



UNIVERSIDADE DA CORUÑA

FACULTAD DE INFORMÁTICA

DEPARTAMENTO DE

TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

TESIS DOCTORAL

METODOLOGÍA PARA ORIENTAR PROCESOS DE EXTRACCIÓN
DE CONOCIMIENTO BASADOS EN COMPUTACIÓN EVOLUTIVA.

APLICACIÓN AL DESARROLLO DE MODELOS Y
FORMULACIONES EN EL ÁMBITO DEL HORMIGÓN
ESTRUCTURAL

Directores:

Dr. Fernando Martínez Abella

Dr. Juan Ramón Rabuñal Dopico

Doctorando:

Juan Luis Pérez Ordóñez

A Coruña, 28 de abril 2010

D. **FERNANDO MARTÍNEZ ABELLA**, Catedrático de Universidad en el área de Ingeniería de la Construcción de la Universidade da Coruña, y D. **JUAN RAMÓN RABUÑAL DOPICO**, Profesor Titular de Universidad en el Área de Ciencias de la Computación e Inteligencia Artificial de la Universidade da Coruña

HACEN CONSTAR QUE:

La memoria “**Metodología para orientar procesos de extracción de conocimiento basados en computación evolutiva. Aplicación al desarrollo de modelos y formulaciones en el ámbito del hormigón estructural**” ha sido realizada por D. **Juan Luis Pérez Ordóñez**, bajo nuestra dirección, en el Departamento de Tecnologías de la Información y las Comunicaciones de la Universidade da Coruña, y constituye la Tesis que presenta para optar al Grado de Doctor en Informática de la Universidade da Coruña.

A Coruña, 28 de abril de 2010

Fdo: Dr. Fernando Martínez Abella Fdo: Dr. Juan Ramón Rabuñal Dopico

*A Ana y
a mis hijos Iago y Antía*

AGRADECIMIENTOS

En primer lugar, mis agradecimientos se dirigen hacia las instituciones que de una manera u otra han sustentado este trabajo: el Ministerio de Ciencia e Innovación del Gobierno de España me ha concedido la ayuda FPI (Ref. BES-2006-13535) que ha permitido realizar esta tesis, la Universidade da Coruña, en concreto el Dpto. de Tecnologías de la Información y las Comunicaciones, el Dpto. de Tecnología de la Construcción, y el Centro de Innovación Tecnológica en Edificación e Ingeniería Civil (CITEEC) han sido el marco en el que se ha desarrollado esta tesis. De igual manera, dado que he empleado medios técnicos y personales derivados de los proyectos de investigación, debo manifestar mi gratitud a las entidades siguientes: La Dirección Xeral de Investigación, Desenvolvemento e Innovación de la Xunta de Galicia (Ref. 07TMT011CT, Ref. 08TIC014CT y Ref. 08TMT005CT), el Ministerio de Educación y Ciencia (Ref BIA2005-09412-C03-01), el Ministerio de Medio Ambiente (Ref. 202/PC08/3-03.2), el Ministerio de Fomento (Ref. CL/EAM-CENIT, AIE) del Gobierno de España. Agradezco también a la Universitat de les Illes Balears y a la Universidade de Aveiro su acogida en las dos estancias de investigación que he realizado durante el transcurso de la tesis.

Me gustaría expresar mi eterna gratitud a mis dos directores de tesis, Fernando Martínez y Juan Rabuñal, ya que con su dedicación, su tiempo, su trabajo y sus prudentes consejos me han sabido conducir durante el desarrollo de esta investigación. Ambos han conseguido motivarme dentro de ámbitos tan dispares como son la Inteligencia Artificial y la Ingeniería Civil. En todo momento me he sentido arropado ante las dificultades, menores con su sabia ayuda. De igual manera he de agradecer la ayuda prestada por mi tutor, Alejandro Pazos, y a Julián Dorado su apoyo y dedicación en las fases tempranas de mi etapa investigadora.

Por supuesto no puedo olvidarme de tanta gente que, de una forma u otra y en mayor o menor medida, me han ayudado durante este largo trabajo. Merece la primera mención Daniel Rivero, ya que esta tesis se ha desarrollado a partir de sus investigaciones. Alba Catoira y Javier Berrocal me ayudaron en la implementación de algunos aspectos del trabajo. Y tantas personas que, aún no habiendo participado directamente en la tesis, siempre me han prestado (y lo siguen haciendo) su apoyo incondicional. Por ello, sería injusto no citar a Nieves Pedreira, a José

Agradecimientos

Manuel Vázquez, a Francisco J. Novoa, a Marcos Gestal, a José Antonio Seoane, a Mónica Miguélez, a Álvaro Rodríguez y a Quique Fernández.

También quiero agradecer a los miembros del grupo de Construcción (gCONS), principalmente a Belén, Isabel, Manuel, Cristina, Javier, Diego, María, César y al ya mencionado Fernando, su acogimiento, comprensión y ayuda durante estos últimos 4 años. Junto a ellos he conocido con cierta profundidad varios aspectos relacionados con la ingeniería civil, lo que ha sido de gran provecho para mí. He de agradecer específicamente la ayuda prestada por Diego y Manuel durante la corrección de este documento.

No quiero olvidarme de aquellas personas que me han acogido durante mis estancias de investigación, en especial de Toni Cladera y de Ana Tomé.

De igual manera quiero agradecer a mis padres el regalo de su apoyo y ayuda en toda mi etapa formativa: sin ellos, con casi total seguridad, no me hubiese sido posible realizar este trabajo. También quiero mostrar mi agradecimiento a mi hermano Álvaro, que tantos quebraderos informáticos me da y que a cambio me brinda su ayuda en cualquier momento. Sería injusto no mencionar a Santiago y a Mercedes, ya que su incondicional ayuda en el ámbito personal me ha permitido evolucionar en el aspecto profesional de mi vida.

Con total seguridad me quedarán en el tintero muchas personas, empezando por los compañeros de trabajo y terminando por tantos amigos con que cuento: a todos los que han sufrido de algún modo este trabajo, mil perdones y muchas gracias.

Y por último quiero agradecer a mi mujer Ana su constante trabajo y su dedicación desinteresada en cada uno de los aspectos de mi vida. Su imprescindible ayuda e inagotable comprensión han sido factores clave para llevar a buen término esta tesis. También he de agradecer a Antía y a Iago sus cálidos abrazos y sus picaronas sonrisas que me regalan todos los días al regresar a casa, y su inocente comprensión en tantos momentos que este trabajo les ha robado mi presencia.

Juan Luis Pérez Ordóñez

RESUMEN

Uno de los problemas típicos ante los que se enfrenta un investigador cuando aborda el estudio de fenómenos reales es la ingente cantidad de información y de datos que es posible extraer para su análisis. En muchos casos, además, la información y los datos son difíciles de relacionar entre sí. Para descubrir, comprender y entender esa interrelación, una de las posibilidades a disposición de los expertos es el empleo de procedimientos de extracción de conocimiento mediante técnicas de Inteligencia Artificial. Su uso, tal como se conocen hoy en día, presenta diversas limitaciones para la mayoría de problemas ya que, por un lado, en ocasiones la solución no es capaz de predecir el comportamiento con un margen de error asumible y, por otro, puede resultar imposible realizar una interpretación de la solución obtenida. En estos casos el experto desearía influir en el proceso de búsqueda, alternativa que no suelen ofrecer los procedimientos disponibles. La tesis se plantea, por tanto, con el objetivo de desarrollar una metodología capaz de obtener soluciones de un problema genérico de extracción de conocimiento construido a partir de datos derivados de ensayos de laboratorio. El aporte fundamental es que ello se hace de forma supervisada, es decir, dirigiendo el proceso de búsqueda (basado en la aplicación de técnicas heurísticas) mediante las indicaciones del experto en el fenómeno concreto, lo que se denomina como “proceso de búsqueda guiado por el usuario”. Definida la metodología, se implementa mediante un sistema basado en Programación Genética desarrollando un conjunto de funcionalidades organizadas por módulos: módulo de operadores y funciones definidos por el usuario, módulo de probabilidad de aparición de nodos, módulo de coeficientes de seguridad, módulo de restricciones y módulo de ajuste de expresiones.

El sistema desarrollado se aplica en primer lugar a un problema de solución teórica conocida y probada (la predicción de la distancia máxima que alcanza un proyectil), lo que permite constatar el buen funcionamiento y las posibilidades del método. En segundo lugar se aborda la aplicación práctica, presentándose para ello tres ejemplos reales de predicción en problemas enmarcados dentro del ámbito del hormigón estructural. El primer caso real elegido es el de la predicción del esfuerzo a cortante que resisten las vigas de hormigón en determinadas condiciones. El segundo corresponde a la predicción de la capacidad adherente entre el hormigón y las barras de acero que actúan como armaduras en las estructuras. Y, por último, se aplica el método a un problema de series temporales, concretamente la predicción de las deformaciones que a lo largo del tiempo sufre el hormigón cuando tiene una carga constante aplicada (deformaciones de fluencia).

Tales ejemplos muestran que la metodología definida y desarrollada permite obtener resultados con menor error y más fácilmente interpretables gracias al proceso guiado, lo que puede extenderse a numerosos campos del conocimiento en los que aparecen problemas de regresión simbólica.

ABSTRACT

One of the most common problems in science related to the study of real phenomena is the huge amount of information and data available. Furthermore, in many cases it is difficult to establish relationships within data. Thus, in order to discover and comprehend these data correlations, Artificial Intelligence techniques of knowledge extraction can be used by experts. Nowadays, the use of these techniques presents some limitations: on one hand, the solution can't provide a behavior prediction with an assumable error; on the other, interpretation of the results obtained may be impossible. In these cases, the expert would desire to guide the search process, but most present day techniques do not allow for this possibility. The objective of this thesis is to develop a new methodology to obtain knowledge from problems, and by data obtained from laboratory testing. The main contribution of this technique is that this process will be done in a supervised way, guiding the search process (based in heuristics) with expert recommendations. This process shall be called "user guided search process". Once the methodology has been defined, it will be implemented using a Genetic Programming based system. This system will have a set of functionalities organized by modules: operators and functions module, module probability of node apparitions module, safety factors module, constraints module and expression adjustment module.

The system thus developed will be applied to a problem with theoretical known solution (the prediction of maximum distance of a projectile) to provide a validation of the proposed method. Additionally, practical application of the method will be carried out with three real examples of prediction problems in the field of structural concrete. The first case chosen is

the prediction of shear strength for concrete beams under determinate circumstances. The second corresponds to the prediction of bond between concrete and steel used as reinforcement within the structure. Finally, the method will be applied to a problem of temporal series, the prediction of strain evolution during the time that concrete endures a constant load (concrete creep).

These examples will show that the defined and implemented methodology provides more precise and easier to interpret results, thanks to the guided process. The methodology proposed can be extended to numerous areas related to symbolic regression problems.

*“El mayor enemigo del conocimiento no es la ignorancia,
sino la ilusión del conocimiento”*

Stephen Hawking

ÍNDICE DE CONTENIDOS

I.	INTRODUCCIÓN.....	1
II.	HIPÓTESIS DE TRABAJO Y OBJETIVOS.....	5
III.	FUNDAMENTOS	9
1	COMPUTACIÓN EVOLUTIVA	10
1.1	Restricciones en la PG.....	17
1.2	Funcionamiento de la PG.....	19
1.2.1	Generación inicial de árboles	20
1.2.2	Operadores genéticos	22
1.2.2.1	Cruce.....	22
1.2.2.2	Reproducción.....	28
1.2.2.3	Selección.....	29
1.2.2.4	Mutación	32
1.2.3	Evaluación	36
1.2.3.1	<i>Fitness</i> Puro: $r(i,t)$	37
1.2.3.2	<i>Fitness</i> Estandarizado: $s(i,t)$	38
1.2.3.3	<i>Fitness</i> Ajustado: $a(i,t)$	38
1.2.3.4	<i>Fitness</i> Normalizado: $n(i,t)$	39
1.2.3.5	Función de ajuste en PG	40
1.2.4	Parámetros de la PG.....	41
1.3	Limitaciones de la técnica.....	44
2	IDENTIFICACIÓN DE SISTEMAS	45
2.1	Definición.....	45
2.2	Conceptos básicos.	46
2.2.1	Estimación	48
2.2.2	Ajuste del modelo	49
2.2.3	Errores.....	49
2.3	Metodología de Identificación de Sistemas.....	50
IV.	ESTADO DE LA CUESTIÓN	55

V. METODOLOGÍA	61
1 ANTECEDENTES	61
2 METODOLOGÍA	64
2.1 <i>Diseño del experimento y conocimiento previo</i>	65
2.1.1 Desarrollo de meta-reglas	67
2.2 <i>Selección del modelo o algoritmo de extracción</i>	68
2.3 <i>Estimación de parámetros</i>	70
2.4 <i>Validación del modelo</i>	71
VI. SISTEMA BASADO EN CONOCIMIENTO APLICADO A LA PROGRAMACIÓN GENÉTICA	75
1 MÓDULO DE OPERADORES Y FUNCIONES DEFINIDOS POR EL USUARIO	78
1.1 <i>Despliegue de funciones</i>	86
1.2 <i>Ejemplo de aplicación</i>	93
1.2.1 Conjunto de datos	94
1.2.2 Configuración de parámetros	95
1.2.3 Resultados	96
2 MÓDULO DE PROBABILIDADES ASOCIADAS A ÁRBOLES	103
2.1 <i>Modificación de algoritmos</i>	103
2.2 <i>Ejemplo de aplicación</i>	105
2.2.1 Resultados	105
3 MÓDULO DE COEFICIENTE DE SEGURIDAD	107
3.1 <i>Ejemplo de aplicación</i>	111
4 MÓDULO DE RESTRICCIONES EN FUNCIONES	117
5 MÓDULO DE OPTIMIZACIÓN DE FUNCIONES CONOCIDAS (NORMATIVAS)	118
5.1 <i>Ejemplo de aplicación</i>	122
5.1.1 Resultados	123
VII. RESULTADOS Y DISCUSIÓN	127
1 CORTANTE	128
1.1 <i>Descripción</i>	128
1.1.1 Conjunto de datos	131

1.1.2	Modelos normativos actuales	133
1.1.2.1	Eurocódigo 2	133
1.1.2.2	ACI318-05	134
1.1.3	Comparación de los modelos presentados y contraste experimental	135
1.2	<i>Aplicación del Método</i>	139
1.2.1	Resultados	145
1.2.1.1	Ecuaciones	150
2	ADHERENCIA	152
2.1	<i>Descripción</i>	152
2.1.1	Modelos normativos actuales	158
2.1.1.1	ACI	158
2.1.1.2	Eurocódigo 2	159
2.1.1.3	EHE	162
2.1.1.4	FIB	165
2.1.2	Conjunto de datos	166
2.2	<i>Aplicación del Método</i>	173
2.2.1	Resultados	175
3	FLUENCIA DEL HORMIGÓN ESTRUCTURAL	183
3.1	<i>Descripción</i>	183
3.1.1	Modelos normativos actuales	187
3.1.1.1	ACI 209 Método del 1978	188
3.1.1.2	Modelo CEB MC90	189
3.2	<i>Aplicación del Método</i>	190
3.2.1	Conjunto de datos	191
3.2.2	Configuración del método	194
3.2.3	Resultados	196
VIII.	CONCLUSIONES	201
IX.	CONCLUSIONS	207
X.	FUTUROS DESARROLLOS	211
XI.	BIBLIOGRAFÍA	213

ÍNDICE DE FIGURAS

Figura 1. Ejemplo de árbol.....	15
Figura 2. Árbol para la expresión $a*((b/2)+5)$	16
Figura 3. Ejemplo de árbol con rama inútil.....	19
Figura 4. Diagrama de flujo de la Programación Genética	20
Figura 5. Árboles seleccionados para la operación de cruce.....	23
Figura 6. Selección de nodo en el 1 ^{er} progenitor y aplicación de la restricción de tipo en el segundo.....	24
Figura 7. Aplicación de restricción de altura	25
Figura 8. Selección de ramas a intercambiar.....	25
Figura 9. Resultado del cruce de dos individuos	26
Figura 10. Cruce de dos árboles iguales	27
Figura 11. Resultado del cruce de dos árboles iguales.....	27
Figura 12. Mutación puntual	33
Figura 13. Árbol seleccionado para la mutación	34
Figura 14. Árbol preparado para la mutación y subárbol generado.....	35
Figura 15. Resultado de aplicar la mutación	35
Figura 16. Abstracción de un sistema.....	46

Índice de figuras

Figura 17. Extracción del modelo mediante IS	46
Figura 18. Etapas de un proceso de identificación (Ljung, 1999)	53
Figura 19. Desarrollo de un modelo físico	62
Figura 20. Modelo de funcionamiento deseado.....	64
Figura 21. Diagrama del proceso de extracción de conocimiento.....	69
Figura 22. Diagrama de flujo de la metodología propuesta	73
Figura 23. Ejemplo de un nuevo operador-expresión	79
Figura 24. Ejemplos de árbol sintáctico con asociatividad por la izquierda y derecha.....	85
Figura 25. Función F_1 y su equivalente desplegado	88
Figura 26. Despliegue de la función F_2	88
Figura 27. Individuo I_1 (18) sin desplegar.....	89
Figura 28. Individuo I_1 (18) con nivel máximo de recursividad a 2	90
Figura 29. Individuo I_1 (18) totalmente desplegado	91
Figura 30. Individuo (20) a efectos de evolución	92
Figura 31. Evaluación del Individuo (20).....	93
Figura 32. Alcance máximo de un proyectil.....	94
Figura 33. Evolución de las pruebas mediante PG clásica	99
Figura 34. Evolución de las pruebas mediante PG clásica (detalle)	100
Figura 35. Evolución de las pruebas mediante PG con funciones definidas.....	101

Figura 36. Evolución de las pruebas mediante PG con funciones definidas (detalle)	102
Figura 37. Evolución del ajuste y ECM de la ejecución 6	102
Figura 38. Ejemplo de ruleta de probabilidades.....	104
Figura 39 Elección mediante el método ruleta.....	105
Figura 40 Evolución de las pruebas mediante PG con funciones definidas y probabilidad en los operadores	107
Figura 41. Importancia del uso de los coeficientes de seguridad.....	112
Figura 42. Mejores individuos PG Clásica y PG con Coeficiente de Seguridad	115
Figura 43. Correlación de los mejores individuos mediante PG Clásica y PG con coeficiente de seguridad.....	116
Figura 44 Evolución del ECM mediante PG con coeficiente de seguridad	116
Figura 45 Evolución del ajuste y ECM de la ejecución 2	117
Figura 46. Ejemplo de solución.....	120
Figura 47. Ejemplo de individuo válido.....	121
Figura 48 Evolución de las pruebas mediante PG con optimización de funciones conocidas	125
Figura 49 Evolución de las pruebas mediante PG con optimización de funciones conocidas (detalle).....	125
Figura 50. Flexión.....	128
Figura 51. Deformación por tensiones de corte.....	129

Índice de figuras

Figura 52. Tensiones en un elemento diferencial ideal.	129
Figura 53. Tensiones de corte principales.	129
Figura 54. Red de isostáticas de una viga.	130
Figura 55. Principales efectos del esfuerzo cortante.	131
Figura 56. Fuerzas internas en una viga de hormigón fisurada sin armadura a cortante.	131
Figura 57. Ejemplo de rotura por falta de armadura transversal.	131
Figura 58. Parámetros de la viga.	132
Figura 59. Distribución del conjunto de datos empleado.	133
Figura 60. Interacción momento cortante según formulación EC-2 , la MCFT.	136
Figura 61. Ajuste experimental de la formulación del EC-2 frente a las variables d y ρ_t	138
Figura 62. Resultados para entrenamiento y test de ejecuciones de PG sin parsimonia.	141
Figura 63. Correlación entre valor de ensayo y valor de predicción.	148
Figura 64. Puntos de demérito cometidos por las fórmulas.	149
Figura 65. Distintas fuerzas existentes entre la unión hormigón-acero con corrugas.	154
Figura 66. Fenómeno de adherencia.	154
Figura 67. Condiciones geométricas a) Posición de barras b) Recubrimiento.	156
Figura 68. Armadura de confinamiento.	156

Figura 69. Tipos de anclaje	162
Figura 70. Distribución del conjunto de datos empleado.....	169
Figura 71. Comparación de ecuaciones actuales.....	170
Figura 72. Correlación valor ensayos/valor predicción de las formulaciones actuales	171
Figura 73. Predicciones de las soluciones alcanzadas.	178
Figura 74. Predicciones de las ecuaciones PG9RSC4, PG8R1 y PG8v2R5 comparándolas con la FIB	179
Figura 75. Predicciones de las ecuaciones PG7v3F2, PG7v3F4, PGcc3 y PGcc6 comparándolas con la FIB.	180
Figura 76 Tensión de barra frente Is/db de la FIB, PG7v3F2 y PGcc6.....	181
Figura 77. Deformación por fluencia	183
Figura 78. Fluencia y recuperación de fluencia	185
Figura 79. Ejemplo de 4 series de fluencia	191
Figura 80. Distribución del conjunto de datos empleado.....	193
Figura 81. Operador " $\ln(t-t_0+1)$ " introducido	194
Figura 82. Diagrama del proceso de extracción de conocimiento	195
Figura 83. Capacidad de predicción de la fórmula F4, comparada con los modelos CEB y B3	198
Figura 84. Relación entre los datos experimentales y los proporcionados por los distintos modelos.....	199

ÍNDICE DE ECUACIONES

(1)	17
(2)	37
(3)	38
(4)	39
(5)	39
(6)	40
(7)	48
(8)	49
(9)	49
(10)	82
(11)	82
(12)	83
(13)	83
(14)	87
(15)	87
(16)	87
(17)	87

Índice de ecuaciones

(18).....	89
(19).....	92
(20).....	92
(21).....	94
(22).....	96
(23).....	96
(24).....	96
(25).....	98
(26).....	98
(27).....	99
(28).....	99
(29).....	99
(30).....	109
(31).....	110
(32).....	111
(33).....	113
(34).....	114
(35).....	114
(36).....	119

(37)	120
(38)	121
(39)	122
(40)	122
(41)	123
(42)	134
(43)	134
(44)	135
(45)	135
(46)	139
(47)	140
(48)	141
(49)	142
(50)	145
(51)	145
(52)	146
(53)	150
(54)	150
(55)	150

Índice de ecuaciones

(56).....	151
(57).....	151
(58).....	151
(59).....	151
(60).....	151
(61).....	151
(62).....	151
(63).....	151
(64).....	151
(65).....	151
(66).....	151
(67).....	152
(68).....	157
(69).....	157
(70).....	157
(71).....	157
(72).....	158
(73).....	158
(74).....	158

(75)	158
(76)	159
(77)	160
(78)	160
(79)	162
(80)	163
(81)	163
(82)	163
(83)	163
(84)	163
(85)	163
(86)	163
(87)	163
(88)	164
(89)	165
(90)	167
(91)	171
(92)	172
(93)	173

Índice de ecuaciones

(94).....	173
(95).....	173
(96).....	174
(97).....	176
(98).....	176
(99).....	176
(100).....	176
(101).....	176
(102).....	176
(103).....	176
(104).....	176
(105).....	176
(106).....	176
(107).....	176
(108).....	177
(109).....	177
(110).....	177
(111).....	182
(112).....	182

(113)	185
(114)	186
(115)	186
(116)	188
(117)	189
(118)	196
(119)	197
(120)	197
(121)	197

ÍNDICE DE TABLAS

Tabla 1. Prioridad de los operadores.....	85
Tabla 2 Nivel de despliegue para las F_1 (14), F_2 (15), F_3 (16) y F_4 (17)	90
Tabla 3. Conjunto de datos empleado.....	95
Tabla 4. Parámetros del ejemplo del proyectil.....	96
Tabla 5. Pruebas realizadas mediante PG Clásica.....	97
Tabla 6. PG con funciones definidas por el usuario.....	100
Tabla 7. Ejemplo de probabilidades de los operadores.....	104
Tabla 8. PG con funciones definidas y probabilidad en operadores	106
Tabla 9. Ejemplo de funcionamiento.....	109
Tabla 10. PG Clásica con el módulo de coeficientes de seguridad	113
Tabla 11. Puntos de demérito en PG Clásica (entrenamiento)	114
Tabla 12. Ejemplo de restricciones a ramas	122
Tabla 13. PG con optimización de funciones conocidas.....	124
Tabla 14. Conjunto datos empleadas	132
Tabla 15. Variables empleadas en el modelo del Eurocódigo 2	134
Tabla 16. Variables empleadas en el modelo ACI318-05	135
Tabla 17. Resultados comparativos entre diferentes formulaciones normativas.....	137

Índice de tablas

Tabla 18. Variabilidad de resultados en ensayos del mismo tipo.....	143
Tabla 19. Parámetros empleados	144
Tabla 20 Restricciones aplicadas	144
Tabla 21. Resultados entrenamiento.....	146
Tabla 22. Resultados Test	147
Tabla 23. Resultados global	148
Tabla 24. Variables empleadas en el modelo de la ACI para el cálculo de la longitud de anclaje	159
Tabla 25. Variables empleadas en el Eurocódigo 2 para el cálculo de tensión de adherencia	160
Tabla 26. Variables empleadas en el Eurocódigo 2 para el cálculo de la longitud de anclaje básica	160
Tabla 27. Variables empleadas en el Eurocódigo 2 para el cálculo de la longitud de anclaje	161
Tabla 28. Variables empleadas en la EHE para el cálculo de la longitud de anclaje	164
Tabla 29. Variables empleadas en la FIB.....	165
Tabla 30. Autores incluidos en el conjunto de datos empleado.....	167
Tabla 31. Conjunto datos empleado.....	168
Tabla 32. Resultados de las expresiones actuales.	172
Tabla 33. Parámetros empleados	175

Tabla 34. Restricciones aplicadas	175
Tabla 35. Resultados obtenidos para el total de ensayos	177
Tabla 36. Variables empleadas en el Modelo ACI 209.....	188
Tabla 37. Variables empleadas en el Modelo MC90	190
Tabla 38. Variables, rangos y normalización empleados en el desarrollo.....	192
Tabla 39. Resultados de los diferentes modelos (valores reales).....	197

I. Introducción

Uno de los problemas típicos ante los que se enfrenta un investigador cuando aborda el estudio de fenómenos reales es la ingente cantidad de información y de datos que es posible extraer para su análisis. En muchos casos, además, la información y los datos son difíciles de relacionar entre sí. Para descubrir, comprender y entender esa interrelación, una de las posibilidades a disposición de los expertos es el empleo de procedimientos de extracción de conocimiento mediante técnicas de Inteligencia Artificial (Graupe, 1997; Holland, 1975; Koza, 1992; Cios, Pedrycz, Swiniarski & Kurgan, 2007; Rivero, Rabuñal, Dorado & Pazos, 2005). Esencialmente se basa en el empleo de diversas herramientas o técnicas de análisis de datos con el objetivo de descubrir patrones y relaciones entre dichos datos que puedan ser empleados para realizar predicciones válidas. Existen distintas áreas de la ciencia en la que se persigue el desarrollo de modelos matemáticos a partir de la extracción de conocimiento de un fenómeno. Una de esas áreas, la Ingeniería Civil, ha sido

el lienzo de trabajo elegido donde se han centrado las pruebas de la metodología propuesta. Concretamente, todas las pruebas están enmarcadas dentro del ámbito de la Tecnología de la Construcción. Cabe destacar que tanto la aplicabilidad de la metodología como las mejoras en las técnicas aquí propuestas van más allá del ámbito de la Ingeniería Civil, ya que serían válidas en otros contextos similares donde se necesite la aplicación de técnicas inductivas sobre datos para la obtención de conocimiento en forma de modelos matemáticos. Concretamente, centrados en problemas de regresión simbólica sobre datos. En la actualidad, dentro de la IA existen diversas técnicas que sirven de apoyo en la realización de estas tareas. Quizás la técnica adaptativa más comúnmente empleada dentro de la IA son las Redes de Neuronas Artificiales (en adelante RR.NN.AA.) puesto que han demostrado su solvencia en numerosos ámbitos de aplicación, desde el mundo de la medicina al mundo ingenieril, siendo una herramienta versátil dadas las ventajas relativas a tolerancia a fallos, capacidad de abstracción, etc. En el caso concreto del presente estudio, esta técnica puede servir de ayuda para el análisis de comportamiento de un modelo, pudiendo obtener predicciones del fenómeno pero no explicándolo mediante ecuaciones matemáticas. Para este cometido la Computación Evolutiva (en adelante CE) si sería una alternativa válida. Ya que con una de las técnicas de CE, concretamente la Programación Genética (en adelante PG) permite extraer mediante programas o expresiones matemáticas el conocimiento inherente en los datos. Esto se hace posible gracias a que la PG es una técnica de búsqueda que permite la resolución de problemas por medio de la inducción automática de programas que describen una relación entre datos de entrada y salida.

Para la redacción de la presente tesis se ha establecido una organización a base de capítulos. Antes del primer capítulo, como se ha podido comprobar, se

encuentra la dedicatoria, los agradecimientos el resumen, y los índices de contenido, figuras, ecuaciones y tablas.

En el presente **Capítulo I** se muestra tanto la introducción a la problemática en la que se enmarca la tesis, como esta propia descripción del contenido de la misma.

En el **Capítulo II** se formula la hipótesis de trabajo y en función de esta hipótesis se dirigen los objetivos a la consecución de una metodología que permita corroborar dicha hipótesis.

Para la consecución de dichos objetivos, se necesita una base sólida de los conocimientos actuales, por ello en el **Capítulo III** se muestran los fundamentos aplicados tanto en el desarrollo de la metodología, como en la implementación de la metodología empleando como técnica inductiva la PG.

El **Capítulo IV** presenta un extracto de las técnicas actuales en las que se aplica, de una manera u otra, mecanismos que influyen en la orientación del proceso de búsqueda dentro del campo de la PG, así mismo se comentan diversos trabajos relacionados con IA dentro del ámbito de la Ingeniería Civil.

El **Capítulo V** es el encargado de mostrar, de forma pormenorizada, la metodología desarrollada, dando un especial énfasis en la incorporación del conocimiento del experto.

Ya en el **Capítulo VI** se muestra una posible implementación de la metodología. En este caso se añaden distintos módulos dentro de la PG que permiten orientar el proceso de búsqueda de las soluciones. Además de explicar de forma detallada cada módulo, se muestra un ejemplo de funcionamiento en un problema de comportamiento conocido consistente en

la predicción de la distancia máxima que alcanza un proyectil. Para poder ver las bondades de cada módulo se compara con la PG Clásica, estableciendo en todo momento la misma configuración para todas las pruebas.

En el **Capítulo VII** se muestra el conjunto de pruebas realizadas en entornos reales agrupados en tres supuestos enmarcados dentro del hormigón estructural. El primero en mostrar es la predicción del esfuerzo a cortante en vigas de hormigón sin armadura transversal. A continuación se aplica el sistema desarrollado en la predicción de la tensión máxima de adherencia alcanzada entre barras de acero y el hormigón. Para finalizar se muestra el problema de realizar la predicción de la fluencia del hormigón sometido a cargas constantes a lo largo del tiempo.

Las conclusiones alcanzadas tras la realización de esta tesis se muestran en el **Capítulo VIII**. En él se indican los logros y las principales dificultades ante las que se ha tenido que hacer frente. Y en el **Capítulo IX**, siguiendo con las exigencias de la actual reglamento de estudios de doctorado sobre la mención “Doctor europeo” (Capítulo VII - artículo 32) se traducen las conclusiones al inglés.

Siendo conscientes de la importancia en la continua mejora de las técnicas de extracción de conocimiento y en aras de mejorar la usabilidad se presenta en el **Capítulo X** los posibles caminos por los que se va a continuar investigando.

En último lugar, en el **Capítulo XI** se enumera la bibliografía referenciada en el texto, puesto que incluir todos los documentos consultados, como por ejemplo los artículos necesarios para completar los conjuntos de datos de ensayos experimentales, resultaría demasiado extenso.

II. Hipótesis de trabajo y Objetivos

En la mayoría de los casos en los que se desea modelar el comportamiento de un fenómeno, el experto, además de basarse en modelos teóricos que lo describen, necesita realizar experimentos que, o bien corroboran y afinan dichos modelos, o los refutan para luego crear nuevos modelos adaptados a la realidad. De esta situación surge la pregunta ¿es posible sustituir al experto para realizar esta tarea? La afirmación de esta pregunta sería un ambicioso objetivo, pero la realidad conduce a responderla negativamente. En la gran mayoría de los casos es necesaria la intervención del experto, ya sea con la incorporación de conocimiento a priori, la definición del modelo de estudio o en la validación de los resultados. Pero otra pregunta surge ¿es posible ayudar

al experto en el modelado de un fenómeno? Y si es posible ¿qué herramientas o técnicas se pueden emplear?

Dentro del área de Inteligencia Artificial existen un conjunto de técnicas que permiten ayudar al experto en la extracción del conocimiento de una infinidad de problemas complejos distribuidos en diversas facetas (clasificación, agrupamiento, regresión, etc.). Una de las principales técnicas es la denominada Redes de Neuronas Artificiales, estas técnicas poseen un potente sistema de aprendizaje que han llevado a su empleo en tareas de investigación clásica (McCulloch & Pitts, 1943; Orchard, 1993; Haykin, 1999) o en investigación aplicada a áreas concretas, como por ejemplo en problemas relacionados con la Ingeniería Civil (Rabuñal, 2008). Otra técnica que podría ser útil en el apoyo al experto sería la aplicación de Sistemas Expertos, especialización de los Sistemas Basados en Conocimiento, permitiendo la solución de un problema en el cual se emplea el conocimiento de un dominio de aplicación concreto. Por ejemplo Bauer et al han desarrollado el Dxpalin, un Sistema Experto que permite ayudar en el diagnóstico médico (Bauer et al, 2002). Otras técnicas que podrían servir de apoyo al trabajo del experto se engloban en la denominada Computación Evolutiva. Este tipo de técnicas permiten la obtención de resultados (optimización, selección de variables, regresión, etc.) mediante un enfoque distinto a las técnicas de programación convencional, debido principalmente a la característica de paralelismo en la búsqueda de soluciones.

El empleo del conjunto de técnicas enumeradas hasta ahora, tal como se conocen hoy en día, presentan diversas limitaciones para la mayoría de problemas. Por ejemplo en las RR.NN.AA. no se conoce el funcionamiento del modelo resultante, es decir, no son capaces de explicar su funcionamiento o los resultados que producen. Aunque hay aproximaciones que intentan

extraer el conocimiento de ellas (Rabuñal, Dorado, Pazos, Pereira & Rivero 2004). Por otra parte también presentan una gran complejidad en su desarrollo, es decir, la configuración de parámetros en la RNA. Si se toma como referencia los SE, las principales limitaciones están relacionadas con los datos. Esta técnica no funciona bien cuando los datos presentan inconsistencias o éstos no son estructurados. Por último, centrándose en la PG, el principal problema es la generación de soluciones demasiado complejas que aunque pueden ser fidedignas con el fenómeno de estudio, sería excesivamente dificultosa su interpretación.

Esta última técnica (PG) tiene la ventaja importante sobre las demás de que plasma, en forma de ecuación matemática, las relaciones entre los datos, permitiendo llegar a comprender la influencia de las distintas variables. Como se ha visto, los resultados pueden llegar a ser demasiado complejos pero ¿sería posible diseñar un sistema de extracción de conocimiento basado en PG que el usuario pueda dirigir?

En la presente tesis se desarrolla una metodología para la aplicación de técnicas heurísticas con la principal característica de que el proceso de búsqueda es guiado mediante las exigencias del usuario. De este modo, el experto en el área en donde se desea extraer conocimiento “dicta” las restricciones, mediante reglas de distinta índole, para que influyan en el proceso de búsqueda.

III. Fundamentos

En este capítulo se describen las bases tecnológicas que se utilizarán en el desarrollo de esta tesis, así como el ámbito de aplicación técnica de la misma. Para poder diseñar y realizar una metodología de extracción de conocimiento a través de sistemas evolutivos es necesario conocer las técnicas disponibles con un cierto grado de profundidad. Para ello, se expondrán los conceptos fundamentales de dichos sistemas. Con este fin se explicarán las técnicas más importantes de Computación Evolutiva. Dado que se usará la PG como técnica principal para el desarrollo del sistema informático, se realizará una concisa descripción de la misma. Así mismo, será necesario describir el funcionamiento de los métodos evolutivos y de los operadores genéticos. De igual manera se muestra cómo se pueden codificar los problemas en forma de individuos, es decir, la forma de convertir el problema en material genético procesable por la PG. También será necesario mostrar cómo actúa el proceso

de evolución natural en la búsqueda de soluciones. Por último se describirá la metodología de Identificación de Sistemas, en la que se inspira esta tesis.

1 Computación Evolutiva

El término Computación Evolutiva (CE) engloba un conjunto de técnicas que se basan en la realización de modelos que emulan ciertas características de la naturaleza, fundamentalmente la capacidad que tienen los seres vivos para adaptarse a su ambiente, lo que ya había sido tomado como base por Darwin para su teoría de la evolución según el principio de selección natural de las especies (Darwin, 1859). Esta teoría sostiene que aquellos individuos de una población que posean los caracteres más ventajosos dejarán proporcionalmente más descendencia en la siguiente generación; y si tales caracteres se deben a diferencias genéticas que pueden transmitirse a los descendientes, tenderá a cambiar la composición genética de la población, aumentando el número de individuos con dichas características. De esta forma, la población completa de seres vivos se adapta a las circunstancias variables de su entorno.

Se puede afirmar que John Holland ha sido el primero en adaptar las ideas de la evolución a la computación. Holland era consciente de la importancia de la selección natural, razón por la cual desarrolló una técnica que permitió incorporarla en un programa de ordenador. A la técnica que inventó Holland se le llamó originalmente *planes reproductivos*, pero se hizo popular bajo el nombre de Algoritmo Genético (en adelante AG) tras la publicación de su libro “Adaptation in Natural and Artificial Systems” en 1975. El objetivo que pretendía con sus desarrollos era que las máquinas lograsen aprender por sí mismas (Holland, 1975).

Un AG es un algoritmo de búsqueda basado en la reproducción sexual y el principio de supervivencia del más apto. Esto permite a las especies biológicas adaptarse a su ambiente y competir por los recursos. Los AA.GG. establecen una analogía entre el conjunto de soluciones de un problema y el conjunto de individuos de una población natural, codificando la información de cada solución en una cadena de valores (bits o números denominados *genes*), llamada *cromosoma*. Cada uno de estos cromosomas evolucionan a través de iteraciones, denominadas *generaciones*. En cada generación, los cromosomas son evaluados usando alguna medida de aptitud (evaluando los valores que actúan como incógnitas y asignando un valor que determina la capacidad de ese individuo para resolver el problema). En cada generación se emplean los operadores genéticos de selección, cruce y mutación para generar los nuevos cromosomas. El conjunto de cromosomas resultante de una generación se denomina *descendencia*.

Se denomina *reproducción* al proceso de formación de una nueva generación de nuevos individuos a partir de los ya existentes. Un factor muy importante en el funcionamiento de este algoritmo es la operación de selección de individuos para la reproducción. Para que el algoritmo funcione, los mejores individuos (aquellos cuyo ajuste o nivel de aptitud sea mejor) deberán reproducirse más veces que el resto. Con esta idea, se han desarrollado múltiples algoritmos de selección.

Todos los algoritmos de selección existentes se basan en el mismo principio: escoger individuos (ya sea de forma probabilística o determinística) dando más posibilidades de supervivencia a los mejores pero posibilitando también la reproducción, en mucha menor medida, de los peores. De no ser así, si solamente se escogiesen los mejores, el algoritmo convergería prematuramente haciendo que toda la población sea igual, lo que se traduce en un cese de la

exploración del espacio de estados posibles para explorar únicamente y de forma intensiva sólo una pequeña región.

Posteriormente a la creación de nuevos individuos, y con una probabilidad baja (comúnmente se suelen emplear valores entre 1% y 5%), cada individuo nuevo es sometido a un proceso de mutación. Un ejemplo de variación sería un cambio al azar de un bit de un individuo, en el caso de que los individuos se codifiquen como una cadena de bits. Finalmente, los nuevos individuos son insertados en la nueva generación formándose así una nueva población que constituye la siguiente generación.

Este proceso se repite hasta que se satisface algún criterio de parada establecido. Los criterios más habituales suelen ser de tres tipos. En el primero se detiene la evolución cuando un individuo ha alcanzado un nivel de ajuste, que es suficiente para dar la solución como válida. El segundo tipo de parada es un control mediante un número de generaciones máximo prefijado. Por último se puede utilizar como criterio de parada del algoritmo la convergencia de la población, criterio que se cumple cuando el 95% de la población es igual (con codificaciones binarias, o valores dentro de un rango especificado, con otro tipo de codificaciones). Cuando esto ocurre la media del ajuste, también denominada bondad de la población se aproxima al ajuste del mejor individuo.

Dado que cada individuo representa una posible solución al problema, la existencia de un gran número de individuos en la población implica que el algoritmo realiza una búsqueda en muchas regiones distintas del espacio de estados a la vez.

Una ventaja adicional de los AG es que la estrategia de resolución de problemas utiliza una medida de ajuste para dirigir la búsqueda, y no se requiere ningún conocimiento específico del espacio de búsqueda, siendo

eficientes incluso en espacios que presenten saltos, ruido, valles, etc. Como cada individuo dentro de la población dirige la búsqueda, el AG realiza una búsqueda en paralelo en numerosos puntos del espacio de estados con numerosas direcciones de búsqueda (Fuchs, 1999).

La implementación de la metodología en el sistema desarrollado utiliza la técnica de PG. Esta surge como una evolución de los AG tradicionales, conservando el mismo principio de selección natural. En este caso, la PG pretende proporcionar soluciones a problemas mediante la inducción de programas y algoritmos que los resuelvan. En el caso de un AG, el resultado que proporciona el algoritmo es el ajuste de una serie de valores numéricos codificados en forma de cromosoma que resuelven el problema. En el caso de la PG, el resultado que proporciona, según la técnica utilizada, puede ser un programa, otro algoritmo, o incluso una expresión matemática que resuelve el problema. La diferencia fundamental entre el AG y la PG es la forma de codificar las soluciones, mientras en un AG se realiza mediante una secuencia de valores, en la PG se representa mediante una estructura denominada *árbol*, en la que se codifica una fórmula matemática. Esta estructura así denominada árbol representa en los *nodos* los operadores (operaciones aritméticas, funciones trigonométricas, etc.) y en las *hojas* o terminales las constantes y variables.

Koza es quien, en el libro “Genetic Programming. On the Programming of Computers by means of Natural Selection”, sienta las bases de lo que a partir de ese momento se conoce como Programación Genética (Koza, 1992) y que originariamente se había implementado en lenguaje de programación LISP. Su técnica es casi independiente del dominio de aplicación y ha sido utilizada en numerosas aplicaciones como la compresión de imágenes, el diseño de

circuitos electrónicos, el reconocimiento de patrones, los movimientos de robots, etc.

Posteriormente este mismo autor (Koza, 1994), extendió su técnica mediante la incorporación de lo que se ha denominado *Automatically Defined Function* (ADF). Funciones que pueden ser reutilizadas como las subrutinas en un programa de ordenador y sirven para incrementar de forma notable la capacidad de la PG para generar programas automáticamente.

Como ya se ha comentado anteriormente, las soluciones se codifican mediante la representación en *árbol*. El árbol es un grafo conexo, acíclico y compuesto por nodos. En esta codificación se definen los siguientes términos:

- Nodo padre: dicese del nodo que tiene sucesores o hijos.
- Nodo hijo: es un nodo sucesor, es decir tiene un nodo padre.
- Nodo raíz es el padre de todos los nodos. En el caso especial de que el árbol esté constituido por un solo nodo también se denomina raíz.
- Nodo hoja: nodo sin hijos, también denominado nodo terminal.
- Nivel de un nodo: distancia desde la raíz, es decir la cantidad de nodos por los que hay que pasar para llegar al nodo seleccionado.
- Subárbol: se define por el nodo y todos los descendientes del nodo. Puede ser cualquier estructura conectada por debajo del nodo raíz.
- Altura o Profundidad de un árbol o subárbol: es el nivel de la hoja que tenga la distancia o camino más largo desde la raíz más 1.

Por ejemplo, dadas las definiciones anteriores, para la Figura 1 se obtiene que los nodos padre serían la potencia (^), la suma (+), la resta (-) y la división (/), los nodos hijos del nodo resta (-) serían la división (/) y la hoja "c". El nodo raíz sería la suma (+), los nodos hoja o terminales serían "a", "2", "b", "8",

“c”. El nivel del nodo “b” sería el 3. El subárbol del nodo división (/) está compuesto por los nodos división, “b” y “8”. La altura del árbol es de 4.

En el *árbol*, tras lo expuesto, se diferencian dos clases de nodos: la primera clase son los nodos **no terminales** donde se albergan los operadores del algoritmo que se quiere desarrollar (por ejemplo suma, resta, etc.). Se caracterizan porque siempre tienen uno o más hijos. La segunda clase son los nodos **terminales**, u hojas del árbol, donde se sitúan los valores constantes y las variables definidas previamente. Estos nodos no tienen hijos.

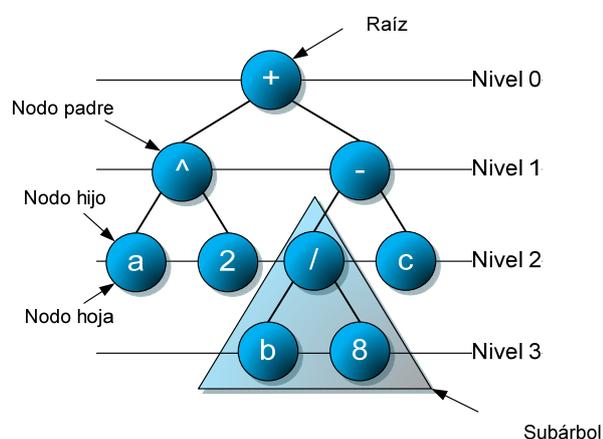


Figura 1. Ejemplo de árbol

Por ejemplo la Figura 2 representa una posible solución a un problema donde se desean relacionar las variables de entrada (a , b) con la salida $f(a,b)$, mediante la expresión $f(a,b) = a*((b/2)+5)$ en este ejemplo los nodos no terminales corresponden con el producto (*), la suma (+) y la división (/), mientras que los nodos terminales son los valores “2” y “5”, junto con las variables “a” y “b”.

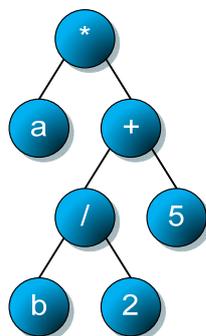


Figura 2. Árbol para la expresión $a*((b/2)+5)$

Una parte fundamental del funcionamiento de la PG es la especificación del conjunto de elementos terminales y no terminales antes del inicio del proceso evolutivo. Con los nodos que se le especifique, el algoritmo construirá los árboles. Por tanto, es necesario un mínimo proceso de análisis del problema para configurar el algoritmo, puesto que hay que decirle qué operadores o funciones (seno, coseno, exponencial, etc.) puede utilizar. Como regla general, es conveniente ajustar el número de operadores sólo a los necesarios, puesto que la adición de elementos no necesarios no provocará que no se encuentre la solución, pero sí que el algoritmo tarde más en encontrarla.

Dado que el proceso de construcción de árboles es un proceso basado en el azar, muchos de los árboles construidos no serán correctos, no por no seguir las reglas de la gramática, sino por la aplicación de operadores (nodos no terminales) a elementos que no están en su dominio. Por esta razón no se aplican estos operadores directamente, sino una modificación de ellos en la que se amplía su dominio de aplicación. El ejemplo más obvio es el operador de división, cuyo dominio es el conjunto de números reales excepto el valor cero. Ampliando su dominio, se define un nuevo operador ($\%$).

$$\%(a, b) = \begin{cases} 1, & b = 0 \\ \frac{a}{b}, & b \neq 0 \end{cases} \quad (1)$$

A esta nueva operación se denomina *operación de división protegida*, y en general cuando se crea una nueva operación que extiende el dominio de otra se denomina *operación protegida*. En la ecuación (1) se muestra el comportamiento de la operación de división protegida.

1.1 Restricciones en la PG

La técnica de PG, al igual que otras técnicas, necesita gran cantidad de recursos tanto computacionales como de memoria. Por este motivo es necesario delimitar y dimensionar el alcance máximo de los elementos. El término de restricción se define como cualquier limitación o reducción impuesta en la creación de nuevos individuos.

En la PG existen principalmente dos restricciones, la primera está relacionada con el tipado de datos y la segunda con la altura máxima que puede tener el árbol al generar una solución.

Las restricciones de tipado permiten establecer reglas sintácticas en la creación de árboles. (Montana, 1995), se establece el tipo de cada nodo (terminales y funciones), y para los no terminales (funciones) el tipo que debe tener cada hijo. De esta forma se especifica la estructura que deben seguir los árboles. Al especificar el tipo de cada nodo, se está especificando una gramática que va a ser la que siga el algoritmo para la construcción de árboles. Esta gramática permitirá que los árboles tengan la estructura deseada.

Los tipos de restricciones más usados son aquellos que tienen que ver con la realización de operaciones aritméticas sobre reales o enteros. Sin embargo, son también comunes otros como el tipo booleano.

Es necesaria también una restricción de altura que evita la creación de árboles demasiado grandes y fuerza una búsqueda de soluciones cuyo tamaño se acota de antemano. Con esta restricción se evita que los árboles posean mucho código redundante y se limita también su crecimiento excesivo (Soule & Foster 1997) (Soule, 1998). Este fenómeno de crecimiento excesivo de los árboles, conocido con el nombre de *bloat*, se produce de forma espontánea al avanzar el proceso evolutivo, debido a que los árboles evolucionan generando partes inútiles que no influyen en su comportamiento. Esto permite paliar los efectos nocivos de los operadores de cruce y mutación, puesto que cuantas más partes inútiles tenga un árbol, menos probabilidades habrá de ser modificado cuando se aplique un operador de cruce o mutación. De esta forma, los individuos se protegen a sí mismos. Por ejemplo, en la solución de un problema de regresión aparece el individuo representado por la Figura 3. Se puede observar que en esta solución aparece una rama inútil, ya que siempre va a estar multiplicado por cero ($y-y$).

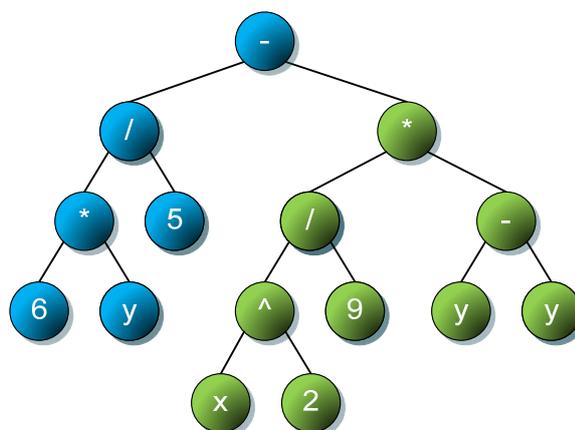


Figura 3. Ejemplo de árbol con rama inútil

1.2 Funcionamiento de la PG

El funcionamiento general de la PG se basa en la obtención de sucesivas generaciones a partir de las anteriores, es un mecanismo análogo al de los algoritmos genéticos. El esquema principal del algoritmo se puede ver en la Figura 4.

Tras la creación inicial de árboles, generalmente mediante técnicas aleatorias, se construyen sucesivas generaciones a partir de copias, cruces y mutaciones de los individuos de cada generación anterior.

Para poder aplicar la PG será necesario especificar varios parámetros, por un lado, el **Conjunto de terminales y funciones**, y por otro, la **función de ajuste** que indica la bondad de cada individuo.

