



---

**Universidad de Valladolid**

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN

DEPARTAMENTO DE TEORÍA DE LA SEÑAL Y COMUNICACIONES E  
INGENIERÍA TELEMÁTICA

TESIS DOCTORAL:

**CONTRIBUCIÓN A LA APLICACIÓN DE TÉCNICAS DE  
INTELIGENCIA ARTIFICIAL PARA EL DISEÑO EFECTIVO DE  
SISTEMAS ADAPTATIVOS DE APRENDIZAJE COMPETITIVO**

Presentada por ELENA VERDÚ PÉREZ para optar al grado de  
doctora por la Universidad de Valladolid

Dirigida por:  
María Jesús Verdú Pérez  
Luisa María Regueras Santos

*Esta tesis doctoral se la dedico a mi familia, especialmente a mis padres porque sin su inestimable ayuda y su apoyo incondicional no hubiera podido realizar este trabajo, y a mis hijas Diana y Elena, que, sin saberlo, han sido mi fuente de energía y mi inyección de voluntad.*

*Además, quería agradecer a mis compañeros y amigos María Jesús, Luisa y Juan Pablo, por su apoyo, consejos, y ayuda durante el desarrollo de esta tesis.*

*Por último, quería también dar las gracias a todos los profesores y compañeros que de una manera u otra me han ayudado en el desarrollo tanto de mi carrera académica como profesional en la Universidad de Valladolid y en CEDETEL.*

*Julio de 2010*

# RESUMEN

El objetivo principal de esta tesis consiste en la *propuesta y validación de métodos basados en técnicas de Inteligencia Artificial para la estimación del nivel de dificultad de los desafíos propuestos en el Entorno On-line de Aprendizaje Competitivo QUESTOURnament, que permita el posterior establecimiento de concursos o itinerarios de aprendizaje para grupos de alumnos según su nivel de conocimiento*. QUESTOURnament es una herramienta telemática que permite el desarrollo de concursos y tiene un fuerte carácter competitivo. Un estudio detallado del estado de arte de los sistemas para el aprendizaje competitivo ha permitido identificar sus ventajas e inconvenientes. Algunas de las ventajas encontradas son la mejora de los resultados académicos, el aumento de la motivación o la protección de la propiedad intelectual. Entre las desventajas se encuentra el aumento de estrés y presión o las diferencias en motivación existentes entre los alumnos según su posición en la clasificación. Se ha realizado también un estudio del estado de arte de los sistemas de aprendizaje adaptativo, estudiando sus funcionalidades y soluciones técnicas y analizando su efectividad respecto a los resultados académicos y al aumento del grado de satisfacción de los estudiantes.

A partir de los dos estudios anteriores se han podido identificar importantes potencialidades de los sistemas adaptativos para su aplicación al aprendizaje competitivo, como la posibilidad de formar grupos de alumnos con similares capacidades y la posibilidad de presentar preguntas adecuadas al nivel de conocimiento del grupo. Así, se propone el sistema QUESTOURnament adaptativo, que incluye tres estrategias fundamentales: estimar el nivel de dificultad de las preguntas o desafíos presentados en QUESTOURnament, formar grupos de alumnos cuyos resultados de aprendizaje son similares para que compitan entre sí y seleccionar las preguntas adecuadas para cada uno de los grupos de alumnos.

El estudio del estado de arte de los sistemas de aprendizaje adaptativo ha permitido conocer las soluciones técnicas empleadas en los sistemas adaptativos, comprobando que no son aplicables directamente al sistema QUESTOURnament. Por tanto, se propone una nueva solución, basada en algoritmos genéticos y lógica difusa que permite estimar el nivel de dificultad de las preguntas en función del comportamiento de los alumnos al responderlas y, así, reclasificar los desafíos propuestos en la herramienta QUESTOURnament. El sistema ha sido validado contra un grupo de expertos y se puede afirmar que el sistema ha estimado el nivel de dificultad de los desafíos, tal y como lo haría un experto humano.



# ABSTRACT

The main objective of this thesis is the *proposal and validation of Artificial Intelligence techniques to estimate the difficulty level of the challenges posed in the competitive learning tool QUESTOURnament*. This will facilitate the later establishment of different contests or learning paths adapted to the current knowledge of the groups of students.

QUESTOURnament is a telematic tool that allows teachers to organize contests in a competitive environment. A detailed study of the state of the art of competitive learning systems has allowed us to identify their advantages and disadvantages. Some of the advantages are the improvement of the academic outcomes, the increase of motivation or the protection of *intellectual property*. The principal drawbacks are the increase of stress and discouragement or the differences in motivation of different students according to their position in the ranking.

The state of the art of adaptive learning systems has been also studied. This second study examines the evidence for the effectiveness of adaptive learning and the satisfaction level of students when using this type of learning.

From these two studies, some important potentialities of adaptive learning have been identified in order to apply them to competitive learning: the possibility of grouping students with similar capabilities and the possibility of posing questions appropriate for the level of knowledge of the group. Then, the adaptive QUESTOURnament system is proposed. It includes three fundamental strategies: estimating the objective difficulty level of the questions posed in QUESTOURnament, grouping students with similar academic outcomes who will compete among themselves and selecting the questions appropriate for the level of knowledge of each group.

The study of the state of the art of adaptive learning systems shows that the technical solutions that are implemented in adaptive systems are not applicable to QUESTOURnament directly. Therefore, a new solution is proposed: a genetic fuzzy expert system that estimates the difficulty level of the questions having into account the real behavior of students and reclassifies the questions posed in QUESTOURnament. The system has been validated against a group of experts and the conclusion is that it performs successfully like a human expert.





3	<b>Sistema QUESTOURnament adaptativo .....</b>	<b>35</b>
<b>CAPÍTULO 4 SISTEMA EXPERTO PARA LA ESTIMACIÓN DEL NIVEL DE DIFICULTAD EN QUESTOURNAMENT .....</b>		<b>41</b>
1	<b>Descripción del problema .....</b>	<b>41</b>
2	<b>Búsqueda de la solución.....</b>	<b>43</b>
3	<b>Arquitectura del sistema experto genético difuso .....</b>	<b>48</b>
4	<b>Diseño del sistema .....</b>	<b>50</b>
4.1	El sistema genético .....	50
4.1.1	Algoritmo genético básico .....	50
4.1.2	Ejecución del algoritmo para el ajuste de sus parámetros .....	63
4.1.3	Algoritmo genético mejorado con técnicas de diversidad.....	68
4.1.4	Explotación de la solución .....	73
4.2	Generación del modelo difuso .....	75
4.2.1	Variables y funciones de pertenencia.....	76
4.2.2	Base de reglas .....	80
4.2.3	Factor de confianza .....	81
4.2.4	Implicación y composición .....	82
4.2.5	Decodificación o <i>defuzzification</i> .....	83
<b>CAPÍTULO 5 EVALUACIÓN DEL SISTEMA EXPERTO .....</b>		<b>85</b>
1	<b>Diseño de experimentos y casos de estudio .....</b>	<b>85</b>
1.1	Caso de estudio 1 .....	86
1.2	Caso de estudio 2.....	87
2	<b>Validación .....</b>	<b>87</b>
2.1	Metodología.....	87
2.2	Resultados .....	92
2.2.1	Caso de estudio 1 .....	92
2.2.2	Caso de estudio 2 .....	100
<b>CAPÍTULO 6 CONCLUSIONES Y LÍNEAS FUTURAS .....</b>		<b>107</b>
1	<b>Conclusiones .....</b>	<b>107</b>
2	<b>Futuras Líneas de Investigación .....</b>	<b>110</b>
<b>BIBLIOGRAFÍA .....</b>		<b>113</b>

# ÍNDICE DE FIGURAS

FIGURA 1. CONTEXTO, OBJETIVOS Y CONTRIBUCIONES DE LA TESIS DOCTORAL.....	5
FIGURA 2. APRENDIZAJE ACTIVO VS. APRENDIZAJE PASIVO.....	10
FIGURA 3. CONO DE DALE. FTE: (DALE 1969).....	11
FIGURA 4. ORIGEN Y CLASIFICACIÓN DE LOS SISTEMAS DE APRENDIZAJE ADAPTATIVO. FUENTE: ELABORACIÓN PROPIA BASADA EN (BRUSILOVSKY Y PEYLO 2003). .....	18
FIGURA 5. PUNTUACIÓN VARIABLE DE UN DESAFÍO.....	25
FIGURA 6: PANTALLA PRINCIPAL DE UN CONCURSO DESARROLLADO CON QUESTOURNAMENT.....	26
FIGURA 7. ESQUEMA DEL <i>CLUSTERER</i> .....	34
FIGURA 8. SISTEMA QUESTOURNAMENT ADAPTATIVO.....	37
FIGURA 9. EJEMPLO DE USO DEL QUESTOURNAMENT ADAPTATIVO.....	40
FIGURA 10. REPRESENTACIÓN TEMPORAL DE LAS LECTURAS (ACCESOS) Y LAS RESPUESTAS DE VARIOS DESAFÍOS REALIZADAS POR UN ALUMNO (COMPORTAMIENTO HABITUAL). .....	43
FIGURA 11. REPRESENTACIÓN TEMPORAL DE LAS LECTURAS (ACCESOS) Y LAS RESPUESTAS DE VARIOS DESAFÍOS REALIZADAS POR UN ALUMNO (COMPORTAMIENTO ANÓMALO EN EL DESAFÍO 4). .....	43
FIGURA 12. ARQUITECTURA DEL SISTEMA EXPERTO DIFUSO-GENÉTICO .....	49
FIGURA 13. FASES DEL ALGORITMO GENÉTICO BÁSICO .....	51
FIGURA 14. ESPACIO DE BÚSQUEDA TRIDIMENSIONAL.....	53
FIGURA 15. MÉTODO DE SELECCIÓN DE LA RULETA .....	59
FIGURA 16. EVOLUCIÓN DEL FITNESS TOTAL DE LA POBLACIÓN A LO LARGO DE LAS GENERACIONES EN LA (A) EJECUCIÓN 1, (B) EJECUCIÓN 2, (C) EJECUCIÓN 3 Y (D) EJECUCIÓN 4. ....	65
FIGURA 17. EVOLUCIÓN DEL FITNESS DEL MEJOR INDIVIDUO DE LA POBLACIÓN A LO LARGO DE LAS GENERACIONES EN LA (A) EJECUCIÓN 1, (B) EJECUCIÓN 2, (C) EJECUCIÓN 3 Y (D) EJECUCIÓN 4.....	66
FIGURA 18. HISTOGRAMAS DE LA VARIABLE <i>NOTA</i> DE LOS PATRONES DE RESPUESTA CORRESPONDIENTES AL NIVEL (A) FÁCIL (B) ASEQUIBLE Y (C) DIFÍCIL .....	67
FIGURA 19. FASES DEL ALGORITMO GENÉTICO (E. VERDÚ, M. J. VERDÚ, ET AL. 2010).....	70

FIGURA 20. DISTRIBUCIÓN DE LOS NICHOS EN LA RUEDA DE LA RULETA DE ACUERDO AL (A) FITNES Y AL (B) FITNESS COMPARTIDO (O SHARED FITNESS).....	72
FIGURA 21. FITNESS DE LOS INTERVALOS DE TIEMPO ( <i>TIME</i> ) Y NOTA ( <i>GRADE</i> ) PARA LOS NIVELES DE DIFICULTAD (A) DIFÍCIL, (B) ASEQUIBLE Y (C) FÁCIL. (E. VERDÚ, M.J. VERDÚ, ET AL. 2010).....	73
FIGURA 22. SISTEMA DE CONTROL DIFUSO BÁSICO .....	76
FIGURA 23. FUNCIONES DE PERTENENCIA TRAPEZOIDALES GENERADAS A PARTIR DE DOS INTERVALOS ADYACENTES.....	77
FIGURA 24. LOS INTERVALOS SOLAPADOS (IZQUIERDA) DAN LUGAR A NUEVOS INTERVALOS (DERECHA) ..	78
FIGURA 25. UN INTERVALO CONTENIDO EN OTRO INTERVALO (IZQUIERDA) DA LUGAR A NUEVOS INTERVALOS (DERECHA) .....	78
FIGURA 26. FUNCIONES DE PERTENENCIA DE LAS VARIABLES DE ENTRADA: ESTRUCTURA GENERAL.....	79
FIGURA 27. FUNCIONES DE PERTENENCIA DE LA VARIABLE DE SALIDA <i>DIFFICULTY</i> .....	80
FIGURA 28. PROCESO PARA CALCULAR LA MEDIDA DE GRUPO. FUENTE: ELABORACIÓN PROPIA BASADA EN (MOSQUEIRA-REY, MORET-BONILLO, Y FERNÁNDEZ-LEAL 2008) .....	89
FIGURA 29. CONJUNTOS DIFUSOS DE LAS VARIABLES DE ENTRADA Y DE SALIDA PARA EL CASO DE ESTUDIO 1.....	93
FIGURA 30. REPRESENTACIÓN DE LAS 16 REGLAS QUE DESCRIBEN EL SISTEMA (CASO DE ESTUDIO 1).....	93
FIGURA 31. REPRESENTACIÓN F.A.M. DE LAS REGLAS (CASO DE ESTUDIO 1).....	94
FIGURA 32. REPRESENTACIÓN TRIDIMENSIONAL DE LA BASE DE REGLAS DEL CASO DE ESTUDIO 1.....	95
FIGURA 33. CONJUNTOS DIFUSOS DE LAS VARIABLES DE ENTRADA Y SALIDA.....	101
FIGURA 34. REPRESENTACIÓN F.A.M DE LAS REGLAS (ACCESOS=L).....	101
FIGURA 35. REPRESENTACIÓN F.A.M. DE LAS REGLAS (ACCESOS=H) .....	102

# ÍNDICE DE TABLAS

TABLA 1. VENTAJAS Y DESVENTAJAS DEL APRENDIZAJE COMPETITIVO (VERDU Y LORENZO 2010).....	16
TABLA 2. SISTEMAS DE APRENDIZAJE ADAPTATIVO .....	21
TABLA 3. TAMAÑO DEL EFECTO (ES) DE LA MEJORA EN LOS RESULTADOS ACADÉMICOS.....	23
TABLA 4. VALOR NORMALIZADO DEL NIVEL DE SATISFACCIÓN DE LOS ESTUDIANTES.....	24
TABLA 5. TÉCNICAS ADAPTATIVAS APLICADAS AL <i>E-LEARNING</i> .....	44
TABLA 6. FITNESS DEL MEJOR CROMOSOMA PARA DISTINTOS OPERADORES DE CRUCE Y DISTINTOS NÚMEROS DE GENERACIONES N (EL TAMAÑO DE LA POBLACIÓN INICIAL ES 200, $p_c=0.6$ Y $p_m=0.01$ )..	62
TABLA 7. EVOLUCIÓN DEL FITNESS TOTAL (F) Y EL FITNESS DEL MEJOR INDIVIDUO (F) A LO LARGO DE LAS GENERACIONES (EL TAMAÑO DE LA POBLACIÓN INICIAL ES 200, EL OPERADOR DE CRUCE ES BLX-A CON $A=0.3$ , $p_c=0.6$ Y $p_m=0.01$ ) .....	64
TABLA 8. INTERVALOS DE <i>TIEMPO</i> , <i>NOTA</i> Y <i>NÚMERO DE ACCESOS</i> ENCONTRADOS POR EL ALGORITMO GENÉTICO PARA CADA NIVEL DE DIFICULTAD ( $p_c=0.6$ , $p_m=0.01$ , OPERADOR DE CRUCE BLX-0.3, 200 INDIVIDUOS, 10 GENERACIONES).....	66
TABLA 9. TABLA DE CONTINGENCIA.....	90
TABLA 10. JUEGOS DE INTERVALOS ENCONTRADOS POR EL ALGORITMO GENÉTICO PARA EL CASO DE ESTUDIO 1 .....	92
TABLA 11. CLASIFICACIÓN DE LOS DESAFÍOS DEL CASO DE ESTUDIO 1 DE ACUERDO AL SISTEMA EXPERTO Y AL EXPERTO HUMANO .....	96
TABLA 12. BASE DE DATOS DE VALIDACIÓN (CASO DE ESTUDIO 1).....	97
TABLA 13. TABLAS DE CONTINGENCIA DE LOS PARES DE EXPERTOS (CASO DE ESTUDIO 1) .....	98
TABLA 14. MATRIZ DE PESOS DEL NIVEL DE ACUERDO/DESACUERDO (CASO DE ESTUDIO 1) .....	99
TABLA 15. TABLA DE PARES CON LOS VALORES DE <i>KAPPA</i> PONDERADA (CASO DE ESTUDIO 1).....	99
TABLA 16. INTERPRETACIÓN DEL VALOR DE <i>KAPPA</i> .....	99
TABLA 17. JUEGOS DE INTERVALOS ENCONTRADOS POR EL ALGORITMO GENÉTICO PARA EL CASO DE ESTUDIO 2 .....	100

TABLA 18. CLASIFICACIÓN DE LOS DESAFÍOS DE ACUERDO AL SISTEMA EXPERTO Y AL EXPERTO HUMANO .....	103
TABLA 19. BASE DE DATOS DE VALIDACIÓN PARA EL CASO DE ESTUDIO 2.....	104
TABLA 20. TABLAS DE CONTINGENCIA DE LOS PARES DE EXPERTOS (CASO DE ESTUDIO 2).....	105
TABLA 21. TABLA DE PARES CON LOS VALORES DE KAPPA PONDERADA (CASO DE ESTUDIO 2) .....	105

# CAPÍTULO 1

## INTRODUCCIÓN

En los últimos años, se ha llevado a cabo importantes estudios en el campo del aprendizaje y de la formación con el fin de adaptar el proceso educativo a las nuevas necesidades de la Sociedad y del Espacio Europeo de Educación Superior. En este sentido, son muchos los trabajos realizados para definir y desarrollar técnicas de *Aprendizaje Activo* (Bonwell y Eison 1991), donde el alumno pase a ser el elemento central del proceso de aprendizaje: debe conocer la información disponible, seleccionarla y analizarla, “hacer” y experimentar, reflexionar, sintetizar los nuevos conocimientos y aplicarlos, construyendo su propio conocimiento y desarrollando así todas sus capacidades (indagación, síntesis, experimentación, creatividad, etc.). En este contexto se enmarca el sistema de interacción educativa QUESTOURnament, anteriormente llamado QUEST (*Quest Environment for Self-managed Training*). Este sistema, que se ha implementado como un módulo integrable dentro de una plataforma Moodle, consiste en un entorno de trabajo, individual o en equipos, en el que se propone una serie de “desafíos” intelectuales que los alumnos tienen que solucionar en un tiempo límite. El trabajo realizado, una vez evaluado, es recompensado mediante un mecanismo de retribución variable sometido a una serie de reglas que regulan el desarrollo de todo el taller. Las sesiones de trabajo se

organizan como un concurso con su tabla de clasificación, la cual se construye en función de los puntos obtenidos por la labor de los participantes (E. Verdú et al. 2006).

La herramienta QUESTOURnament permite hacer un seguimiento continuo del trabajo personal del alumno, además de proporcionar un registro de su participación. Por otra parte, también ofrece a los profesores la posibilidad de definir sus propios instrumentos de evaluación (pruebas objetivas, libres...) en función del tipo de competencias a evaluar. Según el esquema desarrollado, el alumno aprende del propio proceso de evaluación, mucho más que con el tradicional examen; además, aprende de sus compañeros y adquiere habilidades de reflexión crítica. Asimismo, la incorporación de esta herramienta en el proceso educativo promueve y facilita la evaluación continua y el aprendizaje a través de la retroalimentación; y, lo que es más importante, la evaluación se integra de forma natural en el proceso de aprendizaje. Más concretamente, se ha demostrado que el sistema QUESTOURnament mejora tanto la motivación como los resultados académicos de los estudiantes (L.M. Regueras et al. 2009).

Por otra parte, los últimos estudios indican que, siendo el estudiante el centro del proceso de aprendizaje, es necesario adaptar y personalizar el proceso de aprendizaje a cada individuo puesto que no todos los estudiantes procesan y perciben la información de la misma manera (M Lee 2001). La eficacia y conveniencia del uso de distintas estrategias de aprendizaje difiere para distintos estudiantes y diferentes contextos (D.P. Johnson y R.T. Johnson 1999) (Mehlenbacher et al. 2000); de forma que la capacidad de adaptar de forma inteligente el proceso de aprendizaje a las necesidades y al contexto individual de cada estudiante es un elemento clave en el desarrollo de los nuevos *Sistemas de Aprendizaje Inteligentes*. Más aún, el éxito de los nuevos sistemas de aprendizaje depende de la capacidad de adaptar el sistema al nivel cognitivo del estudiante, incorporando técnicas de Inteligencia Artificial y definiendo así *Sistemas Educativos Inteligentes* (Brusilovsky y Peylo 2003) (Ahmad, Basir, y Hassanein 2004). Por tanto, es necesario dotar al entorno de aprendizaje QUESTOURnament de elementos adaptativos inteligentes; de forma que éste sea capaz de adaptarse a las necesidades y particularidades de cada estudiante.

## 1 Objetivos de la Tesis y Contribuciones

El carácter particular y complejo de la herramienta QUESTOURnament, con un fuerte componente competitivo, hace que sea necesario buscar una solución específica para convertir a QUESTOURnament en un *Sistema Educativo Adaptativo Inteligente*, que se adapte a las características y necesidades particulares de cada alumno. Para adaptar el proceso de aprendizaje a las necesidades del alumno es necesario definir y desarrollar herramientas y métodos que permitan, por una parte, medir el nivel de dificultad del material de aprendizaje (Koutsojannis et al. 2007) (Smoline 2008) (H. Liu et al. 2004) y, por otra, modelar y clasificar el conocimiento del alumno en cada instante (C. Romero, S. Ventura, y Garcia 2008). A partir de ahí, se podrá agrupar a los alumnos por nivel de conocimiento y establecer itinerarios de aprendizaje adaptados a cada grupo, con el objetivo final de adaptar el proceso de aprendizaje del alumno a su nivel de conocimiento a la vez que se mantiene la competitividad propia del sistema QUESTOURnament.

Así, el objetivo de la presente tesis es la *propuesta y validación de métodos basados en técnicas de Inteligencia Artificial para la estimación del nivel de dificultad de los desafíos propuestos en el Entorno On-line de Aprendizaje Competitivo QUESTOURnament, que permita el posterior establecimiento de concursos o itinerarios de aprendizaje para grupos de alumnos según su nivel de conocimiento*. Para poder lograr este objetivo se plantea una serie de **objetivos** parciales que permiten su consecución:

- Identificar las posibilidades de los sistemas adaptativos para potenciar las ventajas y mitigar los inconvenientes existentes en los sistemas de aprendizaje competitivo. Para ello se debe realizar estudios sobre el estado de arte tanto de los sistemas *e-learning* competitivos como de los sistemas *e-learning* adaptativos, analizando las soluciones técnicas y los problemas no resueltos.
- Proponer un sistema de aprendizaje adaptativo para la herramienta QUESTOURnament. Tras identificar las propiedades y características particulares del sistema competitivo QUESTOURnament, se determina cuáles deben ser las funcionalidades a incorporar al sistema para llevar a cabo el proceso de adaptación.
- Proponer herramientas y métodos que permitan medir objetivamente el nivel de dificultad de los desafíos del sistema QUESTOURnament. Para lograr este objetivo, se estudia cómo las distintas técnicas adaptativas se ajustan a las características particulares del sistema competitivo QUESTOURnament a la hora de estimar y adaptar el nivel de dificultad de los desafíos propuestos en función de las necesidades y características de cada estudiante. Tras este análisis previo, se diseña una solución adaptativa basada en el uso conjunto de algoritmos genéticos y de lógica difusa, con el fin de clasificar de manera objetiva los desafíos en distintos niveles de dificultad, abordando la problemática existente debida a las particularidades del sistema, como es el caso del necesario control y mantenimiento de la diversidad en las soluciones que el algoritmo encuentra.
- Evaluar la solución propuesta. Una vez desarrollada la herramienta de estimación del nivel de dificultad de los desafíos propuestos en QUESTOURnament, se demuestra su validez en entornos de aprendizaje real.

El desarrollo de esta tesis doctoral ha dado lugar a las **contribuciones** resumidas a continuación.

En relación a los estudios necesarios para identificar las posibilidades de los sistemas adaptativos a la hora de potenciar las ventajas y mitigar los inconvenientes que existen cuando se emplean sistemas de aprendizaje competitivo, la tesis doctoral incluye:

- Estudio del estado de arte de los sistemas *e-learning* para el aprendizaje competitivo, centrado en el análisis de las ventajas y desventajas de estos sistemas. Este estudio ha sido publicado en (E. Verdú et al. 2008a) (Verdu y Lorenzo 2010). El análisis se completa con un estudio del sistema QUESTOURnament, el cual ha sido utilizado con buenos resultados en entornos de trabajo real tal y como se ha descrito en las publicaciones (E. Verdú et al. 2006) (L.M. Regueras, E. Verdú, M.J. Verdú, J.P. de Castro, et al. 2007) (L.M. Regueras, E. Verdú, M.J. Verdú, M.A. Pérez, et al. 2007) (E. Verdú et al. 2007) (L.M. Regueras et al. 2008) (L.M. Regueras et al. 2009).
- Estudio del estado de arte de los sistemas *e-learning* adaptativos. En este análisis, se estudian las funcionalidades de estos sistemas así como las soluciones técnicas empleadas en su implementación. Asimismo se analiza la efectividad del aprendizaje adaptativo en

situaciones reales, atendiendo tanto al rendimiento académico como al grado de satisfacción de los estudiantes, con el fin de demostrar que el aprendizaje adaptativo aporta mejoras significativas en el proceso de aprendizaje de los estudiantes. Estos análisis han sido publicados en (E. Verdú, L. M. Regueras, M.J. Verdú, et al. 2008). Dentro de este estudio, se analizan con especial atención los sistemas *e-learning* adaptativos para el aprendizaje competitivo, con el fin de conocer en qué consiste la adaptación de los sistemas de aprendizaje competitivo existentes, los beneficios que esta adaptación provee, qué soluciones técnicas se están empleando para la adaptación en este tipo de sistemas, y detectar los problemas no resueltos. Este estudio centrado en sistemas adaptativos de *e-learning* competitivo ha sido publicado en (Verdu et al. 2010).

Estos estudios han permitido identificar tanto los aspectos positivos como los negativos del aprendizaje competitivo y, más concretamente, del sistema QUESTOURnament, así como también analizar las distintas técnicas y posibilidades de adaptación con el fin de concretar y proponer una solución adecuada al sistema. Como se ha mencionado anteriormente los sistemas adaptativos deben incorporar herramientas y métodos tanto para medir el nivel de dificultad del material de aprendizaje como para clasificar el conocimiento de los alumnos y poder así establecer los itinerarios de aprendizaje personalizados. La presente tesis doctoral incluye una propuesta de solución adaptativa que cubre estos aspectos y resuelve el primer punto relativo a la medida objetiva del nivel de dificultad del material que, en el caso de QUESTOURnament, son los desafíos (o preguntas) propuestos. Así, las contribuciones de la tesis se completan con las dos siguientes:

- Definición de las funcionalidades del sistema QUESTOURnament adaptativo y propuesta de métodos y tecnologías que resuelvan los distintos aspectos del problema de adaptación.
- Propuesta, diseño y evaluación de una solución adaptativa basada en un sistema híbrido genético difuso, con el fin de clasificar de manera objetiva los desafíos propuestos en QUESTOURnament en distintos niveles de dificultad. Esta solución ha sido publicada parcialmente en (E. Verdú, L.M. Regueras, et al. 2010) (E. Verdú, M. J. Verdú, et al. 2010).

El contexto, objetivos y contribuciones de la presente tesis se recogen de manera esquemática en la Figura 1.

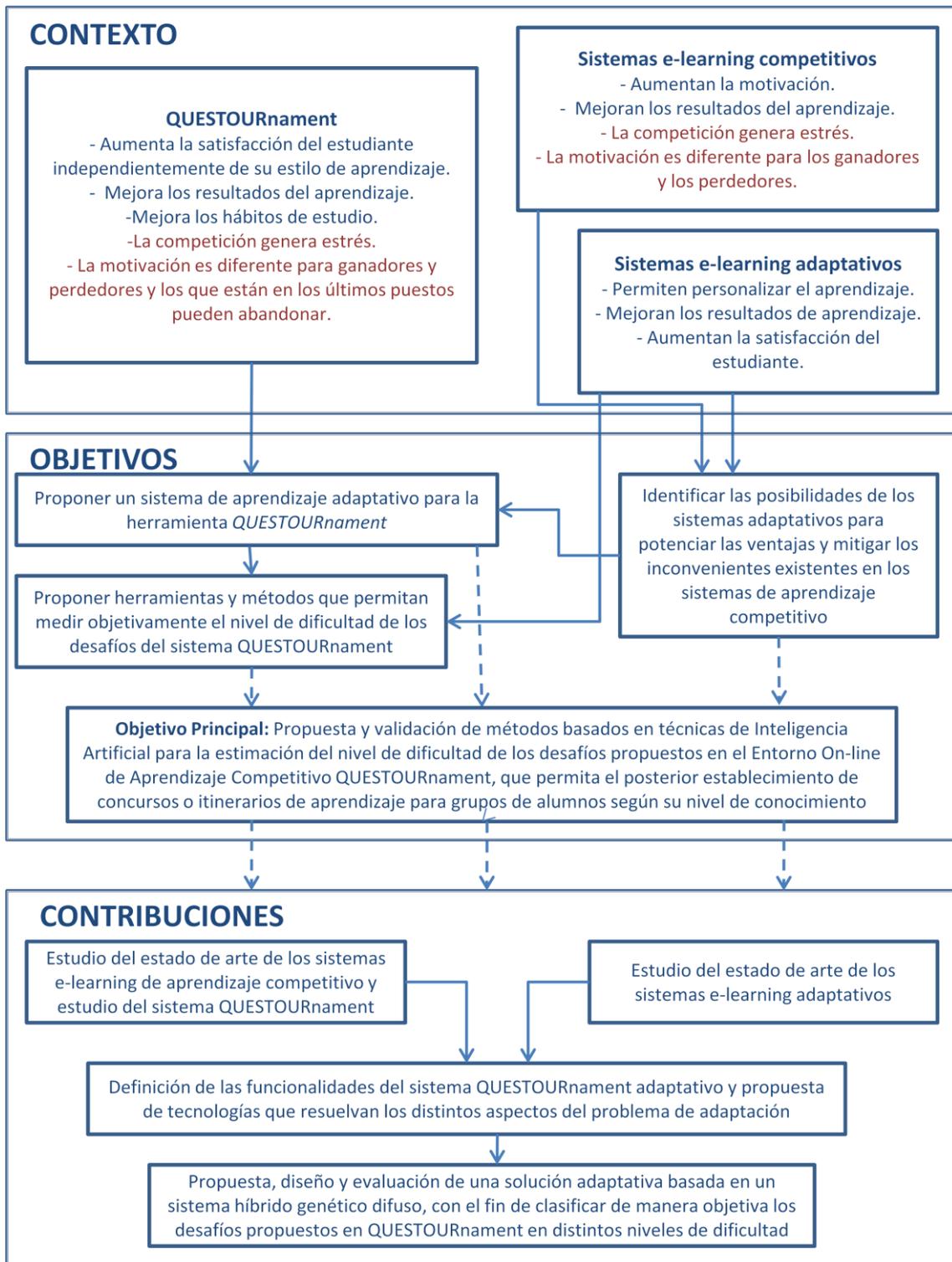


Figura 1. Contexto, objetivos y contribuciones de la tesis doctoral.

## 2 Metodología de la investigación

Para conseguir los objetivos propuestos en esta tesis se ha seguido la metodología de investigación definida en (Adrion 1993), la cual establece que todo proceso de investigación en ingeniería debe seguir las cuatro siguientes etapas:

- *Estudiar las soluciones existentes.* En esta primera fase se identifican las ventajas e inconvenientes existentes en los sistemas de aprendizaje competitivos y se analizan cuáles son las soluciones técnicas aplicadas actualmente a la hora de adaptar los sistemas de aprendizaje a las necesidades y requerimientos de los estudiantes con el fin de determinar cuáles son los problemas aún por resolver (ver Capítulo 2 y apartado 2 del Capítulo 4). Estos estudios se particularizan para un sistema de aprendizaje competitivo concreto, como es la herramienta QUESTOURnament.
- *Proponer mejores soluciones.* Esta segunda etapa consiste en la definición de una solución dirigida a resolver las limitaciones detectadas en la fase anterior. En concreto, a partir del estudio de la literatura realizado se ha podido comprobar cómo dentro de los sistemas de aprendizaje se han aplicado técnicas inteligentes adaptativas con éxito pero en un contexto diferente al aquí presentado debido a las particularidades propias del sistema QUESTOURnament. Asimismo, en este momento se formalizan cuáles deben ser los requisitos y requerimientos del sistema QUESTOURnament adaptativo.
- *Desarrollar una solución mejor.* Una vez propuesta una solución, en la siguiente etapa ésta debe ser desarrollada. En concreto, se define y desarrolla un prototipo de un sistema experto basado en el uso de los algoritmos genéticos y la lógica difusa que permite clasificar de forma objetiva el nivel de dificultad de los desafíos o preguntas propuestas en el sistema QUESTOURnament.
- *Analizar y evaluar la nueva solución.* En esta última etapa, se evalúa la solución propuesta con el fin de validar que ésta supera las limitaciones y problemas que fueron detectados en las fases anteriores. Para ello, se analiza si el sistema experto propuesto permite clasificar los desafíos propuestos en QUESTOURnament tal y como lo haría un experto humano.

Estas cuatro etapas se han ido repitiendo de forma cíclica con el fin de ir mejorando progresivamente la solución final propuesta en la presente tesis.

## 3 Estructura de la memoria

En esta tesis se han cubierto todos los objetivos expuestos anteriormente, tal y como se describe a lo largo de esta memoria, cuya estructura general es la que se muestra a continuación.

Después de esta introducción, en el Capítulo 2 se realiza un detallado estudio del desarrollo y evolución de los sistemas *e-learning* activos; y más concretamente de los sistemas de aprendizaje competitivos y adaptativos. De estos últimos se lleva a cabo también un exhaustivo análisis de su efectividad, ya que son la nueva generación y el futuro de los sistemas educativos. Asimismo, se presenta un sistema particular de aprendizaje competitivo, como es la herramienta QUESTOURnament, y una aproximación a cómo ésta puede ser adaptada con el fin de satisfacer las necesidades y requerimientos de los estudiantes.

En el Capítulo 3 se definen los requisitos del sistema QUESTOURnament adaptativo, que se desglosa en dos módulos: el agrupador o *clusterer*, que agrupa a los alumnos según sus resultados de aprendizaje, y el propio sistema QUESTOURnament adaptativo, que clasifica los desafíos por su nivel de dificultad y propone desafíos adecuados a cada grupo de alumnos que compiten entre sí.

En el Capítulo 4 se propone un sistema experto capaz de estimar el nivel de dificultad de las preguntas propuestas a través de la herramienta competitiva QUESTOURnament. En primer lugar se expone cuáles deben ser los requisitos del sistema y se analizan posibles soluciones técnicas existentes para pasar después a diseñar y desarrollar una solución adecuada a la naturaleza de las preguntas propuestas en el sistema QUESTOURnament.

Una vez definido el sistema experto, en el Capítulo 5 se presentan los resultados obtenidos en el proceso de evaluación y validación del mismo, indicando los experimentos realizados y los casos de estudio llevados a cabo.

Finalmente, en el Capítulo 6 se recogen las conclusiones extraídas a lo largo de esta tesis así como las futuras líneas de investigación que pueden abordarse a partir de la misma.



# CAPÍTULO 2

## ESTADO DE ARTE DE LOS SISTEMAS E-LEARNING COMPETITIVOS Y ADAPTATIVOS

### 1 Aprendizaje Activo

*Los contenidos del presente apartado han sido parcialmente publicados en (E. Verdú et al. 2008a) y (Verdu y Lorenzo 2010)*

Uno de los retos más importantes de los últimos años es la adaptación del sistema educativo universitario a las necesidades y requerimientos del Espacio Europeo de Educación Superior (EEES), lo cual implica la instauración de un nuevo modelo de enseñanza basado en una diferente relación profesor-alumno y en una mayor participación de este último en el

proceso de aprendizaje. En este nuevo modelo el alumno no debe comportarse como un espectador pasivo, recibiendo y memorizando la información que le proporcionan el profesor y el libro de texto; sino que debe tener un papel mucho más activo, debe conocer la información disponible y seleccionarla, analizarla, “hacer” y experimentar, reflexionar, sintetizar los nuevos conocimientos y aplicarlos, construyendo el conocimiento y desarrollando así todas sus capacidades (indagación, síntesis, experimentación, creatividad, etc.).

En este nuevo paradigma, cambian los roles del profesor y el alumno (ver Figura 2). Con la metodología activa, el aprendizaje se centra en el alumno. El papel fundamental del profesor ya no es transmitir la información, sino guiar y orientar el aprendizaje de los alumnos, proveer de los recursos educativos más adecuados, ser tutor, establecer metas, atender las diferencias individuales, desarrollar en los alumnos actitudes positivas, etc.

El aprendizaje activo no es un nuevo método de aprendizaje, aunque en los últimos años ha aumentado mucho el interés por aplicarlo. Numerosos estudios muestran importantes resultados en lo que respecta tanto a la satisfacción e interés de los alumnos como a los resultados académicos obtenidos cuando en el aula se aplican este tipo de metodologías (Mehlenbacher et al. 2000) (Timmerman y Lingard 2003) (R.M. Felder, G.N. Felder, y Dietz 1998). Más aún, las técnicas de aprendizaje activo permiten mejorar la retención a largo plazo, tal y como se muestra en la Figura 3 (Timmerman y Lingard 2003).

No obstante, a pesar de los amplios beneficios que ofrece el aprendizaje activo, actualmente en la enseñanza predomina el modelo tradicional basado en la clase magistral. La aplicación de este nuevo modelo educativo no es una tarea fácil, ya que aparecen diversas dificultades (como el rechazo a los nuevos métodos tanto por parte de los profesores como de los alumnos, el tamaño de los grupos o la disposición de las aulas) que hay que resolver. En este contexto, el uso de las TICs (Tecnologías de la Información y de las Comunicaciones) ofrece un amplio abanico de posibilidades y puede servir para solucionar muchos de estos problemas ya que permite desarrollar programas de trabajo cooperativo y semipresenciales, en los que los alumnos no necesitan desarrollar todo el trabajo en el aula de forma presencial.

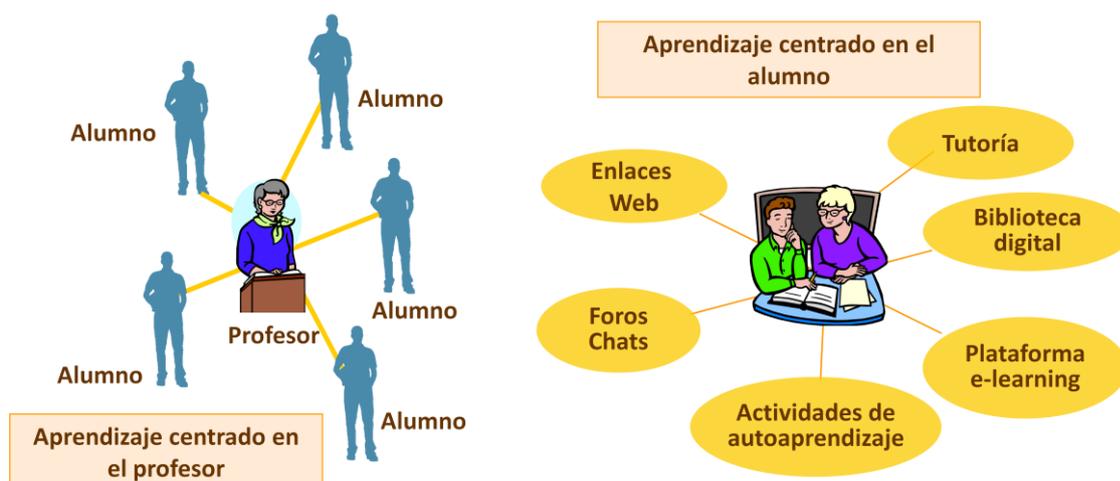


Figura 2. Aprendizaje activo vs. Aprendizaje pasivo.



Figura 3. Cono de Dale. Fte: (Dale 1969)

Los entornos de aprendizaje *on-line* impulsan el pensamiento divergente donde se valoran los múltiples puntos de vista, estimulan la investigación en lugar de imponer pensamientos correctos, ayudan a fomentar la autonomía, pues estimulan que los alumnos creen problemas, seleccionen fuentes y valoren sus juicios, respetando el concepto de comunidad de aprendizaje (Bryndum y Montes 2005). Además, las múltiples posibilidades que ofrecen las tecnologías permiten que las metodologías empleadas se adapten a un mayor rango de estilos de aprendizaje.

Existen varios ejemplos de usos de tecnologías inalámbricas en el aula para mejorar la interacción alumno-profesor con resultados positivos (Y.-F. Chen et al. 2005) (Petr 2005) (Mitra et al. 2005). Gracias a estos sistemas, durante una clase el profesor puede proponer preguntas a los estudiantes con un conjunto de opciones de respuesta que son mostradas en una pantalla. Cada estudiante selecciona la respuesta pulsando un control remoto que la envía al ordenador del profesor. Las respuestas son automáticamente procesadas por el sistema y los resultados pueden ser mostrados gráficamente al profesor y a los estudiantes. Esto es especialmente útil en clases de gran tamaño pues permite la participación de toda la clase. Además el profesor puede controlar, mientras imparte la clase, si los alumnos están comprendiendo los conceptos. Más aún, la realimentación obtenida y el debate que se puede generar pueden resultar muy eficaces.

Por otro lado, los laboratorios virtuales y las simulaciones son otras herramientas muy potentes para el aprendizaje activo (Morozov et al. 2004). (Mickle, Shuman, y Spring 2004) proponen tres técnicas basadas en laboratorios virtuales para que los estudiantes aprendan mejor los conceptos: sitios web con exposiciones multimedia de los experimentos, simulaciones software de experimentos y acceso remoto a equipamiento experimental. La primera de las técnicas no se trata de aprendizaje activo pero sí las otras dos. Un ejemplo de laboratorio virtual que permite la simulación de experimentos de química con gran nivel de interactividad se describe en (Morozov et al. 2004). Por otro lado, un ejemplo de laboratorio controlado remotamente a través de Internet está descrito en (Reguera et al. 2005). Los autores indican que

estos laboratorios estimulan el aprendizaje activo y motivan al estudiante porque “ver es creer” y “hacer es entender”.

También, las nuevas herramientas y servicios relativos a la web 2.0, como *wikis* o *blogs*, presentan nuevas oportunidades para el aprendizaje, facilitando la publicación y compartición de información, la colaboración y la construcción social del conocimiento. En (L.M. Regueras et al. 2008) se explica cómo algunas de estas herramientas de la web 2.0 pueden ser aplicadas a los procesos de aprendizaje.

