	✓	10.0789734	.72804595
25	—	—	✓	✓	✓	9.9321209	.77781972
26	✓	✓	✓	✓	—	9.6700686	.61904096
27	✓	✓	✓	—	✓	10.2007393	.82694984
28	✓	✓	—	✓	✓	10.0380984	.79059693
29	✓	—	✓	✓	✓	9.8294624	.62765095
30	—	✓	✓	✓	✓	10.0755209	.76352835
31	✓	✓	✓	✓	✓	9.7358017	.73253248
32	—	—	—	—	—	11.0853997	1.01619938

Cuadro 6.3: Resultados para problema “grate”.

Combinación	Mejor	Media	Sust.	Dif.	Unión	Aptitud	Desv. Est.
1	✓	—	—	—	—	6.5249001	.18704871
2	—	✓	—	—	—	6.5758347	.11285683
3	—	—	✓	—	—	6.4491334	.15575290
4	—	—	—	✓	—	6.5435211	.15300944
5	—	—	—	—	✓	6.5107038	.18285498
6	✓	✓	—	—	—	6.5912875	.15362569
7	✓	—	✓	—	—	6.4305808	.14545233
8	✓	—	—	✓	—	6.5309936	.23095995
9	✓	—	—	—	✓	6.5406292	.20744616
10	—	✓	✓	—	—	6.5425860	.17063047
11	—	✓	—	✓	—	6.6038136	.19339465
12	—	✓	—	—	✓	6.5897526	.19891968
13	—	—	✓	✓	—	6.5764422	.26240337
14	—	—	✓	—	✓	6.5374660	.16411181
15	—	—	—	✓	✓	6.6388599	.19901155
16	✓	✓	✓	—	—	6.5170950	.12870592
17	✓	✓	—	✓	—	6.5874727	.22086271
18	✓	✓	—	—	✓	6.4969940	.15421469
19	✓	—	✓	✓	—	6.4663830	.16311293
20	✓	—	✓	—	✓	6.4660180	.18705277
21	✓	—	—	✓	✓	6.5168870	.17666975
22	—	✓	✓	✓	—	6.5565985	.17978737
23	—	✓	✓	—	✓	6.5646604	.17861254
24	—	✓	—	✓	✓	6.5407388	.13547227
25	—	—	✓	✓	✓	6.5296238	.13976294
26	✓	✓	✓	✓	—	6.5182525	.19525798
27	✓	✓	✓	—	✓	6.4943424	.18034858
28	✓	✓	—	✓	✓	6.5174883	.14995152
29	✓	—	✓	✓	✓	6.5226438	.13870043
30	—	✓	✓	✓	✓	6.5590296	.21045542
31	✓	✓	✓	✓	✓	6.5380787	.17066994
32	—	—	—	—	—	6.7569128	.10948577

Cuadro 6.4: Resultados para problema “lnlag”.

Combinación	Mejor	Media	Sust.	Dif.	Unión	Aptitud	Desv. Est.
1	✓	—	—	—	—	13.9304366	.44870849
2	—	✓	—	—	—	14.2122691	.32778059
3	—	—	✓	—	—	14.1016562	.43815323
4	—	—	—	✓	—	14.1244620	.38038559
5	—	—	—	—	✓	14.1707004	.38623678
6	✓	✓	—	—	—	13.9479791	.33481436
7	✓	—	✓	—	—	14.1994933	.39513006
8	✓	—	—	✓	—	14.1380659	.33228306
9	✓	—	—	—	✓	14.3403029	.38867583
10	—	✓	✓	—	—	14.3351355	.39573926
11	—	✓	—	✓	—	14.2028840	.30287833
12	—	✓	—	—	✓	14.2371510	.38871203
13	—	—	✓	✓	—	14.1466813	.41914757
14	—	—	✓	—	✓	14.2042761	.40414534
15	—	—	—	✓	✓	14.2952679	.34893533
16	✓	✓	✓	—	—	13.9712420	.35243243
17	✓	✓	—	✓	—	14.2366726	.33898258
18	✓	✓	—	—	✓	14.0199407	.50725982
19	✓	—	✓	✓	—	14.0362938	.43829029
20	✓	—	✓	—	✓	14.1663625	.36916032
21	✓	—	—	✓	✓	14.1450327	.44879340
22	—	✓	✓	✓	—	14.0954434	.38538504
23	—	✓	✓	—	✓	14.4682560	.42307123
24	—	✓	—	✓	✓	14.5199525	.48754649
25	—	—	✓	✓	✓	14.2084424	.42682450
26	✓	✓	✓	✓	—	13.9146963	.34146261
27	✓	✓	✓	—	✓	14.0398862	.34138666
28	✓	✓	—	✓	✓	14.2754789	.42318034
29	✓	—	✓	✓	✓	14.2817771	.43709034
30	—	✓	✓	✓	✓	14.2819681	.40825179
31	✓	✓	✓	✓	✓	14.0818851	.40439613
32	—	—	—	—	—	14.5042481	.46251946

Cuadro 6.5: Resultados para problema “yend”.

Criterio	Con criterio		Sin criterio	
	Aptitud	Desv. Est.	Aptitud	Desv. Est.
Mejor	9.9080541	.79358200	10.1730921	.82545671
Media	10.0126911	.79403136	10.0684550	.84518540
Sust.	10.0132029	.77681676	10.0679432	.86106714
Dif.	9.9296943	.74681495	10.1514518	.87406293
Unión	10.0608374	.79290323	10.0203088	.84667849

Cuadro 6.6: Resultados medios considerando/sin considerar cada criterio. *Leuconostoc*. Problema “grate”.

Criterio	Con criterio		Sin criterio	
	Aptitud	Desv. Est.	Aptitud	Desv. Est.
Mejor	6.5162529	.17703480	6.5672298	.18291322
Media	6.5496266	.17249406	6.5338561	.19032345
Sust.	6.5168084	.17694152	6.5666744	.18315691
Dif.	6.5466767	.18567016	6.5368060	.17770674
Unión	6.5352448	.17529207	6.5482380	.18785626

Cuadro 6.7: Resultados medios considerando/sin considerar cada criterio. *Leuconostoc*. Problema “lnlag”.

Criterio	Con criterio		Sin criterio	
	Aptitud	Desv. Est.	Aptitud	Desv. Est.
Mejor	14.1078466	.40930547	14.2567996	.41397164
Media	14.1775525	.41762994	14.1870937	.41901233
Sust.	14.1583434	.41394451	14.2063028	.42134077
Dif.	14.1865627	.41057405	14.1780835	.42593972
Unión	14.2335425	.42585084	14.1311037	.40425180

Cuadro 6.8: Resultados medios considerando/sin considerar cada criterio. *Leuconostoc*. Problema “yend”.

Combinación	Significación test T
M	0.000
P	0.369
S	0.377
D	0.000
U	0.513
M * P	0.344
M * S	0.904
P * S	0.176
M * P * S	0.149
M * D	0.917
P * D	0.000
M * P * D	0.209
S * D	0.181
M * S * D	0.364
P * S * D	0.408
M * P * S * D	0.236
M * U	0.825
P * U	0.320
M * P * U	0.544
S * U	0.242
M * S * U	0.642
P * S * U	0.176
M * P * S * U	0.148
D * U	0.958
M * D * U	0.467
P * D * U	0.006
M * P * D * U	0.006
S * D * U	0.406
M * S * D * U	0.060
P * S * D * U	0.805
M * P * S * D * U	0.011

Cuadro 6.9: Pruebas de efectos inter-sujetos. Problema “grate”. Criterios: Mejor (M), Media (P, promedio), Diferencia (D), Sustitución (S), Unión (U).

Combinación	Significación test T
M	0.000
P	0.258
S	0.000
D	0.479
U	0.351
M * P	0.223
M * S	0.682
P * S	0.095
M * P * S	0.452
M * D	0.607
P * D	0.788
M * P * D	0.880
S * D	0.095
M * S * D	0.667
P * S * D	0.249
M * P * S * D	0.675
M * U	0.787
P * U	0.430
M * P * U	0.122
S * U	0.021
M * S * U	0.948
P * S * U	0.831
M * P * S * U	0.394
D * U	0.452
M * D * U	0.816
P * D * U	0.309
M * P * D * U	0.090
S * D * U	0.116
M * S * D * U	0.029
P * S * D * U	0.025
M * P * S * D * U	0.011

Cuadro 6.10: Pruebas de efectos inter-sujetos. Problema “lnlag”. Criterios: Mejor (M), Media (P, promedio), Diferencia (D), Sustitución (S), Unión (U).

Combinación	Significación test T
M	0.000
P	0.763
S	0.129
D	0.788
U	0.001
M * P	0.008
M * S	0.870
P * S	0.751
M * P * S	0.039
M * D	0.092
P * D	0.221
M * P * D	0.312
S * D	0.045
M * S * D	0.658
P * S * D	0.003
M * P * S * D	0.318
M * U	0.536
P * U	0.455
M * P * U	0.063
S * U	0.656
M * S * U	0.447
P * S * U	0.969
M * P * S * U	0.180
D * U	0.138
M * D * U	0.086
P * D * U	0.887
M * P * D * U	0.543
S * D * U	0.954
M * S * D * U	0.002
P * S * D * U	0.630
M * P * S * D * U	0.091

Cuadro 6.11: Pruebas de efectos inter-sujetos. Problema “yend”. Criterios: Mejor (M), Media (P, promedio), Diferencia (D), Sustitución (S), Unión (U).

Conclusiones

7.1. Resultados de la aplicación del modelo

El modelo propuesto ha sido aplicado a varios problemas, uno sintético (cobertura de vectores) y otros reales (evoluciones de superficies de respuesta).

Para los problemas de cobertura de vectores (“medio”, “cuarto” y “octavo”) se observa una mejora en los resultados medios en función de la consideración de los criterios de asignación de crédito.

Los resultados de los test ANOVA para el problema “medio” indican la existencia de efectos significativos sobre los resultados en conexión con el criterio de asignación Sustitución y en función de la interacción de los criterios Media, Sustitución, Diferencia y Unión. Los niveles críticos en cada contraste han sido $p < 0,05$.

Para el problema “cuarto” también aparecen diferencias significativas en los contrastes para los criterios Mejor, Media y Sustitución, así como para las combinaciones Media * Sustitución, Sustitución * Unión, Mejor * Media * Diferencia, Mejor * Media * Unión, Mejor * Sustitución * Unión, Media * Sustitución * Unión, Media * Diferencia * Unión, Mejor * Media * Sustitución * Unión y Mejor * Sustitución * Diferencia * Unión. Todos los contrastes con niveles críticos $p < 0,05$.

El problema “octavo” presenta una significancia $p > 0,05$ únicamente en la combinación de criterios Mejor * Media * Sustitución. Todos los demás criterios y combinaciones de los mismos presentan significación $p < 0,05$, pudiéndose rechazar por lo tanto la hipótesis nula $H_0 =$ ‘medias iguales con/sin criterio o combinación’.

Como puede observarse, el número de criterios o combinaciones de los mismos que presentan significación sobre la media aumenta conforme se acrecienta la dificultad del problema. De las 31 posibles combinaciones de criterios (excluyendo la combinación vacía), sólo dos son significativas para el problema “medio”, sólo 12 lo son para el problema “cuarto” y para el problema “octavo” se destacan como significativas 30 combinaciones.

Respecto al problema de aplicación de superficies de respuesta a la predicción de crecimiento microbiano de organismos *Leuconostoc*, los tres subproblemas “grate”, “lnlag” e “yend” presentan resultados que se analizan a continuación.

El subproblema “grate” muestra significación en la aplicación de los criterios Mejor, Diferencia, Media * Diferencia, Media * Diferencia * Unión, Mejor * Media * Diferencia * Unión y Mejor * Media * Sustitución * Diferencia * Unión.

El subproblema “lnlag” presenta valores de significación $p < 0,05$ en las combinaciones Mejor, Sustitución, Sustitución * Unión, Mejor * Sustitución * Diferencia * Unión, Media * Sustitución * Diferencia * Unión y Mejor * Media * Sustitución * Diferencia * Unión.

Para el subproblema “yend” los criterios con significación a nivel 0,05 son: Mejor, Unión, Mejor * Media, Mejor * Media * Sustitución, Sustitución * Diferencia, Media * Sustitución * Diferencia y Mejor * Sustitución * Diferencia * Unión.