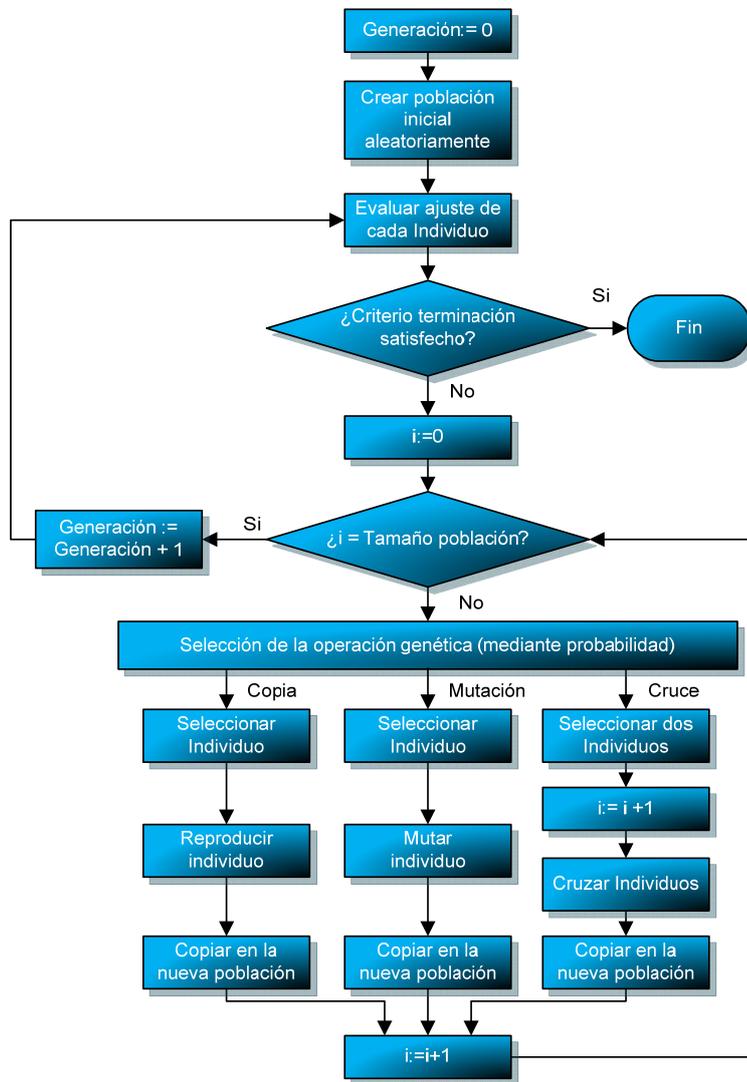


Figura 4. Diagrama de flujo de la Programación Genética

1.2.1 Generación inicial de árboles

El primer paso en el funcionamiento del algoritmo es la generación de la población inicial. En la creación de la generación 0, cada árbol se creará de forma aleatoria, pero en función del algoritmo se introducen restricciones en

los árboles. Dado que los árboles son aleatorios, los individuos de esta población, en general, representan soluciones malas al problema.

Para la creación de un árbol existe una gran variedad de algoritmos, pero son tres los más utilizados (Koza, 1992): parcial, completo e intermedio.

El algoritmo de creación parcial genera árboles limitados por la altura máxima. En cambio el algoritmo completo genera árboles cuyas hojas están todas a un determinado nivel, por tanto, árboles completos. Por último, el algoritmo de creación intermedio es una mezcla de los dos anteriores, con lo que permite mayor variedad en la población inicial, y con ello mayor diversidad genética. Este último algoritmo se basa en ejecuciones alternas de los dos anteriores en la que se toman distintas alturas para crear los elementos de la población. (Rivero, 2007)

Estos tres algoritmos descritos se basan fuertemente en el azar, y la única intervención del usuario está en la introducción de los elementos terminales y funciones. Sin embargo, existen otros algoritmos en los que la creación de árboles no es tan aleatoria. Por ejemplo, Luke asigna una probabilidad de aparición a cada nodo y, de esta forma, se reduce el carácter aleatorio de la creación. Adicionalmente orienta los árboles para que contengan más nodos de una clase que otra. Los algoritmos propuestos por este último autor serían muy similares, con la salvedad de que la elección de los elementos seguiría siendo aleatoria, pero estaría ponderada por esa probabilidad asignada (Luke, 2000).

1.2.2 Operadores genéticos

Como ya se ha comentado anteriormente la PG se basa en los operadores genéticos. Estos operadores son métodos que combinan y modifican los individuos para crear una nueva generación. Estos operadores son el resultado de adaptar a la PG los aplicados a los AG tradicionales. Esta modificación es necesaria debido a la forma que tiene la PG de codificar las soluciones en árbol. Los operadores genéticos son el cruce, la reproducción, la selección y la mutación. A continuación se detalla cada uno de estos operadores.

1.2.2.1 Cruce

Este operador genético es el más importante. En él, dos individuos de la antigua población se combinan para crear otros dos individuos nuevos.

Después de seleccionar a dos individuos como padres, se selecciona un nodo al azar en el primero y otro en el segundo de forma que su intercambio no viole ninguna de las restricciones: los nodos deben ser de igual tipo y los árboles nuevos deben seguir manteniendo la altura máxima. El cruce entre los dos padres se efectúa mediante el intercambio de los subárboles seleccionados en ambos padres.

La restricción de altura en la operación de cruce conlleva a que el resultado nunca supere el tamaño máximo permitido en la altura del árbol. Sea A la altura máxima permitida. Sean I_1 e I_2 dos árboles (individuos) con altura menor o igual. En el primer árbol I_1 se selecciona un nodo para el cruce. Este nodo está a una altura A_1 . Entonces para seleccionar el nodo donde se producirá el cruce en el segundo árbol I_2 habrá que descartar todos aquellos nodos de I_2

que estén en un nivel mayor que $A-A_i$. Esto es necesario puesto que si se selecciona un nodo con nivel mayor que $A-A_i$, el resultado del cruce daría lugar a un árbol (I_2) de altura mayor que A . De la misma forma, si el nodo seleccionado en el primer árbol está a un nivel N_i , se descartarán todos los nodos de I_2 cuya altura sea mayor que $A-N_i$, puesto que de insertar el subárbol seleccionado de I_2 en el nodo de I_1 , este último violaría la restricción de altura máxima.

La Figura 5 muestra un ejemplo de dos árboles seleccionados para la operación de cruce. En este ejemplo se ha limitado la altura máxima a 5 nodos.

En primer lugar se selecciona de forma aleatoria un nodo en el primer árbol. En este caso el nodo “/” (resaltado en color verde en la Figura 6), que representa a un subárbol de altura 3. En el segundo árbol se descartan aquellos nodos de tipo distinto. El único nodo que es de distinto tipo es el correspondiente al operador “<” (resaltado en color rojo).

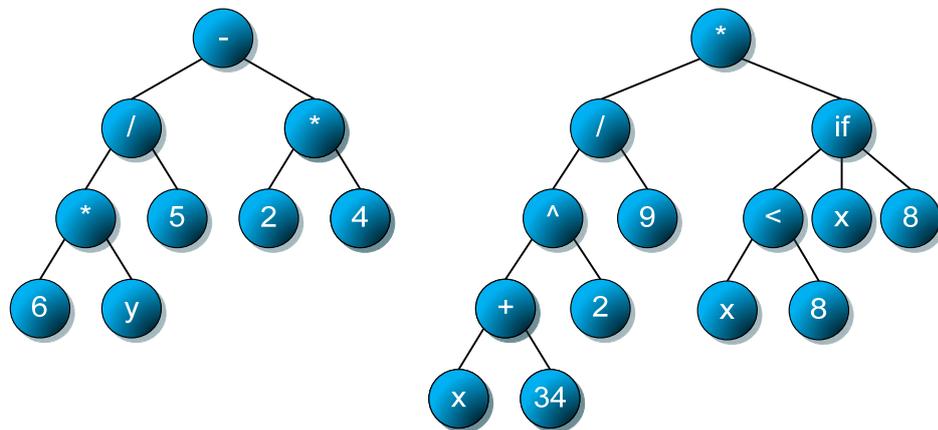


Figura 5. Árboles seleccionados para la operación de cruce

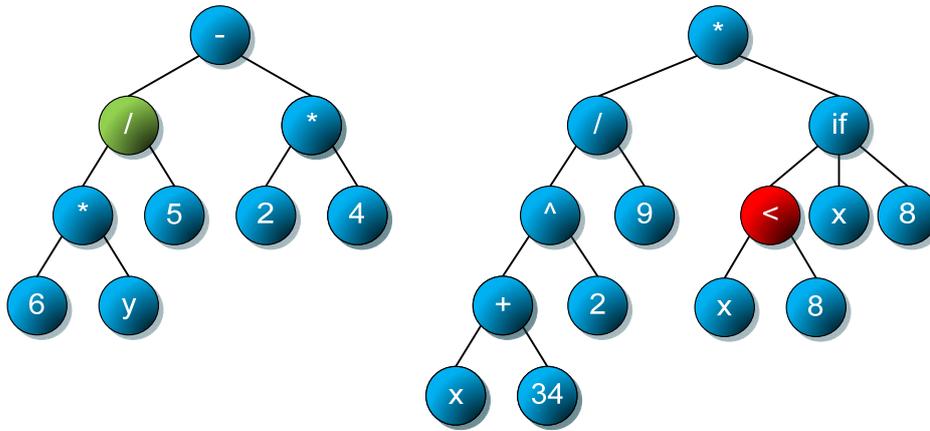


Figura 6. Selección de nodo en el 1^{er} progenitor y aplicación de la restricción de tipo en el segundo

Tras la selección el nodo del primer progenitor y el descarte de los nodos de distinto tipo en el segundo progenitor, se procede a descartar aquellos nodos del segundo progenitor que, en el supuesto de intercambiarse, violarían la restricción de altura de ambos. En el ejemplo de la Figura 6 es preciso descartar el nodo raíz, ya que, si se intercambia, el primer progenitor tendrá una altura de 6. De igual manera se deberán descartar los nodos con alturas de 4 y 5 correspondientes con los dos últimos niveles del árbol, puesto que de realizar el intercambio, el segundo progenitor tendría una altura demasiado alta. En la Figura 7 se muestran en color naranja los nodos descartados para el intercambio por esta restricción.

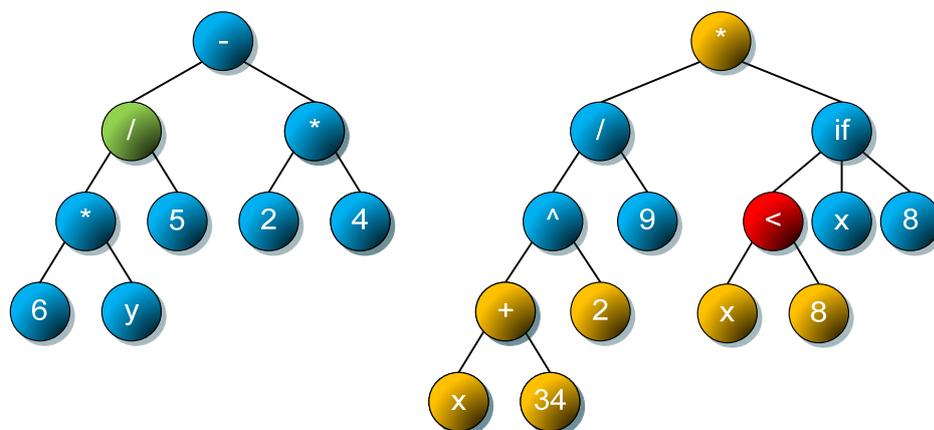


Figura 7. Aplicación de restricción de altura

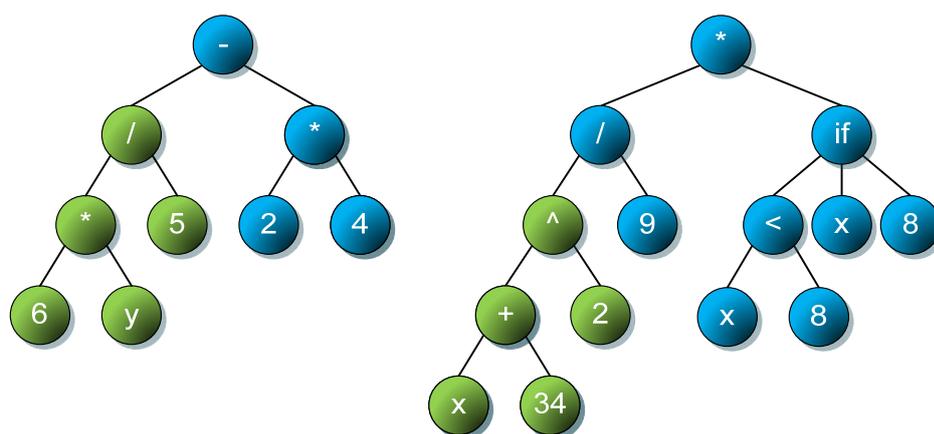


Figura 8. Selección de ramas a intercambiar

A continuación, de entre los nodos restantes del segundo progenitor se selecciona uno de forma aleatoria y se intercambian los subárboles a partir de los dos nodos seleccionados. En el ejemplo se ha seleccionado el nodo “^” (Figura 8). El resultado final de la operación de cruce se puede ver en la Figura 9. Finalmente se introducen los dos árboles creados en la nueva generación.

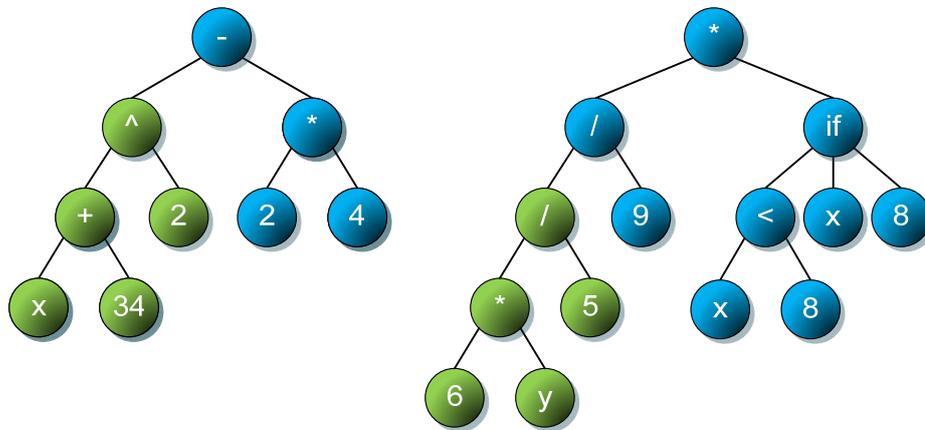


Figura 9. Resultado del cruce de dos individuos

Con la operación de cruce se generan dos individuos nuevos. Esta operación necesita dos individuos, por tanto se cataloga como operación sexual. Este operador es el auténtico motor del funcionamiento de la programación genética e implica la combinación de resolución de subproblemas para la resolución del problema principal. Cada subárbol de la PG puede interpretarse como una forma de resolver un sub-problema. Se ha observado (Soule, 1998) que la mayoría de los cruces provocan la generación de individuos peores, así como la aparición de mucho código redundante dentro de los árboles. Este código redundante previene los posibles efectos nocivos de otros operadores más destructivos como el de mutación, pero provoca un crecimiento exagerado de los árboles en poco tiempo. Para este problema, una posible solución es utilizar *cruces no destructivos*. Es decir, los árboles generados por la operación de cruce son insertados en la nueva generación si son mejores que sus padres. En caso contrario, se insertan en la nueva generación copias de los padres.

Una ventaja de este tipo de cruces sobre los de los algoritmos genéticos tradicionales es que al cruzar dos padres iguales, los hijos, en general, son

distintos a los padres y también distintos entre ellos. Esto no ocurre en el cruce de los algoritmos genéticos, en los que, cuando se cruzan padres idénticos, los hijos son iguales a los padres.

Por ejemplo, en la Figura 10 se muestran dos árboles iguales a los que se les realiza la operación de cruce. Los nodos de color verde son los seleccionados para realizar el intercambio. El resultado que se obtiene, como se puede ver en la Figura 11 son árboles diferentes.

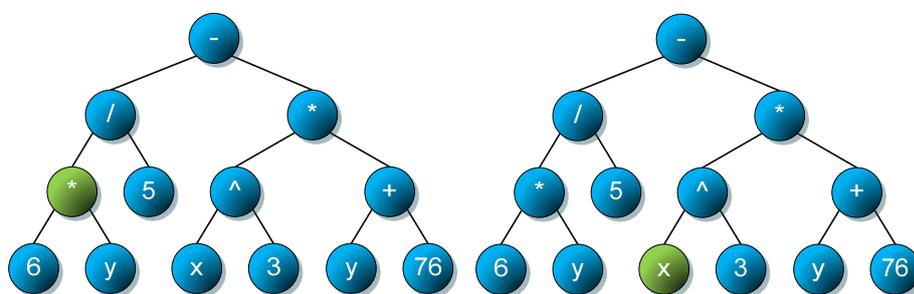


Figura 10. Cruce de dos árboles iguales

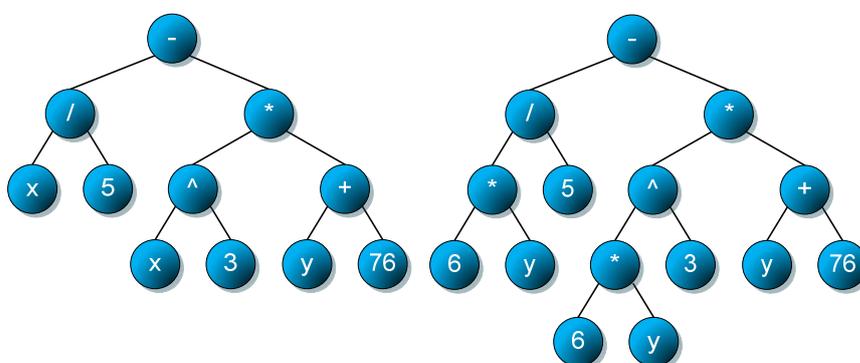


Figura 11. Resultado del cruce de dos árboles iguales

Al realizar un cruce, la forma de seleccionar los nodos de cada árbol se suele realizar generalmente asignando una probabilidad de que el nodo tomado sea no terminal. Esta probabilidad suele ser alta (sobre el 90 %), debido a que en

un árbol la mayoría de los nodos son terminales. Si no se toma esta precaución, la mayoría de los cruces no serían más que permutaciones de elementos terminales entre distintos árboles.

El operador de cruce aquí descrito se aplica de forma igual para todos los individuos. Sin embargo, existen variantes adaptativas de estos operadores (Angeline, 1996) en las que el propio algoritmo se modifica, así como numerosas variantes que tienen como base este algoritmo (Aguirre, Tanaka & Sugimura, 1999) (Pereira, Machado, Costa & Cardoso, 1999). De entre todos los operadores genéticos, el operador de cruce es el principal operador utilizado para la generación de los nuevos árboles y exploración del espacio de estados (Poli, 1996).

1.2.2.2 Reproducción

En ocasiones, tras realizar las operaciones de cruce o mutación, ningún individuo resultante produce un mejor ajuste que los mejores individuos de la generación actual. Por este motivo es interesante que una selección de individuos puedan pasar a la siguiente generación sin ninguna modificación. Para ello se define la operación de reproducción, que simplemente representa la copia de individuos en la nueva generación. En esta operación para generar un individuo solo es necesario un individuo de la generación anterior, por lo tanto la operación se podría clasificar como asexual.

Esta operación, junto con la de cruce, son las que se utilizan más frecuentemente en el proceso genético, y entre las dos tiene una especial predominancia la de cruce. De hecho, el porcentaje de individuos nuevos generados a partir de cruces suele ser superior al 90%, mientras que el resto son generados mediante copias. El aumento del número de individuos

generados por copias aumenta el peligro de predominancia de un individuo sobre el resto de la población, y esto conduce a que tras varias generaciones, toda la población converja hacia ese individuo. Esta es una situación indeseable, puesto que se pierde la diversidad genética inicial y conlleva que la búsqueda en el espacio de estados se focalice en una determinada zona, totalmente en contra de lo que se pretende.

1.2.2.3 Selección

Los algoritmos de selección serán los encargados de escoger qué individuos van a disponer de oportunidades de reproducirse y cuáles no. Puesto que se trata de imitar lo que ocurre en la naturaleza, se ha de otorgar un mayor número de oportunidades de reproducción a los individuos más aptos. Por lo tanto, la selección de un individuo estará relacionada con su valor de ajuste. No se debe, sin embargo, eliminar por completo las opciones de reproducción de los individuos menos aptos, pues en pocas generaciones la población se volvería homogénea (Cantú-Paz, 2000; Goldberg & Deb, 1991).

Para evitar una predominancia de un individuo en la reproducción, ya sea por cruce o copia, es necesario regular bien la selección de individuos antes de aplicarles estos operadores.

En líneas generales todos los algoritmos de selección se basan en la misma idea: que los individuos más aptos tengan más posibilidades de ser escogidos para reproducirse. Aunque debe hacerse sin eliminar por completo las posibilidades de los menos aptos, puesto que sino la población convergería en pocas generaciones.

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

Selección por Ruleta

El método de selección por Ruleta, también denominado como Selección de Montecarlo, ha sido propuesto por De Jong, siendo uno de los métodos más utilizados desde los orígenes de la Computación Evolutiva (Goldberg, 1989). El principio que se aplica es que a cada uno de los individuos de la población se le asigna una parte de una ruleta que es proporcional a su ajuste. Para seleccionar un individuo basta con generar un número aleatorio del intervalo $[0,1]$ y devolver el individuo situado en esa posición de la ruleta. Esta posición se suele obtener recorriendo los individuos de la población y acumulando sus proporciones de ruleta hasta que la suma exceda el valor obtenido. Al asignar posiciones de la ruleta proporcionalmente al ajuste, los mejores individuos recibirán una porción de la ruleta mayor que los peores. Es muy habitual ordenar la población en base al ajuste, por lo que las porciones más grandes suelen encontrarse al inicio de la ruleta.

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

Selección por Torneo

La premisa principal de este método es la selección en base a comparaciones directas entre individuos (Blickle & Thiele, 1995). Dentro de este método se pueden distinguir dos variantes, la primera, donde se eligen de forma determinística y, la segunda, en la cual se eligen de modo probabilístico.

En la versión determinística se selecciona al azar un número p de individuos, es habitual emplear dos individuos, ($p=2$). De entre el conjunto de individuos seleccionados se elige el más apto para introducirlo en la siguiente generación.

En el caso de la variante probabilística, únicamente se diferencia en la forma de seleccionar al individuo ganador del torneo. En vez de escoger siempre el mejor se genera un número aleatorio del intervalo $[0,1]$, en el cual, si es mayor que un parámetro p (fijado para todo el proceso evolutivo) se escoge el individuo más apto y en caso contrario el menos apto. Para el parámetro p se establecen de manera general valores en el intervalo $[0.5, 1]$.

Si se varía el número de individuos que participan en cada torneo se puede modificar la presión de selección. De hecho, cuando participan muchos individuos en cada torneo, la presión de selección es elevada y los peores individuos apenas tienen oportunidades de reproducción. Un caso particular es el *elitismo global*. Consiste en un torneo en el que participan todos los individuos de la población, con lo cual la selección se vuelve totalmente determinista. Cuando el tamaño del torneo es reducido, la presión de selección disminuye y los peores individuos tienen más oportunidades de ser seleccionados.

En síntesis, la elección de un determinado método de selección condicionará significativamente la estrategia de búsqueda empleada en la PG. Si se opta por un método con una alta presión de selección la búsqueda estará centrada en un entorno próximo a las mejores soluciones actuales. Por el contrario, optando por una presión de selección menor se deja el camino abierto para la exploración de nuevas regiones del espacio de búsqueda.

Las posibilidades de elección de algoritmos de selección no están restringidas a las descritas. Existen muchas otras posibilidades, unos buscan mejorar la eficiencia computacional, otros el número de veces que los mejores o peores individuos pueden ser seleccionados. Ejemplos de estos otros algoritmos son: muestreo determinístico, escalamiento sigma, selección por jerarquías, estado uniforme, sobrante estocástico, brecha generacional, etc. (Chakraborty, Deb & Chakraborty 1996).

1.2.2.4 Mutación

El operador de mutación permite la variación de un árbol de la población. Este operador suele usarse antes de introducir un individuo seleccionado en la nueva generación con una probabilidad muy baja, comúnmente menor del 10 %. Existen dos tipos habituales de mutación, la primera, consiste en la variación del valor de un solo nodo, sea terminal o no, y la segunda, consiste en variar una rama del árbol en su totalidad.

En la variación de un solo nodo, también denominada mutación puntual, se realiza según los 3 pasos siguientes:

1. Se selecciona un nodo al azar del árbol.

2. Se elige al azar un nodo del conjunto de terminales o no terminales. Este nodo debe pertenecer al mismo tipo que el seleccionado en el paso 1, es decir, poseer el mismo número de hijos, de forma que sus hijos sean del mismo tipo.
3. Se intercambia el nodo antiguo del árbol por el nuevo, manteniendo los mismos hijos que el antiguo.

Cuando se realiza sobre un elemento no terminal se está provocando que las soluciones se combinen de distinta forma, puesto que cada rama del árbol representa una solución a un sub-problema y el no terminal que las une representa la forma de combinar esas soluciones. Este tipo de mutación apenas se usa debido a que el cambio introducido genera un individuo muy parecido. La Figura 12 ilustra un ejemplo de mutación de este tipo, en la parte de la izquierda de la figura se muestra el individuo original marcando el nodo a mutar (nodo “+” en color verde) y a la derecha se muestra el resultado de mutar el operador “+” por el operador “*” que es compatible tanto en el tipo de nodo como en número de hijos.

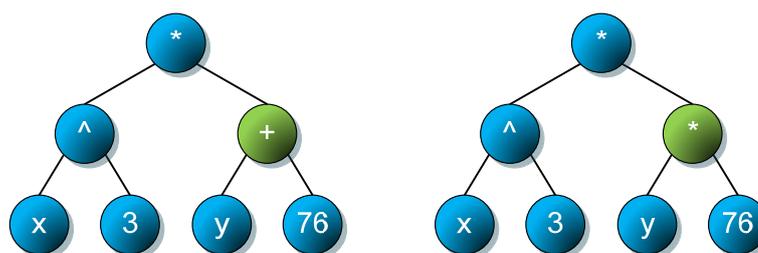


Figura 12. Mutación puntual

En el segundo tipo de mutación mencionado, consistente en variar la sub-rama en su totalidad, se efectúan cambios mayores en el árbol. Para ello se siguen las operaciones que a continuación se detallan:

1. Se selecciona un nodo al azar del árbol
2. Se elimina el subárbol definido por el nodo seleccionado, es decir, se elimina tanto los descendientes del nodo seleccionado como el propio nodo.
3. Se crea un nuevo subárbol del tipo y altura permitidos en el problema y se sitúa en el lugar del nodo eliminado.

De esta forma, al cambiar una rama entera, se consigue que a partir de la solución general se resuelva de otra manera una sub-parte del problema. En la Figura 13 puede verse un ejemplo de un árbol sobre el que se va a aplicar esta operación de mutación. En color verde se muestra el nodo seleccionado, que será el que sufra el proceso de mutación.

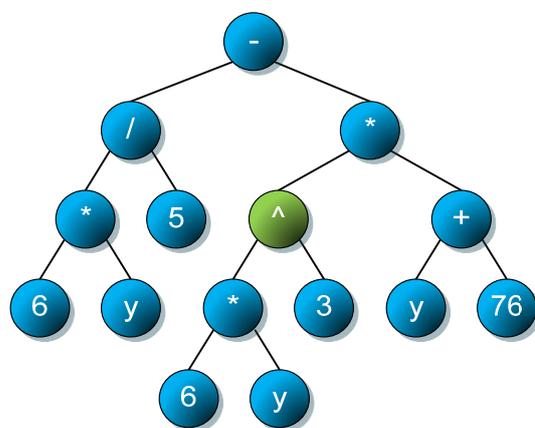


Figura 13. Árbol seleccionado para la mutación

Siguiendo el algoritmo de mutación, será necesario eliminar el nodo seleccionado, y el subárbol que representa, y generar un subárbol nuevo para

insertarlo en ese lugar. Para no violar la restricción de altura, el subárbol nuevo deberá tener una altura máxima de 3, en la Figura 14 se muestran el nuevo árbol generado (derecha) y el árbol preparado para realizar la inserción (izquierda). El último paso, será insertar el subárbol en el hueco dejado por el nodo eliminado. El árbol resultante puede verse en la Figura 15.

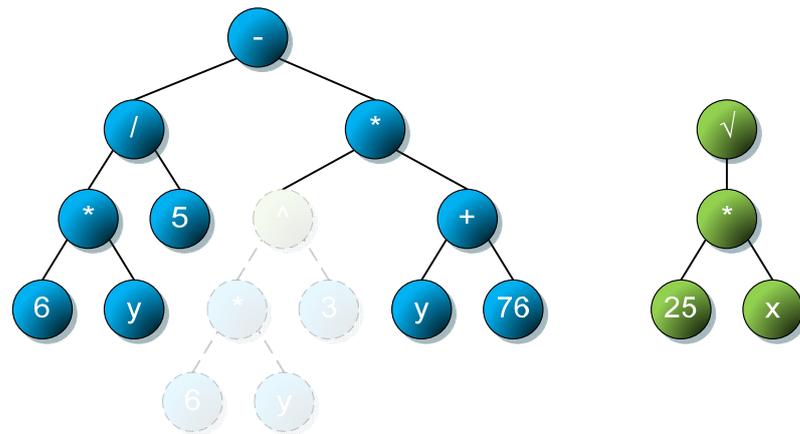


Figura 14. Árbol preparado para la mutación y subárbol generado

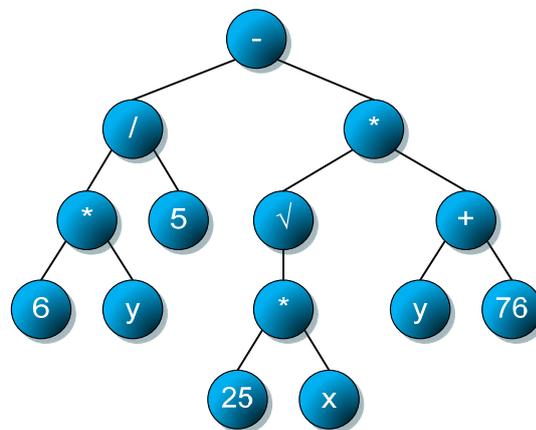


Figura 15. Resultado de aplicar la mutación

El operador de mutación provoca en ese individuo un salto en el espacio de estados, comenzando una búsqueda distinta en otra zona. La mayoría de las

mutaciones son destructivas, es decir, el individuo empeora, y por eso se utilizan con una probabilidad muy baja, para conseguir de este modo una cierta variedad genética. Existen estudios sobre la evolución sin el uso de cruces, en los que la mutación juega un papel fundamental (Chellapilla, 1997), en los que se utilizan distintos tipos de mutaciones, pero los resultados siguen siendo peores que utilizando cruces.

1.2.3 Evaluación

Al igual que en cualquier otra técnica englobada en el término Computación Evolutiva es necesario proporcionar un método que indique si los individuos de la población representan o no buenas soluciones al problema planteado. Este proceso se realiza en la función de evaluación, que permite establecer una medida numérica de la bondad de una solución. Esta medida recibe el nombre de ajuste o *fitness*. Como analogía con la naturaleza, el término de ajuste, o adecuación, de un individuo puede asemejarse a la probabilidad de que un individuo dado sobreviva hasta la edad de reproducción y se reproduzca. Esta probabilidad necesariamente debe estar ponderada con el número de descendientes.

Dentro de la Computación Evolutiva esta medición permite controlar la aplicación de los operadores genéticos, concretamente se suele utilizar para controlar el número de selecciones, cruces, copias y mutaciones.

El cálculo del ajuste o *fitness* se puede realizar de diferentes maneras. La aproximación más común consiste en crear explícitamente una medida de ajuste para cada individuo de la población. A cada uno de los individuos se les asigna un valor de ajuste escalar por medio de un procedimiento de evaluación bien definido que será específico del dominio del problema en el que se aplica

la PG. También puede calcularse el ajuste mediante un método “co-evolutivo”. Por ejemplo, el ajuste de una estrategia de juego puede ser determinado mediante la aplicación de la estrategia codificada por un individuo contra la población entera, o en su defecto una muestra, de estrategias de oposición.

Se diferencian cuatro tipos de ajuste o *fitness* habituales: el *fitness* puro, el *fitness* estandarizado, *fitness* ajustado y por último el *fitness* normalizado (Koza, 1992, 1994). En los siguientes apartados se describen brevemente cada uno de ellos.

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

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

En los problemas de maximización, como sería el de las hormigas mencionado anteriormente, los individuos con un *fitness* puro elevado serán los más interesantes. Al contrario, en los problemas de minimización interesarán los individuos que presenten un *fitness* puro más reducido. La ecuación (2) muestra cómo se calcula el ajuste puro. En esta ecuación, $s(i,j)$ representa el valor deseado, $c(i,j)$ el valor obtenido por el individuo y N_c el número de casos o conjunto de datos sobre el que se desea calcular el ajuste.

$$r(i,t) = \sum_{j=1}^{N_c} |s(i,j) - c(i,j)| \quad (2)$$

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

Para solucionar esta dualidad ante problemas de minimización o maximización se modifica el ajuste puro de acuerdo a la ecuación (3).

$$s(i,t) = \begin{cases} r(i,t) & \text{minimización} \\ r_{max} - r(i,t) & \text{maximización} \end{cases} \quad (3)$$

En el caso de problemas de minimización se emplea directamente la medida de *fitness* puro. Por el contrario, si el problema es de maximización se resta de una cota superior del error, r_{max} , el *fitness* puro.

Empleando esta métrica, la bondad de un individuo será mayor cuanto más cercano esté a cero el valor del ajuste. Por lo tanto, dentro de la generación t , un individuo i siempre será mejor que uno j si se verifica que $s(i,t) < s(j,t)$. Esta medida es muy útil en problemas en los que la cuantificación del nivel de ajuste de los individuos se basa en penalizaciones, como puede ser el error en inducción de fórmulas, error cuadrático medio, número de ejecuciones necesarias para encontrar la solución, etc.

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

El *fitness* ajustado es una variante del *fitness* estandarizado que se obtiene aplicando la transformación reflejada en la ecuación (4). De esta manera, el *fitness* ajustado tomará siempre valores del intervalo (0,1]. Cuando más se aproxime el *fitness* ajustado de un individuo a 1 mayor será su bondad.

$$a(i, t) = \frac{1}{1 + s(i, t)} \quad (4)$$

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

Los diferentes tipos de *fitness* vistos hasta ahora hacen referencia únicamente a la bondad del individuo en cuestión. El *fitness* normalizado introduce un nuevo aspecto, indica la bondad de una solución con respecto al resto de soluciones representadas en la población. La formulación de esta medida se puede ver, para una población de tamaño M , en la ecuación (5).

$$n(i, t) = \frac{a(i, t)}{\sum_{k=1}^M a(k, t)} \quad (5)$$

Al igual que el *fitness* ajustado, esta técnica de medida tomará valores dentro del intervalo $(0,1)$, con mejores individuos cuanto más próxima esté la medida a la unidad. A diferencia de antes, un valor cercano a 1 no sólo indica que ese individuo representa una buena solución al problema, sino que además es una solución destacadamente mejor que las proporcionadas por el resto de la población.

Tal y como puede derivarse de su definición, la suma de los valores de *fitness* normalizado de una población será siempre 1. Este tipo de ajuste es el más comúnmente empleado dentro de los métodos de selección proporcionales al *fitness*.

1.2.3.5 Función de ajuste en PG

Uno de los problemas más comunes de la PG es el crecimiento excesivo de los árboles. Debido principalmente a que los individuos se protegen a sí mismos de los efectos nocivos de los operadores genéticos de cruce y mutación. Para evitar este problema, es común emplear un factor de parsimonia en el cálculo del ajuste (Soule & Foster, 1997; Soule, 1998). Esta técnica se puede usar para reducir la complejidad del individuo que está siendo evaluado, y funciona mediante la penalización en el ajuste del individuo i de la siguiente forma:

$$fitness(i) = a(i, t) + \alpha * s_i \quad (6)$$

Donde $a(i, t)$ es una medida de la bondad del individuo, comúnmente cualquiera de las 4 funciones de *fitness* comentadas en los apartados anteriores. En este caso valores mayores representa una peor solución. α es el nivel de parsimonia y s_i es el tamaño (número de nodos) del individuo. Con este coeficiente se está penalizando el número de nodos de un árbol, y su valor máximo suele ser de 0.1. Con este valor, se necesitará que el árbol tenga 10 nodos para incrementar en una unidad el valor de ajuste. Sin embargo, un valor tan alto ya es muy dañino en la evolución, y se suelen tomar valores menores (0.05, 0.01, etc), en función del rango de valores en los que se espera que estén los ajustes de los individuos.

Utilizando el parámetro de parsimonia, a medida que evoluciona el algoritmo de PG se van seleccionando aquellas soluciones que ante un valor de ajuste similar son más sencillas, lo cual es un factor clave en la calidad del resultado

final que generará el sistema. Con esta funcionalidad, y seleccionando un valor adecuado para la parsimonia, se podrán obtener resultados “entendibles” y por lo tanto interpretables por el usuario.

De esta manera, el usuario podrá decidir si prefiere obtener un valor de ajuste máximo, anulando este parámetro (parsimonia=0), lo que suele conducir a una solución extremadamente compleja. Por otro lado, utilizando un valor de parsimonia adecuado, permite evolucionar el algoritmo y obtener soluciones simples, aunque con un peor valor de ajuste. En regla general se suele iniciar con un valor bajo y posteriormente, cuando se obtienen resultados aceptables, aumentar el valor de la parsimonia para favorecer la sencillez de las soluciones.

1.2.4 Parámetros de la PG

En cada método, algoritmo o procedimiento enmarcado en la PG aparecen distintos parámetros que configuran y matizan el proceso evolutivo de la PG. Estos parámetros permiten, en cierta medida, dirigir, controlar y limitar su funcionamiento. Por lo que, en cada supuesto, es necesario definir una configuración de parámetros a emplear. Siendo determinante la configuración empleada, puesto que la variación de un parámetro puede ser determinante para alcanzar una solución aceptable. Destacan los siguientes: tamaño de la población, porcentaje de cruces, porcentaje de mutaciones, criterio de parada del algoritmo, algoritmo de cruce, selección y mutación seleccionado, altura máxima permitida del árbol, altura máxima al crear los árboles en el proceso inicial, altura máxima de los subárboles creados por la mutación, probabilidad de selección del nodo no terminal y por último el valor de parsimonia.

El tamaño de la población indica el número de individuos que va a tener la población. En general este parámetro se ajusta de forma proporcional a la

complejidad del problema, tomando valores altos cuanto mayor sea ésta. De esta forma, cuanto más complicado es un problema, habrá más opciones de conseguir mejores resultados en un menor número de generaciones, puesto que se generan más individuos nuevos. Sin embargo, un tamaño alto puede no ser siempre la mejor solución: es posible tomar un tamaño menor y confiar más en la evolución durante mayor número de generaciones (Gathercole, Ross & Bridge 1997; Fuchs, 1999).

La tasa de cruces, también denominada porcentaje de cruces, representa la proporción de individuos descendientes, es decir de la siguiente generación, que serán creados a partir de cruces de individuos de la generación actual. Este valor suele ser alto, cercano al total de la población, y generalmente se sitúa sobre el 90% de los individuos. En cambio, la probabilidad de mutación se suele establecer con un valor muy bajo, por lo general en el intervalo [1%-5%].

La elección de los algoritmos de selección y mutación resultan fundamentales a la hora de lograr un buen resultado. Cabe destacar que en este caso, en la PG, se suele utilizar casi siempre el mismo algoritmo para realizar el cruce, pero se establece un parámetro que modifica su comportamiento, que es la probabilidad de selección de nodo no terminal. Este parámetro, se emplea para seleccionar los nodos del árbol que se emplearán para el cruce. De esta manera se pondera positivamente la probabilidad de escoger un nodo no terminal frente a uno terminal (Koza, 1992; Angeline, 1996). Después de la aplicación del algoritmo de mutación se suele emplear una aproximación elitista, en la que, en caso de no obtener ningún individuo con mejor ajuste, se conserva el mejor individuo de la generación anterior.

Durante el proceso de búsqueda es muy difícil, por no decir imposible, encontrar una solución que esté libre de errores, ya sea por los propios errores

de precisión de los datos de entrenamiento, o de la complejidad del problema a resolver, por ello es necesario establecer las condiciones de parada del proceso evolutivo. Lo más habitual es emplear alguno de estos tres criterios: alcanzar el número máximo de generaciones, conseguir un valor de ajuste determinado o que la población haya convergido.

Estos parámetros son comunes a cualquier técnica dentro de la Computación Evolutiva. A continuación, dada la distinta naturaleza de la forma de codificación que tiene la PG, se detallan una serie de parámetros específicos de la PG.

Dado que, las soluciones se codifican en forma de árbol, uno de los parámetros más relevantes es la altura, sobre lo que se establecen tres tipos de restricciones: altura máxima, altura inicial máxima y altura máxima de mutación. La altura máxima del árbol indica la altura máxima que van a poder alcanzar los árboles durante el proceso evolutivo. Este parámetro se ajusta también de forma proporcional a la complejidad del problema, de forma similar al tamaño de la población: un tamaño de población elevado exige elevar la altura máxima, pues, de no ser así existirá un gran número de individuos repetidos en la población, con predominio de uno de ellos, y los problemas que ello acarrea. Fundamentalmente, pérdida de variedad genética y convergencia prematura del algoritmo. Para ajustar los parámetros de altura y tamaño de la población, es necesario valorar la complejidad del problema a resolver. De hecho, para determinar estos dos parámetros es conveniente tener en cuenta el número de elementos que tienen los conjuntos de elementos terminales y no terminales, así como el número de variables y analizar el problema a solucionar. De igual manera se debe establecer el tamaño máximo que podrán alcanzar durante la inicialización del proceso evolutivo. Por último la altura máxima que tendrán los subárboles creados por

una mutación de subárbol. Para una de estas mutaciones, el subárbol generado tendrá una altura máxima que será la mínima entre este parámetro y la altura permitida por la restricción de altura máxima en el nodo seleccionado.

Por último cabe señalar el factor de parsimonia, descrito en detalle anteriormente, que permite penalizar linealmente las soluciones más complejas (mayor número de nodos) obligando al proceso evolutivo a buscar árboles sencillos que permitan resolver el problema en cuestión.

1.3 Limitaciones de la técnica

Al igual que cualquier técnica, la PG no es ajena a diversos problemas o debilidades que en ocasiones hacen inviable su aplicación. En primer lugar se presenta la gran complejidad de las soluciones. Este problema puede venir determinado por muchos factores. Uno de ellos, por ejemplo, es que al experto no se le permite tener control en la altura de los árboles durante el proceso de su creación, simplemente puede definir una altura máxima. Otro de los problemas asociados a éste es el crecimiento excesivo de ramas inútiles en el árbol denominado *bloat*, comentado anteriormente, al que se le puede paliar sus efectos mediante el factor de parsimonia. Pero en muchas ocasiones los mecanismos existentes no son suficientes para obtener buenos resultados. Otro problema que se plantea es la falta de coherencia dimensional en el resultado obtenido, realmente la PG clásica combina los operadores sin tener en cuenta lo que representan, consiguiendo en la mayoría de los casos soluciones incoherentes.

Una de las principales virtudes de la PG, la aleatoriedad en la creación de los árboles, puede convertirse en perjudicial en el caso de un espacio de búsqueda muy extenso. Esto hace difícil obtener resultados satisfactorios. Si fuese

posible paliar el efecto de la aleatoriedad, por ejemplo, obligando al proceso de creación de árboles en una dirección concreta, se conseguiría reducir enormemente el espacio de búsqueda, con lo que se aumentan considerablemente las probabilidades de éxito. Por último y enlazado con lo anterior, la PG clásica presenta limitaciones a la hora de incorporar el conocimiento del experto en la solución de un problema, principalmente se limita su contribución a la elección de los operadores a emplear.

2 Identificación de Sistemas

Existen numerosos aspectos de la ciencia en la que se necesita abstraer el modelo de funcionamiento de un fenómeno a partir de los datos observados. Bien es cierto que debido a la problemática tan extensa y diversa existen numerosas aproximaciones diferentes para su implementación (Ljung, 2010). Para este cometido existen diversas aproximaciones metodológicas que son diferentes en función del área de conocimiento en las que están focalizadas. Dentro del campo de sistemas de control estas técnicas de extracción de modelos se conocen bajo el termino Identificación de Sistemas (en adelante IS).

2.1 Definición

En sentido global la Identificación de Sistemas se puede definir cómo el conjunto de metodologías para la obtención de modelos matemáticos de sistemas dinámicos a partir de mediciones realizadas sobre el sistema.



Figura 16. Abstracción de un sistema

Con el término sistema se engloba cualquier proceso que transforma un conjunto de datos de entrada en un conjunto de datos de salida (Figura 16). Por ejemplo, si se considera que en un sistema intervienen dos variables de entrada x e y , y produce una salida z habrá una relación de transformación $z = f(x,y)$ que relacione las dos variables de entrada con la salida. En este caso las técnicas de IS intentaran descubrir cómo es esa relación, empleando únicamente los datos de entrada y los datos de salida de la forma que se describe en la Figura 17.

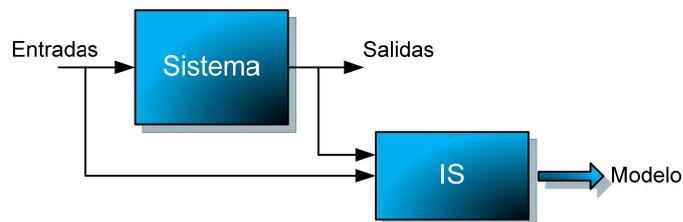


Figura 17. Extracción del modelo mediante IS

2.2 Conceptos básicos.

El núcleo de IS está compuesto por conceptos de naturaleza estadística relacionados con la información, el aprendizaje y la validación o generalización. A continuación se presentan los conceptos básicos de esta metodología.

Modelo: Representa la relación entre los elementos observables. Es decir, un modelo permite realizar predicciones del comportamiento o de las propiedades de un objeto. Típicamente la relación es una expresión matemática, en ocasiones también puede ser una tabla o un grafo. Se denota por el símbolo m .

Modelo Real: A pesar de que en la mayoría de los casos no es realista alcanzar el modelo real de funcionamiento de un objeto, es necesario asumir una abstracción de ese modelo. Ya que en la mayoría de los casos el modelo sería demasiado complejo. Esta abstracción se denota por S

Tipos de Modelos: Representa al conjunto de modelos. Cabe destacar que el conjunto puede ser definido mediante un número finito de parámetros, o no.

Complejidad: Es una medida de “tamaño” de un tipo de modelo. Se denota por el símbolo C

Información: Con este término se engloba tanto la información procedente de los datos observados como de información conocida previamente sobre el objeto a modelar.

Estimación: Es el proceso de “seleccionar” un modelo guiado a través de la información. Los datos empleados para realizar dicha selección se denominan Datos de Estimación o también es comúnmente aceptado *datos de entrenamiento*. Este conjunto se denota por Z_e^N siendo N el tamaño del conjunto de datos. En la actualidad este proceso de estimación se denomina Aprendizaje.

Validación, también denominado generalización: Este proceso permite garantizar que el modelo generado es útil no sólo para los datos de entrenamiento, si no que es válido para otros conjuntos de datos relacionados.

En este sentido se construye un conjunto de datos denominado *datos de validación*, denotado por Z_v .

Ajuste del Modelo: Se sintetiza en un número que mide la bondad de un determinado modelo m . Es decir, permite cuantificar cómo de bueno es un modelo partiendo de un determinado conjunto de datos Z . Se denota de la siguiente manera: $F(m, Z)$.

2.2.1 Estimación

Todos los conjuntos de datos contienen tanto información útil como información irrelevante. Con el objetivo de que la información irrelevante no influya en el proceso se debe prevenir de alguna manera, por ejemplo si se emplea la máxima general de “La naturaleza es simple” el proceso de estimación sería la expuesta en la ecuación (7) donde F es la medida de ajuste elegida y $h(C(m), N)$ es la penalización basada en la complejidad del modelo o del conjunto de modelos en función del número de datos.

$$\hat{m} = \min_{m \in M} [F(m, Z_e^N) + h(C(m), N)] \quad (7)$$

Es decir, el modelo se va construyendo teniendo en cuenta:

- El ajuste del modelo a los datos
- La complejidad, ya que el modelo resultante no debería ser demasiado complejo.

Comúnmente estos dos aspectos son contradictorios, es decir, se intenta buscar un modelo con buen ajuste y poca complejidad. Es por ello que se debe llegar a un compromiso entre ambos aspectos.

Aunque en lo mostrado en el método descrito en la ecuación (7) se muestra un ajuste paramétrico de datos, también puede ser empleado como un modelo no paramétrico.

2.2.2 Ajuste del modelo

Por lo general no es difícil encontrar un modelo que describa la relación entre los datos de entrenamiento. Lo que lo convierte en un buen modelo es si al someterlo a un nuevo conjunto de datos (datos de validación) se obtienen unas buenas predicciones. Es de norma general, que si los conjuntos de datos están bien elegidos, el ajuste en la validación siempre es peor que en el proceso de aprendizaje.

2.2.3 Errores

Si se asume que existe el modelo real S , conceptualmente se puede determinar el error de un modelo m . Para calcular el error se emplea la ecuación (8)

$$S - \hat{m} \tag{8}$$

Los métodos más empleados para calcular el error de un modelo S , son el Error Medio (EM) y el Error Cuadrático Medio (ECM) o variaciones de estos. Si se toma como ejemplo el ECM, este vendría determinado por la ecuación (9). El ECM se divide entre el error de desvío (Bias) identificado por B y el error debido a la varianza denotado por V .

$$ECM = E(S - \hat{m})^2 = (S - m^*)^2 + (\hat{m} - m^*)^2 = B + V \tag{9}$$

El error de desvío se produce por los errores sistemáticos causados principalmente por las características de la señal de entrada y el tipo de modelo empleado. El segundo tipo de error, la varianza, representa los errores aleatorios introducidos por la presencia de ruido en los datos que impiden que el modelo reproduzca exactamente la salida. Uno de los factores principales que afectan a ambos tipos de errores es la cantidad de parámetros del modelo.

Entre los dos tipos de errores existe la siguiente relación: si en un modelo m se incrementa la complejidad C se consigue que disminuya el error de desvío B . En cambio el error debido a la varianza se incrementa con C .

2.3 Metodología de Identificación de Sistemas

Esta metodología (Ljung, 1999) consiste en una serie de etapas bien definidas mediante las cuales se realiza la tarea de modelado y descripción matemática de un sistema dinámico desconocido. En la Figura 18 se muestra de modo esquemático las etapas que se realizan en un proceso de identificación.

A continuación se muestran en detalle, cada una de las etapas en las que consiste el proceso de identificación de un sistema:

1. **Conocimiento previo y diseño de experimentos.** Consiste en la selección de las entradas con las que se excitará al sistema o modelo (S). Esta selección debe de ser lo suficientemente completa como para excitar todos los modos del sistema. Este hecho se denomina persistencia de excitación. Cabe resaltar que sólo se pueden identificar aquellos modos que son observables desde las salidas y que son suficientemente excitados desde las entradas.

2. **Obtención de datos y pre-procesamiento.** En este proceso se extraen los datos de los experimentos diseñados en la etapa anterior. Para ello se establecen las variables a medir, la frecuencia de muestreo, etc.

El conjunto de datos experimentales Z se divide típicamente en dos subconjuntos: entrenamiento Z_e y de validación Z_r . El subconjunto Z_e debe ser lo suficientemente representativo del sistema.

3. **Selección de la estructura de modelo.** Consiste en seleccionar un conjunto de modelos candidatos dentro del cual se seleccionará un modelo apropiado basándose en los datos disponibles de entrada-salida. Es decir, se seleccionará el tipo de modelo que mejor se ajusta, en algún sentido, a los datos. Pueden distinguirse tres enfoques diferentes para la selección de la estructura de modelo:

- a. Estructura en **“Caja Negra”**: los parámetros del modelo no tienen una interpretación física. Un modelo basado en leyes fundamentales es muy complicado o se desconoce los parámetros y no tienen una interpretación física.
- b. Estructura en **“Caja Gris”**: algunas partes del sistema son modeladas basándose en principios fundamentales y otras como una caja negra. Algunos de los parámetros del modelo pueden tener una interpretación física.