La aplicación de las TICs en procesos de aprendizaje colaborativo es lo que se viene denominando Aprendizaje Colaborativo Apoyado por Ordenador (CSCL – *Computer Supported Collaborative Learning*), cuyo objetivo es apoyar la participación activa de los estudiantes en la compartición y construcción del conocimiento a través de ordenadores conectados (Prinsen et al. 2009). Hay muchos ejemplos de su utilización en entornos reales que presentan en general una mejora en el papel activo del estudiante.

(van der Linde 2005) emplea la herramienta de trabajo colaborativo BSCW (*Basic Support for Cooperative Working*) como complemento a las clases presenciales. BSCW es una herramienta que proporciona un espacio de trabajo compartido accesible vía web donde se pueden almacenar documentos, gestionar versiones de los mismos, notificar eventos, gestionar grupos, llevar a cabo debates, etc. El propósito de emplear esta herramienta es incrementar el nivel de discusión y el conocimiento compartiendo información fuera de la clase. Sin embargo, (van der Linde 2005) encuentra que, aunque los estudiantes trabajaron con los contenidos *on-line* y participaron en los debates creados por el profesor, hubo muy poca interacción dentro de los grupos y entre grupos. El autor piensa que esto es debido a la falta de experiencia en el uso de herramientas virtuales por parte de los alumnos. Aún así, sólo el 11% de la clase opinó que la plataforma BSCW no les ayudó en el proceso de aprendizaje.

El Grupo de Sistemas Inteligentes y Cooperativos de la Universidad de Valladolid también utiliza BSCW para experiencias de aprendizaje colaborativo basado en proyectos. En algunos de sus estudios afirman que el soporte que ofrece BSCW para compartir documentos es muy importante incluso en escenarios de clases presenciales (Martinez-Mones et al. 2005).

(Frees y Kessler 2004) emplean un conjunto de herramientas para el aprendizaje colaborativo denominadas *Cimel* (*Cimel Collaborative Tools*). Estas herramientas consisten en una lista de contactos, un *chat*, una base de datos denominada *mFAQ* (*multimedia FAQ*) y un conjunto de aplicaciones para compartir el escritorio del ordenador denominadas colectivamente *ShowMe*. Gracias a la herramienta *ShowMe* el profesor puede ver el escritorio del alumno y enviar anotaciones y mensajes de texto al alumno para guiarle en el estudio. Además de enviar anotaciones, se pueden dibujar figuras, líneas, flechas e incluir texto sobre las ventanas compartidas. Una vez la sesión finaliza, el profesor puede guardar la misma para posteriormente incluirla en la base de datos multimedia de preguntas más frecuentes para que todos los alumnos puedan acceder a las mismas. Los autores de esta herramienta encontraron una respuesta positiva por parte de sus alumnos, indicando la mayoría de ellos que esta herramienta sería muy útil si estuviera disponible en todas sus clases.

(Mickle, Shuman, y Spring 2004) presentan otro sistema empleado para el aprendizaje activo que posibilita el trabajo colaborativo denominado CASCADE (*Computer Augmented Support for Collaborative Authoring and Document Editing*), para la edición de documentos colaborativa. Este sistema permite insertar comentarios o diagramas y representaciones gráficas en textos informando sobre quién ha introducido los comentarios, de tal manera que el profesor puede revisar los comentarios realizados por los estudiantes sobre distintos documentos en poco tiempo, gracias a una representación de hipertexto dinámica que proporciona el sistema automáticamente.

Existen muchos ejemplos de aplicación de las TICs al aprendizaje colaborativo y existen muchos casos de éxito de la aplicación de este tipo de estrategias de aprendizaje colaborativo. La colaboración y las sinergias que se crean entre estudiantes pueden dar como resultado un mejor rendimiento de todos (D.P. Johnson y R.T. Johnson 1999). Los estudiantes que participaron en la experiencia colaborativa relatada en (Hernandez, Asensio, y Y. Dimitriadis 2006) percibieron que ésta suponía una mejora sobre los cursos tradicionales, no colaborativos. Igualmente, los estudiantes mostraron más conocimiento y mejor motivación al emplear un método colaborativo y constructivista en (Moreno et al. 2007). Los resultados de (Terenzini et al. 2001) indican que los estudiantes que participaron en cursos en que se emplearon estrategias de aprendizaje activo y colaborativo, desarrollaron mejor sus capacidades de diseño, de comunicación y de grupo. Más aún, en (Springer, Stanne, y Donovan 1999) se analizan 39 estudios sobre el aprendizaje en pequeños grupos y encuentran que los estudiantes que aprendieron en estos grupos tuvieron mejores resultados académicos, mostraron más persistencia y expresaron actitudes más favorables hacia el aprendizaje que sus compañeros enseñados de manera tradicional. También, en (Watson 1995) se aplicó el aprendizaje activo y colaborativo en tres cursos de ingeniería y se encontraron mejores actitudes en los estudiantes como, por ejemplo, más interés en la materia un año más tarde y menor número de ausencias con respecto a las clases pasivas. De cualquier manera, estas técnicas suelen ser más apreciadas en los cursos más avanzados. De hecho, en (B.A. Oakley et al. 2007) se encuentra que la satisfacción de los estudiantes con su equipo aumenta con el nivel de curso. Algunas estrategias de aprendizaje colaborativo requieren madurez en el estudiante para que sean efectivas o satisfactorias.

Sin embargo, este método también presenta inconvenientes y, al igual que todas las estrategias de aprendizaje, las estrategias que emplean aprendizaje colaborativo deben ser diseñadas por el profesor de manera adecuada. El aprendizaje colaborativo no sólo implica pedir a los estudiantes que hagan un trabajo y que formen grupos para ello. La madurez, experiencia en este tipo de estrategias, y perfil de los estudiantes debe ser tenida en cuenta, ya que los estudiantes sin experiencia que llevan a cabo trabajos dentro de grupos, frecuentemente cometen errores comunes que reducen su efectividad (B. Oakley et al. 2004). Por ejemplo, un error común de los estudiantes al trabajar en grupo es que dividen las partes del trabajo, las distribuyen entre los miembros del equipo, las elaboran individualmente, las juntan y entregan el trabajo al profesor, incluso sin revisar el trabajo final global. Otro problema común es el desequilibrio en la carga de trabajo entre los miembros del equipo, o el hecho de que un miembro del equipo no dedique el mínimo tiempo requerido al trabajo. De hecho, en la experiencia relatada en (Verdu y Lorenzo 2010), un estudiante afirma *“cuando la calidad del trabajo de uno de mis compañeros es baja, no es correcto que influya en mi propia nota”*. Estas prácticas no adecuadas pero comunes suponen que los miembros de los grupos no alcancen los objetivos de aprendizaje.

En (Verdu y Lorenzo 2010) se enumeran algunas recomendaciones extraídas de la literatura para evitar estos inconvenientes, como realizar autoevaluaciones por parte del equipo de trabajo o incluir en la nota final dos aspectos, la contribución individual al grupo así como la nota global del grupo. Además, el uso de herramientas telemáticas facilita que el profesor supervise el trabajo de los alumnos si éste ha sido adecuadamente diseñado, con las interacciones y herramientas a utilizar claramente definidas. Más aún, hay herramientas telemáticas que permiten la creación y desarrollo de scripts o guiones de colaboración como por ejemplo Collage y Gridcole (Hernandez, Asensio, y Y. Dimitriadis 2006). Collage ayuda al profesor en el proceso de crear representaciones de diseños pedagógicos colaborativos efectivos

interpretables por un ordenador, mientras que Gridcole permite la representación del diseño guiando a los estudiantes a través de la secuencia de actividades establecida por el profesor.

Uno de los aspectos que más se valoran en el aprendizaje colaborativo es la motivación. Sin embargo, algunos alumnos se sienten más motivados a través de la competición. Por otra parte, la competición por equipos tiene el *doble carácter competitivo y colaborativo* y, por tanto, muchas posibilidades cuando su aplicación se lleva a cabo con un grupo heterogéneo de alumnos.

## 2 Aprendizaje competitivo

*Los contenidos del presente apartado han sido parcialmente publicados en (E. Verdú et al. 2008a) y (Verdu y Lorenzo 2010)*

A través del aprendizaje competitivo se intenta lograr una mayor eficiencia del proceso de aprendizaje de los estudiantes aumentando su motivación a través de la competición. Este tipo de aprendizaje tiene características que pueden llevar a un mayor compromiso de los estudiantes despertando sus instintos competitivos (Anderson 2006).

Existen experiencias reales que prueban que el aprendizaje competitivo puede ser una estrategia de aprendizaje efectiva. Por ejemplo, (Titcomb, Foote, y Carpenter 1994) describen una competición sobre un diseño de ingeniería para estudiantes que todavía no están en la universidad, que proporciona a los participantes experiencia en diseño y les anima a aprender más de ingeniería. Tanto los profesores como los alumnos involucrados en la competición indican un alto nivel de satisfacción con el concurso. (Striegel y Rover 2002) aplican una estrategia colaborativa y competitiva para el aprendizaje de programación y encuentran los siguientes aspectos positivos en el uso de la competición: aumento del espíritu de equipo, protección de la propiedad intelectual (entre los equipos), importante interacción entre el profesor y los estudiantes, y mejores notas en las soluciones a los problemas. Otro ejemplo de uso de la competición para el aprendizaje de programación es el proyecto TOPS descrito en (White et al. 2007). Cuatro universidades participan en este proyecto organizando un concurso en el que los mejores estudiantes compiten resolviendo problemas de programación en equipos. El objetivo de la experiencia es mantener la motivación en los grupos formados por miembros con distinta experiencia y conocimientos y permitir que aquellos mejores programadores puedan sacar el máximo provecho de su habilidad. En línea con estos concursos de programación, se puede destacar la competición internacional de gran éxito *ACM International Collegiate Programming Contest (ICPC)*, que proporciona a los estudiantes universitarios oportunidades para que interactúen con estudiantes de otras universidades y demuestren y mejoren sus habilidades para programar, resolver problemas y trabajar en grupo. A nivel de educación secundaria, existe una competición similar, también de gran éxito, que se denomina *International Olympiad in Informatics (IOI)*. Se trata de competiciones divertidas que motivan a muchos estudiantes de todo el mundo, que aprenden más gracias a estas competiciones. De hecho, una de las herramientas que los estudiantes usan en el entrenamiento para estas competiciones es el *Juez Online* de la UVA ([uva.onlinejudge.org](http://uva.onlinejudge.org)), sitio web que ganó en 2005 el premio internacional Joe DeBlasi como contribución más importante al aprendizaje competitivo. Este sitio tiene más de 100.000 usuarios y suma más de 8 millones de respuestas a

los problemas propuestos desde su lanzamiento en 1995. Estas cifras demuestran el éxito de este tipo de proyectos competitivos.

En otra línea, el aprendizaje competitivo puede desarrollarse a través de los juegos, los cuales en los últimos años han ido incorporándose cada vez más en el ámbito educativo con el fin de promover un aprendizaje más efectivo (Becker 2001) (L. Smith y Mann 2002). Así, por ejemplo, el sistema Joyce es un juego competitivo donde los jugadores tienen que responder correctamente varias preguntas de elección múltiple para ganar el juego (L.J. Chang et al. 2003). (Siddiqui, Khan, y Akhtar 2008) aplican un juego de simulación y han observado un alto nivel de competitividad entre los estudiantes así como un importante aumento de la motivación. Asimismo, (Shifrony y Ginat 1997) desarrollan igualmente un juego de simulación y comprueban cómo con este método se mejora significativamente el nivel de comprensión y motivación de los estudiantes.

Los juegos tienen un amplio número de características que los hacen atractivos desde el punto de vista pedagógico y educativo, ya que capturan el interés del estudiante, promueven el aprendizaje activo y motivan la participación (T.-H. Wang 2008). (Fasli y Michalakopoulos 2005) afirman que “la naturaleza de los juegos y su carácter competitivo actúan como un incentivo para todos los estudiantes”. Por tanto, la implementación de sistemas de aprendizaje basados en juegos es un método efectivo para aumentar no sólo la motivación sino también la diversión y el aprendizaje (Philpot et al. 2005).

(Franken y Brown 1995) estudian los motivos por los que a algunas personas les gusta la competición y encuentran que para algunas de ellas la competición es una oportunidad para mejorar sus capacidades, mientras que a otras simplemente les gusta porque les proporciona la oportunidad de ganar. En muchos otros casos, la competición supone una mayor motivación para esforzarse más, obteniendo mejores resultados de aprendizaje. En este sentido, algunos autores indican que el aprendizaje efectivo se logra cuando la motivación intrínseca, no extrínseca, se favorece. Por tanto, para evitar que sólo ganar sea la motivación de aprender, es importante promocionar otros factores de motivación, como la diversión o la mejora de habilidades. De cualquier manera, ganar puede mejorar la motivación intrínseca mejorando la competencia percibida (Reeve y Deci 1996).

Por otro lado, el aprendizaje competitivo también presenta algunas desventajas. Ante situaciones competitivas algunos alumnos sufren presión, estrés y ansiedad (D.P. Johnson y R.T. Johnson 1999). Además, los ganadores de una competición no están igual de motivados que los que pierden. Los primeros están más motivados y se sienten más competentes que los segundos (Reeve y Deci 1996) (Vansteenkiste y Deci 2003). Cuando en una competición dos equipos están muy igualados, estos equipos obtienen puntuaciones superiores que si no se hubiera dado esta situación dentro de la competición. Sin embargo, cuando un equipo se desmarca de otro durante la competición, el segundo grupo termina abandonando la lucha por ganar (K. Chu, M. Chang, y Hsia 2004). La Tabla 1 contiene un resumen de las ventajas e inconvenientes del aprendizaje competitivo.

Más ejemplos de aplicación de estrategias de aprendizaje competitivo y un análisis detallado de las ventajas e inconvenientes que presenta el uso de este tipo de estrategias, así como el uso de estrategias de aprendizaje colaborativo, se encuentra en (Verdu y Lorenzo 2010).

Ventajas	Desventajas
Los concursos pueden ser divertidos y aumentar la motivación durante el aprendizaje (Conti, Collins, y Picariello 2001) (L.J. Chang et al. 2003) (Franken y Brown 1995).	Los estudiantes pueden estresarse y sentir ansia (D.P. Johnson y R.T. Johnson 1999) y la presión durante una situación competitiva disminuye la motivación intrínseca (Reeve y Deci 1996).
Protección de la propiedad intelectual (Striegel y Rover 2002). Los trabajos copiados se reducen puesto que los estudiantes evitan compartir sus trabajos.	Los estudiantes pueden centrarse demasiado en premios extrínsecos en vez de en el aprendizaje (D.P. Johnson y R.T. Johnson 1999).
Mejores resultados académicos (Striegel y Rover 2002) (L.M. Regueras et al. 2009).	La motivación es distinta entre los ganadores y los perdedores. Los ganadores están más motivados intrínsecamente (Reeve y Deci 1996) (Vansteenkiste y Deci 2003) y se sienten más competentes que los perdedores (Vansteenkiste y Deci 2003).
Ganar aumenta la motivación mejorando la competencia percibida (Reeve y Deci 1996)	La competición puede tener efectos negativos en la creatividad artística de los niños (Amabile 1982).
Cuando se compite en equipos, el espíritu de equipo aumenta (Striegel y Rover 2002).	Posibles relaciones negativas entre los compañeros y con los profesores (D.P. Johnson y R.T. Johnson 1999).

Tabla 1. Ventajas y desventajas del aprendizaje competitivo (Verdu y Lorenzo 2010)

El sistema QUESTOURnament es una herramienta telemática que permite el aprendizaje competitivo y colaborativo. Se trata de una herramienta versátil, no específica para un tema concreto, sino válida para cualquier materia. Se ha demostrado que el uso de este sistema puede mejorar los resultados de aprendizaje así como aumentar la satisfacción y motivación de los estudiantes (L.M. Regueras et al. 2009); aunque también hace que algunos estudiantes sientan estrés por la competición. Gracias a que el sistema QUESTOURnament está basado en las TICs, los profesores pueden diseñar estrategias de aprendizaje adaptadas a la competición en modo anónimo, por ejemplo, eliminando así algunos de los inconvenientes que presenta el aprendizaje competitivo (E. Verdú et al. 2006).

### 3 Aprendizaje Adaptativo

*Los contenidos del presente apartado han sido parcialmente publicados en (E. Verdú, L. M. Regueras, M.J. Verdú, et al. 2008).*

El concepto de adaptación ha sido un importante tema de investigación para los sistemas educativos en los últimos años. Recientes investigaciones muestran cómo el uso de sistemas adaptativos en la educación puede proporcionar un mejor entorno de aprendizaje, ya que no todos los estudiantes perciben y procesan la información igual (M Lee 2001). De esta forma, los sistemas educativos adaptativos se han convertido en una alternativa a los sistemas educativos

tradicionales, y pueden considerarse como la siguiente generación de *e-learning*. Más aún, algunos autores opinan que el éxito de los nuevos sistemas de aprendizaje depende de la capacidad de adaptar el sistema al nivel cognitivo del estudiante, incorporando técnicas de Inteligencia Artificial y definiendo así Sistemas Educativos Inteligentes (Brusilovsky y Peylo 2003) (Ahmad, Basir, y Hassanein 2004).

A la hora de analizar la literatura existente sobre aprendizaje adaptativo puede haber confusión, ya que la mayoría de las veces los sistemas educativos adaptativos son referenciados como sistemas educativos inteligentes. Sin embargo, ambos términos no siempre son sinónimos. Los sistemas adaptativos “intentan adaptarse a los diferentes estudiantes y grupos de estudiantes teniendo en cuenta información acumulada de los modelos de ese estudiante individual o grupo de estudiantes” (Brusilovsky y Peylo 2003); mientras que los sistemas inteligentes aplican técnicas del campo de la Inteligencia Artificial (IA) para proporcionar un mayor y mejor soporte a los usuarios de sistemas educativos basados en TICs (Brusilovsky y Peylo 2003).

Por otra parte, aunque no hay una única clasificación de los sistemas de aprendizaje adaptativos (Brusilovsky y Peylo 2003) (Mödrtscher 2007), la mayoría de las clasificaciones existentes incluye las dos categorías clásicas: AH (*Adaptive Hypermedia*) e IT (*Intelligent Tutoring*). Así, para el presente análisis de los sistemas de aprendizaje adaptativos se ha establecido una clasificación basada en estas dos categorías. Además, también se tienen en cuenta las tres tecnologías para IT (*Curriculum Sequencing*, *Interactive Problem Solving Support* e *Intelligent Solution Analysis*) definidas por (Brusilovsky y Peylo 2003) y algunas tecnologías inspiradas en Web como aquellas relacionadas con ICL (*Intelligent Collaborative Learning*).

Los sistemas educativos inteligentes o ITS (*Intelligent Tutoring System*) imitan el proceso de aprendizaje entre el profesor y el estudiante, añadiendo nuevas facilidades y utilidades propias de la tecnología. Por lo tanto, un ITS completo debería incluir, aunque no siempre es así, todos los componentes del proceso de aprendizaje: representar el contenido, implementar la estrategia de aprendizaje y proporcionar un mecanismo para evaluar el progreso del estudiante (Mödrtscher 2007).

Por otro lado, los sistemas adaptativos hipermedia o AHS (*Adaptive Hypermedia System*) están inspirados en el IT. Los AHS intentan combinar sistemas educativos adaptativos y basados en hipermedia. Según (Brusilovsky 1998), el sistema adaptativo hipermedia debería satisfacer tres criterios: (1) ser un hipermedia o hipertexto, (2) tener un modelo de usuario, y (3) ser capaz de usar el modelo de usuario. En muchos casos esta adaptación se consigue usando técnicas de IA, por lo que estos sistemas deberían ser considerados también ITS. (Brusilovsky y Peylo 2003) encuadran a estos últimos sistemas dentro de la tecnología *curriculum sequencing*. La Figura 4 muestra una clasificación de los sistemas de aprendizaje adaptativo, la cual está basada en (Brusilovsky y Peylo 2003).

Los sistemas ITES (G. Hwang 2003), Logicando (Lanzilotti y Roselli 2007), IAELS (Y.-H. Chang, T.-Y. Lu, y Fang 2007) y el sistema presentado en (Kavcic 2004) son buenos ejemplos de tecnologías *curriculum sequencing*, que adaptan los contenidos en función del modelo del estudiante o su nivel de conocimiento. Muchos sistemas adaptativos focalizan los esfuerzos en la evaluación (exámenes y auto-evaluaciones) en lugar de en la presentación de contenido. ITES es un sistema basado en Web que usa un sistema experto difuso y un mapa conceptual para construir tests y caminos de aprendizaje basados en el nivel de aprendizaje de cada alumno. Logicando es un sistema hipermedia de aprendizaje, el cual usa un sistema experto y reglas para adaptar el contenido al nivel de conocimiento de cada alumno. IAELS incorpora agentes inteligentes para hacer cursos personales y usa el algoritmo *a priori* para encontrar el mejor camino de aprendizaje para cada estudiante. (Kavcic 2004) describe un sistema que personaliza

la secuencia educativa a través de un modelo de usuario difuso y reglas lingüísticas para su actualización dinámica.

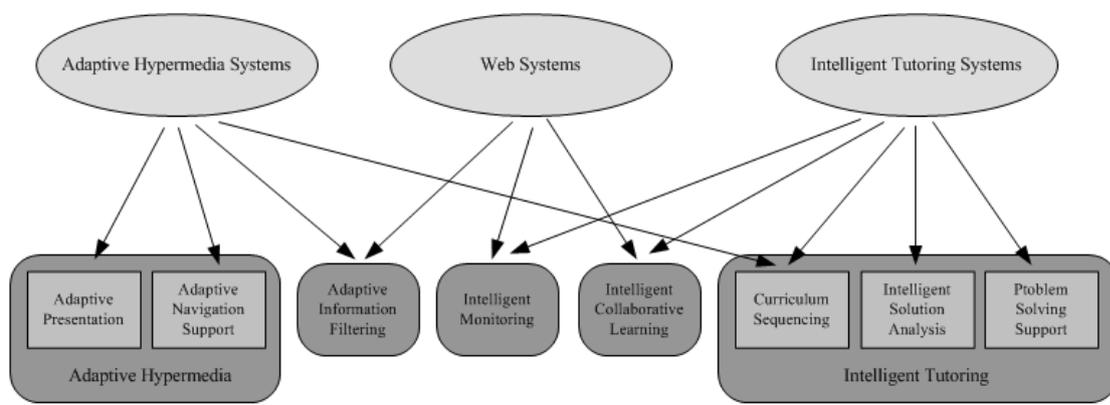


Figura 4. Origen y clasificación de los sistemas de aprendizaje adaptativo. Fuente: elaboración propia basada en (Brusilovsky y Peylo 2003).

Muchos sistemas adaptativos centran la adaptación en la evaluación. Así por ejemplo, SIETTE (Conejo et al. 2004) (Guzman, Conejo, y Perez-de-la-Cruz 2007) emula los exámenes orales e infiere conocimiento del estudiante a través de los tests adaptativos; proponiendo preguntas al estudiante adaptadas a su nivel de conocimiento. Además, los tests de auto-evaluación pueden ofrecer pistas con la pregunta o *feedback* con la respuesta. Otros ejemplos de este tipo de sistemas son los descritos en (Tai, T.-A. Tsai, y F.M.-C. Chen 2001) (Lilley, Barker, y Britton 2004) (Nouh, Karthikeyani, y Nadarajan 2006) (Vozár y Bieliková 2008). Los sistemas enfocados a la evaluación suelen usar la Teoría de Respuesta al Ítem (TRI) como base de su solución.

Se encuentran también otros muchos ejemplos de sistemas AHS (Nirmalakhandan 2007); (Lo, H.-M. Wang, y Yeh 2004) (C.-M. Chen, H.-M. Lee, y Y.-H. Chen 2005) (F. Muñoz y Ortigosa 2006). Por ejemplo, el sistema HELP (Lo, H.-M. Wang, y Yeh 2004) es un sistema de aprendizaje hipertexto para la enseñanza de las preposiciones en inglés que proporciona *feedback* adaptativo e instrucciones de refuerzo, a través de hiperenlaces activos. PEL-IRT (C.-M. Chen, H.-M. Lee, y Y.-H. Chen 2005) usa dos preguntas muy simples como *feedback* de los estudiantes: “¿Comprendes el contenido de los materiales recomendados del curso? (sí o no)” y “¿qué piensas sobre su dificultad? (muy fácil, fácil, moderado, difícil, muy difícil)”. El sistema aplica la MLE (*Maximum Likelihood Estimation*) a las respuestas a la primera pregunta (grado de comprensión) para estimar la habilidad de los estudiantes. Luego, recomienda los materiales del curso adecuados a cada estudiante tomando como base esa estimación. El nivel de dificultad de los materiales se ajusta dinámicamente en función de un esquema de voto colaborativo basado en el *feedback* sobre el nivel de dificultad proporcionado por los estudiantes.

Mientras la mayoría de los sistemas adaptativos utilizan una única fuente de personalización, (J Tseng et al. 2008) y (Hatzilygeroudis, Giannoulis, y Koutsojannis 2005) emplean dos fuentes de personalización, comportamiento del aprendizaje y estilo de aprendizaje, para determinar el nivel de dificultad de los materiales y el estilo de presentación (hipertexto, secuencial...). En concreto, TSAL (J Tseng et al. 2008) usa el estilo de aprendizaje para determinar el estilo de presentación (hipertexto, secuencial...) y, posteriormente, el nivel de dificultad de los materiales se adapta en función del comportamiento de aprendizaje, el cual comprende tanto los resultados como el tiempo utilizado en hacer las tareas (eficiencia de

aprendizaje y grado de concentración). (Hatzilygeroudis, Giannoulis, y Koutsojannis 2005) también incluyen en su modelo tanto el estilo de aprendizaje (teórico, pragmático y constructivista) como el nivel de conocimiento de los estudiantes. Su sistema usa el modelo del estudiante para ofrecer presentación adaptativa, navegación adaptativa y sugerencias personalizadas sobre el camino de aprendizaje más adecuado. Además, para estimar el nivel de conocimiento de los estudiantes usa un sistema experto basado en reglas.

Otros sistemas más complejos, como APeLS (Conlan 2005) (Dagger, Conlan, y Wade 2005) y aLFanet (Van Rosmalen et al. 2006) (Fuentes et al. 2005) (O. C. Santos, Boticario, y Barrera 2004) permiten a los diseñadores de los cursos elegir la adaptación requerida entre múltiples opciones. Por ejemplo, con aLFanet las opciones adaptativas comprenden contenidos adaptativos, auto-evaluaciones adaptativas y recomendaciones dinámicas (objetos de aprendizaje, evaluaciones...) para los estudiantes durante el curso. ALFanet usa diferentes fuentes de información de personalización relacionada con el perfil del estudiante: nivel previo de conocimiento, interés y comportamiento de aprendizaje o progreso en el curso. APeLS es un servicio de e-learning personalizado basado en un motor adaptativo genérico que emplea un esquema de metadatos multimodelo para la adaptación. Este esquema de metadatos se consigue: 1) desarrollando un mecanismo para modelar los estudiantes, capturando tanto su conocimiento inicial como sus preferencias de estilo de aprendizaje así como el conocimiento adquirido a través del uso de cursos personalizados, 2) seleccionando, por medio de un sistema experto, una narrativa apropiada de acuerdo al esquema pedagógico deseado y 3) creando unos metadatos apropiados a los recursos de e-learning describiendo sus características adaptativas con el fin de facilitar poder reutilizar elementos de adaptabilidad.

Como se ha comentado anteriormente, algunos esquemas de ITS tienen en cuenta más de un aspecto del proceso de aprendizaje. Por ejemplo, Logic-ITA (Yacef 2005) proporciona *curriculum sequencing* pero también *interactive problem solving support*, ya que proporciona ayuda inteligente en cada paso guiando al estudiante hacia la solución correcta del problema. KERMIT (Suraweera y Mitrovic 2004) es otro ejemplo de este tipo de sistemas “doble-inteligentes”, el cual usa CBM (*Constraint-Based Modelling*) para implementar el modelo del estudiante y el conocimiento del dominio.

Andes (K. VanLehn et al. 2005) y PAT (Koedinger et al. 1997) son otros dos sistemas *interactive problem solving support*. Andes es un sistema para el aprendizaje de física que proporciona *feedback* inmediato y ayuda en cada paso, tanto si el estudiante la solicita como si comete errores. PAT es otro sistema inteligente del mismo tipo que Andes, pero aplicado al aprendizaje del álgebra.

(G.-L. Hwang 2007) también combina dos tecnologías ITS: *curriculum sequencing* e *intelligent solution analysis*. Este tipo de sistemas analiza la solución dada por un estudiante con el fin de poder informarle de qué está mal o incompleto y de qué conceptos debería repasar para solucionar sus errores (Brusilovsky 1998). CAPIT (Mayo y Mitrovic 2001) y Conceptual Helper (Albacete y K.A. VanLehn 2000) también implementan esta tecnología ITS haciendo uso de las Redes Bayesianas. (Noguez y Sucar 2005) usan Modelos Relacionales Probabilísticos (los cuales integran Redes Bayesianas y las facilidades de los modelos relacionales) para modelar al estudiante. El sistema obtiene los valores iniciales de un nuevo estudiante a partir de otros modelos de estudiantes con características similares. En relación al empleo de estas técnicas, en (C. Gonzalez, Burguillo, y Llamas 2006) se lleva a cabo una comparativa entre las Redes Bayesianas y el Razonamiento Basado en Casos a la hora de modelar a los estudiantes y se decanta por el uso del Razonamiento Basado en Casos, entre otras cosas por la menor carga computacional que implica, lo que es muy importante para el caso de aplicaciones Web, que pueden tener un gran número de usuarios, como se indica en (Mangina y Kilbride 2008).

Los últimos esquemas ITS a analizar son aquellos incluidos en el grupo de *Intelligent Collaborative Learning*. (Jong et al. 2006) usan técnicas para formar grupos adaptativos y *peer help*. Los estudiantes con grafos conceptuales complementarios son agrupados para aprender los unos de los otros mientras trabajan juntos. Otro ejemplo de tecnología *peer help* que usa los mapas conceptuales es el Recomendador de Mentores propuesto en (Wei, L.-Y. Lee, y G.-D. Chen 2004). (G. Chen, C. Chang, y C. Wang 2008) implementan un módulo para recomendar un mentor usando mapas conceptuales y modelos de estudiantes en un entorno de aprendizaje Web ubicuo el cual incluye también algo de *curriculum sequencing*. Además de adaptarse al comportamiento del estudiante y su nivel de conocimiento, el sistema también se adapta al tipo de dispositivo usado (PCs, teléfonos móviles, PDA...) para acceder a los materiales de aprendizaje.

La Tabla 2 recopila y muestra un resumen de los distintos sistemas adaptativos encontrados en la literatura. Estos sistemas aplican técnicas como redes bayesianas, lógica difusa, *clustering* y algoritmos genéticos, entre otras, para adaptar su comportamiento a los objetivos, tareas y características de los alumnos.

Finalmente, si bien son muchas las referencias bibliográficas que comentan las ventajas del aprendizaje adaptativo, no hay demasiados estudios donde se haga un análisis exhaustivo de las mismas. Por tanto, a continuación, se va a examinar cuál es el grado de efectividad y la opinión de los estudiantes con relación a algunos de los sistemas de aprendizaje adaptativo comentados anteriormente.

### 3.1 Efectividad del aprendizaje adaptativo

Para conocer cómo de efectivo es el aprendizaje adaptativo se han estudiado las referencias disponibles de estos sistemas, evaluando su efectividad respecto a los resultados de aprendizaje. Así, se ha tratado de responder a la siguiente cuestión: *¿Pueden realmente los estudiantes mejorar su conocimiento cuando el sistema de aprendizaje se adapta a su perfil y/o rendimiento?*

Además, también se ha analizado la satisfacción de los estudiantes cuando usan sistemas de aprendizaje adaptativo, puesto que al final, ellos son los destinatarios de los mismos y por ello, es importante conocer su percepción de aprendizaje y satisfacción cuando usan sistemas de este tipo.

#### 3.1.1 Metodología

Un problema importante a la hora de determinar la efectividad de un sistema es decidir cuándo una mejora es significativa. No es suficiente comprobar que la mejora media lograda con un grupo experimental es mayor que la mejora media lograda con el grupo de control en un experimento particular, sino que se necesita conocer, además, hasta dónde se puede generalizar este hallazgo cuando se tienen en cuenta las limitaciones del experimento. En definitiva, no es suficiente con identificar la ocurrencia de cierto efecto; se requiere adicionalmente determinar su magnitud o tamaño (Cohen 1992). Con tal propósito se han desarrollado diversas técnicas formales que permiten cuantificar el tamaño del efecto o ES (*Effect Size*).

Referencia	Descripción
(Albacete y K.A. VanLehn 2000)	Conceptual Helper: ITS
(C.-M. Chen, H.-M. Lee, y Y.-H. Chen 2005)	PEL-IRT: Sistema de e-learning personalizado usando TRI
(G. Chen, C. Chang, y C. Wang 2008)	Estado de aprendizaje y <i>peer help</i> en un entorno de aprendizaje ubicuo
(C. Chen 2008)	Sistema de e-learning personalizado usando TRI y algoritmos genéticos
(Y.-H. Chang, T.-Y. Lu, y Fang 2007)	IAELS: Sistema adaptativo de e-learning basado en agentes inteligentes
(Conlan 2005)	APeLS: Servicio de e-learning personalizado adaptativo
(Guzman, Conejo, y Perez-de-la-Cruz 2007)	SIETTE: Tests de auto-evaluación
(Hatzilygeroudis, Giannoulis, y Koutsojannis 2005)	Sistema de educación inteligente basado en Web
(G. Hwang 2003)	Modelo de mapa conceptual para desarrollar ITS
(G.-L. Hwang 2007)	Esquema de pronóstico de Gray
(Jong et al. 2006)	Producción de material de aprendizaje para actividades de grupo
(Kavcic 2004)	Sistema Educativo Hipermedia Adaptativo
(Koedinger et al. 1997)	PAT: ITS para solucionar problemas de álgebra
(Lanzilotti y Roselli 2007)	Logiocando: Sistema hipermedia de tutoría inteligente
(Lilley, Barker, y Britton 2004)	Sistema de tests adaptativos basado en ordenador
(Lo, H.-M. Wang, y Yeh 2004)	HELP: Sistema de aprendizaje de inglés basado en hipermedia
(Mayo y Mitrovic 2001)	CAPIT: Tutor basado en restricciones
(F. Muñoz y Ortigosa 2006)	TANGOW: Hipermedia adaptativo
(Nirmalakhandan 2007)	Esquema de evaluación y tutorial adaptativo
(Noguez y Sucar 2005)	ITS para laboratorios virtuales
(Nouh, Karthikeyani, y Nadarajan 2006)	ITS basado en redes Bayesianas y TRI
(Own 2006)	Curso Web adaptativo
(Suraweera y Mitrovic 2004)	KERMIT: ITS para Modelo Entidad/Relación
(Tai, T.-A. Tsai, y F.M.-C. Chen 2001)	Sistema de aprendizaje adaptativo para el manejo del teclado chino
(J Tseng et al. 2008)	TSAL: Aprendizaje adaptativo de dos fuentes
(Van Rosmalen et al. 2006)	aLFanet: Plataforma de <i>e-learning</i> adaptativa
(K. VanLehn et al. 2005)	Andes: ITS para solucionar problemas de física
(Wei, L.-Y. Lee, y G.-D. Chen 2004)	Mentor adaptativo en un contexto de aprendizaje colaborativo
(Yacef 2005)	Logic-ITA: Sistema inteligente de ayuda a la enseñanza

Tabla 2. Sistemas de aprendizaje adaptativo

ES es una medida estadística que se suele usar para cuantificar la efectividad de una intervención particular, relativa a alguna comparación, por ejemplo, entre un grupo de control y un grupo experimental. De hecho, es una forma estándar de comparar resultados de un experimento pedagógico a otro.

ES pone el énfasis en el tamaño del efecto en lugar de en su significancia estadística; de esta forma proporciona un esquema más científico a la acumulación del conocimiento (Coe 2002). ES puede medirse como la diferencia en medias de una condición de comparación entre un grupo experimental y un grupo de control dividido por la desviación estándar ponderada de los grupos (Frías Navarro, Llobell, y García Pérez 2000). En este estudio, para calcular ES a partir de los datos obtenidos de experimentos publicados se ha utilizado una metodología simplificada proporcionada por (Thalheimer y Cook 2002).

(Cohen 1992) sugiere que valores de ES de 0,20 muestran una pequeña significancia, valores de 0,50 una significancia media y valores de 0,80 una alta significancia. Sin embargo, según (Prince 2004), se deberían tener en cuenta mejoras en logros académicos aunque el ES correspondiente esté por debajo del límite de 0,80.

Finalmente, cabe señalar que la mayoría de investigaciones en el campo del aprendizaje adaptativo no proporcionan datos sobre las mejoras obtenidas en los resultados académicos de los alumnos.

### 3.1.2 Resultados

En la Tabla 3 se muestran los resultados de la evaluación de una serie de sistemas de aprendizaje adaptativo con el fin de probar si este tipo de sistemas mejora el rendimiento de los alumnos. En este estudio se han incluido aquellos sistemas de los que se dispone de los datos necesarios para el análisis ya que existen muchas publicaciones que no ofrecen estos datos. En términos generales, puede verse cómo los resultados obtenidos son positivos aunque con diferente significancia.

Respecto a los sistemas del grupo *Interactive Problem Solving Support*, en el análisis de efectividad realizado en esta tesis, se obtienen resultados positivos aunque de distinta significancia. Por ejemplo, (K. VanLehn et al. 2005) encuentran que los estudiantes del grupo experimental que usaron el sistema Andes obtuvieron mejores resultados académicos que los estudiantes del grupo de control. Sin embargo, el ES es menor en el examen final (0.25) que en los exámenes parciales (0.61). Los experimentos con PAT (Koedinger et al. 1997) dan lugar a distintos ES para distintos tipos de exámenes. Logic-ITA (Yacef 2005) da lugar a ES medianos y grandes, aumentando en los distintos años académicos (0,66 a 1,05). Estos valores podrían estar afectados por el componente *curriculum sequencing* del sistema. De hecho, los sistemas del tipo *curriculum sequencing*, ya sean AHS o ITS, obtienen todos buenos resultados, excepto para el caso del sistema tutor descrito en (Lanzilotti y Roselli 2007) donde la mejora no es significativa.

Uno de los casos más interesantes es el presentado por (J Tseng et al. 2008). Los autores comparan tres grupos. El primero usa un sistema adaptativo basado en el estilo de aprendizaje y la habilidad del estudiante (grupo experimental). El segundo usa el mismo sistema adaptativo pero sólo basado en la habilidad del estudiante (grupo de control 1). El último usa un curso hipermedia no adaptativo (grupo de control 2). El análisis estadístico de los resultados realizado en esta tesis muestra cómo la adaptación ayuda a los estudiantes a mejorar su aprendizaje (con un valor de ES elevado para el grupo de control 1 y el grupo experimental cuando se compara con el grupo de control 2). Además, cuando se comparan los dos sistemas adaptativos, el ES es insignificante (0,14), indicando que el estilo de aprendizaje no afecta a los resultados de los

estudiantes. Sin embargo, la adaptación de acuerdo al estilo de aprendizaje mejora mucho la eficiencia de aprendizaje en términos de tiempo de aprendizaje (con un ES de 4,91 cuando se compara el grupo experimental con el grupo de control 1).

La tecnología hipermedia adaptativa también proporciona mejores resultados cuando se combina con las clases tradicionales (F. Muñoz y Ortigosa 2006). Los resultados de este estudio muestran que los estudiantes que mejoran más fueron aquellos que usaron el sistema de aprendizaje para reforzar los contenidos ya estudiados.

En (Albacete y K.A. VanLehn 2000) se muestra otro interesante resultado. Los autores examinan el efecto del aprendizaje adaptativo en función del aprendizaje previo y encuentran que los estudiantes con menos conocimientos previos son quienes mejoran más. Sin embargo, (Own 2006) encuentra que en un entorno de aprendizaje adaptativo, aunque todos los estudiantes progresan, la diferencia sólo es significativa para los estudiantes con más conocimientos previos.

Referencia	Tamaño del efecto
(G. Hwang 2003)	1
(Lanzilotti y Roselli 2007)	0,1
(Kavcic 2004)	0,97 – 1,3
(Guzman, Conejo, y Perez-de-la-Cruz 2007)	0,93
(Tai, T.-A. Tsai, y F.M.-C. Chen 2001)	0,82
(Nirmalakhandan 2007)	3,86
(Lilley, Barker, y Britton 2004)	0,78 – 1,14
(F. Muñoz y Ortigosa 2006)	0,95
(J Tseng et al. 2008)	0,76 – 0,81
(Own 2006)	0,64
(Yacef 2005)	0,66 – 1,05
(Suraweera y Mitrovic 2004)	0,15
(K. VanLehn et al. 2005)	0,25 – 0,61
(Koedinger et al. 1997)	0,3 – 1,2
(G.-L. Hwang 2007)	1,45
(Mayo y Mitrovic 2001)	0,557
(Albacete y K.A. VanLehn 2000)	0,63
(Jong et al. 2006)	0,57

Tabla 3. Tamaño del efecto (ES) de la mejora en los resultados académicos

### 3.1.3 Opinión de los estudiantes

En este apartado, se va a analizar cuál es la opinión de los estudiantes. En concreto, se va a responder a la siguiente pregunta: *¿Están los estudiantes realmente satisfechos cuando los sistemas de e-learning se adaptan a su perfil, preferencias y/o rendimiento?*

Varios estudios (Bures, Abrami, y Amundsen 2000) (Levy 2007) (Kuh 2001) sugieren que la satisfacción de los estudiantes es un factor importante a la hora de medir el éxito o efectividad del proceso de e-learning y que está asociado con el rendimiento de los estudiantes.

En la Tabla 4 se muestra el grado de satisfacción de los estudiantes con relación a diferentes sistemas educativos adaptativos de entre los estudiados anteriormente. En todos los casos, se ha medido el nivel de satisfacción de los estudiantes preguntándoles explícitamente su opinión a través de una encuesta. Dado que estos cuestionarios están basados en diferentes escalas, los resultados han sido normalizados con el fin de poder examinarlos y compararlos (ver Tabla 4).

Los resultados muestran que la mayoría de los estudiantes opinan que los sistemas educativos adaptativos son buenos para el aprendizaje y satisfacen sus requerimientos. Estudiantes en diferentes entornos de aprendizaje adaptativo muestran un nivel medio de satisfacción próximo a 0,7, excepto para el sistema aLFanet (Fuentes et al. 2005), cuyos resultados muestran un nivel de satisfacción de 0,5 (es decir, medio). En cualquier caso, éste es un resultado global, ya que el sistema aLFanet ha sido usado en cuatro universidades obteniéndose diferentes resultados: desde 0,4 en OUNL (*Open University of the Netherlands*) hasta 0,66 en la UNED (Universidad Nacional de Educación a Distancia). Para analizar estos resultados tan diversos es interesante señalar que las actividades de evaluación del sistema fueron diferentes para cada universidad y que la UNED fue la responsable de la fase de uso y, por tanto, donde más adaptación fue incluida y donde más encuestas se rellenaron (25 de un total de 52 estudiantes).