No se puede hablar de mayor o menor complejidad respecto a los problemas “grate”, “lnlag” e “yend”. El número de combinaciones con significación en los experimentos sobre los mismos ha sido respectivamente de 6, 6 y 7.

Las pruebas experimentales muestran que el modelo propuesto obtiene resultados significativos en medias al combinar diferentes criterios de asignación de crédito. Cada criterio explica una parte de la media de los resultados.

El método es fácilmente aplicable a cualquier problema, siendo el principal aspecto a tener en cuenta la división del problema para permitir que cada subproblema sea asignado a una población de subcomponentes para su evolución.

7.2. Contribuciones

La línea que se ha seguido en este trabajo ha ido concretando progresivamente el foco de atención, hasta terminar con la resolución de los problemas y la experimentación sobre los mismos. Primeramente se ha desarrollado un modelo general detallado y flexible. A continuación se ha implementado dicho modelo en un software configurable y adaptable a diferentes problemas (según un proceso detallado y documentado). Posteriormente se ha usado dicho software para resolver problemas específicos, extendiéndolo para adaptarlo a los mismos. En último término se han realizado experimentos con cada una de las implementaciones sobre problemas para comprobar el correcto funcionamiento del modelo y justificar la mejora de aptitud en función del número de objetivos.

El modelo general es original y engloba otros modelos de coevolución existentes. Con el modelo propuesto se introduce una nomenclatura de los componentes de un sistema coevolutivo que puede servir de referencia debido a su independencia del problema.

Se incluye dentro del modelo propuesto un modelo formal de la asignación de crédito (sección 3.4.9) basado en funciones de selección, extracción y agregación que es general y acepta múltiples métodos existentes de asignación de crédito.

La implementación del modelo se ha realizado teniendo en cuenta características deseables en todo sistema informático software: documentación correcta, eficiencia, facilidad de mantenimiento, flexibilidad y robustez.

Se ha configurado la implementación para su aplicación a problemas sintéticos y a problemas reales. Posteriormente se han realizado del orden de seis mil experimentos (treinta réplicas, seis problemas, treinta y dos combinaciones de criterios de asignación). El control de la distribución de los experimentos (en varias máquinas biprocesador) y de la recogida de datos se ha realizado en lenguaje Python. Posteriormente se han realizado análisis estadísticos para estudiar la influencia del número de objetivos sobre la aptitud final obtenida.

7.3. Futuras mejoras

Los desarrollos futuros incluyen la aplicación del método a nuevos problemas, el estudio y la agregación de nuevos criterios que describan otras características deseables de los subcomponentes, un análisis profundo sobre el comportamiento y aplicación de cada criterio de asignación de crédito, un estudio detallado del comportamiento del algoritmo multiobjetivo sobre las poblaciones de subcomponentes y el uso y evaluación de otros algoritmos multiobjetivo distintos.

Asimismo, se tienen resultados (no publicados en este trabajo por falta de proceso posterior) sobre la aplicación del modelo a otros problemas de modelado de superficies de respuesta. En concreto a tres conjuntos de datos: datos sintéticos, datos de crecimiento microbiano de organismos *Aureus* y datos de cinética química. Estos resultados deberían ser procesados y comparados con los obtenidos en este trabajo, para corroborar la hipótesis de que añadir nuevos criterios de asignación de crédito mejora la aptitud final obtenida.

También se debería comparar el modelo con modelos anteriores (por ejemplo, la coevolución en un nivel propuesta por Potter) y con métodos diferentes (la regresión por métodos de Levenberg-Marquardt, por ejemplo, o las superficies de respuesta generalizadas).

Por último, se han realizado experimentos usando un nuevo objetivo que pondera el tamaño del subcomponente (aplicable al problema de generación de superficies de respuesta). Se obtendrían con este objetivo expresiones con mayor error de predicción pero menor tamaño, lo que en ocasiones podría ser deseable, sobre todo cuando se trate de obtener una expresión interpretable.

Bibliografía

- [Angeline y Pollack, 1993] ANGELINE, P. J. Y POLLACK, J. B. (1993). Competitive environments evolve better solutions for complex tasks. En Forrest, S., editor, *Proceedings of the 5th International Conference on Genetic Algorithms, ICGA-93*, páginas 264–270. Morgan Kaufmann.
- [Axelrod, 1987] AXELROD, R. (1987). The evolution of strategies in the iterated prisoner's dilemma. En Davis, L., editor, *Genetic Algorithms and Simulated Annealing*, Research Notes in Artificial Intelligence, páginas 32–41, London. Pitman Publishing.
- [Balthrop et al., 2002] BALTHROP, J., ESPONDA, F., FORREST, S., Y GLICKMAN, M. (2002). Coverage and generalization in an artificial immune system. En Langdon, W. B., Cantú-Paz, E., Mathias, K., Roy, R., Davis, D., Poli, R., Balakrishnan, K., Honavar, V., Rudolph, G., Wegener, J., Bull, L., Potter, M. A., Schultz, A. C., Miller, J. F., Burke, E., y Jonoska, N., editores, *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference*, páginas 3–10, New York. Morgan Kaufmann Publishers.
- [Baranyi,] BARANYI, J. Dmfit v1.0 (software).
<http://www.ifr.bbsrc.ac.uk/Safety/DMFit/default.html>.
- [Baranyi y Roberts, 1994] BARANYI, J. Y ROBERTS, T. A. (1994). A dynamic approach to predicting bacterial growth in food. *Int. J. Food Microbiol.*, 23:277–294.
- [Barton, 1999] BARTON, R. R. (1999). Simulation metamodels. En *Proc. Winter Simulation Conference*, páginas 167–174. ACM Association for Computing Machinery.
- [Bäck, 1996] BÄCK, T. (1996). *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, New York, NY.
- [Camou et al., 2000] CAMOU, M., GOERZEN, J., Y VAN COUWENBERGHE, A. (2000). *Debian/GNU Linux Unleashed*. Sams Publishing.
- [Carpenter y Barthelemy, 1993] CARPENTER, W. Y BARTHELEMY, J. (1993). A comparison of polynomial approximations and artificial neural nets as response surfaces. *Structural Optimization*, 5:166–174.
- [Collins y Jefferson, 1991] COLLINS, R. J. Y JEFFERSON, D. R. (1991). Selection in massively parallel genetic algorithms. En *Proceedings of the Fourth International*

- Conference on Genetic Algorithms*, páginas 249–256, San Mateo, CA. Morgan Kaufmann.
- [de Jong, 1975] DE JONG, K. A. (1975). *An analysis of the behavior of a class of genetic adaptive systems*. Tesis doctoral, University of Michigan, Ann Arbor, MI.
- [Deb et al., 2002] DEB, K., PRATAP, A., AGARWAL, S., Y MEYARIVAN, T. (2002). A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.
- [Ehrlich y Raven, 1964] EHRLICH, P. R. Y RAVEN, P. H. (1964). Butterflies and plants: a study in coevolution. *Evolution*, 18:586–608.
- [Farmer et al., 1986] FARMER, D. J., PACKARD, N. H., Y PERELSON, A. S. (1986). The immune system, adaptation, and machine learning. *Physica*, 22D:187–204.
- [Ficici y Pollack, 1998] FICICI, S. G. Y POLLACK, J. B. (1998). Challenges in coevolutionary learning: Arms-race dynamics, open-endedness, and mediocre stable states. En Adami, C., Belew, R. K., Kitano, H., y Taylor, C., editores, *Proceedings of the 6th International Conference on Artificial Life (ALIFE-98)*, páginas 238–247, Cambridge, MA, USA. MIT Press.
- [Ficici y Pollack, 2001] FICICI, S. G. Y POLLACK, J. B. (2001). Pareto optimality in coevolutionary learning. *Lecture Notes in Computer Science*, 2159:316–??
- [Fogel et al., 1966] FOGEL, L. J., OWENS, A. J., Y WALSH, M. J. (1966). *Artificial Intelligence through Simulated Evolution*. John Wiley & Sons, New York.
- [Fonseca y Fleming, 1993] FONSECA, C. M. Y FLEMING, P. J. (1993). Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. En Forrest, S., editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, páginas 416–423, San Mateo, California. University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers.
- [Forrest et al., 1993] FORREST, S., JAVORNIK, B., SMITH, R. E., Y PERELSON, A. S. (1993). Using genetic algorithms to explore pattern recognition in the immune system. *Evolutionary Computation*, 1(3):191–211.
- [García-Gimeno et al., 2003a] GARCÍA-GIMENO, R. AND HERVÁS-MARTÍNEZ, C., BARCO ALCALÁ, E., ZURERA-COSANO, G., Y SANZ-TAPIA, E. (2003a). An artificial neural network approach to escherichia coli 0157:h7 growth estimation. *Journal of Food Science*, 68(2):639–645.
- [García-Gimeno et al., 2003b] GARCÍA-GIMENO, R., HERVÁS-MARTÍNEZ, C., SANZ-TAPIA, E., Y ZURERA-COSANO, G. (2003b). Estimation of microbial growth parameters by means of artificial neural networks. *Food Science and Technology International*, 8(2):73–80.

- [García-Pedrajas et al., 2002] GARCÍA-PEDRAJAS, N., HERVÁS-MARTÍNEZ, C., Y MUÑOZ-PÉREZ, J. (2002). Multi-objective cooperative coevolution of artificial neural networks. *Neural Networks*, 15(10):1255–1274.
- [García-Pedrajas et al., 2003] GARCÍA-PEDRAJAS, N., HERVÁS-MARTÍNEZ, C., Y MUÑOZ-PÉREZ, J. (2003). Covnet: A cooperative coevolutionary model for evolving artificial neural networks. *IEEE Trans. Neural Networks*, 14(3).
- [García-Pedrajas et al., 2001] GARCÍA-PEDRAJAS, N., SANZ-TAPIA, E., ORTIZ-BOYER, D., Y HERVÁS-MARTÍNEZ, C. (2001). Introducing multi-objective optimization in cooperative coevolution of neural networks. *Lecture Notes in Computer Science*, 2084:645–??
- [Goldberg, 1989] GOLDBERG, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA.
- [Goldberg y Richardson, 1987] GOLDBERG, D. E. Y RICHARDSON, J. (1987). Genetic algorithms with sharing for multimodalfunction optimization. En Grefenstette, J. J., editor, *Proceedings of the 2nd International Conference on Genetic Algorithms and their Applications*, páginas 41–49, Cambridge, MA. Lawrence Erlbaum Associates.
- [Hillis, 1990] HILLIS, W. D. (1990). Co-evolving parasites improve simulated evolution as an optimization procedure. *Physica D*, 42:228–234. Proceedings of the Ninth Annual Int. Conf. of the Center for Nonlinear Studies, Ed. Stephanie Forrest.
- [Holland, 1975] HOLLAND, J. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press.
- [Horn et al., 1994] HORN, J., GOLDBERG, D. E., Y DEB, K. (1994). Implicit niching in a learning classifier system: Nature's way. *Evolutionary Computation*, 2(1):37 – 66.
- [Horn y Nafpliotis, 1993] HORN, J. Y NAFPLIOTIS, N. (1993). Multiobjective optimization using the niched pareto genetic algorithm. Informe técnico ILLIGAL 93005, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL.
- [Juille y Pollack, 1998] JUILLE, H. Y POLLACK, J. (1998). Coevolutionary arms race improves generalization. En Koza, J. R., editor, *Late Breaking Papers at the Genetic Programming 1998 Conference*, University of Wisconsin, Madison, Wisconsin, USA. Stanford University Bookstore.
- [Kauffman y Johnsen, 1992] KAUFFMAN, S. A. Y JOHNSEN, S. (1992). Co-evolution to the edge of chaos: Coupled fitness landscapes, poised states, and co-evolutionary avalanches. En Langton, C. G., Taylor, C., Farmer, J. D., y Rasmussen, S., editores, *Artificial Life II*, páginas 325–369. Addison-Wesley, Redwood City, CA.
- [Kendall y Stuart, 1977] KENDALL, M. Y STUART, A. (1977). *The Advanced Theory of Statistics*, volumen 1. MacMillan Publishing, New York.