- c. Estructura en **“Caja Blanca”**: la estructura de modelo se obtiene a partir de leyes fundamentales. Los parámetros tienen una interpretación física
4. **Estimación del Modelo**: En este proceso se determina un modelo particular (m) perteneciente a la estructura de modelos, que es el que mejor se ajusta a los datos, es decir el que mejor se ajuste al funcionamiento del sistema. Esto implica normalmente:
 - a. Selección de un criterio a minimizar. Típicamente se emplea una función de ajuste basada en los errores de la predicción del modelo.
 - b. Determinación del vector de parámetros que minimiza el criterio. Es decir la estimación de los parámetros. Para afrontar esta tarea se puede elegir entre un método analítico (sólo posible en algunos casos) o entre métodos numéricos iterativos, también llamados métodos de optimización. Estos métodos pueden ser no lineales y no convexos dando problemas de convergencia y mínimos locales.
5. **Validación del modelo**: A partir del conjunto de datos de validación Z_v se obtiene una medida de confiabilidad del modelo estimado (m). Con este proceso es posible comprobar el funcionamiento del modelo, ya que con los resultados obtenidos se podrá decidir si el modelo es lo suficientemente apropiado. Esto determina la capacidad de predicción del modelo. Además se podrá determinar la distancia que separan el modelo real (S) frente al modelo estimado (m). Esto determina la medida de

incertidumbre del modelo. Por último, el proceso de validación permite determinar si el modelo y los datos son consistentes con las hipótesis sobre la estructura del modelo. Llegados a este punto, si el modelo generado (m) cumple con las expectativas, el proceso terminaría, en caso contrario, puede ser necesario volver a la etapa 1 de la metodología.

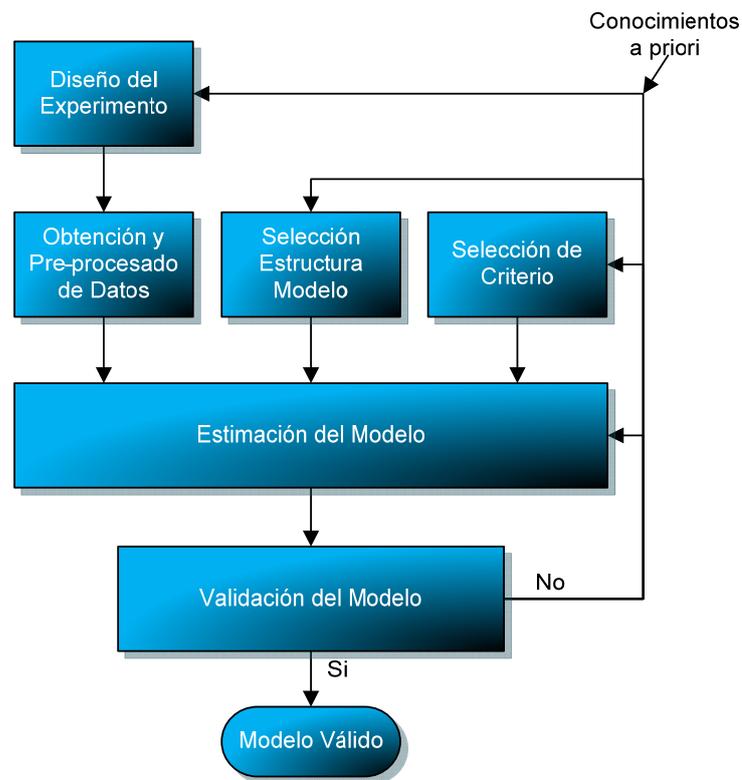


Figura 18. Etapas de un proceso de identificación (Ljung, 1999)

IV. Estado de la cuestión

Esencialmente, en esta tesis se persigue un sistema capaz de obtener soluciones a un problema de una forma, en mayor o menor medida, supervisada, dirigiendo el proceso de búsqueda mediante las indicaciones del experto. En la literatura científica existen aproximaciones a lo que aquí se presenta, pero dichos trabajos resuelven parcialmente las capacidades aquí expuestas y a menudo no solucionan una problemática concreta, sino que presentan la técnica pero no la aplican en ningún escenario real (Oliver & Rodríguez, 2004 ; Zhang & Rockett, 2007).

En cuanto a las tendencias en el campo de la PG relacionados con la orientación del proceso de búsqueda, por lo general se resumen en restricciones sintácticas, por ejemplo Koza emplea este tipo de restricciones a

la hora de generar nuevos individuos (Koza, 1992). Del trabajo original de Koza surgieron distintas propuestas para la resolución de problemas, por ejemplo Angeline y Pollack desarrollaron las estructuras modulares (subrutinas), materializadas en la librería GLIB. En este caso Angeline y Pollack modificaron el genotipo (la estructura de los individuos) sustituyendo una rama determinada de la solución por una función concreta. En cambio no modificaron la función de evaluación (Angeline & Pollack, 1994). Aplicando un procedimiento similar al de la librería GLIB, Koza en 1994 desarrolló un mecanismo denominado *Automatically Defined Functions* (ADF) (Koza, 1994) en el que se podría ver como un caso particular de restricciones sintácticas. Las ADF son funciones o subrutinas “reutilizables” por la PG de estructura fija que pueden ir evolucionando. Otra contribución importante en este ámbito es la que introduce Rosca y Ballard mediante el empleo de técnicas heurísticas para elegir las ramas más útiles. (Rosca & Ballard, 1996). Centrándose en el campo de la regresión simbólica, una posible alternativa de resolución de problemas es el empleo de diferentes tipos de polinomios. Por ejemplo Nikolay y Hitoshi propone una técnica para la construcción rápida de polinomios mediante PG, la cual consiste en la expansión de partes de un polinomio, basándose en la versión discreta de la serie de Volterra. A esta técnica se la denomina *Gabor-Kolmogorov polynomials*, mediante un mecanismo de transferencia de polinomios. Para ello, tanto los coeficientes de los polinomios parciales, como los del nuevo término se calculan mediante el algoritmo RLS (Recursive-Least-Squares) (Nikolaev & Iba, 2001). Otra aproximación en el campo de la regresión simbólica, la aportan Oliver y Rodríguez en la que dividen el problema de regresión en sub-problemas, constituyendo cada uno de ellos una rama que lo va resolviendo individualmente. El resultado final se obtiene mediante la combinación de todos los sub-problemas (ramas). La

técnica empleada, al igual que el ejemplo anterior, se basa en el empleo de polinomios (Oliver & Rodríguez, 2004).

Otro tipo de restricciones habituales son las que afectan al tipo de datos, o la coherencia dimensional de un resultado. En este grupo destaca el caso de Montana (Montana, 1995) que propone un método de PG fuertemente tipado denominado *strongly typed GP (STGP)*, en el que todas las variables, constantes, argumentos de las funciones, resultados de las funciones tienen prefijado un tipo válido de datos. Para ello se necesita que la inicialización y los operadores genéticos sean redefinidos teniendo en cuenta el tipo de datos de cada nodo para poder generar un individuo permitido. Con este método se consigue por ejemplo que el operador “producto” sólo se aplique a variables que sean reales o enteras y no en variables booleanas.

Por último se podría citar el conjunto de técnicas basadas en “*grammar-guided genetic programming*” (GGGP), en este caso las operaciones genéticas están condicionadas por una gramática definida por el usuario. Esta gramática, denominada “*context-free grammars*” (CFG), contiene el conocimiento del experto en el área de estudio. Por ejemplo, y dentro del grupo en cuestión, Hoai y McKay desarrollaron un entorno de trabajo en el que emplean gramáticas para la creación de nuevos individuos (Hoai & McKay, 2001). Un ejemplo aplicado a un problema concreto sería el presentado por García-Arnau et al (García, Manrique, Ríos & Rodríguez, 2007) quienes desarrollan un método denominado “*grammar-based initialization method – GBIM*” que combinado con GGGP resulta útil en tareas de clasificación, como el aplicado para cáncer de mama.

Otra forma posible de abordar la orientación del proceso de búsqueda sería el proceso de creación de los individuos (tanto en la creación inicial, como en la

creación de subárbol). De esta manera se reduce la aleatoriedad del proceso. Para lograrlo una de las alternativas más efectivas es la asignación de probabilidades al conjunto de operadores no terminales. Por ejemplo, Luke propone dos algoritmos que permiten asociar probabilidades a los elementos no terminales durante la creación de nuevos individuos (Luke, 2000).

Lo expuesto anteriormente muestra de forma simplificada un extracto de técnicas que permiten conducir, con mayor o menor flexibilidad, la búsqueda de un modelo. Para ello relacionan, de forma más o menos compleja, los datos de entrada con datos de salida. Centrándose en el ámbito de aplicación de los tres ejemplos presentados en esta tesis, la Ingeniería Civil, se constata que una gran mayoría de las aplicaciones de los métodos de optimización se focalizan en optimizar los recursos empleados. Por ejemplo, en el diseño y construcción de un elemento estructural (viga), a efectos económicos, interesa emplear la mínima cantidad de acero, hormigón, etc. pero garantizando, en todo momento, la seguridad estructural de la viga. Una técnica conocida y empleada para las tareas de optimización es el empleo de Algoritmos Genéticos (Kicinger, Arciszewski & De Jong, 2005). Por ejemplo, el trabajo realizado por Perera R. y Vique J (Perera & Vique, 2009) emplean los AG para el diseño óptimo de vigas de hormigón armado mediante el modelo de bielas y tirantes. En este caso buscan minimizar el producto de la fuerza axial, la longitud y la deformación axial de los elementos de la armadura. En cuanto al empleo de PG en este ámbito se podría citar también el trabajo desarrollado por Sonebi y Cevik, en el que emplean la PG clásica para encontrar una fórmula para modelar el comportamiento tanto en fresco como una vez fraguado del hormigón autocompactante que contiene el aditivo de ceniza volante (Sonebi & Cevik, 2009). Otro ejemplo sería el trabajo realizado por Ashour et al., en este caso se obtiene mediante PG una expresión que, a partir de las variables

previamente normalizadas, es capaz de predecir la resistencia a cortante en vigas de hormigón armado (Ashour, Álvarez & Toropov, 2003). Cabe citar también los trabajos aplicados realizados por Rabuñal, agrupados en su tesis doctoral titulada “Uso de técnicas de Inteligencia Artificial en Ingeniería Civil”, en la cual se ofrece una herramienta de manejo sencillo para los ingenieros civiles, que permite la aplicación de técnicas combinadas entre PG, AG y RR.NN.AA para la predicción de diversos fenómenos, en áreas tan dispersas como la hidrología o la tecnología de la construcción (Rabuñal, 2008). Un empleo más especializado en este ámbito son los trabajos de Ralte et al donde emplea la técnicas de GGPP para la creación de un modelo de comportamiento de un material a partir de datos experimentales (Ratle & Sebag, 2001).

Como resumen, cabe resaltar que en todos estos trabajos, relacionados en el ámbito de la Ingeniería Civil, principalmente aplican las técnicas clásicas de IA, sin que aporten una mejora en la misma. Solo en algunas ocasiones algunas aproximaciones aportan cierta mejora. Por ejemplo en la técnica de PG lo emplean combinado con el factor de parsimonia, o la incorporación de penalizaciones por exceso o por defecto dentro de la función de ajuste en la PG.

V. Metodología

El objetivo que se persigue en esta tesis consiste en encontrar un método de trabajo que, a partir de unos datos experimentales, permita obtener un modelo del comportamiento de los datos lo suficientemente sencillo para que pueda ser interpretable por el experto y manteniendo la premisa subjetiva de que la solución sea la aceptable y útil para el experto.

Para realizar este método de trabajo se empleará la metodología de Identificación de Sistemas comentada en los fundamentos pero adaptada a las necesidades específicas de este trabajo.

1 Antecedentes

Esta tesis se centra en el desarrollo de una metodología para la extracción del conocimiento existente en los datos derivados de ensayos de laboratorio.

Comúnmente, para intentar conocer más un fenómeno físico, se realiza una campaña de ensayos con distintas configuraciones de parámetros. Por ejemplo, para determinar la adherencia de las barras de acero en el hormigón se realizan un conjunto de ensayos en los cuales se varían la cantidad y tipo de hormigón implicado, la longitud del acero introducido, etc. Con el resultado de este conjunto de ensayos se permite corroborar, corregir, mejorar o incluso crear un modelo que describe el comportamiento del fenómeno físico involucrado en la adherencia acero-hormigón. Es decir, el experto, estudiando el conjunto de configuraciones de ensayos (plasmados en variables de entrada) junto con los resultados (variable o conjunto de variables de salida) desarrolla un modelo en el que aúna tanto los resultados teóricos como experimentales, consiguiendo de esta manera un modelo ajustado al mundo real. En la Figura 19 se muestra de manera esquemática el procedimiento clásico para la construcción de un modelo físico apoyado en resultados experimentales.

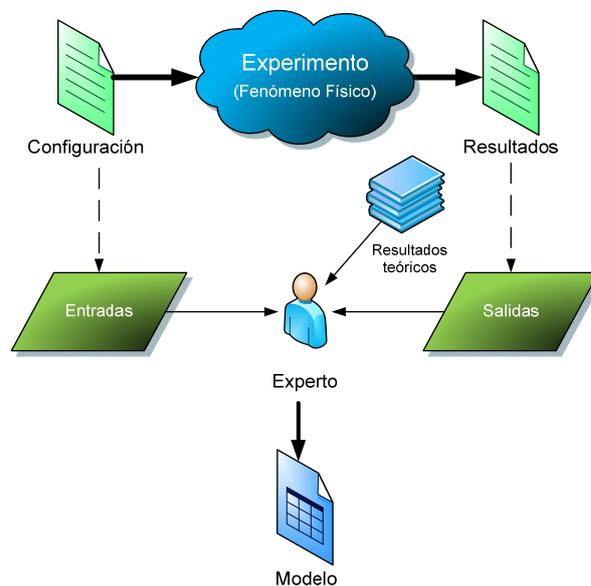


Figura 19. Desarrollo de un modelo físico

Con el objetivo de ayudar al experto en esta tarea, se plantea la posibilidad del empleo de técnicas de Inteligencia Artificial. En este contexto es habitual el empleo de las redes de neuronas artificiales en las que, con los datos experimentales, combinados o no con datos sintéticos de un modelo previo, permite desarrollar un sistema con capacidad de predicción. Las RR.NN.AA han demostrado su solvencia en este tipo de problemas obteniendo muy buenos resultados. El mayor inconveniente de esta técnica es su propia naturaleza de “caja negra”. De todas formas son útiles como “laboratorios virtuales” ya que captan muy bien el comportamiento de los datos y han servido para mejorar modelos físicos (Cladera & Mari, 2004; Iglesias, Rabuñal, Losada, Pachón & Carballo 2008).

Teniendo en mente el objetivo principal de la tesis, es decir, la orientación del proceso de búsqueda en la extracción de conocimiento, se necesita un algoritmo que permita relacionar las entradas y salidas de una forma comprensible por el experto. Para ello se plantea la opción de emplear un algoritmo inductivo, ejemplos de ello serían árboles de decisión, fuzzy logic, PG, expresiones genéticas, etc. En las cuales, dados los datos de entrada y salida, se obtiene de alguna manera, las reglas de funcionamiento del modelo. A priori, la mejor técnica, dada su capacidad de materialización del conocimiento en ecuaciones matemáticas, es el empleo de la técnica de PG. Aunque son conocidos los problemas que afectan a esta técnica como la complejidad de soluciones, la falta de coherencia dimensional, las dificultades en la adición de conocimiento del experto, etc. Todas estas cuestiones hacen necesaria la inclusión de un mecanismo, en forma de Sistema Experto, que condicione y dirija la búsqueda de soluciones en el sentido que el experto desee. En la Figura 20 se muestra un esquema de lo que se persigue en la presente tesis.

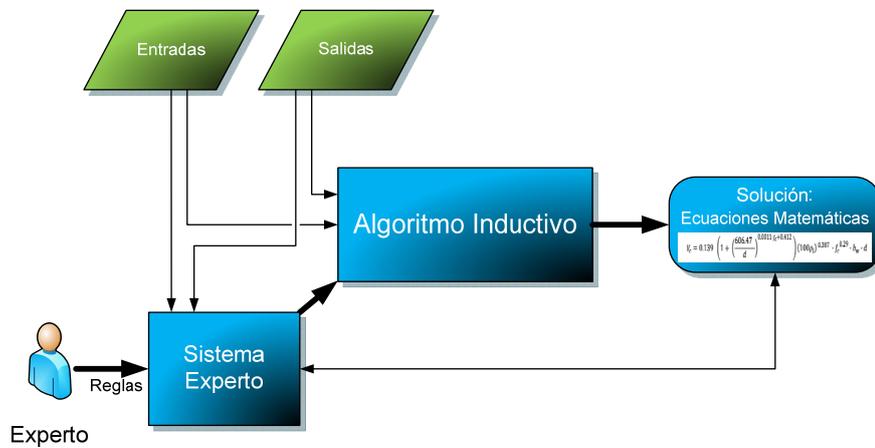


Figura 20. Modelo de funcionamiento deseado

A continuación se detalla pormenorizadamente la metodología que se ha desarrollado y que permitirá estudiar fenómenos diferentes. Estos fenómenos deben de tener en común la existencia de los suficientes datos experimentales para poder extraer la información de dichos fenómenos.

2 Metodología

En el proceso de extracción de conocimiento sobre cualquier tipo de datos, es necesario seguir una determinada pauta, de esta manera se conseguirá desarrollar un modelo de trabajo que, de forma fácil, permite la reproducción del modelo en distintos ámbitos de estudio. A continuación se presenta dicho modelo de funcionamiento plasmado en una metodología de trabajo. Esta metodología sigue las pautas principales de la metodología de Identificación de Sistemas pero adaptándola al objetivo principal de crear modelos a partir de datos experimentales orientando su búsqueda mediante la incorporación del

conocimiento del experto. A continuación se muestran sus cuatro fases principales.

2.1 Diseño del experimento y conocimiento previo

Esta es la primera etapa y una de las más importantes, pues en función de la “buena calidad” de los datos seleccionados para el entrenamiento y las pruebas, los resultados finales serán mejores. En primer lugar es necesario realizar una depuración de los datos, a primera vista puede parecer trivial, pero es fundamental para la obtención de resultados coherentes y no sesgados. El tener una base de datos bien verificada y lo más completa posible será una base sólida para el proceso. Obviamente, los resultados obtenidos estarán restringidos a la representatividad del conjunto de datos proporcionado, además suele ser habitual que en los intervalos en los que ésta sea incompleta, la calidad de los resultados obtenidos disminuya. En caso de no cumplir esta condición se debe realizar un refinamiento, eliminando un intervalo o desarrollando ensayos, hasta alcanzar un conjunto lo suficientemente representativo. En función de la cantidad de variables disponibles en el conjunto de datos puede ser interesante emplear una técnica de selección de variables denominada FSS (Feature Subset Selection) (Yang & Honavar, 1998). La necesidad de métodos de selección de variables surge motivada por la no monotonía de los modelos en relación con el número de variables predictivas, así como por la existencia de ciertas variables predictivas que pueden llegar a ser irrelevantes o incluso redundantes (Gestal, 2009)

Una vez obtenido un conjunto de datos representativo y coherente generalmente se divide en dos sub-conjuntos. El primero, de entrenamiento,

como su propio nombre indica servirá en el proceso de extracción del modelo. Y el segundo, de test, servirá para comprobar si los resultados obtenidos generalizan correctamente. Para realizar este proceso, se puede optar por varios métodos, principalmente se subdividen en métodos aleatorios o determinísticos. El método más empleado es de tipo aleatorio, consiste en dividir el conjunto de datos mediante un porcentaje sobre el total. Por lo general se emplea del 70 al 90 % de patrones para el entrenamiento y el restante (30-10 %) para test. Este método tiene el inconveniente principal de que los conjuntos resultantes son muy diferentes (rango de variables...), para evitar esto se suele recurrir al método cross-validation de “n” niveles que realizan el proceso anterior “n” veces y obteniendo así “n” pares (entrenamiento-test) y el resultado del ajuste sería la media de todos (Cantú-Paz & Kamath 2005; Dietterich, 1998). Por último, se puede optar por realizar el proceso de división manualmente, siendo el experto el encargado de realizar los dos conjuntos. Esta última opción es más compleja y recomendada puesto que garantiza que los conjuntos sean más representativos.

Esta etapa, junto con la sub-etapa “Desarrollo de meta-reglas” que se comentará en el siguiente apartado, es una agrupación de las etapas 1, 2 y 3 de la metodología de IS. Como se puede comprobar la etapa 3 de IS es el análisis y tratamiento de los datos, agrupándolos en estructuras de “caja negra”, “caja blanca” y “caja gris”. Sin embargo, en el caso que nos atañe, la estructura idealmente buscaría una solución de “caja blanca” ya que se pretende crear un modelo a partir de parámetros que tienen interpretación física. De todos modos, dado que es imposible conocer, a priori, todos los parámetros que intervienen en un fenómeno, con carácter general se optará por una solución “caja gris” o “caja pseudo-blanca” ya que con los parámetros disponibles es posible relacionar y explicar el funcionamiento del fenómeno.

2.1.1 Desarrollo de meta-reglas

El desarrollo de las meta-reglas requiere varias fases. En primer lugar se debe recabar toda la información posible acerca de los tipos de datos. Es decir, si son continuos o discretos, si las entradas son lógicas o valores en punto flotante, etc. Dependiendo de la cantidad de información que se disponga se podrá ampliar o reducir el espacio de búsqueda del algoritmo de extracción. Así, por ejemplo, si las entradas y las salidas son lógicas se evita el uso de operaciones matemáticas y trigonométricas complejas en el entrenamiento, redundando en una considerable reducción del espacio de búsqueda, y por ende del tiempo de extracción del conocimiento. De igual manera es en esta etapa donde el experto, a su juicio y basándose en el conocimiento previo del fenómeno de estudio, donde establecerá las reglas que deberán cumplir las soluciones. Con el término de reglas se engloba cualquier información que se puede emplear para dirigir el proceso de búsqueda. Un ejemplo sería el empleo de restricciones, es decir, cómo evitar el uso de una determinada variable, forzar el empleo de constantes, etc. Otros ejemplos de reglas podrían ser la aplicación de coeficientes de seguridad (en el caso de tratarse con problemas donde el error cometido por exceso no importe de la misma manera como si el error es por escasez). También es posible la incorporación del conocimiento de soluciones parciales con el objetivo de contribuir en la solución del problema. De igual manera se podrán añadir como meta-reglas un conjunto de hipótesis, en este caso, el algoritmo inductivo corroborará si las suposiciones ayudan a la solución en el caso de que sean elegidas, o las refutará en el caso de no servirle de ayuda.

Esta sub-etapa de la metodología es crucial para la orientación del proceso de búsqueda. Por si sola representa la obtención del conocimiento, materializadas en diferentes tipos de reglas, por parte del experto.

2.2 Selección del modelo o algoritmo de extracción

La segunda etapa de la metodología consiste en la elección del modelo o algoritmo de extracción de conocimiento. Partiendo de las premisas, características y meta-reglas presentadas anteriormente se debe elegir y configurar el modelo para la extracción del conocimiento. El funcionamiento que el modelo debe cumplir, se ha enunciado anteriormente en apartado “Antecedentes”, concretamente en la Figura 20. Fundamentalmente se centra en inducir reglas a partir de patrones entrada/salida obtenidos del modelo físico real, es decir los datos provenientes del conjunto de experimentos reales.

Todos los métodos de extracción de reglas existentes basan su funcionamiento en el empleo de patrones entrada-salida utilizando distintos métodos. Por tanto, lo que en definitiva se desea buscar es la extracción del conocimiento asociado a esos datos.

El tipo de conocimiento, codificado en meta-reglas, es determinante para la elección del modelo o algoritmo de extracción. Ya que deberá permitir la incorporación del conocimiento obtenido del experto en el algoritmo inductivo. Esto es, si se conoce algún tipo de comportamiento, como por ejemplo reglas que pueden intervenir en el proceso, restricciones en los tipos de datos a procesar, etc. se pueden incorporar en forma de meta-reglas al algoritmo inductivo. Antes de esto, en la etapa anterior, se ha debido

identificar y seleccionar el conocimiento del experto. De esta manera se consigue reducir el espacio de búsqueda del algoritmo inductivo. Así, por ejemplo, si se sabe que el modelo físico real se restringe a valores de entradas booleanas se puede evitar, en el algoritmo inductivo, el uso de simbología que trate valores continuos, con la consiguiente reducción del espacio de búsqueda que el algoritmo necesita explorar.

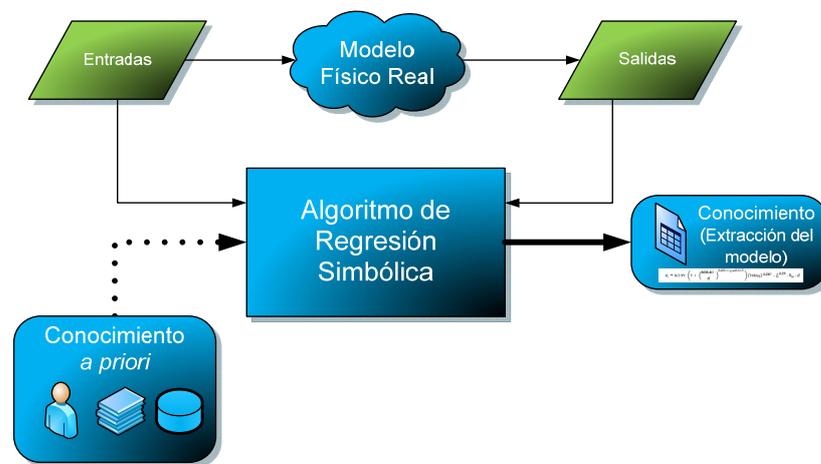


Figura 21. Diagrama del proceso de extracción de conocimiento

Se propone, por tanto, un proceso de extracción de reglas donde el núcleo principal sea el algoritmo inductivo que está influenciado por el conocimiento “a priori”. En el diagrama de la Figura 21 se puede ver que el núcleo del sistema de extracción de reglas radica en el algoritmo inductivo, concretamente se selecciona un algoritmo que permita realizar regresión simbólica sobre datos. Este algoritmo debe ser de carácter general, debe permitir el uso de simbología, debe funcionar por extracción de conocimiento de ejemplos, debe proporcionar un conjunto de reglas con alto nivel expresivo y debe poder incorporar conocimiento “a priori”.

El hecho de trabajar sobre un conjunto de ejemplos hace pensar, por similitud, que éste algoritmo se asemeja a la técnica de “Data Mining” o “Knowledge Discovery in Databases” (KDD)(Kantardzic, 2003). El KDD es un proceso iterativo en el que se van produciendo progresivos descubrimientos y cuantificaciones de relaciones predictivas en los datos y se permite transformar la información disponible en conocimiento útil. Podría ser posible aplicar alguna de las técnicas existentes en el KDD como posibles algoritmos para la extracción de meta-reglas.

2.3 Estimación de parámetros

Una vez desarrollado e implementado el algoritmo inductivo, la siguiente etapa consiste en la determinación y cuantificación de los parámetros que intervienen en dicho algoritmo. Para ello es necesario establecer mecanismos, siempre que sea posible, de obtención de los valores numéricos de estos parámetros. Esta estimación de valores numéricos se puede realizar de diferentes formas, en función del tipo de parámetros y de la estructura del algoritmo. En primer lugar se debe establecer un criterio a minimizar, es decir, la minimización del error cometido por el modelo en la predicción. Por ejemplo el EM, ECM o cualquier otro método que permite cuantificar la bondad del método. Una vez elegido el criterio a minimizar se deberá establecer cada uno de los parámetros, para ello se puede emplear alguna técnica numérica iterativa. Según el algoritmo utilizado y los parámetros involucrados se pueden emplear métodos matemáticos que obtengan los mejores valores para los parámetros. Este conjunto de métodos tiene el inconveniente de que se provoquen problemas de convergencia y de mínimos locales. Otra forma sería mediante un método analítico, dependiendo del caso de estudio, se podrán estimar los valores numéricos de los parámetros

mediante experimentación sobre casos reales. Así, cuando los métodos numéricos son impracticables o no producen resultados satisfactorios se puede recurrir al diseño de casos de prueba para la estimación de los mejores parámetros.

De igual manera que se realiza un refinamiento de los parámetros propios del algoritmo desarrollado, hay que tener en cuenta que el conocimiento del experto expresado en meta-reglas puede estar sesgado, incierto o incluso ser incorrecto en determinadas situaciones. Además el propio experto puede formular conjeturas, hipótesis o pronósticos en los que estime parcialmente el funcionamiento del proceso. Este hecho debe de ser tenido en cuenta a la hora de establecer el conjunto de meta-reglas o los conjuntos posibles de meta-reglas con los que se dirigirá el proceso de búsqueda.

2.4 Validación del modelo

Es la última etapa de la metodología, en esta fase se aplica el modelo extraído (conocimiento extraído) al subconjunto de patrones entrada/salida reservado para test. De esta manera se puede comprobar si los resultados que producen estas reglas se corresponden con los resultados que produce el modelo físico real. Una vez obtenidos los resultados de esta fase, se puede determinar si el modelo es lo suficiente bueno para la extracción de reglas del modelo físico real. De igual manera se podrá calcular cuantitativamente la desviación del modelo frente al modelo físico real. Por último, se podrá concluir si el modelo y los datos son consistentes con las hipótesis sobre la estructura del modelo.

Si llegados a este punto, los resultados obtenidos tras la aplicación de la metodología al completo, no son los esperados habrá que descartar o reestructurar el algoritmo inductivo volviendo a la etapa 2 denominada

“Selección del modelo o algoritmo de extracción”. Incluso puede ser necesario plantearse la validez y completitud del conjunto de datos. De detectarse inconsistencias, los conjuntos de datos deberán ser rechazados, siendo necesario revisar el método para la creación de estos ficheros e incluso modificarlos para incluir o eliminar valores que se consideren necesarios o incorrectos, respectivamente. En este caso habrá que volver a la etapa inicial denominada “Diseño del Experimento y Conocimiento Previo”.

El diagrama de flujo que se presenta en la Figura 22 resume de forma gráfica las fases de la metodología. Se debe resaltar que la fase de especificación de las meta-reglas influirá en la fase de la selección del modelo o algoritmo de extracción y a su vez en el ajuste de parámetros. Ya que dependiendo del tipo de meta-regla, obligarán a la elección de un modelo u otro, y de igual manera intervendrán en los ajustes de los parámetros.

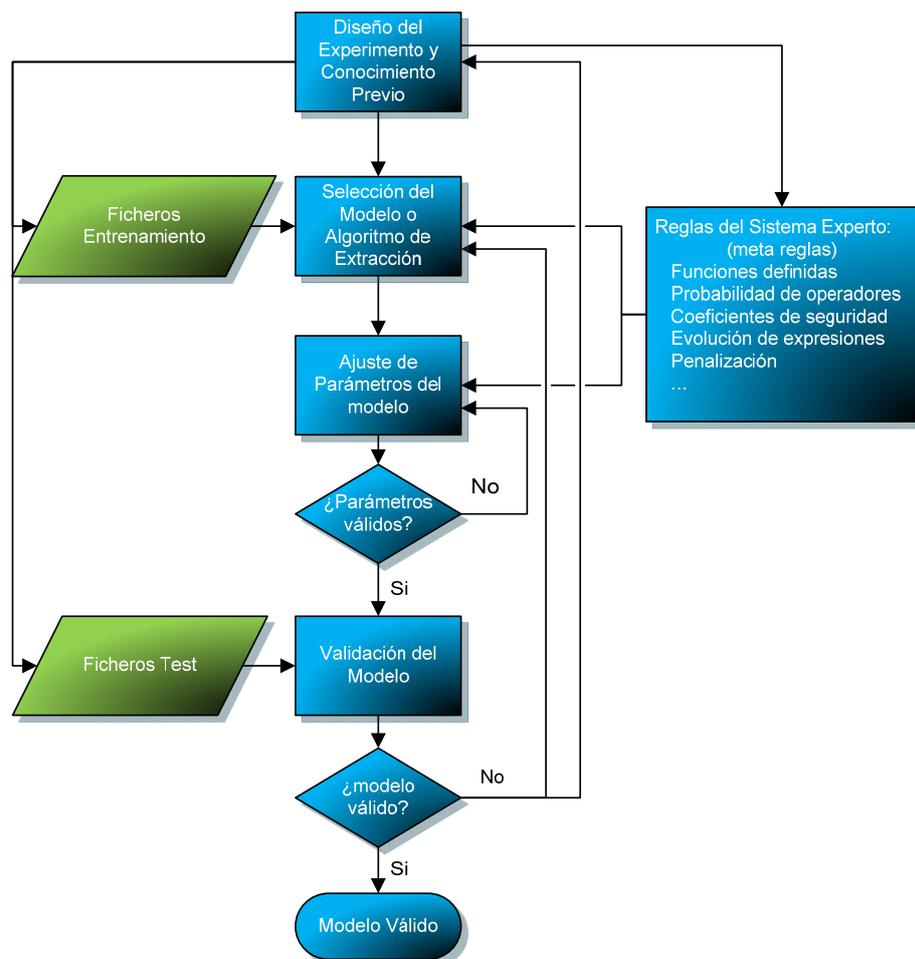


Figura 22. Diagrama de flujo de la metodología propuesta

VI. Sistema basado en conocimiento aplicado a la Programación Genética

Una vez definida la metodología, se aplica a un sistema basado en PG realizando diversas funcionalidades que otorgan la habilidad de dirigir el proceso de búsqueda de soluciones a través de las exigencias del experto. De esta forma se consigue extraer el conocimiento a partir de datos experimentales.

La primera fase de la metodología, como se ha visto, consiste en el diseño del experimento y conocimiento previo, en este caso, el diseño del experimento se centra en la realización de regresión simbólica sobre datos de experimentos procedentes de distinta índole. En esta parte se busca que el sistema desarrollado sea lo más versátil posible y se adecue a cualquier planteamiento donde se requiera realizar procesos de regresión simbólica. La forma elegida para incorporar el conocimiento en el sistema ha sido mediante un conjunto de reglas. Éstas son la base del sistema basado en conocimiento y son las que permiten dirigir la búsqueda de soluciones. Se ha adoptado esta solución debido a que las técnicas de PG tradicionales tienen ciertas limitaciones que se solventarán en el transcurso de este capítulo.

Se resumen a continuación las distintas opciones que incluye el sistema para incorporar el conocimiento del experto.

- **Módulo de operadores y funciones definidos por el usuario.** Este módulo es lo suficientemente versátil como para que el usuario pueda añadir cualquier tipo de función matemática sin necesidad de verse obligado a programarlo y compilarlo. Es importante señalar la diferencia de este módulo con respecto al desarrollo de Koza las *Automatically Defined Functions* (ADF) (Koza, 1994). En estas funciones es el propio mecanismo de PG el que selecciona un conjunto de fragmentos del árbol (sub-funciones) que luego emplea varias veces en la solución final. En cambio, en el módulo desarrollado, el experto es quién diseña las funciones según su conocimiento del problema a resolver. Otra de las funcionalidades que se aporta es la **“desplegabilidad”** de funciones definidas: Se utiliza para convertir un nodo que representa una función definida por el usuario en su árbol

equivalente utilizando los operadores estándar. Esto facilita la evolución de la población permitiendo que se unan subárboles pertenecientes a las funciones definidas por el usuario.

- **Módulo de Probabilidad de aparición de nodos.** Este módulo permite dirigir la creación de nuevos individuos. En la actualidad la creación de individuos se hace, con carácter general, de manera aleatoria. En la técnica desarrollada esta funcionalidad permite que el usuario decida qué operadores tendrán más probabilidad de ser elegidos al comienzo del algoritmo de PG a la hora de crear el conjunto de partida de individuos.
- **Módulo de coeficientes de seguridad.** En este caso el experto podrá decidir si en la evaluación de los individuos penaliza o no el ajuste al obtener valores inseguros. Se podrá penalizar en función de un umbral, por exceso o por defecto, o definiendo unos intervalos de penalización.
- **Módulo de restricciones.** En este módulo se aplican el conjunto de restricciones definidas por el experto. De esta manera se permite dirigir el proceso de búsqueda mediante el control de reglas en la creación / mutación de las soluciones (individuos). De igual manera, se permite forzar la ocurrencia de las variables. Es decir, el usuario podrá “obligar” a que los árboles puedan tener como hijos una o más variables definidas en el conjunto de datos.
- **Módulo de ajuste de expresiones.** En este caso se permite introducir una expresión como punto de partida y emplearla como esqueleto definiendo unos puntos de optimización en donde la PG podrá generar sub-expresiones atendiendo a las diferentes restricciones.

A continuación se muestra de forma más detallada cada uno de los módulos. Además, se emplea un mismo ejemplo para mostrar tanto el funcionamiento como los beneficios de cada módulo. Este ejemplo consiste en la predicción de la distancia máxima que alcanza un proyectil dada una altura, velocidad y ángulo de tiro. De igual manera se emplea la PG clásica con la misma configuración para poder comparar los resultados con los módulos desarrollados.

1 Módulo de operadores y funciones definidos por el usuario.

Este módulo surge de la necesidad de que en muchos de los casos estudiados en Ingeniería, la Programación Genética no puede conseguir crear un modelo capaz de predecir los resultados de un ensayo utilizando los operadores comunes o “clásicos” (suma, resta, multiplicación, etc.). Por otro lado, es muy frecuente, que en los fenómenos de estudio se conozcan distintas relaciones entre datos o soluciones parciales al problema, que a priori, no se pueden introducir en la PG clásica. Estos dos hechos, no conseguir el resultado mediante operadores comunes y el conocimiento de fórmulas, refuerza la idea de que sería de gran interés la introducción de dichas fórmulas a modo de operadores compactos en la PG, es decir que funcionen como un único nodo.

Por ejemplo, para un determinado problema se conoce, bien sea por la experiencia o por intuición, que la fórmula “ F ” (ver Figura 23.a) ayuda en cierta medida a encontrar su solución. En este caso, sería interesante poder definir dentro de la PG el operador no terminal “ F ” que funcionará como otro operador más, al igual que la suma, compuesto por dos variables (a y b) que

proceden de la base de datos de ensayos. A nivel de PG el operador “F” no se modificará, es decir no le afectarán las operaciones de cruce y mutación. Internamente el operador “F” (Figura 23.c) funciona como muestra la Figura 23.a. A este nuevo operador se denomina “operador-expresión” y se representa en el árbol mediante un hexágono.

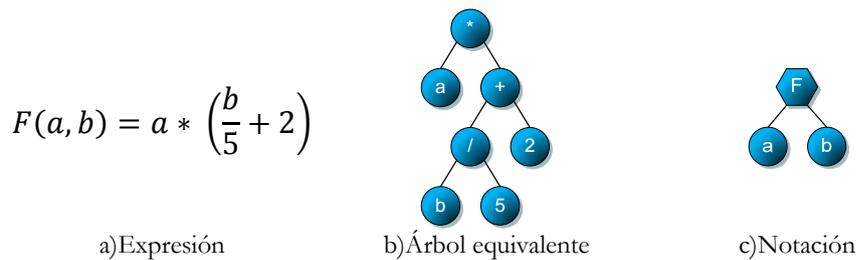


Figura 23. Ejemplo de un nuevo operador-expresión

Para dar soporte a esta nueva funcionalidad dentro de la PG es necesario definir una estructura de datos que permita en primer lugar la representación de una ecuación matemática y en segundo lugar un mecanismo que permita evaluar dicha ecuación. Unido a estos dos puntos se hace necesaria la construcción de un módulo que permita realizar un análisis sintáctico, también denominado “*parser*”, de las fórmulas introducidas mediante ficheros de texto, extrayendo las fórmulas y convirtiéndolas en operadores-expresión.

En cuanto a la representación interna de la ecuación matemática al final se ha optado por la representación de la expresión mediante una pila de elementos en notación postfija o notación polaca inversa también denominada RPN, que surgen de la notación en ingles “*Reverse Polish Notation*”. En esta notación los operandos se presentan antes que el operador que los relaciona. Con esta notación se pueden evaluar los datos directamente en el momento de ser introducidos evitando la ordenación para la evaluación de los mismos. Tal y

como sucede en otras notaciones que requieren introducir todos sus argumentos para evaluar una expresión.

Como ejemplo, si se define un operador-expresión como “ $a+b$ ” internamente se convertiría a la notación RPN que lo manejaría “ $a b +$ ”.

Se ha tomado la decisión de su empleo porque la evaluación de una expresión matemática en RPN es muy eficiente y requiere menos recursos computacionales que con otras alternativas, además no requiere paréntesis ni reglas de preferencia, al contrario que la notación algebraica. Esto es que el proceso de apilamiento permite calcular la expresión por etapas. Además, existen algoritmos que convierten las fórmulas en notación infija (de la forma en que las proporciona el experto) a RPN (notación interna del módulo).

Además de las características que se han descrito, este módulo a nivel de gestión de la información, permite añadir nuevas expresiones. Hace también posible comprobar si una determinada expresión existe, conocer el número de expresiones contenidas, etc. En último lugar, permite verificar que las expresiones introducidas están correctamente tipadas.

Para realizar la tarea de verificación del tipo de las expresiones el módulo necesita un método de tipado que reciba una expresión matemática y pueda tipar cada uno de sus elementos. Para soportar este método se necesita una estructura en la que se pueda almacenar los tipos de los operadores definidos por el usuario. De esta manera el método de tipado puede obtener el tipo resultante de la expresión de manera sencilla. El funcionamiento del método de tipado se basa en recorrer la expresión haciendo uso de una estructura de datos en forma de pila en la que se almacenan los tipos de los elementos, y de una serie de funciones que, dado un operador, devuelven los tipos de los

parámetros de dicho operador. Por ejemplo, si el elemento actual es el operador relacional “*mayor-que*” ($>$), el tipo de dicho operador será booleano, y el tipo de sus dos hijos será (real, real).

Si tras ejecutar el algoritmo de tipado, la pila de tipos está vacía, se concluye que la expresión está bien tipada y que cada operador y función recibe los parámetros necesarios.

Una vez se definidos las estructuras y los métodos para almacenar las expresiones y también la técnica empleada para poder evaluarlas, fue preciso la construcción de un módulo que le facilite al experto la inclusión de los operadores-expresión de forma natural o infija.

Para construir este módulo se sigue las recomendaciones de Aho y Sethi en las que se muestra que la mejor manera de construir un parser de las características descritas es dividirlo en tres subsistemas, un analizador léxico, un analizador sintáctico y por último un sistema de reglas semánticas (Aho, Lam, Sethi & Ullman, 2006).

El **Analizador Léxico** permite examinar un determinado texto almacenado en un fichero recorriendo cada uno de los caracteres y uniéndolos para formar componentes léxicos o *tokens*. Para ello es necesario definir una Tabla de Símbolos (TDS) en la que se halla la información de los *tokens* y un módulo de errores para detectar los símbolos no válidos dentro del léxico del lenguaje.

El **Analizador Sintáctico** permite realizar un examen sintáctico del texto introducido. El resultado obtenido tras su ejecución es la comprobación de que los *tokens* pueden formar una expresión válida del lenguaje. En todo analizador sintáctico existe una gramática que establece unas reglas (sintaxis) que indiquen cómo se pueden generar las expresiones. De esta forma la

gramática está compuesta por símbolos y reglas gramaticales que los relacionan entre sí. Un símbolo es cada tipo de unidad sintáctica o agrupación en una regla.

La gramática definida en este módulo, representada en la ecuación (10) permite reconocer expresiones de tipo numérico, donde el resultado de la evaluación será un valor real, y expresiones de tipo booleano, en este caso el resultado dará un valor de tipo lógico.

$$G = \{S, V_T, V_{NT}, P\} \quad (10)$$

Esta gramática describe un lenguaje basado en la utilización de expresiones aritméticas con sumas, diferencias, productos, divisiones, paréntesis, cambio de signo, etc. De igual manera se pueden definir el uso de constantes e identificadores de variables.

La gramática está compuesta por un vocabulario terminal (VT) formado por los símbolos y tokens que se muestran en la ecuación (11). El vocabulario no terminal (VNT) contiene el símbolo S y los *tokens* que se muestran en la ecuación (12).

$$\begin{aligned} VT = \{ & + \mid - \mid * \mid / \mid ^ \mid \text{-cambio de signo} \mid \text{sqr} \mid \text{sqr3} \mid \text{sqrt} \\ & \mid \text{sqrt3} \mid \text{sen} \mid \text{cos} \mid \text{tan} \mid \text{senh} \mid \text{cosh} \mid \text{tanh} \\ & \mid \text{asen} \mid \text{acos} \mid \text{atan} \mid \text{ln} \mid \text{log10} \mid \text{exp} \mid \text{tansig} \\ & \mid \text{logsig} \mid \text{abs} \mid \text{sign} \mid \text{escalon} \mid (\mid) \mid \text{variable} \\ & \mid \text{true} \mid \text{false} \mid \text{OR} \mid \text{AND} \mid \text{NOT} \mid == \mid > \mid < \\ & \mid >= \mid <= \mid <> \mid \text{constante} \mid \text{otrosCaracteres} \\ & \mid , \mid : \mid \backslash n \mid \text{comentario} \} \end{aligned} \quad (11)$$

$$\begin{aligned}
 \text{VNT} = \{ & \text{opArit} \mid \text{func} \mid \text{termBool} \mid \text{parámetro} \\
 & \mid \text{exprBool} \mid \text{literalBool} \mid \text{relación} \mid \text{factorBool} \\
 & \mid \text{opBool} \mid \text{opRel} \mid \text{factorNot} \mid \text{exprNum} \\
 & \mid \text{expresión} \mid \text{líneaComent} \\
 & \mid \text{coment} \mid \text{sentencia} \mid \text{S} \}
 \end{aligned} \tag{12}$$

Las reglas de producción P que representan al lenguaje se muestran en la ecuación (13) numeradas del (0) al (16). La numeración se utiliza para indicar la regla que se aplica en cada derivación.

$$\begin{aligned}
 (0) \quad & \text{S} \rightarrow \text{sentencia} \mid \text{S sentencia} \\
 (1) \quad & \text{sentencia} \rightarrow \backslash \text{n} \mid \text{variable} : \text{expresión} \backslash \text{n} \mid \text{FIN} \backslash \text{n} \\
 & \quad \mid \text{comentario líneaCom} \backslash \text{n} \\
 (2) \quad & \text{expresión} \rightarrow \text{exprNum} \mid \text{exprBool} \\
 (3) \quad & \text{exprNum} \rightarrow \text{constante} \mid \text{variable} \mid (\text{exprNum}) \\
 & \quad \mid \text{exprNum} + \text{exprNum} \mid \text{exprNum} \\
 & \quad \mid \text{exprNum} \mid \text{exprNum} * \text{exprNum} \\
 & \quad \mid \text{exprNum} / \text{exprNum} \\
 & \quad \mid \text{exprNum} ^ \text{exprNum} \\
 & \quad \mid - \text{cambio de signo} \text{exprNum} \\
 & \quad \mid \text{func} (\text{exprNum}) \\
 & \quad \mid \text{variable} () \mid \text{variable} (\text{parámetro}) \\
 & \quad \mid \text{if exprBool then exprNum else} \\
 & \quad \quad \text{exprNum} \\
 (4) \quad & \text{opArit} \rightarrow + \mid - \mid * \mid / \mid ^ \mid - \text{cambio de signo}
 \end{aligned} \tag{13}$$

- (5) $\text{func} \rightarrow \text{sqr} \mid \text{sqrt} \mid \text{sqr3} \mid \text{sqrt3} \mid \text{sen} \mid \text{cos} \mid \text{tan} \mid$
 $\mid \text{sinh} \mid \text{cosh} \mid \text{tanh} \mid \text{asen} \mid \text{acos} \mid \text{atan}$
 $\mid \text{ln} \mid \text{log10} \mid \text{exp} \mid \text{tansig} \mid \text{logsig} \mid$
 $\mid \text{abs} \mid \text{sign} \mid \text{escalon}$
- (6) $\text{exprBool} \rightarrow \text{termBool} \mid \text{termBool OR termBool}$
 $\mid \text{variable} (\text{parametro})$
- (7) $\text{termBool} \rightarrow \text{factorNot} \mid \text{factorNot AND factorNot}$
- (8) $\text{factorNot} \rightarrow \text{factorBool} \mid \text{NOT factorBool}$
- (9) $\text{factorBool} \rightarrow \text{literalBool} \mid \text{variable} \mid (\text{exprBool})$
 $\mid \text{relacion}$
- (10) $\text{literalBool} \rightarrow \text{true} \mid \text{false}$
- (11) $\text{opBool} \rightarrow \text{OR} \mid \text{NOT} \mid \text{AND}$
- (12) $\text{relacion} \rightarrow \text{exprNum opRel exprNum}$
- (13) $\text{opRel} \rightarrow \text{==} \mid > \mid < \mid \text{>=} \mid \text{<=} \mid \text{<>}$
- (14) $\text{parámetro} \rightarrow \text{expresión} \mid \text{parámetro} , \text{parámetro}$
- (15) $\text{líneaComent} \rightarrow \text{coment} \mid \text{líneaComent coment}$
- (16) $\text{coment} \rightarrow \text{variable} \mid \text{constante} \mid \text{literalBool}$
 $\mid \text{opArit} \mid \text{func} \mid \text{opBool} \mid \text{opRel}$
 $\mid \mid , \mid : \mid (\mid) \mid \text{otrosCaracteres}$

Una característica habitual de una gramática es su ambigüedad, es decir, si existe más de un árbol sintáctico para cualquier sentencia generada por ella. La ambigüedad de una gramática es una propiedad indecidible, lo que significa que no existe ningún algoritmo que determine con certeza en un plazo finito de tiempo si una gramática es ambigua o no. A lo sumo que se ha llegado en el estudio de la ambigüedad es que hay algunas condiciones que de cumplirse determinan que existe ambigüedad, pero en el caso de no cumplirse no se puede decir que la gramática es no ambigua.

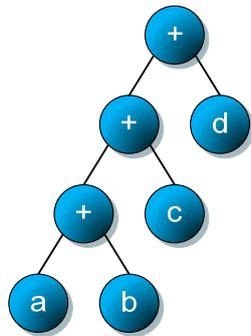
Una de las principales formas de evitar la ambigüedad es definiendo la precedencia y asociatividad de los operadores. Para ello, en la gramática implementada se define el orden de precedencia de evaluación de las expresiones y de los operadores tales como se muestra en la Tabla 1.

Tabla 1. Prioridad de los operadores

Orden	Operador
1	() variable constante
2	- (cambio de signo)
3	^
4	* /
5	+ -

Además, todos los operadores que se emplean presentan *asociatividad por la izquierda*, a excepción del operador potencia (^). En la Figura 24 se muestra un ejemplo de cada tipo.

$$F_1 = a + b + c + d$$



$$F_1 = a^{b^c}$$

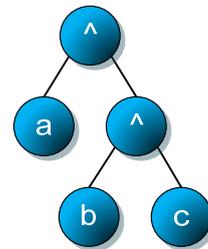


Figura 24. Ejemplos de árbol sintáctico con asociatividad por la izquierda y derecha

Estas dos propiedades, precedencia y asociatividad, son suficientes para convertir la gramática ambigua basada en operadores en no ambigua, es decir,

que cada sentencia tenga sólo un árbol sintáctico. Esto es de vital importancia para poder evaluar correctamente las expresiones introducidas.

1.1 Despliegue de funciones

Como ya se ha comentado, cuando el experto define una función (que generalmente estará compuesta por 2 o más operaciones) dentro del sistema se interpreta como si se tratase de una operación simple, es decir, dentro del árbol la función es manipulada como si se tratase de un único nodo. Una vez establecidos el conjunto de “operadores-expresión” y dado un momento determinado en el proceso evolutivo puede ser interesante descomponer los “operadores-expresión” en los operadores “habituales” en PG. Para ello se define una propiedad dentro de los nodos pudiendo ser desplegables o no desplegables. Todos los nodos, a excepción de los “operadores-expresión” tendrán la propiedad de ser no desplegables. Es decir cualquier nodo, terminal o no, que representa operaciones clásicas o elementos conocidos, como por ejemplo la suma, el coseno, la constante “7”, siempre serán nodos no desplegables, puesto que no se pueden descomponer en otros elementos. En cambio únicamente los nodos “operador-expresión” se pueden desplegar. De esta manera, en el momento que se desean desplegar, se intercambia el nodo operador-expresión por la estructura interna definida por el usuario. Es decir cada vez que se alcance a un nodo susceptible de ser desplegado se crea un árbol equivalente y se añade como subárbol dentro del árbol original. Estos árboles equivalentes estarán compuestos por operaciones más simples y generalmente tendrán mayor altura que el original.

Asociado a la operación de despliegue se define el concepto de **Nivel de Despliegue**. Esta propiedad representa el número de veces que se puede

desplegar un nodo que es desplegable, hasta llegar a un árbol cuyos componentes son todos no desplegables. Dada esta definición, los nodos que no son desplegables tienen nivel de despliegue 0.