Por otra parte, (Hatzilygeroudis, Giannoulis, y Koutsojannis 2005) proporcionan una evaluación del sistema con relación a su versión previa no adaptativa. Los estudiantes usaron las dos versiones del sistema y rellenaron una encuesta, la cual incluía preguntas para evaluar su uso y aprendizaje. Los resultados mostraron una mayor preferencia hacia la versión adaptativa (de aproximadamente del 15%), aunque esta mejora no fue significativa (ES de 0,24).

Referencia	Satisfacción
(Y.-H. Chang, T.-Y. Lu, y Fang 2007)	0,76
(Conejo et al. 2004)	0,79
(C.-M. Chen, H.-M. Lee, y Y.-H. Chen 2005)	0,69
(G. Chen, C. Chang, y C. Wang 2008)	0,76
(Hatzilygeroudis, Giannoulis, y Koutsojannis 2005)	0,78
(Own 2006)	0,81
(Conlan 2005)	0,74
(Fuentes et al. 2005)	0,50
(Suraweera y Mitrovic 2004)	0,66
(K. VanLehn et al. 2005)	0,66
(Wei, L.-Y. Lee, y G.-D. Chen 2004)	0,80

Tabla 4. Valor normalizado del nivel de satisfacción de los estudiantes

## 4 El sistema QUESTOURnament y su evolución hacia la adaptación

El sistema QUESTOURnament es una herramienta telemática para el aprendizaje competitivo que se apoya principalmente en la competitividad, la colaboración y el reconocimiento social como mecanismos de motivación y busca mantener dicha cualidad en la actividad académica de los alumnos. Este sistema ha sido implementado como un módulo integrado dentro la plataforma Moodle y, por tanto, puede ser utilizado por el profesor como cualquier otra actividad disponible en Moodle.

Con QUESTOURnament las sesiones de trabajo se organizan como un concurso en el que se propone una serie de “desafíos” intelectuales que los alumnos tienen que solucionar en un tiempo límite. El trabajo realizado, una vez validado, es recompensado mediante un mecanismo de retribución variable (cuyo comportamiento se muestra en la Figura 5). La máxima puntuación que se puede obtener por contestar un desafío varía en el tiempo, se incrementa hasta que se recibe la primera respuesta correcta, momento a partir del cual empieza a disminuir. El profesor puede configurar el comportamiento de este mecanismo, estableciendo la puntuación inicial, la puntuación máxima, un tiempo al comienzo en que la puntuación no varía y si la variación de la puntuación es lineal o exponencial.

La tarea de los estudiantes no se limita sólo a responder desafíos sino que también pueden proponer nuevos desafíos y evaluar las respuestas de sus compañeros, y ser recompensados por ello. Todos los desafíos propuestos por alumnos y sus evaluaciones de las respuestas a esos desafíos pueden ser validadas previamente por el profesor, si así lo desea. El profesor podrá establecer si se mantiene el anonimato de los autores de desafíos.

Algunas investigaciones señalan que resulta muy útil para los estudiantes el poder acceder a los trabajos realizados por otros estudiantes y que, además, los alumnos realizan mejores trabajos cuando saben que sus compañeros accederán a los mismos (Hislop 1999). Por ello, cuando vence el tiempo límite, el desafío se cierra y las respuestas de los alumnos se hacen públicas para conocimiento general, manteniendo el anonimato de los autores.

La clasificación por puntos de los alumnos o de los equipos (si el concurso es por equipos) está disponible para ser consultada en cualquier momento. Además, en la pantalla de entrada del módulo QUESTOURnament y en la portada de la plataforma de teleformación, puede aparecer de forma permanente un *ranking* con los nombres y fotografías de los primeros clasificados. Esta estética hace énfasis en el espíritu competitivo del taller, tal y como puede observarse en la Figura 6, que muestra la pantalla principal de un concurso, con la lista de desafíos propuestos, su puntuación actual y las primeras posiciones de la clasificación.

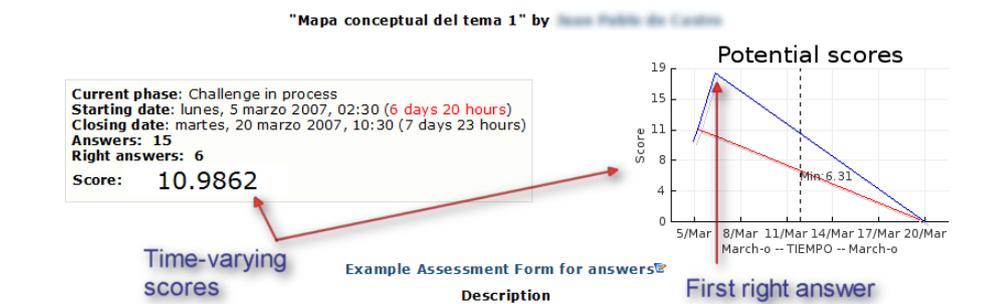


Figura 5. Puntuación variable de un desafío

Teoría de la Comunicación  
Universidad de Valladolid

Teleformación TC07 QUESTS Actividades BONIFICABLES de Teoría de la Comunicación [Update this QUEST](#)

**Actividades BONIFICABLES de Teoría de la Comunicación**

Current phase of Quest: Open QUEST  
 Starting date: Jueves, 1 marzo 2007, 01:00 (11 days 17 hours)  
 Closing date: Sábado, 30 junio 2007, 11:00 (110 days 3 hours)  
 Max right answers by challenge: 10  
 Max. members in team: 3

[Example Assessment Form for challenges](#)

**Description**

Este año se utilizará este sistema de realización de actividades en la asignatura. Para una completa información sobre el funcionamiento del sistema recurre a: [Descripción del sistema](#).  
 Las reglas son las siguientes:

- El profesor propone algunas preguntas que tienen una recompensa en puntos que depende de la dificultad.
- Todos los alumnos pueden proponer Temas** al resto y **obtienen el doble de puntos por ello**. Por supuesto, deben conocer bien la respuesta ya que serán los responsables de evaluar las respuestas que se reciban. (Es más provechoso proponer una buena pregunta para tus compañeros que responder más de una pregunta.)
- Cada Tema es aprobado y precalificado por un profesor. Importante ver previamente el [formulario de calificación](#).
- Durante el plazo abierto para cada pregunta la calificación de recompensa varía en el tiempo.
- La puntuación que se obtiene puede ir desde el máximo al 50% de la recompensa ofrecida según unos criterios de corrección que todos conoceréis de antemano. (Ver [formulario de calificación](#).)
- Se pueden formar equipos (sólo tenéis que decidir un nombre que os represente). Parte de la puntuación de vuestros compañeros se sumará a la puntuación vuestra, pero no se consienten plagios ni enviar el mismo trabajo varias veces. Colaborad, trabajad en equipo pero no permitais parásitos ni falsos favores.

Recordad que el primer clasificado (tomando como mínimo que haya respondido correctamente al menos la mitad de las preguntas planteadas) obtendrá un **40% de bonificación** en su calificación de la asignatura y el resto de participantes la parte proporcional a su puntuación.

NO INCLUYAIS VUESTRO NOMBRE EN EL TÍTULO. EL SISTEMA REGISTRA EL AUTOR SIN RIESGO DE QUE SE PIERDA.  
 Para más información recurre a: [Descripción del sistema](#).

**Ranking**  
 View Summary by Teams

User	Score
USER PROFILE	13.9805
USER PROFILE	13.0689
USER PROFILE	12.3143
USER PROFILE	11.4412
USER PROFILE	10.6714

[View Ranking](#)

**My Place / Manage Teams**

**Add challenge**

Title	Firstname / Lastname ↓	Phase	Answers(Right)[Not assessed]	Starting on	Closing on	Score
Mapa conceptual del tema 1	Usuario Ficticio de Usuario	Challenge in process	15 (7) [2]	03/5/07 14:30	03/ 20/07 10:30	10.5439
Cuestiones de Teoría de la comunicación	Usuario Ficticio de Usuario	Challenge in process	6 (0) [6]	03/7/07 20:05	03/ 21/07 20:05	41.7171
SOLITONES. Trabajo de investigación.	Usuario Ficticio de Usuario	Start pending	0 (0) [0]	03/ 15/07 01:00	04/6/07 02:00	25.0000

A (B) [C] ✓  
 A: Answers, B: Correct Answers.  
 C: Not assessed Answers. ✓ Shows if I have answered yet

Designed By MoodleThemes moodle

You are logged in as Usuario Ficticio de Usuario (Logout)

Figura 6: Pantalla principal de un concurso desarrollado con QUESTOURnament

QUESTOURnament ha sido utilizado en asignaturas de distintas titulaciones universitarias, tanto de ciencias como de letras, con buenos resultados (L.M. Regueras, E. Verdú, M.J. Verdú, M.A. Pérez, et al. 2007) (L.M. Regueras et al. 2008) (L.M. Regueras et al. 2009). Como se comprobó en el estudio presentado en (L.M. Regueras et al. 2009) el sistema permite mejorar los resultados de aprendizaje de los alumnos. Además, los estudiantes que usan QUESTOURnament indican que les gusta el sistema así como participar en concursos y que les gustaría utilizar la herramienta en otras asignaturas (L.M. Regueras, E. Verdú, M.J. Verdú, M.A. Pérez, et al. 2007). El uso de QUESTOURnament contribuye a que los estudiantes lleven al día las asignaturas y mejoren así sus hábitos de estudio (E. Verdú et al. 2007). Además, de acuerdo al estudio descrito en (L.M. Regueras et al. 2008) el nivel de satisfacción con la herramienta no depende del estilo de aprendizaje de los alumnos (participativo, colaborativo y competitivo) de manera significativa, ya que a los alumnos no-competitivos también les gusta aprender con QUESTOURnament (E. Verdú et al. 2007), lo cual es un factor muy positivo.

Sin embargo, se han detectado algunos aspectos negativos del uso del sistema. Por un lado, se trata de una herramienta telemática que puede ser utilizada desde cualquier lugar y en cualquier momento si se tiene acceso a Internet, lo cual aunque en principio es una ventaja, en algunos contextos puede generar injusticias por el hecho de tener más o menos fácil el acceso a Internet (E. Verdú et al. 2007). Por otro lado, debido a la naturaleza competitiva del sistema, su uso causa que algunos alumnos sientan demasiado estrés y dependencia por la evolución del concurso (E. Verdú et al. 2006) (L.M. Regueras, E. Verdú, M.J. Verdú, M.A. Pérez, et al. 2007). Además, aunque, como ya se ha mencionado, el sistema aumenta la motivación, ésta no es la misma para los que están en los primeros puestos de la tabla de clasificación que para los que están en los últimos.

Respecto al primer problema, debido al carácter telemático de QUESTOURnament, si los concursos se realizan durante laboratorios presenciales con un número suficiente de equipos informáticos el problema desaparece. Asimismo, el mecanismo de puntuación variable con el tiempo puede ser configurado por el profesor de tal manera que el tiempo pueda ser un factor no relevante en el concurso. El segundo y el tercer efectos negativos, relativos a la naturaleza inherentemente competitiva de QUESTOURnament, se pueden mitigar diseñando un sistema QUESTOURnament adaptativo que se ajuste a las características y requerimientos de cada estudiante. Sin embargo, a pesar de las ventajas que los sistemas adaptativos proveen, tal y como se ha mostrado en el apartado 3 de este capítulo, es difícil encontrar en la literatura sistemas con el doble carácter competitivo y adaptativo.

Muchos de estos sistemas de aprendizaje competitivo y adaptativo consisten en juegos. Los juegos de ordenador normalmente tienen distintos niveles y, según el usuario practica y obtiene mejores resultados en el juego, el sistema le va proponiendo distintos niveles de dificultad del juego. Esta misma idea es la que se aplica en los sistemas de aprendizaje adaptativo basados en juegos. Por ejemplo, el juego multimedia adaptativo denominado Siege (S. Rodríguez, I. Cheng, y Basu 2007), que permite a los alumnos aprender la física relacionada con el movimiento de un proyectil, presenta distintos episodios de un determinado nivel de dificultad de acuerdo a la actuación del estudiante.

Cuando los alumnos compiten entre sí, se les puede agrupar para que alumnos con similares capacidades compitan entre sí y de esta forma todos los alumnos tengan posibilidades de ganar. Esta es la estrategia que, por ejemplo, se emplea en el juego de tablero *AnswerMatching* (W.M.C. Wu et al. 2007), basado en el funcionamiento del bingo, donde además los alumnos con menor capacidad compiten con menos oponentes.

Otras estrategias que también buscan disminuir el estrés que causa la competición se basan en agentes virtuales acompañantes en el aprendizaje (Chou, Chanb, y C.-J. Lin 2003). Así por ejemplo, existe un sistema denominado *My-Pet* (Z.-H. Chen y T.-W. Chan 2008) en el que, durante una primera fase, cada estudiante tiene que alimentar a su animal de compañía, que es un actor virtual y, en una segunda fase, estos animales de compañía compiten entre sí, evitando de esta manera la competición directa entre estudiantes.

Como se ha mencionado anteriormente, no se encuentra en la literatura muchos ejemplos de sistemas competitivos y adaptativos pero sí que se puede analizar si las funcionalidades de los sistemas adaptativos e ITS presentados en el apartado 3 pueden ser interesantes para el sistema QUESTOURnament adaptativo.

Los sistemas *Curriculum Sequencing* suelen trabajar sobre mapas de conceptos definidos por un experto que contienen los conceptos que se va a aprender y las relaciones entre estos conceptos. Cada nodo del mapa representa un concepto y tiene asociados contenidos y otras actividades, como evaluaciones que permitirán medir el conocimiento por parte del alumno de este concepto. A partir de los conceptos ya conocidos por el estudiante, el sistema le propone el

itinerario adecuado de aprendizaje de los conceptos todavía no aprendidos. La limitación de estos sistemas está en que requieren una definición de un dominio de conocimiento y existen sistemas como QUESTOURnament que son independientes del dominio de conocimiento, esto es, sirven para el aprendizaje de cualquier temática. Para solventar esta limitación, algunos autores proponen herramientas para que el mismo profesor introduzca en el sistema la información del dominio de conocimiento de la materia que imparte. Sin embargo esta tarea es difícil y requiere bastante tiempo por parte del profesor. Por este motivo, (S. Tseng et al. 2007) propone una solución para crear mapas de conceptos a partir de registros históricos de los estudiantes de manera automática. Aunque este aspecto es sin duda, interesante, no se considera un eje fundamental para disminuir los inconvenientes del aprendizaje competitivo presentados en QUESTOURnament. Una estrategia adaptativa de este tipo debería implicar no sólo a QUESTOURnament sino también a muchas otras actividades y contenidos manejados en Moodle.

Los contenidos que QUESTOURnament presenta al estudiante son preguntas y problemas de muy diferente naturaleza y diferentes temáticas. Además son contenidos que no están predefinidos, sino que el profesor los introduce en el sistema para cada concurso particular. Por este motivo, el QUESTOURnament adaptativo se torna principalmente como un sistema adaptativo del tipo *Curriculum Sequencing* independiente del dominio de conocimiento y focalizado en la evaluación. El QUESTOURnament adaptativo deberá ser capaz de proponer desafíos a los alumnos adaptados a su nivel de conocimiento. Así mejorará la eficacia del aprendizaje a través de QUESTOURnament si se obtienen buenos resultados, tal y como ha sucedido en experimentos con sistemas que facilitan este tipo de evaluación, como los descritos en (Guzman, Conejo, y Perez-de-la-Cruz 2007) (Tai, T.-A. Tsai, y F.M.-C. Chen 2001). Por otro lado, el grado de competencia percibido por el alumno que trata de resolver desafíos adaptados a su nivel aumenta y, por tanto, su motivación aumenta también. Si se propone a los estudiantes preguntas de distintas dificultades, se debe habilitar un mecanismo de puntuación que tenga en cuenta los distintos niveles de dificultad, para dar mayor valoración a aquellas cuestiones resueltas más difíciles y así hacer que la competición sea más justa. Sin embargo, una solución de este tipo no permitirá que disminuya el estrés en los alumnos que se encuentran en las últimas posiciones de la tabla de clasificación. Por ello, como en el sistema *AnswerMatching* (W.M.C. Wu et al. 2007), se propone agrupar a los alumnos con niveles de conocimiento similares para que compitan entre sí. De este modo, todos los alumnos pueden percibir que son competitivos en su concurso y la competición se mantiene justa, aumentando la motivación y disminuyendo el estrés.

Siendo los contenidos en QUESTOURnament preguntas y problemas, también cabe proponer un mecanismo de soporte para la resolución de problemas o para el análisis de las soluciones propuestas por los estudiantes, así como para su evaluación automática. Este tipo de sistemas (*Intelligent Solution Analysis* o *Problem Solving Support*) suele ser dependiente del dominio de conocimiento que se está tratando, siendo complicado un diseño general para todo tipo de preguntas y problemas. Aún así, se puede proponer soluciones sencillas que permitan a los profesores, por ejemplo, introducir pistas junto al enunciado de las preguntas y emplear sistemas de evaluación externos para la corrección automática o semiautomática de los diferentes tipos de preguntas propuestas en QUESTOURnament, tal y como permite el sistema EduJudge descrito en (de Castro et al., 2010).

Por último, QUESTOURnament permite llevar a cabo concursos por equipos. Soluciones como las propuestas en (G. Chen, C. Chang, y C. Wang 2008)(Wei, L.-Y. Lee, y G.-D. Chen 2004), que permiten encontrar compañeros mentores, pueden ser la base para agrupar alumnos cuyos conocimientos se complementen y así formar equipos competitivos en

QUESTOURnament. El objetivo de formar este tipo de equipos sería no sólo lograr su competitividad sino también el que los alumnos dentro de su equipo aprendan de sus compañeros. Sin embargo, esta estrategia requiere de la definición de dominios de conocimiento, que inicialmente no se contempla en QUESTOURnament, como ya se ha indicado. Además, al igual que sucede con las estrategias competitivas, las estrategias de aprendizaje colaborativo deben ser también diseñadas cuidadosamente y supervisadas por el profesor, ya que el aprendizaje colaborativo presenta igualmente sus desventajas, como el posible desequilibrio en la carga de trabajo entre los miembros de un equipo o las dificultades que un alumno con menor conocimiento puede encontrar para seguir las explicaciones de sus compañeros y comprender el trabajo realizado (Verdu y Lorenzo 2010).



# **CAPÍTULO 3**

## **PROPUESTA DEL SISTEMA QUESTOURNAMENT ADAPTATIVO**

### **1 Definición de requisitos**

QUESTOURnament registra datos relativos a los resultados y participación de los usuarios, lo que permite tener información relacionada con el trabajo personal de cada alumno (por ejemplo, calificaciones obtenidas) así como información relativa a las interacciones con el sistema (momento en que un alumno accede a un desafío, momento en que responde a un desafío, etc.), datos útiles que permitirán al sistema QUESTOURnament adaptarse a las necesidades de los estudiantes.

Esta adaptación del sistema QUESTOURnament a las necesidades y particularidades de cada alumno conllevará varias estrategias (E. Verdú, L. M. Regueras, M. J. Verdú, et al. 2008b):

- Estimación objetiva del nivel de dificultad de los desafíos propuestos en QUESTOURnament. Dado que para determinar el nivel de conocimiento de los alumnos se utilizará información sobre las respuestas que éstos dan a los desafíos, es fundamental que previamente se haya estimado correctamente el nivel de dificultad de estos desafíos.

- Estimación del nivel de conocimiento y habilidades adquiridas por los estudiantes, así como su progreso durante el concurso, de acuerdo a sus respuestas a los diferentes desafíos y otros resultados en distintas actividades de Moodle.
- Establecimiento de caminos de aprendizaje basados en el progreso del estudiante. Para ello, los estudiantes con similar nivel de conocimiento serán agrupados para que compitan entre sí con el fin de que todos los estudiantes tengan posibilidades de ganar dentro de su grupo y, así, se mantenga alta su motivación y la situación de aprendizaje sea constructiva (D.P. Johnson y R.T. Johnson 1999). Esto además dará como resultado una disminución de la posibilidad de abandono, la cual suele aumentar cuando los alumnos menos aventajados quedan demasiado lejos de las primeras posiciones de la clasificación (K. Chu, M. Chang, y Hsia 2004). Asimismo, el estrés de la competición puede verse disminuido al competir con alumnos del mismo nivel. A cada grupo de alumnos se les presentará un conjunto de preguntas adecuadas a su nivel de conocimiento. Al margen de la competición, para mantener alta la motivación de los estudiantes, es importante proporcionarles preguntas adecuadas a su nivel, ni muy difíciles ni muy fáciles (Kunichika et al. 2002). Presentar preguntas muy difíciles puede hacer que los estudiantes abandonen el intento de resolver los desafíos mientras que responder preguntas fáciles puede dar lugar a distracción, aburrimiento y falta de interés (Vogel et al. 2006).
- Retroalimentación personalizada, pistas y mensajes *ad hoc*, que se muestran a los estudiantes con el fin de ayudarles a responder una pregunta correctamente y progresar así en la competición.

El sistema QUESTOURnament adaptativo debe adaptarse a la capacidad y progreso de los distintos estudiantes pero, teniendo en cuenta el dinamismo subyacente en el sistema QUESTOURnament de partida, es necesario implementar una adaptación progresiva, para evitar un sistema excesivamente dinámico y cambiante que pueda generar confusión en los alumnos. Esta posible confusión podría dar lugar a una agravación del estrés, factor que precisamente se pretende mitigar. Así, en una primera aproximación, la adaptación consistirá en los siguientes dos puntos principales:

- Agrupar a los alumnos de acuerdo a su nivel de conocimiento para que alumnos con similar nivel de conocimiento compitan directamente entre sí dentro de una *liga*.
- Para cada *liga* los desafíos propuestos deberán ser adecuados al nivel de conocimiento de los alumnos que participan en la misma.

Un profesor, al crear un concurso dentro del sistema QUESTOURnament, podrá indicar si desea que se desarrolle como una única competición con un único grupo de alumnos o con distintos grupos de alumnos mediante un sistema de ligas.

Por otro lado, el proceso adaptativo debe ser independiente del dominio de conocimiento; ya que el sistema QUESTOURnament se puede aplicar a cualquier área y dominio de conocimiento, tanto de letras como de ciencias.

Por último, un requisito importante del sistema es que sea modular. Por ello, el módulo QUESTOURnament adaptativo debe ser independiente del módulo que agrupa a los estudiantes, lo que permitirá además reutilizar este último sistema con otros módulos diferentes del QUESTOURnament. Así se puede distinguir dos sistemas: el *clusterer* o “agrupador” y el propio sistema QUESTOURnament Adaptativo.

## 2 Sistema Clusterer

El sistema *Clusterer* permitirá crear *clusters* para el agrupamiento de los alumnos en base a sus características. El sistema es lo suficientemente genérico como para permitir métodos diversos de clasificación, así como distintas posibilidades para los datos de entrada.

Durante el desarrollo de un curso con Moodle, el profesor puede pedir al sistema la creación de grupos o *clusters*, especificando el conjunto de actividades que desea utilizar como datos de entrada para realizar la clasificación de alumnos, así como el método de clasificación a utilizar y el número de grupos que desea. Como datos de entrada para estimar el nivel de conocimiento de los alumnos y la correspondiente agrupación se pueden seleccionar todos aquellos datos de calificación disponibles en Moodle: notas de las tareas, resultados de cuestionarios, puntuaciones de QUESTOURnament, etc. Los métodos de clasificación se implementan como *plugins* que pueden añadirse fácilmente al sistema, según se vayan implementando nuevos métodos. Actualmente se dispone de los siguientes métodos: aleatorio, manual, estadístico (con cuatro opciones: división por intervalos equitativos, por cuantiles, divisiones naturales y medias anidadas) y por ligas. El método de clasificación por ligas permite al profesor simular un campeonato con varias fases y tres ligas simultáneas (avanzada, intermedia y básica). En cada fase los alumnos son asignados a una liga, en función de los resultados de la fase anterior, de forma que algunos alumnos pueden ascender o descender de nivel, en función de sus resultados.

Cada combinación ‘método-fuente de datos’ generará un conjunto de *clusters* o grupos de alumnos identificable, que podrá ser empleado por el módulo QUESTOURnament adaptativo para la creación de ligas (una liga por cada grupo generado).

Aunque la finalidad principal de este módulo es agrupar alumnos en base a su nivel de conocimiento, podrá ser también usado para agrupar alumnos en base a otras características de su perfil como, por ejemplo, hobbies del alumno. Así, cada agrupador o *clusterer* llevará asociada una etiqueta que define la propiedad tenida en cuenta para la agrupación. Dentro de un *cluster* se podrá indicar el grado de pertenencia del alumno al *cluster*, de forma que se proporciona información adicional a lo que sería una clasificación binaria, según la cual un alumno pertenece o no pertenece a un *cluster*. Añadiendo el grado de pertenencia, el profesor, o cualquier módulo del sistema que haga uso de los *clusters* generados, cuenta con mayor información sobre la clasificación realizada y puede saber, por ejemplo, si un alumno es representativo del grupo en el que está o por el contrario está cerca de alguno de los otros niveles. Más adelante se detalla la utilidad de esta información adicional dentro del sistema de ligas del QUESTOURnament adaptativo.

En la Figura 7 se ilustra el funcionamiento del *clusterer* o agrupador. Cada *clusterer* indica el método, la fuente de datos y la propiedad empleada para obtener las agrupaciones de alumnos a partir del estado actual del curso. En el ejemplo de la figura se emplea como entrada la nota obtenida en la actividad cuestionario o *quiz* de Moodle y el método para el agrupamiento es el estadístico por intervalos equitativos, explicado posteriormente. Se obtienen tres clusters a los que se le asigna un valor relativo a la propiedad *Nivel* (*básico*, *medio* o *avanzado*). Los agrupamientos siguen un modelo difuso, con funciones de pertenencia definidas para cada cluster, y cada miembro del cluster tiene un valor de grado de pertenencia a ese cluster, tal y como se ha explicado anteriormente. Así, las agrupaciones están compuestas de diversos usuarios etiquetados con un valor relativo a la variable (propiedad) que se está modelando y que se puede contrastar con la función de pertenencia.

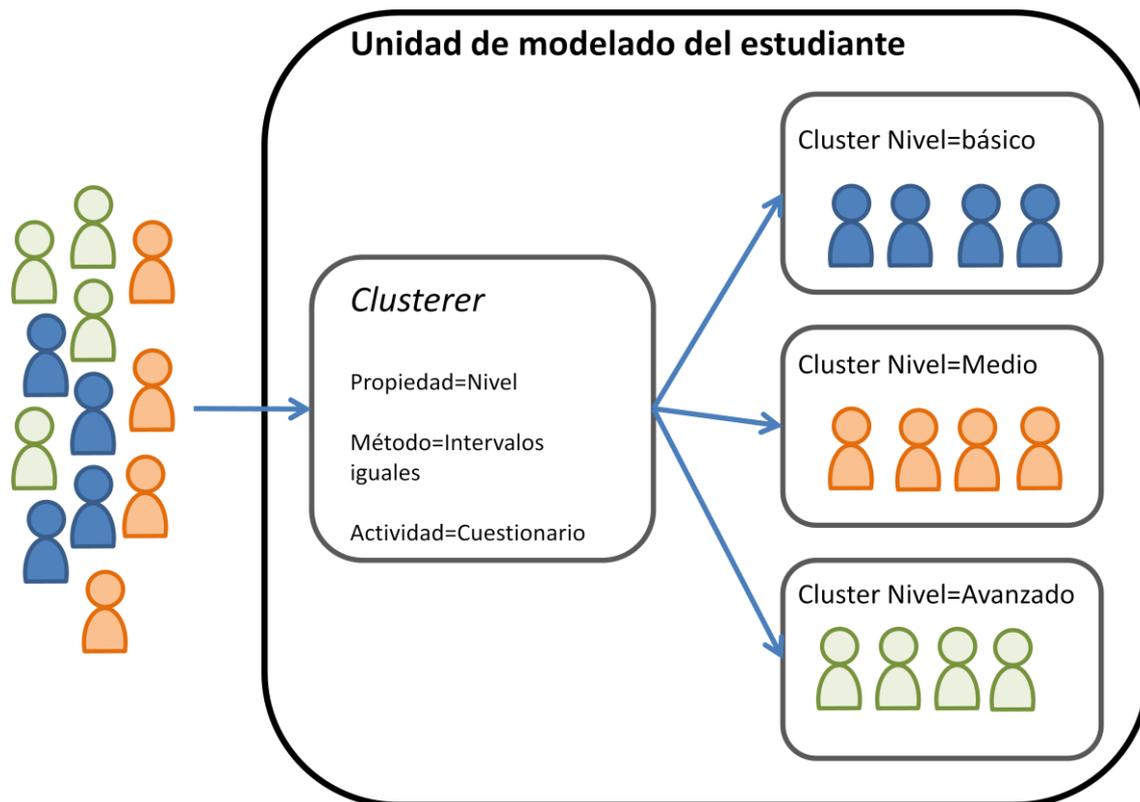


Figura 7. Esquema del *clusterer*

Se describen a continuación con más detalle los métodos de *agrupación* que el *clusterer* puede emplear para generar los grupos:

- Agrupamiento manual: el profesor asigna manualmente usuarios a los grupos.
- Agrupamiento aleatorio: se crean los grupos de manera aleatoria.
- Agrupamiento por ligas: permite modificar los grupos que compiten en el QUESTOURnament en base a los resultados de la competición. Existe un número mínimo de usuarios por *cluster* y a cada usuario se le asigna un grado de pertenencia al grupo según su resultado en la competición.

Este número mínimo de usuarios se define para evitar la creación de grupos demasiado pequeños. Cuando el número de usuarios resultante en un agrupamiento es menor que este mínimo establecido, sus miembros se combinan con los agrupamientos adyacentes. A través del valor de grado de pertenencia al grupo, se puede conocer si un miembro de ese grupo es original del grupo o viene de un grupo con incidencia (grupo demasiado pequeño). Con esta información se puede deducir si existe suficiente diferencia en el nivel de conocimiento dentro del grupo resultante como para incluir preguntas del nivel del grupo con incidencia que desapareció.

- Plugin de agrupamiento estadístico: a partir de los resultados de una actividad de Moodle (QUESTOURnament, cuestionario,...) o de la calificación global de los alumnos disponible en el *Gradebook* o Libro de Calificaciones de Moodle (esta sería la opción por defecto), según seleccione el profesor, se crean los grupos aplicando uno de los métodos estadísticos disponibles:
  - o Cuantiles: en función de la calificación de los alumnos éstos se dividen en varios grupos (tantos como el número de clusters) con el mismo número de

- elementos cada uno. De esta forma, se garantiza que todos los agrupamientos contienen el mismo número de alumnos.
- Intervalos equitativos: se divide la nota o puntuación del usuario que mayor puntuación ha obtenido entre el número de clusters solicitados y se obtiene distintos intervalos de notas, que definen cada agrupamiento.
  - Divisiones naturales: se ordenan las calificaciones de los alumnos de menor a mayor y se crean los agrupamientos separando las notas en función de donde haya un mayor hueco o espaciado de los datos.
  - Medias anidadas: se obtienen las fronteras de los grupos mediante el empleo de la media aritmética. Para ello, se ordenan las calificaciones de mayor a menor, y se calcula la media, generando así dos intervalos (las calificaciones que están por encima y las que están por debajo); este mismo procedimiento se aplica a los nuevos intervalos sucesivamente hasta establecer tantos grupos como número de *clusters*.
  - Divisiones naturales de Jenk: es una optimización de las divisiones naturales, la cual intenta crear grupos reduciendo la varianza dentro de cada grupo y maximizando la varianza entre grupos.
- Agrupamiento mediante algoritmo genético: este método parte de un conjunto inicial de grupos y, en función de los resultados en diferentes actividades de Moodle, reajusta los grupos cuando sea conveniente. Aquellos alumnos que no encajen dentro de su perfil de grupo, serán desplazados a otros grupos. La diferencia de este método con los anteriores es que tomará como entrada los resultados de varias actividades y no sólo de una actividad (o no sólo de la calificación global especificada en el *Gradebook*).
  - Agrupamiento mediante un algoritmo de *clustering*. Un algoritmo de *clustering* permite obtener una clasificación no supervisada (sin clases definidas), formando grupos que tengan características similares. Por ejemplo, (C. Romero, S. Ventura, y Garcia 2008) emplean el algoritmo de clustering *k-means* para agrupar estudiantes en función de las actividades realizadas en Moodle. El algoritmo puede, por ejemplo, agrupar en un mismo cluster a los estudiantes que tienen más de un mensaje enviado a un foro, 3 mensajes leídos y un alto número de cuestionarios contestados y superados. Según los autores, esta información puede ser empleada por el profesor para posteriormente crear equipos de alumnos que desempeñen actividades de trabajo colaborativo con características complementarias, por ejemplo.

En todos estos casos se podrá elegir el número de *clusters* que se quiere formar.

Todos los *clusters* o agrupamientos generados son almacenados en el sistema, que mantiene un histórico de estos y sus relaciones (ya que hay *clusters* que se generan a partir de otros *clusters*) y permite la generación de versiones de estos agrupamientos.

### 3 Sistema QUESTOURnament adaptativo

El sistema propuesto permite el desarrollo de competiciones por ligas a partir de los grupos creados por el módulo *clusterer*. Para cada liga el sistema automáticamente presenta preguntas adecuadas al nivel de conocimiento de los alumnos participantes en la misma (en posteriores diseños de QUESTOURnament adaptativo se podrían tener en cuenta otras propiedades, como

se ha explicado anteriormente), aunque el profesor también puede asignar las preguntas manualmente.

La Figura 8 muestra un esquema del sistema, cuyos componentes son:

- Sistema QUESTOURnament integrado en Moodle: permite la realización de concursos, incluye la interfaz con estudiantes y profesores, los módulos de aplicación, así como la base de datos que incluye los resultados de los estudiantes en las distintas actividades y sus interacciones con el sistema.
- Repositorio de desafíos clasificados por nivel de dificultad (*fácil, asequible y difícil*). Inicialmente el nivel de dificultad asociado a un desafío es el que estima el profesor al crearlo.
- Sistema Experto Genético Difuso: una vez que se disponga de respuestas a los desafíos por parte de los alumnos, el profesor puede invocar a este sistema, descrito en el siguiente capítulo, para que ajuste el nivel de dificultad de las preguntas al comportamiento real de los alumnos al participar en un concurso con QUESTOURnament.
- Unidad de Modelado del Estudiante: consiste en el *clusterer* que agrupa a los estudiantes en función de un método de agrupamiento y de los resultados de actividades (Quiz, QUESTOURnament o el resultado global de varias actividades almacenado en el *GradeBook*, por ejemplo). QUESTOURnament emplea grupos o *clusters* creados en la Unidad de Modelado del Estudiante para formar ligas o competiciones paralelas en las que los miembros de un *cluster* compiten entre sí. Como se ha comentado, este sistema es un módulo extraíble o independiente del QUESTOURnament adaptativo, con el objetivo de que pueda ser utilizado por otras actividades de Moodle.
- Unidad pedagógica: se encarga de proporcionar los caminos de aprendizaje adecuados en los concursos. En definitiva, presenta en cada liga las preguntas adecuadas al grupo de estudiantes que participa en las mismas.

Esta primera versión del QUESTOURnament adaptativo, que pretende un funcionamiento simple y poco confuso, emplea tres niveles de dificultad (*fácil, asequible y difícil*) y se apoya en el módulo *clusterer* como se explica a continuación.

Si al crear la actividad, el profesor indica que el QUESTOURnament se desarrolle por ligas, el sistema le pedirá que seleccione los grupos de estudiantes, previamente generados con el *clusterer*, que conformarán las diferentes ligas. También para simplificar el proceso, se ha establecido que el sistema trabaje con tres ligas (*nivel básico, intermedio y avanzado*), aunque el diseño del mismo permite trabajar en el futuro con más niveles de conocimiento y más niveles de dificultad de los desafíos. En el caso de que no se creen tres *clusters* diferentes porque los alumnos tengan un comportamiento similar o porque algunos de los grupos sean muy poco numerosos, se establecerán menos ligas.

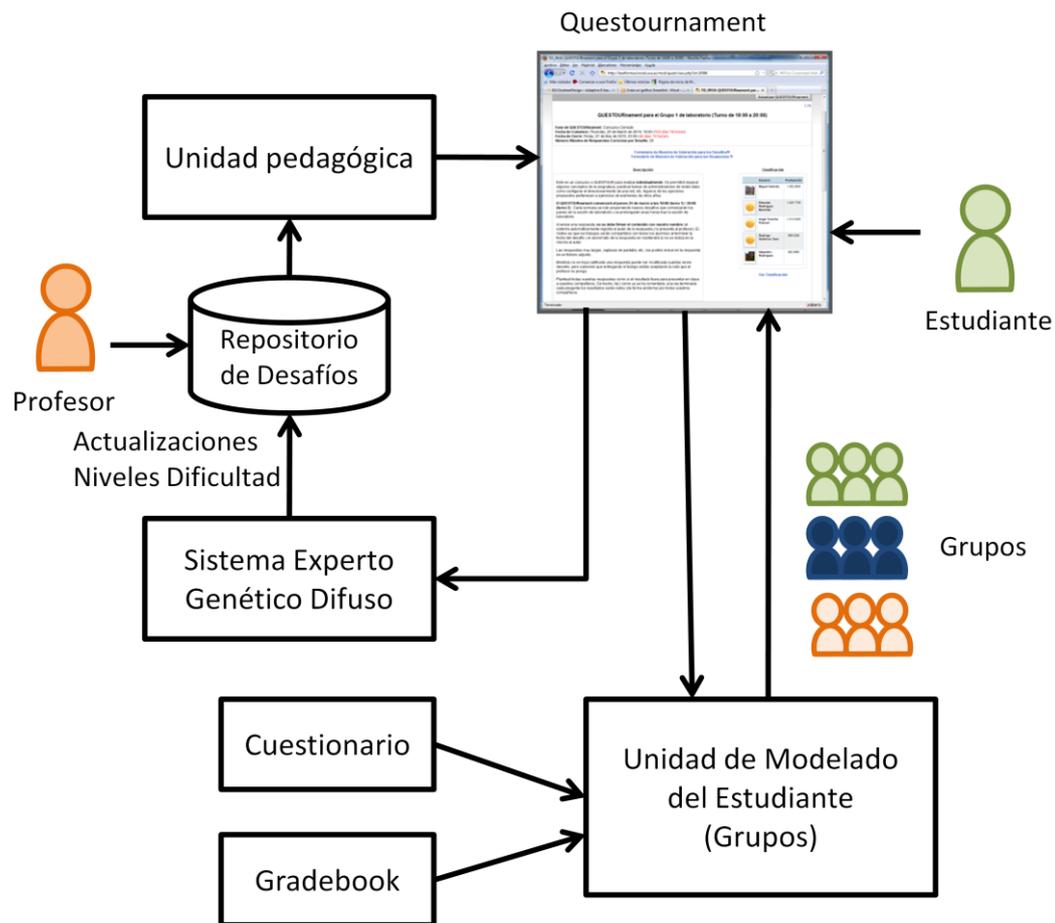


Figura 8. Sistema QUESTOURnament Adaptativo

A cada liga le corresponde un nivel de conocimiento (*básico, intermedio y avanzado*) y el profesor puede introducir los desafíos de cada liga manualmente o solicitar al sistema que asigne desafíos en cada liga, con el nivel de dificultad adecuado al grupo de alumnos correspondiente. Esto no implica que en una liga se asignen desafíos de un único nivel de dificultad. De hecho, el profesor puede configurar qué porcentaje de los desafíos que se presentan en cada liga pueden ser de un nivel de dificultad distinto del correspondiente a esa liga. Además, como se ha explicado anteriormente, puede haber grupos formados por miembros cuyo valor de la propiedad no es el que corresponde al grupo. Esto sucede cuando uno de los grupos que resultan de la aplicación del *clusterer* es demasiado pequeño, con lo que sus miembros son asignados a otro grupo. El sistema puede detectar este hecho porque el usuario tendrá un valor que indica el grado de pertenencia al cluster y así propondrá preguntas tanto del nivel del cluster correspondiente como de niveles adyacentes.

A continuación, un breve ejemplo ilustra cómo un profesor puede utilizar el QUESTOURnament adaptativo. Suponiendo, por ejemplo, que un curso tiene varios temas y el profesor desea establecer varios concursos con preguntas o desafíos sobre cada tema a lo largo del curso, podría hacerlo estableciendo ligas siguiendo los siguientes pasos (ver Figura 9):

- *Paso 1.* El profesor crea un cuestionario en Moodle con una serie de preguntas para determinar el nivel de conocimiento de los alumnos al inicio del curso.

- *Paso 2.* Una vez que los alumnos contestan al cuestionario creado en el paso 1, el profesor puede invocar al *clusterer* para que genere tres grupos de alumnos, seleccionando como datos de entrada al *clusterer* los resultados del cuestionario del paso 1, y el método estadístico de intervalos iguales, por ejemplo. El profesor puede modificar manualmente los grupos generados. Otra manera de crear los grupos iniciales puede ser la generación manual de los mismos, si se dispone de las notas finales de los alumnos en una asignatura previa relacionada con esta asignatura.
- *Paso 3.* El profesor introduce desafíos en el repositorio y les asigna una dificultad estimada: fácil, asequible o difícil.
- *Paso 4.* El profesor crea una actividad QUESTOURnament y elige la modalidad de concurso por ligas. El sistema le pedirá que seleccione el conjunto de *clusters* que son el punto de partida para la formación de las distintas ligas y el profesor seleccionará el generado en el paso 2.

A continuación, el profesor podrá seleccionar los desafíos a presentar en cada liga o también tendrá la opción de que el propio sistema asigne los desafíos adecuados a cada grupo en función del nivel del grupo y de la dificultad del desafío. Si escoge esta opción automática, el profesor podrá indicar un índice de pureza, para que el sistema no sólo presente desafíos de un nivel de dificultad dado sino que también incluya desafíos de otro nivel de dificultad. Por ejemplo, si se indica un nivel de pureza 100% para los desafíos del grupo de nivel avanzado, el sistema sólo presentará desafíos de nivel difícil a ese grupo. Si el profesor indica un nivel de pureza del 80%, el sistema presentará un 80% de desafíos de nivel difícil y un 20% de desafíos de nivel asequible.

Asimismo, si un grupo está formado por alumnos de distintos niveles (debido a que se han fusionado dos niveles por tener un grupo pequeño de alumnos alguno de ellos) el sistema igualmente presentará desafíos correspondientes a dos niveles de dificultad.

El profesor siempre puede modificar manualmente los desafíos propuestos por el sistema.