- [Kernighan y Ritchie, 1988] KERNIGHAN, B. W. Y RITCHIE, D. M. (1988). *The C Programming Language*. Prentice Hall, 2 edición.
- [Knowles y Corne, 2000] KNOWLES, J. D. Y CORNE, D. W. (2000). Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8(2):149–172.
- [Koza, 1994] KOZA, J. R. (1994). *Genetic Programming II. Automatic Discovery of Reusable Programs*. MIT Press, Cambridge, MA.
- [Mahfoud, 1992] MAHFUD, S. W. (1992). Crowding and preselection revisited. En Manner, R. y Manderick, B., editores, *Parallel Problem Solving From Nature 2*, páginas 27–36. Elsevier Science Publishers, Amsterdam, Holland.
- [Matsubara et al., 1996] MATSUBARA, H., NODA, I., Y HIRAKI, K. (1996). Learning of cooperative actions in multiagent systems: A case study of pass play in soccer. En Sen, S., editor, *Working Notes for the AAAI Symposium on Adaptation, Co-evolution and Learning in Multiagent Systems*, páginas 63–67, Stanford University, CA.
- [Moriarty y Miikkulainen, 1996] MORIARTY, D. E. Y MIKKULAINEN, R. (1996). Efficient reinforcement learning through symbiotic evolution. *Machine Learning*, 22:11–32.
- [Moriarty y Miikkulainen, 1998a] MORIARTY, D. E. Y MIKKULAINEN, R. (1998a). Forming neural networks through efficient and adaptive coevolution. *Evolutionary Computation*, 5(4):373–399.
- [Moriarty y Miikkulainen, 1998b] MORIARTY, D. E. Y MIKKULAINEN, R. (1998b). Hierarchical evolution of neural networks. En *Proceedings of the 1998 IEEE World Congress on Computational Intelligence*, páginas 428–433, Anchorage, Alaska, USA. IEEE Press.
- [Morse, 1980] MORSE, J.Ñ. (1980). Reducing the size of the nondominated set: Pruning by clustering. *Computers and Operations Research*, 7(1–2):55–66.
- [Murdock, 1994] MURDOCK, I. (1994). The Debian distribution. *Linux Journal*, 2:20–26.
- [Myers y Montgomery, 2002] MYERS, R. H. Y MONTGOMERY, D. C. (2002). *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. John Wiley & Sons, second edition edición.
- [Potter, 1997] POTTER, M. A. (1997). *The Design and Analysis of a Computational Model of Cooperative Coevolution*. Tesis doctoral, George Mason University.
- [Potter y de Jong, 2000] POTTER, M. A. Y DE JONG, K. A. (2000). Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computation*, 8(1):1–29.

- [Rawlings et al., 1998] RAWLINGS, J. O., PANTULA, S. G., Y DICKEY, D. (1998). *Applied regression analysis: A research tool*. Springer-Verlag, New York.
- [Rosenman y Gero, 1985] ROSENMAN, M. A. Y GERO, J. S. (1985). Reducing the pareto optimal set in multicriteria optimization. *Engineering Optimization*, 8:189–206.
- [Rosin y Belew, 1997] ROSIN, C. D. Y BELEW, R. K. (1997). New methods for competitive coevolution. *Evolutionary Computation*, 5(1):1–29.
- [Sanz-Tapia et al., 2003] SANZ-TAPIA, E., GARCÍA-PEDRAJAS, N., ORTIZ-BOYER, D., Y HERVÁS-MARTÍNEZ, C. (2003). Node level crossover applied to neural network evolution. *Lecture Notes in Computer Science*, 2686:518–525.
- [Schaffer, 1985] SCHAFFER, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. En Grefenstette, J. J., editor, *Proceedings of the 1st International Conference on Genetic Algorithms and their Applications*, páginas 93–100, Pittsburgh, PA. Lawrence Erlbaum Associates.
- [Schildt, 2000] SCHILDT, H. (2000). *C: The Complete Reference*, volumen 1. McGraw Hill, 6 edición.
- [Siever et al., 2000] SIEVER, E., SPAINHOUR, S., HEKMAN, J. P., Y FIGGINS, S. (2000). *Linux in a Nutshell*. O'Reilly, 3rd edición.
- [Smith et al., 1993] SMITH, R. E., FORREST, S., Y PERELSON, A. S. (1993). Searching for diverse, cooperative populations with genetic algorithms. *Evolutionary Computation*, 1(2):127–149.
- [Spears, 1994] SPEARS, W. (1994). Simple subpopulation schemes.
- [Srinivas y Deb, 1995] SRINIVAS, N. Y DEB, K. (1995). Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248.
- [Syswerda, 1989] SYSWERDA, G. (1989). Uniform crossover in genetic algorithms. En Schaffer, J. D., editor, *Proceedings of the 3rd International Conference on Genetic Algorithms*, páginas 2–9, George Mason University. Morgan Kaufmann.
- [van der Linden, 1994] VAN DER LINDEN, P. (1994). *Expert C Programming*. SunSoft Press, Mountain View, CA.
- [Watson, 2002] WATSON, R. A. (2002). *Compositional Evolution: Interdisciplinary Investigations in Evolvability, Modularity, and Symbiosis*. Tesis doctoral, Brandeis University. Department of Computer Science.
- [Welsh y Kaufman, 1996] WELSH, M. Y KAUFMAN, L. (1996). *Running Linux*. O'Reilly & Associates, Inc., second edición.

- [Whitley y Kauth, 1988] WHITLEY, D. Y KAUTH, J. (1988). GENITOR: A different genetic algorithm. En *Proceedings of the Rocky Mountain Conference on Artificial Intelligence*, páginas 118–130. Denver, CO.
- [Wiegand, 1998] WIEGAND, R. P. (1998). Applying diffusion to a cooperative coevolutionary model. *Lecture Notes in Computer Science*, 1498:560–??
- [Wiegand et al., 2001] WIEGAND, R. P., LILES, W. C., Y JONG, K. A. D. (2001). An empirical analysis of collaboration methods in cooperative coevolutionary algorithms. En Spector, L., Goodman, E. D., Wu, A., Langdon, W. B., Voigt, H.-M., Gen, M., Sen, S., Dorigo, M., Pezeshk, S., Garzon, M. H., y Burke, E., editores, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, páginas 1235–1242, San Francisco, California, USA. Morgan Kaufmann.
- [Wiegand et al., 2002] WIEGAND, R. P., LILES, W. C., Y JONG, K. A. D. (2002). The effects of representational bias on collaboration methods in cooperative coevolution. *Lecture Notes in Computer Science*, 2439:257–??
- [Wooldridge, 1999] WOOLDRIDGE, M. J. (1999). *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, Cambridge, Massachusetts.
- [Zhao, 1998] ZHAO, Q. (1998). A general framework for cooperative co-evolutionary algorithms: a society model.
- [Zhao et al., 2000] ZHAO, Q. F., HAMMAMI, O., KURODA, K., Y SAITO, K. (2000). Cooperative co-evolutionary algorithm - how to evaluate a module ? En *Proc. 1st IEEE Symposium on Combinations of Evolutionary Computation and Neural Networks*, páginas 150–157.
- [Zitzler, 1999] ZITZLER, E. (1999). *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. Tesis doctoral, Swiss Federal Institute of Technology Zurich.
- [Zitzler y Thiele, 1999] ZITZLER, E. Y THIELE, L. (1999). Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271.

A

**Resultados del problema de cobertura de
vectores. Subproblema “medio”.**

Resultados completos. Cobertura de Vectores. Problema "medio".						
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.
Sin	Sin	Sin	Sin	Sin	49.25250	.090822
				Con	49.33400	.131955
				Total	49.29325	.119183
			Con	Sin	49.26725	.166934
				Con	49.26525	.159402
				Total	49.26625	.161109
			Total	Sin	49.25988	.132856
				Con	49.29963	.148572
				Total	49.27975	.141459
		Con	Sin	Sin	49.30650	.138593
				Con	49.30850	.129280
				Total	49.30750	.132292
			Con	Sin	49.32100	.138769
				Con	49.31350	.152247
				Total	49.31725	.143835
			Total	Sin	49.31375	.137089
				Con	49.31100	.139432
				Total	49.31238	.137394
	Total	Sin	Sin	49.27950	.118845	
			Con	49.32125	.129584	
			Total	49.30038	.125314	
		Con	Sin	49.29413	.153943	
			Con	49.28938	.155783	
			Total	49.29175	.153901	
		Total	Sin	49.28681	.136843	
			Con	49.30531	.143274	
			Total	49.29606	.139962	
	Con	Sin	Sin	Sin	49.30225	.130510
				Con	49.29000	.144213
				Total	49.29612	.135899
			Con	Sin	49.24825	.125366
				Con	49.28125	.139462
				Total	49.26475	.131953
			Total	Sin	49.27525	.129238
				Con	49.28563	.140097
				Total	49.28044	.134023
Con		Sin	Sin	49.28250	.100884	
			Con	49.28825	.147463	

sigue en la página siguiente

viene de la página anterior						
Resultados completos. Cobertura de Vectores. Problema "medio".						
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.
				Total	49.28538	.124743
			Con	Sin	49.30275	.160177
				Con	49.25775	.166903
				Total	49.28025	.163063
			Total	Sin	49.29263	.132525
				Con	49.27300	.156216
				Total	49.28281	.144274
		Total	Sin	Sin	49.29238	.115570
				Con	49.28913	.143968
				Total	49.29075	.129725
			Con	Sin	49.27550	.144630
				Con	49.26950	.152277
				Total	49.27250	.147590
			Total	Sin	49.28394	.130354
				Con	49.27931	.147571
				Total	49.28163	.138810
	Total	Sin	Sin	Sin	49.27738	.113804
				Con	49.31200	.138244
				Total	49.29469	.127011
			Con	Sin	49.25775	.146033
				Con	49.27325	.148054
				Total	49.26550	.146321
			Total	Sin	49.26756	.130457
				Con	49.29263	.143653
				Total	49.28009	.137358
		Con	Sin	Sin	49.29450	.120266
				Con	49.29838	.137264
				Total	49.29644	.128241
			Con	Sin	49.31187	.148210
				Con	49.28563	.160189
				Total	49.29875	.153904
			Total	Sin	49.30319	.134391
				Con	49.29200	.148359
				Total	49.29759	.141213
		Total	Sin	Sin	49.28594	.116655
				Con	49.30519	.137052
				Total	49.29556	.127229
			Con	Sin	49.28481	.148706
				Con	49.27944	.153388
				Total	49.28213	.150614

sique en la página siguiente

<i>viene de la página anterior</i>						
Resultados completos. Cobertura de Vectores. Problema "medio".						
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.
Con	Sin	Sin	Total	Sin	49.28537	.133225
				Con	49.29231	.145565
				Total	49.28884	.139356
			Sin	Sin	49.27475	.125126
				Con	49.34875	.127195
				Total	49.31175	.130052
			Con	Sin	49.31300	.155059
				Con	49.27475	.098214
				Total	49.29387	.129568
			Total	Sin	49.29388	.140414
				Con	49.31175	.118260
				Total	49.30281	.129299
			Sin	49.36775	.186473	
			Con	49.28800	.150739	
			Total	49.32788	.172165	
			Con	49.29325	.134480	
			Con	49.32400	.168450	
			Total	49.30862	.151251	
		Total	Sin	49.33050	.164846	
			Con	49.30600	.158827	
			Total	49.31825	.161308	
			Sin	49.32125	.163663	
			Con	49.31838	.141060	
			Total	49.31981	.151817	
			Con	49.30313	.143610	
			Con	49.29938	.138366	
			Total	49.30125	.140130	
		Total	Sin	49.31219	.153258	
			Con	49.30888	.139161	
			Total	49.31053	.145928	
			Sin	49.25350	.132507	
			Con	49.25875	.088806	
			Total	49.25613	.111370	
			Con	49.27450	.117506	
			Con	49.26100	.137061	
			Total	49.26775	.126197	
Total	Sin	49.26400	.124072			
	Con	49.25988	.113998			
	Total	49.26194	.118403			
	Con	49.29150	.121093			