Este concepto, novedoso dentro del la PG, de nivel de despliegue tiene un papel relevante dentro del módulo desarrollado ya que permite el despliegue controlado, por niveles, de funciones. Es decir, como se permite definir funciones a partir de otras previamente definidas y así sucesivamente, se puede elegir hasta que nivel de recursión se despliega, definir funciones en las que se empleen otras ya definidas anteriormente y así sucesivamente.

$$F_1(a, b) = \frac{\ln(25 * b)}{a} \quad (14)$$

$$F_2(a, b) = a - F_1(b, 5) \quad (15)$$

$$F_3(a, b, c) = \frac{F_2((F_1(a, 8), 3))}{b} * c^3 \quad (16)$$

$$F_4(a, b) = \text{if}(a < 25) \text{ then } (F_1(a, b)) \text{ else } (a * b) \quad (17)$$

Para comprender el concepto de nivel de despliegue se muestra un ejemplo de su aplicación. Sean las funciones F_1 (14), F_2 (15), F_3 (16) y F_4 (17) expresiones definidas por el usuario, si se realiza una operación de despliegue en la función F_i se obtiene un árbol (Figura 25) en el que todos los nodos son no desplegables. Por lo tanto la F_i tiene el nivel de despliegue de 1, ya que únicamente necesita una operación de despliegue para que todos sus nodos tengan nivel de despliegue de 0. En cambio la función F_2 , es necesario realizar la operación de despliegue dos veces (Figura 26), por lo que se establece su nivel de despliegue en 2. De forma análoga se calcula el nivel para las funcione F_3 obteniendo un nivel de 3 y para F_4 que obtiene un nivel de 2.

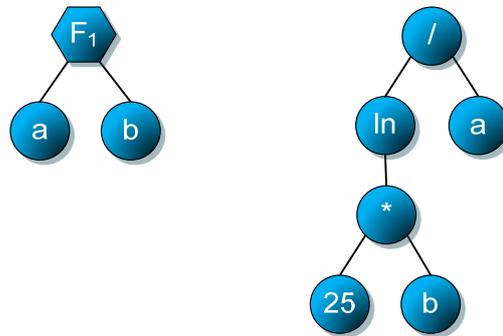


Figura 25. Función F_1 y su equivalente desplegado

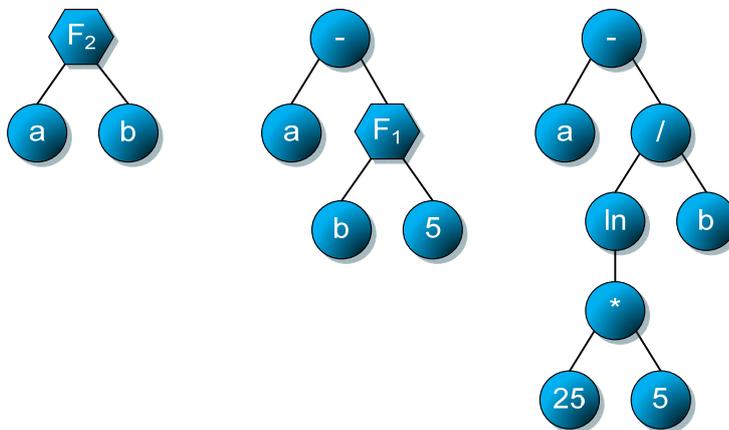


Figura 26. Despliegue de la función F_2

Dentro del módulo definido se añade el parámetro de **nivel máximo de recursividad permitida**, de esta manera el experto puede decidir hasta qué profundidad se despliegan las funciones. Para este cometido se permiten 3 variantes:

- No permitir el despliegue de las funciones.
- Funciones totalmente desplegadas. Cualquier operador-expresión se descompondrá en los nodos habituales de la PG. Para ello se establece el nivel máximo de recursividad permitida a 0.

- Establecido un nivel “n” de recursividad. Las funciones se despliegan hasta que el nivel máximo de despliegue de los nodos es “n”

De este modo, si en un determinado momento del proceso evolutivo de la PG se desea desplegar las ecuaciones tan solo sería necesario establecer hasta qué nivel de recursividad se permite. Por ejemplo, dado el individuo codificado en la ecuación (18) el usuario desea establecer el nivel máximo de recursividad permitida a 2. En la Figura 27 se muestra, en forma de árbol, el individuo I_1 sin desplegar (con los operadores-expresión).

$$I_1 = \frac{F_2(2,9)}{F_4(x,16)} - F_3(y,27,8) \quad (18)$$

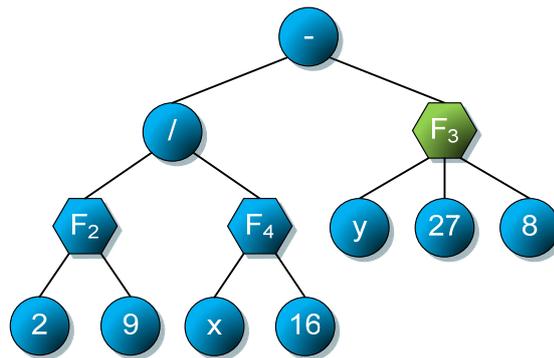


Figura 27. Individuo I_1 (18) sin desplegar

De esta manera se van desplegando todos los nodos operadores-expresión del individuo cuyo nivel de despliegue es superior al valor máximo de recursividad permitida. En este caso todos aquellos nodos superiores a 2, por esta razón los nodos que contienen F_3 deben ser desplegados (en la Tabla 2 se muestra un resumen de las ecuaciones y su correspondiente nivel de despliegue). En la Figura 27 se muestra en color verde el nodo a desplegar. Y en la Figura 28 el resultado después de realizar las operaciones de despliegue.

Tabla 2 Nivel de despliegue para las F_1 (14), F_2 (15), F_3 (16) y F_4 (17)

Función	Nivel de Despliegue
$F_1(a, b) = \frac{\ln(25 * b)}{a}$	1
$F_2(a, b) = a - F_1(b, 5)$	2
$F_3(a, b, c) = \frac{F_2((F_1(a, 8), 3))}{b} * c^3$	3
$F_4(a, b) = \text{if}(a < 25) \text{ then } (F_1(a, b)) \text{ else } (a * b)$	2

En cambio, si se establece el nivel máximo de recursividad permitida en 1 se deberían desplegar todas las funciones a excepción de la F_1 . Por último, en la Figura 29 se muestra el resultado de aplicar la operación de despliegue a todos los nodos, en correspondencia con el nivel máximo de recursividad permitida igual a 0.

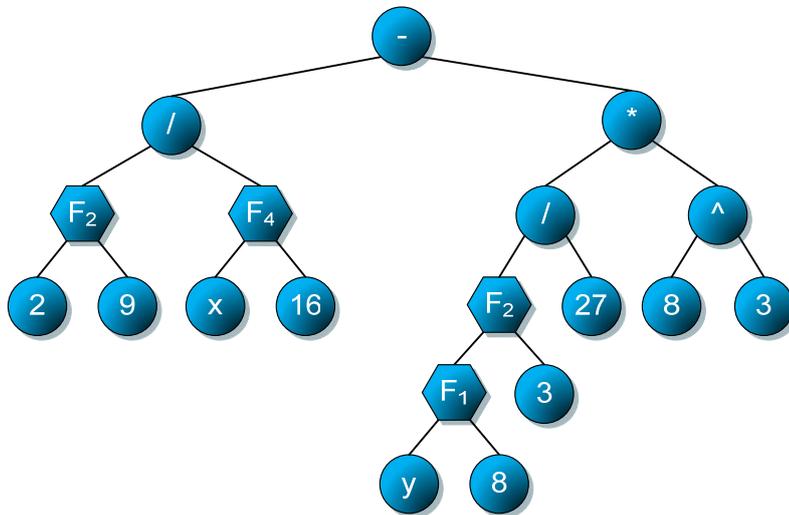


Figura 28. Individuo I_1 (18) con nivel máximo de recursividad a 2

Cabe resaltar que el módulo desarrollado hace distinción entre variables provenientes del conjunto de entrenamiento e incógnitas definidas en el conjunto de expresiones. Para este cometido es importante identificar correctamente las variables de entrada. Esto es muy útil, por ejemplo si se desea predecir el comportamiento de un fenómeno a partir de un conjunto de ensayos experimentales. En este caso, el conjunto de datos estará compuesto por las entradas a , b , c (conjunto de características del ensayo) y una salida d (valor a predecir). Si en la definición del operador expresión se emplean las variables de entrada a , b , c , se ligarán automáticamente con el conjunto de datos. Es decir durante el proceso de evaluación del individuo se tomarán los valores de cada variable del conjunto de datos. En cambio, si aparece otra variable, ésta se convertirá en el punto de creación de sub-ramas por parte de la PG. Esta última variable es la única parte del operador expresión donde la PG podrá realizar las operaciones genéticas de cruce y mutación.

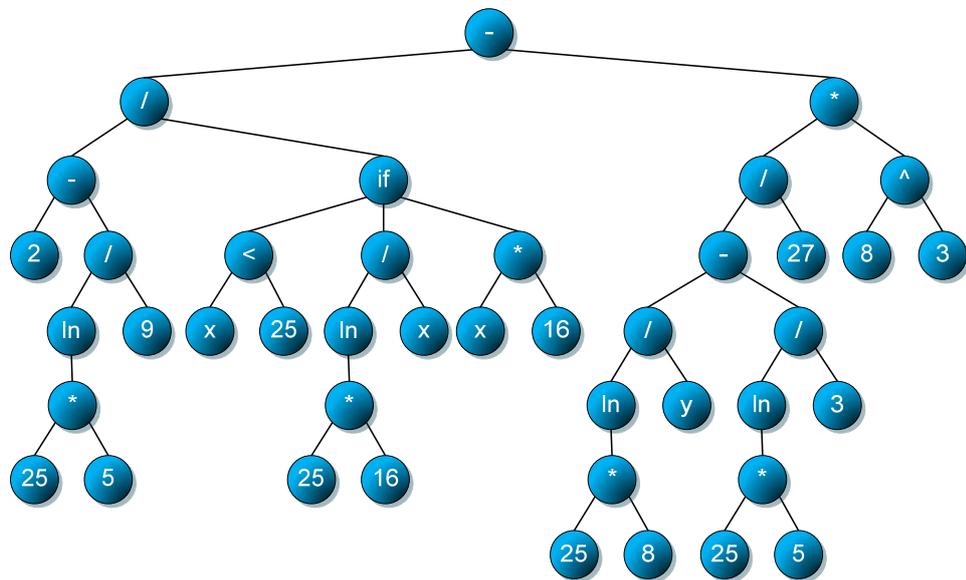


Figura 29. Individuo I_1 (18) totalmente desplegado

$$E_1(a, b, x) = \sqrt{\frac{x * a}{b}} \quad (19)$$

Dentro del proceso de PG se distinguen dos comportamientos diferentes para los operadores expresión. El primero tendría lugar durante la evolución de los individuos (aplicación de operaciones genéticas) y el segundo se corresponde con la evaluación de dichos individuos.

$$\text{Individuo} = 25 * a + E_1(3) \quad (20)$$

Continuando con el ejemplo anterior, si se retoma el conjunto de datos y se añade como operador expresión la ecuación (19), en este caso, en la evolución de la población, el operador expresión E_1 , sólo tiene como variable de entrada x , ya que las otras dos: a y b provienen del conjunto de datos. Siguiendo con el mismo ejemplo, tomando el individuo descrito en la ecuación (20), el operador expresión E_1 , a efectos evolutivos toma únicamente la constante 3 . En la Figura 30 se muestra este individuo en forma de árbol. A partir de éste, y como en cualquier otro árbol, la PG podrá evolucionar en cualquier nodo terminal (25 , a y 3) como no terminal ($*$, $+$, E_1).

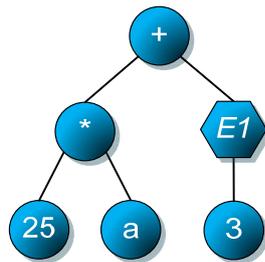


Figura 30. Individuo (20) a efectos de evolución

En cambio, durante la evaluación de los individuos, el operador expresión E_1 estará formado por 3 variables (a , b y x) de las cuales provienen las dos

primeras del conjunto de datos y la última del individuo. Siguiendo el mismo ejemplo para la evaluación del individuo codificado en la ecuación (20) el operador expresión E_1 , a efectos de la evaluación toma como entradas a y b del conjunto de datos y la constante 3 . En la Figura 31 se muestra gráficamente el resultado de la evaluación del individuo (20) con el conjunto de datos.

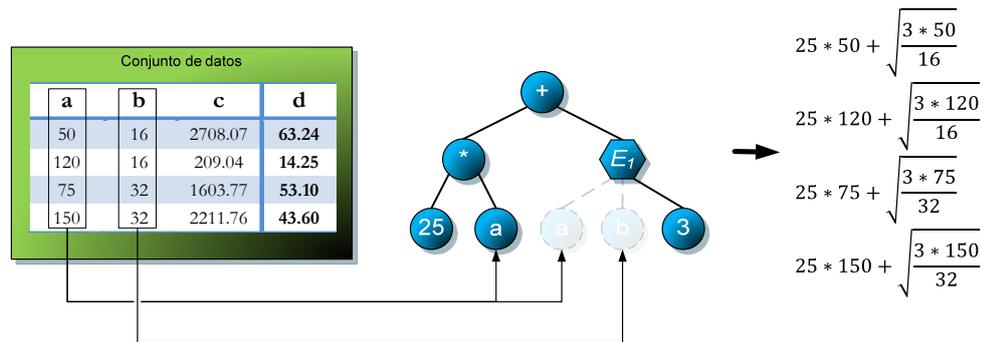


Figura 31. Evaluación del Individuo (20)

1.2 Ejemplo de aplicación

Con el objetivo de ilustrar las capacidades del módulo desarrollado se va a aplicar sobre un problema típico en el campo de la física: determinar la longitud que alcanzará un proyectil en tiro parabólico que sólo se ve afectado por la velocidad inicial y la fuerza de la gravedad (Giancoli, 2005). Para simplificar el ejemplo, en la ecuación que describe el fenómeno (21), no se tiene en cuenta las otras fuerzas que pueden influir en el resultado, como por ejemplo la fricción de la resistencia del aire. En la Figura 32 se muestra de forma gráfica el fenómeno físico que se ha utilizado.

$$d = \frac{v_0 * \cos(\theta)}{g} \left(v_0 * \sin(\theta) + \sqrt{(v_0 * \sin(\theta))^2 + 2 * g * y_0} \right) \quad (21)$$

La ecuación (21) da como resultado la distancia alcanzada por un proyectil lanzado con una velocidad inicial v_0 , las variables que intervienen se describen a continuación:

- g : la aceleración gravitacional (establecido en $9,81 \text{ m/s}^2$).
- θ : el ángulo bajo el cual se ha lanzado el proyectil (rad).
- v_0 : la velocidad inicial a la que fue lanzado el proyectil (m/s)
- y_0 : la altura inicial del proyectil (m). Es un valor positivo si está por encima de la superficie terrestre, negativo si está por debajo.
- d : La distancia horizontal total recorrida durante el vuelo (m).

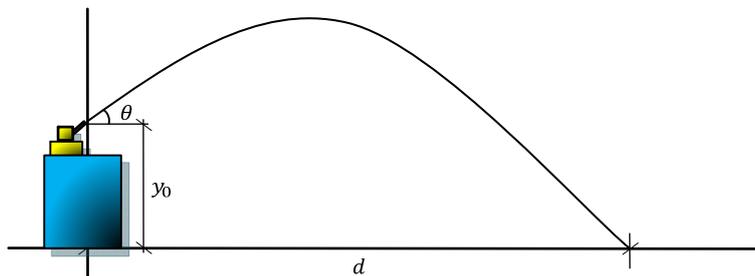


Figura 32. Alcance máximo de un proyectil

1.2.1 Conjunto de datos

Los datos que se han empleado para este ejemplo se han generado mediante la aplicación de la ecuación (21). En la Tabla 3 se muestra el intervalo creado para cada variable, y el paso (de cuanto en cuanto) se ha creado. En total se ha generado un conjunto de datos de 405 patrones (9 valores de $\theta * 5$ valores de

v_0 * 9 valores de y_0). Para el conjunto de entrenamiento se han empleado 284 patrones y 121 para el conjunto de test.

Tabla 3. Conjunto de datos empleado

Variable	Min	Max	paso
θ	5	85	10
v_0	20	100	20
y_0	0	200	25
d	7.08	1190.60	-

1.2.2 Configuración de parámetros

Aunque ya se conoce la ecuación del fenómeno físico que describe la relación entre las entradas y salida, se asume como desconocida y que consultamos a un experto que, debido a su experiencia, estima que las ecuaciones $E_1(22)$, $E_2(23)$ y $E_3(24)$ podrían ayudar a encontrar la solución dentro del proceso de búsqueda, Por ello se incluirán como posibles operadores dentro de la configuración del algoritmo de PG. Adicionalmente se añaden otros operadores que están disponibles para la evolución: suma, producto, división protegida, raíz cuadrada, cuadrado. Para los nodos terminales se pueden emplear las variables procedentes del conjunto de datos (θ , v_0 e y_0) y la constante de la aceleración gravitacional (9.81).

En la Tabla 4 se muestra la configuración empleada para el conjunto de pruebas realizadas. El ajuste se ha realizado mediante el error cuadrático medio combinado con el valor de parsimonia. El criterio de terminación que se ha adoptado para el conjunto de pruebas ha sido que, o bien se alcanza la generación máxima, (establecido en 20000 generaciones) o bien se alcanza un $ECM \approx 0$.

$$E_1 = v_0 * \frac{\cos(\theta)}{9.81} \quad (22)$$

$$E_2 = v_0 * \text{sen}(\theta) \quad (23)$$

$$E_3 = \sqrt{\text{valor} + (2 * 9.81 * y_0)} \quad (24)$$

Tabla 4. Parámetros del ejemplo del proyectil

Parámetro	Valor
Generación máxima	20000
Tamaño Población	500
Tasa de Cruces	95%
Tasa de Selección no Terminal	90%
Probabilidad Mutación:	5%
Algoritmos	Selección: Torneo Creación: Intermedio Mutación: Subárbol
Altura inicial de los individuos	4
Altura máxima de mutación	4
Altura máxima de los individuos	6
Estrategia elitista	si
Parsimonia	0.1

1.2.3 Resultados

Se realizaron un conjunto de 10 pruebas mediante la PG clásica y otras 10 mediante el módulo aquí presentado. Con el objetivo de que ambos métodos puedan ser comprobables, se ha empleado la misma configuración de parámetros, con la particularidad de que, como es lógico, la eliminación en el

caso de la PG clásica, de los operadores expresión y de la inclusión de los operadores seno y coseno.

En la Tabla 5 se muestran los resultados del conjunto de pruebas realizadas. Por orden aparece el nombre de la ejecución, las generaciones realizadas (Gen), el error medio (EM) y error cuadrático medio (ECM) en entrenamiento (E) y Test (T), el tiempo empleado (Tiempo) en segundos y las generaciones por minuto (GPM). Cabe destacar que ninguna de las ejecuciones ha conseguido obtener un ajuste perfecto. Si se toma como referencia el ECM los mejores individuos de la ejecución 1 y 7 obtienen un error de 1263 metros, en cambio con los patrones de test se ha alcanzado un ECM de 1627 metros. Por el contrario si se toma como referencia el EM se obtiene que el mejor individuo se ha obtenido en la ejecución 4 siendo el error de 23.21 y 27.48 para entrenamiento y test respectivamente.

Tabla 5. Pruebas realizadas mediante PG Clásica

Nombre	Gen.	EM (E)	ECM (E)	EM (T)	ECM (T)	Tiempo (s)	GPM
Ej.1	20000	26.39	1263	30.27	1627	5796	207.0
Ej.2	20000	24.26	1742	28.26	2342	6222	192.9
Ej.3	20000	70.51	7019	74.60	9002	4880	245.9
Ej.4	20000	23.21	1548	27.48	2166	5913	202.9
Ej.5	20000	65.23	7737	70.21	10822	5947	201.8
Ej.6	20000	37.15	2153	38.40	2540	5609	213.9
Ej.7	20000	26.39	1263	30.27	1627	4827	248.6
Ej.8	20000	122.90	22550	140.78	32162	5486	218.7
Ej.9	20000	44.86	3661	49.20	4215	8108	148.0
Ej.10	20000	27.26	1270	31.03	1634	7278	164.9
Promedio:	20000	46.82	5021	52.05	6814	6006.6	204.5

Por término medio para realizar una ejecución de 20000 generaciones ha sido necesario 6006.6 segundos (1h 40' 6.6"). Visto de otra forma se obtiene que en

un minuto, con la configuración empleada, se consigue por término medio realizar 204.5 generaciones. Para tener una aproximación del esfuerzo computacional en función del tiempo de cálculo empleado, cabe destacar que las pruebas se han realizado en equipos con el mismo microprocesador (Intel® Core™ i7 920@ 2.67GHz) asignando a cada ejecución un solo núcleo.

$$PG_{clásica}(Ej.1) = \frac{2 v_0^2 * \cos(\theta) * \text{sen}(\theta)}{9.81} + \sqrt{\frac{v_0 * y_0}{\theta}} \quad (25)$$

$$PG_{clásica}(Ej.7) = \frac{v_0^2 * \text{sen}(2\theta)}{9.81} + \frac{\sqrt{v_0} * \sqrt{y_0}}{\sqrt{\theta}} \quad (26)$$

En este caso, con la configuración que se ha comentado anteriormente, la mejor solución obtenida se ha alcanzado en dos ocasiones (ejecución 1 y 7). Estas dos soluciones son equivalentes ya que la estructura correspondiente del árbol es diferente. Además el ajuste obtenido difiere un poco en la ejecución 1, se obtiene un ajuste de 1264.89 frente a 1264.99 del ajuste de la ejecución 7. En las ecuaciones (25) y (26) se muestran el mejor individuo de las ejecuciones 1 y 7 respectivamente.

Para poder comparar el módulo desarrollado frente a los métodos clásicos de PG se muestran en la Figura 33 y en la Figura 34 la evolución de ECM del mejor individuo de cada ejecución durante las generaciones. De forma gráfica se muestra que los individuos 1 y 7 son los que obtienen un mejor ajuste. Cabe resaltar que, modificando los parámetros de la PG, se podría llegar a obtener resultados similares o incluso mejores en cuanto a los errores cometidos pero a costa de obtener una solución mucho más compleja.

Ahora bien, si se emplea el módulo desarrollado con las expresiones definidas por el usuario $E_1(22)$ $E_2(23)$ y $E_3(24)$. Se obtienen resultados muy satisfactorios. Por norma general se obtiene la misma fórmula objetivo (21) o

fórmulas equivalentes, ecuaciones (27) y (28). En ocasiones y debido al carácter estocástico de la técnica, puede suceder que no se alcance la ecuación analítica. Esta situación ha sucedido en la ejecución 7, puesto que ha sido incapaz de llegar a la solución óptima. De todas formas, cómo se puede ver en la Tabla 6, la ejecución 7 obtiene un EM (4.68 m) y un ECM (53 m) más pequeño que cualquier ejecución empleando la PG clásica (EM 23.21 m, ECM 1263 m). Es probable que modificando dinámicamente los valores de la parsimonia la ejecución 7 obtuviese el resultado óptimo.

$$PG_{FD}(Ej. 1) = E_2 * E_1 + E_1 * E_3(E_2^2) \quad (27)$$

$$PG_{FD}(Ej. 8) = (E_3(E_2^2) + E_2) * E_1 \quad (28)$$

$$PG_{FD}(Ej. 7) = E_1 * (E_2^2 + \sqrt{E_1}) + \frac{(\sqrt{y_0} + E_3(y_0)) * \sqrt{E_1}}{E_2} + \frac{\frac{\sqrt{y_0} * E_1}{\sqrt{y_0} + 2E_2}}{\frac{E_2 + 9.81}{E_3(E_2)}} + \frac{E_3(E_2)}{4E_2} \quad (29)$$

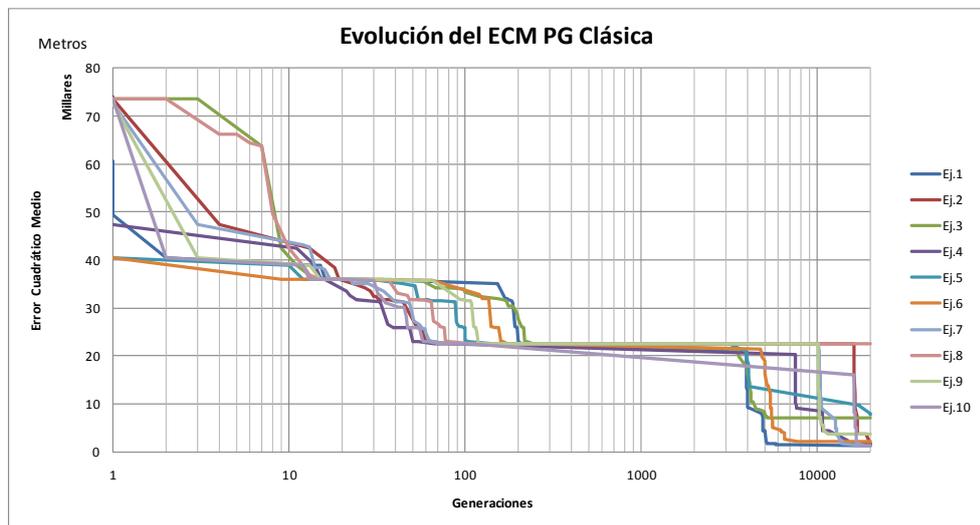


Figura 33. Evolución de las pruebas mediante PG clásica

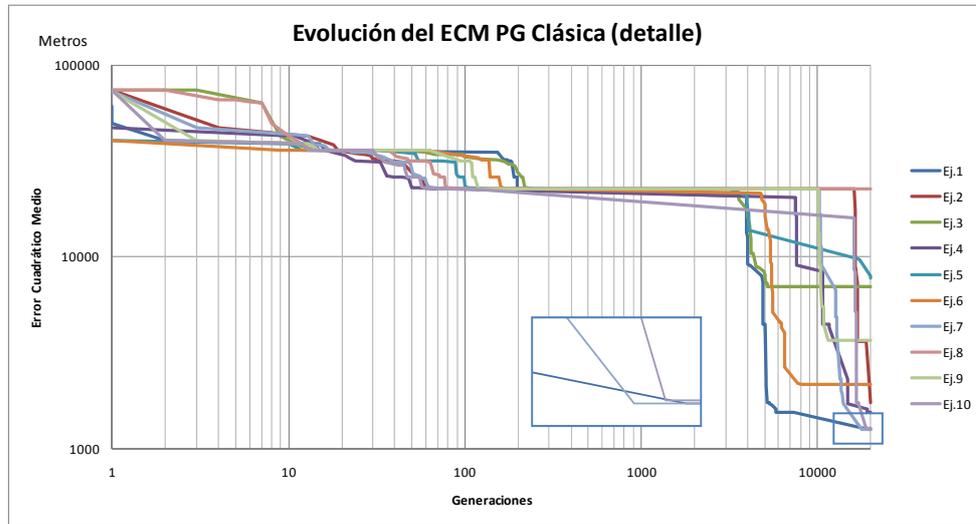


Figura 34. Evolución de las pruebas mediante PG clásica (detalle)

Tabla 6. PG con funciones definidas por el usuario

Nombre	Gen.	EM(E)	ECM(E)	EM(T)	ECM(T)	Tiempo(s)	GPM
Ej.1	99	≈ 0	≈ 0	≈ 0	≈ 0	536	11.1
Ej.2	51	≈ 0	≈ 0	≈ 0	≈ 0	167	18.3
Ej.3	46	≈ 0	≈ 0	≈ 0	≈ 0	140	19.7
Ej.4	122	≈ 0	≈ 0	≈ 0	≈ 0	665	11.0
Ej.5	269	≈ 0	≈ 0	≈ 0	≈ 0	2021	8.0
Ej.6	159	≈ 0	≈ 0	≈ 0	≈ 0	849	11.2
Ej.7	20000	4.68	53	5.69	78	132893	9.0
Ej.8	82	≈ 0	≈ 0	≈ 0	≈ 0	378	13.0
Ej.9	525	≈ 0	≈ 0	≈ 0	≈ 0	5322	5.9
Ej.10	76	≈ 0	≈ 0	≈ 0	≈ 0	335	13.6
Promedio:	2142.9	0.47	5	0.57	8	14330.6	12.1

Esta funcionalidad del módulo desarrollado, que permite la adición de cualquier expresión como operador, tiene un coste computacional alto. En la Tabla 6 se mostraba el tiempo empleado, por término medio en un minuto se puede realizar 12 generaciones frente a las 204.5 generaciones con los métodos tradicionales. Este hecho está motivado por la necesidad de realizar más

operaciones, principalmente vinculadas con la gestión y evaluación de las expresiones. De igual manera, se constata un incremento del consumo de memoria. De todos modos, con la excepción de la ejecución 7, con menos de 160 generaciones se consigue un ajuste óptimo.

En la Figura 35 y con más detalle en la Figura 36 se muestra la evolución del mejor individuo de cada ejecución. De forma visual se muestra cómo la Ej. 7 no consigue obtener la expresión. Cabe destacar en la Figura 36 las ejecuciones 1, 5, 6, 10 las variaciones del ECM en el intervalo 1 al 0.001. Estas variaciones son producidas por la influencia del factor de parsimonia en el cálculo del ajuste.

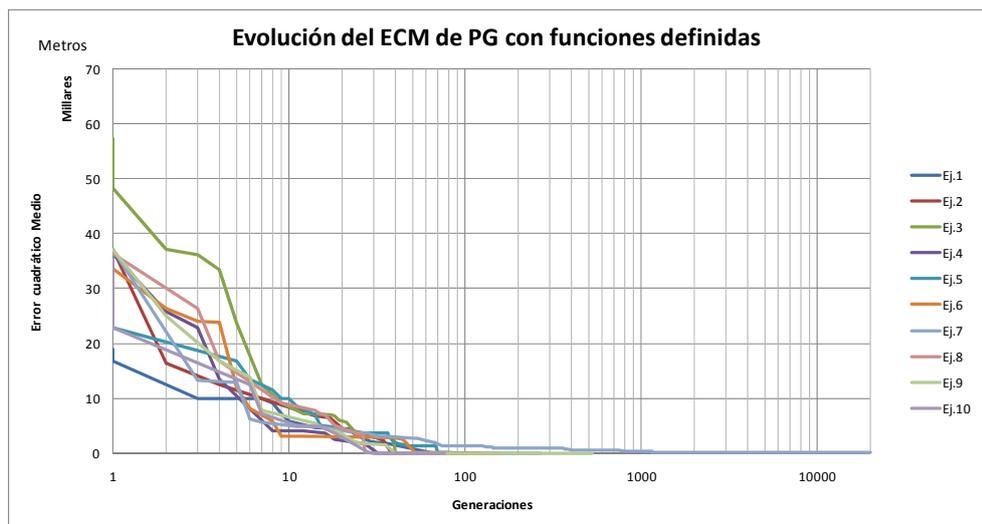


Figura 35. Evolución de las pruebas mediante PG con funciones definidas

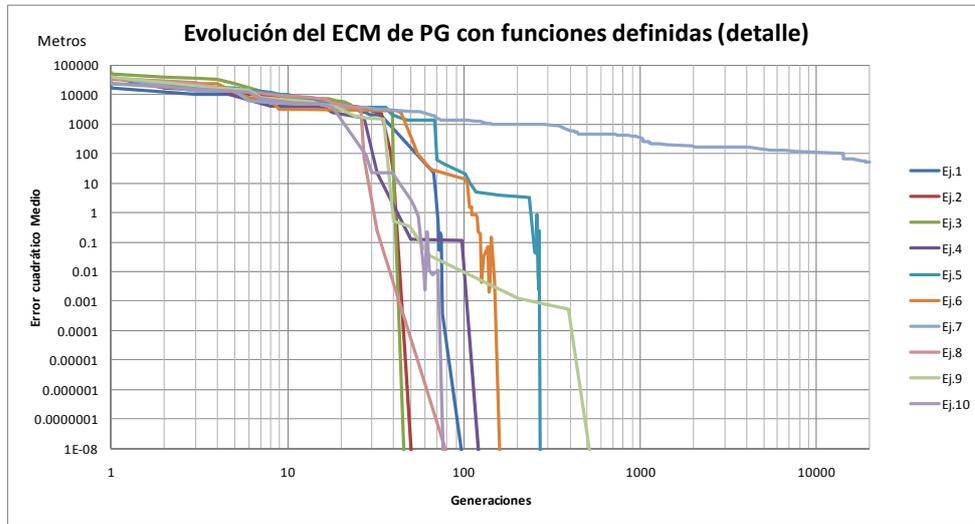


Figura 36. Evolución de las pruebas mediante PG con funciones definidas (detalle)

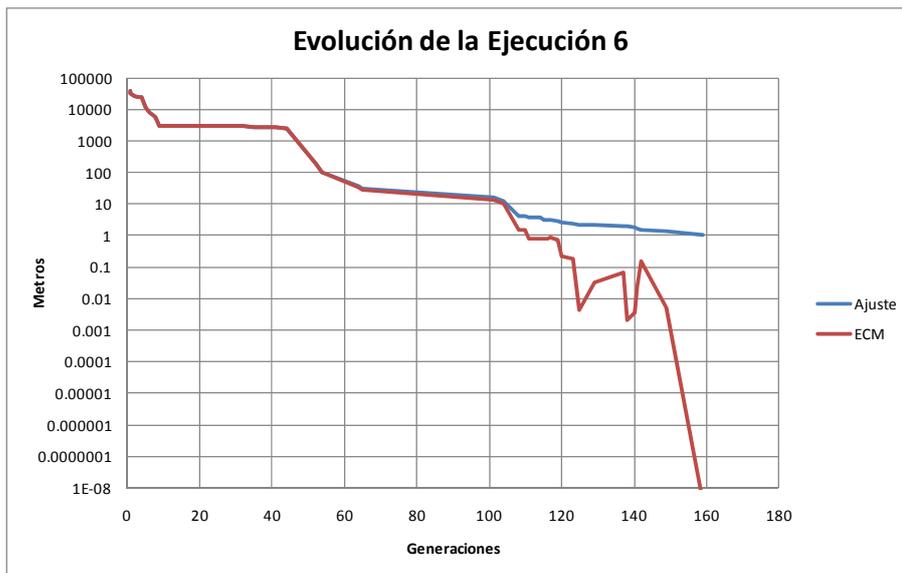


Figura 37. Evolución del ajuste y ECM de la ejecución 6

2 Módulo de probabilidades asociadas a operadores

Como se ha visto anteriormente, de manera habitual el proceso de creación de individuos es totalmente aleatorio. Este hecho, desde el punto de la orientación en el proceso de búsqueda, presenta la limitación de que el usuario no puede tener control sobre el crecimiento del árbol, simplemente establece una altura máxima. Por este motivo se ha desarrollado dentro del Sistema Basado en Conocimiento (SBC) la funcionalidad de poder asignarle a cada prototipo (operador) una probabilidad de ser elegido. A mayor probabilidad, mayor será el número de veces que aparece un prototipo tras realizar la creación inicial de individuos.

2.1 Modificación de algoritmos.

Para albergar esta funcionalidad en el SBC es necesario modificar el comportamiento tanto del algoritmo de creación como del algoritmo de mutación con el objetivo de cambiar el comportamiento aleatorio en la elección de los elementos. De esta manera, mediante probabilidades, se orientan ambos procesos (creación y mutación).

Para la aplicación de este método se ha elegido el algoritmo basado en “Ruleta de Probabilidades”. Este método consiste en asignar un espacio de la ruleta, proporcional a la probabilidad asignada, a cada operador. En la Figura 38 se muestra la ruleta resultante de la Tabla 7. Conceptualmente se obtiene el

operador girando la ruleta y en el lugar donde se ha parado será el espacio elegido de un operador.

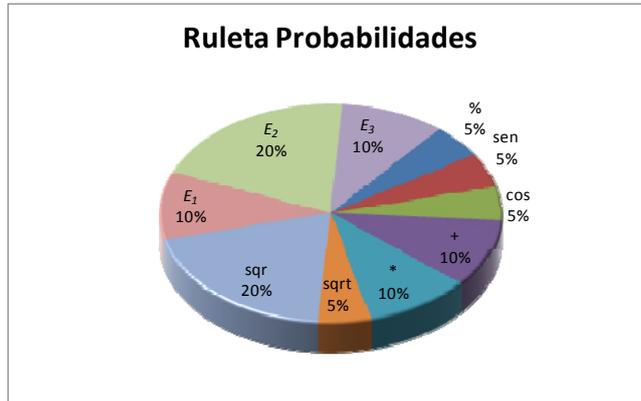


Figura 38. Ejemplo de ruleta de probabilidades

Tabla 7. Ejemplo de probabilidades de los operadores

Operador	Probabilidad
%	5%
sen	5%
cos	5%
+	10%
*	10%
sqrt	5%
sqr	20%
E_1	10%
E_2	20%
E_3	10%

Este método se traduce, en implementación, a la generación de una estructura con las probabilidades acumuladas de cada operador. Por ejemplo dada la Tabla 7 de asignación de probabilidades se puede trasladar a una estructura similar a la de la Figura 39. A continuación para obtener el operador es necesario generar un número aleatorio del intervalo [0,1]. Con el número

generado, por ejemplo el 0.5, se localiza el valor en la figura, obteniendo el operador *sqr*.

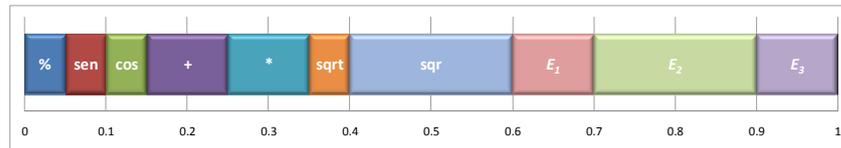


Figura 39 Elección mediante el método ruleta

2.2 Ejemplo de aplicación

Siguiendo con el ejemplo del alcance máximo del tiro parabólico de un proyectil presentado en el apartado 1.2 (página 93) se añade al módulo anterior (Módulo de operadores y funciones definidos por el usuario) las probabilidades de los operadores descritas en la Tabla 7. De igual manera se ha tomado la misma configuración descrita en el apartado Configuración de parámetros (página 95). A continuación se muestran los resultados obtenidos.

2.2.1 Resultados

Al igual que las pruebas descritas anteriormente se han realizado un total de 10 ejecuciones empleando la misma configuración que en el ejemplo del módulo anterior pero asignando, en este caso, las probabilidades de los operadores. En la Tabla 8 se muestra un resumen de las ejecuciones realizadas, cómo puede comprobarse en todas ellas se alcanza la solución exacta. Es importante señalar que para llegar a dicha solución por término medio solo han sido necesarias menos de 23 generaciones frente a las 160 requeridas en el caso de no emplear este módulo (sin tener en cuenta la ejecución 7 que no había convergido). Incluso el tiempo empleado para realizar las pruebas ha sido

inferior, ya que por término medio se han empleado (1' 31'') lo que representa cerca de 31 generaciones por minuto.

Tabla 8. PG con funciones definidas y probabilidad en operadores

Nombre	Gen.	EM (E)	ECM (E)	EM (T)	ECM (T)	Tiempo (s)	GPM
Ej.1	9	≈ 0	≈ 0	≈ 0	≈ 0	15	36.00
Ej.2	24	≈ 0	≈ 0	≈ 0	≈ 0	66	21.82
Ej.3	124	≈ 0	≈ 0	≈ 0	≈ 0	711	10.46
Ej.4	6	≈ 0	≈ 0	≈ 0	≈ 0	10	36.00
Ej.5	8	≈ 0	≈ 0	≈ 0	≈ 0	13	36.92
Ej.6	16	≈ 0	≈ 0	≈ 0	≈ 0	32	30.00
Ej.7	10	≈ 0	≈ 0	≈ 0	≈ 0	17	35.29
Ej.8	6	≈ 0	≈ 0	≈ 0	≈ 0	10	36.00
Ej.9	8	≈ 0	≈ 0	≈ 0	≈ 0	14	34.29
Ej.10	13	≈ 0	≈ 0	≈ 0	≈ 0	25	31.20
Promedio:	22.4	0	0	0	0	91.3	30.80

En la Figura 40 se muestra la evolución del mejor individuo de cada ejecución empleando el módulo de probabilidades junto con el de funciones definidas por el usuario.

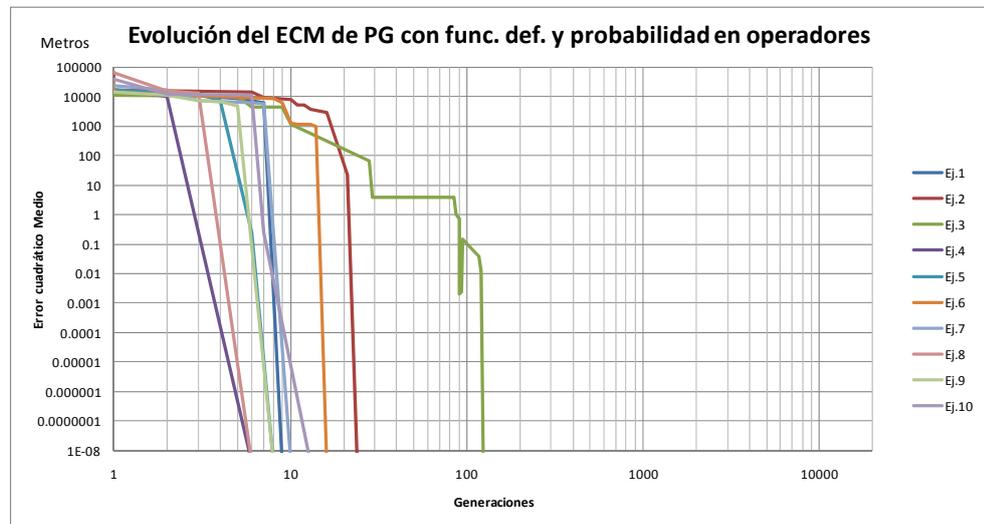


Figura 40 Evolución de las pruebas mediante PG con funciones definidas y probabilidad en los operadores

3 Módulo de coeficiente de seguridad

En ciertos fenómenos, cuando se intenta modelar el comportamiento mediante ecuaciones matemáticas, resulta de interés poder controlar el proceso de búsqueda, es decir, desviar la predicción hacia ciertas características y no sólo hacia el mejor resultado. A modo de ejemplo, para realizar el cálculo de un puente se emplea un modelo de cálculo, después de los cálculos, es imprescindible que el resultado final, la estructura construida, se sitúe dentro de unos márgenes de seguridad. En este caso al método de dimensionamiento se le aplican unos coeficientes de seguridad que permiten sobredimensionar la estructura garantizando de esta manera las restricciones impuestas. Ejemplos de este sobredimensionamiento es ponderar al alza la carga máxima que debe soportar, o también minusvalorar la resistencia de los materiales.

Es por ello la importancia de introducir un mecanismo que permita incorporar mecanismos de seguridad a la hora de buscar una expresión que permita predecir el comportamiento de un proceso. En esencia, a partir de los datos experimentales prima la seguridad en las fórmulas frente a la precisión de la predicción. Volviendo al ejemplo anterior, si se desea predecir la resistencia que podrá soportar una viga, teniendo en cuenta los factores en su construcción, resulta fundamental que dicha predicción no sea superior al valor real, pues de lo contrario, la viga podría llegar a colapso, suponiendo una catástrofe desde el punto de vista ingenieril. Surge de estos motivos la necesidad de incorporar a la PG un módulo de coeficientes de seguridad.

El proceso de búsqueda de expresiones es dinámico, de modo que a la vez que se evalúan las expresiones obtenidas, se penalizan aquellas que producen valores inseguros en función del criterio establecido, consiguiendo de esta manera obtener expresiones desviadas hacia la seguridad.

Este método se basa en introducir la penalización a la función de ajuste pura, ecuación (2) descrita en Fundamentos, capítulo III apartado 1.2.3.1 (página 37), asociado al factor de parsimonia, ecuación (6) página 40. La función resultante se puede ver en la ecuación (30) en la que x_i corresponde al valor predicho por la fórmula genética, d_i es el valor real, n es el número de patrones, α es el nivel de parsimonia y s_i representa la complejidad de la función empleando para ello el tamaño, es decir el número de nodos del individuo, p_i es la penalización debido a la seguridad.

$$fitness(i) = \frac{\sum_{i=0}^n p_i * |x_i - d_i|}{n} + \alpha * s_i \tag{30}$$

$$p_i = \begin{cases} k_1 & \text{si } x_i > d_i \\ k_2 & \text{si } x_i \leq d_i \end{cases}$$

El método propuesto permite configurar la penalización p_i por exceso o por defecto, es decir, permite sobredimensionar (cuando k_1 se establece en 1 y k_2 a un valor mayor que 1) o sub-dimensionar los resultados obtenidos (k_2 establecido en 1 y k_1 a valor mayor que 1). Tomando como ejemplo la fuerza necesaria que hay que aplicar a una estructura para lograr su colapso, si el valor de la predicción (x_i) es superior al valor real del ensayo (d_i) provocará, en global, un peor ajuste, con lo que ese individuo con alta probabilidad será desechado en detrimento de otro con peor predicción pero dentro del lado de la seguridad. Por ejemplo, en la Tabla 9 se muestra el cálculo del ajuste para dos individuos. Para simplificar el ejemplo se obvia el factor de parsimonia y se establece como penalización de seguridad p_i los valores $k_1=2$ y $k_2=1$ (penalización por sobredimensionamiento). En este caso, teniendo en cuenta el EM el individuo 1 sería mejor que el individuo 2 pero obtiene un peor ajuste ya que en dos ocasiones las predicciones son inseguras.

Tabla 9. Ejemplo de funcionamiento

d_i	x_i I1	x_i I2	EM I1	EM I2	p_i I1	p_i I2	Ajuste I1	Ajuste I1
110	105	105	5	5	1	1	5	5
120	120	115	0	5	1	1	0	5
130	135	120	5	10	10	1	50	10
90	92	85	2	5	4	1	8	5
EM:			3	6.25	Ajuste:		15.75	6.25

Es posible un refinamiento del método de penalización, se pueden construir distintos intervalos de penalización, de forma que cuanto más error se produzca mayor será la penalización. De igual manera se puede añadir el número de intervalos deseados, el rango en el que se aplica la penalización, y el valor de dicha penalización. En la ecuación (31) se muestran 3 intervalos determinados por los valores g_1 y g_2 con las penalizaciones de k_1 , k_2 y k_3 .

$$fitness(i) = \frac{\sum_{i=0}^n p_i * |x_i - d_i|}{n} + \alpha * s_i \quad (31)$$

$$p_i = \begin{cases} k_1, & x_i - d_i < g_1 \\ k_2, & g_1 \leq x_i - d_i < g_2 \\ k_3, & x_i - d_i \geq g_2 \end{cases}$$

En ocasiones, debido al carácter experimental de los ensayos, a misma configuración de parámetros se obtiene una variabilidad en los resultados. Por esta razón, se incorpora un mecanismo corrector que permite paliar esta variabilidad de los resultados. Este factor de corrección está asociado a la dispersión de los resultados experimentales ante un mismo ensayo (las mismas características de ensayo), ya que para un determinado tipo de ensayos interviene una gran cantidad de parámetros en el fenómeno físico. Por este motivo, las predicciones que se realicen en el modelado del fenómeno deben tener en cuenta esta variabilidad y obtener un resultado seguro para la mayoría de los casos. Para solventar esta variabilidad, en la ecuación (32) se añade el factor “ I_{bias} ” que corresponde con el esviate (bias) que se producen en el conjunto de datos. Además, en esta variante del algoritmo, se cambia la forma de calcular el error cometido, ya que en vez de restar el valor del ensayo (d_i) al valor de la predicción (x_i), se calcula el cociente. El resultado del cociente será 1 cuando la predicción sea exacta, mayor que 1 cuando la predicción sea inferior al valor real y menor que 1 cuando sea lo contrario. Dada esta

configuración, para poder minimizar el error, el resultado del cociente se debe restar a 1 ya que así se consigue obtener la predicción exacta.

$$fitness(i) = \frac{\sum_{i=0}^n p_i * \left| l_{bias} - \frac{d_i}{x_i} \right|}{n} + \alpha * s_i$$

$$p_i = \begin{cases} k_1, & \frac{d_i}{x_i} < g_1 \\ k_2, & g_1 \leq \frac{d_i}{x_i} < g_2 \\ k_3, & \frac{d_i}{x_i} \geq g_2 \end{cases} \quad (32)$$

Por ejemplo, para evitar la dispersión de los datos, se podría situar el valor objetivo (l_{bias}) del cociente entre el valor del ensayo (d_i) y el valor de la predicción (x_i) al de 1.1 frente al 1.0 (que como se ha visto corresponde con la predicción exacta). El modelo resultante se situará ligeramente en el lado de la seguridad (el valor de la fórmula obtenida será inferior al valor del ensayo) obteniendo de esta manera una formulación conservadora, útil para su aplicación en códigos técnicos de ingeniería.

3.1 Ejemplo de aplicación

De igual manera que en los módulos anteriores, se basa el ejemplo de funcionamiento de este módulo, en el cálculo del alcance máximo del tiro parabólico de un proyectil (página 93). Imagínese que se necesita diseñar las dimensiones de un campamento protegido por una torre de combate. En este caso será necesario determinar el alcance máximo de los proyectiles para establecer en qué zonas se puede construir el campamento. Supóngase que se

desconoce la ecuación (21) por lo que se diseña una campaña experimental para intentar determinarlo (conjunto de datos descrito en el apartado 1.2.1 de este capítulo). A efectos de diseño, obviamente resulta perjudicial que la predicción sea superior al valor real. Ya que de esta manera el campamento construido dentro de la distancia x_1 estaría en zona insegura (ver Figura 41.1). Es decir, es preferible que las predicciones sean conservadoras (menos distancia) ya que así en la realidad el campamento estará en el lado de la seguridad (dentro de la distancia x_2), como puede observarse en la Figura 41.2

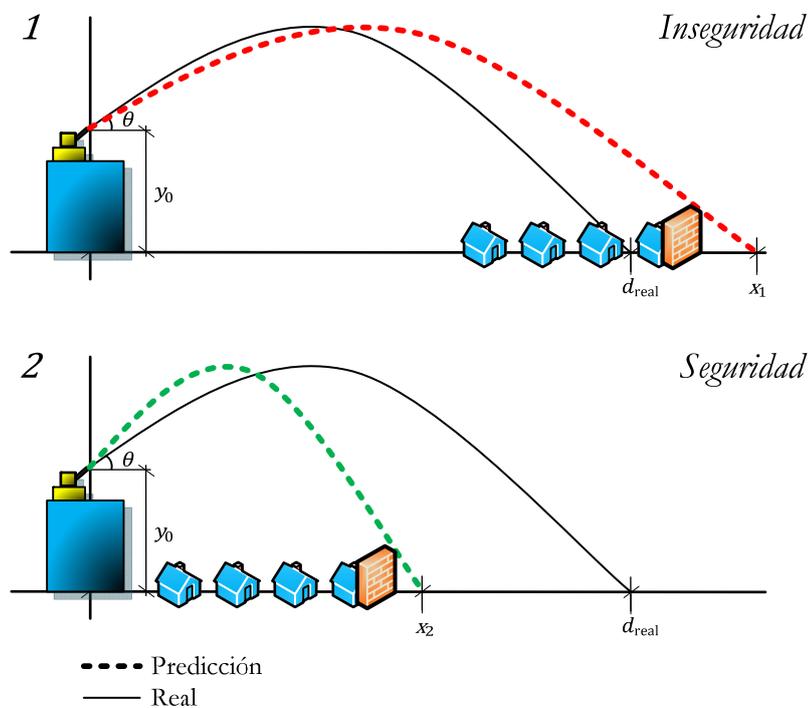


Figura 41. Importancia del uso de los coeficientes de seguridad

En este caso se configuraron los intervalos de penalización como muestra la ecuación (33). El resto de parámetros de la configuración del algoritmo de PG es la misma que en los ejemplos anteriores, en concreto la misma que la empleada en la PG clásica, con la excepción de la parsimonia. Esto se debe a

que al cambiar el método de ajuste, se produce un cambio de magnitud. Así pues se estableció la parsimonia a 0.001

$$p_i = \begin{cases} 10, & \frac{d_i}{x_i} < 0.5 \\ 5, & 0.5 \leq \frac{d_i}{x_i} < 1 \\ 1, & 1 \leq \frac{d_i}{x_i} < 1.25 \\ 2, & 1.25 \leq \frac{d_i}{x_i} < 1.5 \\ 3, & \frac{d_i}{x_i} \geq 1.5 \end{cases} \quad (33)$$

Tabla 10. PG Clásica con el módulo de coeficientes de seguridad

Nombre	Gen.	EM (E)	ECM (E)	EM (T)	ECM (T)	Tiempo (s)	GPM	PD
Ej.1	20000	113.65	32909	119.98	33370	6292	190.7	800
Ej.2	20000	33.79	4286	38.02	5067	5473	219.3	348
Ej.3	20000	75.06	11818	76.07	12125	5160	232.6	511
Ej.4	20000	56.50	9147	58.90	8859	5441	220.5	887
Ej.5	20000	91.09	19448	98.99	25561	5522	217.3	871
Ej.6	20000	106.13	27125	113.75	28529	4875	246.2	773
Ej.7	20000	65.46	9255	66.75	9802	5446	220.3	587
Ej.8	20000	59.64	8670	61.02	9002	5972	200.9	212
Ej.9	20000	50.28	6237	50.17	6001	5513	217.7	400
Ej.10	20000	90.33	18455	97.21	23266	4359	275.3	708
Promedio:	20000	74.19	14735	78.08	16158	5405.3	224.1	609.7

En la Tabla 10 se muestra los resultados del conjunto de pruebas realizadas. Al igual que en caso de la PG clásica, ninguna de las ejecuciones ha conseguido obtener un ajuste perfecto. De hecho los errores empeoran debido al desplazamiento que imprime los coeficientes de seguridad. Para determinar el mejor individuo del conjunto de 10 ejecuciones, además de fijarse en el EM y

ECM se emplea la técnica denominada “puntos de demérito” (PD) de forma similar a la que ha empleado Collins para categorizar sus predicciones (Collins, 2001). En la ecuación (34) se muestra cómo calcularlo, siendo n el número de patrones, d_i el valor real y x_i el valor de predicción.

$$PD(\text{individuo}) = \sum_{i=0}^n k_i, \quad k_i = \begin{cases} 10, & \frac{d_i}{x_i} < 0.5 \\ 5, & 0.5 \leq \frac{d_i}{x_i} < 1 \\ 0, & 1 \leq \frac{d_i}{x_i} < 1.25 \\ 2, & 1.25 \leq \frac{d_i}{x_i} < 1.5 \\ 3, & \frac{d_i}{x_i} \geq 1.5 \end{cases} \quad (34)$$

Teniendo en cuenta los errores y los PD se determina que el individuo, representado en la ecuación (35), obtenido de la ejecución 2 es el más seguro de los obtenidos ya que cometiendo un EM 33.79 metros y ECM 4286 metros consigue una puntuación de 384 en demérito, en cambio aunque el individuo de la ejecución 8 obtiene una puntuación menor de punto demérito (212) los errores que comete son superiores (EM 59.64 y ECM 8670).

$$PG_{CS}(Ej. 2) = \frac{v_0^2 * \sin(2 * \theta)}{9.81} + \frac{y_0 * \sin(2 * \theta)}{2 * \theta} \quad (35)$$

Tabla 11. Puntos de demérito en PG Clásica (entrenamiento)

Nombre	Ej.1	Ej.2	Ej.3	Ej.4	Ej.5	Ej.6	Ej.7	Ej.8	Ej.9	Ej.10	Promedio
PD	900	638	1098	722	1023	1115	900	1195	1169	1065	982.5

Para poder comparar este módulo se muestran en la Tabla 11 los puntos de demérito de las ejecuciones resultantes de la PG clásica (presentadas en la Tabla 5), de este modo se puede comparar, resultando que en término medio

la PG clásica obtiene una puntuación de 982.5 frente a los aproximadamente 610 puntos obtenidos mediante el método basado en los coeficientes de seguridad.