- *Paso 5.* Tiene lugar el concurso durante el cual los alumnos compiten dentro de sus ligas y al finalizar el concurso, una tabla de clasificación en cada liga muestra las posiciones de los alumnos dentro de sus ligas.
- *Paso 6.* El profesor puede solicitar al sistema el reajuste del nivel de dificultad de los desafíos con respecto al comportamiento real de los alumnos al resolverlos. Este reajuste lo realiza el Sistema Experto Genético Difuso. El profesor selecciona los desafíos que desea reajustar. Reglas sencillas permiten el reajuste de algunos casos como, por ejemplo, si un desafío clasificado como fácil no es respondido correctamente por un gran número de alumnos en el grupo de nivel medio, ese desafío no es fácil. Igualmente, si un desafío asequible o difícil fuera respondido correctamente por los alumnos del nivel básico, se puede deducir que está mal clasificado. Además, es conveniente tener en cuenta otros datos de interacción de los alumnos con los desafíos como el número de lecturas o el tiempo empleado, ya que, por ejemplo, un desafío asequible podría ser resuelto por gran parte de los alumnos tras una buena inversión de tiempo. Por tanto, el sistema experto genético difuso utiliza también esos datos para realizar la reclasificación. Este proceso de reajuste de los desafíos deberá hacerse paulatinamente y el nivel de dificultad estimado para los mismos se irá perfeccionando según son analizados y reclasificados por el sistema.

No se puede obtener una clasificación óptima de una vez si, por ejemplo, un desafío ha sido contestado únicamente por alumnos de un único nivel de conocimiento. La mejor manera de conseguir buenas reclasificaciones es tener respuestas a los desafíos de alumnos de todos los niveles. Si se establecen índices de pureza menores que 100, los desafíos son respondidos por alumnos de distintos niveles y su reclasificación será más precisa. Cuando se trabaja con niveles de pureza intermedios, el sistema propondrá aquellos desafíos que tienen menos respuestas de alumnos en los distintos niveles para mejorar su posterior reclasificación.

- *Paso 7.* El profesor invoca al módulo *clusterer* que reagrupará a los alumnos a partir de sus resultados en el QUESTOURnament que se ha desarrollado según el paso 5. En función del método que escoja para las agrupaciones, se generarán distintos grupos:
  - Agrupamiento por ligas: se toma el *clusterer* de partida (grupos generados en el paso 2) y los resultados en el QUESTOURnament (para cada liga, las posiciones de sus miembros) y se cambia de categoría a aquellos alumnos que han quedado en las primeras posiciones y/o últimas de su liga (los primeros suben de categoría y los últimos bajan de categoría como en las competiciones deportivas).
  - Agrupamiento mediante un algoritmo genético teniendo en cuenta los resultados de distintas actividades: se toma el *clusterer* de partida (grupos generados en el paso 2) y los resultados de los alumnos en las actividades que el profesor seleccione (por ejemplo, se ha podido llevar a cabo varias actividades simultáneamente al concurso que pueden servir para valorar los resultados globales académicos de los estudiantes) y el sistema proporciona unos nuevos *clusters* recolocando a los alumnos que no encajan en su *cluster* original, porque su rendimiento ha sido diferente al de sus compañeros durante el desarrollo de estas actividades.
  - Agrupamiento mediante un algoritmo genético teniendo en cuenta las respuestas a los distintos desafíos propuestos en el QUESTOURnament desarrollado en el paso 5 y tomando como *clusters* de partida los grupos generados en el paso 2. Si únicamente se ha realizado la actividad QUESTOURnament durante el curso, se puede analizar las respuestas de los estudiantes a los distintos desafíos y cambiar de grupo a aquellos alumnos cuyo comportamiento ante los desafíos haya sido distinto al de su grupo.
- *Paso 8.* El profesor lanza un nuevo concurso por ligas con la nueva distribución de grupos. Este paso sería similar al paso 4, repitiéndose los pasos 4-5-6-7 tantas veces como concursos quiera el profesor desarrollar. El profesor podrá ir solicitando al sistema el reajuste de las dificultades de los desafíos y podrá introducir desafíos nuevos. El reajuste de las dificultades de los desafíos será válido para el grupo de alumnos de ese curso y cualquier otro grupo de alumnos de similares características (la clasificación de los desafíos puede valer de un año académico a otro, por ejemplo, si los antecedentes y formación de los alumnos son similares).

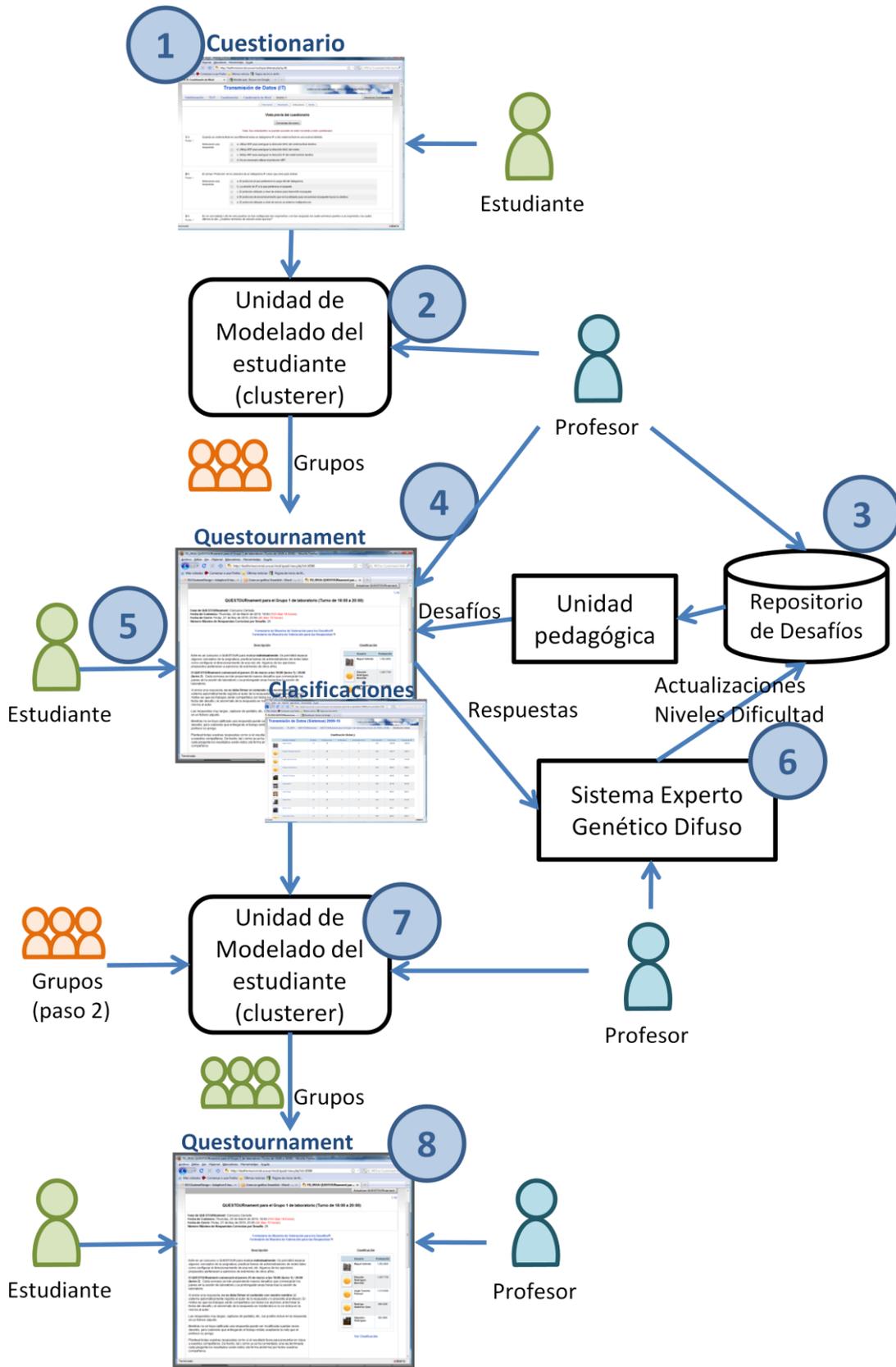


Figura 9. Ejemplo de uso del QUESTOURnament adaptativo

# **CAPÍTULO 4**

## **SISTEMA EXPERTO PARA LA ESTIMACIÓN DEL NIVEL DE DIFICULTAD EN QUESTOURNAMENT**

### **1 Descripción del problema**

Partiendo de la clasificación inicial del nivel de dificultad de los desafíos que realiza el profesor, se pretende optimizar esta clasificación de manera automática adaptándola a la dificultad real que encuentran los alumnos a la hora de tratar de responder a los desafíos. Se asume que el profesor estima la dificultad correctamente de muchos de los desafíos pero puede cometer errores en la estimación para algunos de ellos y estos errores deben ser corregidos por el sistema. Por ejemplo, podría suceder que un profesor considere que resolver un determinado desafío es fácil y, sin embargo, los estudiantes encuentran dificultades mientras lo resuelven por lo que el desafío no debería ser clasificado como fácil, de acuerdo al comportamiento real de los estudiantes. Observado este comportamiento, el profesor re-clasificaría la dificultad del desafío, estableciéndola en asequible o difícil. Esta tarea de re-clasificación es la que el sistema debe realizar automáticamente.

En la base de datos del sistema QUESTOURnament y de Moodle se almacena información relativa a las interacciones del alumno con el sistema (momentos en que el usuario accede a los desafíos y momento en que envía la respuesta) así como la nota obtenida al responder los desafíos. Esta información se puede utilizar para la estimación del nivel de dificultad de los desafíos. Concretamente, a partir de la información de la base de datos se puede conformar un conjunto de parámetros que podrá usar el sistema para la estimación: la *nota*, el número de *accesos* o lecturas por parte de un alumno a un determinado desafío y el *tiempo* que ha dedicado a resolver el desafío. Este último parámetro, sin embargo, no se puede obtener de forma precisa, ya que depende de muchos factores no controlables en un entorno como el de QUESTOURnament.

Así, se considera como *tiempo* de resolución del desafío al período en minutos transcurrido desde el último acceso al desafío por parte del alumno hasta el momento en que se envía la respuesta al sistema. Se ha escogido el último acceso puesto que, tras un análisis del comportamiento de los alumnos cuando se enfrentan a distintos desafíos simultáneamente, se ha encontrado que en muchos casos los alumnos tienden a realizar una primera lectura consecutiva de varios desafíos, seleccionando uno de ellos para responder. Posteriormente pueden realizar de nuevo una lectura consecutiva de los desafíos que les queda por responder, seleccionando uno de ellos y así sucesivamente. Un ejemplo real de este comportamiento se refleja en la Figura 10. También hay casos en que los alumnos no realizan el barrido inicial sino que acceden a un desafío y responden al mismo directamente, y en este caso también es correcto considerar el último acceso al desafío.

Sin embargo hay ejemplos con accesos adicionales como en el caso del desafío 4 de la Figura 11, que hacen incorrecto el tomar este valor como tiempo empleado en resolver este desafío y, por ello, este aspecto tendrá que ser depurado en futuros trabajos.

Tiempos anómalos en la resolución de ejercicios podrían ser eliminados teniendo en cuenta, por ejemplo, un tiempo mínimo y un tiempo estándar que adaptativamente se actualizan en función del conjunto de respuestas a cada pregunta (H. Liu et al. 2004). Para la solución actual se ha decidido que el profesor introduzca un tiempo mínimo estimado necesario para responder el desafío, de tal manera que se detecten aquellos casos en que el alumno resuelve un desafío *off-line* y accede al sistema sólo para enviar la respuesta, con lo que el tiempo calculado no sería el correcto.

A partir de estos parámetros sobre las respuestas de los alumnos a los distintos desafíos y la clasificación inicial de desafíos por parte del profesor, el sistema debe ser capaz de localizar desafíos mal clasificados y reclasificarlos, tal y como haría el profesor si analizara las respuestas de sus alumnos tras responder a los mismos.

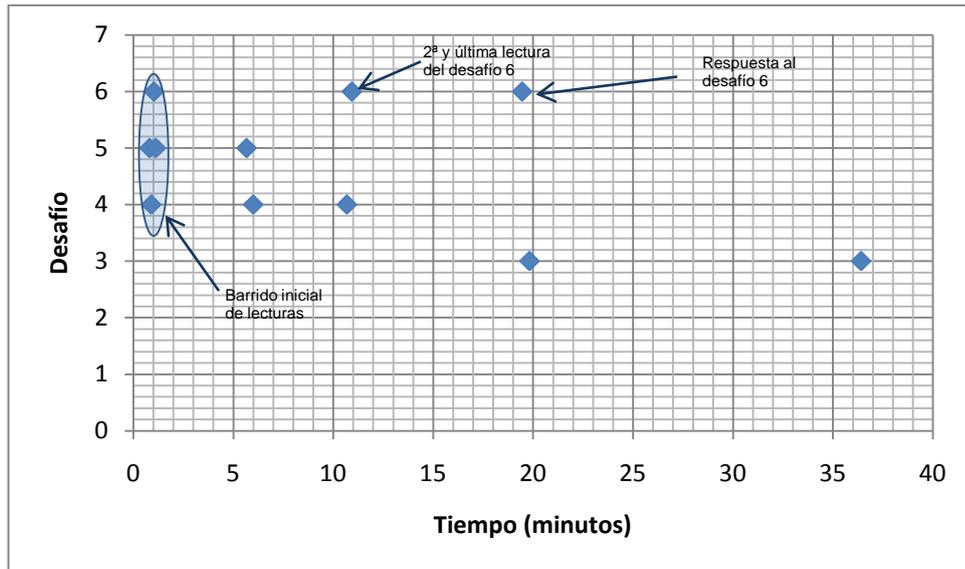


Figura 10. Representación temporal de las lecturas (accesos) y las respuestas de varios desafíos realizadas por un alumno (comportamiento habitual).

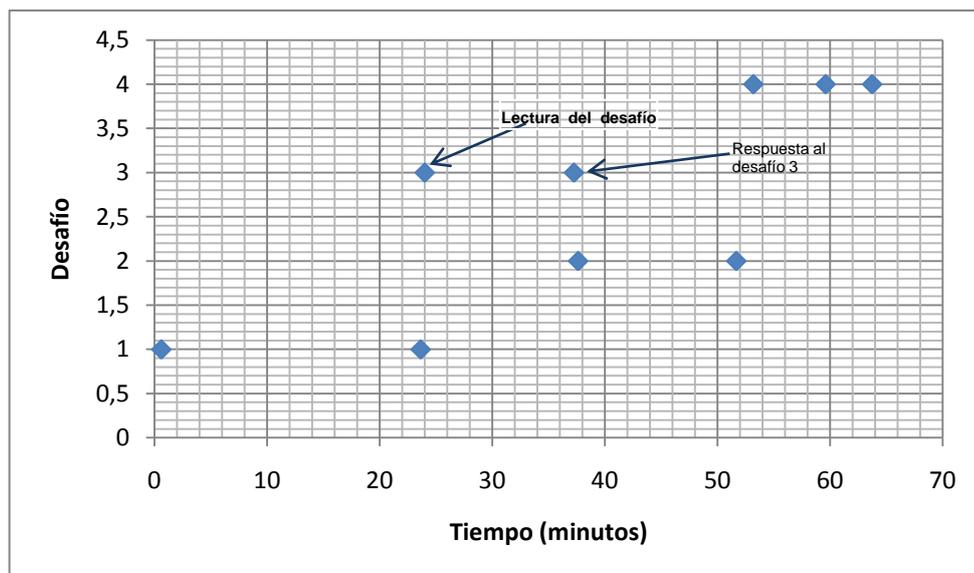


Figura 11. Representación temporal de las lecturas (accesos) y las respuestas de varios desafíos realizadas por un alumno (comportamiento anómalo en el desafío 4).

## 2 Búsqueda de la solución

Como se ha expuesto en el apartado 3 del capítulo 2, los sistemas *e-learning* adaptativos utilizan una amplia variedad de técnicas para lograr su objetivo. En la Tabla 5 se resumen las técnicas empleadas en los distintos sistemas adaptativos que se han analizado como paso previo al diseño de la solución técnica adecuada para el sistema QUESTOURnament.

<b>Referencia</b>	<b>Técnica</b>
(Albacete y K.A. VanLehn 2000)	Redes Bayesianas
(Capuano et al. 2003)	Lógica difusa
(W.-C. Chang y H.-C. Yang 2009)	TRI y clustering
(C.-M. Chen, H.-M. Lee, y Y.-H. Chen 2005)	TRI
(C. Chen 2008)	Algoritmos genéticos y TRI
(H. Chu et al. 2009)	Clustering
(Conejo et al. 2004)	TRI
(Guzman, Conejo, y Perez-de-la-Cruz 2007)	TRI
(Koutsojannis et al. 2007)	Algoritmos genéticos
(Hibou y Labat 2004)	Redes Bayesianas
(G. Hwang 2003)	Lógica difusa y mapas conceptuales
(Kavcic 2004)	Lógica difusa y reglas lingüísticas
(Lanzilotti y Roselli 2007)	Sistema experto
(Lilley, Barker, y Britton 2004)	TRI (3-PL)
(Mayo y Mitrovic 2001)	Redes Bayesianas y teoría decisión
(Noguez y Sucar 2005)	Redes Bayesianas
(Nouh, Karthikeyani, y Nadarajan 2006)	Redes Bayesianas y TRI
(C. Romero, S. Ventura, y Garcia 2008)	Clustering
(Tai, T.-A. Tsai, y F.M.-C. Chen 2001)	TRI
(Vozár y Bieliková 2008)	TRI
(Zakrzewska 2008)	Clustering

Tabla 5. Técnicas adaptativas aplicadas al *e-learning*

Las peculiaridades del sistema QUESTOURnament y el hecho de que muchas de las propuestas que se encuentran en la literatura se centran en productos específicos, hace difícil la directa aplicación de las soluciones existentes al presente problema. Las preguntas o desafíos que se proponen en el sistema QUESTOURnament no tienen restricciones en cuanto a:

- El tiempo disponible para contestar una pregunta. El profesor puede configurar cualquier tiempo límite para que los alumnos envíen la respuesta.
- El nivel de dificultad de las preguntas.
- El tipo de preguntas, que pueden ser de tipo verdadero/falso, elección múltiple, respuestas abiertas, problemas, etc., y que pueden estar relacionadas con cualquier temática o dominio de conocimiento.
- Los criterios de evaluación, que el profesor puede particularizar para cada desafío concreto.
- La motivación y estrategia seguida por cada alumno.

- El instante en que cada alumno accede y lee una pregunta. El sistema QUESTOURnament, al ser una herramienta telemática, permite participar en los concursos desde cualquier lugar y en cualquier momento.

Así, tanta libertad en el uso de la herramienta hace complicado el utilizar modelos predefinidos como los que aporta la Teoría de Respuesta al Ítem (TRI), que se aplica frecuentemente en los tests adaptativos informatizados.

Muchos sistemas tutores inteligentes se centran en estimar el conocimiento de los estudiantes y proporcionarles un camino de aprendizaje adecuado a éste. Algunos de estos sistemas organizan los contenidos de aprendizaje en dominios de conocimiento y presentan los contenidos adecuados para aprender los distintos conceptos en base a los conceptos previamente aprendidos. Este tipo de sistemas se basan en técnicas como las redes bayesianas (Hibou y Labat 2004) (Nouh, Karthikeyani, y Nadarajan 2006) y requieren la definición previa del dominio de conocimiento por un experto humano, el modelado de la red y sus dependencias, elementos que en muchos casos no es sencillo generalizar para los diferentes estudiantes (Noguez y Sucar 2005).

Por otro lado, existen sistemas adaptativos independientes del dominio de conocimiento que tratan de estimar el nivel de conocimiento del estudiante a partir de sus resultados al responder a un test. Muchos de estos sistemas aplican la Teoría de Respuesta al Ítem (Muñiz 1990) (Olea y Ponsoda 2003) para estimar tanto la dificultad de las preguntas como el nivel de conocimiento de los estudiantes (C.-M. Chen, H.-M. Lee, y Y.-H. Chen 2005) (Lilley, Barker, y Britton 2004) (Nouh, Karthikeyani, y Nadarajan 2006). Estos sistemas se acercan más al funcionamiento del QUESTOURnament adaptativo, pero las soluciones propuestas no se pueden aplicar a este sistema. La correcta aplicación de las teorías tradicionales para tests implica asumir ciertas condiciones que no se cumplen en la mayoría de los contextos en los que se aplica QUESTOURnament.

Los modelos típicamente usados de la Teoría de Respuesta al Ítem asumen que los ítems son unidimensionales, esto es, que la respuesta que se da a un ítem depende de un simple rasgo, que normalmente es el conocimiento de los estudiantes. Además, otra condición a asumir (que directamente se cumple si se da la unidimensionalidad) es la independencia local, esto es, que la respuesta que un sujeto da a una pregunta no viene influida por las respuestas que da a otras preguntas (Barbero 1999) (Muñiz 1990). Por ello, los tests adaptativos basados en la TRI se deben diseñar muy cuidadosamente para que se cumplan las anteriores condiciones ya que existen numerosas situaciones en que las condiciones anteriores pueden no cumplirse como cuando el orden de presentación de los ítems afecta a las respuestas.

Más aún, la TRI está pensada para ser usada en tests de potencia puros, asumiendo que los alumnos tienen un tiempo ilimitado para responder las preguntas (Barbero 1999). Por tanto, son varios los factores que podrían distorsionar los resultados obtenidos por el método TRI si se aplicara a un sistema competitivo como QUESTOURnament. Por un lado, los estudiantes pueden leer las preguntas tantas veces como deseen y, dentro de un plazo determinado, pueden responder cuando lo deseen, e incluso pueden resolver varias preguntas simultáneamente y luego responderlas. Por otro lado, la variación de la puntuación en el tiempo implica que el estudiante trata de resolver los desafíos lo más rápido posible. Claramente, el tiempo es un factor crucial en el sistema QUESTOURnament que debería ser tenido en cuenta en el modelo. Existen modelos basados en la TRI que tienen en cuenta el tiempo, como el descrito por (Roskam 1997) para tests de velocidad con límite de tiempo, que consideran tanto el tiempo invertido en responder como la corrección de la respuesta.

Sin embargo, las condiciones de nuestro sistema hacen que este tipo de modelos tampoco puedan ser empleados ya que se sigue presuponiendo que la respuesta es correcta o no es

correcta (sin tener en cuenta el grado de corrección) y no considera otros parámetros como la motivación o la estrategia. Frente a los modelos para ítems dicotómicos (de respuesta correcta o incorrecta), surgen modelos como los presentados por (Samejina 1997) y (Andersen 1997), que tratan con distintas categorías de respuestas (ej. “Muy bien”, “bien”, “regular”, “mal”, “muy mal”), permitiendo deducir la probabilidad de que un alumno con un conocimiento determinado reciba una determinada puntuación.

Aun resolviendo el problema del tipo de respuesta, el carácter competitivo de la herramienta QUESTOURnament, hace que los distintos factores de personalidad puedan influir mucho en la respuesta final a un desafío. Por ejemplo, hay estudiantes que son muy persistentes y dedican mucho tiempo a resolver una cuestión, asegurándose mucho de que la respuesta es la correcta, mientras que otros estudiantes se estresan con la competición y se precipitan enviando respuestas prematuras, buscando ser los primeros en responder. Por otro lado, como se ha mencionado anteriormente, los profesores no tienen limitaciones con el tipo de preguntas que pueden proponer ni con la estrategia de concurso que deseen plantear.

Hay distintos modelos de la TRI que cubren aspectos parciales del presente problema pero no existe un modelo que cubra todos los aspectos requeridos por el sistema. En este sentido, (van der Linden 2007) propone una solución jerárquica “*plug-and-play*” que consta de un modelo TRI, un modelo de distribución de los tiempos de respuesta y una estructura superior que tiene en cuenta las dependencias entre los parámetros de las personas y de los ítems en estos modelos. La ventaja de este modelo es que se puede escoger el modelo TRI más adecuado para el tipo de ítems e, independientemente, el modelo adecuado para los tiempos de respuesta. De cualquier manera una solución basada en esta idea, que tomara en cuenta todos los factores que afectan a la respuesta que un estudiante da a un desafío dentro de QUESTOURnament, sería demasiado compleja. Por tanto, es necesario buscar una solución específica para convertir al sistema QUESTOURnament en un Sistema Educativo Adaptativo Inteligente. Además, la herramienta QUESTOURnament se integra en la plataforma de teleformación Moodle, por lo que otra de sus peculiaridades es que está sujeta a un gran número de usuarios concurrentes, interactuando simultáneamente con el sistema, a diferencia de otras aplicaciones que no son accesibles desde Internet, sino desde una intranet o, simplemente desde un ordenador personal. Por ello, la carga computacional añadida por los algoritmos adaptativos seleccionados debe ser minimizada en la medida de lo posible, de forma que el tiempo de respuesta del sistema no se incremente drásticamente según aumenta el número de usuarios.

Hay otras soluciones más simples para tener en cuenta la dificultad del material de aprendizaje como las propuestas en (Hung et al. 2004) (Jong et al. 2006), que dan un peso a los materiales calculado como el cociente del número de veces que un ítem ha sido correcta o incorrectamente contestado entre el número total de veces que ha sido contestado.

Por otro lado, se encuentran otras soluciones enfocadas a la temática objetivo, como la propuesta por (Kunichika et al. 2002), que calculan la complejidad de preguntas sobre sentencias en inglés como una suma de factores ponderados, siendo los factores la complejidad de una sentencia por ejemplo (calculada teniendo en cuenta el número de proposiciones de la sentencia), etc.

Los algoritmos genéticos se han usado con éxito en problemas de búsqueda y optimización en el campo del *e-learning*. Hay ejemplos de su aplicación para proporcionar respuestas automáticas a las preguntas de los estudiantes (G. Hwang et al. 2008), obtener información útil del estudiante (C. Romero et al. 2002), predecir el rendimiento del estudiante (Minaei-Bidgoli et al. 2003), definir caminos de aprendizaje (C. Chen 2008) así como estimar el nivel de dificultad de los ejercicios propuestos en un sistema web (Koutsojannis et al. 2007).

Los algoritmos genéticos son técnicas útiles para testear y ajustar modelos cuantitativos. Tras construir un modelo a partir por ejemplo de un experimento científico, se recogen datos reales para probar y validar el modelo. Si se encuentran discrepancias, el modelo es modificado y el problema se repite hasta que éste es resuelto. Cuando se emplean algoritmos genéticos para probar y ajustar parámetros cuantitativos se están buscando parámetros que optimicen una función de *fitness*. El problema de determinar la dificultad de los desafíos se puede resolver empleando un algoritmo genético que ajuste la dificultad de los desafíos en función de algunos parámetros extraídos de las respuestas reales de los alumnos a los distintos desafíos. Partiendo de la clasificación de la dificultad de los desafíos que inicialmente estima el profesor, estas dificultades pueden ir ajustándose a los datos contenidos en las respuestas reales, mediante el empleo de un algoritmo genético.

Los algoritmos genéticos permiten buscar soluciones a un problema y, a diferencia de otros métodos de búsqueda dirigidos, combinan la búsqueda estocástica y dirigida, lo que les hace más robustos (Michalewicz 1996). Otra propiedad muy interesante de los algoritmos genéticos es que trabajan con un conjunto de posibles soluciones a un problema mientras que otros métodos sólo trabajan con un punto del espacio de búsqueda (Michalewicz 1996) (Deb 2001). Así, los algoritmos genéticos permitirán encontrar múltiples soluciones a un problema si hay más de una buena solución.

Por todos estos motivos, los algoritmos genéticos han sido la técnica escogida para resolver el problema de cómo determinar objetivamente la dificultad real de los desafíos. Asimismo, dada la complejidad y comportamiento no lineal del sistema, es adecuado el uso de la lógica difusa (Zadeh 1965), la cual ya ha sido aplicada con gran éxito en el campo de la educación (C.-M. Chen, H.-M. Lee, y Y.-H. Chen 2005) (Biswas 1995) (Semerci 2004) (Brugos et al. 1998) y de los sistemas de control de diferente índole (Dadone 2001) (Tomescu 2001).

El uso de la lógica difusa proporciona grandes ventajas al problema bajo estudio, dado que un sistema basado en reglas tradicional sólo puede reaccionar en comparación con umbrales. Los umbrales son suficientes si sólo hay unas pocas alternativas de adaptación donde poder elegir; sin embargo, si hay muchas alternativas y además cada una tiene necesidades de recursos superpuestas, intentar hacer adaptaciones correctas basadas sólo en umbrales es difícil. Por otra parte, los sistemas difusos ofrecen una solución natural a un escenario tan complejo. Asimismo, el modelo difuso es inherentemente genérico y muy configurable; tanto las reglas como las funciones de pertenencia para las variables lingüísticas pueden configurarse para que sean específicos de la aplicación particular.

Finalmente, el uso de la lógica difusa aumenta el espacio de soluciones posibles evitando la existencia de soluciones inviables como ocurre con los sistemas tradicionales, proporcionando así una buena solución a este problema. Al perder el enfoque binario, se cubren más variantes con una regla y se razona de una forma más sencilla y con un razonamiento más potente (Brule 1985). Asimismo, a medida que aumenta la complejidad de un problema, disminuye la posibilidad de analizarlo en términos precisos. Es decir, a medida que los problemas se vuelven más complejos, resulta más difícil usar la teoría lógica tradicional y construir métodos cuantitativos, que ponen demasiado énfasis en la precisión.

Como se ha mencionado previamente, (Koutsojannis et al. 2007) utilizan algoritmos genéticos para determinar el nivel de dificultad de los ejercicios propuestos en un sistema web que denominan *Artificial Intelligence Teaching System* (AITS). Sin embargo esta solución no es válida para QUESTOURnament por los siguientes motivos:

- AITS permite a los estudiantes realizar múltiples intentos hasta responder bien una pregunta. Asimismo los estudiantes pueden recibir pistas. Ninguna de estas características se cumple en QUESTOURnament.

- Para estimar el nivel de dificultad de las cuestiones, AITS tiene en cuenta el número de intentos, el número de pistas y el tiempo empleado. El sistema trabaja con preguntas contestadas correctamente y utiliza estos parámetros para estimar su dificultad. Sin embargo, con QUESTOURnament no todas las preguntas tienen por qué estar correctamente contestadas. Por eso, en QUESTOURnament la nota debe ser un parámetro de entrada al algoritmo genético.
- Con AITS el profesor debe asignar una dificultad inicial a las preguntas y además tiene que definir unas reglas iniciales basadas en los tres parámetros anteriormente mencionados para cada uno de los niveles de dificultad. Definir estas reglas resulta costoso para el profesor y es un paso que se desea evitar con QUESTOURnament adaptativo.

Dado que las soluciones existentes no son adecuadas para QUESTOURnament, se ha diseñado un nuevo sistema basado en el uso de los algoritmos genéticos, combinados con la lógica difusa, adecuado a las funcionalidades y requerimientos del sistema QUESTOURnament.

### 3 Arquitectura del sistema experto genético difuso

El sistema propuesto se basa en un sistema experto difuso que emplea un algoritmo genético para definir unas reglas difusas que se utilizan para clasificar las preguntas de acuerdo a su nivel de dificultad. Habitualmente, los sistemas expertos emplean bases de conocimiento generadas por expertos humanos de manera manual (Negnevitsky 2005). Sin embargo, las diferentes posibilidades de uso de QUESTOURnament no permiten que podamos definir unas reglas generales, sino que es necesario particularizar para cada caso. Por ejemplo, un profesor puede proponer preguntas de tipo test a sus alumnos mientras que otro profesor puede proponer trabajos más laboriosos. Asimismo, un profesor puede decidir que el tiempo de envío de la respuesta sea más o menos importante en la puntuación final del concurso, lo que influye finalmente en la rapidez con que los alumnos resuelven los desafíos. Por tanto, para cada caso particular, nuestro sistema aprende esta base de conocimiento mediante un algoritmo genético que caracteriza las preguntas de cada nivel de dificultad a partir de las respuestas de los alumnos a los desafíos de un curso concreto.

Partiendo de la clasificación inicial del nivel de dificultad que realiza el profesor, se pretende optimizar esta clasificación de manera automática adaptándola a la dificultad real que encuentran los alumnos a la hora de tratar de responder a los desafíos. El sistema trabaja con tres niveles de dificultad: fácil, asequible y difícil. Según se vayan enviando las diferentes respuestas a los distintos desafíos, los datos de interacción de los alumnos se van almacenando en la base de datos de hechos del sistema. En concreto para el cálculo de la dificultad se utiliza el siguiente patrón para cada una de las respuestas: *<tiempo, nota, accesos >*.

Como se ha explicado anteriormente, el parámetro *tiempo* refleja el período en minutos transcurrido desde el último acceso al desafío por parte del alumno hasta el momento en que se envía la respuesta al sistema.

El parámetro *nota* corresponde a la calificación obtenida por dicha respuesta y será un valor comprendido entre 0 y 100 (el sistema permite distintas escalas para puntuar las respuestas, pero se ha tomado la escala más utilizada por los usuarios).

El parámetro *accesos* indica el número de accesos o lecturas por parte de un alumno a un determinado desafío.

Por último, cada patrón de respuesta lleva asociado un *nivel de dificultad*, que se corresponde con la dificultad actual del desafío, asignada inicialmente por el profesor en el momento de su creación. De esta manera se pueden agrupar los patrones de respuesta por niveles de dificultad.

Una vez que el sistema albergue una cantidad razonable de respuestas (patrones) para los distintos desafíos planteados, se puede llevar a cabo la adaptación de la dificultad de los mismos mediante el empleo del sistema genético difuso. Para cada nivel de dificultad se buscarán reglas con el fin de clasificar los desafíos en función de los datos de interacción de los alumnos.

Así, la estimación del nivel de dificultad de los desafíos se realiza en las siguientes fases (ver Figura 12):

1. El algoritmo genético toma de la base de hechos todos los patrones de respuesta correspondientes a los desafíos de un nivel de dificultad dado y caracteriza esos patrones de respuesta mediante conjuntos clásicos.
2. A partir de la caracterización del algoritmo genético, el Generador del Modelo Difuso, genera los correspondientes conjuntos y reglas difusos, actualizando la base de conocimiento.
3. Una vez definidas las reglas y conjuntos difusos el Motor de Inferencia infiere para cada patrón de respuesta el nivel de dificultad que le corresponde.
4. La dificultad de un desafío es finalmente calculada a partir de la mediana resultante de los niveles de dificultad inferidos para todos los patrones de respuesta de un determinado desafío.

En los siguientes apartados se describe más detalladamente el funcionamiento de los distintos componentes del sistema.

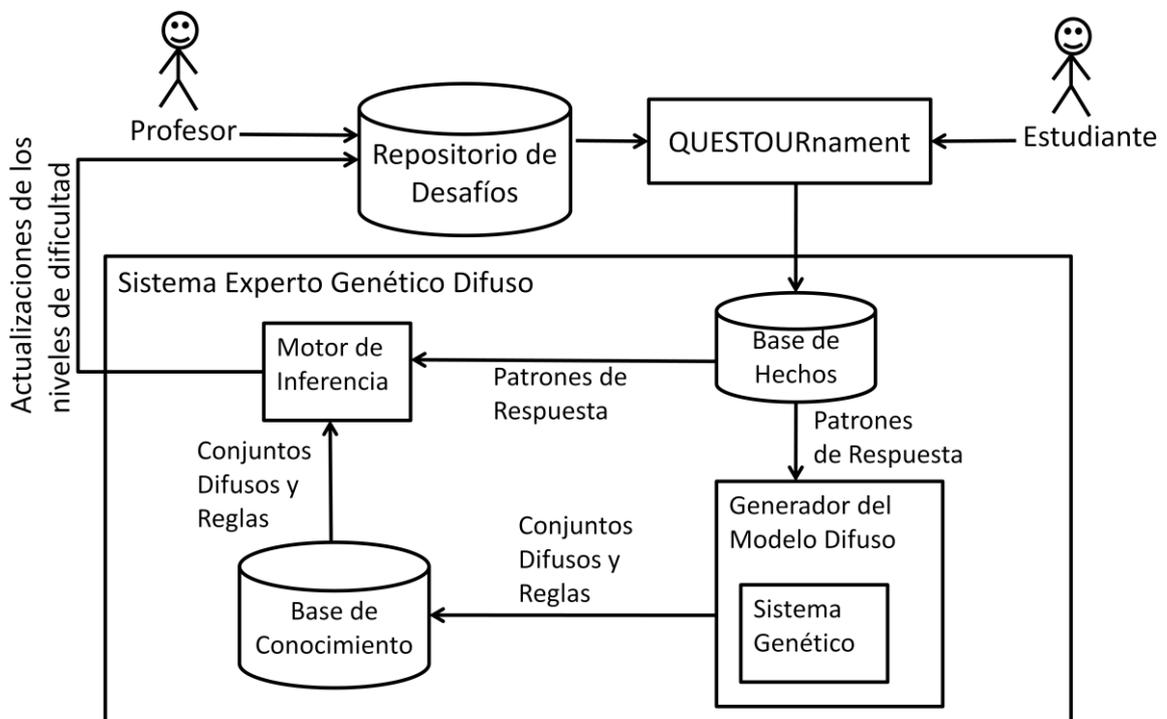


Figura 12. Arquitectura del sistema experto difuso-genético

## 4 Diseño del sistema

### 4.1 El sistema genético

Los algoritmos genéticos emulan la selección natural. Un algoritmo genético parte de una población inicial de individuos (también denominados cromosomas), que representan posibles soluciones a un problema. Esta población inicial suele ser generada aleatoriamente aunque también es posible establecer pautas para definirla. Como sucede en la selección natural, los mejores individuos se cruzan dando lugar a hijos que generalmente mejoran la especie. Así, un aspecto crucial en estos sistemas es cómo medir la aptitud, bondad o *fitness* de cada individuo o solución. Para ello se define una función de *fitness*, dependiente del problema específico que se esté resolviendo, que da una medida de esa bondad. Asimismo se definen las probabilidades de los individuos de cruzarse e incluso de sufrir mutación. Como se ha indicado, los individuos con mayor bondad o *fitness* tienen más posibilidades de cruzarse, con lo que el resultado es que, generación tras generación, los individuos aumentan su *fitness*. El algoritmo genético finaliza tras un determinado número de iteraciones, cuando no se producen mejoras en la población o cuando se cumple una determinada condición.

El sistema genético se ha diseñado en dos fases principales. La primera fase da como resultado un algoritmo genético básico, que no cubre todas las necesidades del problema, pero que es el punto de partida para el algoritmo genético final, al que se le añaden técnicas de diversidad, para que el algoritmo sea capaz de encontrar varias soluciones buenas al problema y no una única solución.

#### 4.1.1 Algoritmo genético básico

*Los contenidos del presente apartado han sido parcialmente publicados en (E. Verdú, L.M. Regueras, et al. 2010)*

El algoritmo genético diseñado inicialmente tiene las fases típicas de un algoritmo genético común, representadas en la Figura 13, y que se resumen a continuación:

*Fase 1. Generación de la población inicial.* Se generan N individuos o cromosomas de manera aleatoria que componen la población inicial:  $C_1, C_2, C_3, \dots, C_N$ . Estos individuos representan posibles soluciones al problema.

*Fase 2. Evaluación del fitness.* Para cada cromosoma se calcula su valor de fitness  $f(C_1), f(C_2), f(C_3), \dots, f(C_N)$ . Para el cálculo de este fitness se emplean los patrones de respuesta de la base de hechos.

A continuación, si se cumple el criterio de parada el algoritmo termina, en caso contrario, continúa en la siguiente fase.

*Fase 3. Selección (Ruleta).* Se seleccionan N cromosomas para ser cruzados en la siguiente fase. La probabilidad de selección de cada cromosoma es proporcional a su fitness según la siguiente expresión:

$$P_s(C_i) = \frac{f(C_i)}{\sum_{j=1}^N f(C_j)} \quad (1)$$

*Fase 4. Cruce.* Los cromosomas seleccionados en la etapa previa son emparejados. Con una probabilidad de cruce  $p_c$  se aplica en cada pareja un operador de cruce para dar lugar a dos hijos.

*Fase 5. Mutación.* Con una probabilidad  $p_m$  cada elemento hijo generado en la anterior fase puede sufrir una mutación definida por un operador de mutación uniforme. Al final de esta fase se tiene  $N$  hijos que sustituyen a los  $N$  individuos de la anterior población.

Las fases 2 a 5 se repiten hasta que se satisface el criterio de parada. Una vez que este criterio se cumple el algoritmo devuelve el cromosoma de mayor fitness que representa la mejor solución encontrada al problema.

Durante el proceso de diseño de este algoritmo genético, se han seguido los pasos típicos especificados por (Negnevitsky 2005):

1. Especificación del problema.
2. Representación del problema como un cromosoma o individuo.
3. Definición de la función de fitness.
4. Definición de los operadores genéticos.
5. Ejecución del algoritmo para el ajuste de sus parámetros.

Estos pasos son detallados en los siguientes apartados.

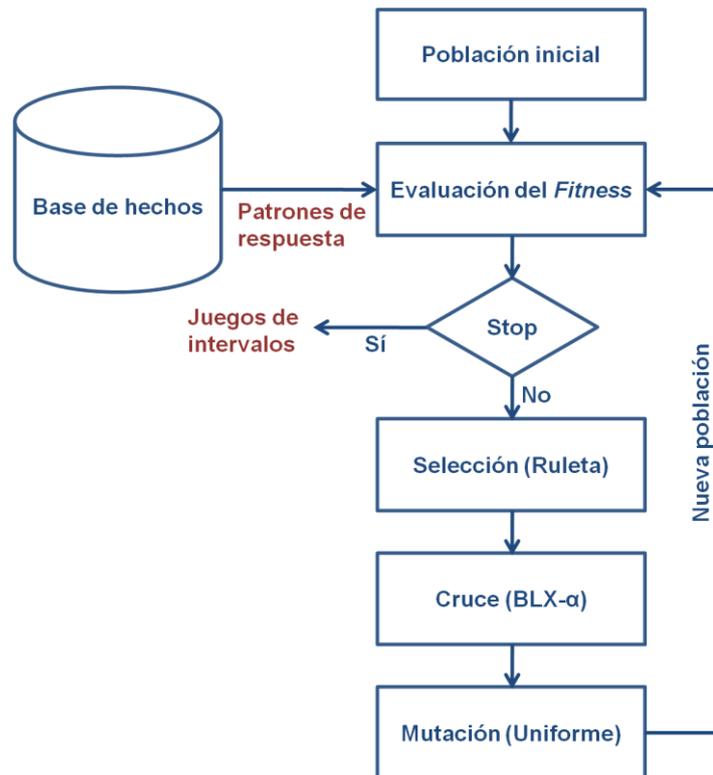


Figura 13. Fases del algoritmo genético básico

#### 4.1.1.1 Especificación del problema

Mediante el algoritmo genético se pretende caracterizar los patrones de respuesta  $\langle tiempo, nota, accesos \rangle$  correspondientes a los desafíos de un nivel de dificultad dado. El objetivo es encontrar un juego de intervalos  $[t_1, t_2][g_1, g_2][a_1, a_2]$  que contenga un mayor número de esos patrones. Un patrón  $\langle tiempo, nota, accesos \rangle$  pertenece o está contenido en un juego de intervalos  $[t_1, t_2][g_1, g_2][a_1, a_2]$  si:

$$t_1 < tiempo < t_2 \quad Y \quad g_1 < grade < g_2 \quad Y \quad a_1 < accesos < a_2$$

Además, es importante que la amplitud de estos intervalos no sea tal que cubra gran parte del dominio de las variables *tiempo*, *nota* y *accesos*, puesto que en este caso el juego de intervalos no aportaría información interesante.