sigue en la página siguiente

<i>viene de la página anterior</i>						
Resultados completos. Cobertura de Vectores. Problema "medio".						
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.
				Con	49.31675	.177366
				Total	49.30412	.150443
			Con	Sin	49.33200	.153369
				Con	49.32775	.213877
				Total	49.32987	.183710
			Total	Sin	49.31175	.137927
				Con	49.32225	.194016
				Total	49.31700	.167339
		Total	Sin	Sin	49.27250	.126759
				Con	49.28775	.141530
				Total	49.28013	.133715
			Con	Sin	49.30325	.137964
				Con	49.29437	.180499
				Total	49.29881	.159688
			Total	Sin	49.28787	.132545
				Con	49.29106	.161194
				Total	49.28947	.147110
	Total	Sin	Sin	Sin	49.26413	.127661
				Con	49.30375	.117477
				Total	49.28394	.123516
			Con	Sin	49.29375	.137187
				Con	49.26788	.117898
				Total	49.28081	.127759
			Total	Sin	49.27894	.132509
				Con	49.28581	.118325
				Total	49.28238	.125269
		Con	Sin	Sin	49.32963	.159921
				Con	49.30238	.163119
				Total	49.31600	.161087
			Con	Sin	49.31262	.143719
				Con	49.32588	.190033
				Total	49.31925	.167538
			Total	Sin	49.32113	.151313
				Con	49.31413	.176360
				Total	49.31763	.163835
		Total	Sin	Sin	49.29688	.147503
				Con	49.30306	.141241
				Total	49.29997	.143985
			Con	Sin	49.30319	.139921
				Con	49.29687	.159817

sique en la página siguiente

<i>viene de la página anterior</i>									
Resultados completos. Cobertura de Vectores. Problema "medio".									
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.			
Total	Sin	Sin	Total	Total	49.30003	.149759			
				Sin	49.30003	.143344			
				Con	49.29997	.150372			
			Sin	Sin	Sin	Sin	Total	49.30000	.146670
							Sin	49.26363	.108504
							Con	49.34138	.128142
			Con	Sin	Sin	Con	Total	49.30250	.124293
							Sin	49.29013	.160706
							Con	49.27000	.130772
			Total	Sin	Sin	Total	Total	49.28006	.145927
							Sin	49.27688	.136892
							Con	49.30569	.133561
			Con	Sin	Con	Sin	Total	49.29128	.135583
							Sin	49.33713	.165106
							Con	49.29825	.138996
			Total	Sin	Sin	Con	Total	49.31769	.152898
							Sin	49.30713	.135608
							Con	49.31875	.158570
			Con	Sin	Sin	Total	Total	49.31294	.146717
							Sin	49.32212	.150877
							Con	49.30850	.148517
			Total	Sin	Total	Sin	Total	49.31531	.149386
							Sin	49.30038	.143657
							Con	49.31981	.134591
			Con	Sin	Con	Total	Total	49.31009	.139101
							Sin	49.29863	.147991
							Con	49.29438	.146483
			Total	Sin	Sin	Total	Total	49.29650	.146790
							Sin	49.29950	.145383
							Con	49.30709	.140799
			Con	Sin	Sin	Total	Total	49.30330	.142935
							Sin	49.27787	.132142
							Con	49.27438	.119267
			Total	Sin	Con	Sin	Total	49.27612	.125082
							Sin	49.26137	.120666
							Con	49.27113	.136867
Total	Sin	Total	Total	Total	49.26625	.128296			
				Sin	49.26963	.126004			
				Con	49.27275	.127565			
Con	Sin	Total	Total	Total	49.27119	.126397			

sique en la página siguiente

<i>viene de la página anterior</i>						
Resultados completos. Cobertura de Vectores. Problema "medio".						
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.
		Con	Sin	Sin	49.28700	.110104
				Con	49.30250	.161642
				Total	49.29475	.137638
			Con	Sin	49.31738	.155493
				Con	49.29275	.192647
				Total	49.30506	.174388
			Total	Sin	49.30219	.134738
				Con	49.29763	.176761
				Total	49.29991	.156682
		Total	Sin	Sin	49.28244	.120938
				Con	49.28844	.141849
				Total	49.28544	.131428
			Con	Sin	49.28937	.141131
				Con	49.28194	.166396
				Total	49.28566	.153841
			Total	Sin	49.28591	.131055
				Con	49.28519	.154158
				Total	49.28555	.142850
	Total	Sin	Sin	Sin	49.27075	.120348
				Con	49.30788	.127534
				Total	49.28931	.124997
			Con	Sin	49.27575	.141940
				Con	49.27056	.133006
				Total	49.27316	.137137
			Total	Sin	49.27325	.131197
				Con	49.28922	.131230
				Total	49.28123	.131251
		Con	Sin	Sin	49.31206	.141698
				Con	49.30038	.149803
				Total	49.30622	.145466
			Con	Sin	49.31225	.145055
				Con	49.30575	.175800
				Total	49.30900	.160688
			Total	Sin	49.31216	.142935
				Con	49.30306	.162828
				Total	49.30761	.153032
		Total	Sin	Sin	49.29141	.132671
				Con	49.30413	.138728
				Total	49.29777	.135670
			Con	Sin	49.29400	.144221

sigue en la página siguiente

<i>viene de la página anterior</i>						
Resultados completos. Cobertura de Vectores. Problema "medio".						
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.
				Con	49.28816	.156387
				Total	49.29108	.150219
			Total	Sin	49.29270	.138355
				Con	49.29614	.147806
				Total	49.29442	.143057

Cuadro A.1: Resultados completos. Cobertura de Vectores. Problema "medio".

B

Resultados del problema de cobertura de vectores. Subproblema “cuarto”.

Resultados completos. Cobertura de Vectores. Problema "cuarto".						
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.
Sin	Sin	Sin	Sin	Sin	43.43625	.189465
				Con	43.81525	.152923
				Total	43.62575	.256344
			Con	Sin	43.68200	.207595
				Con	43.73350	.192942
				Total	43.70775	.199528
			Total	Sin	43.55912	.232312
		Con		43.77437	.176756	
		Total		43.66675	.231940	
		Con	Sin	Sin	43.79875	.179992
				Con	43.76850	.155590
				Total	43.78363	.166768
			Con	Sin	43.76875	.160385
				Con	43.69325	.160470
				Total	43.73100	.162907
				Total	Sin	43.78375
		Total	Sin	Sin	43.73088	.160595
				Con	43.75731	.165928
				Total	43.61750	.258777
			Con	Sin	43.79187	.154101
				Con	43.70469	.229085
	Total			43.72538	.188300	
	Total			Con	43.71338	.176342
	Total	Sin	Sin	43.71937	.181362	
			Con	43.67144	.231320	
			Total	43.75263	.169218	
		Con	Sin	43.71203	.206087	
			Con	43.77625	.198593	
			Total	43.77900	.198823	
	Con	Sin	Sin	Sin	43.77762	.196149
				Con	43.75800	.112734
				Total	43.75200	.153592
			Con	Sin	43.75500	.133017
				Con	43.75500	.133017
				Total	43.76713	.159659
		Total	Sin	Sin	43.76550	.175893
				Con	43.76631	.166907
				Total	43.76631	.166907
			Con	Sin	43.86375	.126635
				Con	43.75875	.144029
				Total	43.75875	.144029

sique en la página siguiente

viene de la página anterior						
Resultados completos. Cobertura de Vectores. Problema "cuarto".						
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.
				Total	43.81125	.144034
			Con	Sin	43.73875	.130766
				Con	43.75050	.148492
				Total	43.74463	.138232
			Total	Sin	43.80125	.141950
				Con	43.75463	.144450
				Total	43.77794	.144217
		Total	Sin	Sin	43.82000	.170264
				Con	43.76888	.171668
				Total	43.79444	.171818
			Con	Sin	43.74837	.120901
				Con	43.75125	.149116
				Total	43.74981	.134890
			Total	Sin	43.78419	.151084
				Con	43.76006	.160013
				Total	43.77213	.155593
	Total	Sin	Sin	Sin	43.60625	.257572
				Con	43.79713	.176035
				Total	43.70169	.239319
			Con	Sin	43.72000	.169316
				Con	43.74275	.172385
				Total	43.73138	.170158
			Total	Sin	43.66312	.224009
				Con	43.76994	.175262
				Total	43.71653	.207519
		Con	Sin	Sin	43.83125	.157096
				Con	43.76362	.148069
				Total	43.79744	.155450
			Con	Sin	43.75375	.145235
				Con	43.72188	.155331
				Total	43.73781	.150272
			Total	Sin	43.79250	.155297
				Con	43.74275	.152236
				Total	43.76763	.155308
		Total	Sin	Sin	43.71875	.240316
				Con	43.78037	.162498
				Total	43.74956	.206807
			Con	Sin	43.73688	.157652
				Con	43.73231	.163376
				Total	43.73459	.160050

sigue en la página siguiente

<i>viene de la página anterior</i>						
Resultados completos. Cobertura de Vectores. Problema "cuarto".						
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.
Con	Sin	Sin	Total	Sin	43.72781	.202795
				Con	43.75634	.164204
				Total	43.74208	.184775
				Sin	43.73825	.155895
				Con	43.71975	.197114
				Total	43.72900	.175661
				Con	43.76400	.221143
				Con	43.75225	.107526
				Total	43.75813	.171736
				Total	43.75113	.189302
				Con	43.73600	.157583
				Total	43.74356	.173227
				Sin	43.78625	.141643
				Con	43.78875	.153879
				Total	43.78750	.145985
				Con	43.79050	.116550
				Con	43.74750	.152880
				Total	43.76900	.135936
				Total	43.78838	.128049
				Con	43.76813	.152835
				Total	43.77825	.140463
				Sin	43.76225	.149014
				Con	43.75425	.178004
				Total	43.75825	.163157
				Con	43.77725	.174994
				Con	43.74988	.130480
				Total	43.76356	.153987
				Total	43.76975	.161668
				Con	43.75206	.155086
				Total	43.76091	.158161
				Con	43.76000	.132387
				Con	43.72025	.175420
				Total	43.74013	.154710
				Con	43.79000	.149701
				Con	43.86275	.152380
				Total	43.82638	.153581
	Total	43.77500	.140311			
	Con	43.79150	.177512			
	Total	43.78325	.159197			
	Con	43.74375	.114855			

sigue en la página siguiente

<i>viene de la página anterior</i>						
Resultados completos. Cobertura de Vectores. Problema "cuarto".						
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.
				Con	43.78175	.156905
				Total	43.76275	.137080
			Con	Sin	43.78400	.146957
				Con	43.78425	.151608
				Total	43.78413	.147374
			Total	Sin	43.76387	.131770
				Con	43.78300	.152294
				Total	43.77344	.141825
		Total	Sin	Sin	43.75188	.122609
				Con	43.75100	.167199
				Total	43.75144	.145679
			Con	Sin	43.78700	.146453
				Con	43.82350	.155210
				Total	43.80525	.151057
			Total	Sin	43.76944	.135359
				Con	43.78725	.164390
				Total	43.77834	.150367
	Total	Sin	Sin	Sin	43.74913	.143178
				Con	43.72000	.184175
				Total	43.73456	.164562
			Con	Sin	43.77700	.186859
				Con	43.80750	.141689
				Total	43.79225	.165480
			Total	Sin	43.76306	.165994
				Con	43.76375	.169099
				Total	43.76341	.167027
		Con	Sin	Sin	43.76500	.129090
				Con	43.78525	.153435
				Total	43.77512	.141253
			Con	Sin	43.78725	.130958
				Con	43.76587	.151429
				Total	43.77656	.141075
			Total	Sin	43.77612	.129686
				Con	43.77556	.151781
				Total	43.77584	.140722
		Total	Sin	Sin	43.75706	.135686
				Con	43.75262	.171597
				Total	43.75484	.154216
			Con	Sin	43.78212	.160406
				Con	43.78669	.147206

sigue en la página siguiente

<i>viene de la página anterior</i>						
Resultados completos. Cobertura de Vectores. Problema "cuarto".						
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.
Total	Sin	Sin	Total	Total	43.78441	.153480
				Sin	43.76959	.148626
				Con	43.76966	.160277
			Sin	Total	43.76963	.154319
				Sin	43.58725	.229595
				Con	43.76750	.180721
			Con	Total	43.67738	.224437
				Sin	43.72300	.215742
				Con	43.74288	.154463
			Total	Total	43.73294	.186698
				Sin	43.65513	.231660
				Con	43.75519	.167497
			Sin	Total	43.70516	.207660
				Sin	43.79250	.159992
				Con	43.77862	.153084
			Con	Total	43.78556	.155738
				Sin	43.77963	.138820
				Con	43.72038	.157119
		Total	Total	43.75000	.150297	
			Sin	43.78606	.148971	
			Con	43.74950	.156891	
		Sin	Total	43.76778	.153599	
			Sin	43.68987	.222093	
			Con	43.77306	.166504	
		Con	Total	43.73147	.200058	
			Sin	43.75131	.182491	
			Con	43.73163	.155221	
		Total	Total	43.74147	.169161	
			Sin	43.72059	.204949	
			Con	43.75234	.161795	
		Sin	Total	43.73647	.185032	
			Sin	43.76813	.166794	
			Con	43.74963	.187444	
		Con	Total	43.75888	.176538	
			Sin	43.77400	.131803	
			Con	43.80738	.161090	
Total	Total	43.79069	.147203			
	Sin	43.77106	.149395			
	Con	43.77850	.176069			
			Total	43.77478	.162806	