El tiempo medio empleado es similar con ambas técnicas con una variación de 10 minutos a favor de la explicada en este módulo. Para seleccionar el mejor individuo de la PG clásica, en términos de seguridad, se ha empleado el mismo procedimiento resultando como representante la ejecución 2. En la Figura 42 se muestra la cantidad de patrones distribuidos en los intervalos, en este caso el de PG clásica (Ej.2) obtiene una predicción muy mala (< 0.5) y 115 malas ($\geq 0.5, < 1$), en total 116 predicciones inseguras frente a las 53 del PG con coeficiente de seguridad.

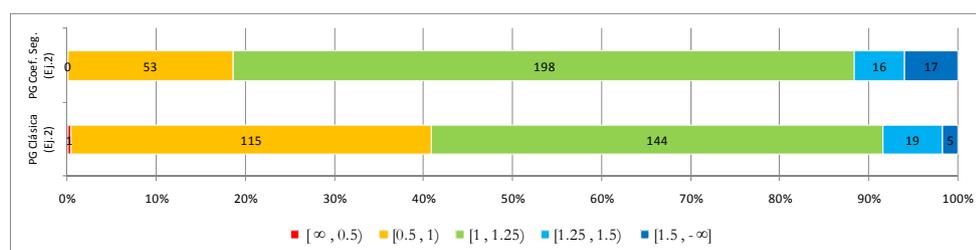


Figura 42. Mejores individuos PG Clásica y PG con Coeficiente de Seguridad

De forma gráfica se muestran las correlaciones de los dos individuos en la Figura 43. Como puede observarse la zona sombreada en rojo representa las predicciones inseguras. A simple vista se puede concluir que con el método de coeficiente de seguridad la gran mayoría de las predicciones son seguras.

En la Figura 44 se muestra la evolución del mejor individuo de cada ejecución empleando el módulo de coeficientes de seguridad. Como puede observarse, el error cometido por los individuos oscila debido a que la función de ajuste no

está basada en el ECM. En la Figura 45 se muestra la evolución del ajuste y el ECM de la ejecución 2.

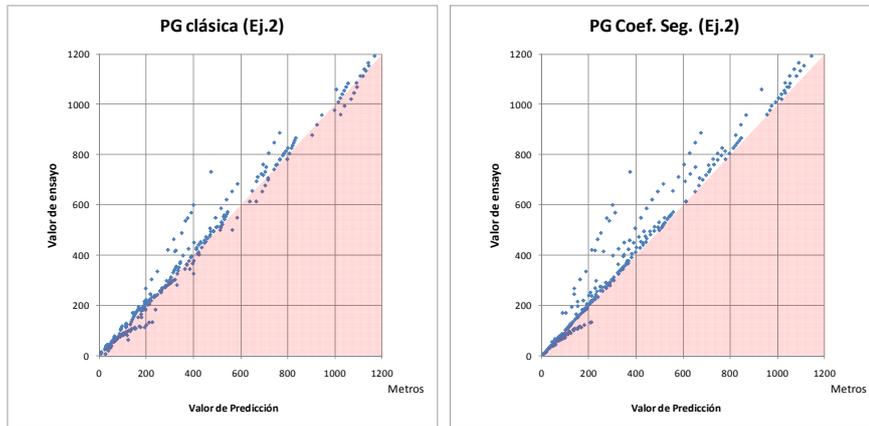


Figura 43. Correlación de los mejores individuos mediante PG Clásica y PG con coeficiente de seguridad

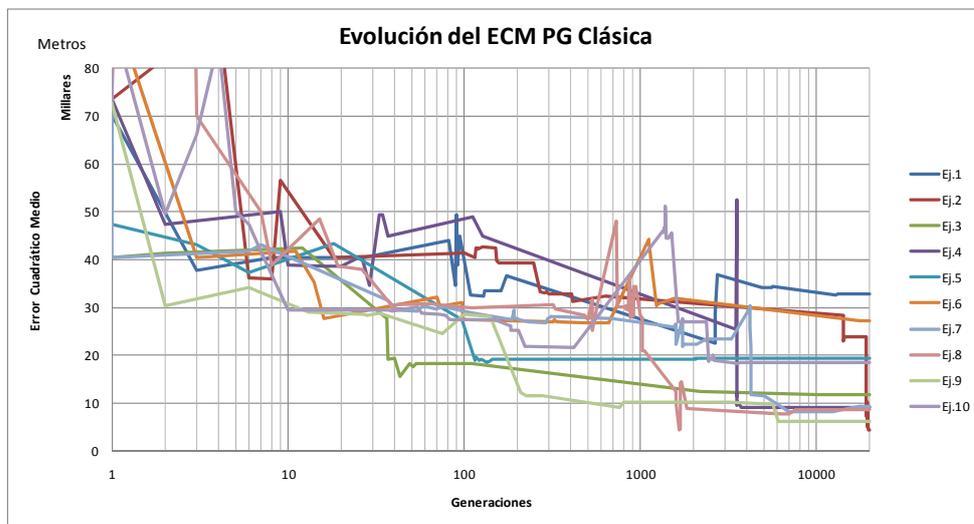


Figura 44 Evolución del ECM mediante PG con coeficiente de seguridad

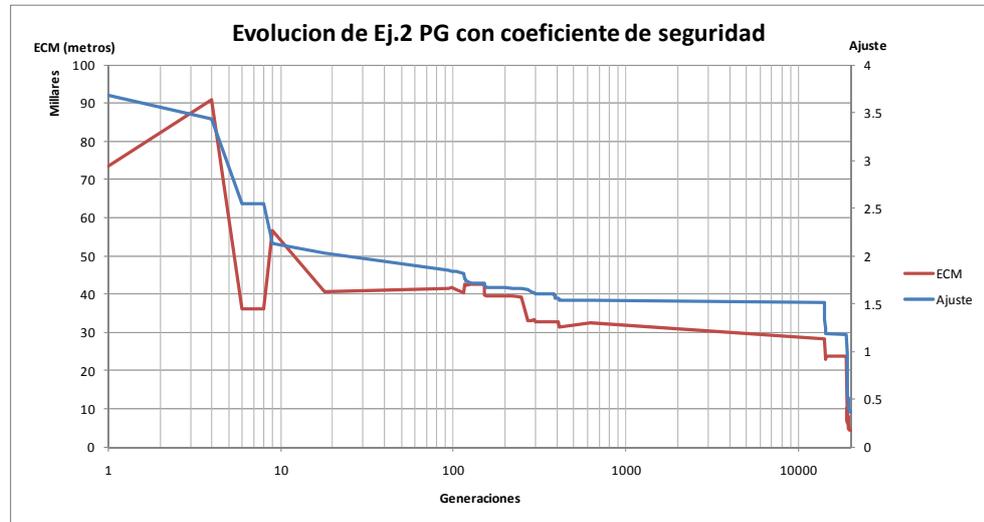


Figura 45 Evolución del ajuste y ECM de la ejecución 2

4 Módulo de restricciones en funciones

En numerosas ocasiones, cuando se aplica directamente la PG sobre datos experimentales con la intención de obtener una fórmula matemática que explique el comportamiento del fenómeno a estudiar, se obtienen expresiones demasiado complejas. El ejemplo más significativo sería cuando en una expresión aparece n veces la misma variable. Por norma general estas formulaciones no son prácticas para incluirlas en códigos y normativas técnicas en el mundo de la Ingeniería Civil, ya que hacen más complicada su interpretación.

Por esta razón se ha desarrollado este módulo que permite definir las restricciones siguientes:

- número máximo de repeticiones permitidas de una variable dentro de una expresión. A partir de una lista de pares (variable, número repeticiones) se desechará toda solución que no cumpla con la restricción. De esta manera se puede obligar a que en una solución no aparezca una determinada variable, o que únicamente emplee valores constantes, etc...
- número de veces que puede aparecer un determinado operador: de igual manera que el caso anterior.

Este módulo cobra más sentido, si cabe, unido al módulo siguiente, ya que permite definir un conjunto de restricciones diferentes para cada una de las sub-ramas definidas.

5 Módulo de optimización de funciones conocidas (normativas)

En un cierto número de ocasiones el empleo de una técnica determinada es válida para la resolución de un problema propuesto, pero a menudo se necesita realizar ajustes, o combinaciones de técnicas para solventar con éxito determinados planteamientos. La Computación Evolutiva, como se ha comentado anteriormente, por sí sola es capaz de resolver multitud de cuestiones en diversos ámbitos, como por ejemplo en la Ingeniería Civil. En este ámbito de estudio aparecen diversos problemas donde las técnicas de Inteligencia Artificial pueden ayudar al enriquecimiento de la ciencia. En la gran mayoría de los problemas se intenta abstraer un fenómeno físico, complejo y con gran cantidad de variables, en un modelo matemático para simular y predecir dicho fenómeno. En la mayoría de casos de estudio ya

existen modelos matemáticos sobre el fenómeno, es decir, existen diversos modelos que permiten ajustar el comportamiento físico/químico mediante ecuaciones. En este contexto resulta de gran interés el empleo de técnicas de Inteligencia Artificial para la optimización o mejora, si cabe, de dichos modelos. Esta es la motivación para el desarrollo de este módulo específico que permite, mediante técnicas de PG, la optimización de fórmulas matemáticas a partir de los datos experimentales (patrones de aprendizaje). Es importante hacer hincapié que este módulo está fuertemente relacionado con el módulo anterior, ya que permite añadir el conocimiento del experto mediante restricciones y mediante la inducción de las ecuaciones matemáticas antes mencionadas.

El módulo permite la optimización de cualquier expresión matemática en los puntos que seleccione el experto. Dada una ecuación f en donde intervienen n variables, se pretende optimizarla mediante un conjunto de datos experimentales. Este conjunto de datos puede contener valores para más variables que las que intervienen en la función. Por ejemplo, en la ecuación (36) intervienen las variables a , b y c , en cambio, en el conjunto de datos experimentales hay valores para las variables a , b , c , d , y e . Es decir, es posible que además de las variables a , b y c que están en la función, se empleen en la optimización las variables d y e . El experto para orientar la solución selecciona qué puntos de la ecuación desea optimizar, en este caso 4 puntos (valores entre corchetes de la ecuación: 2, $\frac{1}{2}$, $a+c$ y 5)

$$f(a, b, c) = [2] * \left(\frac{1}{a}\right)^{[1/2]} * \left(\frac{b}{[a+c]} * c\right)^{[5]} \quad (36)$$

Para optimizar esta expresión con PG es necesario utilizar una codificación que permita desarrollar las 4 ramas que se quieren optimizar. La solución a

este problema vendrá determinada mediante un árbol que en este caso contendrá 4 ramas (Figura 46), donde cada rama representa la parte de la fórmula que se sustituirá en la ecuación original.

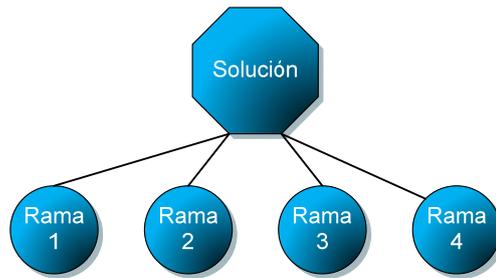


Figura 46. Ejemplo de solución

Para la adecuación del algoritmo de PG al problema propuesto es necesario definir cómo van a ser las soluciones. Es decir, la definición del genotipo de la solución. Para ello se parte de la función objetivo a optimizar. En la ecuación (37) se muestra el genotipo resultante de la ecuación (36) que se empleará para evaluar los individuos. Se representa como $rama_i$ cada una de las ramas de la solución obtenida y el conjunto de variables (a, b, c) se obtienen del conjunto de datos.

$$rama_1 * \left(\frac{1}{a}\right)^{rama_2} * \left(\frac{b}{rama_3} * c\right)^{rama_4} \quad (37)$$

Una vez definida la estructura entra en juego el módulo de restricciones, definido anteriormente, y personalizado para cada una de las ramas. De esta manera, se podría obligar a que en una rama se impida el uso de una determinada variable, o que sólo se pueda utilizar un subconjunto de funciones (suma y resta). En este caso, si al crear, mutar o cruzar un individuo no cumplierse las restricciones en cada una de las ramas dadas por el experto,

ese individuo se “poda” (eliminación de modos y ramas) hasta que cumpla todas las restricciones.

Por ejemplo, en una rama donde la restricción dada impide la aparición de una variable, en el caso de que tras una operación de cruce o mutación apareciese una variable restringida, ésta se sustituye por un valor constante. En la Figura 47 se muestra un ejemplo de individuo válido que optimiza la ecuación (37) que cumple las restricciones enumeradas en la Tabla 12 (la *P* significa valor permitido). En la ecuación (38) se muestra el resultado final, una vez sustituida cada rama.

$$a * b^2 * \left(\frac{1}{a}\right)^3 * \left(\frac{b}{d + c * 6} * c\right)^{e + \sqrt{a} * 2} \quad (38)$$

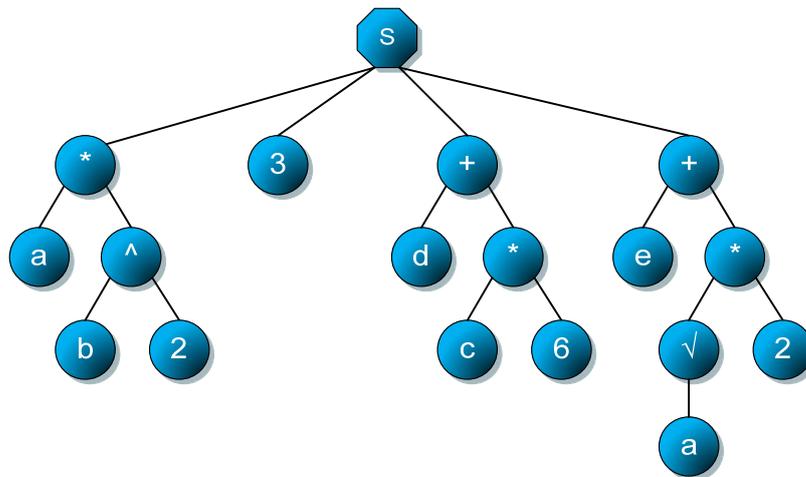


Figura 47. Ejemplo de individuo válido

Tabla 12. Ejemplo de restricciones a ramas

Restricción	a	b	c	d	e	Constantes
Rama 1	P	P	P			P
Rama 2						P
Rama 3			P	P		P
Rama 4	P	P			P	P

5.1 Ejemplo de aplicación

Para comentar las bondades del módulo desarrollado se vuelve a emplear el mismo ejemplo de alcance máximo en tiro parabólico ya empleado en los módulos anteriores (presentado en la página 93). Supóngase que debido a la experiencia, se conoce que la ecuación (39) ayuda a resolver el problema planteado, con esta información se podría resolver parcialmente el problema planteado. En este caso el experto establece dos puntos marcados como *rama₁* y *rama₂* en donde se supone que la PG ayudará a mejorar la expresión. Este esqueleto de función se empleará dentro de la función de ajuste del método desarrollado. Para ello se evalúa el valor de cada rama y se sustituye en la ecuación (40) y se evalúa la ecuación conjunta. Una vez obtenido el valor se calcula el error cometido con cualquier método disponible, por ejemplo el promedio del error cometido por cada patrón.

$$d = \frac{1}{g} (v_0 * \sin(\theta) + \sqrt{2 * g * y_0}) \quad (39)$$

$$d = \frac{rama_1}{g} (v_0 * \sin(\theta) + \sqrt{rama_2 + 2 * g * y_0}) \quad (40)$$

En cuanto a la configuración empleada se ha utilizado la misma que en casos anteriores, es decir como operadores disponibles se establecen la suma,

producto, división protegida, raíz cuadrada, cuadrado, seno y coseno. Para los nodos terminales se pueden emplear las variables procedentes del conjunto de datos $(\theta, v_0 \text{ e } y_0)$ y la constante de la aceleración gravitacional (9.81). De igual manera se establece como parámetros de configuración de la PG la misma que en casos anteriores (Tabla 4, página 96). En la ejecución de este ejemplo no se ha establecido ninguna restricción en el uso de operadores y variables en cada rama.

5.1.1 Resultados

De igual manera que en los ejemplos anteriores se han realizado 10 ejecuciones empleando el módulo aquí presentado. En la Tabla 13 se muestran los resultados. Como puede observarse, todas las ejecuciones a excepción de la ejecución 5 (41) han logrado obtener la solución (21), pero no con el mismo ajuste, ya que las soluciones encontradas son equivalentes pero no iguales a nivel de la PG. Por ejemplo, en la *rama₂* para obtener el cuadrado de un valor en ocasiones se emplea el operador *sqr(valor)* y en otras simplemente se multiplica por sí mismo el propio valor (*valor*valor*) con lo cual aunque el resultado sea el mismo, el ajuste asociado al individuo es diferente, por el valor de la parsimonia, ya que el tamaño del árbol para representar los individuos es diferentes.

$$\begin{aligned}
 PG_{fd}(Ej. 5) &= \frac{rama_1}{g} (v_0 \text{ sen}(\theta) + \sqrt{rama_2 + 2gy_0}) \\
 rama_1 &= \text{cos}(\theta) * v_0 \\
 rama_2 &= \theta^2 * \sqrt{\text{cos}(\theta)} * (v_0 + \text{sen}(v_0))^2
 \end{aligned}
 \tag{41}$$

Aun en el caso (ejecución 5) en la que no se alcanzó la solución óptima, los errores cometidos (EM 1.89 y ECM 12) son más pequeños que los obtenidos con el mejor individuo de la PG clásica (EM 26.39 y ECM 1263).

Tabla 13. PG con optimización de funciones conocidas

Nombre	Gen.	EM(E)	ECM(E)	EM(T)	ECM(T)	Tiempo(s)	GPM
Ej.1	7665	≈ 0	≈ 0	≈ 0	≈ 0	4481	102.6
Ej.2	43	≈ 0	≈ 0	≈ 0	≈ 0	6	430.0
Ej.3	2026	≈ 0	≈ 0	≈ 0	≈ 0	1219	99.7
Ej.4	12210	≈ 0	≈ 0	≈ 0	≈ 0	4662	157.1
Ej.5	20000	1.89	12	2.24	17	9181	130.7
Ej.6	1911	≈ 0	≈ 0	≈ 0	≈ 0	1096	104.6
Ej.7	3815	≈ 0	≈ 0	≈ 0	≈ 0	1563	146.4
Ej.8	1467	≈ 0	≈ 0	≈ 0	≈ 0	804	109.5
Ej.9	6098	≈ 0	≈ 0	≈ 0	≈ 0	3197	114.4
Ej.10	11141	≈ 0	≈ 0	≈ 0	≈ 0	6723	99.4
Promedio:	6637.6	0.19	1	0.22	2	3293.2	149.5

La funcionalidad que aporta este módulo tiene un coste computacional más elevado que el de la PG clásica. Por término medio, en un minuto, con la configuración anteriormente explicada, se realizan cerca de 150 generaciones frente a las casi 205 mediante PG clásica. Cabe resaltar la disparidad entre el número necesario de generaciones para alcanzar la solución, en el mejor de los casos con tan sólo 43 generaciones (y en sólo 6 segundos), en cambio, en el peor a excepción de la ejecución 5 que no ha convergido, se necesitaron 12210 generaciones (1 h 17' 42"). Esta situación se podría haber mejorado estableciendo otra configuración, por ejemplo, aumentando el factor de parsimonia con lo que se alcanzaría la solución en menos generaciones.

De forma gráfica se puede ver la evolución del ECM de las ejecuciones en la Figura 48 y en Figura 49 un detalle de la misma en cercanía de los valores mínimos. Al igual que en los casos anteriores, en las proximidades de encontrar la solución exacta aparecen variaciones en el ECM provocados por la influencia del factor de parsimonia.

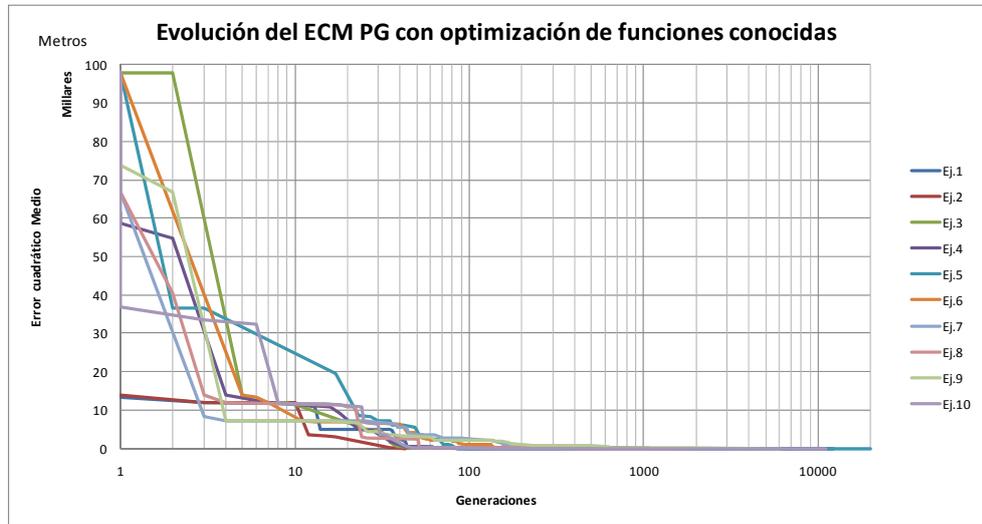


Figura 48 Evolución de las pruebas mediante PG con optimización de funciones conocidas

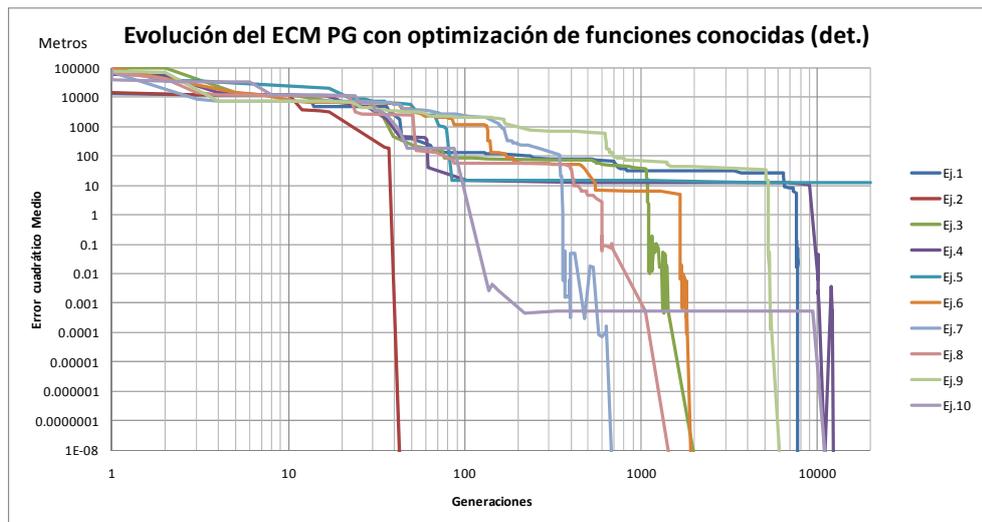


Figura 49 Evolución de las pruebas mediante PG con optimización de funciones conocidas (detalle)

Tas la realización de estas pruebas iniciales para cada uno de los módulos desarrollados, se puede concluir, que todos los módulos aceleran la búsqueda

de la solución, resolviendo en casi la totalidad de los casos el problema planteado. Incluso, en los casos que no ha conseguido resolverlo los errores cometidos siempre han sido inferiores comparados con los cometidos por la PG clásica. Desde el punto de vista de los Ingenieros Civiles, el desarrollo del módulo de seguridad presenta un avance frente a la PG clásica.

VII. Resultados y Discusión

A continuación se muestra la aplicabilidad práctica del sistema desarrollado en tres ejemplos reales de predicción en problemas enmarcados dentro del campo específico del Hormigón Estructural. En primer lugar se afronta la predicción del esfuerzo a cortante que sufren las vigas de hormigón en determinadas condiciones. En este caso se parte de una ecuación conocida, procedente de normativas vigentes en el ámbito y mediante restricciones se consigue mejorar las predicciones obtenidas. En el segundo ejemplo se desea predecir el comportamiento entre la unión de las barras de acero con el hormigón determinando la tensión máxima que es capaz de alcanzar. En este caso se parte de una fórmula ya establecida en la que se modifica su estructura y se establecen diversas ramas para lograr un conjunto de expresiones, que además de obtener mejores resultados son más seguras a nivel estructural que las

normativas vigentes. El último ejemplo, se basa en determinar el comportamiento del hormigón a lo largo del tiempo, en concreto la propiedad que se desea predecir es la deformación producida por la fluencia que sufre el hormigón cuando es sometido a cargas constantes a lo largo del tiempo. En este problema influye de forma determinante el instante de tiempo en el que se realiza la predicción, constituyendo un problema con comportamiento dinámico.

Con la aplicación de estos tres ejemplos, del mismo ámbito, pero de distinta complejidad y característica se muestra las capacidades y versatilidad de la metodología y su implementación planteada en esta tesis.

1 Cortante

1.1 Descripción

El trabajo de las vigas de hormigón sometidas a cargas externas suele vincular un doble trabajo interno, pudiéndose así distinguir entre el comportamiento a flexión y a cortante. El trabajo a flexión se esquematiza de modo coherente según la Figura 50, que ilustra el trabajo de cada sección de viga con el par de tensiones Compresión/Tracción. La incapacidad del hormigón para resistir tracciones motiva el armado habitual de estas piezas.

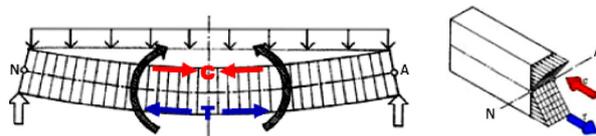


Figura 50. Flexión.

El trabajo a cortante también se desarrolla, evitando éste el que la viga se deforme por tensiones de corte de acuerdo a la Figura 51.

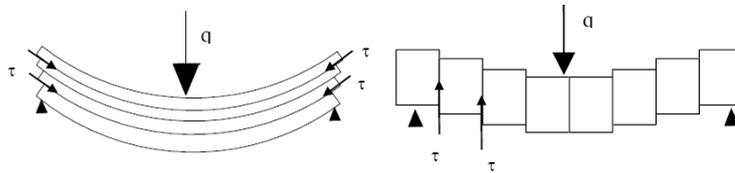


Figura 51. Deformación por tensiones de corte.

Las tensiones de corte tienden a distorsionar un elemento diferencial ideal (Figura 52), siendo su acción equivalente a la que realizarían una compresión y una tracción actuando bajo las direcciones llamadas principales (Figura 53). Tensiones de flexión y cortante se combinan en cada uno de los puntos de la viga, generando una red de tensiones principales esquematizada en la Figura 54, antesala del modelo general de armado de una viga. En efecto, en ésta se suelen disponer armaduras longitudinales y transversales en forma de jaulas de armado.

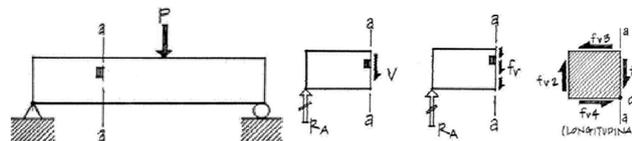


Figura 52. Tensiones en un elemento diferencial ideal.

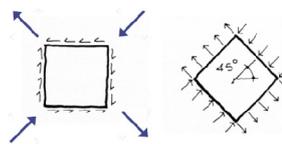


Figura 53. Tensiones de corte principales.

El comportamiento de las piezas estructurales de hormigón armado sometidas a esfuerzos cortantes es más complejo que el que se produce bajo

solicitaciones normales. La resistencia a compresión y a tracción del hormigón simple, la presencia de armadura longitudinal, la proximidad de cargas concentradas y la concomitancia de esfuerzos normales, son algunos de los factores que definen los mecanismos que se desarrollan dentro de los elementos estructurales para resistir las fuerzas cortantes. La teoría de cortante en vigas, desarrollada para materiales homogéneos, isótropos y elásticos, puede ser utilizada como punto de partida, pero debe ser modificada para tener en cuenta el resto de factores involucrados.

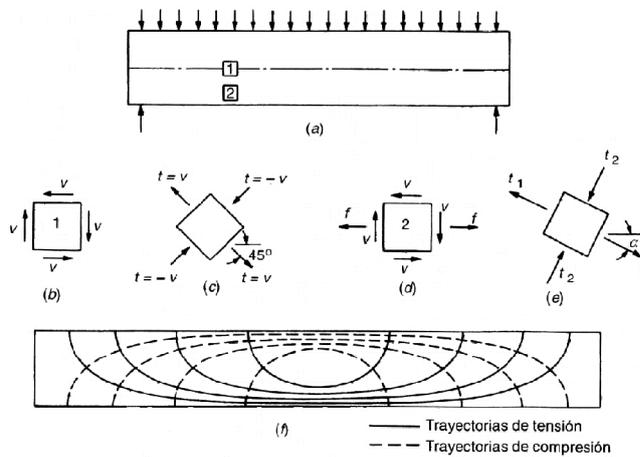


Figura 54. Red de isostáticas de una viga.

El trabajo a cortante, no obstante, no requiere siempre la presencia de armadura transversal. Otros mecanismos resistentes se dan cita y pueden ser suficientes si las tensiones se encuentran acotadas. Entre esos mecanismos se encuentran los siguientes: las tensiones de corte (o tangenciales) resistidas en el hormigón no fisurado, el mecanismo de cortante – fricción en las fisuras (provocado por el engranamiento entre áridos), el efecto dovela aportado por la armadura longitudinal (que evita la deformación en “gajos” visto en Figura 51), el efecto arco (especialmente en vigas con cantos considerables) y las tensiones de tracción residuales (ASCE-ACI Committee 426, 1973; ASCE-

ACI Committee 445, 1998). La Figura 55 resume los principales efectos. Estos diferentes mecanismos se traducen de forma simplificada en las fuerzas internas desarrolladas en una viga fisurada. En la Figura 56, y más en detalle en la Figura 57, se muestra las fisuras resultantes en una viga, en la que en la zona izquierda hay ausencia de armadura transversal (González, 2002).

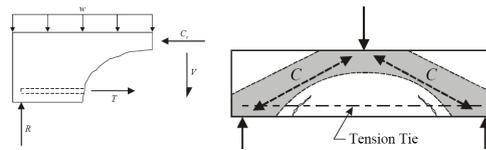


Figura 55. Principales efectos del esfuerzo cortante.

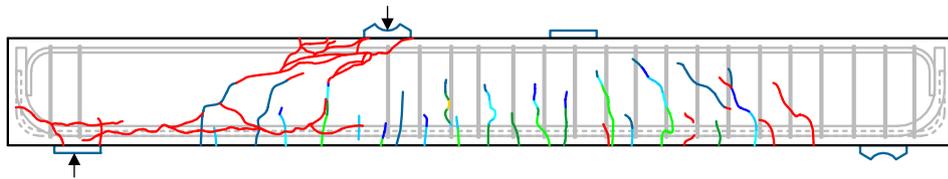


Figura 56. Fuerzas internas en una viga de hormigón fisurada sin armadura a cortante



Figura 57. Ejemplo de rotura por falta de armadura transversal.

1.1.1 Conjunto de datos

Para la realización de las pruebas para el estudio del fenómeno a cortante se ha empleado principalmente una base de datos recopilada por Collins et al (Collins, Bentz, Sherwood & Xie, 2008). Esta base de datos incluye los resultados experimentales de 1149 vigas cuya relación a/d es mayor o igual a

2.5 y que las roturas de las vigas se han identificado como roturas a cortante. Del conjunto de datos se han seleccionado 5 variables (4 entradas – 1 salida). En la Figura 58 se muestran gráficamente alguno de los parámetros. El ancho de la viga, b , el canto útil, d , la resistencia a compresión del hormigón, f_c , y la cuantía geométrica de la armadura longitudinal traccionada ($\rho_{lw} = A_s/b*d$).

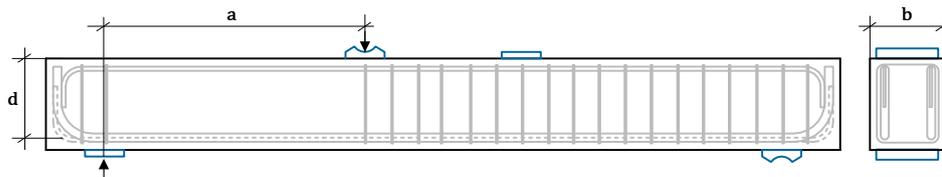


Figura 58. Parámetros de la viga.

Para realizar las pruebas se ha dividido en dos conjuntos. Para el entrenamiento se han empleado 518 casos. Para el test se han dejado los 631 casos restantes. Se ha elegido esta división debido a exigencias del experto. En la Tabla 14 se muestran las características de los conjuntos de datos empleados. En la Figura 59 se muestra, gráficamente, la distribución del conjunto total de datos.

Tabla 14. Conjunto datos empleadas

Conjuntos de datos	Cantidad	b (mm)		d (mm)		f _c (MPa)		ρ _{lw} (%)		V _u (kN)	
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
Entrenamiento	518	21	3000	41	3000	6.1	127.5	0.14	9.5	1.9	1575.00
Test	631	21	1829	41	2000	6.3	127.5	0.17	9.42	2.1	789.00

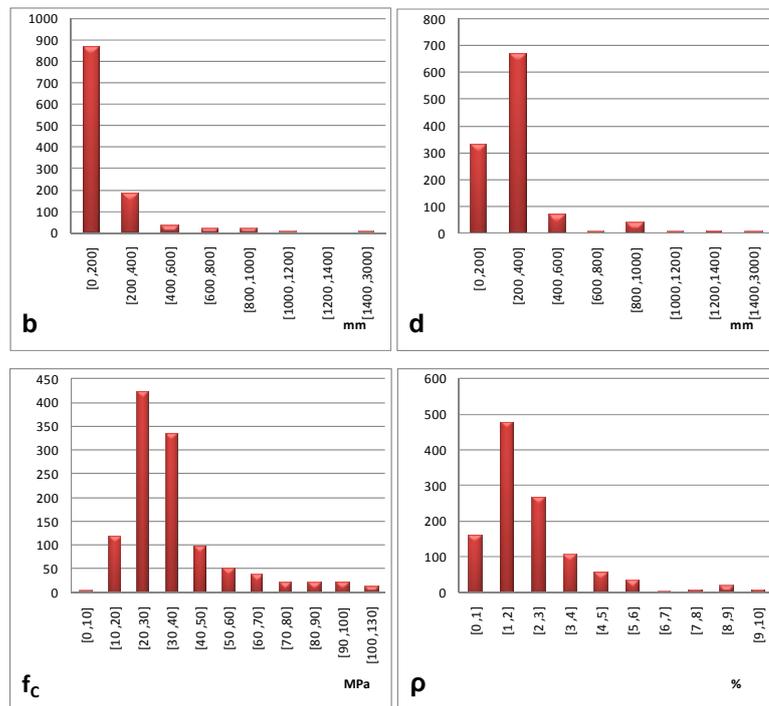


Figura 59. Distribución del conjunto de datos empleado

1.1.2 Modelos normativos actuales

Existe un amplio abanico de modelos teóricos de elevada complejidad que, debido a la dificultad de plasmar la importancia de todas las variables implicadas, derivan en diferentes propuestas normativas, tratándose en gran parte de casos de formulaciones de origen empírico. A continuación se presentan el código europeo: EC-2 (Eurocode 2, 2002) y el código americano: ACI 318-05 (ACI Committee 318, 2005).

1.1.2.1 Eurocódigo 2

En la ecuación (42) se muestra la formulación adoptada por el Eurocódigo 2 (Eurocode 2, 2002) en el caso de vigas sin esfuerzo axial y una vez eliminado el

factor de seguridad. El valor resultado de aplicar la ecuación (42) siempre debe ser mayor que el valor mínimo dado por la ecuación (43). En la Tabla 15 se muestran las distintas variables empleadas en esta formulación.

$$V_c = 0.18 * k * (100 * \rho_l * f_c)^{1/3} * b_w * d \quad (42)$$

$$V_{Rd,c \min} = 0.035 * k^{3/2} * f_c^{1/2} * b_w * d \quad (43)$$

Tabla 15. Variables empleadas en el modelo del Eurocódigo 2

Variables	Descripción
k	$k = 1 + \sqrt{\frac{200}{d}} \leq 2.0$
ρ_l	Cuantía geométrica de la armadura longitudinal traccionada, anclada a una distancia igual o mayor que d a partir de la sección de estudio $\rho_l = \frac{A_s}{b_w * d} \leq 0.02$
A_s	Armadura traccionada (mm ²)
f_c	Resistencia del hormigón a compresión (N/mm ²) $f_c \leq 90MPa$
b_w	Ancho del alma
d	Distancia desde la fibra extrema en compresión hasta el centroide del refuerzo longitudinal en tracción.

1.1.2.2 ACI318-05

El código ACI (*American Concrete Institute*) presenta dos posibles formulaciones para el cálculo de la resistencia a cortante en este tipo de elementos (ACI Committee 318, 2005). El método más sencillo se define mediante la ecuación 11-3 del Código ACI y se presenta en la ecuación (44). También se presenta un método alternativo, dado por la ecuación 11-5, en que se tiene en cuenta la influencia de la armadura longitudinal y de la relación $V \cdot d/M$ (45). En la Tabla 16 se muestra el significado de las variables utilizadas.

$$V_c = 0.17\sqrt{f_c} * b_w * d \quad (44)$$

$$V_c = \left(0.16\sqrt{f_c} + 0.17\rho_l * \frac{V * d}{M} \right) * b_w * d \quad (45)$$

$$\geq 0.29\sqrt{f_c} * b_w * d$$

Tabla 16. Variables empleadas en el modelo ACI318-05

Variables	Descripción	
ρ_l	Cuantía geométrica de la armadura longitudinal traccionada, anclada a una distancia igual o mayor que d a partir de la sección de estudio	$\rho_l = \frac{A_s}{b_w * d}$
A_s	Armadura traccionada (mm ²)	
$\frac{V \cdot d}{M}$	M momento concomitante al cortante V en la sección considerada	$\frac{V * d}{M} \geq 1$
f_c	Resistencia del hormigón a compresión (N/mm ²).	$\sqrt{f_c} \leq 8.3 \text{ N/mm}^2$
b_w	Ancho del alma.	
d	Distancia desde la fibra extrema en compresión hasta el centroide del refuerzo longitudinal en tracción.	

1.1.3 Comparación de los modelos presentados y contraste experimental

Por su sencillez y la correcta correlación frente a resultados experimentales en general, la ecuación dada en el Eurocódigo 2 se está extendiendo a diferentes normativas nacionales y europeas, como es el caso de la Instrucción Española (EHE-08, 2008). Una de las limitaciones que plantea, sin embargo, es el hecho de no considerar interacción entre momentos flectores últimos y cortantes últimos hasta valores elevados del momento flector, en los que el cortante resistido disminuye al plastificar la armadura traccionada. Para una sección

dada, según la formulación del EC-2, el cortante último es independiente del momento flector concomitante para momentos flectores alejados del que produce la plastificación de la armadura longitudinal. Por el contrario, modelos más complejos como la Modified Compression Field Theory (MCFT) (Vecchio & Collins, 1986) predicen, para una sección dada, una reducción del cortante de rotura a medida que aumenta el momento flector concomitante para cualquier valor del momento flector (Figura 60).

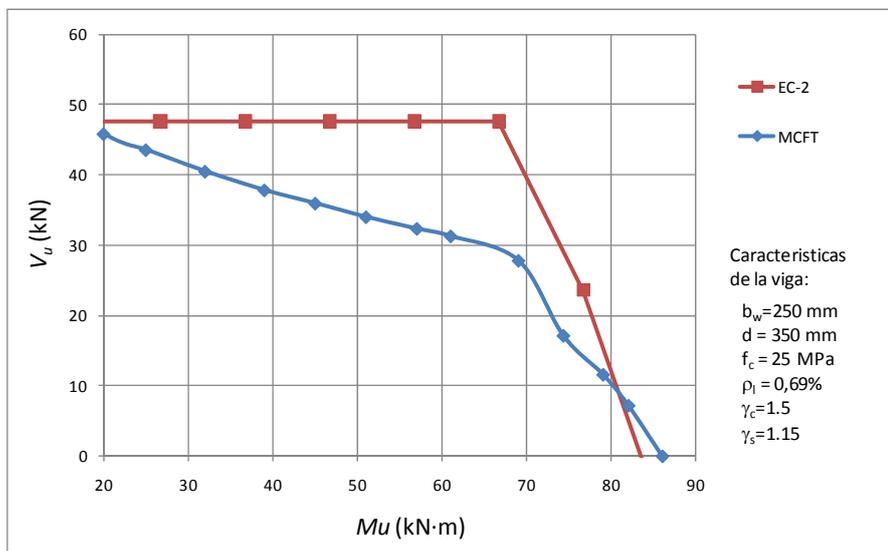


Figura 60. Interacción momento cortante según formulación EC-2 , la MCFT

El tratamiento de la influencia de la cuantía de armadura longitudinal varía también notablemente de una normativa a otra. La formulación dada por el Eurocódigo 2 propone que la resistencia a cortante es proporcional a la cuantía de armadura longitudinal. Sin embargo, otros modelos proponen que el cortante último es proporcional al valor $\rho_l V d / M$, como es el caso de uno de los métodos propuestos en el Código ACI. En el dimensionamiento real de vigas, la cuantía de armadura longitudinal crece proporcionalmente con el

momento flector concomitante y, por tanto, el parámetro $\rho_l Vd/M$ es prácticamente constante. Sin embargo, es habitual en ensayos de laboratorio utilizar valores de ρ_l desproporcionadamente elevados para evitar roturas a flexión e, incluso, realizar series de ensayos en los que a/d disminuye, disminuyendo por tanto el momento flector concomitante, sin variar el armado longitudinal. Por este motivo, algunos autores (Collins et al., 2008) sostienen que el ajuste de ecuaciones con estos ensayos no realistas puede producir desviaciones para elementos reales con las combinaciones habituales de las distintas variables. En el caso de la formulación dada por el Eurocódigo 2, un aumento de la cuantía longitudinal supondría siempre un incremento en la resistencia a cortante independientemente del momento flector concomitante.

Se ha contrastado las tres formulaciones normativas presentadas empleando el conjunto de datos definido anteriormente. Los resultados comparativos se presentan en la Tabla 17.

Tabla 17. Resultados comparativos entre diferentes formulaciones normativas

	EC-2	ACI 11-3	ACI 11-5
Promedio $V_{\text{test}}/V_{\text{pred}}$	1.12	1.49	1.33
Mediana	1.07	1.47	1.31
Desv. Estándar	0.28	0.50	0.36
COV (%)	24.85	33.39	26.92
$(V_{\text{test}}/V_{\text{pred}})_{1\%}$	0.67	0.34	0.50
$(V_{\text{test}}/V_{\text{pred}})_{99\%}$	2.85	3.34	2.97

Se observa que el mejor ajuste viene dado por la formulación propuesta en el EC-2, con un valor promedio del ratio $V_{\text{test}}/V_{\text{pred}}$ de 1.12 y un coeficiente de variación (COV) de 24.85%, frente a los valores de 1.49 (COV = 33.39%) para la formulación dada por la ecuación 11-3 del Código ACI318-05 y 1.33

(COV = 26.92%) para la ecuación 11-5 del ACI318-05. En la Tabla 17 también se presentan los valores del percentil del 1% y 99%, suponiendo una distribución normal de la mitad de los valores inferiores y superiores. Llama la atención que la ecuación 11-3 del código ACI ofrece un valor mínimo de 0.34, lo que sería un valor claramente del lado de la inseguridad aunque se incluyeran los coeficientes de seguridad correspondientes.

Si bien el EC-2 ofrece para el conjunto de la base de datos un resultado bastante satisfactorio, al analizar el comportamiento con detalle se observa que el ajuste podría ser mejorado. A modo de ejemplo, y a pesar de que el EC-2 trata de considerar el efecto tamaño, en la Figura 61 se observa que el ratio V_{test}/V_{pred} disminuye a medida que aumenta el canto, obteniéndose valores inferiores a 1.0 para cantos elevados.

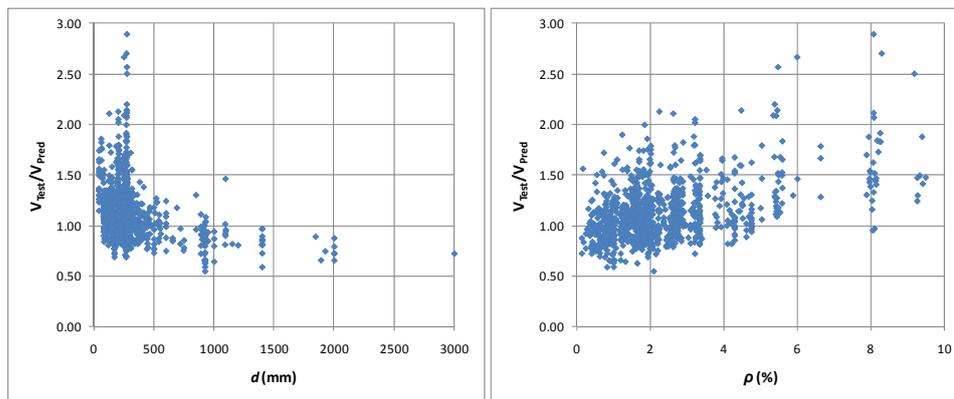


Figura 61. Ajuste experimental de la formulación del EC-2 frente a las variables d y ρ_l

Cabe destacar que la formulación del EC-2 sería insegura para muchas vigas con baja cuantía de armadura longitudinal, mientras que sería muy conservadora para vigas con cuantías elevadas.

1.2 Aplicación del Método

El objetivo perseguido en este ejemplo es lograr una expresión que permita predecir la resistencia a cortante de vigas de hormigón. Para ello y de acuerdo con los expertos del área de construcción se ha establecido la ecuación (46) como punto de partida para realizar la optimización mediante el algoritmo desarrollado. Esta ecuación representa el genotipo de la solución. Véase el gran parecido con la formulación del Eurocodigo 2 (42)

$$V_c = Rama_1 \left(1 + \left(\frac{Rama_2}{d} \right)^{Rama_3} \right) \rho^{Rama_4} * fpc^{Rama_5} * b * d \quad (46)$$

Para la resolución de este problema se emplean los módulos 3, 4 y 5 definidos en el apartado “Sistema basado en conocimiento aplicado a la Programación Genética”. En primer lugar es necesario definir la evaluación de los individuos en la función de ajuste. En el Módulo de coeficiente de seguridad se deben definir los parámetros k y l_{bias} definidos en la ecuación (32). Para ello se han realizado pruebas empleando técnicas clásicas de PG. Para la realización de estas pruebas se ha optado por emplear el 1.0, 1.1 y 1.3. El valor de V_{test}/V_{pred} de 1.0 representa el valor exacto de la predicción, frente al 1.1 que se sobredimensiona. De esta manera el modelo se situará ligeramente del lado de la seguridad (el valor de la fórmula obtenida será inferior que el valor del ensayo) con el objetivo de obtener una predicción conservadora.

Además, se han realizado las pruebas con y sin la ponderación del error en función de en qué intervalo esté. En la ecuación (47) se muestra el valor de “ p_i ” inspirado en el empleo de la técnica de “puntos de demérito” empleado

por Collins para categorizar los diferentes modelos de predicción (Collins, 2001).

$$p_i = \begin{cases} 10, & \frac{V_{test}}{V_{pred}} < 0.5 \\ 5, & 0.5 \leq \frac{V_{test}}{V_{pred}} < 0.67 \\ 3, & 0.67 \leq \frac{V_{test}}{V_{pred}} < 0.85 \\ 1, & 0.85 \leq \frac{V_{test}}{V_{pred}} < 1.3 \\ 2, & 1.3 \leq \frac{V_{test}}{V_{pred}} < 2 \\ 3, & \frac{V_{test}}{V_{pred}} \geq 2 \end{cases} \quad (47)$$

En total se han realizado un total de 5 conjuntos de pruebas diferentes (objetivo 1.0 con y sin penalización y objetivo 1.1 con y sin penalización, objetivo 1.3 con penalización) empleando la misma configuración. La configuración empleada para realizar las pruebas ha sido de población 1000 individuos, tasa de cruces del 80%, probabilidad de mutación del 20%, algoritmo de selección se ha empleado el torneo y altura máxima de 7 nodos de profundidad (esta configuración se ha elegido tras la ejecución de unas pruebas preliminares y son los que mejor han resultado). A mayores se repitieron las pruebas empleado distintos valores de parsimonia, un grupo sin tener en cuenta el tamaño y los otros con 0.01 y 0.000001. En total se realizaron 150 ejecuciones del algoritmo.

En Ingeniería Civil para ver las bondades de una predicción se suele emplear el Coeficiente de Variación COV (%). El COV se define como el cociente de la

desviación estándar entre el promedio de la muestra y multiplicado por 100.