En la Figura 14 se muestra el espacio de búsqueda completo y se representan las posibles soluciones al problema como cubos, cuyas aristas vienen determinadas por los intervalos de tiempo, nota y accesos. Estos cubos contienen un determinado número de patrones de respuesta, que son puntos situados en el espacio representados como aspas. El juego de intervalos  $[t_1, t_2][g_1, g_2][a_1, a_2]$  se torna más interesante que el juego de intervalos  $[t_1', t_2'][g_1', g_2'][a_1', a_2']$  dado que el primero, representado por el cubo más pequeño, contiene casi el mismo número de patrones en unos intervalos mucho más estrechos que el cubo grande. Así, la solución representada por el cubo pequeño es más específica y, por tanto, es más interesante, dado que indica dónde hay una mayor concentración de patrones. Por otro lado, la posible solución correspondiente a las notas más altas (cubo de la derecha) sólo incluye un patrón de entrada, luego no es una solución buena. El dominio de valores posibles para los intervalos de tiempo, nota y accesos es variable puesto que se puede configurar el tiempo máximo  $T_{max}$ , la nota máxima  $G_{max}$  y el número de accesos máximos  $A_{max}$ , aunque inicialmente se ha establecido un tiempo máximo de 160 minutos, una nota máxima de 100 y un número de accesos máximo de 8.

Como ya se ha mencionado previamente, la entrada al algoritmo es el conjunto de patrones de respuesta, que están almacenados en la base de hechos, y que tienen la forma  $\langle tiempo, nota, accesos \rangle$ . Además, cada patrón de respuesta lleva asociado el *nivel de dificultad* actual del desafío correspondiente. Cuando el objetivo del algoritmo genético es describir las características de cada grupo determinado por una clase (nivel de dificultad), se puede dejar fija la clase y ejecutar el algoritmo genético para cada uno de los valores posibles de esta variable objetivo o clase (Gündoğan, Alataş, y Karcı 2004)(C Romero et al. 2009). Por tanto, dado que se trabaja con tres niveles de dificultad, el algoritmo genético será ejecutado tres veces, para caracterizar cada uno de los niveles de dificultad. Los intervalos resultantes del algoritmo serán la base para generar un conjunto de reglas con la forma “IF *condicion* THEN *clase*” que permitirán clasificar los desafíos. A partir de los juegos de intervalos encontrados se determinará el antecedente de la regla (la condición) correspondiente a cada clase o nivel de dificultad (fácil, asequible o difícil).

Las diferentes fases del algoritmo genético básico se muestran en la Figura 13. Como ya se ha comentado, todas ellas son las fases típicas de estos algoritmos, excepto la fase relativa a la función de fitness, que es específica del problema en cuestión y, por tanto, depende bastante de la especificación del problema aquí expuesta. Asimismo, los operadores de cruce y mutación, aunque basados en operadores existentes, se han adaptado de acuerdo a la codificación de los cromosomas utilizada.

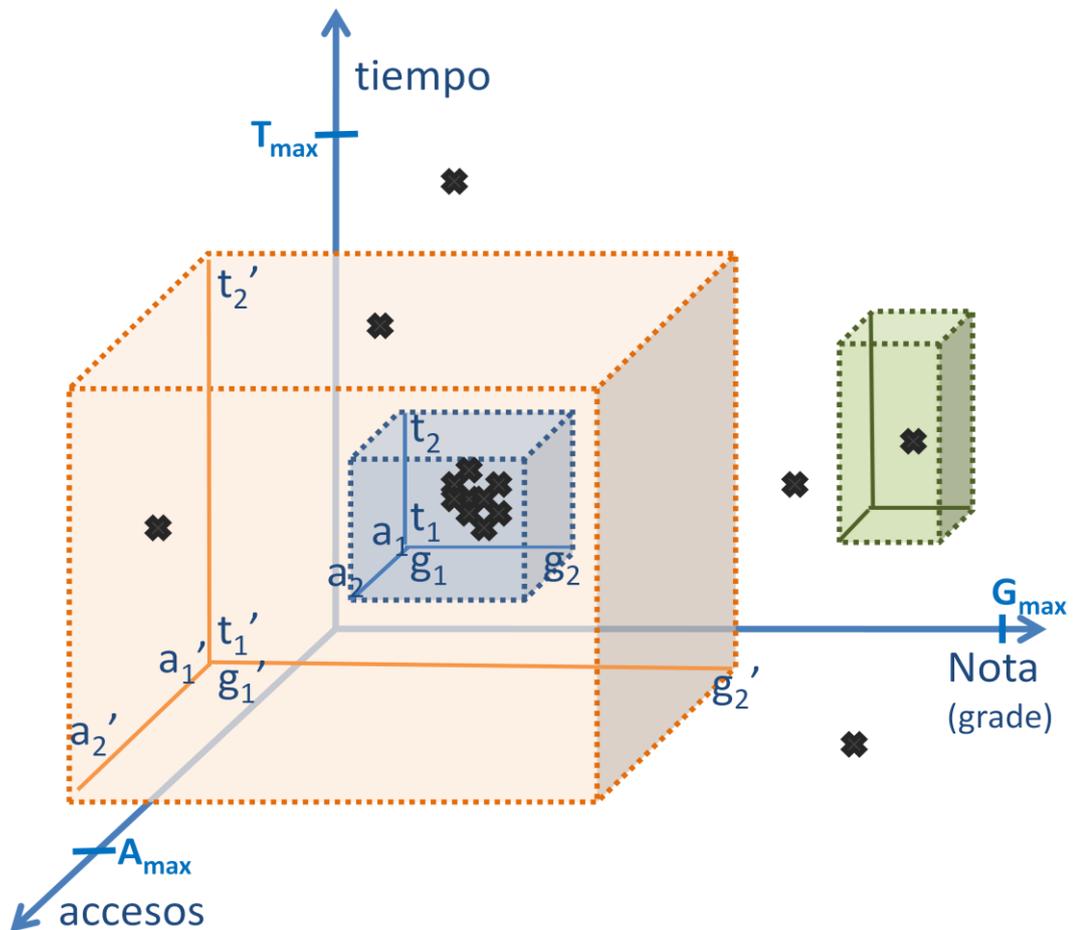


Figura 14. Espacio de búsqueda tridimensional

#### 4.1.1.2 Representación del problema mediante un individuo o cromosoma

Los cromosomas o individuos del algoritmo genético representan posibles soluciones al problema. Estos cromosomas están compuestos de genes, que son partes elementales indivisibles (Negnevitsky 2005). Típicamente los algoritmos genéticos trabajan con poblaciones de individuos o cromosomas codificados de forma binaria (M. Mitchell 2001), siendo cada gen un bit, aunque a veces un gen contiene un grupo de elementos o bits (Gündoğan, Alataş, y Karci 2004). Sin embargo, en muchos casos es posible y recomendable trabajar con poblaciones representadas por números reales. Por ejemplo, en problemas de optimización de parámetros reales, la representación real permite mayor precisión y rapidez en la ejecución que la representación binaria (Michalewicz 1996) (Wright 1991). En problemas de alta precisión o multidimensionales, la representación binaria da lugar a largas cadenas de bits que suponen un enorme espacio de búsqueda (Taplin y Qiu 2001) (Wen et al. 1998). Además, la representación real facilita el diseño de operadores de cruce y mutación que incorporen conocimiento específico del problema (Michalewicz 1996) (Wright 1991). Más aún, empleando una representación real se ahorra tiempo de CPU evitando la conversión entre la representación real y binaria (Taplin y Qiu 2001). Por tanto, para el presente algoritmo se ha escogido una representación real de los individuos, dado que se pretende optimizar parámetros reales.

Puesto que los individuos representan potenciales soluciones al problema y dado que las soluciones que se buscan son juegos de intervalos  $[t_1, t_2]$   $[g_1, g_2]$   $[a_1, a_2]$  que contienen simultáneamente un mayor número de patrones de entrada, los individuos de la población se representan por seis números reales  $[t_1, t_2, g_1, g_2, a_1, a_2]$  siendo  $t_1$  y  $t_2$  los límites inferior y superior respectivamente del intervalo de tiempo,  $g_1$  y  $g_2$  corresponden a los límites inferior y superior del intervalo de nota o *grade* y, por último,  $a_1$  y  $a_2$  corresponden a los límites inferior y superior del intervalo que se refiere al número de accesos.

Como previamente se ha comentado, a partir de estos intervalos se crearán reglas. Por tanto, se está empleando el método de codificación “Cromosoma=regla” (cada cromosoma codifica una regla) frente al método “Cromosoma=conjunto de reglas”, también llamado *Pittsburgh*, en el que cada cromosoma codifica un conjunto de reglas, siendo estos dos métodos los más extendidos en la literatura (Cordón 2004) (del Jesus et al. 2007). Para el presente problema es más adecuado que cada cromosoma represente una regla, ya que el objetivo es encontrar un juego reducido de reglas en que la calidad de cada regla sea independiente de las demás, no siendo necesario evaluar el conjunto de reglas simultáneamente.

Como se muestra en la Figura 13 la primera fase del algoritmo genético corresponde a la generación inicial de la población que se realiza de manera aleatoria. Lo más común es generar la población inicial aleatoriamente pero se encuentran casos en que la población inicial se genera de manera controlada. Por ejemplo, (Gündoğan, Alataş, y Karci 2004) propone crear la población inicial mediante un método de generación uniforme. Se puede generar la población inicial repartiendo los individuos uniformemente por el espacio de búsqueda. También se puede generar la mitad de la población aleatoriamente y la otra mitad a partir de la información de los individuos ya generados (Díaz-Gomez y Hougen 2006). En el presente problema, simplemente a cada una de las seis variables del cromosoma se le asigna un valor numérico real generado aleatoriamente dentro del dominio de valores posibles para cada una de las variables, cumpliendo la restricción de que el número que representa el límite inferior de un intervalo sea inferior al número que representa el correspondiente límite superior (si esto no se cumple, simplemente los valores se intercambian).

#### 4.1.1.3 Definición de la función de fitness

La fase de evaluación del fitness se trata de una fase crucial. El algoritmo genético requiere la función de fitness para evaluar la calidad de los individuos de la población, especificando los atributos deseados en una solución óptima (Riekert, Malan, y Engelbrecht 2009). Por tanto, diseñar una función de fitness adecuada al problema es muy importante para la efectividad y eficiencia del algoritmo genético (Fan et al. 2004) (López-Pujalte, Guerrero-Bote, y de Moya-Anegón 2003). Dado que la función de fitness debe caracterizar aquello que se quiere optimizar, ésta es dependiente del problema específico. Por ejemplo, hay funciones de fitness específicas para problemas de recuperación de información (López-Pujalte, Guerrero-Bote, y de Moya-Anegón 2003) (G. Hwang et al. 2008) (Fan et al. 2004) o específicas para resolver el problema de la “*Serpiente en la Caja*” (Díaz-Gomez y Hougen 2006). En algunos problemas de descubrimiento de subgrupos, los autores basan su función de fitness en una o más medidas clásicas encontradas en la literatura, que objetivamente evalúan las reglas inferidas (del Jesus et al. 2007).

El objetivo del algoritmo genético del presente problema es encontrar los intervalos de *tiempo*, *nota* y *número de accesos* que contengan un mayor número de patrones de respuesta correspondientes a los desafíos de un nivel de dificultad dado. Así, el algoritmo genético caracteriza las respuestas correspondientes a un nivel de dificultad. A partir de esta caracterización, se generarán reglas de clasificación que permitan clasificar los desafíos.

De acuerdo a este objetivo se ha definido la función de *fitness*. Puesto que las soluciones se van a utilizar para definir reglas de clasificación, previamente a la definición de la función de *fitness* se han analizado distintas medidas que se encuentran en la literatura para la evaluación de reglas inferidas. A continuación se describen algunas de estas medidas, teniendo en cuenta que una regla inferida  $R^i$  tiene la forma “IF  $Cond^i$  THEN  $Class_j$ ”, siendo  $Cond^i$  los atributos y sus valores (antecedente de la regla) y  $Class_j$ , el valor de la clase (consecuente de la regla):

- *Cobertura (Coverage)* de una regla  $R^i$ : mide el porcentaje de ejemplos cubiertos de media por una regla y se define con la siguiente fórmula (del Jesus et al. 2007) (Lavrač et al. 2004) (Webb y Songmao Zhang 2005):

$$Cov(R^i) = p(Cond^i) = \frac{n(Cond^i)}{n_s} \quad (2)$$

donde  $n(Cond^i)$  es el número de ejemplos que verifican la condición  $Cond^i$  descrita en el antecedente de la regla (independientemente de la clase  $Class_j$  a la que pertenece) y  $n_s$  es el número total de ejemplos. La cobertura es, por tanto, una medida de la *generalidad* de la regla (Lavrač, Flach, y Zupan 1999).

- *Soporte (Support)* de una regla  $R^i$ : se refiere al número de ejemplos que satisfacen tanto el antecedente como el consecuente de una regla y se define con la siguiente fórmula (del Jesus et al. 2007) (Lavrač et al. 2004) (Webb y Songmao Zhang 2005) (H Wang et al. 2005):

$$Sup(R^i) = p(Class_j \cdot Cond^i) = \frac{n(Class_j \cdot Cond^i)}{n_s} \quad (3)$$

donde  $n(Class_j \cdot Cond^i)$  es el número de ejemplos que satisfacen las condiciones del antecedente  $Cond^i$  y a su vez pertenecen a la clase  $Class_j$  indicada en el consecuente de la regla. En la anterior expresión se calcula el soporte dividiendo por el número total de ejemplos pero también se puede calcular dividiendo por el número de ejemplos de la clase u otras variaciones. Al soporte también se le denomina *frecuencia* (Lavrač, Flach, y Zupan 1999).

- *Tamaño*: medida de complejidad calculada como el número de reglas inducidas (del Jesus et al. 2007) (Lavrač et al. 2004). También se puede medir la complejidad como el número medio de reglas obtenidas por clase o el número medio de variables por regla.
- *Significancia* de una regla  $R^i$ : se refiere a la razón o cociente de verosimilitud (*likelihood ratio*) de una regla y se define con la siguiente fórmula (del Jesus et al. 2007) (Lavrač et al. 2004):

$$Sig(R^i) = 2 \cdot \sum_{j=1}^{n_c} n(Class_j \cdot Cond^i) \log \frac{n(Class_j \cdot Cond^i)}{n(Class_j) \cdot p(Cond^i)} \quad (4)$$

donde  $p(Cond^i)$  se usa como un factor de normalización,  $n_c$  es el número de valores para la variable objetivo, esto es, la clase. Nótese que aunque cada regla predice una clase específica, esta medida mide únicamente la significancia de la condición de la regla (Lavrač et al. 2004).

- *Unusualness* de una regla  $R^i$ : se refiere a la *precisión relativa ponderada* (*Weighted Relative Accuracy -WRAcc*) propuesta por (Lavrač, Flach, y Zupan 1999).

$$WRAcc(R^i) = \frac{n(Cond^i)}{n_s} \left( \frac{n(Class_j, Cond^i)}{n(Cond^i)} - \frac{n(Class_j)}{n_s} \right) \quad (5)$$

La precisión relativa ponderada de una regla se puede describir como el balance entre la cobertura de la regla  $p(Cond^i)$  y su ganancia en precisión ( $p(Class_j, Cond^i) - p(Class_j)$ ). A través del cálculo de esta ganancia se pretende tener una idea del volumen del efecto asociado a la interacción entre el antecedente y el consecuente de la regla (Webb y Songmao Zhang 2005). Además, se evita obtener alta precisión relativa con reglas muy específicas (reglas poco generales, en las que pocos ejemplos cumplen el antecedente), introduciendo la cobertura como factor. Una regla requiere ser suficientemente general además de tener una alta precisión relativa para que sea interesante. La precisión relativa ponderada (WRA) la denominan en (Webb y Songmao Zhang 2005) *leverage o ganancia*.

- *Confianza* de una regla  $R^i$ : mide la frecuencia relativa de los ejemplos que satisfacen antecedente y consecuente entre aquellos que satisfacen sólo el antecedente (H Wang et al. 2005). (Webb y Songmao Zhang 2005) la expresan como el cociente entre el soporte y la cobertura:

$$Conf(R_i) = \frac{Sup(R^i)}{Cov(R^i)} = \frac{n(Class_j, Cond^i)}{n(Cond^i)} \quad (6)$$

De acuerdo a (Lavrač, Flach, y Zupan 1999) en la inducción predictiva orientada a la clasificación, la medida más usada es la *precisión (accuracy)*, que definen como la probabilidad condicional de que se cumpla el consecuente si se cumple el antecedente, esto es, la confianza aquí definida. La precisión se puede emplear para medir la fiabilidad de la regla en la predicción de casos positivos.

- *Novedad* de una regla  $R_i$ : se define como la diferencia relativa entre el número de ejemplos para los que se cumple el antecedente y consecuente de la regla y el valor esperado bajo independencia (Lavrač, Flach, y Zupan 1999):

$$Nov(R^i) = p(Class_j, Cond^i) - p(Class_j) \cdot p(Cond^i) \quad (7)$$

Así, según la definición de novedad, sólo estamos interesados en un soporte alto si  $p(Class_j)$  y/o  $p(Cond^i)$  son relativamente bajos.

Una vez conocidas las anteriores medidas, se analiza su conveniencia para el presente problema, dado que el valor real de una regla es subjetivo y depende fuertemente del dominio particular y de los objetivos buscados (Zhou et al. 2009). El objetivo del algoritmo genético es caracterizar los diferentes niveles de dificultad en función del comportamiento de los alumnos al responder los desafíos de esos niveles. Este comportamiento viene definido por los patrones de respuesta <tiempo, nota, accessos> y el algoritmo trata de encontrar aquellos intervalos  $[t1, t2]$   $[g1, g2]$   $[a1, a2]$  que contengan un mayor número de estos patrones de respuesta en un nivel determinado. A partir de estos intervalos se generarán unas reglas para clasificar los desafíos. Como se indica, se pretende caracterizar los comportamientos de los alumnos al responder los distintos niveles y es interesante conocer si para distintos niveles, existen comportamientos similares. Por esto, medidas como la *confianza* o la *precisión relativa ponderada*, que a priori se

presentan como interesantes, deben ser descartadas, dado que si el antecedente de la regla lo cumplen muchos ejemplos en distintos niveles de dificultad, la regla podría ser descartada, lo que no interesa que suceda en el presente problema. Teniendo esto en cuenta el *soporte* se presenta como una medida adecuada puesto que indica el porcentaje de ejemplos dentro de una clase (o nivel de dificultad) que están cubiertos por esa regla. Además, la medida de *soporte* es sencilla y permite trabajar únicamente con los patrones del nivel de dificultad que se está caracterizando, lo que supone disminuir el coste computacional de manera importante. Emplear otras medidas como *confianza* o *precisión relativa ponderada* implica operar con patrones de todos los niveles de dificultad, lo que incrementa el coste computacional de manera importante. Dado que la función de fitness es calculada múltiples veces a lo largo de la ejecución de un algoritmo genético, el tiempo necesario para su evaluación es un aspecto crítico (Minaei-Bidgoli et al. 2003). Más aún, cuando se está interesado en reglas que maximizan alguna medida específica tal y como es el *soporte* en este caso, incluir reglas que también maximicen otras medidas dará lugar a mayor complejidad y confusión en las reglas descubiertas (Webb y Songmao Zhang 2005).

Por tanto, la función de fitness se basa en el *soporte* y trabaja con patrones o ejemplos pertenecientes a una clase (nivel de dificultad), por lo que el algoritmo debe ser ejecutado al menos una vez por clase (Gündoğan, Alataş, y Karci 2004).

El *soporte* de una regla se define como la frecuencia de los ejemplos cubiertos por la regla y correctamente clasificados, esto es, los que satisfacen tanto el antecedente como el consecuente de la regla. Así, la función de *fitness* del presente problema se define mediante la siguiente ecuación:

$$Fitness(C_i) = \frac{n_j^i}{n_j} \cdot w_t \cdot w_g \cdot w_a \quad (8)$$

donde  $n_j^i$  representa el número de patrones de dificultad  $j$  contenidos en el juego de intervalos definidos por el cromosoma  $C_i$ , y  $n_j$  es el número total de patrones de dificultad  $j$ . De esta manera el cociente entre  $n_j^i$  y  $n_j$  correspondería a la medida de *soporte* si se considera que cada nivel de dificultad corresponde a una clase y que la condición de pertenecer a esa clase (antecedente) es que un patrón esté contenido en el juego de intervalos definidos por el cromosoma  $C_i$ . Dado que los intervalos muy amplios tenderán a tener mayor soporte, se emplean los pesos  $w_t$ ,  $w_g$  y  $w_a$  para penalizar aquellos intervalos con una amplitud lo suficientemente grande como para dejar de ser una solución interesante y útil para resolver el problema. El objetivo de estos pesos es encontrar un equilibrio entre la amplitud del intervalo y el número de patrones incluidos (E. Verdú, L.M. Regueras, et al. 2010). Por ejemplo, si se tienen 100 patrones de respuesta para caracterizar las preguntas del nivel de dificultad fácil y todos tienen una nota alta entre 80 y 100, el soporte de un cromosoma que cuyo intervalo para nota sea [20,100] será el máximo ya que todos los patrones están dentro de ese intervalo (se están ignorando las variables tiempo y accesos para mayor claridad de esta explicación). Sin embargo este intervalo tan amplio no aporta información útil sobre la nota que obtienen los estudiantes al resolver los desafíos fáciles. Por otro lado, un cromosoma que defina el intervalo para nota [80,100] tendrá el mismo soporte pero es mucho más informativo puesto que es más específico. Así, se debe incluir un factor en la medida de soporte para premiar aquellos cromosomas que representan intervalos más estrechos. En concreto para este ejemplo, en que el intervalo correspondiente al primer cromosoma es 4 veces más amplio que el del segundo, el peso  $w_g$  de este segundo cromosoma es igual a 1.2 mientras que el del primer cromosoma toma el valor 0.1, por lo que su valor de *fitness* es mucho más pequeño. Los valores de los pesos para las distintas amplitudes han sido determinados experimentalmente.

#### 4.1.1.4 Definición de los operadores genéticos

Una vez que se ha especificado la función de fitness, se debe definir tanto el método de selección como los operadores de cruce y mutación.

##### 4.1.1.4.1 Selección

En la fase de selección, como su propio nombre indica, se seleccionan los individuos que serán cruzados durante la siguiente fase dando lugar a descendientes. El objetivo de los métodos de selección, también denominados operadores de reproducción, es identificar y hacer múltiples copias de buenas soluciones de una población así como eliminar las malas soluciones, manteniendo el tamaño de la población constante (Deb 2001). Por ello, cuanto mayor fitness tiene un individuo, mejor solución representa y, por tanto, mayor es la probabilidad de que sea seleccionado para el cruce posterior.

El método de selección implementado en el presente algoritmo es el *método de la ruleta*, que emula el comportamiento del juego de azar de la ruleta de los casinos, siendo uno de los métodos de selección más utilizados (Negnevitsky 2005). Por ejemplo, (Gündoğan, Alataş, y Karci 2004) (Koutsojannis et al. 2007) (López-Pujalte, Guerrero-Bote, y de Moya-Anegón 2003) (Pró, Ruiz, y Yañez 2005) emplean este método de selección.

El método de la ruleta consta de las siguientes operaciones (Michalewicz 1996):

1. Cálculo del valor de fitness  $f_i(C_i)$  de cada uno de los cromosomas  $C_i$  de la población ( $i=1,2,\dots,N$ ), siendo  $N$  el tamaño de la población.
2. Cálculo del fitness total de la población  $F$ , que es la suma de los valores de fitness de todos los cromosomas de la población:

$$F = \sum_{i=1}^N f_i(C_i) \quad (9)$$

3. Cálculo de la probabilidad de selección  $p_{si}$  de cada cromosoma  $C_i$  de la población ( $i=1,2,\dots,N$ ), que corresponde al fitness del cromosoma  $C_i$  dividido por el fitness total:

$$p_{si} = \frac{f_i(C_i)}{F} \quad (10)$$

4. Cálculo de la probabilidad acumulada  $p_{ai}$  de cada cromosoma  $C_i$  de la población ( $i=1,2,\dots,N$ ):

$$p_{ai} = \sum_{j=1}^i p_{sj} \quad (11)$$

5. Selección de un cromosoma de la población generando un número aleatorio  $r$  entre 0 y 1, con el siguiente procedimiento:

*Si  $r < p_{a1}$  entonces*

*seleccionar  $C_1$*

*Si no entonces*

*Seleccionar  $C_i$  tal que  $p_{a(i-1)} < r \leq p_{ai}$*

*Fin Si*

Este paso se repite  $N$  veces para seleccionar un número de cromosomas igual al tamaño de la población de partida.

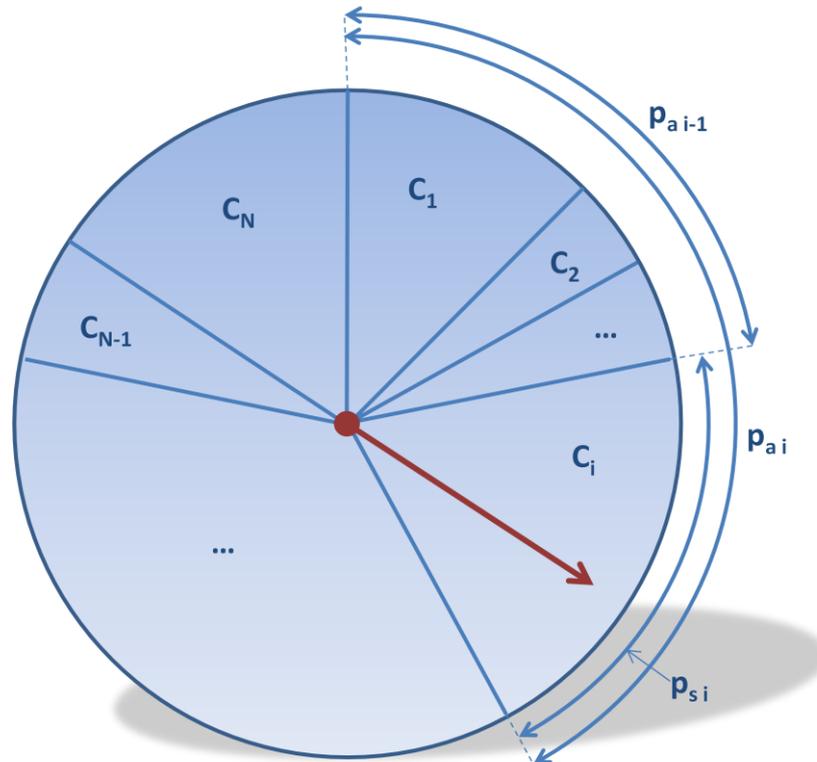


Figura 15. Método de selección de la ruleta

En la Figura 15 se representa mediante un esquema el funcionamiento de la técnica de la ruleta de selección para una más fácil comprensión así como para ilustrar el por qué se denomina de esta manera. A cada cromosoma  $C_i$  le corresponde una porción de la ruleta cuya área es proporcional a su probabilidad de selección  $p_{si}$ , esto es, proporcional a su *fitness*. La probabilidad acumulada de un cromosoma  $C_i$  es la porción que incluye el área del cromosoma  $C_i$  más las áreas de los cromosomas  $C_j$ ,  $j=1\dots i-1$ . Cada vez que se genera el número aleatorio  $r$  especificado en el paso 5 de la enumeración anterior es como si se tirase la ruleta y, lógicamente, tienen mayor probabilidad de ser seleccionados aquellos individuos que tienen mayor porción de ruleta. Como la porción de la ruleta es proporcional al *fitness*, los individuos con más *fitness* tienen mayor probabilidad de ser seleccionados. La ruleta se tira tantas veces como número de individuos de la población.

#### 4.1.1.4.2 Cruce

Cuando se trabaja con cromosomas codificados con números reales no se puede utilizar las operaciones triviales de cruce o mutación típicamente usadas en los algoritmos genéticos y en ocasiones, se convierte en un reto el seleccionar unos métodos adecuados para el cruce y la mutación de los cromosomas (Deb 2001). Existen muchas variantes de operadores de cruce en la literatura. La adecuación de un operador de cruce determinado depende de la función de *fitness* particular y de la codificación concreta empleada (M. Mitchell 2001).

Algunos operadores de cruce son los siguientes:

- *Cruce simple* (Taplin y Qiu 2001) (Michalewicz 1996):  $c_v^t=(v_1,v_2,\dots,v_N)$  y  $c_w^t=(w_1,w_2,\dots,w_N)$  son cruzados en la posición  $k$  dando lugar a los descendientes

- $c_v^{t+1}=(v_1,\dots,v_k,w_{k+1},\dots,w_N)$  y  $c_w^{t+1}=(w_1,\dots,w_k,v_{k+1},\dots,v_N)$ , donde  $k$  se selecciona aleatoriamente entre los valores 2 y  $N-1$ .
- *Cruce aritmético completo* (Taplin y Qiu 2001) (Michalewicz 1996): el cruce de  $c_v^t$  y  $c_w^t$  se basa en una combinación lineal que da lugar a los descendientes  $c_v^{t+1} = r(c_v^t) + (1-r)(c_w^t)$  y  $c_w^{t+1} = r(c_w^t) + (1-r)(c_v^t)$ , siendo  $r$  un valor aleatorio entre 0 y 1.
  - *Cruce heurístico* (Michalewicz 1996): el cruce de  $c_v^t$  y  $c_w^t$  da lugar a un hijo  $c_z^{t+1} = c_v^t + r(c_v^t - c_w^t)$ , siendo  $r$  un valor aleatorio entre 0 y 1 y no siendo el valor de fitness del padre  $c_v^t$  peor que el del padre  $c_w^t$ . De esta manera la búsqueda se produce en una determinada dirección.
  - *Cruce lineal* (Deb 2001): a partir de dos padres  $c_v^t$  y  $c_w^t$  se crean tres hijos  $c_v^{t+1} = 0.5(c_v^t + c_w^t)$ ,  $c_w^{t+1} = 1.5c_v^t - 0.5c_w^t$  y  $c_z^{t+1} = (-0.5c_v^t + 1.5c_w^t)$ .
  - *Cruce mixto (BLX- $\alpha$ )* (Deb 2001): si  $c_v^t$  y  $c_w^t$  son los padres y  $c_v^t < c_w^t$ , los descendientes se calculan con la siguiente fórmula en la que  $u$  es un número aleatorio entre 0 y 1 y  $\alpha$  es una constante cuyo valor se debe definir:

$$c_v^{t+1} = (1 - \gamma)c_v^t + \gamma c_w^t$$

$$c_w^{t+1} = (1 - \gamma)c_w^t + \gamma c_v^t \quad (12)$$

$$\text{siendo } \gamma = (1 + 2\alpha)u - \alpha$$

Este operador es interesante puesto que permite una búsqueda de soluciones adaptativa (Deb 2001). En definitiva el resultado de las anteriores operaciones es que se selecciona un valor aleatoriamente dentro del rango  $[c_v^t - \alpha(c_w^t - c_v^t), c_w^t + \alpha(c_w^t - c_v^t)]$ , si  $c_v^t < c_w^t$ . Por tanto, cuando la diferencia entre los cromosomas padres es pequeña, la diferencia entre cromosomas hijos y padres es también pequeña y, en caso contrario, la diferencia es grande. Así, cuando el algoritmo genético comienza, la población generada aleatoriamente será diversa y el cruce dará lugar a hijos diversos explorando el espacio de búsqueda más eficazmente. En el inicio, por tanto, se mejora la etapa de exploración del algoritmo. Sin embargo, en las fases más avanzadas en que se está más cerca de la solución, mediante este operador se explora en regiones más localizadas, mejorando la etapa de explotación del algoritmo.

Se puede encontrar más definiciones de operadores de cruce así como de mutación en (Deb 2001) (Michalewicz 1996) (Booker et al. 2000) (Gen 1997).

Para el presente problema se han diseñado operadores de cruce específicos, adaptados al formato de los cromosomas, pero basados en los operadores existentes. Concretamente tres operadores de cruce han sido diseñados, implementados y probados, para seleccionar finalmente el operador BLX- $\alpha$ . Se describe a continuación el diseño de estos operadores:

- *Cruce aritmético*: a partir de dos padres  $c_v^t = (t_1, t_2, g_1, g_2, a_1, a_2)$  y  $c_w^t = (t_1', t_2', g_1', g_2', a_1', a_2')$  se generan los hijos

$$c_v^{t+1} = (rt_1 + (1-r)t_1', rt_2 + (1-r)t_2', rg_1 + (1-r)g_1', rg_2 + (1-r)g_2', ra_1 + (1-r)a_1', ra_2 + (1-r)a_2' )$$

$$c_w^{t+1} = (rt_1' + (1-r)t_1, rt_2' + (1-r)t_2, rg_1' + (1-r)g_1, rg_2' + (1-r)g_2, ra_1' + (1-r)a_1, ra_2' + (1-r)a_2 )$$

- *Cruce BLX- $\alpha$* : a partir de dos padres  $c_v^t = (t_1, t_2, g_1, g_2, a_1, a_2)$  y  $c_w^t = (t_1', t_2', g_1', g_2', a_1', a_2')$  se generan los elementos de los vectores de los hijos  $c_v^{t+1}$  y  $c_w^{t+1}$  de la siguiente manera:

$$t_1^{t+1} \leftarrow \begin{cases} \text{random}(t_1' - \alpha(t_1 - t_1'), t_1 + \alpha(t_1 - t_1')) & \text{si } t_1 \geq t_1' \\ \text{random}(t_1 - \alpha(t_1' - t_1), t_1' + \alpha(t_1' - t_1)) & \text{si } t_1' \geq t_1 \end{cases}$$

$$t_2^{t+1} \leftarrow \begin{cases} \text{random}(t_2' - \alpha(t_2 - t_2'), t_2 + \alpha(t_2 - t_2')) & \text{si } t_2 \geq t_2' \\ \text{random}(t_2 - \alpha(t_2' - t_2), t_2' + \alpha(t_2' - t_2)) & \text{si } t_2' \geq t_2 \end{cases}$$

$$g_1^{t+1} \leftarrow \begin{cases} \text{random}(g_1' - \alpha(g_1 - g_1'), g_1 + \alpha(g_1 - g_1')) & \text{si } g_1 \geq g_1' \\ \text{random}(g_1 - \alpha(g_1' - g_1), g_1' + \alpha(g_1' - g_1)) & \text{si } g_1' \geq g_1 \end{cases}$$

$$g_2^{t+1} \leftarrow \begin{cases} \text{random}(g_2' - \alpha(g_2 - g_2'), g_2 + \alpha(g_2 - g_2')) & \text{si } g_2 \geq g_2' \\ \text{random}(g_2 - \alpha(g_2' - g_2), g_2' + \alpha(g_2' - g_2)) & \text{si } g_2' \geq g_2 \end{cases}$$

$$a_1^{t+1} \leftarrow \begin{cases} \text{random}(a_1' - \alpha(a_1 - a_1'), a_1 + \alpha(a_1 - a_1')) & \text{si } a_1 \geq a_1' \\ \text{random}(a_1 - \alpha(a_1' - a_1), a_1' + \alpha(a_1' - a_1)) & \text{si } a_1' \geq a_1 \end{cases}$$

$$a_2^{t+1} \leftarrow \begin{cases} \text{random}(a_2' - \alpha(a_2 - a_2'), a_2 + \alpha(a_2 - a_2')) & \text{si } a_2 \geq a_2' \\ \text{random}(a_2 - \alpha(a_2' - a_2), a_2' + \alpha(a_2' - a_2)) & \text{si } a_2' \geq a_2 \end{cases}$$

Si tras estas operaciones  $t_1^{t+1}, g_1^{t+1}$  o  $a_1^{t+1}$  son mayores que  $t_2^{t+1}, g_2^{t+1}$  o  $a_2^{t+1}$ , respectivamente, se intercambian los valores para que el elemento que representa el límite superior del intervalo represente el inferior y el elemento que representa el límite inferior pase a representar el límite superior.

Por otro lado, también se controla que los elementos del vector no tomen valores fuera del dominio permitido. En el caso en que ocurre esto, los valores son modificados al mínimo o máximo permitido, según se esté superando el valor mínimo permitido o el máximo permitido, respectivamente.

- *Cruce de Intercambio o Swap*: en este operador de cruce los padres simplemente intercambian elementos de su vector dando lugar a los hijos. Partiendo de dos padres  $c_v^t = (t_1, t_2, g_1, g_2, a_1, a_2)$  y  $c_w^t = (t_1', t_2', g_1', g_2', a_1', a_2')$  se generan dos hijos  $c_v^{t+1}$  y  $c_w^{t+1}$  intercambiando los valores de los tiempos, de las notas o de los accesos. Así, la variable  $r$  puede tomar aleatoriamente los valores 1, 2 ó 3 y en función de este valor, se realizan intercambios de distintas variables:

$$\text{Si } r=1, c_v^{t+1}=(t_1', t_2', g_1, g_2, a_1, a_2) \text{ y } c_w^{t+1} = (t_1, t_2, g_1', g_2', a_1', a_2')$$

$$\text{Si } r=2, c_v^{t+1}=(t_1, t_2, g_1', g_2', a_1, a_2) \text{ y } c_w^{t+1} = (t_1', t_2', g_1, g_2, a_1', a_2')$$

$$\text{Si } r=3, c_v^{t+1}=(t_1, t_2, g_1, g_2, a_1', a_2') \text{ y } c_w^{t+1} = (t_1', t_2', g_1', g_2', a_1, a_2)$$

Los buenos resultados que se obtengan cuando se aplica únicamente este operador de intercambio dependen en gran medida de la población inicial aleatoriamente generada, dado que este operador sólo intercambia elementos pero no los modifica, luego debería utilizarse combinado con otro operador para poder garantizar un buen rendimiento.

La Tabla 6 contiene una muestra de las pruebas realizadas para determinar qué operador de cruce resulta más conveniente en el presente problema. Los datos de entrada corresponden a datos reales de un concurso desarrollado mediante QUESTOURnament en la asignatura Transmisión de Datos de la titulación Ingeniería Técnica de Telecomunicación Especialidad Sistemas de Telecomunicación durante el curso 2008-2009. El algoritmo se ejecuta tres veces para clasificar los desafíos de los tres niveles de dificultad (*fácil*, *asequible* y *difícil*) y la Tabla 6 muestra el *fitness* del mejor cromosoma.

Operador de Cruce	N=5			N=10			N=15			N=20		
	Fácil	Aseq.	Dif.									
Aritmético	0.07	0.150	0.040	0.075	0.327	0.030	0.054	0.083	0.092	0.225	0.076	0.102
Swap	0.096	0.176	0.102	0.261	0.301	0.153	0.161	0.184	0.079	0.401	0.167	0.122
BLX- $\alpha$ ( $\alpha=3$ )	0.279	0.218	0.138	0.564	0.301	0.138	0.657	0.276	0.153	0.631	0.352	0.138
BLX- $\alpha$ ( $\alpha=5$ )	0.290	0.301	0.084	0.657	0.251	0.143	0.580	0.352	0.125	0.631	0.251	0.138

Tabla 6. Fitness del mejor cromosoma para distintos operadores de cruce y distintos números de generaciones N (el tamaño de la población inicial es 200,  $p_c=0.6$  y  $p_m=0.01$ )

Empleando el cruce aritmético no se obtienen resultados satisfactorios incluso aumentando el número de generaciones (el número de veces que itera el algoritmo). Nótese que esta tabla se incluye a modo ilustrativo, para determinar qué operador de cruce emplear se requiere repetir múltiples veces las pruebas, ya que en ocasiones un algoritmo genético puede encontrar un buen resultado como, por ejemplo, el caso del mejor cromosoma encontrado para las preguntas asequibles con N=10 y empleando cruce aritmético (Tabla 6). Un buen resultado aislado se puede deber al azar si, por ejemplo, en la generación aleatoria inicial se genera una solución cercana a una solución óptima.

Mediante el cruce swap se obtienen mejores soluciones que con el operador de cruce aritmético. Sin embargo, como ya se ha comentado, el buen resultado que se obtenga mediante este método depende bastante de la generación de la población inicial. Asimismo, se puede observar cómo al aumentar el número de generaciones no siempre mejora la solución (los mejores resultados se obtienen con N=10). Con el cruce swap se pretende que si, por ejemplo, un cromosoma tiene un alto fitness porque tiene muy bien ajustados los intervalos de tiempo y accesos y otro cromosoma tiene muy bien ajustado el intervalo de nota, su combinación podría dar lugar a un cromosoma mejor, que combine los intervalos que tienen bien ajustados ambos cromosomas padres. Sin embargo también puede suceder que el intercambio dé lugar a un peor ajuste del juego de intervalos. Por ello este operador debería incluir la condición de sólo realizar el intercambio cuando algún descendiente tiene mejor fitness que los padres y no incluir en los descendientes aquellos con peor fitness que el de los padres. En ese caso, el mejor padre pasaría a la descendencia. Aún así, este operador de cruce debería combinarse con otro operador, por ejemplo, el operador BLX- $\alpha$ , para que los valores de los elementos se puedan modificar y no sólo intercambiar.

Se observa en la Tabla 6 que se obtienen los mejores resultados cuando se usa el operador BLX- $\alpha$ , pero no se percibe una mejora importante cuando se establece  $\alpha=0.5$  en vez de  $\alpha=0.3$ , tal y como muchas investigaciones sugieren (Deb 2001).

#### 4.1.1.4.3 Mutación

La mutación se produce raramente en la naturaleza y, de igual manera, la probabilidad de mutación en los algoritmos genéticos suele ser pequeña, típicamente entre 0.001 y 0.01 (Negnevitsky 2005). Este operador contribuye a evitar la convergencia prematura y el estancamiento en soluciones óptimas locales.

Cuando se emplean codificaciones de los cromosomas mediante cadenas de bits, el operador de mutación selecciona aleatoriamente algunos de estos bits y cambia su valor (T. Mitchell 1997). Como en el caso del operador de cruce, hay operadores de mutación específicos para cuando se trabaja con números reales como por ejemplo:

- *Mutación aleatoria.* Existen varias variantes, como la más simple que consiste en crear una solución aleatoriamente a partir de todo el espacio de búsqueda, que sustituye al cromosoma que muta. Otra opción es crear una solución cercana a la solución de partida  $c_i^t = (v_1^t, v_2^t, \dots, v_N^t)$  en que las variables se modifican mediante la fórmula (Deb 2001):

$$v_j^{t+1} = v_j^t + (r_j - 0.5)\Delta_j \quad (13)$$

Siendo  $\Delta_j$  la variación máxima definida permitida en la variable  $j$ .

- *Mutación uniforme:* Si  $c_v^t = (v_1, \dots, v_N)$  es un cromosoma, existe la misma probabilidad para cada elemento  $v_i$  de que se le aplique la operación de mutación, que consiste en sustituir el elemento  $v_i$  por un valor generado aleatoriamente dentro del espacio de dominio correspondiente a esa variable (Michalewicz 1996).

Otros operadores de mutación se pueden encontrar en (Bäck et al. 2000) (Deb 2001) (Michalewicz 1996).