sigue en la página siguiente

<i>viene de la página anterior</i>						
Resultados completos. Cobertura de Vectores. Problema "cuarto".						
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.
		Con	Sin	Sin	43.80375	.133909
				Con	43.77025	.149117
				Total	43.78700	.141823
			Con	Sin	43.76137	.139201
				Con	43.76738	.149104
				Total	43.76437	.143354
			Total	Sin	43.78256	.137378
				Con	43.76881	.148171
				Total	43.77569	.142593
		Total	Sin	Sin	43.78594	.151353
				Con	43.75994	.168612
				Total	43.77294	.160242
			Con	Sin	43.76769	.134842
				Con	43.78738	.155535
				Total	43.77753	.145434
			Total	Sin	43.77681	.143177
				Con	43.77366	.162279
				Total	43.77523	.152794
	Total	Sin	Sin	Sin	43.67769	.219180
				Con	43.75856	.183165
				Total	43.71812	.205386
			Con	Sin	43.74850	.179478
				Con	43.77513	.160132
				Total	43.76181	.170070
			Total	Sin	43.71309	.202818
				Con	43.76684	.171694
				Total	43.73997	.189528
		Con	Sin	Sin	43.79813	.146701
				Con	43.77444	.150213
				Total	43.78628	.148476
			Con	Sin	43.77050	.138433
				Con	43.74388	.154018
				Total	43.75719	.146581
			Total	Sin	43.78431	.142851
				Con	43.75916	.152421
				Total	43.77173	.148019
		Total	Sin	Sin	43.73791	.195476
				Con	43.76650	.167164
				Total	43.75220	.182150
			Con	Sin	43.75950	.160150

sigue en la página siguiente

<i>viene de la página anterior</i>						
Resultados completos. Cobertura de Vectores. Problema "cuarto".						
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.
				Con	43.75950	.157392
				Total	43.75950	.158528
			Total	Sin	43.74870	.178735
				Con	43.76300	.162135
				Total	43.75585	.170654

Cuadro B.1: Resultados completos. Cobertura de Vectores. Problema "cuarto".

C

**Resultados del problema de cobertura de
vectores. Subproblema “octavo”.**

Resultados completos. Cobertura de Vectores. Problema "octavo".							
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.	
Sin	Sin	Sin	Sin	Sin	41.37225	.147537	
				Con	42.45925	.120823	
				Total	41.91575	.566289	
			Con	Sin	42.40900	.145300	
				Con	42.42950	.108104	
				Total	42.41925	.126833	
			Total	Sin	41.89062	.544511	
		Con		42.44438	.114159		
		Total		42.16750	.480035		
		Con	Sin	Sin	42.40925	.155583	
				Con	42.44025	.133658	
				Total	42.42475	.144022	
			Con	Sin	42.41125	.170069	
				Con	42.47775	.184772	
				Total	42.44450	.178487	
			Total	Sin	42.41025	.160887	
		Con		42.45900	.160301		
		Total		42.43463	.161449		
		Total	Sin	Sin	Sin	41.89075	.546015
					Con	42.44975	.126125
					Total	42.17025	.483882
	Con			Sin	42.41012	.156133	
				Con	42.45363	.151404	
	Total		Sin	42.43187	.154370		
			Con	42.15044	.476975		
			Total	42.45169	.138468		
	Con		Sin	Sin	42.22275	.133500	
				Con	42.40325	.138177	
				Total	42.31300	.162290	
			Con	Sin	42.37825	.149167	
				Con	42.38175	.188779	
	Total		Sin	42.30050	.160383		
			Con	42.39250	.163652		
		Total	42.34650	.167520			
	Con	Sin	Sin	42.42500	.181898		
			Con	42.41575	.106218		

sigue en la página siguiente

<i>viene de la página anterior</i>						
Resultados completos. Cobertura de Vectores. Problema "octavo".						
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.
				Total	42.42038	.147098
			Con	Sin	42.44025	.127893
				Con	42.43825	.193454
				Total	42.43925	.161870
			Total	Sin	42.43262	.155394
				Con	42.42700	.154463
				Total	42.42981	.153972
		Total	Sin	Sin	42.32387	.187857
				Con	42.40950	.121812
				Total	42.36669	.163105
			Con	Sin	42.40925	.140692
				Con	42.41000	.190821
				Total	42.40963	.166577
			Total	Sin	42.36656	.170408
				Con	42.40975	.159064
				Total	42.38816	.165736
	Total	Sin	Sin	Sin	41.79750	.452506
				Con	42.43125	.131216
				Total	42.11437	.459636
			Con	Sin	42.39363	.146178
				Con	42.40562	.153753
				Total	42.39963	.149183
			Total	Sin	42.09556	.448998
				Con	42.41844	.142606
				Total	42.25700	.369454
		Con	Sin	Sin	42.41713	.167259
				Con	42.42800	.119806
				Total	42.42256	.144660
			Con	Sin	42.42575	.149248
				Con	42.45800	.187790
				Total	42.44187	.169320
			Total	Sin	42.42144	.157563
				Con	42.44300	.157236
				Total	42.43222	.157276
		Total	Sin	Sin	42.10731	.460537
				Con	42.42963	.124854
				Total	42.26847	.373176
			Con	Sin	42.40969	.147670
				Con	42.43181	.172552
				Total	42.42075	.160472

sigue en la página siguiente

<i>viene de la página anterior</i>								
Resultados completos. Cobertura de Vectores. Problema "octavo".								
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.		
Con	Sin	Sin	Total	Sin	42.25850	.373117		
				Con	42.43072	.150133		
				Total	42.34461	.296753		
			Sin	Sin	42.17100	.147440		
				Con	42.39800	.133174		
				Total	42.28450	.180120		
				Con	Sin	42.39975	.104119	
					Con	42.39500	.110048	
					Total	42.39738	.105769	
				Total	Sin	42.28537	.171140	
					Con	42.39650	.120593	
					Total	42.34094	.157368	
			Con	Sin	Sin	42.41850	.175395	
					Con	42.51525	.069462	
					Total	42.46687	.140492	
					Con	Sin	42.46100	.154380
						Con	42.49225	.152415
						Total	42.47662	.152246
					Total	Sin	42.43975	.164504
						Con	42.50375	.117488
						Total	42.47175	.145639
				Total	Sin	Sin	42.29475	.203186
						Con	42.45663	.120482
						Total	42.37569	.184881
					Con	Sin	42.43038	.133620
						Con	42.44363	.140151
						Total	42.43700	.136218
					Total	Sin	42.36256	.183989
						Con	42.45012	.130022
						Total	42.40634	.164767
			Con		Sin	Sin	42.30000	.214709
						Con	42.45800	.138786
						Total	42.37900	.195560
					Con	Sin	42.37400	.114784
						Con	42.45800	.149186
						Total	42.41600	.138097
		Total		Sin	42.33700	.174017		
				Con	42.45800	.142220		
				Total	42.39750	.169237		
	Con	Sin		Sin	42.40075	.129800		
				Sin	42.40075	.129800		

sigue en la página siguiente

<i>viene de la página anterior</i>						
Resultados completos. Cobertura de Vectores. Problema "octavo".						
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.
				Con	42.45650	.152696
				Total	42.42863	.142703
			Con	Sin	42.36900	.138323
				Con	42.49650	.121103
				Total	42.43275	.143647
			Total	Sin	42.38488	.133371
				Con	42.47650	.137529
				Total	42.43069	.142282
		Total	Sin	Sin	42.35038	.182400
				Con	42.45725	.144026
				Total	42.40381	.171920
			Con	Sin	42.37150	.125485
				Con	42.47725	.135528
				Total	42.42438	.140258
			Total	Sin	42.36094	.155919
				Con	42.46725	.139318
				Total	42.41409	.156735
	Total	Sin	Sin	Sin	42.23550	.193175
				Con	42.42800	.137649
				Total	42.33175	.192762
			Con	Sin	42.38688	.108950
				Con	42.42650	.133269
				Total	42.40669	.122578
			Total	Sin	42.31119	.173445
				Con	42.42725	.134618
				Total	42.36922	.165347
		Con	Sin	Sin	42.40963	.152565
				Con	42.48588	.120809
				Total	42.44775	.142013
			Con	Sin	42.41500	.151995
				Con	42.49437	.135893
				Total	42.45469	.148717
			Total	Sin	42.41231	.151338
				Con	42.49013	.127827
				Total	42.45122	.144987
		Total	Sin	Sin	42.32256	.193878
				Con	42.45694	.131934
				Total	42.38975	.178514
			Con	Sin	42.40094	.132156
				Con	42.46044	.138024

sigue en la página siguiente

<i>viene de la página anterior</i>									
Resultados completos. Cobertura de Vectores. Problema "octavo".									
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.			
Total	Sin	Sin	Total	Total	42.43069	.137963			
				Sin	42.36175	.169997			
				Con	42.45869	.134600			
			Sin	Sin	Sin	Sin	Total	42.41022	.160596
							Sin	41.77163	.429866
							Con	42.42863	.129283
			Con	Sin	Sin	Con	Total	42.10012	.456894
							Sin	42.40438	.124855
							Con	42.41225	.109081
			Total	Sin	Sin	Total	Total	42.40831	.116556
							Sin	42.08800	.447525
							Con	42.42044	.119135
			Con	Sin	Sin	Sin	Total	42.25422	.366557
							Sin	42.41388	.163713
							Con	42.47775	.111786
			Con	Sin	Sin	Con	Total	42.44581	.142945
							Sin	42.43613	.162285
							Con	42.48500	.167343
			Total	Sin	Sin	Total	Total	42.46056	.165623
							Sin	42.42500	.162352
							Con	42.48138	.141446
	Total	Sin	Sin	Sin	Total	42.45319	.154390		
					Sin	42.09275	.457035		
					Con	42.45319	.122602		
	Con	Sin	Sin	Con	Total	42.27297	.379388		
					Sin	42.42025	.144750		
					Con	42.44863	.145047		
	Total	Sin	Sin	Total	Total	42.43444	.145141		
					Sin	42.25650	.375735		
					Con	42.45091	.133891		
	Con	Sin	Sin	Sin	Total	42.35370	.297960		
					Sin	42.26137	.180753		
					Con	42.43062	.139478		
	Con	Sin	Sin	Con	Total	42.34600	.181618		
					Sin	42.37613	.131390		
					Con	42.41988	.172324		
	Total	Sin	Sin	Total	Total	42.39800	.153840		
					Sin	42.31875	.167288		
					Con	42.42525	.155862		
					Total	42.37200	.169788		

sigue en la página siguiente

<i>viene de la página anterior</i>						
Resultados completos. Cobertura de Vectores. Problema "octavo".						
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.
		Con	Sin	Sin	42.41288	.156455
				Con	42.43613	.131459
				Total	42.42450	.144057
			Con	Sin	42.40463	.136351
				Con	42.46737	.162011
				Total	42.43600	.152094
			Total	Sin	42.40875	.145875
				Con	42.45175	.147432
				Total	42.43025	.147776
		Total	Sin	Sin	42.33713	.184456
				Con	42.43338	.134696
				Total	42.38525	.168077
			Con	Sin	42.39037	.133814
				Con	42.44362	.167895
				Total	42.41700	.153674
			Total	Sin	42.36375	.162835
				Con	42.43850	.151811
				Total	42.40112	.161568
	Total	Sin	Sin	Sin	42.01650	.409969
				Con	42.42962	.133627
				Total	42.22306	.367855
			Con	Sin	42.39025	.128141
				Con	42.41606	.143348
				Total	42.40316	.136146
			Total	Sin	42.20337	.356103
				Con	42.42284	.138304
				Total	42.31311	.291237
		Con	Sin	Sin	42.41337	.159109
				Con	42.45694	.123040
				Total	42.43516	.143449
			Con	Sin	42.42038	.149769
				Con	42.47619	.163893
				Total	42.44828	.158980
			Total	Sin	42.41688	.154063
				Con	42.46656	.144779
				Total	42.44172	.151319
		Total	Sin	Sin	42.21494	.368391
				Con	42.44328	.128769
				Total	42.32911	.298302
			Con	Sin	42.40531	.139756

sigue en la página siguiente

<i>viene de la página anterior</i>						
Resultados completos. Cobertura de Vectores. Problema "octavo".						
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.
			Total	Con	42.44612	.156413
				Total	42.42572	.149490
				Sin	42.31012	.294054
				Con	42.44470	.143042
				Total	42.37741	.240657

Cuadro C.1: Resultados completos. Cobertura de Vectores. Problema "octavo".