En la ecuación (48) se describe su cálculo.

$$COV = \frac{\sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}}}{\frac{\sum_{i=1}^n (x_i)}{n}} * 100 \quad (48)$$

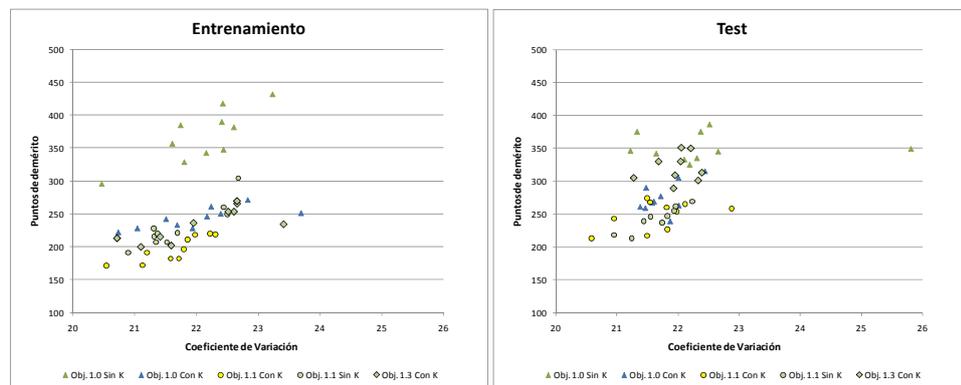


Figura 62. Resultados para entrenamiento y test de ejecuciones de PG sin parsimonia

En la Figura 62 se muestra tanto para entrenamiento como para test el resultado de combinar la técnica de “puntos de demérito” frente al Coeficiente de Variación. Cabe destacar que en el test hay dos casos que superan los límites del gráfico en cuanto al COV, obteniendo 378 (Obj. 1.1 Sin K) y 764 (Obj. 1.3 Con K). En la ecuación (49) se muestra cómo se calculan los puntos de demerito (PD) de una solución, siendo n el número de patrones.

$$PD(solución) = \sum_{i=0}^n k_i, \quad k = \begin{cases} 10, & \frac{V_{real}}{V_{pred}} < 0.5 \\ 5, & 0.5 \leq \frac{V_{test}}{V_{pred}} < 0.67 \\ 2, & 0.67 \leq \frac{V_{test}}{V_{pred}} < 0.85 \\ 1, & 1.3 \leq \frac{V_{test}}{V_{pred}} < 2 \\ 2, & \frac{V_{test}}{V_{pred}} \geq 2 \end{cases} \quad (49)$$

Como se puede observar empíricamente tras las pruebas realizadas se corrobora tanto en entrenamiento como en test (obsérvese Figura 62) que en la función de ajuste se debe emplear el factor de seguridad l_{bias} con valor 1.1 y el factor de ponderación del error, parámetros k_p ,

Como justificación del empleo de 1.1 como factor de esviaje (l_{bias}) se tiene en cuenta que en este tipo de ensayos existe una gran variabilidad en los resultados. Dada la misma configuración de ensayo está constatada una desviación de hasta un 10%-15% de los datos. Para ejemplificar esta situación se ha seleccionado un conjunto de datos de la base de datos (ver Tabla 18) que tengan las mismas variables de entrada (características de la viga) pero con diferente salida. En este ensayo seleccionado se constata una desviación típica de 5.15 kN, que representa un 7,86 % de diferencia frente al valor promedio de los 6 ensayos.

Para la realización de las pruebas se han establecido diversos conjuntos de pruebas, persiguiendo el objetivo de conducir el espacio de búsqueda según las indicaciones de los expertos en el área de trabajo. A la hora de realizar las ejecuciones del algoritmo desarrollado se ha establecido como configuración por defecto la listada en la Tabla 19. Se han elegido estos parámetros debido a que han sido los que mejores resultados han producido en las pruebas

iniciales. Cabe resaltar que los datos de entrada al algoritmo no han sido normalizados, con el objetivo de que los resultados se puedan aplicar directamente a los datos.

Tabla 18. Variabilidad de resultados en ensayos del mismo tipo

b_w	d	f_c	ρ_l	V_u
170	270	53.7	1.87	71.10
170	270	53.7	1.87	71.60
170	270	53.7	1.87	66.60
170	270	53.7	1.87	63.80
170	270	53.7	1.87	59.20
170	270	53.7	1.87	61.00
Promedio:				65.55
Desviación Típica				5.15
%				7.86%

Por defecto, se ha elegido como operadores o nodos no terminales la suma, resta, producto y división protegida. Para los nodos terminales se emplean las variables procedentes del conjunto de datos (b , d , f_c y ρ), constantes aleatorias comprendidas entre el intervalo $(-1, 1)$ junto con enteros entre los intervalos $[-10, 10]$.

Después de estudiar conjuntamente con los expertos del área de construcción se establecieron los 8 conjuntos de restricciones definidos en la Tabla 20, donde se muestran las restricciones empleadas para cada rama. Con el símbolo “Cte.” Se indica que se permite el empleo de cualquier constante (generada mediante PG, a base de operaciones suma, resta, multiplicación y división protegida de constantes: reales en el intervalo $(-1, 1)$ y enteros en el intervalo $[-10, 10]$). Con los símbolos d , f_c y ρ o cualquier combinación de ellos indica que se pueden emplear dichas variables de la base de datos o cualquier constante relacionada mediante los operadores suma, resta, multiplicación y

división protegida. Por último, cuando aparezca un valor indica que sólo se empleará ese valor en esa rama. Por ejemplo, las restricciones denominadas “F” definidas en la Tabla 20, muestran el potencial del método empleado. En él se fija el valor 0.18 y 0.37 en las ramas 1 y 4 respectivamente, permitiendo únicamente el empleo de las variables d para la rama 2 y f_c para las ramas 3 y 5.

Tabla 19. Parámetros empleados

Parámetro	Valores por Defecto	Otros valores
Tamaño Población	1000	
Tasa de Cruces:	80%	
Tasa de Selección no Terminal:	90%	
Probabilidad Mutación:	20%	
Algoritmos:	Selección: Torneo Creación: Intermedio Mutación: Subárbol	
Estrategia elitista	Si	
Parsimonia	0	0.1, 0.01, 0.001, 0.0001, $1 \cdot 10^{-7}$ ó $1 \cdot 10^{-10}$
Altura Inicial	3	6
Altura Máxima	6	8 ó 10
Altura Máxima mutación	3	7 ó 8

Tabla 20 Restricciones aplicadas

Restricción	R ₁	R ₂	R ₃	R ₄	R ₅
A	Cte	$df_c \varrho$	$df_c \varrho$	$df_c \varrho$	$df_c \varrho$
B	Cte	$df_c \varrho$	Cte	Cte	Cte
C	Cte	Cte	$df_c \varrho$	Cte	Cte
D	Cte	Cte	Cte	$df_c \varrho$	Cte
E	Cte	Cte	Cte	Cte	$df_c \varrho$
F	0.18	d	f_c	0.37	f_c
G	Cte	Cte	$0.40 + 0.001f_c$	0.37	Cte
H	Cte	Cte	Cte	Cte	Cte

1.2.1 Resultados

Los resultados obtenidos surgen de la realización de más de 300 ejecuciones diferentes del algoritmo con un promedio por ejecución de más de 6700 generaciones. En las siguientes tablas se muestran los valores de las normativas actuales junto con los resultados obtenidos de los individuos más significativos en cada caso. Este grupo se ha subdividido en tres categorías. La primera representa a las ecuaciones generadas mediante el algoritmo desarrollado en este artículo, la segunda corresponde con la PG clásica y por último la PG clásica con ponderación del error. Para cada categoría se muestra en negrita el mejor valor, subrayando el mejor en global. En cada tabla aparece el nombre de la ecuación (formado por el Nombre_restricciones + n°_de_solución) por ejemplo la ecuación A3 significa restricciones “A”, fórmula seleccionada número “3” (En el apartado 1.2.1.1 de este capítulo se pueden observar las fórmulas obtenidas). A continuación del nombre están el coeficiente de variación COV (48), el V_{ens}/V_{form} representa el cociente entre el valor obtenido en el ensayo y el valor calculado mediante los distintos modelos, el R^2 es el cuadrado del coeficiente de correlación de momento del producto Pearson, calculado mediante la ecuación (50). Por último figuran el Error Cuadrático Medio (51) y el Error Medio (52) y los puntos de demérito calculados mediante el método empleado por Collins para categorizar un modelo (Collins, 2001)

$$R^2 = \left(\frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 (y_i - \bar{y})^2}} \right)^2 \quad (50)$$

$$ECM(X, Y) = \frac{1}{n} \sum_{i=1}^n (x_i - y_i)^2 \quad (51)$$

$$EM(X, Y) = \frac{1}{n} \sum_{i=1}^n abs(x_i - y_i) \quad (52)$$

En la Tabla 21 se muestran los resultados para el conjunto de entrenamiento. En función de qué criterio a seguir se obtendría que las ecuaciones A1(53), A2(54) y A3(55) son las mejores para predecir el fenómeno a cortante en este conjunto de datos. Si se toma como referencia los puntos de demérito sería la A3 la mejor función. En cambio si se toma como referencia el ECM sería la A1 la mejor adaptada. En cuanto al conjunto de datos de test (ver Tabla 22) la ecuación que mayor capacidad de generalización tiene es la A3.

Tabla 21. Resultados entrenamiento

		COV	V _{ens} /V _{form}	R ²	ECM	EM	PD
Modelos internacionales	EC-2	23.40	1.10	0.960	2165.38	17.84	229
	ACI 11-3	33.23	1.44	0.867	18820.12	41.11	636
	ACI 11-5	28.72	1.30	0.876	18962.56	38.13	551
Modelos generados por el algoritmo de PG desarrollado	A1 (53)	17.32	1.09	0.978	387.92	11.36	106
	A2 (54)	15.31	1.08	0.977	402.98	10.64	77
	A3 (55)	16.51	1.10	0.972	542.31	11.84	72
	B1(56)	17.88	1.11	0.960	806.80	13.30	116
	C1 (57)	18.37	1.09	0.968	767.09	12.55	122
	D1 (58)	17.94	1.09	0.976	433.11	11.89	123
	E1 (59)	17.61	1.10	0.977	446.50	12.03	114
	F1 (60)	15.30	1.08	0.973	542.31	11.84	76
	G1 (61)	17.77	1.09	0.979	409.90	11.70	116
H1 (62)	17.83	1.10	0.976	441.53	11.91	117	
PG Clásica	P1.0 (63)	18.40	0.98	0.967	944.29	13.37	273
	P1.1 (64)	18.59	1.08	0.965	803.00	13.60	165
PG Clásica con ponderación error (K)	P1.0K (65)	18.49	1.03	0.948	2552.57	15.78	199
	P1.1K (66)	18.33	1.10	0.961	705.53	13.94	145
	P1.1KC (67)	16.78	1.10	0.971	685.41	12.49	89

Por último en la Tabla 23 se muestran los resultados teniendo en cuenta los dos conjuntos de datos. En este caso tanto la ecuación A1 como la A3 presentan el mejor ajuste. Tanto la A1(53) como la P1.1K (66) son ecuaciones

relativamente sencillas siendo la A1 mejor teniendo en cuenta su simplicidad. En cambio la A3 aun obteniendo una buena clasificación es demasiado compleja (55) para su uso por parte de los ingenieros civiles.

Cabe destacar el aumento de seguridad conseguido tanto con el coeficiente de seguridad “1.1” aplicado como con el factor “K” de ponderación del error. Este efecto se puede ver en la Figura 63 donde se muestra la correlación de las predicciones de Eurocódigo 2 y la ecuación A1, la zona sombreada en rojo corresponde con valores inseguros (deficiencia estructural)

Tabla 22. Resultados Test

		COV	V_{ens}/V_{form}	R^2	ECM	EM	PD
Modelos internacionales	EC-2	25.80	1.14	0.928	627.68	13.23	267
	ACI 11-3	33.26	1.53	0.829	1902.22	25.62	639
	ACI 11-5	25.36	1.35	0.845	1778.20	21.98	488
Modelos generados por el algoritmo de PG desarrollado	A1 (53)	20.11	1.11	0.941	548.76	12.18	195
	A2 (54)	26.90	1.13	0.937	619.48	13.28	244
	A3 (55)	20.07	1.14	0.962	497.30	12.50	177
	B1(56)	22.44	1.16	0.931	673.01	13.73	213
	C1 (57)	20.20	1.11	0.942	527.19	11.95	195
	D1 (58)	20.56	1.12	0.950	536.59	12.59	221
	E1 (59)	20.90	1.13	0.944	631.20	13.28	221
	F1 (60)	21.44	1.11	0.787	503.08	12.52	274
	G1 (61)	20.65	1.12	0.950	567.85	12.84	210
	H1 (62)	20.54	1.12	0.949	548.31	12.71	209
PG Clásica	P1.0 (63)	21.23	1.01	0.934	507.54	11.65	346
	P1.1 (64)	20.95	1.12	0.956	477.34	12.49	218
PG Clásica con ponderación error (K)	P1.0K (65)	21.46	1.08	0.922	623.14	12.02	259
	P1.1K (66)	20.58	1.13	0.959	503.57	12.73	213
	P1.1KC (67)	22.67	1.14	0.944	804.65	14.28	229

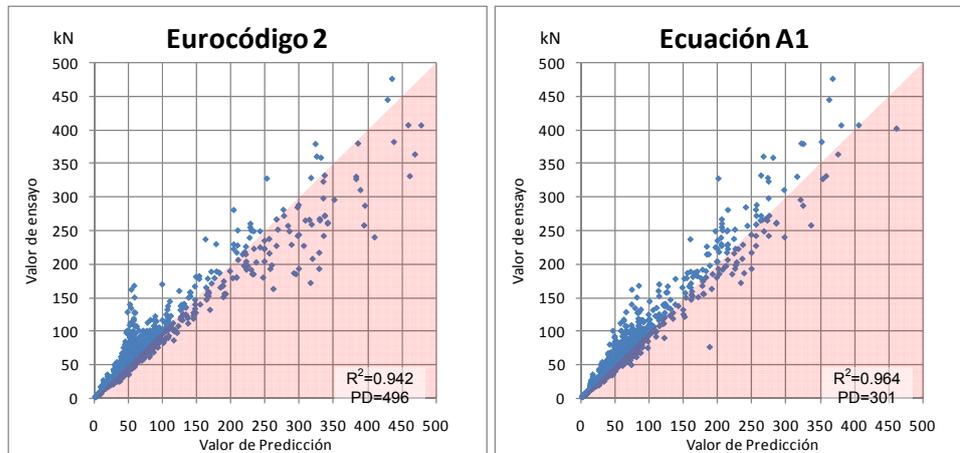


Figura 63. Correlación entre valor de ensayo y valor de predicción

Tabla 23. Resultados global

		COV	V_{ens}/V_{form}	R^2	ECM	EM	PD
Modelos internacionales	EC-2	24.85	1.12	0.942	1320.92	15.31	496
	ACI 11-3	33.39	1.49	0.816	9529.26	32.60	1275
	ACI 11-5	26.92	1.33	0.830	9525.37	29.26	1039
Modelos generados por el algoritmo de PG desarrollado	A1 (53)	18.95	1.10	0.964	476.25	11.81	301
	A2 (54)	22.80	1.11	0.960	521.88	12.09	321
	A3 (55)	18.69	1.12	0.967	517.59	12.20	249
	B1(56)	20.70	1.14	0.945	733.33	13.54	329
	C1 (57)	19.43	1.10	0.952	635.34	12.22	317
	D1 (58)	19.47	1.11	0.965	489.94	12.27	344
	E1 (59)	19.58	1.12	0.962	547.93	12.72	335
	F1 (60)	19.06	1.10	0.906	520.76	12.22	350
	G1 (61)	19.46	1.11	0.966	496.64	12.33	326
H1 (62)	19.43	1.11	0.964	500.17	12.35	326	
PG Clásica	P1.0 (63)	20.12	1.00	0.952	704.44	12.42	619
	P1.1 (64)	20.05	1.10	0.952	624.15	12.99	383
PG Clásica con ponderación error (K)	P1.0K (65)	20.40	1.06	0.929	1492.98	13.72	458
	P1.1K (66)	19.69	1.12	0.954	594.62	13.28	358
	P1.1KC (67)	20.37	1.12	0.958	750.89	13.47	318

En la Figura 64 se muestra el cómputo de puntos de demérito cometidos por cada uno de los modelos y ecuaciones (mejor cuanto menos). A efectos de clarificación se ha limitado el gráfico a 800 puntos ya que los modelos de la ACI superaban todos los 1000 puntos. Dado esto se puede observar que los modelos internacionales son los que peor clasifican seguidos de la PG clásica, que con la ecuación P1.1 obtiene mejores resultados que cualquiera de los modelos internacionales. Estos resultados se mejoran empleando el factor “k” de ponderación del error, y por último, al aplicar lo desarrollado en esta tesis, se consigue refinar aun más y cometer menos errores en la predicción.

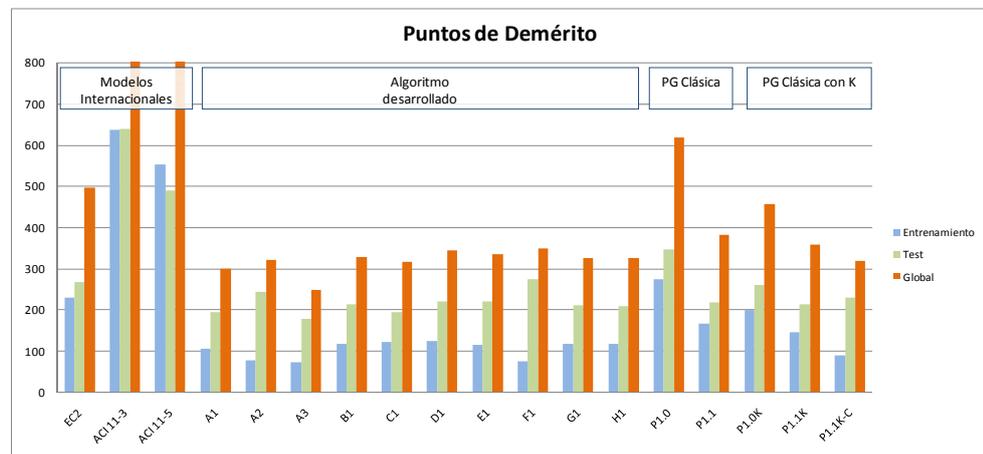


Figura 64. Puntos de demérito cometidos por las fórmulas

Algunos de los resultados alcanzados en el estudio del cortante se han presentado en los congresos IECC'5 (Rabuñal, Pérez, Cladera & González 2008) y CST-2010 (Cladera, Pérez & Martínez 2010). Además se han publicado dos capítulos de libro en la *“Encyclopedia of Artificial Intelligence”* (Pérez, González & Martínez, 2008; Pérez, Martínez & Herrador, 2008) y en la actualidad se han enviado a diversas revistas para su publicación.

1.2.1.1 Ecuaciones

$$A1 = 1.6399 * \left(\frac{\text{abs} \left(\frac{-14.2569}{-40 + 8} - fc \right)}{d} \right)^{\frac{1}{8} \frac{d}{fc - 50 + \frac{\rho_l}{-3 + \frac{1}{16}d} + 50}} * \rho_l^{0.3706} * fc^{\frac{3}{20} \frac{5}{7} + fc + \frac{1}{252}d} * b_w * d \quad (53)$$

$$A2 = 0.198919 * \left(1 + \left(\frac{\text{abs}(Rama2)}{d} \right)^{Rama3} \right) * \rho_l^{Rama4} * fc^{Rama5} * b_w * d$$

Donde:

$$Rama2 = \frac{-903 + \frac{25}{2} * d + \frac{-80}{3} - \frac{1}{3} * fc + \frac{-7}{4} * \frac{-378 * d + \frac{1701}{2}}{-d + 199} + 34}{\frac{63}{17} - \frac{8}{7} * \rho_l} * \frac{-\frac{611}{7} + 19 * fc + d}{\frac{280}{997} * \frac{\rho_l}{-108 * \rho_l + 364} - 68 - \frac{1}{2} * fc} \quad (54)$$

$$Rama3 = \frac{-100 + d + \rho_l - \frac{4}{6 - fc}}{\frac{2023}{2} - \frac{7}{2} * fc + \frac{112}{-5 + \rho_l}}$$

$$Rama4 = \frac{\frac{2}{7} - \frac{\frac{4}{7}}{\frac{72}{17} - \frac{8}{7} * \rho_l}}$$

$$Rama5 = \frac{-\frac{9}{32} - \frac{65}{6}}{\frac{-71}{4} + \frac{1}{4} * fc} - 34$$

$$A3 = 1 * \left(1 + \left(\frac{Branch2}{d} \right)^{\frac{-\frac{1}{320} * (-1 + \frac{1}{10} * fc) * (100 * \rho_l - \frac{1}{3} * fc) + \frac{1}{30} * d}{-fc + 3 + 100 * \rho_l}} \right) * (100 * \rho_l)^{Branch4} * fc^{\frac{1}{10} * \frac{1}{4} + \frac{3}{4 - fc} * d} * b_w * d$$

Donde:

$$Branch2 = \frac{\frac{-4 + fc - 100 * \rho_l}{d - 12 - 10 * fc}}{65 - \frac{1}{3} * d} * \frac{\frac{1}{3} * \frac{d}{100 * \rho_l - 0.63} - \frac{31}{3} * fc + \frac{1}{27} * (36 * fc - \frac{1}{3} * d + 10) * d}{\frac{1}{3} * d - \frac{36 + 10 * fc}{100 * \rho_l}} * \frac{1 - \frac{9}{10} * fc - \frac{9}{d}}{\left(-\frac{5}{9} - \frac{1}{3} * d \right) * \left(d + \frac{10}{d} * (fc + 3) \right) - 105.6838 - \frac{1}{3} * d} \quad (55)$$

$$\text{Branch}_4 = \frac{8 * f_c - 45 - \frac{1}{100} * \frac{d^2}{(100 \rho_l - f_c) * (f_c - 5) - 100} + \frac{\frac{300}{100 \rho_l - 0.63} - \frac{7}{3} * d + 3}{\frac{-50}{69} * d}}{\frac{\frac{0.63 - \frac{d}{3}}{-10} - \frac{1}{3} * d}{100 - \frac{1}{3} * d} + \frac{\frac{f_c}{8} - \frac{1}{90} * d}{\frac{1}{90} * d} + 18 * f_c}$$

$$B1 = \left(\frac{24}{49} + \frac{24}{49} * \left(\frac{\text{abs} \left(\frac{100}{3} * \rho_l - 10 * f_c + \frac{10}{3} + \frac{1}{6} * d \right)}{d} \right)^{\frac{9}{19}} \right) * \frac{\rho_l^{\frac{5}{12}}}{f_c^{\frac{1}{168}}} * b_w * d \quad (56)$$

$$C1 = 0.4617 * \left(1 + \left(\frac{3.67}{d} \right)^{\frac{\left(\frac{d}{9} \right) + \left(\frac{f_c - 6}{-3} \right)}{\left(\frac{f_c - \rho_l}{0.23087253} \right) + \left(7 + \left(\frac{d}{5} \right) \right)} \right) * \rho_l^{0.3565} * f_c^{0.0823} * b_w * d \quad (57)$$

$$D1 = 0.0918912 * \left(1 + \left(\frac{2424.57}{d} \right)^{0.392987} \right) * \rho_l^{0.392987} * f_c^{0.31492} * b_w * d \quad (58)$$

$$E1 = 0.3571 * \left(1 + \left(\frac{9.8250}{d} \right)^{0.1765} \right) * \rho_l^{0.4006} * \frac{f_c^{16-3*\rho_l+2*f_c}}{d+3+8*f_c} * b_w * d \quad (59)$$

$$F1 = 1 * \left(1 + \left(\frac{0.944982}{d} \right)^{0.40+0.001*f_c} \right) * \rho_l^{0.37} * f_c^{-0.164647} * b_w * d \quad (60)$$

$$G1 = 0.1235 * \left(1 + \left(\frac{1600}{d} \right)^{0.40+\frac{f_c}{1000}} \right) * (100\rho_l)^{0.37} * f_c^{1/3} * b_w * d \quad (61)$$

$$H1 = 0.094 * \left(1 + \left(\frac{1600}{d} \right)^{0.42} \right) * (100\rho_l)^{0.37} * f_c^{1/3} * b_w * d \quad (62)$$

$$P1.0 = \frac{10 * f_c + b_w * d}{-2 * \rho_l - 0.8612} + \rho_l * b_w * f_c + b_w * f_c + b_w * d + \frac{-5 * b_w * f_c + 2 * b_w * d}{b_w * d + f_c^2} * (b_w + b_w * f_c) \quad (63)$$

$$P1.1 = \frac{b_w^2 * f_c + b_w * d * (d + f_c)}{-50 * \rho_l * f_c - d - b_w} + \rho_l * b_w * f_c + b_w * d - 20 * f_c - 10 * \rho_l^2 + 10 * b_w \quad (64)$$

$$P1.0K = \frac{b_w * d + f_c^2}{-3 * \rho_l - 0.9119} + (-10 + b_w) * \rho_l * f_c + b_w * f_c + b_w * d + \frac{-3 * b_w * f_c + b_w * d - 9 * b_w}{b_w * d + f_c^2} * (b_w * f_c - 10 * d) \quad (65)$$

$$P1.1K = d * \frac{-5 * d + d * b_w}{-42 * \rho_l * f_c - \frac{2}{3} * b_w - d} + b_w * \rho_l * \frac{f_c + 5}{-\frac{1}{10} * d + 1} + b_w * \rho_l * (f_c + 4) + d * b_w - 10 * f_c \quad (66)$$

$$\begin{aligned}
 & P1.1KC \\
 & = \frac{b_w^2}{b_w + 42} + 4 * d + 10 * b_w - 46 + \rho_l + b_w * \rho_l * f_c + d * b_w + 18 * \frac{b_w}{5 * \rho_l * f_c - 35} + \frac{5}{18} * \rho_l * f_c * \frac{d}{3 + d + \rho_l - f_c} \\
 & + \frac{36}{7} * \frac{d}{-18 + 6 * \rho_l} - \frac{f_c + 10 * d}{\rho_l} + \frac{\rho_l - f_c - d}{\rho_l - 4} - 7 * \rho_l^2 * f_c \\
 & + \frac{\alpha + (b_w + 2) * b_w * \rho_l + b_w * \rho_l * d + 81 * b_w - 324 - 72 * f_c - d * b_w * (-9 + b_w + d) + \frac{b_w + d}{\rho_l - 2} * (9 * d + \rho_l)}{\frac{10}{9} * d - 7 + 2 * b_w + 2 * \rho_l + \frac{-5 + d}{2 * f_c + 5} + (33 - 0.2273 * f_c) * (\rho_l^2 * f_c + \rho_l * f_c)} \quad (67)
 \end{aligned}$$

Donde:

$$\alpha = 2 * (\rho_l * f_c + b_w + d) * \rho_l * f_c * \frac{d}{-\frac{3}{2} * d + \rho_l * f_c + f_c}$$

2 Adherencia

2.1 Descripción

La adherencia entre el hormigón y el acero que conforma su armadura pasiva se puede definir como el conjunto de efectos en las zonas de contacto hormigón-acero, asociados al deslizamiento relativo entre ambos. El planteamiento de la formulación de adherencia para hormigón armado suele basarse en la determinación de la longitud de anclaje necesaria para conseguir la correcta fijación de la barra. Se puede definir, entonces, la longitud de anclaje como la necesaria para garantizar la resistencia del anclaje por adherencia hasta la rotura del acero. Actualmente no existen formas de medir empíricamente la longitud de anclaje, por lo que se emplean distintas aproximaciones en forma de normativas.

Puede considerarse que, desde el punto de vista español, tres escuelas constituyen una referencia a la hora de plantear el cálculo de la longitud de anclaje de las barras de armadura pasiva. La escuela americana, iniciada en la primera mitad del siglo XX y que, con el impulso de Orangun y Jirsa, se

plasmó en el código ACI en los años '60 sin que, hasta la fecha, haya experimentado grandes variaciones. La escuela europea, vinculada a la FIB y desarrollada a partir de figuras fundamentales como Tepfers, ha sido el germen del Código Modelo y el Eurocódigo-2. Y la escuela española, con poca trascendencia más allá de nuestras fronteras, inspiradora de la redacción del apartado correspondiente de la EHE, cuya potencia se apoyó en los miles de ensayos que durante la década de los '70 se desarrollaron en el Instituto Eduardo Torroja, aún vigentes.

Pese a los planteamientos variados, las tres escuelas tienen un nexo común: los planteamientos se desarrollan a partir de las evidencias experimentales. A partir de una expresión básica de la tensión de adherencia, dependiente de una variable principal, se incorporan con carácter multiplicativo los efectos de otras variables.

Existen tres fenómenos asociados a la adherencia: la adherencia por contacto o adhesión, la fricción o rozamiento, y el acuñaamiento. Los dos primeros desaparecen rápidamente en barras a tracción debido al coeficiente de Poisson. El último factor, que sólo está presente cuando se emplean barras corrugadas, depende de la altura y separación de las corrugas, siendo un parámetro constante en todas las barras. Con este factor aparecen unas fuerzas que son transmitidas al hormigón. Eventualmente el deslizamiento del hormigón paralelo a la barra puede fisurarlo, produciendo una caída en la adherencia. En la Figura 65 se muestran las distintas fuerzas producidas entre el acero corrugado y el hormigón, cuando se somete a una carga (Thompson, Jirsa Breen & Klingner, 2002).

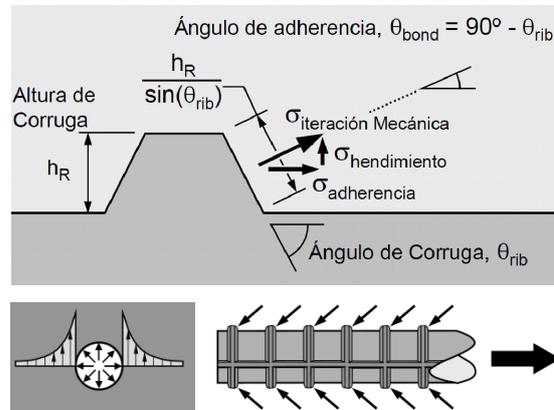


Figura 65. Distintas fuerzas existentes entre la unión hormigón-acero con corrugas

Cuando se ensaya una barra frente arrancamiento, alrededor de la misma se genera un estado de tracciones radiales que puede provocar la rotura del hormigón circundante, circunstancia que puede mitigarse con la disposición de la armadura transversal, por una parte, y con la adopción de una corona resistente de suficiente magnitud (suficiente hormigón circundante). En la Figura 66 se muestra gráficamente el fenómeno de adherencia de una barra. (Cairns, 2006)

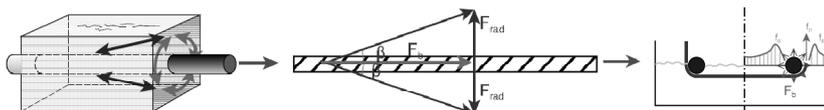


Figura 66. Fenómeno de adherencia

Existen diversos factores que, en mayor o menor medida, influyen en la capacidad adherente entre los elementos. Generalmente son varios y se asocian a diversos orígenes: los materiales empleados, las condiciones geométricas y por último las cargas aplicadas.

Teniendo en cuenta los materiales, esencialmente está compuesto por el hormigón y el acero. En el caso del **hormigón** aparecen tensiones de tracción y compresión. Por este motivo el resultado final de la adherencia depende de la resistencia a tracción, debido a las tensiones generadas por la componente radial de las fuerzas por acuíamiento en las corrugas (posible splitting), y la resistencia a compresión. Las tensiones de compresión vienen determinadas por la componente longitudinal de las fuerzas por acuíamiento en las corrugas ya que genera tensiones de compresión en el hormigón que tienden a provocar su aplastamiento. El otro elemento que influye es el **Acero**, en este caso será determinante si se aplican barras lisas o corrugadas, el estado superficial de las barras, ya sea por una posible oxidación, o por una contaminación debido a sustancias desencofrantes, etc. Y por último el diámetro de la barra, de este último se conoce la siguiente relación: a igualdad de parámetros, la tensión de adherencia decrece a medida que aumenta el diámetro de la barra.

El segundo tipo de factores vienen determinados por las **condiciones geométricas**, principalmente se dividen entre los factores asociados a la armadura de tracción, y la existencia o no de armadura de confinamiento. En el primer caso, se identifican cuatro factores, el primero es la geometría de las barras, es decir, la superficie relativa de la corruga. El segundo corresponde con el posicionamiento de las barras ya que influye la dirección del hormigonado (cuando se emplean barras horizontales se pueden generar bolsas de lechada). El tercer factor viene determinado por los grupos de barras, ya que se puede presentar dificultad a la hora del hormigonado. Por último, el cuarto factor viene determinado por el recubrimiento, ya que existe una relación directa con la adherencia, es decir, si se aumenta el recubrimiento se aumenta proporcionalmente la adherencia. En la Figura 67 se muestran

diferentes formas de posicionar una barra dentro del hormigón (a) y distintos tipos de recubrimiento junto con las fracturas típicas que se producen (b).

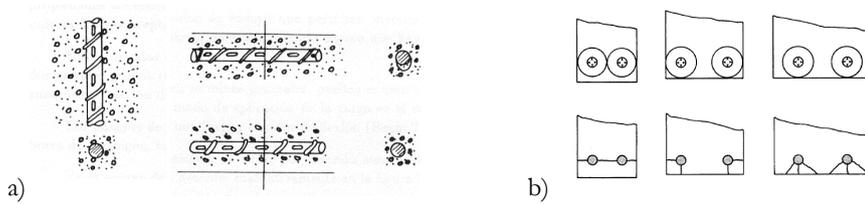


Figura 67. Condiciones geométricas a) Posición de barras b) Recubrimiento

En cuanto al segundo tipo de factores asociados a condiciones geométricas vienen determinados por la existencia o no de la armadura de confinamiento. En este caso otorga una capacidad mecánica y separación, por un lado absorbe parte de las tracciones haciendo inferior la influencia de la longitud. En la Figura 68 se muestra el efecto de la armadura de confinamiento (ACI Committee 318, 2005).

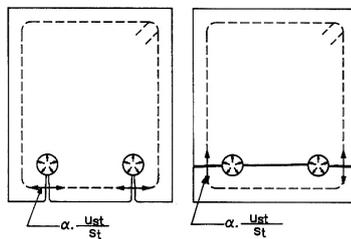


Figura 68. Armadura de confinamiento

El tercer tipo, y último, de los factores que influyen en la adherencia viene determinado por los ciclos de cargas, ya que a más cantidad de ciclos más disminuye la adherencia, es decir, influye la fatiga de los materiales.

El planteamiento de la formulación de la adherencia se suele hacer diseñando la longitud de anclaje necesaria para conseguir el correcto anclaje de la barra

(en estado límite último). Para ello se han acometido a lo largo del tiempo muchas campañas experimentales destinadas a conocer y a cuantificar la influencia de las distintas variables. La primera, probablemente, fue la de Abrams, quien determinó en la primera década del siglo XX las condiciones de anclaje en barras lisas (Abrams 1913). Los planteamientos más recientes, inspiradores de la formulación americana de la ACI, surgen de los trabajos de Orangun et al. (Orangun, Jirsa & Breen, 1977), quien acometió el problema en tres fases (la definición de variables aparece en la expresión normativa ACI posterior):

- Fase 1: tras ensayos experimentales se postula una formulación para la tensión de adherencia, ecuación (68), y normativamente se propone una fórmula de cálculo de l_d mediante ecuación (69)

$$u = \frac{9.5 * \sqrt{f_c}}{d_b} \quad (68)$$

$$l_d = 0.04 \frac{A_d * f_y}{\sqrt{f_c}} \quad (69)$$

- Fase 2: aparece la dependencia con parámetros geométricos dependientes de la armadura de tracción, proponiendo a su vez una formulación normativa de l_d , ver ecuaciones (70) y (71).

$$\frac{u}{\sqrt{f_c}} = 1.22 + 3.23 \frac{c}{d_b} + 53.0 \frac{d_b}{l_d} \quad (70)$$

$$\frac{L_d}{d_b} = \frac{f_y - 200 * \sqrt{f_c}}{\left(1 + \frac{5}{2} * \frac{c}{d_b}\right) * 4.8 * \sqrt{f_c}} \quad (71)$$

- Fase 3: se descubre la influencia de la armadura transversal, proponiéndose la correspondiente formulación normativa de l_d (se consideran ambos efectos aditivos). Ver ecuaciones (72) y (73).

$$\frac{u}{\sqrt{f_c}} = \frac{1}{500} \frac{A_{tr} * f_{yt}}{s * d} \quad (72)$$

$$\frac{l_d}{d_b} = \frac{f_y - 200 * \sqrt{f_c}}{\left(1 + \frac{5}{2} * \frac{c}{d_b} + \frac{A_{tr} * f_{yt}}{600 * s * d}\right) * 4.8 * \sqrt{f_c}} \quad (73)$$

2.1.1 Modelos normativos actuales

2.1.1.1 ACI

La actual norma ACI (ACI Conmittee 318, 2005) plantea una expresión para el cálculo de la longitud de anclaje(74). En ella se considera que la tensión del acero traccionado (al igual que hizo Orangun) es la correspondiente al límite elástico de cálculo, lo que equivale a considerar que la barra desliza a la máxima carga de la barra (y no a su teórica tensión de trabajo, que puede ser menor). La ecuación de la ACI (74) está acotada por las restricciones que aparecen en la ecuación (75), en la Tabla 24 se muestra la descripción de cada variable empleada.

$$l_d = \frac{0.075 * f_y * \alpha * \beta * \lambda * d_b}{\sqrt{f_c} * \left(\frac{c + k_{tr}}{d_b}\right)} \quad (74)$$

$$\left(\frac{c + k_{tr}}{d_b}\right) < 2.5 \quad y \quad a * b < 1.7 \quad (75)$$

Tabla 24. Variables empleadas en el modelo de la ACI para el cálculo de la longitud de anclaje

Variables	Descripción
f_y	límite elástico de las barras de armadura (psi)
α	1.3 para barras “top” (superiores), de adherencia deficiente 1.0 para el resto de casos
β	1.0 para barras sin recubrimiento epoxídico 1.5 para barras con epoxi con recubrimiento $< 3d_b$ o espaciamiento $< 6 d_b$ 1.2 para barras con epoxi con recubrimiento $\geq 3d_b$ o espaciamiento $\geq 6 d_b$
γ	0.8 para barras de diámetro #10 - #19 1.0 para barras de diámetro #22 - #57
λ	1.0 para hormigones de densidad normal
d_b	diámetro nominal de la barra (in)
f'_c	resistencia a compresión del hormigón (psi)
c	menor de las siguientes distancias: recubrimiento lateral o inferior del hormigón, la mitad de la separación entre barras (medida de centro a centro de barra) (in)
K_{tr}	$K_{tr} = A_{tr} \frac{f_{yt}}{1500 * s * n}$ (psi*in)
A_{tr}	área total de armadura transversal separada “s”, que atraviesa el posible plano de deslizamiento dónde las barras están siendo ancladas (in ²)
f_{yt}	límite elástico del acero de armadura transversal (psi)
s	máxima separación entre centros de armadura transversal en longitud l_d (in)
n	número de barras ancladas a lo largo del plano de deslizamiento

2.1.1.2 Eurocódigo 2

Los estudios en el ámbito europeo se desarrollaron en el C.U.R. y contribuyeron los ensayos de Tepfers, así como los planteamientos de Eligehausen y Kempt. Fruto de este conocimiento se ha desarrollado la normativa propuesta para el nuevo código Eurocódigo 2 (EC-2) (Eurocode 2, 2002). Dicha norma propone que la longitud de anclaje (78) se defina en función de la tensión de adherencia (76) y de la longitud de anclaje básica (77).

$$f_{db} = 2.25 * \eta_1 * \eta_2 * f_{ctd} \quad (76)$$

Tabla 25. Variables empleadas en el Eurocódigo 2 para el cálculo de tensión de adherencia

Variables	Descripción
f_{yctd}	valor de la resistencia a tracción del hormigón limitada a la de un C60 (MPa)
η_1	relativo a la condición de adherencia según la posición de la barra: 1.0 para buenas condiciones 0.7 para otros casos
η_2	relativo al diámetro de la barra: 1.0 para $d_b \leq 32$ mm $\frac{132-d_b}{100}$ para $d_b > 32$ mm

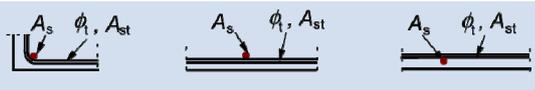
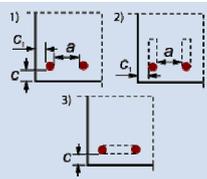
$$f_{b,rqd} = \frac{d_b}{4} * \frac{\sigma_{sd}}{f_{fcd}} \quad (77)$$

Tabla 26. Variables empleadas en el Eurocódigo 2 para el cálculo de la longitud de anclaje básica

Variables	Descripción
σ_{sd}	límite elástico del acero de la barra a anclar (MPa)
f_{bd}	tensión de adherencia definida en la ecuación (76) (MPa)
d_b	diámetro de la barra:

$$l_{bd} = \alpha_1 * \alpha_2 * \alpha_3 * \alpha_4 * \alpha_5 * l_{b,rqd} \geq l_{d,min} \quad (78)$$

Tabla 27. Variables empleadas en el Eurocódigo 2 para el cálculo de la longitud de anclaje

Variables	Descripción
α_1	Forma de las barras. 0.7 si c_d en el caso de tipo de anclaje diferente al recto y cuando la barra trabaja a tracción. 1.0 en todos los demás casos
α_2	Recubrimiento del hormigón (límites $\geq 0.7 \leq 1.0$) 1.0 si las barras trabajan a compresión $1 - 0.15 * \frac{c_d - d_b}{d_b}$ para barras en tracción y tipo de anclaje recto $1 - 0.15 * \frac{c_d - 3*d_b}{d_b}$ para barras en tracción y en otro tipo de anclaje.
α_3	Confinamiento con armadura transversal no soldada (límites $\geq 0.7 \leq 1.0$) 1.0 si las barras trabajan a compresión $1 - K * \lambda$ si las barras trabajan a tracción
α_4	Confinamiento con armadura transversal no soldada 0.7
α_5	Confinamiento con pretensado transversal (límites $\geq 0.7 \leq 1.0$) $1 - 0.04 * p$ si las barras trabajan a tracción.
λ	$\frac{\sum A_{st} - \sum A_{st,min}}{A_s}$ $\sum A_{st}$ es el área de armadura transversal a lo largo de l_{bd} $\sum A_{st,min}$ es el área correspondiente a armadura transversal mínima (0.25* A_s para vigas o placas) A_s área de una barra anclada correspondiente a la de diámetro máximo
K	 $K = 0,1$ $K = 0,05$ $K = 0$
p	Tensión (presión) transversal correspondiente al estado límite último a lo largo del l_{bd}
c_d	Se calcula en función de la <div style="display: flex; justify-content: space-between; align-items: flex-start;"> <div style="width: 60%;"> <p>$c_d = \min\left(\frac{a}{2}, c1, c\right)$ Para barras rectas (1)</p> <p>$c_d = \min\left(\frac{a}{2}, c1\right)$ Para barras dobladas(2)</p> <p>$c_d = c$ Para barras en lazo (3)</p> </div> <div style="width: 35%; text-align: center;">  </div> </div>

2.1.1.3 EHE

En España el Instituto Eduardo Torroja desarrolló un completo trabajo experimental (cerca de 4000 ensayos) que sirvieron de base para el planteamiento de la determinación de la longitud de anclaje en la Norma Española (EHE-08, 2008). Los resultados de los ensayos *beam-test* realizados determinaron la existencia de una relación, ecuación (79), entre la tensión de adherencia y el diámetro de la barra.

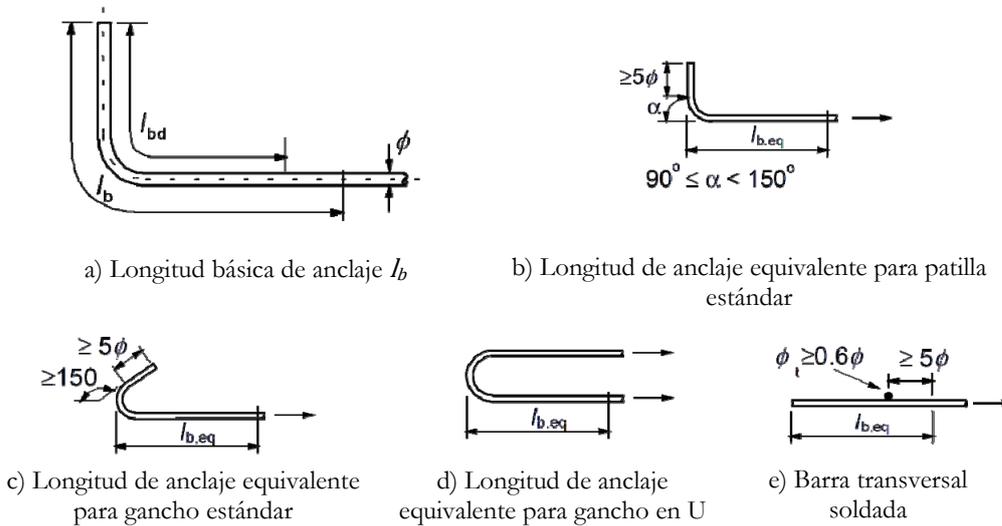


Figura 69. Tipos de anclaje

$$\tau_{m,10} = 80 - 12 * d_b \quad (79)$$

Además se observaron que en rangos normales de longitud de anclaje se llega a una relación lineal entre la ecuación (80) y τ . Esto es, cuando n es igual a 50 el valor de la tensión de adherencia es la mitad que cuando n vale 10.

$$n = \frac{l_{bl}}{d_b} \quad (80)$$

Para valores de n entre el intervalo [10,50] se cumple la ecuación (81) y si n es mayor que 50 se cumple la ecuación (82)

$$\tau_{m,n} = \frac{90 - n}{80} * \tau_{m,10} = \eta * \tau_{m,10} \quad (81)$$

$$\tau_{m,n} = 0.5 * \tau_{m,10} = \eta * \tau_{m,10} \quad (82)$$

Otra relación descubierta es cuando la longitud de anclaje es 50 o más veces el diámetro de la barra, entonces se cumple la ecuación (83).

$$\tau_{bm,50} = 0.5 * \tau_{bm,10} \quad (83)$$

En el caso de prolongación recta se cumple la relación enunciada en la ecuación (84) en la que si se establece el valor de θ a $(f_{ck}/22.5)^\alpha$, entonces se cumple la ecuación (85).

$$f_{yd} * A_s = u * l_b * \tau_m \quad (84)$$

$$f_{yd} * \frac{\pi * d_b^2}{4} = \pi * d_b * n * d_b * \eta * \theta \tau_{m,10} \quad (85)$$

Sustituyendo las expresiones y adoptado $\alpha=0.5$ se llega a las siguientes ecuaciones. Para valores de n entre el intervalo [10,50] se cumple la ecuación (86) y si n es mayor que 50 se cumple la ecuación (87) logrando la actual norma española(88).

$$n = 45 - \sqrt{2025 - \frac{20f_{yd}}{\theta\tau_{m,10}}} \quad (86)$$

$$n = \frac{f_{yd}}{2} * \theta \tau_{m,10} \quad (87)$$

$$l_{db} = l_b * \beta \frac{A_s}{A_{s,real}} \tag{88}$$

En la Tabla 28 se muestran un resumen de las variables empleadas en el cálculo de la longitud de anclaje en la EHE.

Tabla 28. Variables empleadas en la EHE para el cálculo de la longitud de anclaje

Variables	Descripción																		
l_b	Para barras posición I: $l_b = m * d_b^2$ no menor de $\frac{f_{yk}}{20} * d_b$																		
	Para barras posición II: $l_b = 1.4 * m * d_b^2$ no menor de $\frac{f_{yk}}{14} * d_b$																		
β	Factor de reducción determinado por el tipo de anclaje 1.0 para prolongación recta																		
	0.7 Para patilla, gancho y gancho en U si la barra trabaja a tracción y 1.0 si la barra trabaja a compresión 0.7 para barra transversal soldada																		
d_b	Diámetro de la barra (cm)																		
m	Coeficiente numérico, obtenido a partir de los resultados experimentales																		
	<table border="1"> <thead> <tr> <th>$f_c \left(\frac{N}{mm^2} \right)$</th> <th>25</th> <th>30</th> <th>35</th> <th>40</th> <th>≥ 45</th> </tr> </thead> <tbody> <tr> <td>B 400 S</td> <td>12</td> <td>10</td> <td>9</td> <td>8</td> <td>7</td> </tr> <tr> <td>B 500 S</td> <td>15</td> <td>13</td> <td>12</td> <td>11</td> <td>10</td> </tr> </tbody> </table>	$f_c \left(\frac{N}{mm^2} \right)$	25	30	35	40	≥ 45	B 400 S	12	10	9	8	7	B 500 S	15	13	12	11	10
	$f_c \left(\frac{N}{mm^2} \right)$	25	30	35	40	≥ 45													
B 400 S	12	10	9	8	7														
B 500 S	15	13	12	11	10														
m																			
f_{yk}	Límite elástico garantizado del acero (N/mm ²)																		

Suele adoptarse simplifícadamente un valor de tensión de adherencia media a lo largo de toda la barra, aunque realmente los efectos descritos tienen un gradiente importante en distancias situadas a 10/30 veces el diámetro de la barra desde el plano de anclaje (l_b). A la vista de los resultados experimentales, la tensión radial que es capaz de absorber el hormigón puede expresarse como producto de la longitud (l_b) por un parámetro que depende de los recubrimientos mínimo y máximo de la barra (c), por la resistencia a tracción

del hormigón y por un parámetro ligado al diámetro de la barra. Si a esta resistencia se suma la contribución de la armadura transversal, se obtiene finalmente una expresión experimental que acota la carga máxima que puede desarrollar una barra de un diámetro dado, en función de las variables indicadas.

2.1.1.4 FIB

El grupo de trabajo TG4.5 de la *Fédération Internationale du Béton* (FIB) trabaja desde hace tiempo en el análisis del anclaje y la adherencia de barras de armaduras. Basándose en los trabajos de Canbay y Frosch, el grupo ha propuesto una nueva ecuación (89) que proporciona la tensión de trabajo que puede alcanzar una barra anclada, dependiente de los parámetros hasta ahora vistos y amparada por un contraste con una base de datos experimental importante. En la Tabla 29 se muestran las variables de la ecuación propuesta por la FIB.

$$f_{st} = 25 * f_c^{0.25} * \left(\frac{20}{d_b}\right)^{0.2} * \left(\frac{l_s}{d_b}\right)^{0.55} * \left(\frac{c_{min}}{d_b}\right)^{0.33} * \left(\frac{c_{max}}{c_{min}}\right)^{0.1} * (1 + 10 * K_{tr}) \quad (89)$$

Tabla 29. Variables empleadas en la FIB

Variables	Descripción
f_{st}	Tensión de trabajo
f_c	Resistencia a compresión del hormigón en probeta cilíndrica
d_b	Diámetro de la barra
l_s	Longitud de barra anclada
$c_{min} c_{max}$	Recubrimientos de la barra
K_{tr}	Contribución de la armadura transversal

2.1.2 Conjunto de datos

Para la realización de las pruebas para el estudio de la tensión de barra, se ha empleado una base de datos mantenida por la FIB, que en su formato actual contiene 807 vigas. Se han añadido y completado datos a partir de las referencias bibliográficas relacionadas con Tabla 30.

Los distintos planteamientos vistos conducen a expresiones formalmente diferentes y de resultados también dispares para variables iguales. La existencia de una importante base de datos, recopilada por la FIB, permite atestiguar esta afirmación, como se verá a continuación.

La base de datos definitiva se nutre de los ensayos de los investigadores referenciados en la siguiente Tabla 30.