El operador diseñado para el presente algoritmo se basa en la mutación uniforme. Partiendo de un cromosoma  $c_v^t = (t_1, t_2, g_1, g_2, a_1, a_2)$  se genera otro cromosoma  $c_v^{t+1}$  en función del valor de la variable  $r$  que aleatoriamente puede tomar los valores 1, 2 ó 3:

- Si  $r=1$ ,  $c_v^{t+1} = (t_1', t_2', g_1, g_2, a_1, a_2)$ ,  $t_1'$  y  $t_2'$  toman valores aleatorios pertenecientes al dominio de la variable tiempo. En el caso de que  $t_1'$  sea mayor que  $t_2'$ , se intercambian los valores.
- Si  $r=2$ ,  $c_v^{t+1} = (t_1, t_2, g_1', g_2', a_1, a_2)$ ,  $g_1'$  y  $g_2'$  toman valores aleatorios pertenecientes al dominio de la variable nota (grade). En el caso de que  $g_1'$  sea mayor que  $g_2'$ , se intercambian los valores.
- Si  $r=3$ ,  $c_v^{t+1} = (t_1, t_2, g_1, g_2, a_1', a_2')$ ,  $a_1'$  y  $a_2'$  toman valores aleatorios pertenecientes al dominio de la variable número de accesos. En el caso de que  $a_1'$  sea mayor que  $a_2'$ , se intercambian los valores.

#### 4.1.2 Ejecución del algoritmo para el ajuste de sus parámetros

El establecimiento de los parámetros de control suele ser específico del problema (Riekert, Malan, y Engelbrecht 2009) y, por ello, se ha de realizar pruebas para su ajuste. En el apartado previo, que describe los distintos operadores de cruce, ya se han mostrado resultados de pruebas en las que se pretende averiguar qué operador de cruce de los tres diseñados es más adecuado al problema. Finalmente se concluye que el operador BLX- $\alpha$  es el más adecuado.

Además de los operadores genéticos, se debe ajustar también las probabilidades  $p_c$  y  $p_m$ , el tamaño de la población o el número de generaciones.

Para las pruebas y ejemplos mostrados en el presente apartado se han empleado, al igual que en las pruebas de los operadores de cruce, los datos reales correspondientes al concurso desarrollado mediante QUESTOURnament en la asignatura Transmisión de Datos de la titulación Ingeniero Técnico de Telecomunicación Especialidad Sistemas de Telecomunicación durante el curso 2008-2009. Los patrones de respuesta de entrada corresponden a las respuestas dadas por los estudiantes de este curso a 12 desafíos. Inicialmente el profesor clasifica 4 de estos desafíos como fáciles, 3 como asequibles y 5 como difíciles, siendo 134 el número total de respuestas para el nivel fácil, 103 para asequible y 169 para difícil. El algoritmo se ejecuta tres veces para clasificar los desafíos de los tres niveles de dificultad (*fácil*, *asequible* y *difícil*).

Primeramente, se muestra un ejemplo de cómo se ha ajustado el número de generaciones empleando estos datos de entrada. La Tabla 7 muestra el valor del fitness total  $F$ , esto es, la suma del fitness de todos los individuos de la población, y el valor del fitness del mejor individuo  $f$ , según el algoritmo progresa, en las generaciones quinta, décima, quinceava y veinteava. A diferencia de la Tabla 6, en este caso los resultados en las distintas generaciones corresponden a una misma ejecución del algoritmo. La Figura 16 ilustra la evolución del fitness total a lo largo de las generaciones basándose en los resultados mostrados en la Tabla 7. Por otro lado, la Figura 17 muestra la evolución del fitness del mejor individuo de la población a lo largo de las generaciones. Se puede observar que la mejora del fitness en el caso del mejor individuo o cromosoma de la población no es importante si comparamos la generación décima con la quinceava o la veinteava en gran parte de los casos. Por este motivo, se establece como criterio de parada del algoritmo el alcanzar la generación décima, evitando así mayores costes computacionales.

Ejecución		N=5		N=10		N=15		N=20	
		$F$							
1	<i>Fácil</i>	33.3	0.406	82.58	0.631	91.38	0.631	94.36	0.631
	<i>Asequible</i>	30.55	0.327	39.71	0.377	51.09	0.402	52.81	0.377
	<i>Difícil</i>	8.47	0.102	11.04	0.138	17.39	0.153	19.57	0.153
2	<i>Fácil</i>	67.33	0.564	87.31	0.631	93.2	0.657	95.37	0.657
	<i>Asequible</i>	15.08	0.209	25.88	0.352	35.03	0.352	34.55	0.352
	<i>Difícil</i>	15.03	0.143	18.98	0.167	21.85	0.167	22.84	0.167
3	<i>Fácil</i>	64.01	0.541	72.93	0.657	82.75	0.657	85.64	0.657
	<i>Asequible</i>	5.35	0.117	24.93	0.327	48.93	0.352	53.41	0.352
	<i>Difícil</i>	5.50	0.107	12.40	0.112	16.93	0.143	18.12	0.143
4	<i>Fácil</i>	59.83	0.564	80.30	0.631	89.15	0.631	88.50	0.631
	<i>Asequible</i>	5.20	0.117	22.9	0.352	50.35	0.377	51.64	0.377
	<i>Difícil</i>	8.51	0.107	12.5	0.138	16.45	0.153	18.9	0.153

Tabla 7. Evolución del fitness total ( $F$ ) y el fitness del mejor individuo ( $f$ ) a lo largo de las generaciones (el tamaño de la población inicial es 200, el operador de cruce es BLX- $\alpha$  con  $\alpha=0.3$ ,  $p_c=0.6$  y  $p_m=0.01$ )

Finalizar el algoritmo genético tras un número establecido de generaciones es una práctica común, puesto que es difícil determinar cuándo se ha encontrado la mejor solución, ya que el

algoritmo genético es un método de búsqueda estocástica y podría permanecer estable durante un número de generaciones hasta que aparece una mejor solución (Negnevitsky 2005).

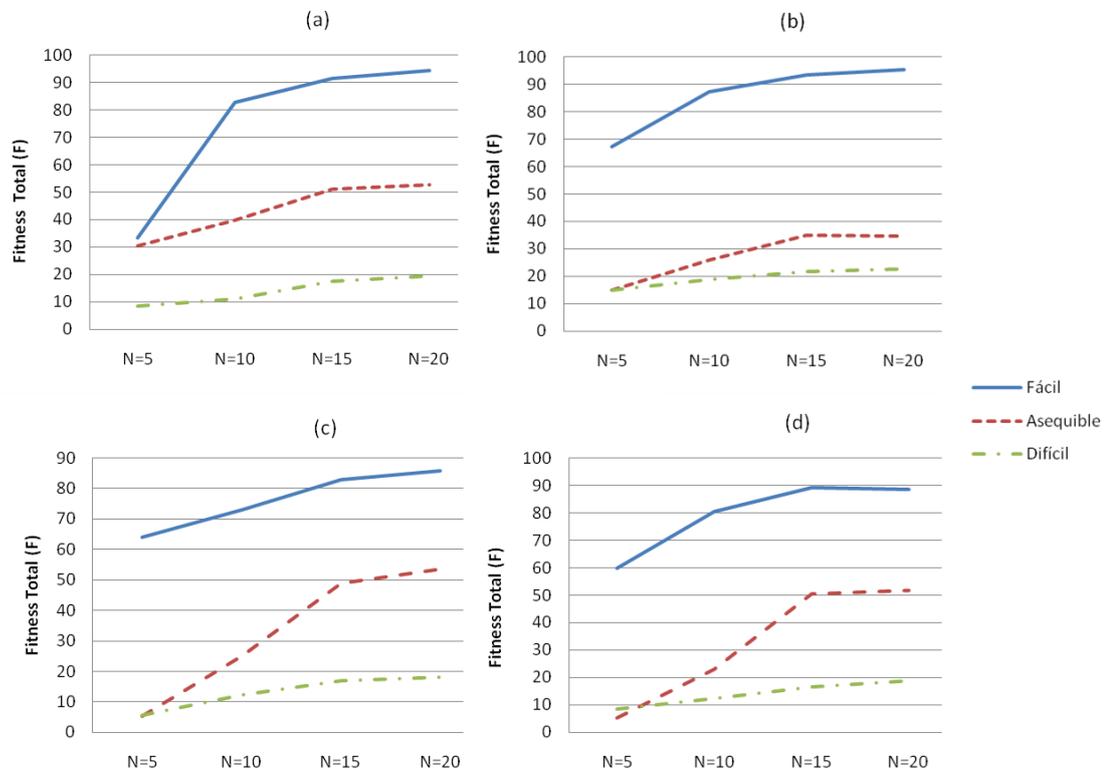


Figura 16. Evolución del fitness total de la población a lo largo de las generaciones en la (a) ejecución 1, (b) ejecución 2, (c) ejecución 3 y (d) ejecución 4.

El resto de parámetros se han ajustado experimentalmente, buscando aquellos valores que optimizan el fitness del mejor individuo. Se resumen a continuación los valores de parámetros y operadores que permiten obtener una mejor solución al problema preservando el coste computacional:

*Probabilidad de cruce  $p_c = 0.6$*

*Probabilidad de mutación  $p_m = 0.01$*

*Operador de cruce: BLX- $\alpha$  ( $\alpha = 0.3$ )*

*Población inicial N: 200 individuos*

*Número de generaciones: 10 generaciones*

La Tabla 8 muestra la mejor solución encontrada en 5 ejecuciones del algoritmo para los tres niveles de dificultad y con estos valores de operadores y parámetros.

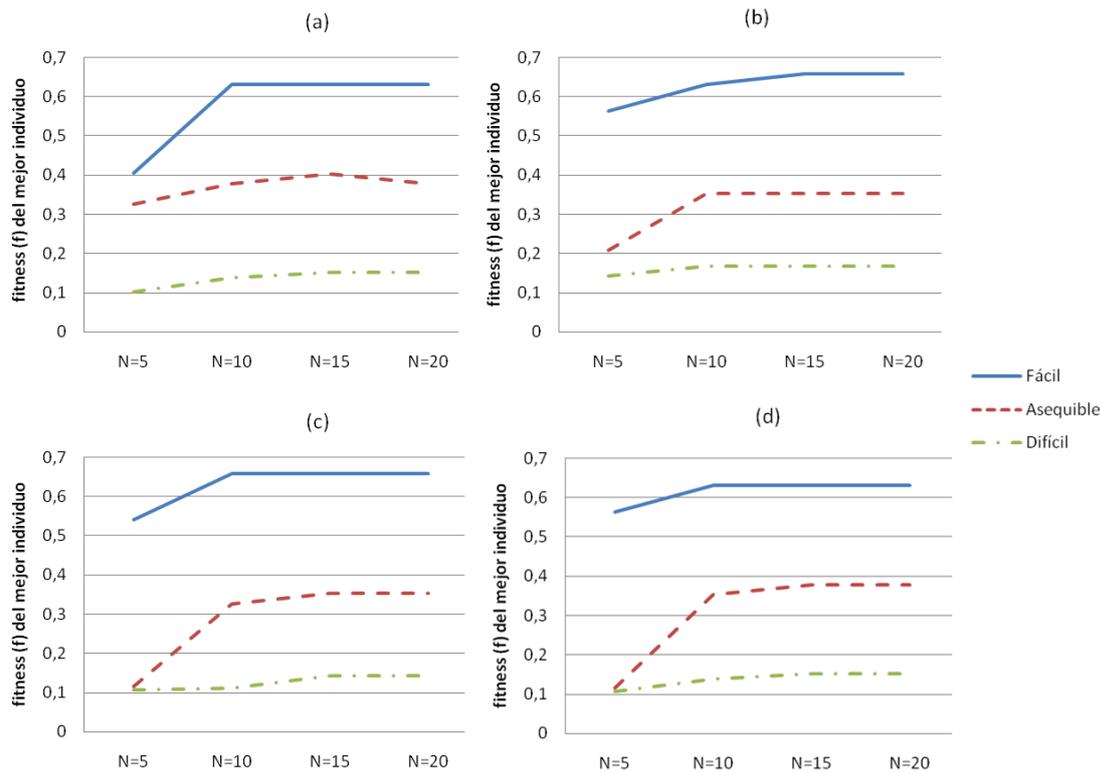


Figura 17. Evolución del fitness del mejor individuo de la población a lo largo de las generaciones en la (a) ejecución 1, (b) ejecución 2, (c) ejecución 3 y (d) ejecución 4.

Nivel fácil				Nivel Asequible				Nivel Difícil			
Tiempo	Nota	Accesos	Fitness	Tiempo	Nota	Accesos	Fitness	Tiempo	Nota	Accesos	Fitness
[8,40]	[80,100]	[1,2]	0.657	[2,33]	[30,50]	[1,2]	0.352	[15,42]	[48,67]	[2,3]	0.138
[7,38]	[80,100]	[1,2]	0.657	[8,34]	[35,55]	[1,2]	0.301	[6,55]	[0,16]	[3,4]	0.132
[8,52]	[81,100]	[1,2]	0.399	[0,29]	[37,50]	[1,2]	0.301	[1,47]	[0,9]	[3,4]	0.155
[2,25]	[80,100]	[1,3]	0.419	[7,38]	[33,50]	[1,2]	0.402	[3,55]	[0,10]	[3,4]	0.167
[4,34]	[85,100]	[1,2]	0.502	[6,36]	[31,50]	[1,2]	0.377	[13,60]	[49,68]	[2,3]	0.132

Tabla 8. Intervalos de *tiempo*, *nota* y *número de accesos* encontrados por el algoritmo genético para cada nivel de dificultad ( $p_c=0.6$ ,  $p_m=0.01$ , operador de cruce BLX-0.3, 200 individuos, 10 generaciones)

Puede observarse en la Tabla 8 que los valores de fitness del mejor cromosoma que encuentra el algoritmo al trabajar con los patrones de nivel de dificultad *fácil* es bastante superior al obtenido con los patrones de nivel de dificultad *asequible* o *difícil*. El motivo es que muchos de los patrones de respuesta del nivel *fácil* siguen un patrón común, mientras que en el resto de casos los valores se encuentran más dispersos a lo largo del dominio y es, por tanto, más difícil encontrar la mejor solución. Este hecho se puede observar a partir de los histogramas de la Figura 18, que representan la frecuencia de los valores de nota en los patrones de respuesta de entrada para el nivel (a) *fácil*, (b) *asequible* y (c) *difícil*. De cualquier manera, nótese que el algoritmo no sólo está teniendo en cuenta el número de patrones cuyo valor de nota está dentro del intervalo definido para nota en la solución sino que simultáneamente debe cumplirse

también que el valor de tiempo y el valor de número de accesos se encuentre dentro de los intervalos de tiempo y número de accesos definidos por la solución.

Como se puede observar en la Tabla 8, se encuentra un problema con el presente algoritmo. Aunque su comportamiento es el esperado, sucede que en algunos casos existen varias soluciones buenas al problema, esto es, las preguntas de un nivel de dificultad dado pueden caracterizarse por varios juegos de intervalos. Por ejemplo, en el caso del nivel de dificultad difícil de este ejemplo, la solución encontrada en la segunda ejecución tiene el mismo fitness que la solución encontrada en la quinta ejecución y, sin embargo, las soluciones son muy distintas, aunque igual de buenas de acuerdo a su fitness. Por ello, el algoritmo genético debe trabajar con múltiples buenas soluciones simultáneamente, y mantenerlas y explotarlas a lo largo de las generaciones.

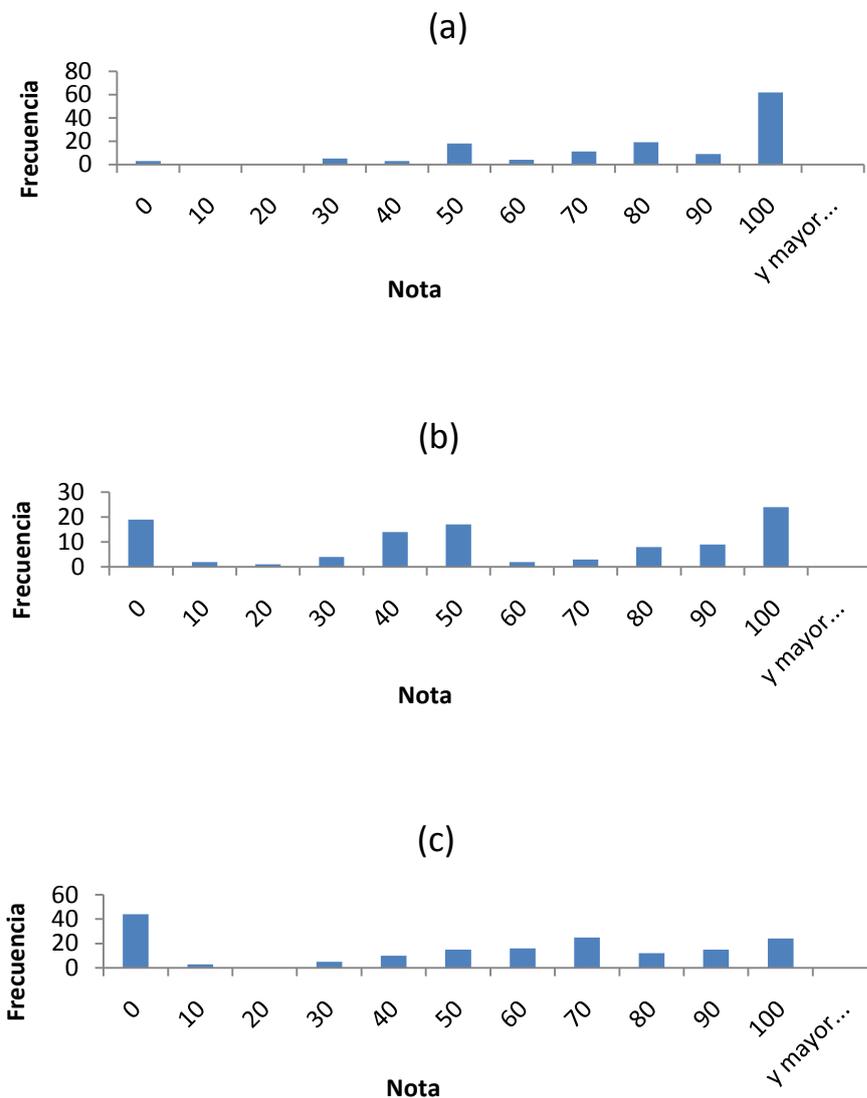


Figura 18. Histogramas de la variable *nota* de los patrones de respuesta correspondientes al nivel (a) fácil (b) asequible y (c) difícil

### 4.1.3 Algoritmo genético mejorado con técnicas de diversidad

*Los contenidos del presente apartado han sido parcialmente publicados en (E. Verdú, M.J. Verdú, et al. 2010).*

Como se acaba de exponer, tras realizar una serie de experimentos con una primera versión del algoritmo, se encuentra que las preguntas correspondientes a un mismo nivel de dificultad se pueden caracterizar por más de un juego de intervalos. Los distintos comportamientos y características de los alumnos dan lugar a diferentes perfiles de respuestas. Por ello el algoritmo genético debe encontrar no sólo la mejor solución al problema sino una serie de soluciones diversas, tal y como sucede con muchos problemas de optimización reales, que requieren la localización de soluciones múltiples, ya sean locales o globales (Sareni y Krahenbuhl 1998). Además, implementar una solución que promueva la diversidad de soluciones, evitará un problema tradicional de los algoritmos genéticos, que es la convergencia prematura a una solución local que no es la mejor. El problema de la convergencia prematura puede darse cuando se pierde diversidad y los cromosomas de la población son muy similares ya en iteraciones tempranas del algoritmo, con lo que la población casi no evoluciona y la búsqueda se estanca prematuramente (Shimodaira 2001) (Lozano, F Herrera, y Cano 2008). La baja diversidad es frecuentemente la razón de rendimientos pobres en los algoritmos evolutivos (Burke, Gustafson, y Kendall 2004). Reproduciendo individuos no muy buenos, se puede escapar de óptimos locales y mejorar los valores de fitness final. Por ello, la prevención de la convergencia prematura promoviendo la diversidad debe considerarse durante el diseño del algoritmo genético (Chaiyaratana, Piroonratana, y Sangkawelert 2006).

Un método sencillo para encontrar diversas soluciones consiste en ejecutar el algoritmo genético varias veces y tomar las distintas soluciones que el algoritmo proporciona en cada ejecución (del Jesus et al. 2007). El problema de este método es que se debe ejecutar el algoritmo varias veces con el consiguiente coste computacional que supone. Para evitar realizar distintas ejecuciones que den lugar al mismo resultado, se puede incluir una penalización en la función de fitness de tal manera que aquellos cromosomas similares a soluciones ya encontradas en ejecuciones previas vean disminuido su fitness. Para promover y controlar la diversidad es común usar operadores de cruce y de mutación, así como estrategias de reemplazo y selección poco comunes (Burke, Gustafson, y Kendall 2004). Algunas técnicas específicas son: inyección de cromosomas nuevos en la población (Chaiyaratana, Piroonratana, y Sangkawelert 2006) (P.-C. Chang, W.-H. Huang, y Ting 2010) (Sultan et al. 2006), marcado de individuos con etiquetas de tal manera que no se permita etiquetas duplicadas en la población (Ronald 1995), control adaptativo de las tasas de cruce y de mutación (Zhu 2003), uso de redes neuronales artificiales para distribuir uniformemente soluciones que están convergiendo en posibles óptimos locales (Kobayashi, Hiroyasu, y Miki 2007), métodos de nichos (Rudolph y Ziegenhirt 2000), sustitución de individuos duplicados por individuos nuevos generados aleatoriamente y, para el resto de individuos, aplicación de una probabilidad de supervivencia, que depende de la similitud entre los individuos y el mejor individuo (Chaiyaratana, Piroonratana, y Sangkawelert 2006), etc. Aumentar las probabilidades de mutación y de cruce puede mejorar la diversidad; de hecho el objetivo del operador de mutación es mejorar la diversidad y facilitar el escapar de óptimos locales. Sin embargo, para el presente algoritmo no se encontraron mejoras en la diversidad de la población al realizar pruebas con las probabilidades de cruce y de mutación igual a 0.8 y 0.1, respectivamente. Por ello, se tuvo que analizar otros métodos para garantizar diversidad de buenas soluciones.

En la solución propuesta, para lograr diversidad de soluciones, el algoritmo incluye una estrategia de nichos denominada *sharing*. Los métodos de nichos son una extensión a los algoritmos genéticos que permiten que el algoritmo genético encuentre y mantenga diversas soluciones dentro de la población a lo largo de las generaciones. Además, permiten evitar la convergencia prematura en caso de que sólo se requiera una solución (Mengshoel y Goldberg 2008). Como su propio nombre indica los métodos de nichos pretenden formar y mantener poblaciones estables, denominadas nichos, dentro de la población (Rudolph y Ziegenhirt 2000). Hay dos métodos de nichos empleados frecuentemente que se denominan *fitness sharing* (*compartición de fitness*) y *crowding* (*apiñamiento*).

En los métodos de *crowding* los individuos similares compiten entre sí mediante torneos locales y los mejores individuos tienden a sustituir a los peores (Rudolph y Ziegenhirt 2000) (Mengshoel y Goldberg 2008). Existen distintos métodos de *crowding* como el denominado *crowding* determinístico, que es simple, rápido y no requiere parámetros adicionales a mayores (Mengshoel y Goldberg 2008). En concreto, en este método compiten padres e hijos y cada hijo reemplaza al padre más similar si tiene mayor *fitness* (Sareni y Krahenbuhl 1998). Una variante de este método el *crowding* probabilístico, en el que la regla de reemplazo es probabilística y no determinística (Mengshoel y Goldberg 2008).

Por otro lado, mediante la técnica *sharing* los cromosomas ven reducido su *fitness* en función de su similitud al resto de cromosomas (Rudolph y Ziegenhirt 2000). Una ventaja del método *sharing*, siendo éste un motivo por el que este método ha sido seleccionado junto a los buenos resultados obtenidos, es que es sencillo de implementar, pues se trata de una operación de escalado que sólo afecta a la etapa de cálculo de *fitness* (Mahfoud 2000). Por contrapartida, el principal inconveniente de usar *sharing* es el tiempo adicional requerido para calcular los *fitness* compartidos (Rudolph y Ziegenhirt 2000), aunque muchos de estos métodos que buscan diversidad incluyen cálculos adicionales como las operaciones para obtener las distancias entre individuos y así agruparlos en nichos o en parejas para que compitan en torneos.

En la solución propuesta, además de la técnica *sharing*, se ha establecido otro mecanismo adicional para aumentar la diversidad. Como en (Chaiyaratana, Piroonratana, y Sangkawelert 2006) (Sultan et al. 2006) (P.-C. Chang, W.-H. Huang, y Ting 2010), el algoritmo propuesto inserta nuevos individuos en la población que reemplazan a los individuos similares. Así, mediante la inyección de nuevos cromosomas se promueve la búsqueda por todo el espacio de soluciones. Se trata de una estrategia casi elitista ya que el algoritmo conserva el mejor individuo de un nicho y elimina aquellos individuos que están demasiado cercanos a él. El elitismo consiste en conservar el mejor individuo garantizando que pasa a la siguiente generación para evitar la pérdida de soluciones óptimas (Deb 2001) (López-Pujalte, Guerrero-Bote, y de Moya-Anegón 2003). En este caso no se garantiza que el mejor individuo pase a la siguiente generación pero se eliminan aquellos elementos demasiado similares a él con lo cual su *fitness* se ve menos penalizado en la posterior aplicación de *sharing* y tiene, por tanto, mayores posibilidades de contribuir a la siguiente generación. Mediante la técnica *sharing*, los individuos similares comparten el *fitness* y, por tanto, se reduce el *fitness* de los individuos ubicados en regiones altamente pobladas. La solución, por tanto, introduce distintos elementos para promover la diversidad siendo una combinación de estrategia cuasi-elitista, junto con el método *sharing* y la inyección de nuevos individuos en la población.

Dado que las distintas operaciones tienen en cuenta la similitud de individuos, es preciso definir una manera de medir esta similitud. Esta medida puede tener en cuenta el genotipo (estructura del cromosoma) o el fenotipo (*fitness* del cromosoma).

En este caso, la medida es genotípica y basada en la distancia Euclídea entre dos vectores (Hiroyasu et al. 2008) (Taplin y Qiu 2001). Así, la distancia entre dos cromosomas  $c_v = (t_1, t_2, g_1, g_2, a_1, a_2)$  y  $c_w = (t_1', t_2', g_1', g_2', a_1', a_2')$  se calcula con la siguiente fórmula:

$$D(c_v, c_w) = \sqrt{\left(\frac{t_1 - t_1'}{T_{max}}\right)^2 + \left(\frac{t_2 - t_2'}{T_{max}}\right)^2 + \left(\frac{g_1 - g_1'}{G_{max}}\right)^2 + \left(\frac{g_2 - g_2'}{G_{max}}\right)^2 + \left(\frac{a_1 - a_1'}{A_{max}}\right)^2 + \left(\frac{a_2 - a_2'}{A_{max}}\right)^2} \quad (14)$$

siendo  $T_{max}$ ,  $G_{max}$  y  $A_{max}$  los valores máximos de tiempo, nota y accesos respectivamente, incluidos para normalizar.

En la Figura 19 se muestran las distintas fases del algoritmo genético. Se incluyen las etapas típicas de un algoritmo genético tales como la generación de la población inicial, la selección, el cruce y la mutación, la evaluación de *fitness* y, por otro lado, aquellas etapas más específicas del problema de la diversidad de soluciones como la asignación de nichos o el escalado de *fitness*.

Durante la fase de asignación de nichos se crean los nichos que contienen cromosomas similares. La similitud se determina con la anteriormente explicada medida de distancia y se necesita definir un radio de nicho  $r_n$  para determinar aquellos individuos que por su cercanía pertenecen a un mismo nicho. A continuación, en la siguiente fase, los cromosomas más cercanos al de mayor *fitness* en su nicho son borrados y reemplazados por individuos aleatoriamente generados, incrementándose así la diversidad y evitándose la convergencia prematura. Por ello se define también otro radio  $r_c$  que determina qué individuos están demasiado cercanos al mejor individuo de su nicho. El valor de ambos radios ha sido establecido experimentalmente.

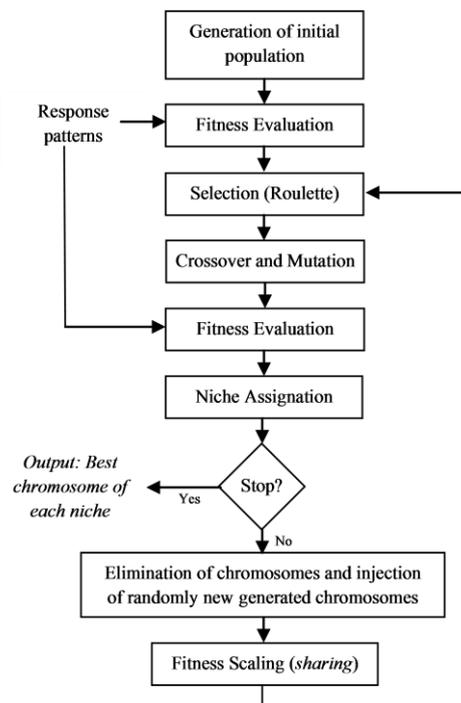


Figura 19. Fases del Algoritmo Genético (E. Verdú, M. J. Verdú, et al. 2010)

En la fase siguiente de escalado de fitness, el valor de fitness del resto de individuos del nicho se divide por una cantidad que mide la similitud entre ese individuo y el resto de individuos de la población, empleando la función de sharing ampliamente utilizada indicada a continuación (Sareni y Krahenbuhl 1998):

$$f'(C_i) = \frac{f(C_i)}{\sum_{j=1}^N sh(d(C_i, C_j))} \quad (15)$$

Donde  $f(C_i)$  es el fitness del cromosoma  $C_i$ ,  $N$  es el tamaño de la población,  $d(C_i, C_j)$  es la distancia entre el cromosoma  $C_i$  y el cromosoma  $C_j$  y  $sh(d)$  es la función de sharing, que mide la similitud entre dos elementos y se calcula empleando la siguiente fórmula:

$$sh(d) = \begin{cases} 1 - \left(\frac{d}{r_n}\right)^\alpha, & \text{si } d < r_n \\ 0, & \text{si } d \geq r_n \end{cases} \quad (16)$$

donde  $\alpha$  es una constante que habitualmente toma el valor 1 (Sareni y Krahenbuhl 1998).

La técnica *sharing* es un mecanismo de escalado de fitness, que se aplica justo antes de la etapa de selección. La Figura 20 muestra el efecto del mecanismo *sharing* sobre la selección de individuos en la décima generación. Como previamente se ha indicado, el método de selección emula el juego de la ruleta y aquellos individuos o cromosomas con mayor fitness ocupan una mayor porción de la ruleta con lo que tienen una mayor probabilidad de que en cada tirada de la ruleta puedan ser seleccionados.

La distribución de los nichos en la rueda de la ruleta antes de aplicar el mecanismo de *sharing* se puede observar en la Figura 20 (a). El número que se indica encima de los porcentajes identifica un nicho. Así, cada porción de ruleta representada en la figura agrupa en realidad a las distintas porciones que les corresponde a los individuos de un mismo nicho, que son, por tanto, similares. Los elementos del nicho 1 tienen mayor fitness y ocupan una gran porción de la ruleta. Si no se aplicase *sharing* el resultado es que la mayoría de los elementos seleccionados para cruce pertenecen al nicho 1, perdiéndose diversidad y corriendo el riesgo de una convergencia prematura. Se puede observar que tras aplicar el mecanismo de *sharing*, las porciones de la ruleta quedan distribuidas más equitativamente (zona (b) de la figura) y la porción de ruleta que ocupan los nichos mayores (nichos 1 y 6) se reduce a favor del resto de nichos. La operación de escalado de fitness ha reducido el fitness en aquellos cromosomas parecidos a muchos otros individuos. El modo natural de operar de un algoritmo genético (zona (a) de la figura) promueve la repetición de los individuos de mayor fitness y esto es deseable que ocurra en últimas iteraciones del algoritmo, cuando se busca una única solución, pero no en fases tempranas donde, como ya se ha comentado, se puede encontrar una solución óptima local y el algoritmo podría estancarse repitiendo individuos cercanos a esta solución local. Eliminando los individuos que se encuentran muy cercanos a los mejores individuos dentro de su nicho y sustituyéndolos por nuevos individuos se evita la convergencia prematura y se promueve la diversidad. Si, además, para el resto de individuos del nicho se aplica el escalado de fitness de tal manera que cromosomas similares disminuyen su fitness, compartiendo fitness, mientras que los cromosomas aislados mantienen su fitness, se da el caso de la zona (b) de la figura. El resultado que se muestra refleja claramente el efecto de este mecanismo, que promueve la diversidad y el mantenimiento de distintas soluciones buenas a lo largo de las distintas ejecuciones del algoritmo, mejorándose la búsqueda en todo el espacio de soluciones y la localización de individuos en diferentes regiones óptimas.

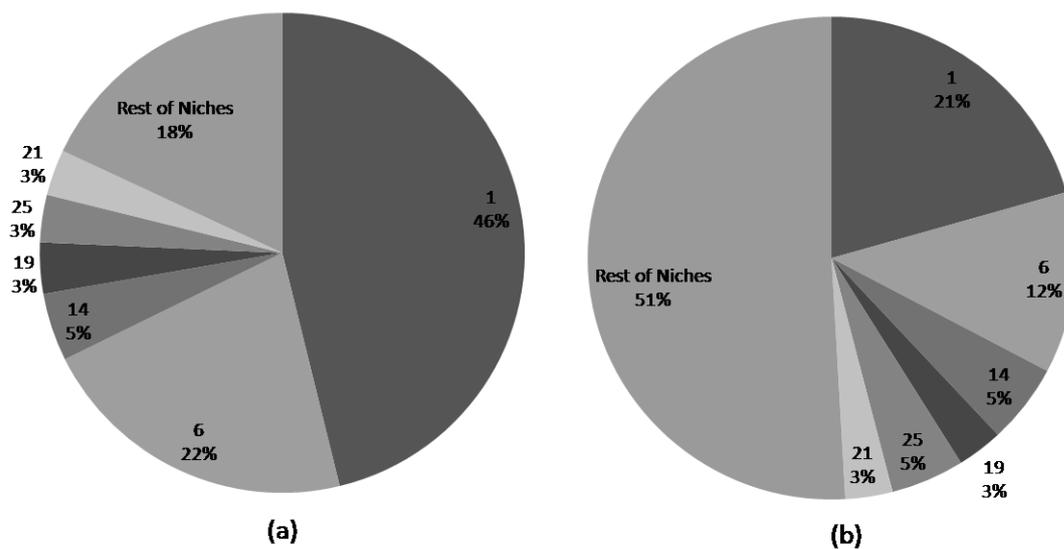


Figura 20. Distribución de los nichos en la rueda de la ruleta de acuerdo al (a) fitness y al (b) fitness compartido (o shared fitness)

Tras un número de generaciones, el algoritmo da como resultado varios cromosomas de alto fitness. En concreto, el algoritmo devuelve el mejor cromosoma de la población y aquellos mejores cromosomas de otros nichos, cuyo fitness es superior a la mitad del fitness del mejor cromosoma (este valor se ha escogido para evitar obtener muchas soluciones diversas, lo que hace más compleja la etapa de post-procesamiento de la salida del algoritmo genético). Dado que los cromosomas que el algoritmo genético entrega pertenecen a distintos nichos, se garantiza que son buenas soluciones diferentes.

En la Figura 21 se muestran los distintos juegos de intervalos obtenidos por el algoritmo genético mejorado con técnicas de diversidad en los distintos niveles de dificultad de las preguntas relativas a la asignatura Transmisión de Datos cuyos datos se han descrito en el apartado 4.1.2 del presente capítulo. Se puede observar cómo las preguntas difíciles se caracterizan por dos juegos de intervalos principales. Muchas respuestas tienen nota muy baja mientras que también hay un importante número de respuestas que tienen una nota media. Estos conjuntos de respuestas con diferentes características podrían responder a diferentes perfiles de estudiantes y diferentes niveles de conocimiento. Para las preguntas asequibles se encuentra tres juegos de intervalos que las caracterizan. Uno relativo a las notas más bajas, otro de notas medias y, por último, otro de notas altas pero tiempos altos también. Esto podría indicar que para resolver correctamente los desafíos asequibles, los alumnos tienen que dedicar más tiempo que para resolver los desafíos fáciles. Estos últimos vienen caracterizados por un único juego de intervalos, correspondiente a notas altas y tiempos bajos. Como cabe esperar, el juego de intervalos con mayor fitness se va desplazando hacia las notas más bajas, conforme el nivel de dificultad aumenta, tal y como se puede ver en la Figura 21. Más aún, cabe preguntarse si cada uno de los tres diferentes grupos que se detectan en el nivel asequible podría ser representativo de los diferentes niveles de conocimiento de los alumnos.

Una vez se resuelve el problema de obtener diversas soluciones, la salida del algoritmo genético se ajusta mediante un algoritmo de búsqueda local de tipo *escalada* o *máximo gradiente* con el fin de mejorar las soluciones encontradas (etapa de explotación).

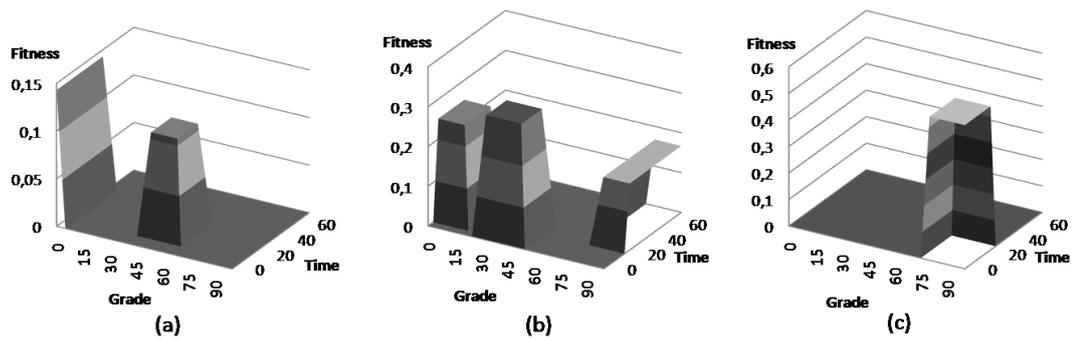


Figura 21. Fitness de los intervalos de tiempo (*time*) y nota (*grade*) para los niveles de dificultad (a) difícil, (b) asequible y (c) fácil. (E. Verdú, M.J. Verdú, et al. 2010)

#### 4.1.4 Explotación de la solución

La búsqueda en el espacio de posibles soluciones requiere el equilibrio de dos objetivos: explorar el espacio de búsqueda (exploración) y optimizar las mejores soluciones encontradas (explotación). Los algoritmos de búsqueda local permiten explotar la mejor solución encontrada para mejorarla pero no realizan una exploración por todo el espacio de búsqueda. Por otro lado una búsqueda aleatoria explora el espacio de búsqueda pero no optimiza las buenas soluciones. Los algoritmos genéticos consiguen un equilibrio entre estos dos aspectos, exploración y explotación (Michalewicz 1996). Sin embargo, un algoritmo genético que emplea la técnica *sharing* no converge hacia la solución óptima de la misma manera que un algoritmo genético común, ya que la etapa de explotación está afectada por la reducción de fitness que sufren los cromosomas similares. Una manera de lograr la convergencia hacia la mejor solución en cada uno de los nichos es ejecutar un algoritmo de escalada (*hillclimbing*), por ejemplo, después del algoritmo genético (Rudolph y Ziegenhirt 2000). Aun no empleando técnicas como *sharing* el resultado de la etapa de explotación del algoritmo genético puede ser optimizado aplicando un algoritmo de búsqueda local cuyo punto de partida sea la solución encontrada por el algoritmo genético tal y como se realiza en los sistemas descritos en (H.-S. Hwang 1998) (C Romero et al. 2009). El objetivo es en muchos casos aumentar la generalidad de la regla extraída (C Romero et al. 2009).

Otro método para mejorar la etapa de explotación es el método de especies (Deb 2001), que, en la ejecución del algoritmo genético, restringe el cruce a individuos similares. De esta manera, la técnica de nichos encuentra y mantiene subpoblaciones de soluciones alrededor de múltiples óptimos y la técnica de especies realiza una búsqueda paralela en cada óptimo para encontrar múltiples soluciones óptimas. Se ha preferido emplear una solución de búsqueda local en vez de este método de especies, ya que no se puede predecir que el cruce de dos buenas soluciones vaya a dar lugar a otra buena solución. Asimismo, aunque se eliminan individuos muy similares al óptimo y se disminuye el fitness de los individuos en función de su similitud con el resto de cromosomas, estos individuos, si parten de un alto fitness, siguen manteniendo un peso importante en la ruleta para ser seleccionados y cruzados algunos de ellos entre sí (ver Figura 20), mejorando la explotación. De cualquier manera, la inclusión de una etapa posterior de búsqueda local permite obtener soluciones más genéricas y optimizar las soluciones encontradas.

Las técnicas de búsqueda local o de mejora iterativa consisten en partir de un estado que representa una solución al problema y realizar pequeños cambios en la configuración de este estado buscando mejoras en la solución (M.C. Rodríguez et al. 2008).

Los algoritmos de búsqueda local parten de un punto del espacio de soluciones posibles y en cada iteración calculan un conjunto de soluciones vecinas mediante una regla de vecindad. Cada una de estas soluciones es evaluada y se selecciona la mejor solución vecina, reemplazando a la actual si es mejor o si cumple otro criterio que se especifique. El proceso se repite hasta cumplir con un criterio de finalización, que suele ser llegar a un número determinado de iteraciones o que no se produzcan mejoras en la solución en las últimas iteraciones (Michalewicz 1996) (M.C. Rodríguez et al. 2008).

Hay distintas versiones de este tipo de algoritmos que difieren en el modo en que una nueva solución vecina es seleccionada para ser comparada con la solución actual (Michalewicz, 1996), siendo de uso muy extendido los algoritmos de escalada o máximo gradiente, la búsqueda tabú y el temple simulado (o *simulated annealing*) (M.C. Rodríguez et al. 2008).

En los algoritmos de escalada o máximo gradiente se selecciona una solución vecina siempre que sea mejor o igual que la actual. Estos algoritmos presentan problemas en regiones planas, en que los vecinos son igual de buenos que la solución actual, o con pendientes muy suaves en dirección del óptimo global, cuando es fácil desviarse hacia los lados de la cresta (M.C. Rodríguez et al. 2008). Asimismo, dado que se trata de un algoritmo de búsqueda local, no permite encontrar un óptimo global, sino sólo óptimos locales que dependen de la selección del punto de partida (Michalewicz 1996). Se puede ejecutar el algoritmo desde distintos puntos de partida en el espacio de posibles soluciones, que no tienen por qué seleccionarse aleatoriamente sino en función de la solución obtenida en ejecuciones previas, para obtener múltiples posibles soluciones al problema, pero aún así puede ser complicado encontrar el óptimo global (Michalewicz 1996) (M.C. Rodríguez et al. 2008).