D

Resultados del problema de evolución de superficies de respuesta. Subproblema “Leuconostoc” .

134D. Resultados del problema de evolución de superficies de respuesta. Subproblema "Leuconostoc".

Resultados completos. Superficies de respuesta. Problema "Inlag".							
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.	
Sin	Sin	Sin	Sin	Sin	6.7569128	.10948577	
				Con	6.5107038	.18285498	
				Total	6.6338083	.19409403	
			Con	Sin	6.5435211	.15300944	
				Con	6.6388599	.19901155	
				Total	6.5911905	.18174562	
			Total	Sin	6.6502169	.17006347	
				Con	6.5747819	.19948840	
				Total	6.6124994	.18805399	
			Con	Sin	Sin	6.4491334	.15575290
					Con	6.5374660	.16411181
					Total	6.4932997	.16413471
		Con		Sin	6.5764422	.26240337	
				Con	6.5296238	.13976294	
				Total	6.5530329	.20886228	
		Total		Sin	6.5127878	.22252921	
				Con	6.5335449	.15050990	
				Total	6.5231663	.18904637	
		Total		Sin	Sin	6.6030231	.20481122
					Con	6.5240849	.17202908
					Total	6.5635540	.19208215
			Con	Sin	6.5599816	.21267038	
				Con	6.5842418	.17852471	
				Total	6.5721117	.19547611	
			Total	Sin	6.5815023	.20857953	
				Con	6.5541634	.17680408	
				Total	6.5678329	.19322389	
			Con	Sin	Sin	6.5758347	.11285683
					Con	6.5897526	.19891968
					Total	6.5827937	.15978715
		Con		Sin	6.6038136	.19339465	
				Con	6.5407388	.13547227	
				Total	6.5722762	.16787621	
		Total		Sin	6.5898241	.15692993	
				Con	6.5652457	.16980662	
				Total	6.5775349	.16292704	
Con	Sin	Sin		6.5425860	.17063047		
		Con		6.5646604	.17861254		

sigue en la página siguiente

D. Resultados del problema de evolución de superficies de respuesta. Subproblema "Leuconostoc".135

viene de la página anterior						
Resultados completos. Superficies de respuesta. Problema "Inlag".						
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.
				Total	6.5536232	.17277520
			Con	Sin	6.5565985	.17978737
				Con	6.5590296	.21045542
				Total	6.5578141	.19320135
			Total	Sin	6.5495922	.17315254
				Con	6.5618450	.19268691
				Total	6.5557186	.18212162
		Total	Sin	Sin	6.5592103	.14377972
				Con	6.5772065	.18703176
				Total	6.5682084	.16600150
			Con	Sin	6.5802060	.18584991
				Con	6.5498842	.17494217
				Total	6.5650451	.17998035
			Total	Sin	6.5697082	.16543443
				Con	6.5635454	.18046249
				Total	6.5666268	.17259406
	Total	Sin	Sin	Sin	6.6663737	.14301201
				Con	6.5502282	.19279203
				Total	6.6083010	.17849662
			Con	Sin	6.5736673	.17481170
				Con	6.5897994	.17522776
				Total	6.5817333	.17409792
			Total	Sin	6.6200205	.16540479
				Con	6.5700138	.18412933
				Total	6.5950172	.17626012
		Con	Sin	Sin	6.4958597	.16805331
				Con	6.5510632	.16986139
				Total	6.5234614	.17016916
			Con	Sin	6.5665203	.22224607
				Con	6.5443267	.17696336
				Total	6.5554235	.19992136
			Total	Sin	6.5311900	.19897307
				Con	6.5476949	.17238078
				Total	6.5394425	.18575063
		Total	Sin	Sin	6.5811167	.17719976
				Con	6.5506457	.18053572
				Total	6.5658812	.17896590
			Con	Sin	6.5700938	.19870361
				Con	6.5670630	.17646911
				Total	6.5685784	.18732981

sigue en la página siguiente

viene de la página anterior									
Resultados completos. Superficies de respuesta. Problema "Inlag".									
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.			
Con	Sin	Sin	Total	Sin	6.5756053	.18774745			
				Con	6.5588544	.17814216			
				Total	6.5672298	.18291322			
			Sin	Sin	6.5249001	.18704871			
				Con	6.5406292	.20744616			
				Total	6.5327647	.19512491			
				Con	Sin	6.5309936	.23095995		
					Con	6.5168870	.17666975		
					Total	6.5239403	.20308703		
				Total	Sin	6.5279469	.20746553		
					Con	6.5287581	.19056683		
					Total	6.5283525	.19793114		
		Con	Sin	Sin	Sin	6.4305808	.14545233		
				Con	6.4660180	.18705277			
				Total	6.4482994	.16635730			
				Con	Sin	6.4663830	.16311293		
					Con	6.5226438	.13870043		
					Total	6.4945134	.15213707		
		Total	Sin	Total	Sin	6.4484819	.15361457		
					Con	6.4943309	.16504616		
					Total	6.4714064	.16009142		
					Sin	Sin	6.4777404	.17214259	
					Con	6.5033236	.19859110		
					Total	6.4905320	.18510614		
	Con			Sin	Con	Sin	6.4986883	.20004894	
						Con	6.5197654	.15680149	
						Total	6.5092268	.17890419	
						Total	Sin	6.4882144	.18573254
						Con	6.5115445	.17797692	
						Total	6.4998794	.18170036	
	Con	Sin	Sin	Sin	6.5912875	.15362569			
				Con	6.4969940	.15421469			
				Total	6.5441408	.15926032			
			Con	Sin	Con	Sin	6.5874727	.22086271	
						Con	6.5174883	.14995152	
						Total	6.5524805	.18967099	
						Total	Sin	6.5893801	.18779340
						Con	6.5072411	.15049364	
						Total	6.5483106	.17406566	
			Con	Sin	Sin	Sin	6.5170950	.12870592	
						Sin	6.5170950	.12870592	

sigue en la página siguiente

D. Resultados del problema de evolución de superficies de respuesta. Subproblema "Leuconostoc".137

<i>viene de la página anterior</i>						
Resultados completos. Superficies de respuesta. Problema "Inlag".						
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.
				Con	6.4943424	.18034858
				Total	6.5057187	.15507669
			Con	Sin	6.5182525	.19525798
				Con	6.5380787	.17066994
				Total	6.5281656	.18128848
			Total	Sin	6.5176738	.16323181
				Con	6.5162105	.17471967
				Total	6.5169421	.16800146
		Total	Sin	Sin	6.5541913	.14484303
				Con	6.4956682	.16563151
				Total	6.5249297	.15737634
			Con	Sin	6.5528626	.20872806
				Con	6.5277835	.15891456
				Total	6.5403231	.18475463
			Total	Sin	6.5535270	.17850875
				Con	6.5117258	.16208474
				Total	6.5326264	.17124596
	Total	Sin	Sin	Sin	6.5580938	.17225847
				Con	6.5188116	.18176808
				Total	6.5384527	.17705945
			Con	Sin	6.5592332	.22487790
				Con	6.5171876	.16174205
				Total	6.5382104	.19577333
			Total	Sin	6.5586635	.19903263
				Con	6.5179996	.17095627
				Total	6.5383316	.18606323
		Con	Sin	Sin	6.4738379	.14246524
				Con	6.4801802	.18192668
				Total	6.4770090	.16238559
			Con	Sin	6.4923178	.17951524
				Con	6.5303612	.15370110
				Total	6.5113395	.16714604
			Total	Sin	6.4830778	.16129188
				Con	6.5052707	.16923114
				Total	6.4941742	.16516404
		Total	Sin	Sin	6.5159659	.16268261
				Con	6.4994959	.18173527
				Total	6.5077309	.17212735
			Con	Sin	6.5257755	.20495722
				Con	6.5237744	.15691113

sigue en la página siguiente

138D. Resultados del problema de evolución de superficies de respuesta. Subproblema "Leuconostoc".

viene de la página anterior							
Resultados completos. Superficies de respuesta. Problema "Inlag".							
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.	
Total	Sin	Sin	Total	Total	6.5247749	.18194990	
				Sin	6.5208707	.18451404	
				Con	6.5116352	.16968031	
			Sin	Total	6.5162529	.17703480	
				Sin	6.6409064	.19153976	
				Con	6.5256665	.19360828	
			Con	Total	6.5832865	.19994617	
				Sin	6.5372573	.19347714	
				Con	6.5778735	.19574397	
			Total	Total	6.5575654	.19445505	
				Sin	6.5890819	.19827028	
				Con	6.5517700	.19521835	
			Con	Total	6.5704259	.19702138	
				Sin	6.4398571	.14904255	
				Con	6.5017420	.17741414	
			Con	Total	6.4707995	.16575427	
				Sin	6.5214126	.22273912	
				Con	6.5261337	.13748151	
			Total	Total	6.5237732	.18392649	
				Sin	6.4806348	.19272383	
				Con	6.5139379	.15817791	
			Total	Total	6.4972863	.17653583	
				Sin	6.5403818	.19826981	
				Con	6.5137043	.18490112	
		Con	Total	6.5270430	.19156613		
			Sin	6.5293350	.20745046		
			Con	6.5520036	.17007064		
		Total	Total	6.5406693	.18942768		
			Sin	6.5348584	.20234884		
			Con	6.5328539	.17812011		
		Total	Total	6.5338561	.19032345		
			Sin	6.5835611	.13328211		
			Con	6.5433733	.18185050		
		Con	Total	6.5634672	.15969974		
			Sin	6.5956431	.20507190		
			Con	6.5291136	.14154201		
		Total	Total	6.5623783	.17824674		
			Sin	6.5896021	.17195249		
			Con	6.5362434	.16207175		
					Total	6.5629228	.16869540

sigue en la página siguiente

D. Resultados del problema de evolución de superficies de respuesta. Subproblema "Leuconostoc".139

viene de la página anterior						
Resultados completos. Superficies de respuesta. Problema "Inlag".						
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.
		Con	Sin	Sin	6.5298405	.14973629
				Con	6.5295014	.18070937
				Total	6.5296709	.16489347
			Con	Sin	6.5374255	.18627512
				Con	6.5485541	.18942314
				Total	6.5429898	.18674695
			Total	Sin	6.5336330	.16796648
				Con	6.5390278	.18419181
				Total	6.5363304	.17573160
		Total	Sin	Sin	6.5567008	.14341831
				Con	6.5364373	.18026501
				Total	6.5465691	.16269165
			Con	Sin	6.5665343	.19684696
				Con	6.5388338	.16643143
				Total	6.5526841	.18223104
			Total	Sin	6.5616176	.17174563
				Con	6.5376356	.17294392
				Total	6.5496266	.17249406
	Total	Sin	Sin	Sin	6.6122338	.16647431
				Con	6.5345199	.18684142
				Total	6.5733768	.18064921
			Con	Sin	6.5664502	.20025951
				Con	6.5534935	.17148622
				Total	6.5599719	.18595521
			Total	Sin	6.5893420	.18499419
				Con	6.5440067	.17901657
				Total	6.5666744	.18315691
		Con	Sin	Sin	6.4848488	.15519246
				Con	6.5156217	.17847973
				Total	6.5002352	.16742812
			Con	Sin	6.5294190	.20417363
				Con	6.5373439	.16483835
				Total	6.5333815	.18500957
			Total	Sin	6.5071339	.18215023
				Con	6.5264828	.17159964
				Total	6.5168084	.17694152
		Total	Sin	Sin	6.5485413	.17268051
				Con	6.5250708	.18237944
				Total	6.5368060	.17770674
			Con	Sin	6.5479346	.20244295

sigue en la página siguiente

140D. Resultados del problema de evolución de superficies de respuesta. Subproblema "Leuconostoc".

viene de la página anterior						
Resultados completos. Superficies de respuesta. Problema "lnlag".						
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.
				Con	6.5454187	.16786093
				Total	6.5466767	.18567016
			Total	Sin	6.5482380	.18785626
				Con	6.5352448	.17529207
				Total	6.5417414	.18165698

Cuadro D.1: Resultados completos. Superficies de respuesta. Problema "lnlag".