A continuación se han aplicado los siguientes filtros sobre el conjunto total de datos existentes. En primer lugar se descartan todas las vigas en las que la resistencia media a compresión del hormigón (f_c) sean menor o igual a 20 MPa, ya que las normas y códigos actuales establecen una cota inferior de 25 MPa a la resistencia del hormigón. De igual manera se establece como máximo de longitud de barra anclada (l_a) a 2100 mm, el diámetro de la barra (d_b) menor que 26 mm, recubrimiento lateral (c_{so}) como máximo 136 mm, y separación entre barras ($2*c_{sv}$) menor que 294 mm. Por último se establece que todas las vigas deben de tener un factor de rotura de barra, determinado por la ecuación (90), menor que 1.05. La razón de este filtro se basa en la duda que ofrece un ensayo en el que la barra alcanza una tensión muy superior a la de su propia categoría resistente.

$$\text{Factor de rotura} = \frac{S_{su}}{f_y} \quad (90)$$

Tabla 30. Autores incluidos en el conjunto de datos empleado

Autor	Total
Azizinamini, Chisala, Ghosh 1995	7
Azizinamini, Pavel, Hatfield and Ghosh 1997	57
Azizinamini, Stark, Roller, Ghosh 1993	13
Betzle 1980	5
Chamberlin 1956	11
Chamberlin 1958	6
Chinn, Ferguson and Thompson 1955	34
Choi, Haje-Ghaffari, Darwin and McCabe 1990 1991	8
Darwin, Tholen, Idun and Zuo 1995	73
DeVries, Mochle and Hester 1991	10
Eligehausen 1979	7
Ferguson and Breen 1965	28
Hamad 1999	8
Hamad, Mansour 1996	16
Hegger, Burkhardt 1998	9
Hester, Salamizavaregh, Darwin and McCabe 1991 1993	17
Hwang, Lee, Lee 1994	8
Hwang, Leu, Hwang 1996	10
Jeanty, Mitchell, Mirza 1988	4
Kadoriku 1994	33
Kemp 1977	4
Mathey and Watstein 1961	14
Olsen 1990	21
Rehm, Eligehausen 1977	16
Rezansoff, Akanni and Sparling 1993	15
Rezansoff, Konkankar and Fu 1991	34
Tepfers 1973	111
Thompson, Jirsa, Breen and Meinheit 1975	12
Zekany, Neumann, Jirsa and Breen 1981	12
Zuo and Darwin 1998	91
Total	694

Una vez aplicados todos los filtros se dispone de un total de 694 vigas válidas para realizar el estudio, de las cuales 525 se seleccionan para el conjunto de entrenamiento y las restantes 169 para el conjunto test. La selección de estos dos conjuntos la ha realizado los expertos del área. En la Tabla 31 se muestran las variables empleadas y los máximos y mínimos para cada conjunto de datos. En la Figura 70 se muestra gráficamente los intervalos de cada variable.

Tabla 31. Conjunto de datos empleado

		Entrenamiento	Test
Cantidad		525	169
l_s (mm)	Max	2095.5	2032
	Mín	60	120
d_b (mm)	Max	36	36
	Mín	8	8
c_{min} (mm)	Max	77	76
	Mín	8	8
c_{max} (mm)	Max	140	140
	Mín	20	18
f_c (MPa)	Max	114	110
	Mín	20	20
K_{tr}	Max	0.229	0.229
	Mín	0	0
σ_{su}	Max	810	814
	Mín	126	182

A partir de esta base de datos puede analizarse, en primer lugar, el nivel de confianza que aportan las actuales expresiones que desarrollan la longitud de anclaje. Realmente, en los ensayos de adherencia jamás se obtienen longitudes de anclaje, que son un valor fijado en el ensayo. El dato esencial que se obtiene en el ensayo es la tensión máxima que alcanza la barra anclada o, con más propiedad, solapada, puesto que se ensayan barras con solapo en la zona de máxima tracción de una viga. Las condiciones de todos los ensayos corresponden a barras en posición favorable a la adherencia.

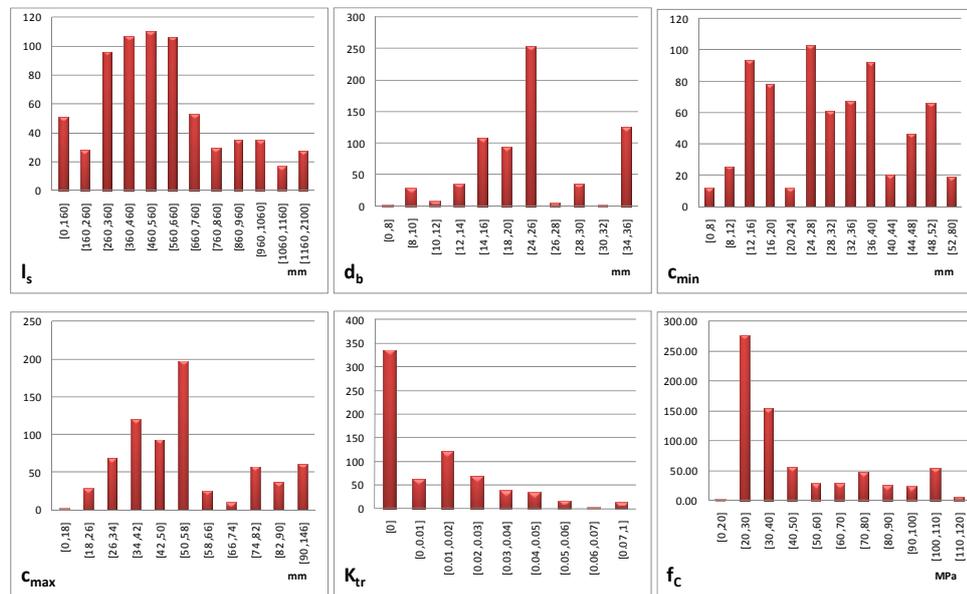


Figura 70. Distribución del conjunto de datos empleado

Se desarrollan, por tanto, las expresiones para cada una de las fórmulas existentes que despeje el valor de la tensión de la barra a partir de la longitud de anclaje, siendo la ecuación de la FIB la única con el resultado ya aislado.

La aplicación de tales expresiones sobre la base de datos produce los resultados dispares que antes se han comentado. Se pueden representar adecuadamente mediante la variable (tensión medida en ensayo)/(tensión dada por la fórmula), de forma que valores menores que 1 resultan inseguros y al contrario los mayores que 1. Para los casos de la EHE y del EC-2, se aportan los resultados relativos a las situaciones de anclaje y de solapo, mientras que los resultados de ACI y FIB formulan ambas situaciones de forma coincidente.

Se observa que la distribución (ver Figura 71) más concentrada en la unidad es la de la FIB, a pesar de haberse retirado los coeficientes de seguridad de las otras fórmulas.

Otra forma de ver el asunto la proporcionan las gráficas (Figura 72) que comparan valor de ensayo y valor predicho por cada expresión. Se presentan a continuación, pudiéndose observar el nivel de confianza y seguridad que proporciona cada fórmula.

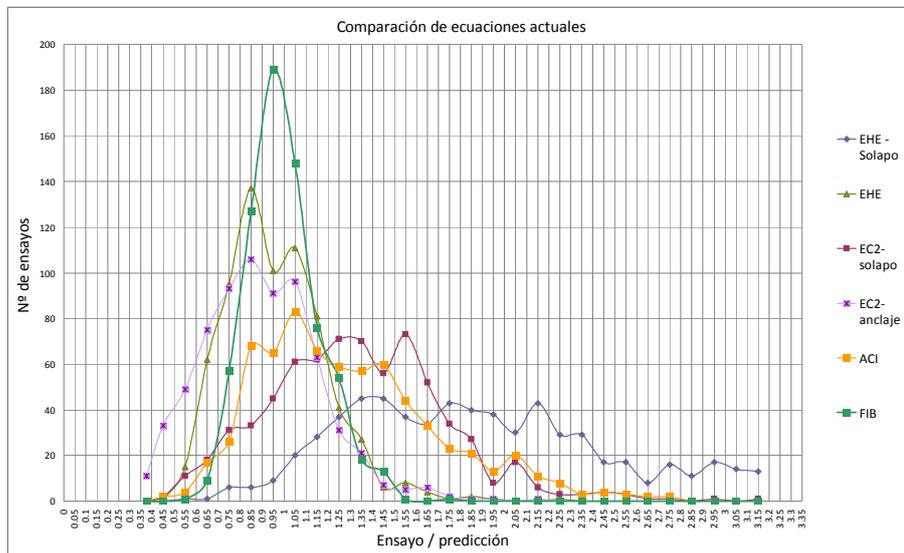


Figura 71. Comparación de ecuaciones actuales

El resumen numérico de la bondad de cada una de las expresiones puede observarse en la Tabla 32, donde el COV es el coeficiente de variación explicado en la ecuación (48) de la página 141, el $\sigma_{\text{test}}/\sigma_{\text{pred}}$ representa el cociente entre el valor de ensayo frente al valor de la fórmula, el R^2 es el cuadrado del coeficiente de correlación de momento del producto Pearson, ecuación (50) página 145. El error cuadrático medio y el error medio, y por último los puntos de demérito calculados según la ecuación (91).

$$PD(\text{solución}) = \sum_{i=1}^n p_i, \quad p_i = \begin{cases} 16, & \frac{\sigma_{test}}{\sigma_{pred}} < 0.5 \\ 8, & 0.5 \leq \frac{\sigma_{test}}{\sigma_{pred}} < 0.67 \\ 4, & 0.67 \leq \frac{\sigma_{test}}{\sigma_{pred}} < 0.85 \\ 2, & 0.85 \leq \frac{\sigma_{test}}{\sigma_{pred}} < 1.0 \\ 0, & 1.0 \leq \frac{\sigma_{test}}{\sigma_{pred}} < 1.3 \\ 3, & 1.3 \leq \frac{\sigma_{test}}{\sigma_{pred}} < 2 \\ 6, & \frac{\sigma_{test}}{\sigma_{pred}} \geq 2 \end{cases} \quad (91)$$

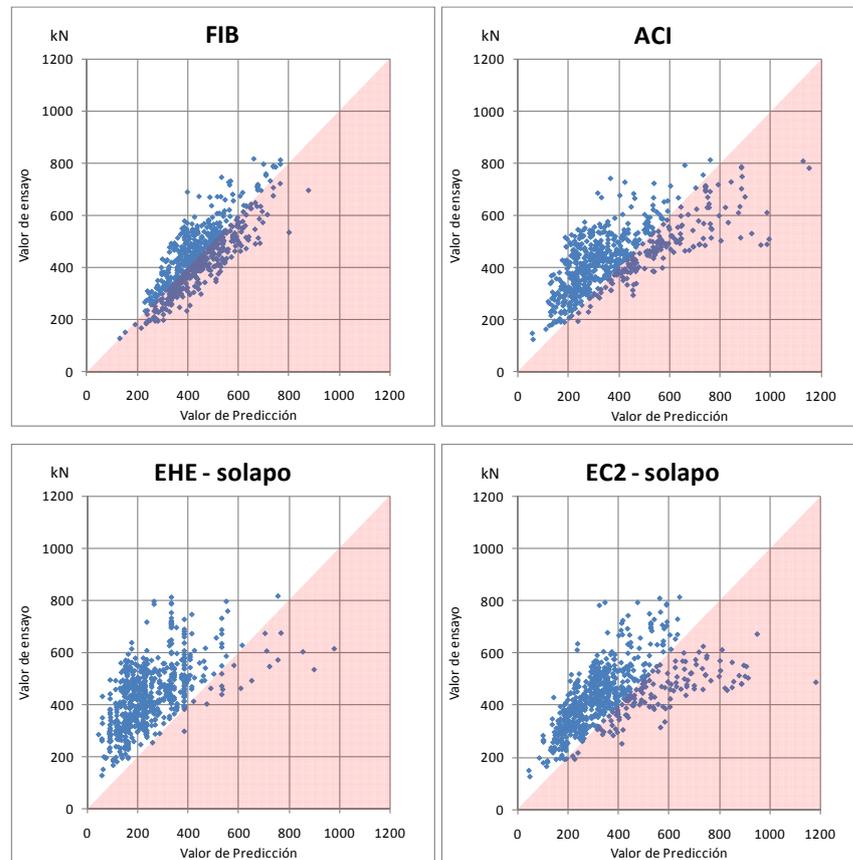


Figura 72. Correlación valor ensayos/valor predicción de las formulaciones actuales

Tabla 32. Resultados de las expresiones actuales.

	EC-2 anclaje	EC-2 solapo	EHE	EHE Solapo	ACI	FIB
COV	30.1499	30.1499	23.8561	39.5273	31.9305	16.4556
$\sigma_{\text{test}}/\sigma_{\text{pred}}$	0.8899	1.3349	0.9618	2.0353	1.2963	0.9997
R²	0.3680	0.3680	0.4450	0.3113	0.5129	0.6609
ECM	52068.6	22896.2	12124.2	48053.2	19839.8	4934.6
EM	146.5	124.1	85.4	194.8	111.5	54.5
P.demerito	2635	1756	1693	2748	1645	1111

Sin duda puede concluirse que es la expresión no normativa de la FIB la que ofrece mejores resultados. Realmente la FIB introduce unos límites a tal fórmula, de acuerdo al rango de datos de la base experimental, lo que mejora algo más el balance. Tales límites son los siguientes descritos en la ecuación (92).

$$0.5 < \frac{c_{min}}{db} < 3.0 \text{ (límites de los datos)}$$

$$\frac{C_{max}}{C_{min}} < 5 \text{ (límites de los datos)} \tag{92}$$

$$K_{tr} = \sum \frac{A_{sw}}{l_s * db * n_s} < 0.04 \text{ como máximo el } K_{tr} \text{ contribuye con } 0.04.$$

Las extraordinarias diferencias entre expresiones son un buen motivo para aplicar la metodología desarrollada, a lo que invita también el hecho de que todas las expresiones tienen, como se ha comentado anteriormente, origen experimental y ninguna guarda coherencia dimensional. Por tanto, este caso práctico va a pretender obtener nuevas o mejoradas fórmulas que sean de utilidad para predecir con aproximación y seguridad la tensión que puede desarrollar una barra por adherencia. Cabe indicar que resulta útil, a efectos

normativos, disponer de la posibilidad de aislar la variable correspondiente a la longitud anclada o solapada, por lo que en varios casos se guiará el proceso en esa dirección.

2.2 Aplicación del Método

El objetivo perseguido en este ejemplo de aplicación es la construcción de un modelo de comportamiento que satisfaga las exigencias de los expertos y consiga determinar la longitud de anclaje de forma más segura.

Para lograr este objetivo y de acuerdo con los expertos del área de construcción se ha establecido un conjunto de ecuaciones (93)(94)(95) como punto de partida para realizar la optimización mediante el algoritmo desarrollado.

$$\sigma_{su} = Rama_1^{Rama_2} * f_c^{Rama_3} * \left(\frac{ls}{db}\right)^{Rama_4} * \left(\frac{cmin}{db}\right)^{Rama_5} * \left(\frac{cmax}{cmin}\right)^{Rama_6} * (1 + Rama_7 * Ktr)^{Rama_8} \quad (93)$$

$$\sigma_{su} = ls^{Rama_1} * Rama_2 * (1 + Rama_3 * Ktr)^{Rama_4} \quad (94)$$

$$\sigma_{su} = Rama_0 * f_c^{Rama_1} * \left(\frac{Rama_2}{db}\right)^{Rama_3} * \left(\frac{ls}{db}\right)^{Rama_4} * \left(\frac{cmin}{db}\right)^{Rama_5} * \left(\frac{cmax}{cmin}\right)^{Rama_6} * (1 + Rama_7 * Ktr)^{Rama_8} \quad (95)$$

Para la resolución de este problema se emplean los módulos 3 (Módulo de coeficiente de seguridad), 4 (Módulo de restricciones en funciones) y 5 (Módulo de optimización de funciones conocidas (normativas)) definidos en el apartado “Sistema basado en conocimiento aplicado a la Programación Genética”. En primer lugar es necesario definir la evaluación de los individuos en la función de ajuste. En el Módulo de coeficiente de seguridad se deben definir los parámetros k y l_{bias} definidos en la ecuación (32). Después de realizar diversas pruebas se estableció el valor de l_{bias} a 1.0 y en la ecuación (96) se

muestra el valor de “ p_i ” inspirado en el empleo de la técnica de “puntos de demérito”.

$$p_i = \begin{cases} 16, & \frac{\sigma_{test}}{\sigma_{pred}} < 0.5 \\ 8, & 0.5 \leq \frac{\sigma_{test}}{\sigma_{pred}} < 0.67 \\ 4, & 0.67 \leq \frac{\sigma_{test}}{\sigma_{pred}} < 0.85 \\ 2, & 0.85 \leq \frac{\sigma_{test}}{\sigma_{pred}} < 1.0 \\ 1, & 1.0 \leq \frac{\sigma_{test}}{\sigma_{pred}} < 1.3 \\ 3, & 1.3 \leq \frac{\sigma_{test}}{\sigma_{pred}} < 2 \\ 6, & \frac{\sigma_{test}}{\sigma_{pred}} \geq 2 \end{cases} \quad (96)$$

Para la realización de las pruebas se ha establecido como configuración por defecto la listada en la Tabla 33. Se han elegido estos parámetros debido a que han sido los que mejores resultados han dado en las pruebas iniciales. Cabe destacar que los datos de entrada al algoritmo no han sido normalizados. De esta manera no se necesita aplicar ninguna conversión de datos sobre las soluciones obtenidas, pudiéndolas aplicar directamente.

Por defecto, se ha elegido como operadores o nodos no terminales la suma, resta, producto y división protegida. Para los nodos terminales se emplean las variables procedentes del conjunto de datos (ls , d_b , c_{min} , c_{max} y f_i), y los enteros comprendidos en el intervalo $[-10, 10]$. En la Tabla 34 se muestran las restricciones aplicadas en las ecuaciones de partida comentadas anteriormente. Se puede observar que para la ecuación (93) se establecen tres tipos de restricciones denominadas (A, B y C) para la ecuación (94) se establece la restricción denominada D y por último para la ecuación (95) se establece la restricción E.

Tabla 33. Parámetros empleados

Parámetro	Valores por Defecto	Otros valores
Tamaño Población	1000	
Tasa de Cruces:	80%	
Tasa de Selección no Terminal:	90%	
Probabilidad Mutación:	20%	
Algoritmos:	Selección: Torneo Creación: Intermedio Mutación: Subárbol	
Estrategia elitista	Si	
Parsimonia	0	0.0001, $1*10^{-6}$ ó $1*10^{-9}$
Altura Inicial	4	5
Altura Máxima	6	7
Altura Máxima mutación	4	5

Tabla 34. Restricciones aplicadas

Ecuación	Restr.	R ₁	R ₂	R ₃	R ₄	R ₅	R ₆	R ₇	R ₈	R ₉
(93)	A	l_s, d_b c_{min} c_{max}, f_c	Cte 2 dec.	Cte 2 dec.	Cte 2 dec.	Cte 2 dec.	Cte 2 dec.	Cte 2 dec.	Cte 2 dec.	-
(93)	B	d_b, c_{min} c_{max}, f_c	Cte 2 dec.	Cte 2 dec.	Cte 2 dec.	Cte 2 dec.	Cte 2 dec.	Cte 2 dec.	Cte 2 dec.	-
(93)	C	d_b, c_{min} c_{max}, f_c	Cte 2 dec.	Cte 2 dec.	0.5	Cte 2 dec.	Cte Ent.	Cte 2 dec.	Cte 2 dec.	-
(94)	D	Cte 2 dec.	d_b, c_{min} c_{max}, f_c	Cte 2 dec.	Cte 2 dec.	-	-	-	-	-
(95)	E	Cte 2 dec.	Cte 2 dec.	Cte 2 dec.	Cte 2 dec.	Cte 2 dec.	Cte 2 dec.	Cte 2 dec.	Cte 2 dec.	Cte 2 dec.

2.2.1 Resultados

Este conjunto de escenarios se activa hasta desarrollar más de 4500 ejecuciones. Los resultados se analizan fundamentalmente a través de los indicadores ya mencionados (COV, max, min, R^2 , ECM, EM y PD). Una vez seleccionado un conjunto de ellos (para cada uno de los escenarios, entendidos como pares ecuación/restricción), se estudia la expresión para eliminar aquellas que conducen a divisiones protegidas en el denominador. En este

conjunto de resultados de PG, además, no se tienen en cuenta las ecuaciones de gran complejidad, salvo que sus resultados permitan aceptar que se ha conseguido un “laboratorio virtual”, situación que no se produce.

Las ecuaciones más sobresalientes del estudio realizado se presentan a continuación, entre las que no se incluyen las derivadas de la PG clásica al no proporcionar mejoras significativas.

$$PG9RSC4\sigma_{su} = 27 * fc^{0.27} * \left(\frac{29}{db}\right)^{0.23} * \left(\frac{ls}{db}\right)^{0.47} * \left(\frac{cmin}{db}\right)^{0.16} * \left(\frac{cmax}{cmin}\right)^{0.13} * (1 + 286 * Ktr)^{0.11} \quad (97)$$

$$PG8R1\sigma_{su} = \left(\frac{92.37}{cmin} - \frac{ls}{8 * cmax} + 16\right) * fc^{0.28} * \left(\frac{ls}{db}\right)^{0.6} * \left(\frac{cmin}{db}\right)^{0.34} * \left(\frac{cmax}{cmin}\right)^{0.05} * (1 + 107.52 * Ktr)^{0.18} \quad (98)$$

$$PG8v2R5\sigma_{su} = \left(\frac{cmax}{36} - \frac{fc}{36} + \frac{150}{cmin} + 19.294\right) * fc^{0.29} * \left(\frac{ls}{db}\right)^{0.49} * \left(\frac{cmin}{db}\right)^{0.29} * \left(\frac{cmax}{cmin}\right)^{0.04} * (1 + 239.29 * Ktr)^{0.12} \quad (99)$$

$$PG7v3F1\sigma_{su} = Rama_1 * fc^{0.41} * \left(\frac{ls}{db}\right)^{0.5} * \left(\frac{cmin}{db}\right)^{0.34} * \left(\frac{cmax}{cmin}\right)^{0.1} * (1 + 157 * Ktr)^{0.14} \quad (100)$$

$$Rama_1 = \frac{54}{5 * \left(\frac{fc}{5} + 6\right)} - \frac{fc}{5 * \left(\frac{9}{cmin} + 5\right)} + \frac{66}{cmin} + \frac{9}{db} + 13$$

$$PG7v3F2\sigma_{su} = \left(\frac{48}{cmin} - \frac{fc - 44}{\frac{cmin}{48} + 20} + 9\right) * fc^{0.5} * \left(\frac{ls}{db}\right)^{0.5} * \left(\frac{cmin}{db}\right)^{0.34} * \left(\frac{cmax}{cmin}\right)^{0.01} * (1 + 173 * Ktr)^{0.14} \quad (101)$$

$$PG7v3F4\sigma_{su} = \left(\frac{fc}{\frac{cmax}{38} - 24} + \frac{38}{cmin} + 11\right) * fc^{0.5} * \left(\frac{ls}{db}\right)^{0.5} * \left(\frac{cmin}{db}\right)^{0.33} * \left(\frac{cmax}{cmin}\right)^{0.16} * (1 + 225 * Ktr)^{0.12} \quad (102)$$

$$PG7v3F8\sigma_{su} = \left(\frac{70}{cmin} + \frac{70}{db} + 24\right) * fc^{0.23} * \left(\frac{ls}{db}\right)^{0.5} * \left(\frac{cmin}{db}\right)^{0.25} * \left(\frac{cmax}{cmin}\right)^{0.07} * (1 + 148 * Ktr)^{0.15} \quad (103)$$

$$PG7v3F10\sigma_{su} = \left(\frac{51}{5 * db} + 5\right)^2 * fc^{0.23} * \left(\frac{ls}{db}\right)^{0.5} * \left(\frac{cmin}{db}\right)^{0.13} * \left(\frac{cmax}{cmin}\right)^{0.11} * (1 + 227 * Ktr)^{0.12} \quad (104)$$

$$PG7v3F11\sigma_{su} = \left(\frac{23}{db} + 8\right)^{1.52} * fc^{0.24} * \left(\frac{ls}{db}\right)^{0.5} * \left(\frac{cmin}{db}\right)^{0.14} * \left(\frac{cmax}{cmin}\right)^{0.12} * (1 + 173 * Ktr)^{0.14} \quad (105)$$

$$PG7v3F12\sigma_{su} = \left(\frac{89}{db} + \frac{447}{19}\right) * fc^{0.25} * \left(\frac{ls}{db}\right)^{0.5} * \left(\frac{cmin}{db}\right)^{0.16} * \left(\frac{cmax}{cmin}\right)^{0.13} * (1 + 121 * Ktr)^{0.15} \quad (106)$$

$$PG7v3F13\sigma_{su} = \left(\frac{49}{\frac{db}{fc} + 7}\right)^{0.88} * fc * \left(\frac{ls}{db}\right)^{0.5} * \left(\frac{cmin}{db}\right)^{0.13} * \left(\frac{cmax}{cmin}\right)^{0.1} * (1 + 256 * Ktr)^{0.12} \quad (107)$$

$$PGcc1\sigma_{su} = l_s^{0.5} * \left(\frac{5 * \left(\frac{fc}{db} + \frac{5}{db} + 1 \right)}{db} + \frac{5 * \left(\frac{5 * (fc + 2)}{2} + \frac{5 * \left(\frac{cmax}{3} + cmin + 5 \right)}{db} + 6 \right)}{db} + \frac{cmax * \left(\frac{fc}{8} + \frac{fc}{cmin} \right)}{2 * cmin^2} + 8 \right) * (1 + 401 * Ktr)^{0.11} \quad (108)$$

$$PGcc3\sigma_{su} = l_s^{0.5} * \left(- \left(\frac{10 * fc}{db * (db + 10)} + 2 \right) * \left(\frac{db}{9} + \frac{10}{cmax} - 9 \right) \right) * (1 + 164.25 * Ktr)^{0.15} \quad (109)$$

$$PGcc6\sigma_{su} = l_s^{0.5} * \left(\left(\frac{75}{db} + \frac{19}{7} \right) * \left(\frac{fc}{5 * db} + \frac{cmax * fc}{360 * db} + 2 \right) \right) * (1 + 165 * Ktr)^{0.15} \quad (110)$$

Los indicadores de ajuste se presentan en la Tabla 35, quedando patente la sustancial mejora que se produce, en todos los casos seleccionados, respecto de la expresión propuesta por la FIB.

Tabla 35. Resultados obtenidos para el total de ensayos

	COV	Ens / Form	R2	ECM	EM	PD
FIB	16.456	1.000	0.661	4934.61	54.51	1111
PG9RSC4	14.508	1.017	0.714	4035.00	49.91	860
PG8R1	13.946	1.040	0.732	3998.42	48.82	755
PG8v2R5	14.198	1.024	0.727	3916.84	48.89	819
PG7v3F1	13.958	1.037	0.736	3953.45	48.83	753
PG7v3F2	13.952	1.039	0.736	3980.27	48.71	745
PG7v3F4	13.972	1.038	0.733	3958.59	48.46	759
PG7v3F8	14.405	1.041	0.713	4316.40	51.09	758
PG7v3F10	14.405	1.045	0.712	4368.60	51.50	745
PG7v3F11	14.335	1.046	0.716	4313.27	51.12	749
PG7v3F12	14.422	1.043	0.710	4353.01	51.11	753
PG7v3F13	14.305	1.033	0.719	4109.45	50.29	782
PGcc1	14.438	1.022	0.714	4067.38	48.88	837
PGcc3	15.219	1.052	0.669	5030.92	55.52	767
PGcc6	14.616	1.025	0.711	4129.55	49.82	826

Se representan todas ellas en la siguiente gráfica (Figura 73), donde se observa el grado de aproximación alcanzado.

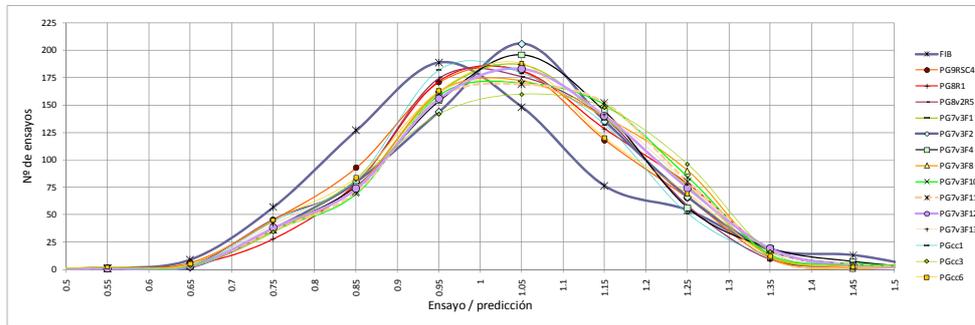


Figura 73. Predicciones de las soluciones alcanzadas.

Algunas de las expresiones destacan por diferentes aspectos. La PG9RSC4 es una simple mejora de la ecuación de la FIB, obtenida mediante el mejor ajuste de los exponentes y constantes. La PG8R1 tiene el máximo en 1, aunque no permite despejar el valor de la longitud ya que se forma a partir de la ecuación de la FIB, aunque modificando la constante inicial por una función cualquiera de las variables del problema. Para evitar esta situación se condiciona la búsqueda proponiendo una primera función libre, no dependiente de la longitud, y un ajuste de coeficientes para el resto de parámetros. Surge así la ecuación PG8v2R5. La marcada tendencia que exhibe el exponente de (l_s/db) hacia el valor 0.5 (lo que facilita mucho la expresión en l_s), induce un nuevo conjunto de ejecuciones en las que se fija esa constante. Se consiguen de ese modo las fórmulas de la serie PG7vFj, entre las que destacan la PG7vF2 y la PG7vF4, la primera de ellas con una magnífica distribución. El último grupo destacable es aquel en el que se permite la aparición de una función libre (sin l_s) que multiplica a l_s con exponente constante y al término clásico de contribución de la armadura mejorado con constantes. Se hallan así las expresiones PGcc3 y PGcc6, la primera con pocos valores inseguros y la segunda con fuerte concentración alrededor de la unidad.

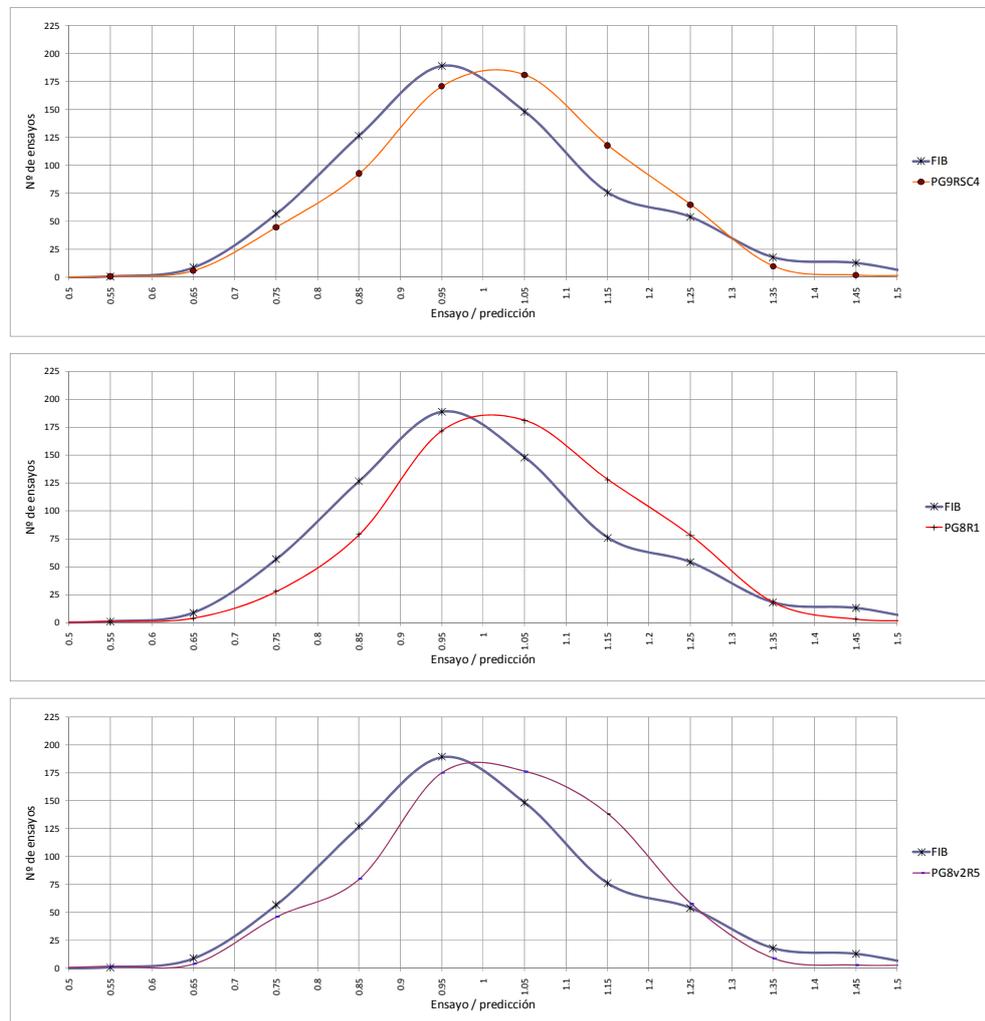


Figura 74. Predicciones de las ecuaciones PG9RSC4, PG8R1 y PG8v2R5 comparándolas con la FIB

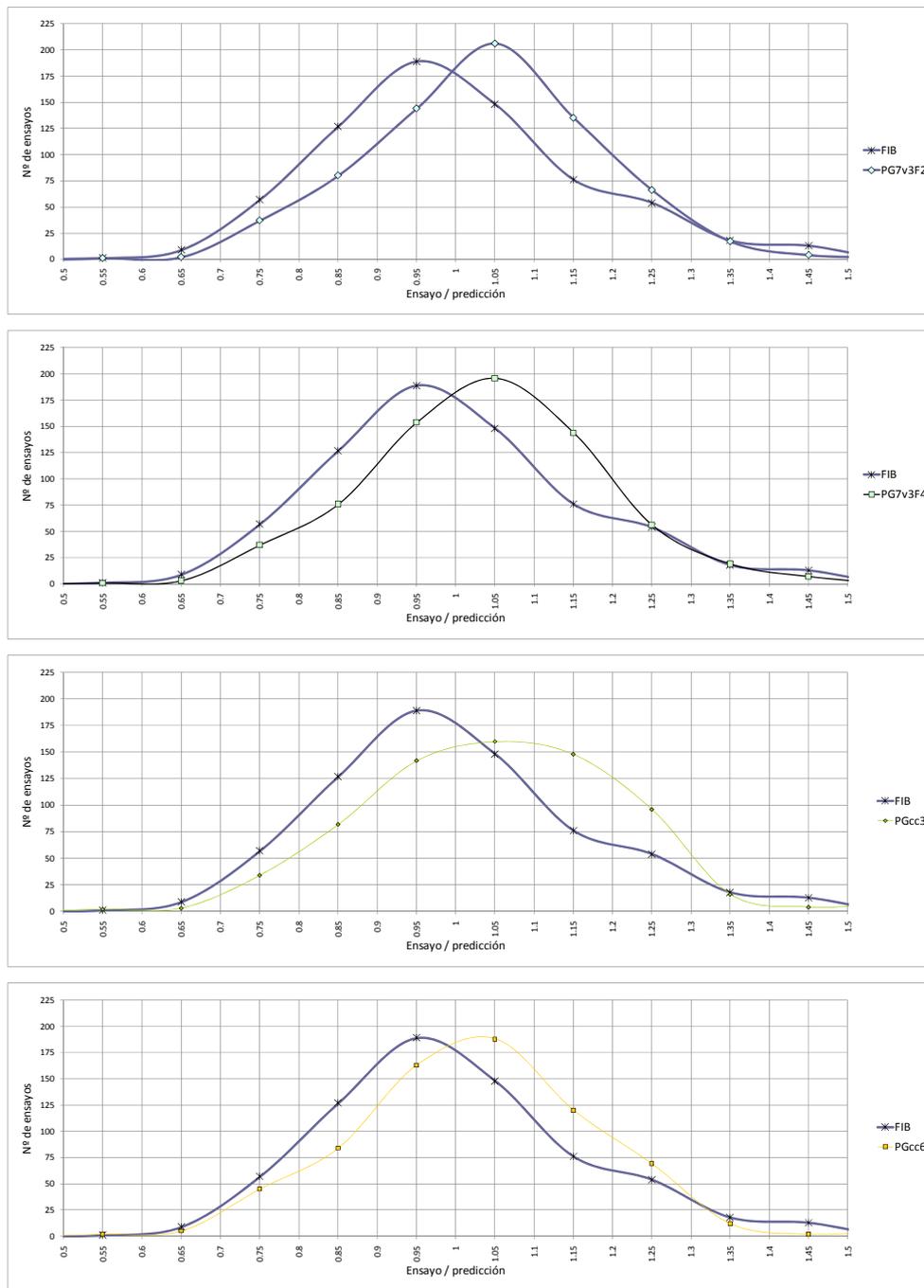


Figura 75. Predicciones de las ecuaciones PG7v3F2, PG7v3F4, PGcc3 y PGcc6 comparándolas con la FIB.

Se comparan a continuación las tensiones que pueden desarrollarse para algunas variables concretas, comparadas en dos de las ecuaciones halladas con las deducidas de la FIB. Se observa la similitud del planteamiento, incluso para ecuaciones que no nacen de la estructura de la FIB.

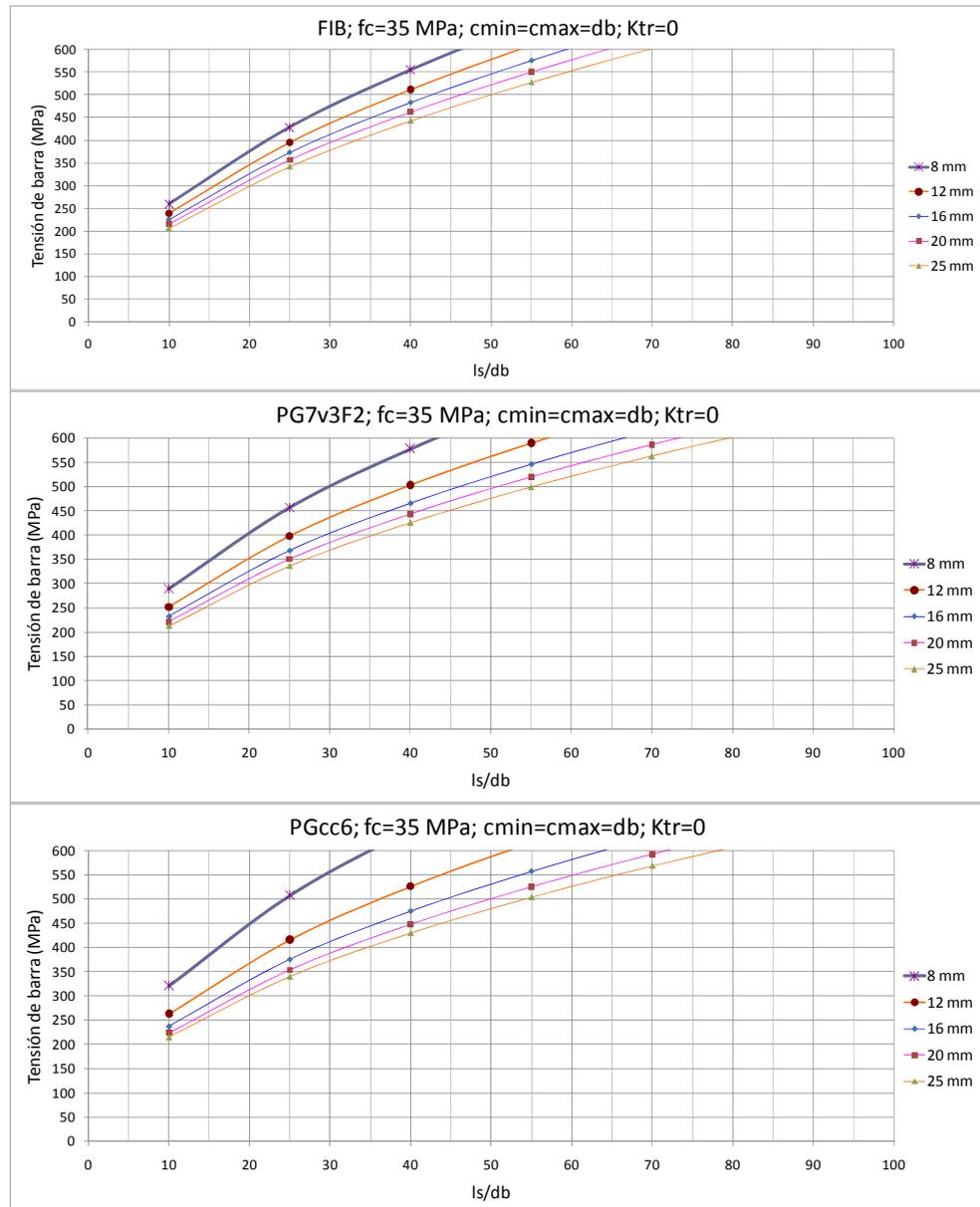


Figura 76 Tensión de barra frente l_s/d_b de la FIB, PG7v3F2 y PGcc6

A la vista de lo expuesto anteriormete, puede recomendarse adoptar la expresión PG7v3F2 como una buena ecuación para captar el comportamiento adherente de las armaduras pasivas en el hormigón. La expresión que recogería el diseño de los anclajes/solapos sería la presentada en la ecuación (111).

$$l_{bd} = \frac{d_b * \left(\frac{d_b}{c_{\min}}\right)^{0.68} * \left(\frac{c_{\min}}{c_{\max}}\right)^{0.2} * \sigma_{sd}^2}{f_{cd} * A^{0.28} * B^2} \quad (111)$$

donde σ_s es la tensión que alcanzará la barra en Estado Límite Último (ELU), y los parámetros A y B se determinan siguiendo la ecuación (112).

$$A = (1 + 173 * K_{tr})$$

$$B = \frac{1}{d_b} * \left(\frac{48}{c_{\min}/d_b} - \frac{f_{cd} - 44}{(c_{\min} + 20)/d_b} \right) + 9 \quad (112)$$

Parte de los resultados alcanzados en el estudio del fenómeno de adherencia entre el hormigón y el acero se ha presentado en el congreso IECC'5 (Rabuñal et al., 2008). También se ha escrito un capítulo del libro "TIC en Ingeniería Civil" (Pérez & Carro 2007).

3 Fluencia del hormigón estructural

3.1 Descripción

Se define la fluencia como el aumento de deformación que sufre el hormigón cuando se encuentra bajo una carga constante. Para su determinación se observa la evolución de las deformaciones que experimenta una probeta sometida a una tensión constante de compresión a lo largo del tiempo.

Se observa examinando la evolución de las deformaciones que experimenta una pieza de hormigón (probeta) sometida a una tensión de compresión constante a lo largo del tiempo (ver Figura 77). En este tipo de ensayos se producen, en primer lugar, una deformación inicial elástica instantánea (Figura 77, variable ε_i en el instante t_1), y a continuación una deformación creciente que tiende a un valor asintótico (en un instante t_x la deformación total es la suma de ε_i más la deformación producida por la fluencia ε_f)

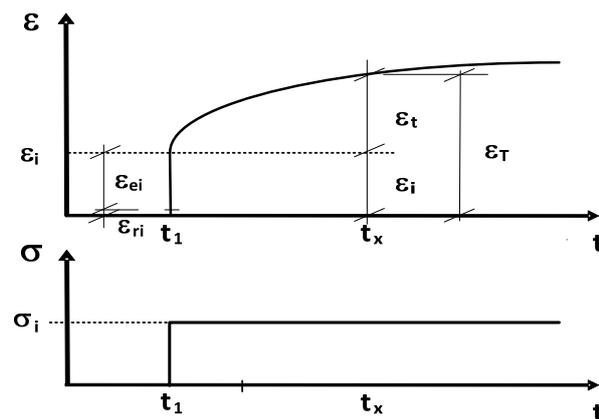


Figura 77. Deformación por fluencia

Para cuantificar la deformación de fluencia se resta la deformación elástica de la deformación total registrada. Habitualmente se define como deformación elástica la producida durante la aplicación de la carga, si bien en la deformación instantánea se incluye de igual manera un componente plástico. Las demás deformaciones se atribuyen a la fluencia.

Principalmente se identifican dos tipos diferentes de fluencia, denominados básica y de secado, siendo la suma de las dos la fluencia total. Se denomina fluencia básica a la deformación producida bajo carga constante en ausencia de gradientes de humedad. En cambio, la fluencia de secado, vinculada al efecto Pickett, se produce cuando el hormigón soporta cargas constantes y al mismo tiempo se está secando.

Se debe resaltar que la fluencia no es un fenómeno reversible en su totalidad. En efecto, tras eliminar la carga aplicada, instantáneamente las deformaciones que acumula el hormigón disminuirán en una cantidad similar a la deformación elástica inicial. En la Figura 78 se muestra gráficamente el fenómeno. La recuperación disminuye con la edad del hormigón en el momento de la descarga, ya que se rigidiza por envejecimiento. A continuación se va produciendo una recuperación parcial de la fluencia hasta llegar a un valor asintótico de deformación. Aproximadamente la fluencia básica se recupera en un 25%. En cambio la fluencia de secado es imposible de recuperar. Por norma general, si se somete el hormigón a carga constante durante 25 años, puede asumirse que 1/4 de la fluencia total se produce en las primeras semanas tras la aplicación de la carga. La mitad ocurriría a los tres meses siguientes y alrededor del 75% cuando se cumpliera un año.

Entre la fluencia y la tensión aplicada existe una relación lineal siempre y cuando la ratio (tensión aplicada)/(resistencia a compresión) no supere el

índice 0.5 ó 0.6. A partir de dicho nivel se incrementan tanto el número como el tamaño de las microfisuras, desarrollándose el fenómeno de la fluencia bajo una acusada no linealidad con la carga. Para valores entre 0.8 y 0.9 se llega a producir la rotura del hormigón por cansancio.

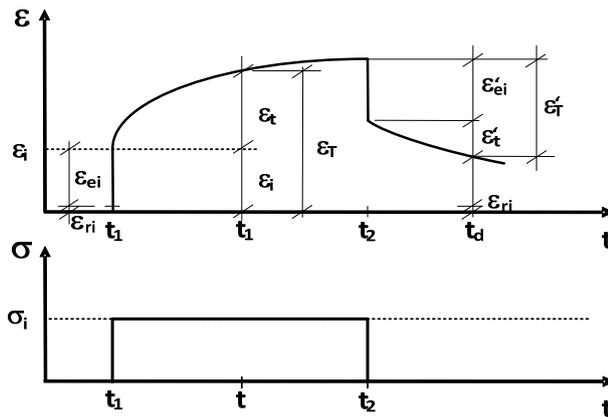


Figura 78. Fluencia y recuperación de fluencia

En la actualidad, existe un conjunto de parámetros que caracterizan las deformaciones por fluencia. En primer lugar la función de fluencia $J(t, t')$, ecuación (113) que describe la deformación total, es decir, la suma de ϵ_{α} y ϵ_{α} en el tiempo t , producida por la tensión unitaria aplicada en el instante de tiempo t' . La función de fluencia está expresada en MPa^{-1} , estando su valor entre 30 y $300 \mu\epsilon/\text{MPa}$ (Bazant & Baweja, 2000). Este parámetro es recomendado por el comité 209 de la ACI puesto que no es necesario separar las deformaciones iniciales de las producidas por la fluencia (ACI Committee 209, 1982)

$$J(t, t') = \frac{\text{deformación elástica} + \text{deformación por fluencia}}{\text{tensión aplicada}} \quad (113)$$

Además de la función de fluencia existe otro parámetro denominado coeficiente de fluencia, ecuación (114), que permite definir la fluencia. La deformación diferida es proporcional a la instantánea en los casos en los que la tensión esté entre el 0.4 y el 0.6 de la resistencia. Partiendo de este hecho, empleando el coeficiente de fluencia φ , es posible relacionar la deformación por fluencia ε_{cc} y la deformación instantánea ε_{ci} . Este parámetro presenta tres ventajas: no tiene unidades, considera las propiedades elásticas del hormigón y por último es independiente de la tensión aplicada.

$$\varphi = \frac{\text{deformación por fluencia}}{\text{deformación elástica}} \quad (114)$$

En la ecuación (115) se muestra cómo es posible relacionar deformación por fluencia ε_{cc} ($\mu\varepsilon$) y la deformación instantánea o elástica ε_{ci} ($\mu\varepsilon$), siendo J ($\mu\varepsilon/\text{MPa}$) la función de fluencia, E_0 (MPa) el módulo de deformación longitudinal del hormigón a la edad de carga t_0 (días), E_{28} (MPa) el módulo de elasticidad a los 28 días, $\sigma(t_0)$ (MPa) la tensión unitaria aplicada a la edad t_0 , y $\varphi(t, t_0)$ el coeficiente de fluencia del hormigón a la edad t (días) y bajo una tensión aplicada a la edad t_0 .

$$\begin{aligned} \varepsilon_{\sigma} &= \varepsilon_{ci} + \varepsilon_{cc} \\ \varepsilon_{\sigma} &= \sigma(t_0) \left[\frac{1}{E_0} + \frac{\varphi}{E_{28}} \right] \quad , \quad J = \sigma(t_0) \left[\frac{1}{E_0} + \frac{\varphi}{E_{28}} \right] \\ \varepsilon_{cc} &= \sigma(t_0) \left[\frac{\varphi}{E_{28}} \right] \quad \Rightarrow \quad \varepsilon_{cc} = \varphi \varepsilon_{ci} \end{aligned} \quad (115)$$

Además de las dos relaciones explicadas existen otras, documentadas en la literatura científica, en las que se relacionan otras variables con la fluencia.

La fluencia en el hormigón comienza a considerarse a principios del siglo XX. El primer estudio conocido sobre fluencia pertenece a Woolson (Woolson, 1905). A mediados del siglo XX, Troxel et al, (Troxell, Raphael & Davis, 1958) fueron los primeros en relacionar la humedad durante el proceso de curado con la fluencia del hormigón.

3.1.1 Modelos normativos actuales

Existen diferentes aproximaciones para predecir las deformaciones de fluencia, generalmente plasmadas a través de códigos o normas. El más extendido es la Normativa ACI 209 desarrollado por el Committee 209 de la American Concrete Institute (ACI), así como el propuesto por el Comité Europeo de Béton (CEB), resumido en el modelo CEB MC90 (Müller & Hilsdorf, 1990) del año 1990. Este modelo ha sido adoptado como la propuesta para el Eurocode 2 (EC-2) (Eurocode 2, 2002). Otros de los modelos existentes hoy en día han sido, por ejemplo, el desarrollado por Bazant y denominado B3 (Bazant & Baweja 1995) y por último el modelo desarrollado por Gardner y Lockman denominado GL2000 (Lockman, 2000). Estos modelos están desarrollados a partir de resultados experimentales, sostenidos en cierta medida (especialmente el B3) por planteamientos teóricos. Se han verificado mediante ensayos contrastados, específicos o agrupados en bases uniformes. Probablemente la base experimental más sólida es la puesta a punto por la RILEM (2009), a partir del trabajo del comité TC 107, que incluye ensayos comparables entre distintos hormigones y en los que intervienen las mismas variables relacionadas con la fluencia. Se comparan a continuación los métodos de ACI y CEB-MC90.