Los algoritmos de temple simulado, sin embargo, permiten salir de óptimos locales ya que sí es posible con cierta probabilidad el paso a una solución peor. Primero se selecciona aleatoriamente una solución vecina y, si es mejor que la actual, reemplaza a la solución actual. En caso de que no lo sea, la nueva solución se acepta con una cierta probabilidad que depende de dos parámetros: la temperatura y el incremento de energía. Al principio de la búsqueda la temperatura tiene un valor alto, así la probabilidad de aceptar una solución peor a la actual es alta. Según progresan las iteraciones la temperatura disminuye y, con ello, la probabilidad de aceptar una solución que empeora la actual. El criterio de terminación suele ser que la temperatura alcance un determinado valor suficientemente pequeño (M.C. Rodríguez et al. 2008). Así, en el temple simulado las soluciones no dependen del punto de partida y normalmente se acercan al punto óptimo (Michalewicz 1996). El problema de este método es el ajuste de los parámetros que depende del problema y por ello hay que realizarlo de manera experimental (M.C. Rodríguez et al. 2008).

Por otro lado, el método de búsqueda tabú memoriza las transiciones que dieron lugar a las últimas soluciones y evita la generación de algunos vecinos, no empleando estas transiciones memorizadas que se consideran tabú, pudiendo establecerse excepciones. Como en el método de temple simulado, el inconveniente es la necesidad de ajustar parámetros como el tamaño de la lista tabú, la elección de las transiciones que se deben registrar o el criterio de aspiración, que permite establecer excepciones a la lista tabú (M.C. Rodríguez et al. 2008).

En la solución propuesta, el algoritmo genético ya está proporcionando una solución global buena al problema y el objetivo es optimizarla y generalizarla, con lo que el problema presente en los algoritmos de máximo gradiente, que no son capaces de hacer búsquedas globales, no es importante. Por ello, se diseña una solución basada en el funcionamiento de los algoritmos de

escalada o máximo gradiente buscando disminuir además el coste computacional de esta etapa, al igual que en el sistema descrito en (del Jesus et al. 2007) (C Romero et al. 2009), que post-procesan la salida del algoritmo genético con un algoritmo de escalada.

Como se ha mencionado previamente, en los algoritmos de búsqueda local, se evalúa cada una de las soluciones vecinas. Esto supone un alto coste computacional cuando existe un gran número de soluciones vecinas o si la evaluación de cada solución es costosa (Rodríguez, 2008). Las soluciones con las que se está trabajando tienen 6 parámetros  $[t1, t2, g1, g2, a1, a2]$  y hay un buen número de posibles combinaciones de soluciones vecinas que además hay que evaluar con una función de evaluación (función de fitness) relativamente costosa, especialmente si el número de patrones de respuesta es grande. Por este motivo, se desestima el evaluar todas las soluciones vecinas. Analizando las salidas del algoritmo genético, se observa que el parámetro nota o *grade* es el parámetro que discrimina mejor y el más controlado. Por ello, se va a optimizar primero los valores relativos a la nota, luego los del tiempo y, finalmente, los del número de accesos. Así inicialmente se dejan fijos los parámetros relativos a tiempo y a número de accesos, y se trabaja con los vecinos que resultan de modificar los valores de nota  $g1$  y  $g2$ . De esta manera no se puede asegurar que se va a obtener la mejor solución pues no se están teniendo en cuenta a todos los vecinos simultáneamente pero se obtiene un equilibrio en lo referente a la calidad de la solución y al coste computacional.

En el algoritmo diseñado se busca una mejor solución a la encontrada por el algoritmo genético, aumentando la amplitud de los intervalos encontrados por el algoritmo genético, si esa ampliación supone una mejora del soporte no pequeña. Se pretende encontrar soluciones con mayor soporte pero se sigue controlando que la amplitud de los intervalos no sea tan amplia que abarque gran parte del dominio y por eso la solución pierda el interés. Por ello, la función de evaluación empleada por el algoritmo de escalada es similar a la función de fitness del algoritmo genético, aunque los pesos que penalizan los intervalos según su amplitud son menores para así tratar de obtener una regla más general. Por tanto, el algoritmo amplía los intervalos de las soluciones propuestas por el algoritmo genético si se obtienen mejores soluciones (ni disminuye ni “traslada” intervalos, los nuevos intervalos siempre contendrán a los intervalos de partida).

El resultado es un conjunto de intervalos que caracteriza cada nivel de dificultad y a partir del cual se debe generar el modelo difuso.

## 4.2 Generación del modelo difuso

El Generador del Modelo Difuso recoge los juegos de intervalos que el algoritmo genético encuentra y define, a partir de los mismos, las reglas y conjuntos difusos que modelan las respuestas correspondientes a los distintos niveles de dificultad. Para la implementación del sistema de control difuso se deben seguir los pasos definidos en cualquier sistema difuso.

En un sistema de control difuso, asumiendo que hay un problema particular a resolver, lo primero que hay que hacer es determinar los conjuntos difusos que describen el problema para después, establecer la base de reglas del sistema. Posteriormente, con estas reglas y a partir de los conjuntos difusos de las variables de entrada, se infieren los conjuntos difusos de las variables de salida; asimismo, luego éstos deben combinarse de alguna forma para, en último lugar, obtener conclusiones precisas. Hay variaciones en esta aproximación pero en esencia, un sistema difuso está compuesto por los elementos que se muestran en la Figura 22.

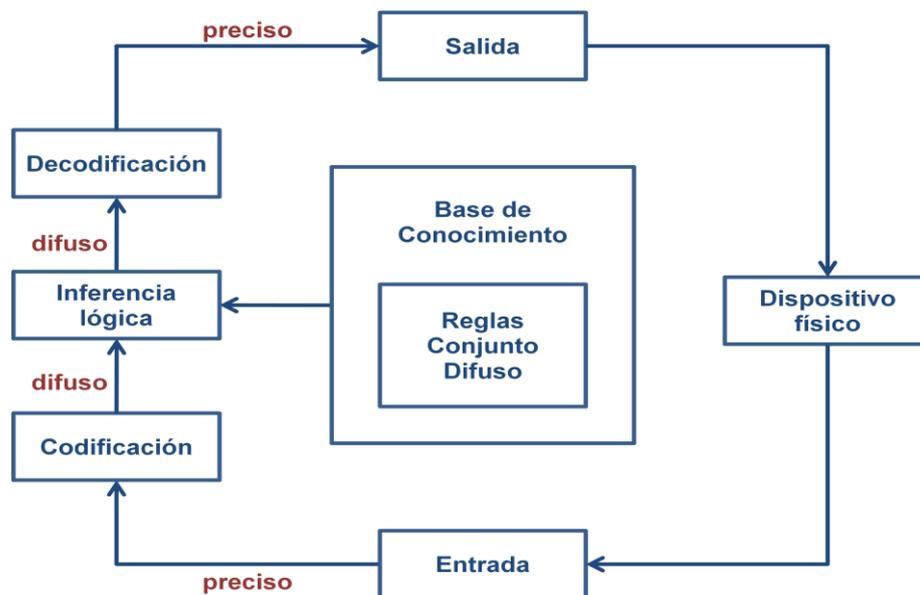


Figura 22. Sistema de control difuso básico

Antes de pasar a detallar cada uno de estos elementos es importante comentar, dado que esto va a afectar a su desarrollo, que el sistema se ha simulado con Fuzzy Logic Toolbox de Matlab.

#### 4.2.1 Variables y funciones de pertenencia

A la hora de establecer un sistema de control difuso el primer paso es definir la gramática del sistema. En concreto, como se ha visto ya anteriormente, se tomarán como variables de entrada el tiempo, la nota (*grade*) y el número de accesos y como variable de salida el nivel de dificultad (*difficulty*). Además, cada variable difusa tiene asociado un conjunto de términos difusos que también hay que especificar. Así, una vez definida la gramática, el siguiente paso es establecer las funciones “ser-miembro-de” o funciones de pertenencia correspondientes a cada uno de estos términos con la forma y posición adecuada. Aquí, en todos los casos, a la hora de describir los conjuntos difusos se van a considerar funciones de pertenencia trapezoidales y triangulares; si bien con diferentes simetrías.

El uso de este tipo de funciones viene justificado porque, como se señala en (Zimmerman 1991), estas funciones además de ser las más simples (están formadas mediante el uso de líneas rectas) y populares, son muy adecuadas para trabajar en tiempo real y de hecho, son las que se eligen normalmente en los sistemas de control. Además de simplificar el proceso de computación, frecuentemente proporcionan una representación adecuada del conocimiento del experto (Negnevitsky 2005).

En este sentido, es importante señalar que, mientras los conceptos y funciones de pertenencia de las variables de entrada serán diferentes en función de cada aplicación particular, las funciones de pertenencia de la variable de salida serán siempre las mismas.

En este sistema, los conjuntos difusos y las funciones de pertenencia se definen a partir de los resultados obtenidos tras la aplicación del algoritmo genético. Las funciones de pertenencia trapezoidales se pueden expresar con cuatro parámetros  $[a, b, c, d]$ . La Figura 23 muestra cómo se crean dos funciones trapezoidales adyacentes a partir de los intervalos de notas (*grade*) obtenidos por el algoritmo genético  $[g_1, g_2]$  y  $[g_1', g_2']$ . Siendo  $[g_1, g_2]$  y  $[g_1', g_2']$  dos intervalos adyacentes, los parámetros  $c, d, a'$  y  $b'$ , que definen las pendientes superpuestas de sus respectivos conjuntos difusos trapezoidales, se calculan empleando las siguientes fórmulas:

$$b' = g_1' + \frac{1}{4}w_n \left(1 - \frac{w_g}{G_{max}}\right) \quad (17)$$

$$c = g_2 - \frac{1}{4}w_n \left(1 - \frac{w_g}{G_{max}}\right) \quad (18)$$

$$a' = c \quad (19)$$

$$d = b' \quad (20)$$

donde  $G_{max}$  es la nota máxima,  $w_n$  es la amplitud del intervalo más estrecho de los dos intervalos adyacentes, y  $w_g$  es la amplitud del hueco entre los dos intervalos. Estos dos últimos parámetros se obtienen a partir de las siguientes fórmulas:

$$w_n = \min(g_2 - g_1, g_2' - g_1') \quad (21)$$

$$w_g = g_1' - g_2 \quad (22)$$

Para definir los valores de  $a$ ,  $b$ ,  $c'$  y  $d'$  se necesitaría conocer datos de los intervalos anteriores y posteriores a los mostrados en la Figura 23. Cuando no hay un intervalo previo, la función de pertenencia toma el valor 1 hasta  $x=0$ ; mientras que, cuando no existe un intervalo posterior, la función de pertenencia toma el valor 1 hasta  $G_{max}$ .

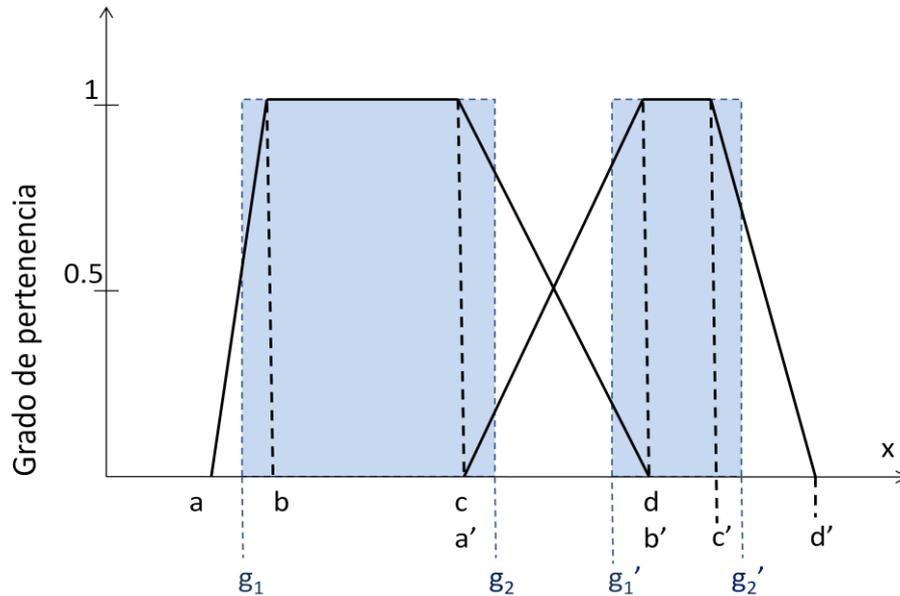


Figura 23. Funciones de pertenencia trapezoidales generadas a partir de dos intervalos adyacentes.

Puede suceder que haya solapamientos en los intervalos encontrados por el algoritmo genético. En estos casos, la región solapada dará lugar a un nuevo conjunto difuso. Así, si el algoritmo genético devuelve dos intervalos solapados  $[g_1, g_2]$  y  $[g_1', g_2']$ , estos son reemplazados por otros nuevos, los cuales serán utilizados a la hora de generar los conjuntos difusos, de acuerdo a los dos casos expuestos a continuación:

- Si  $g_1 < g_1' < g_2$  y  $g_2 < g_2'$ , los intervalos  $[g_1, g_2]$  y  $[g_1', g_2']$  son sustituidos por los intervalos  $[g_1, g_1']$ ,  $[g_1', g_2]$  y  $[g_2, g_2']$  (ver Figura 24).
- Si  $g_1 < g_1' < g_2$  y  $g_2' < g_2$  (un intervalo está contenido en otro), los intervalos  $[g_1, g_2]$  y  $[g_1', g_2']$  son sustituidos por los intervalos  $[g_1, g_1']$ ,  $[g_1', g_2']$  y  $[g_2', g_2]$  (ver Figura 25).

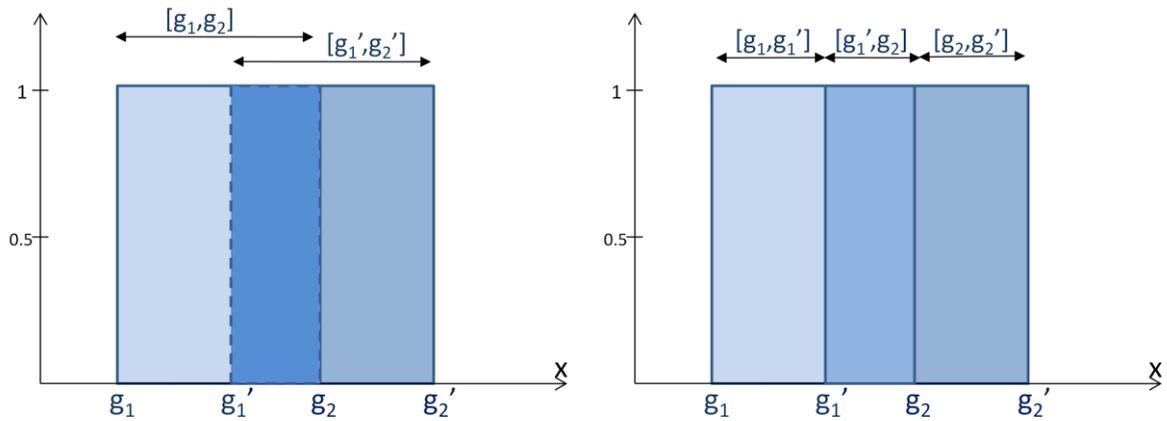


Figura 24. Los intervalos solapados (izquierda) dan lugar a nuevos intervalos (derecha)

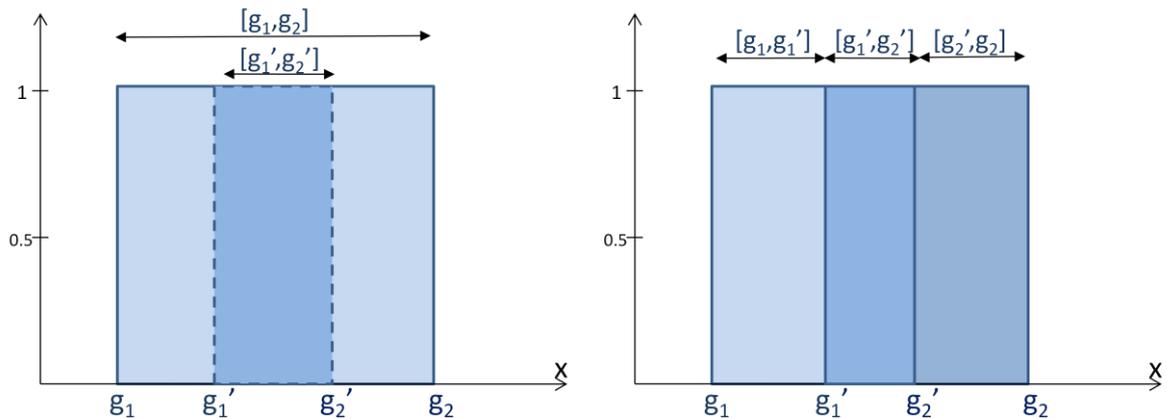


Figura 25. Un intervalo contenido en otro intervalo (izquierda) da lugar a nuevos intervalos (derecha)

Por otra parte, para evitar la existencia de intervalos pequeños y poco relevantes, los cuales dotarían de una mayor complejidad al sistema, se establecen las siguientes excepciones:

- Si  $[g_1, g_2]$  y  $[g_1', g_2']$  son dos intervalos que se solapan y se cumple que  $(g_1 - g_1') + (g_2 - g_2') < (G_{max}/10)$  (los intervalos solapados son parecidos), los intervalos  $[g_1, g_2]$  y  $[g_1', g_2']$  son sustituidos por un único intervalo  $[(g_1 + g_1')/2, (g_2 + g_2')/2]$ .
- Si por un solapamiento se genera un nuevo intervalo  $[g_1, g_2]$  tal que  $(g_2 - g_1) < (G_{max}/20)$ , este intervalo se fusionará con el intervalo adyacente  $[g_1', g_2']$  dando lugar al intervalo  $[(g_2 - g_1)/2, g_2']$ . En caso de ser adyacente a dos intervalos, la mitad del intervalo se fusionará con el intervalo anterior y la otra mitad, con el intervalo posterior.

El uso de más funciones de pertenencia aumentaría la precisión de las reglas lingüísticas; de forma que el comportamiento del sistema podría ajustarse mejor a las necesidades de los

diseñadores. Sin embargo, por otra parte, cuanto mayor es el número de conceptos empleados en la definición del conjunto de términos, más complicado se hace el proceso de diseño y los requerimientos de procesamiento del sistema difuso son mayores.

Este mismo procedimiento se aplica a la hora de determinar las funciones de pertenencia correspondientes a las variables de entrada *tiempo* y *número de accesos*.

Para cada variable de entrada, los valores lingüísticos que corresponden a cada conjunto difuso variarán dependiendo del número de conjuntos difusos creados en función de los intervalos obtenidos anteriormente. En el caso de tener dos conjuntos difusos, sus valores lingüísticos serían *alto* (*high-H*) y *bajo* (*low-L*). Si se tienen tres conjuntos difusos, los valores serían *alto* (*high-H*), *medio* (*médium-M*) y *bajo* (*low-L*). En definitiva, si el número de conjuntos difusos es impar, se comienza nombrando el conjunto difuso del centro como medio y los conjuntos situados a la derecha de este conjunto se denominarán *alto* (*high-H*), *muy alto* (*very high-VH*), *muy muy alto* (*very very high-VVH*)... según corresponda, mientras que los conjuntos situados a la izquierda se denominarán *bajo* (*low-L*), *muy bajo* (*very low-VL*), *muy muy bajo* (*very very low-VVL*), etc. En el caso de que el número de conjuntos difusos sea par se comienza nombrando los conjuntos situados en el medio como *alto* y *bajo* según corresponda (no existe el valor *medio*) y se procede de igual manera que en el caso anterior para los distintos conjuntos. En la Figura 26 se muestran las funciones de pertenencia para dos variables de entrada, una de las cuales está definida por un número par de funciones de pertenencia y la otra, por un número impar.

Finalmente, las funciones de pertenencia de la variable de salida *dificultad* (*difficulty*) son fijas y siempre toman la forma trapezoidal que se muestra en la Figura 27.

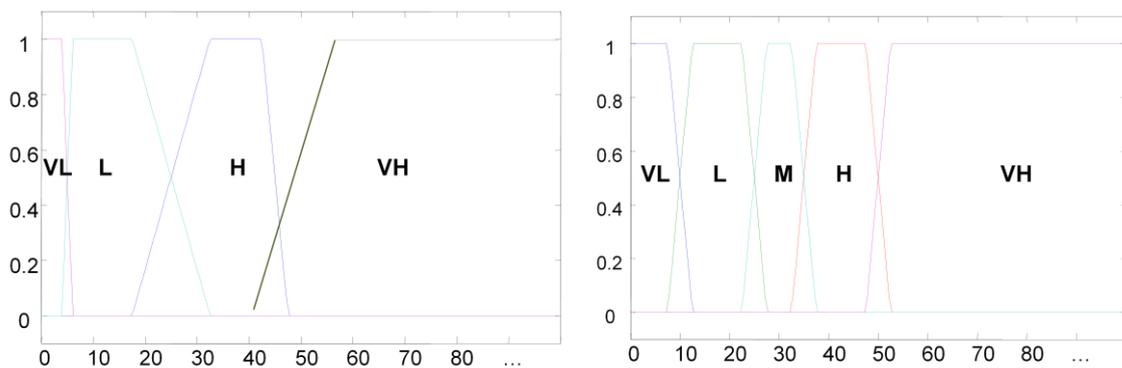


Figura 26. Funciones de pertenencia de las variables de entrada: Estructura general

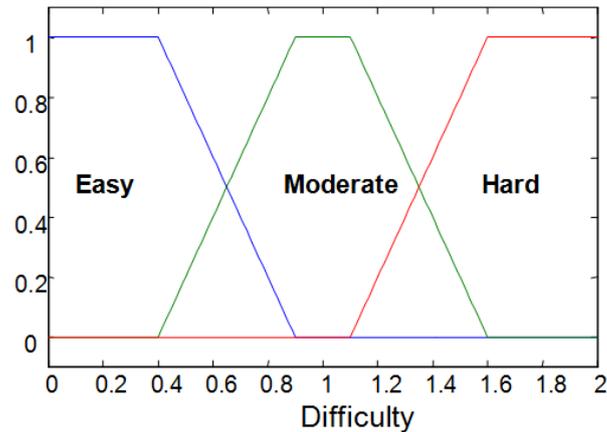


Figura 27. Funciones de pertenencia de la variable de salida *Difficulty*

#### 4.2.2 Base de reglas

Una vez que se han creado las funciones de pertenencia, el siguiente paso a la hora de desarrollar un sistema de control difuso es determinar las reglas difusas. En realidad, una regla difusa IF-THEN no es otra cosa que una declaración condicional expresada de la siguiente forma:

IF <Proposición difusa> THEN <Proposición difusa>

La base de reglas para cada nivel de dificultad se define a partir de los conjuntos difusos definidos anteriormente y de los resultados del algoritmo genético. Para mostrar de forma simplificada cómo funciona el sistema, puede suponerse que, por ejemplo, el algoritmo genético ha encontrado el juego de intervalos  $[g_1, g_2]$ ,  $[t_1, t_2]$  y  $[a_1, a_2]$ , que caracteriza las preguntas del nivel fácil. Además, tras la definición de los conjuntos difusos se comprueba que al intervalo  $[g_1, g_2]$  le corresponden dos de estos conjuntos difusos (*high -H-* y *very high-VH-*), mientras que los intervalos  $[t_1, t_2]$  y  $[a_1, a_2]$  están representados directamente por un único conjunto difuso (*low -L-*). Según los datos de este ejemplo simplificado, las reglas generadas serían:

IF GRADE IS VH AND TIME IS L AND ACCESSES IS L THEN DIFFICULTY IS EASY

IF GRADE IS H AND TIME IS L AND ACCESSES IS L THEN DIFFICULTY IS EASY

Esta relación normalmente se representa mediante el uso de una matriz relacional difusa o FAM (*Fuzzy Associative Memory*), es decir, una matriz cuyos elementos son los valores lingüísticos que toma la variable de salida para cada combinación de los posibles valores de las variables de entrada.

Finalmente, a cada regla se le asignará un *factor de incertidumbre o confianza*, describiendo el grado de confianza de la regla. Así, cuando hay dos reglas con un mismo antecedente pero distinto consecuente, el factor de incertidumbre de cada regla determina qué nivel de dificultad está mejor representado por su correspondiente regla.

### 4.2.3 Factor de confianza

El factor de confianza permite manejar la incertidumbre que ocurre cuando no hay absoluta certeza sobre una información determinada. La gestión de la incertidumbre es algo importante en el diseño de los sistemas expertos puesto que mucha de la información en la base de conocimiento de un sistema experto típico es imprecisa, incompleta o no totalmente fiable (Zadeh 1983) y el sistema experto debe ser capaz de utilizar esta información para llegar a una conclusión sobre el problema (Oh y Kandel 1991).

Cabe indicar en este punto que el factor de confianza refleja la falta de certeza sobre una información mientras que la difuminación que se maneja en los sistemas de lógica difusa se refiere a que el límite de la información no está claro (Leung y Lam 1998).

Cuando un sistema difuso experto es definido por un experto humano, este experto asigna a las reglas un factor de confianza que es el grado en que el fenómeno representado por la regla es cierto de acuerdo a anteriores experiencias y conocimientos de este experto. Así, se trata de un concepto empírico y subjetivo (Qian Hao y Tao Lu 2009).

El factor de confianza  $FC(h, e)$  es un valor numérico entre 0 y 1 que indica la probabilidad de que la hipótesis  $h$  se cumpla si se da la evidencia  $e$  (Leung y Lam 1998) (Negnevitsky 2005) (Qian Hao y Tao Lu 2009).

Entonces, se puede trabajar con dos tipos de reglas tal y como se define en (Cordón, del Jesús, y F. Herrera 1998):

1. Reglas difusas sin factor de confianza:

$$R_k: IF x_1 \text{ is } A_1^k \text{ AND } x_2 \text{ is } A_2^k \text{ AND } \dots \text{ AND } x_N \text{ is } A_N^k \text{ THEN } y \text{ is } C_j$$

donde  $x_1, x_2, \dots, x_N$  son los valores de los atributos empleados para la clasificación,  $A_1^k, \dots, A_N^k$  son los valores lingüísticos que discretizan el dominio de las variables,  $e$  y es la clase  $C_j$  a la cual el patrón pertenece.

2. Reglas difusas con factor de confianza:

$$R_k: IF x_1 \text{ is } A_1^k \text{ AND } x_2 \text{ is } A_2^k \text{ AND } \dots \text{ AND } x_N \text{ is } A_N^k \text{ THEN } y \text{ is } C_j \text{ with } FC^k$$

Donde  $FC^k$  es el factor de confianza de la clasificación en la clase  $C_j$  para un patrón perteneciente a la región delimitada por el antecedente.

Para definir un factor de confianza en las reglas que el sistema genera, el sistema se basa en las medidas de soporte y confianza de reglas de clasificación especificadas en las ecuaciones (3) y (6).

El soporte (calculado como el cociente entre el número de patrones de un nivel dado que cumple la regla y el número total de patrones de ese nivel) da una indicación de la significancia de la regla para describir los patrones de un nivel dado (ej. Si un 95% de los patrones de un nivel cumplen una determinada condición, la regla es muy significativa y descriptiva de los patrones de ese nivel). Por otro lado, la confianza indica la probabilidad de que un patrón que cumpla el antecedente de la regla, pertenezca a la clase de la regla. Utilizar para el factor de confianza únicamente la confianza no es conveniente en el caso bajo estudio, ya que los patrones de entrada que se usan para entrenar al sistema no tienen por qué estar equitativamente repartidos en los tres niveles, y podría haber un nivel con muchos más patrones que otro nivel. Por ejemplo, se supone el caso en que el sistema encuentra dos reglas con el mismo antecedente para clasificar dos clases diferentes (fácil y asequible) y hay un 20% de los 500 patrones de

nivel asequible que cumplen ese antecedente y un 95% de 100 patrones de nivel fácil que cumplen también el antecedente. Si sólo se tuviera en cuenta la confianza, el factor de confianza sería 0.51 para la regla del nivel asequible y 0.48 para la regla del nivel fácil (valor pequeño teniendo en cuenta que esa regla la cumplen un 95% de los patrones de nivel fácil). Si sólo se tuviera en cuenta el soporte, el factor de confianza sería 0.2 para la regla del nivel asequible y 0.95 para la regla del nivel fácil, valores tampoco apropiados ya que también es importante en la fiabilidad de la regla el número de patrones con los que se entrena. Si tenemos en cuenta conjuntamente el soporte y la confianza, para la regla del nivel asequible el factor de confianza es 0.10 mientras que el factor de confianza de la regla de nivel fácil es 0.46. Así, el factor de confianza de una regla de un nivel con muchos patrones, que no es característica de ese nivel, disminuye.

Por tanto el factor de confianza  $FC^k$  se calcula a partir de los intervalos (conjuntos clásicos) que define el algoritmo genético, una vez tratados los solapamientos, tal y como se ha indicado en el apartado 4.2.1, a partir de la siguiente fórmula:

$$FC_k = \frac{n_j^i}{n_j} \cdot \frac{n_j^i}{n^i} \quad (23)$$

donde  $n_j^i$  representa el número de patrones de dificultad  $j$  contenidos en el juego de intervalos definidos por el antecedente, y  $n_j$  es el número total de patrones de dificultad  $j$ . De esta manera el cociente entre  $n_j^i$  y  $n_j$  correspondería a la medida de soporte especificada en la ecuación (3). Por otro lado, como ya se ha indicado,  $n_j^i$  representa el número de patrones de dificultad  $j$  contenidos en el juego de intervalos definidos por el antecedente, y  $n^i$  es el número total de patrones (de cualquier clase) contenidos en este juego de intervalos. Esta medida corresponde a la medida de confianza de la regla expuesta en la ecuación (6).

#### 4.2.4 Implicación y composición

Dado que las proposiciones difusas son explicadas como relaciones difusas, el siguiente paso es ver cómo interpretar las operaciones IF-THEN. Por lo tanto, una vez definida la base de reglas se debe establecer la estrategia de implicación y el método de composición del sistema de control difuso.

Los *métodos de implicación* calculan el grado de verdad de las premisas de las reglas y lo aplican a la conclusión, dando como resultado un subconjunto difuso asignado a cada una de las variables de salida. Normalmente, se suele utilizar uno de los dos métodos de implicación descritos a continuación (MathWorks 2000):

- *MIN*: calcula el mínimo de la función de pertenencia de los datos de entrada para cada uno de los antecedentes. Gráficamente, la función de pertenencia de la salida quedará truncada a la altura del grado de verdad de la premisa de la regla.
- *PRODUCT*: calcula el producto de la función de pertenencia de los datos de entrada para cada uno de los antecedentes. Gráficamente, la función de pertenencia de la salida quedará escalada a la altura del grado de verdad de la premisa de la regla.

Aunque el método MIN es el más comúnmente utilizado, el método PRODUCT o de escalado pierde menos información al mantener la forma de la función de pertenencia de salida. Sin embargo, el método MIN se suele preferir porque supone menos coste de cálculo (Negnevitsky 2005).

Una vez evaluadas todas las reglas activas en un momento dado y obtenidos los conjuntos de salida modificados, es necesario combinar todos los subconjuntos difusos asignados a cada variable de salida en un único subconjunto difuso. Para ello, se puede aplicar uno de los siguientes *métodos de composición o agregación* (MathWorks 2000):

- *MAX*: obtiene como conjunto de salida el valor máximo de los conjuntos de cada una de las reglas.
- *SUM*: toma como conjunto de salida la suma de los conjuntos de cada una de las reglas. Por lo tanto, este método de composición presenta el problema de que puede dar lugar a valores mayores que 1. Para evitar esto, se puede usar la suma limitada (*bounded sum*) que toma el mínimo entre el resultado del método SUM y el número 1 (Lowen 1995).

En la práctica el método más común de implicación/composición es el método *min-max*, el cual es la opción por defecto implementado en Matlab Fuzzy Logic Toolbox. En cualquier caso, mientras que de otros componentes sí que existen aproximaciones y sus propiedades teóricas son bien conocidas, en la literatura hay poca información sobre cómo se debe elegir el método de composición más adecuado a la hora de desarrollar un sistema difuso.

#### 4.2.5 Decodificación o *defuzzification*

Una vez aplicados los métodos de composición de las reglas, la salida es un conjunto difuso; sin embargo, para poder usar esta información como señal de control, ésta tiene que ser un número preciso. Por lo tanto, como último paso, en cualquier sistema de control difuso se debe definir el método de decodificación o *defuzzification* a emplear.

En el proceso de decodificación, el conjunto difuso obtenido para el grado de calidad se convierte a un valor preciso. En la decodificación se pueden utilizar diferentes métodos (Jantzen 1998):

- *Centroid* o centroide: utiliza como salida del sistema, el centro de gravedad de la función de pertenencia de salida, es decir, el punto central del área cubierta por la función de pertenencia. Matemáticamente, se calcula siguiendo la siguiente fórmula:

$$\bar{y} = \frac{\int y \cdot \mu(y) dy}{\int \mu(y) dy} \quad (24)$$

donde  $y$  es la variable de salida y  $\mu(y)$  es la función de pertenencia de salida.

- *MOM (Mean of Maximum)*: elige como salida la media de los máximos. Esto mismo se puede ver de una manera más formal en la siguiente expresión matemática:

$$\bar{y} = \sum_{n=1}^N \frac{y_n}{N} \quad (25)$$

donde  $y_n$  es la coordenada de cada máximo y  $N$  es el número de máximos.

- *SOM (Smallest of Maximum)*: establece como salida el valor del menor máximo.
- *LOM (Largest of Maximum)*: establece como salida el valor de mayor máximo.

En el caso de que sólo haya un máximo, MOM, SOM y LOM proporcionarían el mismo resultado.

- *Bisector*: establece como salida el punto donde el conjunto difuso es dividido en dos sub-regiones de igual área. En muchos casos, coincide con el resultado obtenido a través del centroide.

$$\bar{y} = \left\{ y \mid \int_{min}^y \mu(y) dy = \int_y^{max} \mu(y) dy \right\} \quad (26)$$

donde  $y$  es la variable de salida,  $\mu(y)$  es la función de pertenencia de salida,  $min$  es el valor más a la izquierda del universo de la variable de salida y  $max$  es el valor más a la derecha del universo de la variable de salida.

En este trabajo, como método de decodificación se ha probado tanto con *mom* como con *centroid* con el fin de determinar cuál es el que ofrece mejores resultados. Finalmente, dado que con el método *mom* se han obtenido resultados más ajustados al funcionamiento deseado del sistema, éste ha sido el método escogido. Además, el coste computacional que supone el método *mom* es menor que el *centroide*, que implica calcular una integral.

# CAPÍTULO 5

## EVALUACIÓN DEL SISTEMA EXPERTO

### 1 Diseño de experimentos y casos de estudio

A lo largo de la investigación se ha seguido un proceso cíclico con fases de diseño, desarrollo y evaluación, que ha permitido reorientar y refinar progresivamente la solución propuesta.

En una primera etapa, se diseña, prueba y ajusta un algoritmo genético que busca la mejor solución al problema. En el apartado 4.1.2 del capítulo 4 se incluye una descripción de estos primeros experimentos en los que se detecta la necesidad de encontrar no sólo la mejor solución al problema, sino varias soluciones buenas y diversas. Por ello, se diseña y desarrolla la siguiente versión del algoritmo genético que incluye el método *sharing* (y algún otro mecanismo adicional), que permite encontrar diversas soluciones al problema. Las pruebas realizadas con esta versión del algoritmo genético están descritas en (E. Verdú, M. J. Verdú, et al. 2010).

Las anteriores pruebas se refieren únicamente al algoritmo genético. En el presente apartado, describimos las pruebas realizadas con el Sistema Experto Genético Difuso completo. Para la realización de las pruebas se ha simulado el comportamiento del motor de inferencia mediante la herramienta *Fuzzy Logic Toolbox* de *MATLAB*.

En los experimentos se ha contado con datos reales pertenecientes a dos asignaturas de Telemática de dos titulaciones distintas y se ha empleado un método para validar el sistema contra un grupo de expertos, descritos en los siguientes apartados.

Por último, cabe indicar que evaluar un sistema inteligente consiste en comprobar que su funcionamiento satisface las necesidades y requisitos del usuario (Negnevitsky 2005), por tanto la evaluación de un sistema se refiere a todos los aspectos del mismo: utilidad, eficacia, coste, facilidad de uso, etc, siendo la validación del sistema un aspecto de la evaluación (O'Keefe, Balci, y E.P. Smith 1986). Dado que durante el desarrollo de este trabajo no se dispone de una implementación de parte del sistema (el motor de inferencia), no se puede llevar a cabo una evaluación completa del mismo, pero sí se puede validar el diseño del sistema experto, comprobando que tenga un nivel de precisión aceptable. A la hora de evaluar sistemas de clasificación hay que tener en cuenta su posterior uso y en este sentido se puede distinguir aquellos sistemas que trabajan de manera autónoma y aquellos que son una herramienta de ayuda en los procesos de decisión del usuario. Lo que más importa en el primer tipo de sistemas, donde se encuadra el sistema propuesto, es obtener clasificaciones precisas, mientras que en el segundo tipo de sistemas hay que tener en cuenta otros aspectos como la comprensibilidad o la versatilidad (Cordón, del Jesús, y F. Herrera 1998).

## 1.1 Caso de estudio 1

La primera experiencia se ha realizado con datos de la asignatura Transmisión de Datos de la titulación Ingeniero Técnico de Telecomunicación Especialidad Sistemas de Telecomunicación de la Universidad de Valladolid, una asignatura obligatoria de segundo curso del programa de dicha titulación.

El objetivo de la asignatura es el estudio y comprensión de conceptos fundamentales de redes de datos. En concreto, se trata de que el alumno se familiarice con el funcionamiento de los niveles de red, transporte y aplicación de las redes TCP/IP (*Transmission Control Protocol / Internet Protocol*). La mitad de la docencia de la asignatura corresponde a clases presenciales en las que se exponen los contenidos teóricos de la asignatura y se resuelven problemas prácticos en la pizarra.

De forma paralela a las clases teóricas, se realizan clases prácticas en laboratorio, de manera que complementen y afiancen los conceptos aprendidos en teoría.

Varias sesiones de laboratorio se dedican a realizar prácticas con QUESTOURnament. El profesor propone semanalmente una serie de desafíos que los alumnos deben responder.

Durante el curso 2008-2009, se desarrolla un concurso, de varias semanas de duración, durante el cual el profesor propone 12 desafíos que los alumnos deben resolver individualmente. El sistema pide al profesor indicar la dificultad de los desafíos al crearlos en el sistema y el profesor clasifica 4 desafíos como fáciles, 3 desafíos como asequibles y 5 como difíciles.

Los desafíos son propuestos unas horas antes de comenzar la sesión de laboratorio (las clases tienen lugar por la tarde y los desafíos comienzan por la mañana) así que los alumnos pueden tratar de resolver los desafíos durante la mañana y acudir al laboratorio si tienen dificultades en la resolución de algunos desafíos, para consultar al profesor. El concurso está configurado de tal manera que cuanto antes se resuelvan los desafíos más puntuación se obtiene y, por tanto, los alumnos tratan de resolver los desafíos durante la mañana. Se trata de una situación en que no se puede controlar si los alumnos están tratando de resolver desafíos conjuntamente y, por supuesto, no se puede saber si el tiempo que el sistema utiliza como

tiempo de resolución del desafío es cercano a la realidad, ya que el alumno puede contestar al teléfono o realizar cualquier otra actividad en su casa una vez ha comenzado a resolver el desafío. Esta es una situación realista y común en el uso de la herramienta QUESTOURnament, que muchas veces se ha utilizado para el desarrollo de concursos de participación optativa, en que el alumno participa remotamente en un concurso.

En este caso, la participación en el QUESTOURnament es obligatoria, 38 alumnos participan en el concurso y se almacenan un total de 406 respuestas en la base de hechos.

## 1.2 Caso de estudio 2

El segundo caso de estudio corresponde a la asignatura también denominada Transmisión de Datos de segundo curso de la titulación Ingeniero de Telecomunicación.

El objetivo de esta asignatura es el estudio y la aplicación práctica de conceptos fundamentales de redes de comunicaciones en el marco del modelo de referencia OSI y la arquitectura de comunicaciones TCP/IP.

La mitad de la docencia corresponde a clases presenciales en las que se exponen los contenidos teóricos del programa de la asignatura, se describen ejemplos y se desarrollan ejercicios prácticos. De forma paralela a las clases en aula, se realizan clases prácticas en laboratorio, de manera que complementen y afiancen los conceptos aprendidos en teoría. También se realizan prácticas con QUESTOURnament.

Se desarrolla tanto un concurso optativo como uno obligatorio, siendo los datos del concurso obligatorio los que corresponden a este caso de estudio. La diferencia fundamental de este caso con el caso anterior es que en esta asignatura el concurso obligatorio comienza en la misma sesión de laboratorio a la que los alumnos acuden y la situación está más controlada. Los alumnos resuelven las cuestiones de manera individual y no conjunta pero tienen libertad para ir resolviendo los desafíos en el orden que quieran.

En este caso se proponen 24 desafíos, 8 clasificados inicialmente por el profesor como fáciles, otros 8 como asequibles y 8 como difíciles. El total de respuestas es 1016 por parte de 33 alumnos, que quedan almacenadas en la base de hechos.

## 2 Validación

### 2.1 Metodología

La mayoría de las definiciones de sistema experto incluyen alguna referencia a la capacidad del sistema de comportarse de manera próxima a cómo lo haría un experto humano (O'Keefe, Balci, y E.P. Smith 1986). Los sistemas inteligentes suelen resolver problemas que no tienen una clara definición de cuál es la solución correcta (Negnevitsky 2005). Por tanto, los sistemas expertos típicamente deben ser validados contra expertos, excepto si existen resultados conocidos que proporcionen una base útil para validarlos (O'Keefe, Balci, y E.P. Smith 1986).

No disponiendo de resultados previos que ayuden en esta tarea, la validación de la solución propuesta ha consistido fundamentalmente en comprobar que el resultado proporcionado por el sistema experto, que consiste en una clasificación de la dificultad de los desafíos, coincide con la clasificación que los profesores harían observando los datos disponibles.

A partir de datos relativos a respuestas reales de los alumnos a los desafíos propuestos en las asignaturas antes descritas, el sistema ha reclasificado los desafíos de acuerdo a su dificultad real estimada. Por otro lado, tres expertos, profesores del Área de Telemática de la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universidad de Valladolid, han analizado el comportamiento real de los alumnos al contestar los desafíos y han determinado cuál es la dificultad que ellos asignarían a cada desafío. Los datos de cada respuesta que los profesores han usado para el análisis son los mismos que el sistema emplea como entrada (base de hechos). El análisis realizado por los profesores ha sido “manual” y ha consistido en observar el tiempo empleado, la nota obtenida y el número de accesos realizados por parte de los alumnos al contestar a cada uno de los desafíos.

Este tipo de validación “*contra el experto*” se suele denominar validación retrospectiva frente a la validación prospectiva, que se puede aplicar cuando existe una referencia estándar que permite establecer si la interpretación del sistema es correcta (Mosqueira-Rey y Moret-Bonillo 2000). El validar contra un grupo de expertos y no sólo un experto tiene la ventaja de que mejora la objetividad y permite comparar el grado de consistencia que existe entre los expertos del dominio.