D. Resultados del problema de evolución de superficies de respuesta. Subproblema "Leuconostoc".141

Resultados completos. Superficies de respuesta. Problema "grate".								
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.		
Sin	Sin	Sin	Sin	Sin	11.0853997	1.01619938		
				Con	10.0656645	.78688834		
				Total	10.5755321	1.03507505		
			Con	Sin	9.8860130	.73705045		
				Con	10.3256759	.69290413		
				Total	10.1058445	.74035447		
			Total	Sin	10.4857064	1.06611529		
				Con	10.1956702	.74356967		
				Total	10.3406883	.92485256		
			Con	Sin	Sin	Sin	10.3543580	.48970112
						Con	10.4529281	.55740489
						Total	10.4036430	.52027609
	Con	Sin			9.7403148	.82063520		
		Con			9.9321209	.77781972		
		Total			9.8362178	.79515139		
	Total	Sin		10.0473363	.73593178			
		Con		10.1925245	.71809529			
		Total		10.1199304	.72613588			
		Sin		Sin	10.7198788	.87002976		
				Con	10.2592963	.70105612		
				Total	10.4895876	.81854728		
	Con	Sin	9.8131639	.77342569				
		Con	10.1288984	.75389806				
		Total	9.9710311	.77532516				
Total		Sin	10.2665214	.93655079				
		Con	10.1940973	.72630423				
		Total	10.2303093	.83619677				
Con	Sin	Sin	Sin	9.7303497	.94518820			
			Con	10.2825936	.84125217			
			Total	10.0064717	.92639868			
		Con	Sin	10.1191002	.69907463			
			Con	10.0789734	.72804595			
			Total	10.0990368	.70479075			
	Total	Sin	9.9247249	.84384505				
		Con	10.1807835	.78335207				
		Total	10.0527542	.81918609				
		Con	Sin	9.9958808	.93501854			
			Con	10.2785920	.89589936			
			Total					

sigue en la página siguiente

142D. Resultados del problema de evolución de superficies de respuesta. Subproblema "Leuconostoc".

viene de la página anterior						
Resultados completos. Superficies de respuesta. Problema "grate".						
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.
				Total	10.1372364	.91511899
			Con	Sin	10.3659880	.59664560
				Con	10.0755209	.76352835
				Total	10.2207544	.69215347
			Total	Sin	10.1809344	.79653836
				Con	10.1770564	.82801883
				Total	10.1789954	.80727510
		Total	Sin	Sin	9.8631152	.93767628
				Con	10.2805928	.85779438
				Total	10.0718540	.91729254
			Con	Sin	10.2425441	.65356363
				Con	10.0772471	.73637403
				Total	10.1598956	.69676245
			Total	Sin	10.0528297	.82545140
				Con	10.1789200	.80087964
				Total	10.1158748	.81315996
	Total	Sin	Sin	Sin	10.4078747	1.18707140
				Con	10.1741290	.81148256
				Total	10.2910019	1.01713669
			Con	Sin	10.0025566	.71880102
				Con	10.2023246	.71255760
				Total	10.1024406	.71821035
			Total	Sin	10.2052156	.99614687
				Con	10.1882268	.75890785
				Total	10.1967212	.88276037
		Con	Sin	Sin	10.1751193	.75874947
				Con	10.3657601	.74174625
				Total	10.2704397	.75167761
			Con	Sin	10.0531514	.77581726
				Con	10.0038209	.76421969
				Total	10.0284861	.76555359
			Total	Sin	10.1141354	.76492456
				Con	10.1847905	.77012698
				Total	10.1494629	.76593304
		Total	Sin	Sin	10.2914970	.99678110
				Con	10.2699445	.77845526
				Total	10.2807208	.89155446
			Con	Sin	10.0278540	.74354031
				Con	10.1030727	.74091210
				Total	10.0654634	.74085088

sique en la página siguiente

D. Resultados del problema de evolución de superficies de respuesta. Subproblema "Leuconostoc".143

viene de la página anterior								
Resultados completos. Superficies de respuesta. Problema "grate".								
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.		
Con	Sin	Sin	Total	Sin	10.1596755	.88647369		
				Con	10.1865086	.76213194		
				Total	10.1730921	.82545671		
			Sin	Sin	10.0235861	.93750627		
				Con	10.2072342	.94780708		
				Total	10.1154101	.93514205		
			Con	Sin	9.9116626	.81154689		
				Con	9.5970226	.59215054		
				Total	9.7543426	.71907606		
			Total	Sin	9.9676243	.86733115		
				Con	9.9021284	.83901975		
				Total	9.9348764	.84851528		
		Con	Sin	Sin	10.0935295	.80350096		
				Con	10.0530673	.90273068		
				Total	10.0732984	.84377912		
				Con	Sin	9.5372406	.80004381	
					Con	9.8294624	.62765095	
				Total	9.6833515	.72501523		
		Con	Sin	Total	Sin	9.8153851	.84006361	
					Con	9.9412648	.77572944	
					Total	9.8783250	.80589578	
					Sin	Sin	10.0585578	.86253986
						Con	10.1301507	.91692955
					Total	10.0943543	.88523163	
			Con	Sin	9.7244516	.81770173		
				Con	9.7132425	.61367857		
			Total	9.7188471	.71835586			
			Total	Sin	9.8915047	.85183624		
				Con	9.9216966	.80310580		
			Total	9.9066007	.82536125			
	Sin	Sin	Sin	Sin	9.8837692	.79238848		
				Con	9.8199029	.87148737		
				Total	9.8518361	.82276604		
			Con	Sin	10.0320456	.77725785		
				Con	10.0380984	.79059693		
			Total	10.0350720	.77384585			
	Con	Sin	Total	Sin	9.9579074	.77836177		
				Con	9.9290007	.82868809		
				Total	9.9434540	.79894694		
			Sin	Sin	9.8956342	.65468121		

sigue en la página siguiente

144D. Resultados del problema de evolución de superficies de respuesta. Subproblema "Leuconostoc".

viene de la página anterior						
Resultados completos. Superficies de respuesta. Problema "grate".						
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.
				Con	10.2007393	.82694984
				Total	10.0481868	.75221907
			Con	Sin	9.6700686	.61904096
				Con	9.7358017	.73253248
				Total	9.7029352	.67024113
			Total	Sin	9.7828514	.63917742
				Con	9.9682705	.80622922
				Total	9.8755610	.72889013
		Total	Sin	Sin	9.8897017	.71745027
				Con	10.0103211	.86043747
				Total	9.9500114	.78948304
			Con	Sin	9.8510571	.71736271
				Con	9.8869501	.76769939
				Total	9.8690036	.73846110
			Total	Sin	9.8703794	.71311662
				Con	9.9486356	.81258529
				Total	9.9095075	.76307283
	Total	Sin	Sin	Sin	9.9536776	.85970538
				Con	10.0135685	.91985223
				Total	9.9836231	.88514731
			Con	Sin	9.9718541	.78669945
				Con	9.8175605	.72471936
				Total	9.8947073	.75554141
			Total	Sin	9.9627658	.81883054
				Con	9.9155645	.82868577
				Total	9.9391652	.82151953
		Con	Sin	Sin	9.9945819	.73032886
				Con	10.1269033	.85776507
				Total	10.0607426	.79433663
			Con	Sin	9.6036546	.70925723
				Con	9.7826321	.67497701
				Total	9.6931433	.69380268
			Total	Sin	9.7991182	.74185109
				Con	9.9547677	.78622068
				Total	9.8769430	.76593948
		Total	Sin	Sin	9.9741298	.79284724
				Con	10.0702359	.88554187
				Total	10.0221828	.83921181
			Con	Sin	9.7877543	.76693661
				Con	9.8000963	.69606555

sigue en la página siguiente

D. Resultados del problema de evolución de superficies de respuesta. Subproblema "Leuconostoc".145

<i>viene de la página anterior</i>						
Resultados completos. Superficies de respuesta. Problema "grate".						
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.
Total	Sin	Sin	Total	Total	9.7939253	.73007846
				Sin	9.8809421	.78314201
				Con	9.9351661	.80542835
			Sin	Total	9.9080541	.79358200
				Sin	10.5544929	1.10470424
				Con	10.1364493	.86281470
			Con	Total	10.3454711	1.00708386
				Sin	9.8988378	.76530138
				Con	9.9613492	.73543578
			Total	Total	9.9300935	.74641490
				Sin	10.2266653	1.00021466
				Con	10.0488993	.80142662
		Con	Total	10.1377823	.90782218	
			Sin	10.2239438	.66992753	
			Con	10.2529977	.76770924	
		Con	Total	10.2384707	.71605386	
			Sin	9.6387777	.80652982	
			Con	9.8807916	.69954973	
		Total	Total	9.7597846	.75996304	
			Sin	9.9313607	.79333317	
			Con	10.0668947	.75340486	
		Total	Total	9.9991277	.77418041	
			Sin	10.3892183	.92286740	
			Con	10.1947235	.81357868	
		Con	Total	10.2919709	.87267065	
			Sin	9.7688077	.79207715	
			Con	9.9210704	.71431103	
		Total	Total	9.8449391	.75569056	
			Sin	10.0790130	.91198497	
			Con	10.0578970	.77538925	
		Con	Total	10.0684550	.84518540	
			Sin	9.8070594	.86438536	
			Con	10.0512483	.87731438	
		Con	Total	9.9291539	.87402331	
			Sin	10.0755729	.73099251	
			Con	10.0585359	.75044434	
Total	Total	10.0670544	.73612876			
	Sin	9.9413162	.80678265			
	Con	10.0548921	.81117280			
			Total	9.9981041	.80844226	

sigue en la página siguiente

146D. Resultados del problema de evolución de superficies de respuesta. Subproblema "Leuconostoc".

viene de la página anterior						
Resultados completos. Superficies de respuesta. Problema "grate".						
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.
		Con	Sin	Sin	9.9457575	.79831505
				Con	10.2396657	.85190229
				Total	10.0927116	.83352517
			Con	Sin	10.0180283	.69591840
				Con	9.9056613	.75830612
				Total	9.9618448	.72536695
			Total	Sin	9.9818929	.74500213
				Con	10.0726635	.81877525
				Total	10.0272782	.78161998
		Total	Sin	Sin	9.8764085	.82966390
				Con	10.1454570	.86442580
				Total	10.0109327	.85526796
			Con	Sin	10.0468006	.70973050
				Con	9.9820986	.75353188
				Total	10.0144496	.73037485
			Total	Sin	9.9616045	.77432886
				Con	10.0637778	.81246491
				Total	10.0126911	.79403136
	Total	Sin	Sin	Sin	10.1807762	1.05486781
				Con	10.0938488	.86563245
				Total	10.1373125	.96284902
			Con	Sin	9.9872053	.74889048
				Con	10.0099426	.73987830
				Total	9.9985739	.74214109
			Total	Sin	10.0839907	.91703599
				Con	10.0518957	.80378048
				Total	10.0679432	.86106714
		Con	Sin	Sin	10.0848506	.74550157
				Con	10.2463317	.80577849
				Total	10.1655911	.77800793
			Con	Sin	9.8284030	.77241638
				Con	9.8932265	.72499578
				Total	9.8608147	.74742961
			Total	Sin	9.9566268	.76754198
				Con	10.0697791	.78429659
				Total	10.0132029	.77681676
		Total	Sin	Sin	10.1328134	.91177153
				Con	10.1700902	.83710840
				Total	10.1514518	.87406293
			Con	Sin	9.9078042	.76251976

sigue en la página siguiente

D. Resultados del problema de evolución de superficies de respuesta. Subproblema “Leuconostoc”.147

<i>viene de la página anterior</i>						
Resultados completos. Superficies de respuesta. Problema “grate”.						
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.
				Con	9.9515845	.73251084
				Total	9.9296943	.74681495
			Total	Sin	10.0203088	.84667849
				Con	10.0608374	.79290323
				Total	10.0405731	.81984048

Cuadro D.2: Resultados completos. Superficies de respuesta. Problema “grate”.