3.1.1.1 ACI 209 Método del 1978

La normativa ACI 209 establece como modelo de fluencia el enunciado en la ecuación (116). En la Tabla 36 se muestran las variables empleadas en este modelo.

$$J(t, t_0) = \frac{1 + \phi}{E_{cmto}} \quad (116)$$

$$\phi = 2.35 \gamma_1 \gamma_2 \gamma_3 \gamma_4 \gamma_5 \gamma_6 \frac{(t - t_0)^{0.6}}{(t - t_0)^{0.6} + 10}$$

Tabla 36. Variables empleadas en el Modelo ACI 209

Variables	Descripción
t	Edad del hormigón en carga (en días)
a	Contenido de aire (expresado en decimal)
CM	Método de curado
Ψ	Proporción de árido fino frente total (en peso)
E_{cmto}	Módulo de elasticidad del hormigón cuando comienza la el proceso de carga (en MPa)
h	Humedad relativa (expresado en decimal)
s	Altura de cono del hormigón fresco (mm)
(V / S)	Ratio volumen/superficie (mm) del elemento
γ_1	Corrección en función de la edad de carga Para $t_0 \geq 3$ días $\gamma_1 = 1.25t_0^{-0.094}$ Para $t_0 \geq 7$ días $\gamma_1 = 1.25t_0^{-0.118}$
γ_2	Corrección en función de la humedad $\gamma_2 = 1.27 - 0.67h$ Para $h \geq 0.40$

Variables	Descripción												
	Corrección en función del tamaño Si el $V/S \leq 37.5$ mm												
	<table border="1"> <thead> <tr> <th>V/S</th> <th>12.5</th> <th>18.8</th> <th>25</th> <th>31.2</th> <th>37.5</th> </tr> </thead> <tbody> <tr> <td>γ_3</td> <td>1.3</td> <td>1.17</td> <td>1.11</td> <td>1.04</td> <td>1</td> </tr> </tbody> </table>	V/S	12.5	18.8	25	31.2	37.5	γ_3	1.3	1.17	1.11	1.04	1
V/S	12.5	18.8	25	31.2	37.5								
γ_3	1.3	1.17	1.11	1.04	1								
γ_3	Si V/S esta en el intervalo 37.5 - 95 mm $\gamma_3 = 1.14 - 0.00368 (V / S)$ (t-t0) \leq 365 días $\gamma_3 = 1.10 - 0.00268 (V / S)$ (t-t0) $>$ 365 días												
	Si el $V/S \geq 95$ mm $\gamma_3 = \frac{2}{3} \left[1 + 1.13e^{(1-0.0213(V/S))} \right]$												
γ_4	$\gamma_4 = 0.82 + 0.00264s$												
γ_5	$\gamma_5 = 0.82 + 0.24\Psi$												
γ_6	$\gamma_6 = 0.46 + 9a \geq 1.0$												

3.1.1.2 Modelo CEB MC90

El modelo CEB MC90 ha desarrollado la ecuación (117). En la Tabla 37 se muestran las variables empleadas en este modelo.

$$J(t, t_0) = \frac{1}{E_{cm_{t_0}}} + \frac{\phi}{E_{cm_{28}}} \quad (117)$$

$$\phi = \phi_{RH} \beta(f_{cm_{28}}) \beta(t_0) \left(\frac{t - t_0}{t - t_0 + \beta_H} \right)$$

Tabla 37. Variables empleadas en el Modelo MC90

Variables	Descripción
t	Edad del hormigón en carga (en días)
f_{cm28}	Resistencia del hormigón a los 28 días
E_{cm28}	Módulo de elasticidad el hormigón a los 28 días (MPa)
E_{cm0}	Módulo de elasticidad del hormigón cuando comienza la el proceso de carga (en MPa)
h	Humedad relativa (expresado en decimal)
(V / S)	Ratio volumen/superficie (mm) del elemento
	Corrección relativa a la humedad
ϕ_{RH}	$\phi_{RH} = 1 + \frac{t - h}{0.1(2V / S)^{1/3}}$
	Corrección relativa a la humedad
β_H	$\beta_H = 1.5 \left[1 + (1.2h)^8 \right] (2V / S) + 250 < 1500 \text{ dias}$
	Corrección relativa a la resistencia del hormigón
$\beta(f_{cm28})$	$\beta(f_{cm28}) = \frac{16.8}{\sqrt{f_{cm28}}}$
	Corrección relativa a la edad de la carga
$\beta(t_0)$	$\beta(t_0) = \frac{1}{0.1 + t_0^{0.2}}$

3.2 Aplicación del Método

El objetivo perseguido en este ejemplo es predecir correctamente la serie temporal a partir únicamente de las características del hormigón. Se trata, por tanto, de predecir un comportamiento dinámico a partir de datos estáticos. Para ello se trabajará con una base de ensayos presentada en el siguiente punto, de forma que para cada uno de ellos, como se puede observar en la Figura 79, se tiene una serie temporal con los valores de deformación a determinados instantes de tiempo.

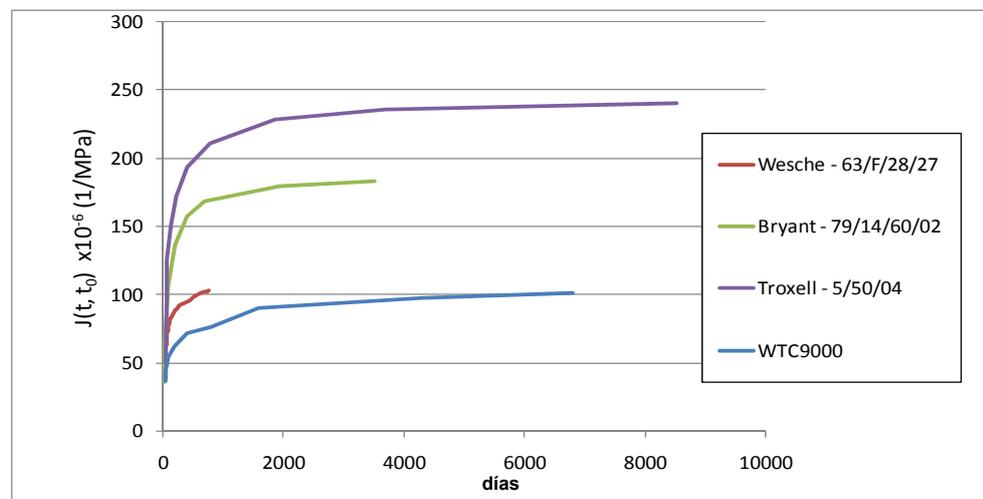


Figura 79. Ejemplo de 4 series de fluencia

3.2.1 Conjunto de datos

Se dispone de un conjunto de series experimentales, agrupadas internacionalmente en la base de datos de la *Réunion Internationale des Laboratoires et Experts des Matériaux, systèmes de construction et ouvrages* - RILEM (RILEM, 2009). Este conjunto de datos, puesto a punto por el comité TC 107, incluye ensayos comparables con distintos hormigones y en los que intervienen las mismas variables relacionadas con la fluencia. El conjunto de datos contiene valores que caracterizan cada ensayo (datos no dependientes del tiempo) y valores dependientes del momento en que se toma la medida. Dentro del primer grupo están: la relación volumen/área de la probeta (V/S , en mm), la humedad relativa del ambiente (HR , en %), el tiempo de curado del hormigón (t_c , en días), la edad del hormigón en la que se aplica la carga (t_0 , en días), el tipo de cemento (*Type*), la cantidad de cemento (c , en kg/m^3), la relación agua / cemento (w/c), la cantidad de árido (a , en kg/m^3), el módulo de deformación longitudinal en la edad de carga inicial ($E_{cm(t_0)}$, en MPa) y la

resistencia a compresión del hormigón a 28 días (f_{cm28} , en MPa). En cuanto a datos dependientes del tiempo se tienen: el día en que se realiza la medida (t , en días) y la deformación que se ha producido desde el inicio del ensayo (computada por unidad de carga y restada la deformación inicial mediante $(J(t, t_0) - J(t_0)) \times 10^{-6}$), en MPa^{-1}).

En conjunto se dispone de 2323 lecturas repartidas entre los 185 ensayos. De cada ensayo se tiene, como mínimo, alguna lectura de edad superior a 500 días. Las lecturas de cada probeta se producen a intervalos irregulares: en algunos casos existen pocos registros (5 ó 6 lecturas experimentales) y en el mejor de los casos 48 lecturas. En la Tabla 38 se muestran las variables, los rangos de éstas, y los valores de normalización empleados en el sistema desarrollado. Gráficamente en la Figura 80 se muestra la distribución de las variables.

Tabla 38. Variables, rangos y normalización empleados en el desarrollo

Variable	Nombre de la variable en las expresiones	Extremos de normalización (min – max)	Extremos de variación reales (min – max)
V/S (mm)	VS	0 – 220	17.5 – 200
HR (%)	H	0 – 110	20 – 100
t_c (días)	tc	0 – 440	0.4 – 400
t_0 (días)	$t0$	0 – 3600	0.5 – 3300
t (días)	t	0 – 10000	0.5 – 8525
$Ctype$	$Ctype$	1 – 2 – 3	1 – 2 – 3
c (kg/m ³)	C	250 – 600	275 – 564
a (kg/m ³)	A	1500 – 2300	1661 – 2110
w/c	W/C	0.2 – 0.8	0.25532 – 0.70957
E_{cmt0} (MPa)	$Ecmt0$	10000 – 50000	10554 – 47620
f_{cm28} (MPa)	fc_m28	0 – 130	17.2 – 119
$J(t, t_0) - J(t_0) \times 10^{-6}$ (1/MPa)	SALIDA	0 – 300	0 – 262

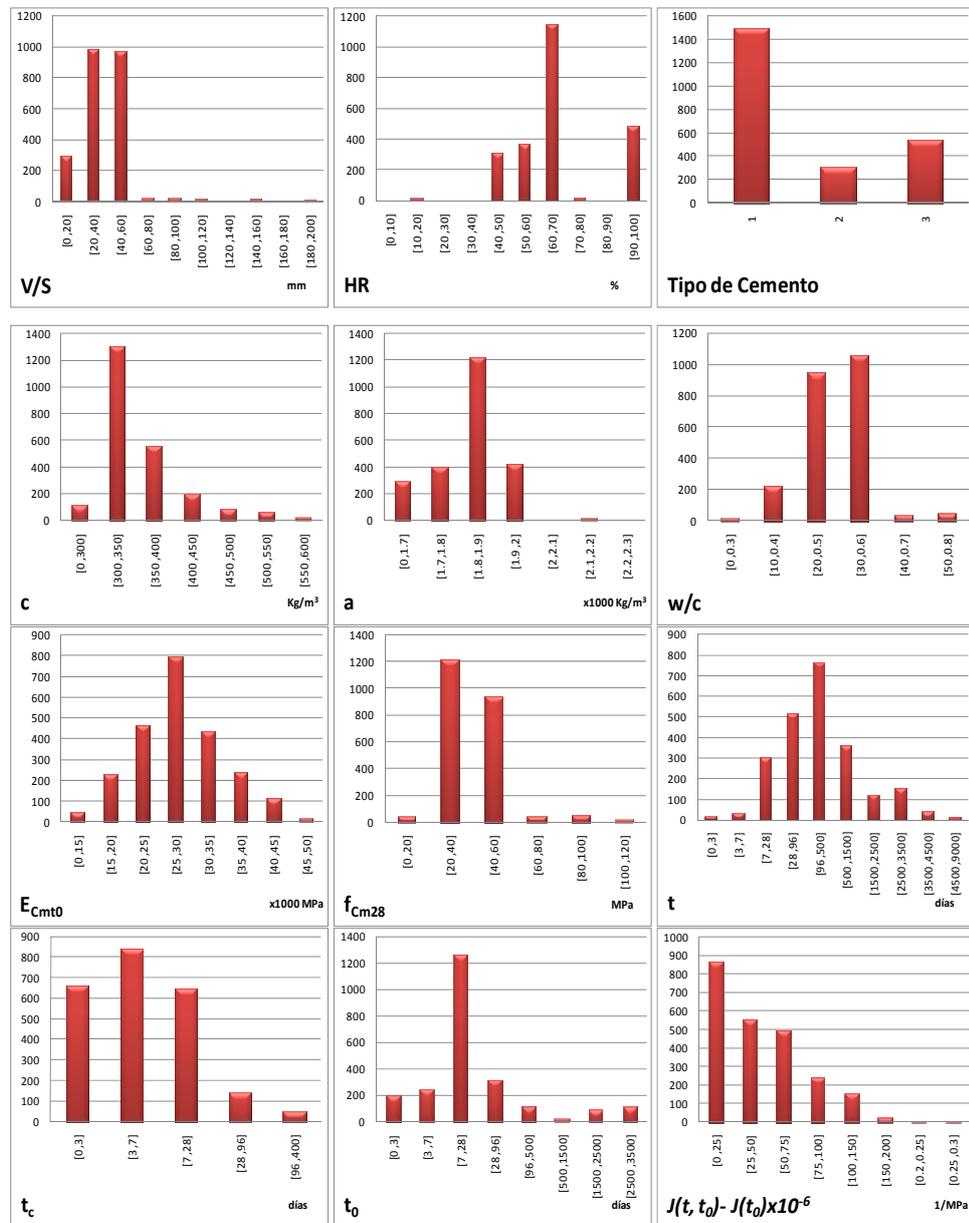


Figura 80. Distribución del conjunto de datos empleado

3.2.2 Configuración del método

Tras un análisis del comportamiento del conjunto de casos del que se ha dispuesto, se ha constatado que con la expresión “ $\ln(t-t_0+1)$ ” se obtiene una tendencia adecuada en este tipo de ensayos ya que el carácter temporal de la serie imprime una trayectoria asintótica en la deformación de fluencia a medida que va pasando el tiempo (Figura 77, Figura 79 y Figura 81)

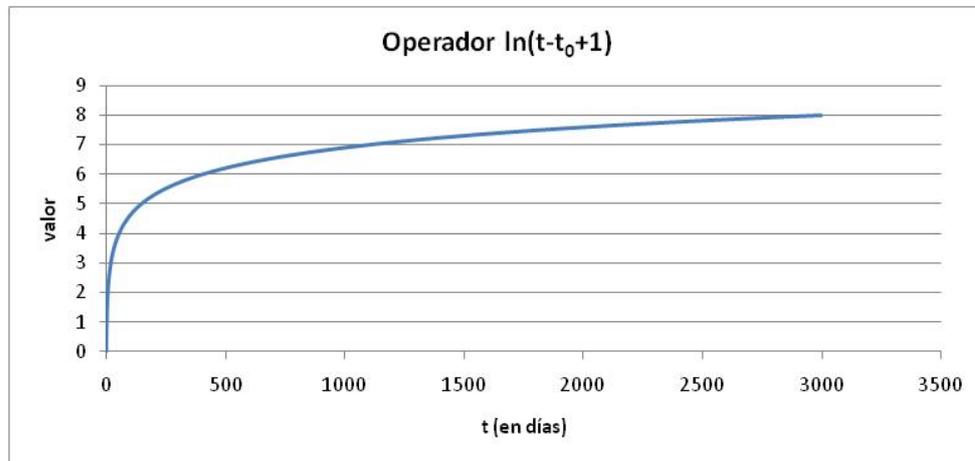


Figura 81. Operador “ $\ln(t-t_0+1)$ ” introducido

Con esto se concluiría la fase 1 de la metodología “Diseño del experimento y conocimiento previo”. A continuación en la fase 2 “Selección del modelo o algoritmo de extracción” se muestran las peculiaridades del modelo elegido y la adaptación de los algoritmos que se han empleado.

Como en todos los ejemplos que se plantean en este trabajo se emplearán las técnicas desarrolladas en el capítulo VI “Sistema basado en conocimiento aplicado a la Programación Genética” explicando en cada momento la adaptación que se realiza para lograr los objetivos.

En las primeras aproximaciones realizadas se ha empleado la técnica de PG clásica para la realización de regresión simbólica empleando las base de datos comentada. Se han realizado diversas configuraciones de parámetros, por ejemplo distintos algoritmos de selección (torneo, ruleta ...), distintos porcentajes de cruce y mutación, diferentes tamaños de población entre el intervalo [100, 1500]. En este primer grupo de pruebas se han obtenido resultados poco satisfactorios aproximándose al error medio obtenido por los modelos existentes pero en ningún caso mejorando estos. Esto fue debido a que de la forma clásica la PG tiene dificultad a adecuarse al carácter temporal de este tipo de series. Es por este motivo que este ejemplo ha sido candidato para la aplicación del sistema desarrollado. En este caso se configura el algoritmo de extracción de conocimiento de manera que se introduzca el conocimiento del experto en forma de ecuación matemática citada anteriormente. En la Figura 82 se muestra de forma esquemática la configuración realizada. A mayores, y para lograr un mayor ajuste, aun no siendo estrictamente necesario en la PG, se ha realizado un proceso de normalización de todas las variables.

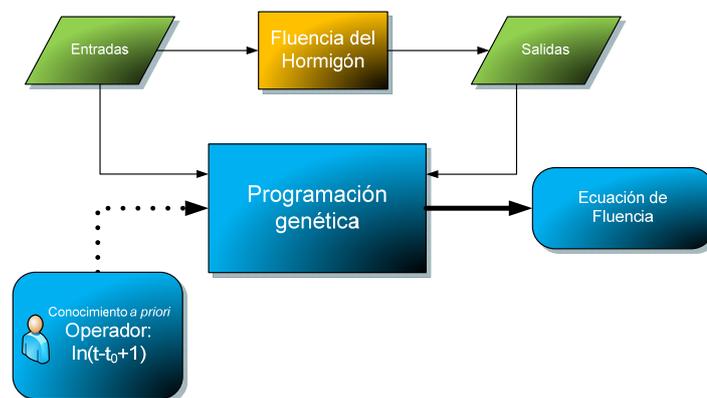


Figura 82. Diagrama del proceso de extracción de conocimiento

Una vez finalizadas la fase 1 y 2 de la metodología se está a disposición de la estimación de los mejores parámetros (fase 3). En este caso, después de hacer un batería de pruebas se ha estimado que la mejor configuración es la que se detalla a continuación: Como nodos no terminales (o funciones) sería la suma, la resta, el producto, la división protegida, la potencia y el “operador-expresión” $\ln(t-t_0+1)$ definido anteriormente. En cuanto a los nodos terminales se han empleado las variables procedentes de la base de datos ($V/S, HR, t_s, t_p, t, cType, c, a, w/c, f_{cm28}, E_{cm0}$) y constantes comprendidas entre el intervalo [-1,1] dado que dichas variables han sido normalizadas en ese rango. En cuanto a los parámetros específicos de la PG serían: para el algoritmo de selección se ha empleado el torneo, el porcentaje de cruces se ha establecido al 95%, el porcentaje de mutaciones en el 20%, el tamaño de la población se determinó en 400 individuos y la altura máxima del árbol en 9 nodos. Por último, para evitar el excesivo crecimiento de los árboles se estableció la parsimonia en 0.00001.

3.2.3 Resultados

En las siguientes ecuaciones se muestran las mejores soluciones obtenidas empleado el método propuesto. Para la obtención de la fórmula con ajustes similares a los modelos actuales (fórmula F1) hicieron falta 450 generaciones de individuos, mientras que para la F4 (la que obtiene mejor ajuste) ha sido necesario emplear cerca de 70.000 generaciones.

$$f1 = 0.0685 * \frac{W}{C} * \ln(t - t_0 + 1) \quad (118)$$

$$f1 = \alpha * \sqrt{-1.049 + t_0 - \frac{W}{C} + E_{cmt_0} + H^2} \quad (119)$$

$$\alpha = 0.05(\ln(t - t_0 + 1) + \sqrt[3]{t_c} - \sqrt[6]{t_c} - 2 * \sqrt{t_0} - 2 * VS * C_{type} - 4 * E_{cmt_0}^2)$$

$$f3 = -0136 * \left(H \left(-0.36 * \frac{W}{C} + 0.694 \right) + 0.583 * t_0 - 0.835 \right) * \ln(t - t_0 + 1) * (-0.796 * E_{cmt_0} + 0.899) \quad (120)$$

$$f4 = -0187 * \alpha * \ln(t - t_0 + 1) * (-0.768 * E_{cmt_0} - 0.086 * t_0 + 0.927)$$

$$\alpha = H * \left(1.268 * \left(\frac{V}{C} \right)^2 * (VS - 0.366) + 0.504 \right) - 0.1471 * t_c - 0.64 + \frac{0.4172 * t_0}{t_0 + VS} \quad (121)$$

En la Tabla 39 se pueden observar (ordenados de menor a mayor error por el Error Medio) los errores y el coeficiente de correlación que se obtienen aplicando los modelos actuales y las formulaciones obtenidas. En todos los cálculos se ha procedido previamente a deshacer la normalización.

Tabla 39. Resultados de los diferentes modelos (valores reales)

	F4	F3	F2	GL	CEB	ACI	F1	B3
Error Medio	10.5	10.8	11.2	12.0	13.7	14.7	17.4	17.6
Error Cuadrático Medio	276.3	305.1	287.1	362.7	410.4	549.0	679.5	622.8
Coficiente R ²	0.842	0.825	0.838	0.794	0.782	0.764	0.614	0.716

Eliminando la normalización, el grado de ajuste de la expresión F4 obtenida puede observarse en la figura 3.

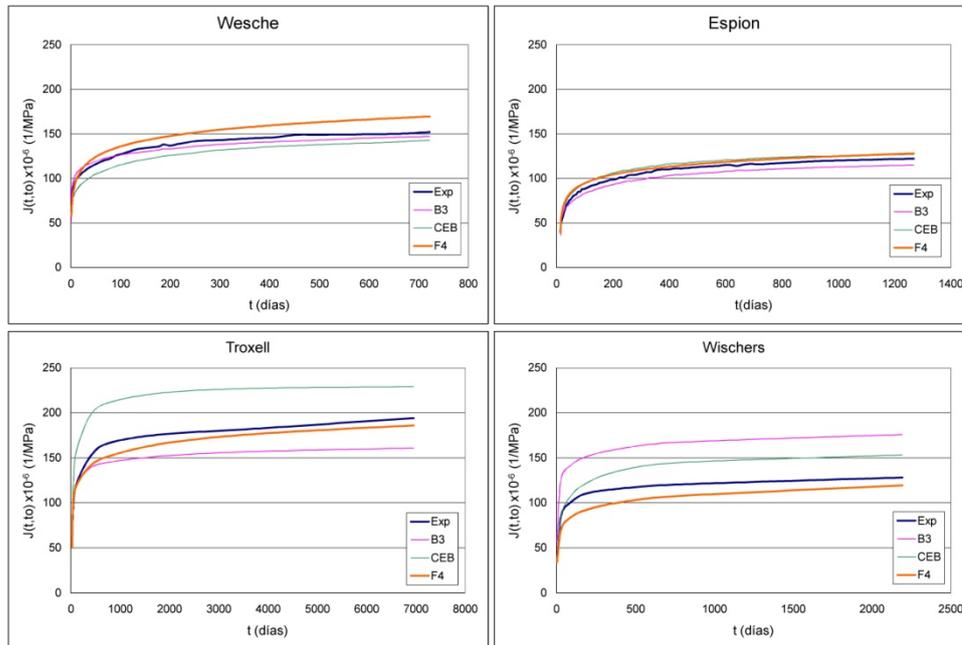


Figura 83. Capacidad de predicción de la fórmula F4, comparada con los modelos CEB y B3

A la vista de los resultados obtenidos se puede concluir que la fórmula F4 es la que mejor modela el comportamiento de la fluencia, y para ello necesita emplear 7 variables. En cambio, la fórmula F2, aún empleando más variables (9), consigue un peor ajuste. Por el contrario la fórmula F1, empleando sólo 4 variables, mejora el error medio del modelo B3, que utiliza todas las variables.

La correlación que se obtiene al aplicar los modelos actuales al conjunto total de datos no supera el valor de 0.795, siendo GL2000 el mejor modelo hasta el momento. Con la PG se ha conseguido mejorar la correlación llegando con la Formula 4 una correlación cercana a 0.842. La Figura 84 muestra la relación entre los datos experimentales y los distintos modelos.

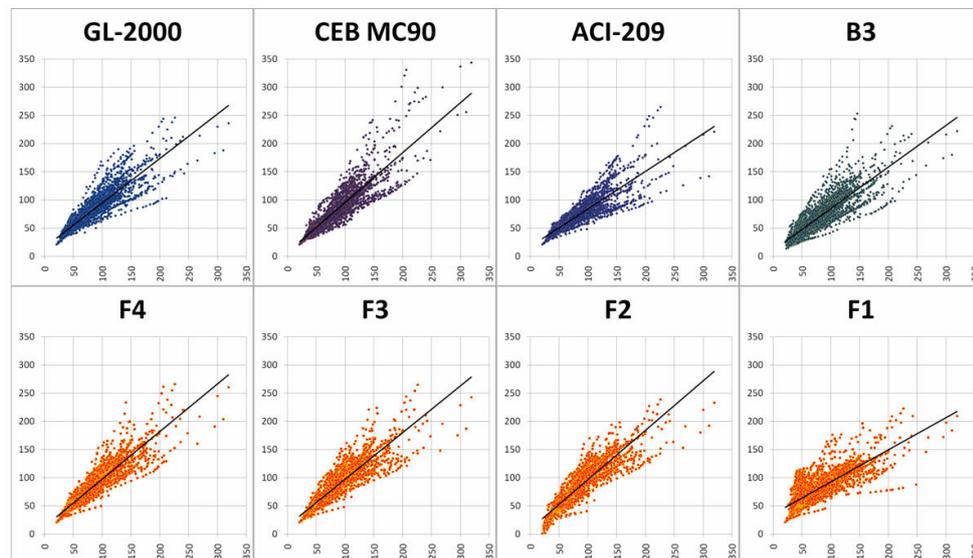


Figura 84. Relación entre los datos experimentales y los proporcionados por los distintos modelos

Parte de los resultados obtenidos del estudio de éste fenómeno con técnicas de computación evolutiva se ha divulgado en congresos internacionales como son IBERAMIA 2008 (Pérez, Miguélez, Rabuñal & Martínez, 2008), ACHE 2008 (Pérez, Rabuñal, Herrador & Martínez 2008), IWANN 2009 (Pérez, Martínez, Franco & Berrocal 2009) e ICSA2010 (Pérez, Eiras & Martínez 2010). De igual manera se ha publicado un capítulo en el libro *“Soft Computing Methods for Practical Environment Solutions: Techniques and Studies”* (Pérez, Rabuñal & Martínez 2010).

VIII. Conclusiones

El objetivo principal de la presente tesis es el desarrollo de una metodología basada en Inteligencia Artificial que permita extraer información de un sistema apoyándose en el conocimiento de un experto. Se ha alcanzado este objetivo y se ha demostrado que es posible incorporar el conocimiento del experto en dos sentidos: por un lado éste aporta información inicial del problema a resolver y, por otro, también interviene en el proceso de búsqueda de la solución. Este último aspecto tiene gran relevancia, ya que el sistema que se ha desarrollado como parte de la demostración de dicha metodología permite que el experto obtenga soluciones orientadas, es decir, válidas dentro del ámbito científico o ingenieril según su criterio.

La técnica en la que se ha basado la implementación del sistema diseñado es la de Programación Genética (PG). Esta técnica, tal y como se conoce hoy en día, no satisface necesariamente los criterios exigidos por los expertos. La

razón de ello es que las soluciones que se obtienen empleando PG Clásica no contemplan aspectos relacionados con la seguridad estructural, la interpretación física del fenómeno, etc. Por esta razón, tras desarrollar el sistema de extracción de información se ha puesto a prueba sobre tres problemas del ámbito de la Ingeniería Civil.

El contexto en el que se desarrollan los ejemplos es el de la ingeniería de la construcción, que tiene características bastante específicas. En primer lugar, los ingenieros civiles priman la aplicabilidad de las soluciones frente al ajuste de las mismas. Muestra de ello es el caso de la predicción a cortante, en la que, aunque se han conseguido mejores ajustes con fórmulas complejas, en primera instancia se prefiere la solución basada únicamente en constantes. Además, resulta de suma importancia la seguridad estructural implícita en las fórmulas predictivas, conseguida en el proceso de búsqueda penalizando más las soluciones que yerran generando inseguridad estructural.

La validez de aplicación de la metodología desarrollada ha quedado patente ya que en todos los ejemplos de aplicación a problemas reales se ha conseguido obtener formulaciones que mejoran la predicción de los resultados frente a las normas y códigos vigentes, y requiriendo los criterios de satisfacción impuestos por los expertos del área.

En cuanto a la implementación de la metodología, cabe destacar que la incorporación de los mecanismos de orientación a la PG ha supuesto un incremento en tiempo computacional frente a la técnica clásica. De todas formas este aumento se ve compensado por los resultados que se obtienen, los cuales satisfacen los criterios de los expertos, al contrario de lo que sucedía con PG Clásica.

Detallando aspectos de la implementación, cabe destacar que se han desarrollado algunos módulos novedosos, que fueron, en general, adaptados de técnicas conocidas de computación evolutiva. A continuación se detalla alguno de estos módulos.

- El módulo de operadores y funciones definidos por el usuario consigue dirigir la búsqueda de soluciones mediante fórmulas predefinidas por el experto.
- Con el módulo de probabilidad de aparición de nodos, es posible dirigir la búsqueda de soluciones otorgando más probabilidad a los operadores que el experto indica “a priori” que influyen más en el proceso.
- Mediante el módulo de coeficientes de seguridad se consigue obtener modelos predictivos seguros, es decir, se desvía la predicción hacia la seguridad. Este condicionante es considerado de gran importancia para los expertos y representa un avance interesante en la técnica de PG.
- El módulo de ajuste de expresiones combinado con el módulo de restricciones permite obtener mejores formulaciones matemáticas basadas en modelos ya conocidos, mejorando determinadas partes de éstos. Este procedimiento demostró ser útil y válido según el criterio de los expertos del área.

La metodología desarrollada se ha aplicado sobre tres problemas de forma satisfactoria. Cabe señalar que los resultados obtenidos son buenos, pero no en términos de ajuste óptimo, sino por el hecho de alcanzar un buen ajuste y siendo simultáneamente válido para el experto.

En el caso de ejemplo del proyectil se ha podido comprobar que la metodología, en el caso de existir una fórmula que relaciona los datos de

entrada con la salida, es capaz de obtener dicha fórmula en un corto periodo de tiempo. Sin embargo con la PG clásica, aún consiguiendo, en un tiempo considerablemente alto, formulaciones con errores similares, la solución obtenida distaría bastante de la formulación matemática que rige dicho fenómeno físico.

Se abordaron otros tres casos de ejemplo, el análisis del cortante en formulación de hormigón armado, el estudio de la adherencia pasiva de hormigón acero, fundamental en el diseño de elementos estructurales y la fluencia del hormigón sometido a carga constante a lo largo del tiempo. En estos tres casos se consiguió, apoyándose siempre en el experto, mejorar las formulaciones habituales que modelan estos fenómenos físicos. El grado de mejora ha sido variable y ha sido posible perfilar varios protocolos de actuación en función de aquello que busca el experto, bien sea obtener una fórmula que modele una situación compleja, bien la mejora de una formulación preexistente.

Las principales dificultades aparecidas en la aplicación de la técnica en entornos reales fueron, por un lado, la dificultad en la extracción de las restricciones por parte del experto y, por otro, la recopilación de los datos experimentales. La primera limitación ha motivado la realización de multitud de pruebas que ha permitido mejorar la metodología. La segunda dificultad relevante tiene su origen en la procedencia de los resultados experimentales, un compendio de ensayos realizados en campañas de distintos autores y años. Esto implicó un arduo proceso de recopilación y depuración de inconsistencias y errores en los datos, realizado para cada uno de los tres ejemplos tratados.

Como conclusión y síntesis final cabe resaltar que se ha conseguido desarrollar una metodología funcional para extraer conocimiento de datos experimentales apoyándose en la perspectiva de un experto en la materia. Además, esta metodología, aún siendo general ha sido aplicada a casos concretos de ingeniería civil, demostrando su aplicabilidad con buenos resultados, tanto en ajuste como en validez, a instancias del experto. En definitiva se puede obtener una herramienta de apoyo al experto en la modelización de un fenómeno apoyado en datos experimentales.

IX. Conclusions

The main objective of this thesis was to develop an artificial intelligence system that would extract information from a system relying on knowledge from an expert. This goal has been achieved and has been shown that it is possible to incorporate expert knowledge in two ways: firstly, by providing initial information relevant to the solution of the problem, and secondly, by getting involved in the process of finding the solution. This last issue is very important, because the system presented allows the expert to get oriented solutions, i. e. only valid under the expert's discretion within the scientific field.

The base technique of the implementation of the designed system is Genetic Programming (GP). This technique is nowadays known not to meet the criteria required by the experts. The cause for this failure is that the solutions obtained using Classic GP do not include aspects related to structural safety,

physical interpretation of the phenomenon, and so on. For this reason, the system was applied to three engineering problems.

The context in which the examples were developed is Structural Engineering, which has quite specific characteristics. First, structural engineers value the applicability of a given solution over its precision. Proof of this is the case of shear resistance prediction, where the preferred solution is based on constants, over better fits obtained with complex formulae. Moreover, it was important to include structural safety prediction implied the formulae, by penalizing solutions where errors were committed towards structural unsafety.

The validity of the application of the methodology hereby developed has been proven, since in all examples of application to real problems there have been obtained formulations that improve prediction of results against those provided by up-to-date standards and codes.

Regarding the implementation of the methodology, it is noteworthy that the incorporation of orientation mechanisms to GP has translated into an increase in computational time compared to the classical technique. Anyway, this increase is offset by the quality of the results obtained, which satisfy the criteria of the experts, contrary to what was happening to Classic GP.

When detailing certain aspects of the implementation, it must be noted that some new modules have been developed, which were generally known techniques adapted from evolutive computing. Some of these modules are presented below.

- The user defined operators and functions module directs the search for solutions through predefined formulas.

- With the node probability appearance module, it is possible to direct the search for solutions by giving operators that the expert considers to have “a priori” more probability to influence the process.
- Through the safety factors module, models with included safety can be achieved, i.e. the prediction is deflected to safety. This condition is considered of great importance to the experts and represents an interesting development in the technique of GP.
- The expressions adjustment module combined with the constraints module yields better models based on known models, improving certain parts of them. This procedure proved to be useful and valid, as judged by the experts.

The methodology has been satisfactorily applied to three problems. Note that the results are good, not in terms of optimal fit, but rather as the combination between a good fit and a valid tool for the expert.

In the projectile example has been shown that the methodology, in the case where a formula exists that relates the input data with the output, is able to obtain this formula in a short time. However, with classic GP, even though formulations with similar accuracy are obtained after a rather long period of time, the solution is far from the mathematical formulation governing the physical phenomenon.

Three more sample cases were addressed: the analysis of shear in reinforced concrete design; the study of lap splice lengths on steel bars for concrete, fundamental in the design of structural elements; and concrete creep, the evolution over time of strains in concrete subject to constant load. In all three cases, by relying on the expert, the formulations achieved improve over the standard formulations that model these physical phenomena. The degree of

improvement was variable and it has been possible to outline a number of protocols in terms of what is the expert looking for, whether it is to obtain a formula that models a complex situation, or refine an existing formulation.

The main difficulties encountered in the application of the technique to real cases were, on one hand, the difficulty found by the expert to isolate the constraints, and on the other, the collection of experimental data. The first limitation has motivated the realization of many different tests that have improved the methodology. The second important restriction is based on the origin of the experimental results, consisting on a compilation of tests performed in campaigns by various authors and different years. This involved an arduous process of collection and treatment of inconsistencies and errors in the data, and this for each of the three application examples.

As a final conclusion and summary it should be highlighted that success has been achieved in the development a functional methodology to extract knowledge from experimental data by drawing on the perspective of an expert in the field. In addition, this methodology, while generic, has been applied to specific cases of Structural Engineering, demonstrating its applicability with good results both in fit as valid instances of the expert.

X. Futuros Desarrollos

Como en cualquier desarrollo de un proceso ingenieril, tras su finalización se identifican posibles mejoras que en una futura trabajo podrían ser contempladas.

Una de las mejoras más demandadas por parte de los ingenieros civiles es la disponibilidad de expresiones continuas dentro del rango válido de todas las variables implicadas en el problema. Para un conjunto de datos experimentales en el intervalo (+2 , +10) que no incluyan el valor 5 puede resultar propicia la expresión $1/(x-5)$, que sin embargo generará una discontinuidad en $x=5$. Debe evitarse en futuros desarrollos de este método, por tanto, que la expresión generada pueda integrar el resultado. En la actualidad, este proceso se está depurando de forma manual. .

Otra posible mejora en el funcionamiento global sería añadir restricciones de coherencia dimensional en los resultados obtenidos. Aunque en la mayoría de normativas nacionales e internacionales no se tenga en cuenta este hecho, resultaría interesante poder asociar a cada variable una unidad de medida y que los resultados obtenidos sean de una unidad determinada, coherente con la ecuación presentada. Ello, además, daría validez a la expresión en los diferentes sistemas de medida.

Uno de los procesos tediosos es encontrar la configuración de parámetros óptima, y se podría ampliar el sistema basado en conocimiento para que aplique automáticamente métodos habituales en el devenir de la Programación Genética. Ejemplo de ello sería la variación paulatina del valor de parsimonia a medida que va evolucionando la población consiguiendo expresiones más simples pero sin limitar al inicio del proceso evolutivo, o el aumento automático de la ratio de mutación cuando la población está estancada.

Uno de los inconvenientes conocidos en cualquier método basado en aprendizaje de patrones es el tiempo dedicado a la evaluación de las soluciones. La Programación Genética no es ajena a este problema, y por ello una de las futuras mejoras que se plantea sería la distribución del proceso de evaluación de los individuos, es decir, la paralelización del algoritmo de evaluación para obtener los resultados en un tiempo menor.

XI. Bibliografía

- Abrams D. A. (1913) *Tests of Bond Between Concrete and Steel*. University of Illinois Bulletin N° 71, Urbana, Illinois, USA: University of Illinois.
- ACI Committee 209. (1982). Prediction of Creep, Shrinkage and Temperature Effects in Concrete Structures. Detroit: ACI 209.
- ACI Conmittee 318. (2005). Building code requirements for structural concrete and commentary. USA: ACI 318.
- Aguirre, H. E., Tanaka, K. & Sugimura, T. (1999). Cooperative crossover and mutation operators in genetic algorithms. *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-99*. Orlando-Florida.
- Aho, A. V., Lam, M. S., Sethi, R. & Ullman, J. D. (2006). *Compilers: Principles, Techniques, and Tools* (2ª edition). Addison Wesley.

Angeline, P. J. & Pollack, J. B. (1994). Co-evolving high level representation. Artificial Life III. ed Langton CG: Addison Wesley.

Angeline, P. J. (1996) An investigation into the sensitivity of genetic programming to the frequency of leaf selection during subtree crossover. Genetic Programming 1996: *Proceedings of the First Annual Conference GP-96* (pp. 21-29). Stanford University, California, USA: MIT Press

ASCE-ACI Committee 426. (1973). The shear strength of reinforced concrete members. *Journal Structural División*, 99 (6), 1091-1187.

ASCE-ACI Committee 445. (1998). Recent approaches to shear design of structural concrete. *Journal Structural Engineering*, 124 (12), 1375-1417.

Ashour, A. F., Álvares, L. F. & Toropov, V. V. (2003). Empirical modelling of shear strength of RC deep beams by genetic programming. *Computers & Structures*, 81, 331-338.

Bauer, B. A., Lee, M., Bergstrom, L., Wahner-Roedler, D. L., Bundrick, J., Litin, S., Hoffer, E., Kim, R. J., Famiglietti, K., Barnett, G. O. & Elkin, P. L. (2002). Introduced on a Teaching Hospital Service. *Proceedings of AMLA Annual Symposium. Internal Medicine Resident Satisfaction with a Diagnostic Decision Support System (Dxplain)*, (págs. 31-35). San Antonio, TX.

Bazant, Z. P. & Baweja, S. (1995). Creep and Shrinkage Prediction Model for Analysis and Design of Concrete Structures-Model B3. *Materials & Structures*, 28 (6), 357-365.

-
- Blickle, T. & Thiele, L. (1995). A Mathematical Analysis of Tournament Selection. En L. Eshelman, ed. Genetic Algorithms. *Proceedings of the Sixth International Conference (ICGA95)*. Morgan Kaufmann, San Francisco, CA.
- Cairns, J (2006, noviembre) *Model for strength of lapped joints and anchorages*. Ponencia presentada en la Reunión del grupo TG 4.5 de la International Federation for Structural Concrete, Stuttgart, Alemania.
- Cantú-Paz, E. (2000). Selection Intensity in Genetic Algorithms with Generation Gaps. *The Genetic and Evolutionary computation conference*. Las Vegas, Nevada, USA.
- Cantú-Paz, E. K. & Kamath, C. (2005). An empirical comparison of combinations of evolutionary algorithms and neural networks for classification problems. *IEEE Transactions on systems, Man and Cybernetics Part B- Cybernetics*, 35 (5), 915-927.
- Chakraborty, U. K., Deb, K. & Chakraborty, M. (1996). Analysis of Selection Algorithms: A Markov Chain Approach. *Evolutionary Computation*. Vol. 4, No. 2, Pages 133-167.
- Chellapilla, K. (1997). Evolutionary programming with tree mutations: Evolving computer programs without crossover Genetic Programming: *Proceedings of the Second Annual Conference*. (pp. 431-438). CA, USA: Stanford University.
- Cios, K., Pedrycz, W., Swiniarski, R. & Kurgan, L. (2007). *Data Mining: A Knowledge Discovery Approach*. New York: Springer.

- Cladera, A. & Marí, A. R. (2004). Shear design procedure for reinforced normal and high-strength concrete beams using artificial neural networks. Part I: beams without stirrups. *Engineering Structures*. 26 (7), 917-926.
- Cladera, A., Pérez, J. L. & Martínez-Abella, F. (2010). Bending Moment-Shear Force Interaction for Elements without Stirrups: Improvement of the Eurocode 2 Shear Procedure. *The Tenth International Conference on Computational Structures Technology*. Valencia. España.
- Collins, M. P., Bentz, E. G., Sherwood, E. G. & Xie, L. (2008). An adequate theory for the shear strength of reinforced concrete structures, in *Morley Symposium on Concrete Plasticity and its Application*. University of Cambridge: Cambridge.
- Collins, M.P (2001). A evaluation of shear design procedures for concrete structures. *A report prepared for the CSA Technical Committee on Reinforced Concrete Design*. Canada.
- Darwin, C. R. (1859). *On the Origin of Species by Means of Natural Selection*. London: John Murray.
- Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, Vol. 10 (7), págs. 1895-1924. MIT Press Cambridge, MA, USA.
- EHE-08. (2008). *Instrucción de Hormigón Estructural*. España: Publicaciones del Ministerio de Fomento. Secretaría General Técnica.

- Eurocode 2. (2002) Design of concrete structures Part I: General rules and rules for buildings. European Committee for Standardization. Bruselas, EC-2.
- FIB Task Group 4.5 “Bond models” (s.f). Recuperado el 21 de mayo de 2009, de <http://fibtg45.dii.unile.it/>
- Fuchs, M. (1999). Large Populations Are Not Always The Best Choice In Genetic Programming. *GECCO-99. Proceedings of the Genetic and Evolutionary Computation Conference*, págs. 1033-1038, Morgan-Kauffman. Orlando, Florida, USA.
- García-Arnau, M., Manrique, D., Ríos, J. & Rodríguez-Patón, A. (2007). Initialization method for grammar-guided genetic programming. *Knowledge-Based Systems* , 20, 127-133.
- Gathercole, C. , Ross, P. & Bridge, S. (1997). Small Populations over many Generations can beat Large Populations over Few Generation, in Genetic Programming: *Proceedings of the Second Annual Conference* (1997), pp. 111-118. Stanford University, CA: Morgan Kaufmann.
- Gestal, M. (2009). *Computation Evolutiva para el Proceso de Selección de Variables en espacios de Búsqueda Multimodales*. Tesis doctoral. Departamento de Tecnologías de la Información y las Comunicaciones: Universidad de A Coruña, A Coruña.
- Giancoli, Douglas C. (2005). *Physics: Principles with Applications, sixth edition*. Upper Saddle River, N.J.: Prentice Hall.

- Goldberg, D. E. & Deb, K. (1991). A Comparative Analysis of Selection Schemes Used in Genetic Algorithms. *Foundations of Genetic Algorithms*, 51, 61801-62996.
- Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Reading, MA: Addison-Wesley.
- González B. (2002). *Hormigones con áridos reciclados procedentes de demoliciones: dosificaciones, propiedades mecánicas y comportamiento estructural a cortante*. Tesis Doctoral, Departamento de Tecnología de la Construcción, Universidade da Coruña, A Coruña, España
- Graupe, D. (1997). Principles of Artificial Neural Networks. *Advances series on Circuits and Systems*, Vol 3, in World Scientific Publishing Co. Singapore.
- Haykin, S. (1999). *Neural Networks second Edition*. (Prentice Hall ed.). NJ: Englewood Cliffs.
- Hoai, N. M. & McKay, R. (2001). A framework for tree adjunct grammar guided genetic programming. En H. A. Abbass (Ed.), *Proceedings of the Post-graduate ADFEA: PACCS '01. Conference on Computer Science* (págs. 93-99). ADFEA.
- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. Michigan: University of Michigan Press.
- Iglesias, G., Rabuñal, J. R., Losada, M. A., Pachón, H., Castro, A. & Carballo, R. (2008). A virtual laboratory for stability tests of rubble-mound breakwaters. *Ocean Engineering*, 35, 1113– 1120.

-
- Kantardzic, C. (2003). *Data mining: concepts, models, methods and algorithms*. Totowa, NJ: Wiley-IEEE Press.
- Kicinger, R., Arciszewski, T. & Jong, K. (2005). Evolutionary computation and Structural design: A survey of the state-of the-art. *Computers and Structures* , 83 (23-24), 1943-1978.
- Koza, J. R. (1992). *Genetic Programming: On the programming of computers by means of natural selection*. Cambridge: MIT Press.
- Koza, J. R. (1994). *Genetic Programming II: Automatic Discovery of Reusable Programs*. Cambridge, MA: MIT Press.
- Ljung, L. (1999). *Systems Identification: Theory for the User (2nd ed.)*. Linköping University, Sweden: PTR Prentice Hall.
- Ljung, L. (2010). Perspectives on system identification, *Annual reviews in control*. 34.
- Lockman, M. J. (2000). *Compliance, relaxation and creep recovery of normal strength concrete*. Thesis of Departament of Civil engineering: Ottawa, Canada.
- Luke, S. (2000). Two Fast Tree-Creation Algorithms for Genetic Programming. *IEEE-Transactions on Evolutionary Computation* , 4 (3), 274-283.
- McCulloch, W. S. & Pitts, W. (1943). A Logical Calculus of Ideas Immanent in Nervous Activity. *Bulletin of Mathematical biophysics* , 5 (5), 115-133.
- Montana, D. J. (1995). Strongly typed genetic programming. *Evolutionary Computation* , 3 (2), 199-230.

- Müller, H. S. & Hilsdorf, H. K. (1990). Evaluation of the Time Dependent Behavior of Concrete. CEB Comite Euro-International du Beton. France: *Bulletin d' Information n° 199*.
- Nikolaev, N. I. & Iba, I. (2001). Accelerated Genetic Programming of Polynomials. *Genetic Programming and Evolvable Machines*, 2 (3), 231-257.
- Oliver Morales, C. & Rodríguez Vázquez, K. (2004). Symbolic Regression Problems by Genetic Programming with Multi-branches. *En MICAI 2004: Advances in Artificial Intelligence* (Springer Berlin/ Heidelberg ed., Vol. 2972, págs. 717-726). Berlin: LNCS.
- Orangun, C. O., Jirsa, J. O., & Breen, J. E. (1977). A Reevaluation of Test Data on Development Length and Splices. *ACI Journal*. 74(11), 114-122.
- Orchad, G. (1993). *Neural Computing, Research and Applications*. Londres: Institute of Physics Publishing.
- Pereira, F. B. , Machado, P., Costa, E. & Cardoso A. (1999). Graph based crossover-A case study with the busy beaver problem. *Proceedings of the Genetic and Evolutionary Computation conference*. GECCO-99, págs. 1149-1155. Orlando-Florida-USA, Morgan Kaufmann.
- Perera, R. & Vique, J. (2009). Strut-and-tie modelling of reinforced concrete beams using genetic algorithms optimization. *Construction and Building Materials*, 23 (8), 2914-2925.

- Pérez Ordóñez, J. L. ; Miguélez Rico, Mónica; Rabuñal Dopico, J. R. & Martínez Abella, F. (2008). Applying Genetic Programming to Civil Engineering in the Improvement of Models, Codes and Norms. *Iberamia 2008- 11th Ibero-American Conference on Artificial Intelligence*. Lisboa-Portugal.
- Pérez Ordóñez, J. L. ; Rabuñal Dopico, J. R.; Herrador Barrios, M.F. Martínez Abella, F. (2008). Aplicación de técnicas de Programación Genética en la determinación de funciones de fluencia. *IV Congreso de ACHE*. Valencia.
- Pérez Ordóñez, J. L. ; Martínez Abella, F. ; Catoira Franco, A. & Berrocal Benito, J. (2009). Special Time Series. Prediction: Creep of Concrete. *10th International Work-Conference on Artificial Neural Networks*, Iwann 2009, pgs: 1184-1191, Salamanca.
- Pérez, J. L. & Carro; L. (2007). Modelos Avanzados Utilizando Computación Evolutiva para la Mejora de Normativa de Diseño de Estructuras. *TIC en Ingeniería Civil* (ISBN 84-934497-7-6) Colección Informática 17, pgs: 107-126. Ed: Fundación Alfredo Brañas. Santiago de Compostela.
- Pérez, J. L., Rabuñal, J.R. & Martínez, F. (2010). Softcomputing techniques in Civil Engineering: time series forecast. *Soft computing Methods for Practical Environmental Solutions: Techniques and Studies*. Hersey: IGI global.
- Pérez, J. L.; Eiras, J. & Martínez, F. (2010). Artificial Intelligence techniques applied to reinforced concrete. *Structures & Architecture*. Guimarães, Portugal.

Pérez, J. L.; González, B. & Martínez. F. (2008). Techniques of EC in the structural concrete field. *Encyclopedia of Artificial Intelligence* (ISBN 978-1-59904-849-9), pgs: 526-531, Ed: Information Science Reference. EU

Pérez, J. L.; Martínez, I. & Herrador, M. (2008). Application of Artificial Neural Networks in the field of Structural Concrete. *Encyclopedia of Artificial Intelligence* (ISBN 978-1-59904-849-9), pgs: 118-124, Ed: IGI Global. Londres.

Polí, R. (1996). Some Steps Towards a Form of Parallel Distributed Genetic Programming. *Proceedings of the First On-Line Workshop on Soft Computing*. UK.

Rabuñal Dopico, J. R. , Pérez Ordóñez, J. L. , Cladera Bohigas, A. & González Fonteboa, B. (2008). Shear and bond analysis on structural concrete using artificial neural networks. *5th International Engineering and Construction Conference, IECC'5*. Irvine. USA.

Rabuñal J.R., Dorado J., Pazos A., Pereira J., Rivero D. (2004). A New Approach to the Extraction of ANN Rules and to Their Generalization Capacity Through GP. *Neural Computation*, (16) 7, pág. 1483-1523.

Rabuñal, J. R. (2008). *Uso de técnicas de Inteligencia Artificial en Ingeniería Civil*. Thesis Department of Mathematics and Representation Methods . University of A Coruña.

- Ratle, A. & Sebag, M. (2001). Grammar-guided genetic programming and dimensional consistency: application to non-parametric identification in mechanics. *Applied Soft Computing*, 1 (1), 105-118.
- RILEM 2009. (s.f.). Obtenido de Réunion Internationale des Laboratoires et Experts des Matériaux, systèmes de construction et ouvrages. Retrieved from : <http://www.rilem.net>.
- Rivero, D. (2007). *Desarrollo y simplificación de Redes de Neuronas Artificiales mediante el uso de técnicas de Computación Evolutiva*. Tesis Doctoral, Departamento de Tecnologías de la Información y las Comunicaciones, Universidade da Coruña, A Coruña, España
- Rivero, D., Rabuñal, J. R., Dorado, J. & Pazos, A. (2005). Time Series Forecast with Anticipation using Genetic Programming. En J. Cabestany & A. S. Prieto, *LNC3: Computational Intelligence and Bioinspired Systems*, IWANN 2005. (Springer ed., Vol. 3512, págs. 968-975). Verlag Berlin-Heidelberg.
- Rosca, J. P. & Ballard, D.H.(1996). Discovery of sboutines in genetic programming. En *Advances in genetic programming: volume 2* (Vol. 2, pp. 177-201). Cambridge, MA, USA: MIT Press .
- Sonebi, M. C. & Cevik, A. (2009). Genetic programming based formulation for fresh and hardened properties of self-compacting concrete containing pulverised fuel ash, Constr Build Mater. *Construction and Building Materials* , 23 (7), 2614-2622.

- Soule, T. F. & Foster, J. A. (1997). Code Size and Depth Flows in Genetic Programming. *Genetic Programming. In Proceedings of the Second Annual Conference.* (pp. 313-320). San Francisco, CA .
- Soule, T. F. (1998). *Code Growth in Genetic Programming.* Tesis doctoral, University of Idaho, Moscow, USA.
- Thompson, M. K., Jirsa, J. O., Breen, J. E., & Klingner R. E. (2002) *Anchorage behavior of headed reinforcement: literature review.* Research Report 1855-. The University of Texas at Austin. Austin:USA
- Troxell, G. E., Raphael, J. M. & Davis, H. M. (1958). Long term creep and shrinkage test of plain and reinforced concrete. *Proceedings of the American Society for Testing and Materials (ASTM)*, 58, págs. 1101-1120. Philadelphia, PA.
- Vecchio, F. J. & Collins, M. P. (1986). The modified compression field theory for reinforced concrete elements subjected to shear. *ACI Structural Journal* , 83 (2), 219-231.
- Woolson, I. H. (1905). Some remarkable tests indicating flow of concrete under pressure. *Engineering News* , 54, 454.
- Yang, J. H. & Honavar, V. G. (1998). Feature Subset Selection Using a Genetic Algorithm. *IEEE Intelligent Systems*, 13 (2), 44-49.
- Zhang, Y. & Rockett, P.I. (2007). Multiobjective Genetic Programming Feature Extraction with Optimized Dimensionality. *Soft Computing in Industrial Applications*, ASC 39, 159-168.

*“Lo que cuenta al final no son los años de tu vida sino
la vida de tus años”
Abraham Lincoln*