148D. Resultados del problema de evolución de superficies de respuesta. Subproblema "Leuconostoc".

Resultados completos. Superficies de respuesta. Problema "grate".								
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.		
Sin	Sin	Sin	Sin	Sin	14.5042481	.46251946		
				Con	14.1707004	.38623678		
				Total	14.3374742	.45323622		
			Con	Sin	14.1244620	.38038559		
				Con	14.2952679	.34893533		
				Total	14.2098650	.37052558		
			Total	Sin	14.3143550	.46010329		
				Con	14.2329842	.36874459		
				Total	14.2736696	.41630474		
			Con	Sin	Sin	Sin	14.1016562	.43815323
						Con	14.2042761	.40414534
						Total	14.1529661	.41928605
					Con	Sin	14.1466813	.41914757
						Con	14.2084424	.42682450
						Total	14.1775618	.41871468
					Total	Sin	14.1241687	.42383682
						Con	14.2063592	.41028174
						Total	14.1652640	.41652400
	Total	Sin		Sin	14.3029521	.48918953		
				Con	14.1874882	.39056196		
				Total	14.2452202	.44364136		
		Con		Sin	14.1355716	.39523173		
				Con	14.2518551	.38730337		
				Total	14.1937134	.39318090		
		Total		Sin	14.2192619	.44982966		
				Con	14.2196717	.38782126		
				Total	14.2194668	.41864876		
	Con	Sin	Sin	Sin	14.2122691	.32778059		
				Con	14.2371510	.38871203		
				Total	14.2247100	.35512356		
			Con	Sin	14.2028840	.30287833		
				Con	14.5199525	.48754649		
				Total	14.3614182	.43159288		
			Total	Sin	14.2075766	.31153933		
				Con	14.3785517	.45817197		
				Total	14.2930641	.39868143		
Con		Sin	Sin	14.3351355	.39573926			
			Con	14.4682560	.42307123			

sigue en la página siguiente

D. Resultados del problema de evolución de superficies de respuesta. Subproblema "Leuconostoc".149

viene de la página anterior						
Resultados completos. Superficies de respuesta. Problema "grate".						
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.
				Total	14.4016957	.40992771
			Con	Sin	14.0954434	.38538504
				Con	14.2819681	.40825179
				Total	14.1887057	.40308257
			Total	Sin	14.2152894	.40420912
				Con	14.3751120	.42106546
				Total	14.2952007	.41791269
		Total	Sin	Sin	14.2737023	.36401977
				Con	14.3527035	.41773919
				Total	14.3132029	.39133760
			Con	Sin	14.1491637	.34642187
				Con	14.4009603	.45991684
				Total	14.2750620	.42393221
			Total	Sin	14.2114330	.35859110
				Con	14.3768319	.43721973
				Total	14.2941324	.40712535
	Total	Sin	Sin	Sin	14.3582586	.42239974
				Con	14.2039257	.38395375
				Total	14.2810921	.40852020
			Con	Sin	14.1636730	.34170167
				Con	14.4076102	.43366425
				Total	14.2856416	.40687558
			Total	Sin	14.2609658	.39409169
				Con	14.3057679	.41966939
				Total	14.2833669	.40642103
		Con	Sin	Sin	14.2183958	.42872183
				Con	14.3362660	.42969833
				Total	14.2773309	.43058905
			Con	Sin	14.1210623	.39827105
				Con	14.2452052	.41393005
				Total	14.1831338	.40840207
			Total	Sin	14.1697291	.41405594
				Con	14.2907356	.42170558
				Total	14.2302324	.42098013
		Total	Sin	Sin	14.2883272	.42868604
				Con	14.2700959	.41032041
				Total	14.2792115	.41838207
			Con	Sin	14.1423677	.36933275
				Con	14.3264077	.42907341
				Total	14.2343877	.40959539

sique en la página siguiente

150D. Resultados del problema de evolución de superficies de respuesta. Subproblema "Leuconostoc".

viene de la página anterior									
Resultados completos. Superficies de respuesta. Problema "grate".									
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.			
Con	Sin	Sin	Total	Sin	14.2153474	.40551436			
				Con	14.2982518	.41943148			
				Total	14.2567996	.41397164			
			Sin	Sin	13.9304366	.44870849			
				Con	14.3403029	.38867583			
				Total	14.1353698	.46342260			
				Con	Sin	14.1380659	.33228306		
					Con	14.1450327	.44879340		
					Total	14.1415493	.38977997		
				Total	Sin	14.0342512	.40364920		
					Con	14.2426678	.42602843		
					Total	14.1384595	.42548011		
					Con	Sin	14.1994933	.39513006	
						Con	14.1663625	.36916032	
						Total	14.1829279	.37780444	
					Con	Sin	14.0362938	.43829029	
						Con	14.2817771	.43709034	
						Total	14.1590355	.44956963	
				Total	Sin	14.1178935	.42009271		
					Con	14.2240698	.40358725		
					Total	14.1709817	.41277894		
				Total	Sin	14.0649650	.43899038		
					Con	14.2533327	.38437993		
					Total	14.1591488	.42078298		
				Con	Sin	14.0871798	.38734048		
					Con	14.2134049	.44271298		
					Total	14.1502923	.41815940		
				Total	Sin	14.0760724	.41149516		
					Con	14.2333688	.41243105		
					Total	14.1547206	.41817565		
			Con	Sin	Sin	Sin	13.9479791	.33481436	
						Con	14.0199407	.50725982	
						Total	13.9839600	.42579142	
						Con	Sin	14.2366726	.33898258
							Con	14.2754789	.42318034
							Total	14.2560758	.37896229
		Total			Sin	14.0923259	.36326956		
					Con	14.1477098	.47890060		
					Total	14.1200179	.42325531		
		Con		Sin	13.9712420	.35243243			

sique en la página siguiente

viene de la página anterior						
Resultados completos. Superficies de respuesta. Problema "grate".						
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.
				Con	14.0398862	.34138666
				Total	14.0055641	.34423591
			Con	Sin	13.9146963	.34146261
				Con	14.0818851	.40439613
				Total	13.9982907	.37900162
			Total	Sin	13.9429692	.34370812
				Con	14.0608856	.37000276
				Total	14.0019274	.35975627
		Total	Sin	Sin	13.9596106	.33950521
				Con	14.0299134	.42689302
				Total	13.9947620	.38486240
			Con	Sin	14.0756844	.37331843
				Con	14.1786820	.42015029
				Total	14.1271832	.39828737
			Total	Sin	14.0176475	.35932514
				Con	14.1042977	.42745100
				Total	14.0609726	.39600850
	Total	Sin	Sin	Sin	13.9392079	.39087137
				Con	14.1801218	.47462715
				Total	14.0596649	.44869422
			Con	Sin	14.1873692	.33505974
				Con	14.2102558	.43558412
				Total	14.1988125	.38629069
			Total	Sin	14.0632885	.38266973
				Con	14.1951888	.45288575
				Total	14.1292387	.42313368
		Con	Sin	Sin	14.0853677	.38721152
				Con	14.1031243	.35675253
				Total	14.0942460	.37003774
			Con	Sin	13.9754951	.39265912
				Con	14.1818311	.42777504
				Total	14.0786631	.42098777
			Total	Sin	14.0304314	.39139290
				Con	14.1424777	.39336564
				Total	14.0864545	.39516134
		Total	Sin	Sin	14.0122878	.39350853
				Con	14.1416231	.41897646
				Total	14.0769554	.41032227
			Con	Sin	14.0814321	.37802310
				Con	14.1960434	.42919463

sigue en la página siguiente

152D. Resultados del problema de evolución de superficies de respuesta. Subproblema "Leuconostoc".

<i>viene de la página anterior</i>								
Resultados completos. Superficies de respuesta. Problema "grate".								
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.		
Total	Sin	Sin	Total	Total	14.1387378	.40722315		
				Sin	14.0468599	.38618861		
				Con	14.1688333	.42366073		
			Sin	Total	14.1078466	.40930547		
				Sin	14.2173423	.53547495		
				Con	14.2555016	.39198239		
			Con	Total	14.2364220	.46666177		
				Sin	14.1312639	.35260387		
				Con	14.2201503	.40401709		
			Total	Total	Sin	Total	14.1757071	.37942042
						Sin	14.1743031	.45255454
					Con	Total	14.2378260	.39591769
		Con				14.2060645	.42503691	
		Sin			Sin	14.1505748	.41478246	
					Con	14.1853193	.38253599	
		Con	Total	Sin	Total	14.1679470	.39683712	
					Con	14.0914875	.42696701	
				Con	Total	14.2451097	.42802753	
					Total	14.1682986	.43175859	
				Total	Sin	14.1210311	.41930161	
					Con	14.2152145	.40446261	
		Total	Sin	Total	Total	14.1681228	.41335964	
					Sin	14.1839585	.47708897	
					Con	14.2204105	.38644577	
					Total	14.2021845	.43315816	
				Con	Sin	14.1113757	.38958312	
					Con	14.2326300	.41374359	
					Total	14.1720029	.40516979	
					Total	14.1476671	.43569111	
				Total	Con	14.2265202	.39911361	
					Total	14.1870937	.41901233	
					Sin	14.0801241	.35336356	
					Con	14.1285459	.45941949	
		Con	Sin	Total	Total	14.1043350	.40796258	
					Con	14.2197783	.31775097	
				Con	Con	14.3977157	.46730415	
					Total	14.3087470	.40701832	
				Total	Sin	14.1499512	.34120957	
					Con	14.2631308	.47994179	
					Total	14.2065410	.41894645	

sigue en la página siguiente

D. Resultados del problema de evolución de superficies de respuesta. Subproblema "Leuconostoc".153

viene de la página anterior						
Resultados completos. Superficies de respuesta. Problema "grate".						
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.
		Con	Sin	Sin	14.1531887	.41323396
				Con	14.2540711	.43706936
				Total	14.2036299	.42565557
			Con	Sin	14.0050698	.37085983
				Con	14.1819266	.41368393
				Total	14.0934982	.40037564
			Total	Sin	14.0791293	.39718085
				Con	14.2179988	.42439075
				Total	14.1485641	.41559491
		Total	Sin	Sin	14.1166564	.38378965
				Con	14.1913085	.44999116
				Total	14.1539824	.41856338
			Con	Sin	14.1124241	.35973971
				Con	14.2898211	.45174897
				Total	14.2011226	.41666905
			Total	Sin	14.1145403	.37079365
				Con	14.2405648	.45215873
				Total	14.1775525	.41762994
	Total	Sin	Sin	Sin	14.1487332	.45602779
				Con	14.1920237	.42910412
				Total	14.1703785	.44190985
			Con	Sin	14.1755211	.33645974
				Con	14.3089330	.44313477
				Total	14.2422270	.39785796
			Total	Sin	14.1621272	.39969216
				Con	14.2504784	.43873828
				Total	14.2063028	.42134077
		Con	Sin	Sin	14.1518817	.41138238
				Con	14.2196952	.40956459
				Total	14.1857885	.41059289
			Con	Sin	14.0482787	.39973149
				Con	14.2135182	.41945080
				Total	14.1308984	.41674373
			Total	Sin	14.1000802	.40764689
				Con	14.2166067	.41324315
				Total	14.1583434	.41394451
		Total	Sin	Sin	14.1503075	.43291425
				Con	14.2058595	.41835734
				Total	14.1780835	.42593972
			Con	Sin	14.1118999	.37377783

sique en la página siguiente

154D. Resultados del problema de evolución de superficies de respuesta. Subproblema "Leuconostoc".

<i>viene de la página anterior</i>						
Resultados completos. Superficies de respuesta. Problema "grate".						
Mej.	Med.	Sust.	Dif.	Union	Media	Desv.típ.
				Con	14.2612256	.43275077
				Total	14.1865627	.41057405
			Total	Sin	14.1311037	.40425180
				Con	14.2335425	.42585084
				Total	14.1823231	.41802152

Cuadro D.3: Resultados completos. Superficies de respuesta. Problema "grate".