El método seguido es, por tanto, una “*validación contra un grupo de expertos*” y está basado en el método descrito en (Mosqueira-Rey, Moret-Bonillo, y Fernández-Leal 2008). En resumen, se trata de dar una medida que indique el grado de consenso entre los expertos humanos y comprobar si el sistema experto se comporta como un experto humano más, pudiendo ser incorporado al grupo de expertos sin empeorar el grado de consenso. Los pasos de la validación se ilustran en la Figura 28 y se resumen a continuación:

1. *Creación de una base de datos de validación.* Los expertos del dominio así como el sistema experto realizan una clasificación de la dificultad de los desafíos.
2. *Creación de las tablas de contingencia para los posibles pares de expertos.* Para cada emparejamiento posible de expertos se crea una tabla de contingencia que muestra los acuerdos y desacuerdos en la clasificación entre los pares de expertos.
3. *Cálculos de las medidas de pares.* A partir de las tablas de contingencia se calculan las medidas de pares, que indican el grado de acuerdo entre dos expertos, dando lugar a la tabla de pares que recopila las medidas de pares de todos los expertos.
4. *Cálculo de la medida de grupo a partir de las tablas de pares.* A partir de los datos de las tablas de pares se calcula la medida de grupo, que resume la información contenida en las tablas de pares y da una medida del grado de consenso dentro de un grupo.
5. Con el objetivo final de probar si el sistema experto puede “pasar” como un experto humano, se determina si el sistema experto está de acuerdo con los expertos del grupo de referencia en el mismo grado que los expertos están de acuerdo entre ellos mismos.

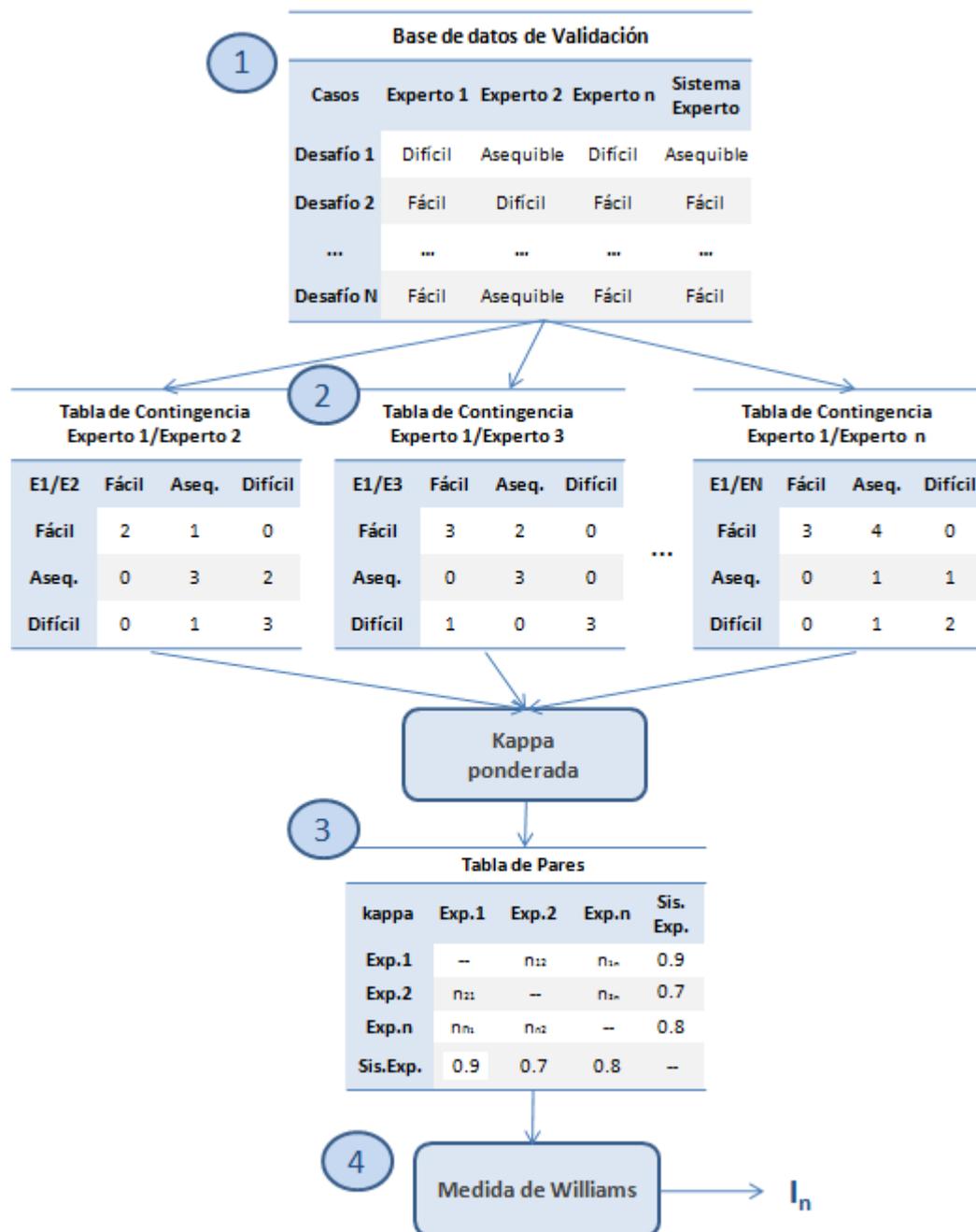


Figura 28. Proceso para calcular la medida de grupo. Fuente: Elaboración propia basada en (Mosqueira-Rey, Moret-Bonillo, y Fernández-Leal 2008)

		<i>Experto A</i>				
		<i>Clase 1</i>	<i>Clase 2</i>	...	<i>Clase m</i>	<i>Total</i>
<i>Experto B</i>	<i>Clase 1</i>	$n_{11}$	$n_{12}$	...	$n_{1m}$	$n_{1.} = \sum_{i=1}^m n_{1i}$
	<i>Clase 2</i>	$n_{21}$	$n_{22}$	...	$n_{2m}$	$n_{2.} = \sum_{i=1}^m n_{2i}$
	...	...	...	...	...	
	<i>Clase m</i>	$n_{m1}$	$n_{m2}$	...	$n_{mm}$	$n_{m.} = \sum_{i=1}^m n_{mi}$
	<i>Total</i>	$n_{.1} = \sum_{i=1}^m n_{i1}$	$n_{.2} = \sum_{i=1}^m n_{i2}$	...	$n_{.m} = \sum_{i=1}^m n_{im}$	$N$

Tabla 9. Tabla de contingencia

En la Tabla 9 se muestra un esquema de tabla de contingencia. Cada celda tiene un valor  $n_{ij}$  que corresponde al número de veces que el experto A y el experto B han clasificado un caso como clase  $j$  y clase  $i$ , respectivamente.

Hay varias medidas que permiten calcular las medidas de pares. El *índice de acuerdo* es una medida muy popular que se calcula como el cociente entre el número de observaciones en que la opinión de los expertos coincide y el número total de observaciones (Mosqueira-Rey, Moret-Bonillo, y Fernández-Leal 2008). Sin embargo, se puede dar el caso en que la opinión de los expertos coincida por casualidad en algunos casos. Esto es más probable que ocurra si se tiene un pequeño número de clases o categorías (por ejemplo, si en vez de tener en cuenta cinco niveles de clases para una validación, se tienen en cuenta tres clases, la posibilidad de que la opinión de los expertos coincida es lógicamente mayor). Por ello, para el cálculo de las medidas de pares se ha empleado la medida denominada *kappa*, medida estadísticamente robusta (Bendavid 2008), que tiene en cuenta esta posibilidad de coincidencia por casualidad y está definida por la siguiente expresión (Bendavid 2008) (Mosqueira-Rey, Moret-Bonillo, y Fernández-Leal 2008) (Viera y Garrett 2005):

$$K = \frac{p_a - p_c}{1 - p_c} \quad (27)$$

siendo  $p_a$  la proporción de acuerdo observado o el *índice de acuerdo* previamente mencionado y  $p_c$  la proporción de acuerdo por casualidad esperado.

La proporción de acuerdo esperado  $p_a$  se calcula con la siguiente fórmula (Mosqueira-Rey y Moret-Bonillo 2000):

$$p_a = \frac{\sum_{i=1}^m n_{ii}}{N} = \sum_{i=1}^m p_{ii} \quad (28)$$

siendo  $N$  el total de casos clasificados (total de observaciones),  $m$  el número de clases o categorías, y  $n_{ii}$  es el número indicado en la celda  $ii$  de la tabla de contingencia (siendo  $p_{ii}$  la frecuencia relativa de la celda  $ii$ ). En resumen, esta proporción de acuerdo se calcula como la suma de los valores de la diagonal de la tabla de contingencia y, por tanto, se trata de una medida del acuerdo entre los expertos.

La proporción de acuerdo por casualidad esperado  $p_c$  se calcula como la suma de los productos de probabilidades marginales según la siguiente expresión:

$$p_c = \sum_{i=1}^m \frac{n_i}{N} \cdot \frac{n_i}{N} \quad (29)$$

siendo  $n_i$  y  $n_i$  la suma total de los valores en las conlumnas y filas de la tabla de contingencia (ver Tabla 9), respectivamente.

Así el valor de *kappa* se puede calcular de manera sencilla a partir de los valores en la tabla de contingencia tal y como se indica en la siguiente ecuación:

$$K = \frac{N \cdot \sum_{i=1}^m n_{ii} - \sum_{i=1}^m n_i \cdot n_i}{N^2 - \sum_{i=1}^m n_i \cdot n_i} \quad (30)$$

El valor de *Kappa* varía de -1 (desacuerdo total) a 1 (acuerdo total) pasando por el 0 (clasificación aleatoria) (Bendavid 2008).

En el problema bajo estudio sucede que no tiene la misma intensidad un desacuerdo en el que un experto clasifica una pregunta como fácil y otro la clasifica como difícil que un desacuerdo en que un experto clasifica una pregunta como fácil y otro la clasifica como asequible. De hecho, con categorías ordinales, los coeficientes ponderados de *kappa* son frecuentemente usados, ponderando los desacuerdos por la magnitud de la discrepancia (Brenner y Kliebsch 1996). Por ello, hay que ponderar los niveles de desacuerdo y, para ello, se puede usar el valor de *Kappa* ponderado. Entonces se debe ponderar el nivel de acuerdo o desacuerdo definiendo una matriz de pesos con valores  $w_{ij}$  que denotan el peso del valor en la fila  $i$  y columna  $j$  de la matriz de contingencia, siendo  $0 \leq w_{ij} \leq 1$ .  $w_{ij}$  tomará el valor 1 en los valores situados en la diagonal de la tabla que indican un total acuerdo. Por otro lado,  $w_{ij}$  valdrá 0 en los casos que se refleje un total desacuerdo.

Los pesos se pueden definir libremente, aunque hay dos esquemas de pesos típicamente utilizados, el lineal y el cuadrático (Brenner y Kliebsch 1996). Los pesos lineales se pueden calcular con la siguiente expresión:

$$w_{ij} = 1 - \frac{|i-j|}{m-1} \quad (31)$$

y los cuadráticos, con la siguiente:

$$w_{ij} = 1 - \frac{(i-j)^2}{(m-1)^2} \quad (32)$$

En este problema se va a usar los coeficientes lineales, puesto que los cuadráticos penalizan menos el desacuerdo (por ejemplo, si hay cinco clases, el peso lineal para un desacuerdo entre la clase 1 y la clase 3 es 0.5 mientras que el peso cuadrático toma el valor de 0.75).

Una vez definidos los pesos, se puede calcular el valor de *kappa* ponderado con la expresión (Bendavid 2008):

$$K_w = \frac{\sum_{i=1}^m \sum_{j=1}^m w_{ij} p_{ij} - \sum_{i=1}^m \sum_{j=1}^m w_{ij} P_i \cdot P_j}{1 - \sum_{i=1}^m \sum_{j=1}^m w_{ij} P_i \cdot P_j} \quad (33)$$

siendo  $w_{ij}$  los pesos y  $P_i$  y  $P_j$  las probabilidades marginales.

Una vez se tienen las medidas de pares, se puede realizar la medida de grupo con el objetivo de conocer el nivel de acuerdo que existe entre los expertos del grupo. Para ello se emplea la *medida de Williams* (Mosqueira-Rey, Moret-Bonillo, y Fernández-Leal 2008) que define las siguientes ecuaciones:

$$P_n = 2 \frac{\sum_{a=1}^{n-1} \sum_{b=a+1}^n P_{(a,b)}}{n(n-1)} \quad (34)$$

$$P_0 = \frac{\sum_{a=1}^n P_{(0,a)}}{n} \quad (35)$$

$$I_n = \frac{P_0}{P_n} \quad (36)$$

donde  $n$  corresponde al número de expertos de referencia y  $P_{(a,b)}$  es la medida de pares entre los expertos  $a$  y  $b$ .

$I_n$  es el índice de Willians y si toma un valor igual o mayor que 1 significa que el sistema experto coincide con el consenso del grupo de expertos (si es igual a 1 coincide exactamente igual a como los miembros del grupo coinciden entre sí). Por el contrario si es menor que 1 el acuerdo entre el sistema experto y el grupo de expertos es menor que el acuerdo entre los propios miembros del grupo (Mosqueira-Rey, Moret-Bonillo, y Fernández-Leal 2008).

Por tanto, con este valor  $I_n$  se podrá determinar si el comportamiento del sistema experto se puede asimilar al comportamiento de un experto humano.

## 2.2 Resultados

### 2.2.1 Caso de estudio 1

Partiendo de los 406 patrones de respuesta disponibles en la base de hechos, el sistema genético, en primer lugar, caracteriza las respuestas correspondientes a los distintos niveles de dificultad. En concreto para caracterizar las preguntas de nivel fácil, en este experimento se cuenta con 134 patrones de respuesta fáciles; para las asequibles se cuenta con 103 patrones; y para las difíciles, con 169 patrones. Los resultados del algoritmo genético se muestran en la Tabla 10.

Nivel Difícil (169 patrones)				Nivel Asequible (103 patrones)				Nivel fácil (134 patrones)			
Tiempo	Nota	Accesos	Fitness	Tiempo	Nota	Accesos	Fitness	Tiempo	Nota	Accesos	Fitness
[1,48]	[0,5]	[3,4]	0.143	[4,28]	[30,50]	[1,2]	0.301				
[13,36]	[44,60]	[1,2]	0.107	[7,35]	[0,18]	[1,2]	0.251	[0,25]	[80,100]	[1,2]	0.541
				[27,74]	[84,100]	[1,2]	0.167				

Tabla 10. Juegos de intervalos encontrados por el algoritmo genético para el caso de estudio 1

Los juegos de intervalos encontrados por el algoritmo genético (ver Tabla 10) son la base que emplea el *Generador del Modelo Difuso* para definir, siguiendo los procedimientos descritos en el apartado 4.2 del capítulo 4, las funciones de pertenencia de las variables de entrada (ver Figura 29) así como las 16 reglas representadas en la Figura 30, que describen el comportamiento de los estudiantes al enfrentarse a las preguntas de los distintos niveles de dificultad. Esta figura esquematiza el funcionamiento del motor de inferencia con un ejemplo. La entrada al motor de inferencia es un patrón de respuesta con los valores de tiempo, nota (*grade*) y número de accesos iguales a 31, 67 y 3, respectivamente. La inferencia tiene lugar en cuatro pasos. Primero, las variables de entrada se difuminan durante la fase “*fuzzification*”. A continuación las 16 reglas son evaluadas y el resultado es que el patrón de entrada cumple el antecedente de dos de las reglas. Los métodos *Min* y *Max* implementan los operadores difusos AND y OR. Durante el tercer paso, los consecuentes de las reglas son agregados dando lugar a un único conjunto difuso. Por último, la fase “*defuzzification*” da lugar a un número como salida. En este ejemplo, al patrón de entrada le corresponde una dificultad de 1.24 que corresponde a un nivel de dificultad a medio camino entre asequible y difícil, más cercano a asequible, como se puede ver en la Figura 29.

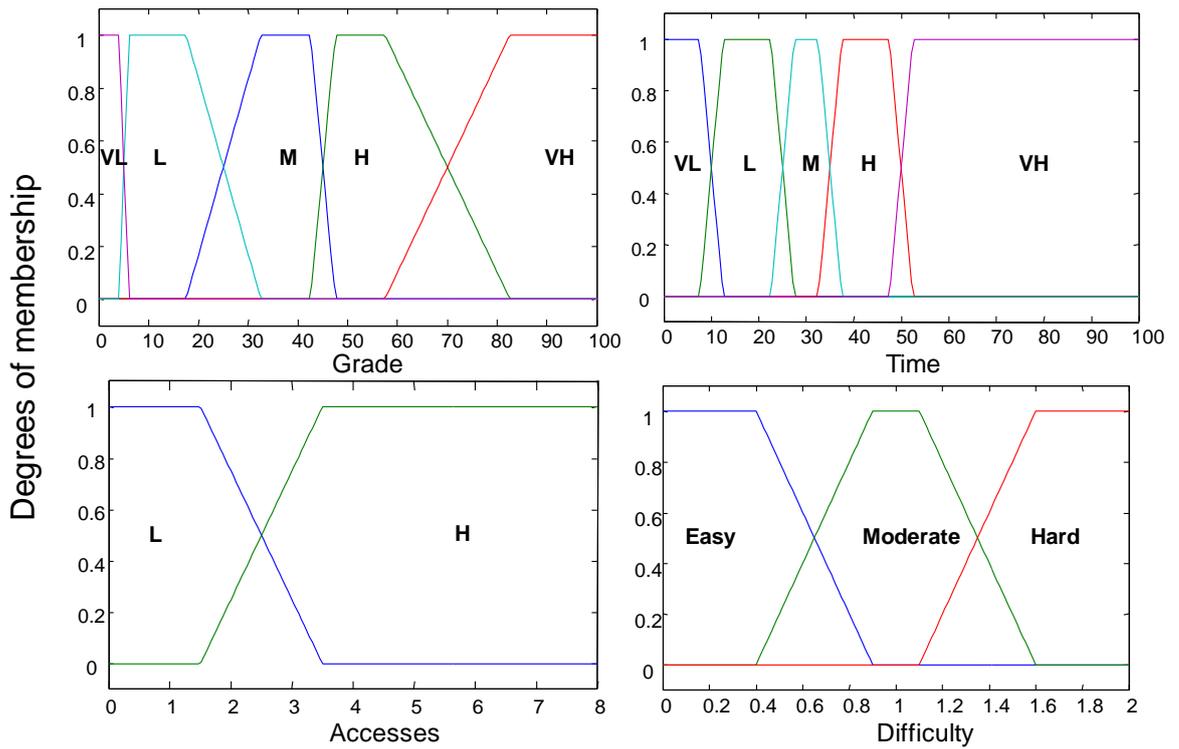


Figura 29. Conjuntos difusos de las variables de entrada y de salida para el caso de estudio 1

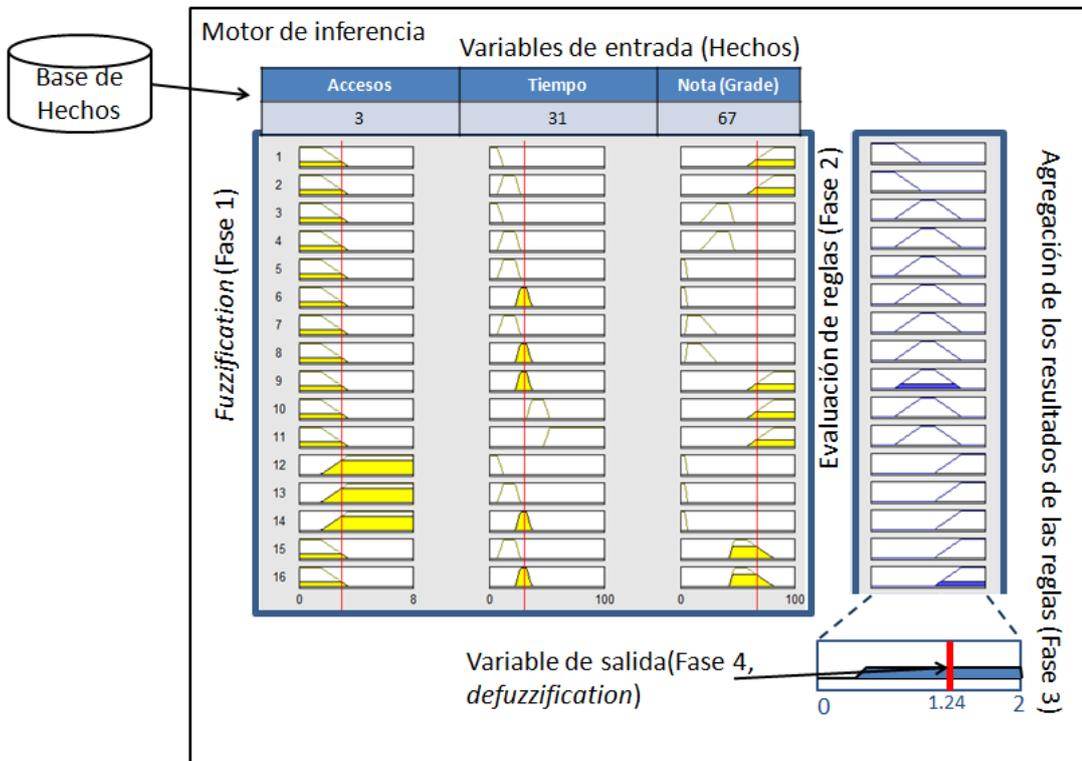


Figura 30. Representación de las 16 reglas que describen el sistema (caso de estudio 1)

Las reglas difusas que se han generado se pueden representar gráficamente mediante la Memoria Asociativa Difusa o F.A.M (*Fuzzy Associative Memory*), tal y como se muestra en la Figura 31.

Con ayuda de esta representación, de un vistazo, se puede detectar comportamientos habituales en los alumnos. Por ejemplo, cuando los alumnos no saben cómo responder a un desafío, lo leen varias veces y, finalmente, envían una respuesta elaborada en poco tiempo buscando un acierto en algunos casos y, en otros casos, envían una respuesta más meditada empleando más tiempo (H. Liu et al. 2004). Este comportamiento se refleja en las respuestas a las preguntas difíciles, que se caracterizan por un alto número de accesos (H), tiempo pequeño (VL, L) a medio (M) para resolverlas y nota muy baja (VL). Hay otras reglas correspondientes a las preguntas difíciles en las que la nota es alta (H). Estas reglas podrían reflejar el comportamiento de los alumnos con mayor nivel de conocimiento.

Por otra parte, las preguntas fáciles se caracterizan porque los alumnos obtienen notas muy altas (VH), y emplean poco tiempo para resolverlas (L o VL). También hay respuestas a preguntas asequibles que obtienen notas muy altas (VH) y, sin embargo, los alumnos necesitan más tiempo para resolverlas.

La representación tridimensional que se muestra en la Figura 32 permite también de un vistazo conocer cómo se comportan los estudiantes ante las preguntas de distintas dificultades. En la gráfica se representa la dificultad respecto al tiempo (*time*) y la nota (*grade*). Como ya se podía observar con la representación F.A.M., cuando las preguntas son fáciles los alumnos dedican poco tiempo para responder y obtienen buenas notas. Cuando los alumnos dedican más tiempo para responder y la nota obtenida disminuye, se encuentran ante preguntas más difíciles.

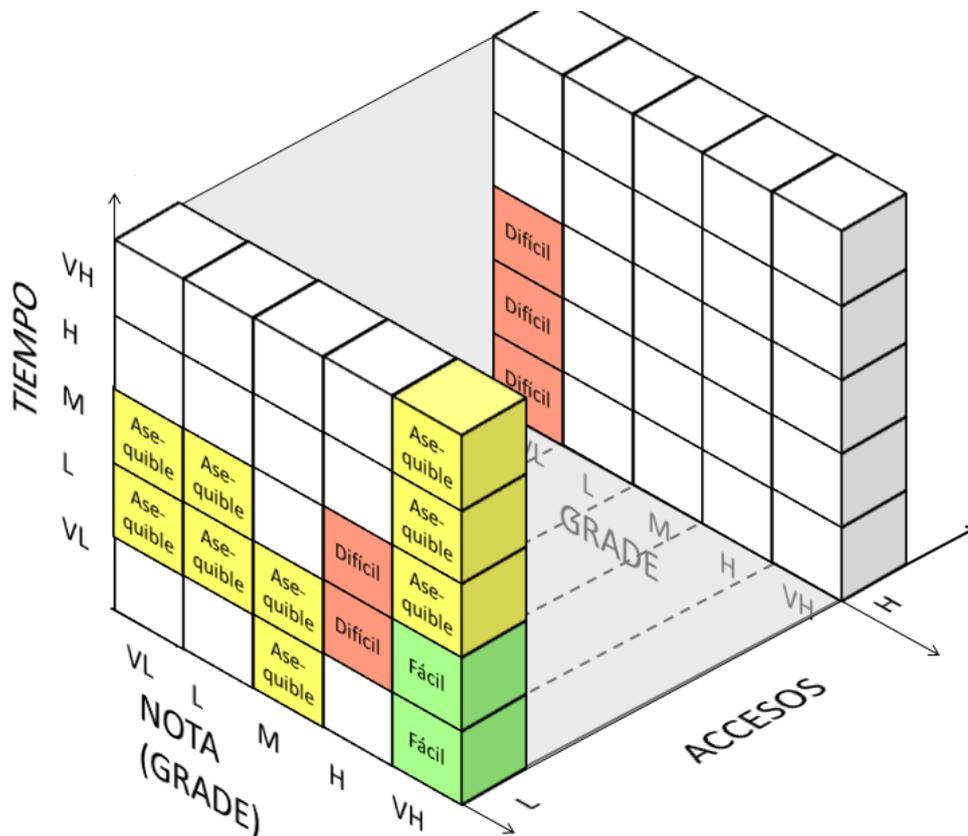


Figura 31. Representación F.A.M. de las reglas (caso de estudio 1)

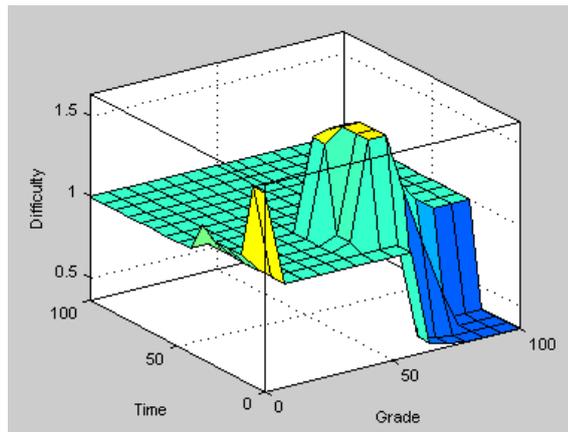


Figura 32. Representación tridimensional de la base de reglas del caso de estudio 1

La Tabla 11 muestra el nivel de dificultad estimado por el sistema Experto Genético Difuso para los 12 desafíos del experimento. La primera columna contiene unos números que son los identificadores de los desafíos. La segunda columna muestra la clasificación inicial hecha por el profesor. La tercera columna muestra un número que representa la dificultad del desafío obtenida por el sistema, que puede tomar un valor entre 0 y 2 (ver Figura 29). La cuarta columna representa el valor lingüístico que corresponde a la salida numérica de la anterior columna. La última columna muestra la clasificación realizada por un experto humano, esto es, el profesor de la asignatura, que, tras analizar las respuestas reales dadas por los alumnos a las preguntas, reclasificó estas preguntas de acuerdo a su nivel de dificultad.

El siguiente paso es actualizar el repositorio de desafíos con el nivel de dificultad de cada desafío corregido de acuerdo a lo estimado por el sistema. Así, conforme a la clasificación realizada por el sistema, tres desafíos deberían ser reclasificados (los desafíos con identificador 2, 8 y 10), ya que el nivel de dificultad inicialmente estimado no coincide con la dificultad asignada por el sistema.

A primera vista, si se comparan las clasificaciones hechas por el sistema y por el experto, se puede afirmar que el sistema se comporta adecuadamente y es capaz de realizar la labor de reclasificación sustituyendo al profesor.

Pero para confirmar este hecho se va a validar el sistema contra un grupo de expertos humanos siguiendo la metodología previamente descrita. El grupo de expertos está formado por el profesor de la asignatura y otros dos profesores expertos en la temática.

Aunque el sistema trabaja con tres niveles de dificultad, hay valores para la variable de salida en que la pertenencia no es exclusiva a un conjunto sino a dos. Por este motivo la clasificación se va a desglosar en cinco niveles, incluyendo aquellas regiones difusas entre dos niveles: *fácil*, *entre fácil* y *asequible*, *asequible*, *entre asequible* y *difícil*, *difícil*. Esto viene motivado también por las dudas que surgen a los expertos a la hora de clasificar una pregunta analizando las respuestas, que a veces no es clara.

Identificador de desafío	Clasificación inicial del profesor	Sistema Experto Genético Difuso (Salida numérica)	Sistema Experto Genético Difuso (Valor lingüístico)	Experto humano
1	Asequible	1,08	Asequible	Asequible
2	Asequible	1,51	Difícil tirando a asequible	Difícil tirando a asequible
3	Fácil	0,38	Fácil	Fácil
4	Fácil	0,52	Fácil tirando a asequible	Fácil
5	Asequible	1,07	Asequible	Asequible
6	Difícil	1,48	Difícil tirando a asequible	Difícil
7	Fácil	0,37	Fácil	Fácil
8	Fácil	1,08	Asequible	Asequible
9	Difícil	1,68	Difícil	Difícil
10	Difícil	1,32	Asequible tirando a difícil	Asequible
11	Difícil	1,62	Difícil	Difícil
12	Difícil	1,55	Difícil tirando a asequible	Asequible tirando a difícil

Tabla 11. Clasificación de los desafíos del caso de estudio 1 de acuerdo al sistema experto y al experto humano

La Tabla 12 recopila las clasificaciones realizadas por el sistema experto y el grupo de expertos dando lugar a la base de datos de validación. A partir de la base de datos de validación se crean las tablas de contingencia que representan los acuerdos y desacuerdos entre los expertos (ver Tabla 13). Para medir el nivel de acuerdo o desacuerdo entre los pares de expertos se utiliza la medida de *kappa ponderada*. Los pesos que se aplican según el nivel de acuerdo o desacuerdo están indicados en la Tabla 14. Los resultados de la medida de *kappa* se muestran en la Tabla 15, donde de un vistazo se puede observar el consenso que existe entre pares de expertos. En la Tabla 15 se muestra una escala comúnmente citada que indica el nivel de acuerdo que corresponde a diferentes rangos del valor de *Kappa* (Viera y Garrett 2005).

El valor de *Kappa* indica que el nivel de acuerdo entre el experto 1 y el experto 2 es casi perfecto. Sin embargo el nivel de acuerdo entre los expertos 1 y 3 y entre los expertos 2 y 3 se encuentra en el límite entre un acuerdo substancial y un acuerdo casi perfecto. De cualquier manera se encuentra un nivel alto de acuerdo entre los expertos. Respecto al nivel de acuerdo del sistema experto con cada uno de los expertos, éste varía desde un acuerdo substancial (valor de *Kappa* igual a 0,747) hasta un acuerdo casi perfecto (valor de *Kappa* igual a 0,947).

En principio, estos parecen buenos resultados pero se va a comprobar si al introducir al sistema experto dentro del grupo de expertos disminuye el nivel de consenso en el grupo. Para ello se calcula por un lado la medida de grupo de los tres expertos, obteniéndose un valor  $P_n$  igual a 0,840, y por otro lado la medida del acuerdo del sistema experto con los expertos humanos,  $P_0$ , que toma un valor igual a 0,844. Así, el índice de Williams es 1,005, lo que significa que introducir al sistema experto en el grupo de expertos no supondría una disminución del consenso. Se confirma de esta manera que el comportamiento del sistema

experto es el adecuado, y puede sustituir en la tarea de clasificación de la dificultad de los desafíos al experto humano.

<b>Identificador del desafío</b>	<b>Sistema Experto</b>	<b>Experto 1</b>	<b>Experto 2</b>	<b>Experto 3</b>
1	Asequible	Asequible	Asequible	Asequible
2	Asequible-Difícil	Asequible-Difícil	Difícil	Difícil
3	Fácil	Fácil	Fácil	Fácil
4	Fácil-Asequible	Fácil	Fácil	Fácil-Asequible
5	Asequible	Asequible	Asequible	Asequible
6	Asequible-Difícil	Difícil	Difícil	Asequible-Difícil
7	Fácil	Fácil	Fácil	Fácil
8	Asequible	Asequible	Asequible	Asequible
9	Difícil	Difícil	Difícil	Difícil
10	Asequible-Difícil	Asequible	Asequible	Asequible-Difícil
11	Difícil	Difícil	Difícil	Difícil
12	Asequible-Difícil	Asequible-Difícil	Difícil	Asequible-Difícil

Tabla 12. Base de datos de validación (caso de estudio 1)

<b>E2\E1</b>	<b>F</b>	<b>F-A</b>	<b>A</b>	<b>A-D</b>	<b>D</b>	<b>Total</b>
<b>F</b>	3					3
<b>F-A</b>						
<b>A</b>			4			4
<b>A-D</b>						
<b>D</b>				2	3	5
<b>Total</b>	3		4	2	3	12

<b>E3\E1</b>	<b>F</b>	<b>F-A</b>	<b>A</b>	<b>A-D</b>	<b>D</b>	<b>Total</b>
<b>F</b>	2					2
<b>F-A</b>	1					1
<b>A</b>			3			3
<b>A-D</b>			1	1	1	3
<b>D</b>				1	2	3
<b>Total</b>	3		4	2	3	12

<b>E3\E2</b>	<b>F</b>	<b>F-A</b>	<b>A</b>	<b>A-D</b>	<b>D</b>	<b>Total</b>
<b>F</b>	2					2
<b>F-A</b>	1					1
<b>A</b>			3			3
<b>A-D</b>			1		2	3
<b>D</b>					3	3
<b>Total</b>	3		4		5	12

<b>SE\E1</b>	<b>F</b>	<b>F-A</b>	<b>A</b>	<b>A-D</b>	<b>D</b>	<b>Total</b>
<b>F</b>	2					2
<b>F-A</b>	1					1
<b>A</b>			3			3
<b>A-D</b>			1	2	1	4
<b>D</b>					2	2
<b>Total</b>	3	0	4	2	3	12

<b>SE\E2</b>	<b>F</b>	<b>F-A</b>	<b>A</b>	<b>A-D</b>	<b>D</b>	<b>Total</b>
<b>F</b>	2					2
<b>F-A</b>	1					1
<b>A</b>			3			3
<b>A-D</b>			1		3	4
<b>D</b>					2	2
<b>Total</b>	3		4		5	12

<b>SE\E3</b>	<b>F</b>	<b>F-A</b>	<b>A</b>	<b>A-D</b>	<b>D</b>	<b>Total</b>
<b>F</b>	2					2
<b>F-A</b>		1				1
<b>A</b>			3			3
<b>A-D</b>				3	1	4
<b>D</b>					2	2
<b>Total</b>	2	1	3	3	3	12

Tabla 13. Tablas de contingencia de los pares de expertos (caso de estudio 1)

	Fácil	Fácil-Asequible	Asequible	Asequible-Difícil	Difícil
Fácil	1	0,75	0,5	0,25	0
Fácil-Asequible	0,75	1	0,75	0,5	0,25
Asequible	0,5	0,75	1	0,75	0,5
Asequible-Difícil	0,25	0,5	0,75	1	0,75
Difícil	0	0,25	0,5	0,75	1

Tabla 14. Matriz de pesos del nivel de acuerdo/desacuerdo (caso de estudio 1)

	Experto 1	Experto 2	Experto 3	Sistema Experto
Experto 1	--	0,901	0,805	0,837
Experto 2	0,901	--	0,813	0,747
Experto 3	0,805	0,813	--	0,947
Sistema Experto	0,837	0,747	0,947	--

Tabla 15. Tabla de pares con los valores de Kappa ponderada (caso de estudio 1)

Kappa	Nivel de acuerdo
< 0	Menor que el acuerdo por casualidad
0	Acuerdo por casualidad
0,01-0,20	Acuerdo leve
0,21-0,40	Acuerdo regular
0,41-0,60	Acuerdo moderado
0,61-0,80	Acuerdo substancial
0,81-0,99	Acuerdo casi perfecto

Tabla 16. Interpretación del valor de *Kappa*

## 2.2.2 Caso de estudio 2

Para el caso de estudio 2 se tiene 24 desafíos con 1016 patrones de respuesta. La Tabla 17 muestra los juegos de intervalos que el algoritmo genético encuentra para caracterizar los distintos niveles de dificultad.

<i>Nivel Difícil (338 patrones)</i>				<i>Nivel Moderado (338 patrones)</i>				<i>Nivel Fácil (340 patrones)</i>			
<i>Tiempo</i>	<i>Nota</i>	<i>Accesos</i>	<i>Fitness</i>	<i>Tiempo</i>	<i>Nota</i>	<i>Accesos</i>	<i>Fitness</i>	<i>Tiempo</i>	<i>Nota</i>	<i>Accesos</i>	<i>Fitness</i>
[5,20]	[50,69]	[1,2]	0.213	[3,18]	[67,87]	[1,3]	0.335	[0,23]	[80,100]	[1,2]	0.48
[3,18]	[23,40]	[1,2]	0.200	[2,25]	[80,100]	[1,2]	0.183	[0,15]	[54,86]	[1,2]	0.254
[4,19]	[0,17]	[1,3]	0.118	[3,18]	[43,67]	[1,3]	0.157				
[6,29]	[73,89]	[1,3]	0.106								

Tabla 17. Juegos de intervalos encontrados por el algoritmo genético para el caso de estudio 2

Aunque hay un mayor número de patrones que en el caso de estudio 1, la búsqueda no es más sencilla porque en este caso los patrones de respuesta están más dispersos y es más difícil detectar las distintas agrupaciones de patrones. Esta dispersión también se refleja en que se está encontrando un mayor número de juegos de intervalos.

Por ello, como se observa en la Figura 33, la cantidad de conjuntos difusos y sus valores lingüísticos aumenta para las variables *nota* (grade) y *tiempo*. Más aún, la cantidad de reglas asciende a 96 reglas. En este caso el factor de confianza asignado a las reglas tiene un papel importante ya que hay diferentes reglas con mismo antecedente para clasificar distintas clases, como se observa en la matriz F.A.M (la Figura 34 muestra la matriz F.A.M para accesos bajos (L) y la Figura 35 muestra la matriz F.A.M. para accesos altos (H)).

Aquellas respuestas con notas *muy muy baja* (VVL) o *muy baja* (VL), como cabe esperar, corresponden a desafíos difíciles. Cuando los accesos son *bajos* (L), el tiempo es *muy muy muy bajo* (VVVL) y las notas son *muy muy altas* (VVH), se trata de patrones correspondientes a preguntas fáciles, como también se espera. Para las notas *muy muy altas* (VVH), diferentes tiempos y accesos *bajos* (L), se tienen patrones de respuesta correspondientes a preguntas fáciles y asequibles. Sin embargo, en este caso el factor de confianza es el doble para las reglas de la clase fácil. Por ello, si se tiene un patrón que tenga notas *muy muy altas* (VVH), será más probable que sea un patrón fácil que un patrón asequible.

Aunque la nota sea *muy muy alta* (VVH), si el tiempo empleado es *muy alto* (VH), el patrón se clasificaría como asequible.

Las casillas del centro de la matriz muestran que para el mismo antecedente se tienen distintas clasificaciones. Según los tiempos aumentan y las notas disminuyen se observa una tendencia a que vayan desapareciendo las reglas de la clase fácil y aparezcan las reglas de la clase difícil.

Para el caso de que el número de accesos sea *alto* (H), caso representado en la matriz F.A.M de la Figura 35, los patrones corresponderán a la clase asequible o difícil. Como es de esperar normalmente los alumnos resuelven los desafíos fáciles en cuanto los leen, con lo cual el número de accesos para los desafíos fáciles suele ser bajo.

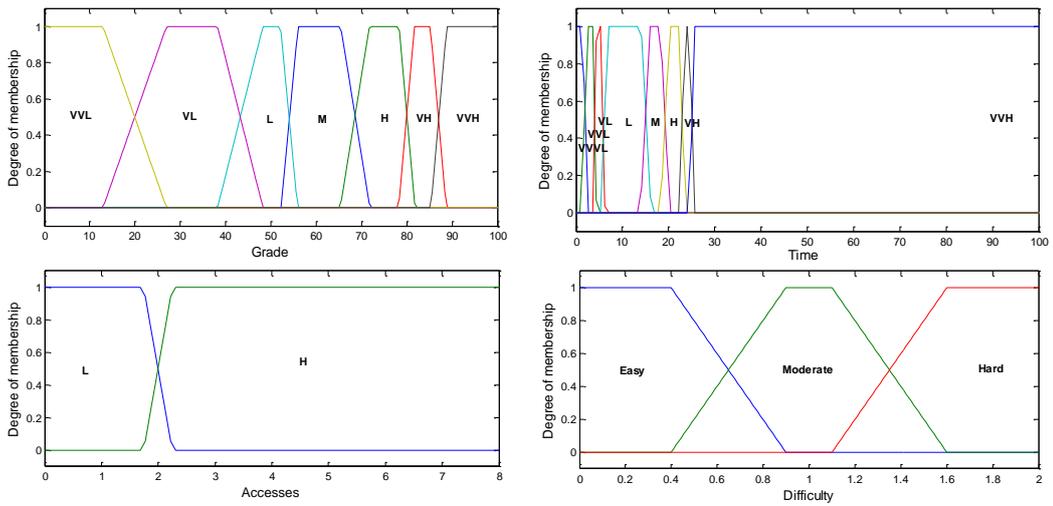


Figura 33. Conjuntos difusos de las variables de entrada y salida

<i>Time</i>							<i>Accesses (Low)</i>	
VVH					Difícil	Difícil		
VH					Difícil	Difícil Aseq.	Aseq.	
H					Difícil	Difícil Aseq. Fácil	Aseq. Fácil	
M	Difícil	Difícil	Difícil Aseq.	Difícil Aseq.	Difícil Aseq.	Difícil Aseq. Fácil	Aseq. Fácil	
L	Difícil	Difícil	Difícil Aseq.	Difícil Fácil	Difícil Aseq. Fácil	Difícil Aseq. Fácil	Aseq. Fácil	
VL	Difícil	Difícil	Difícil Aseq.	Difícil Fácil	Aseq. Fácil	Aseq. Fácil	Aseq. Fácil	
VVL		Difícil	Aseq.	Aseq. Fácil	Aseq. Fácil	Aseq. Fácil	Aseq. Fácil	
VVVL				Aseq.	Aseq.	Fácil	Fácil	
	VVL	VL	L	M	H	VH	VVH	<i>Grade</i>

Figura 34. Representación F.A.M de las reglas (Accesos=L)

<i>Time</i>		<i>Accesses (High)</i>					<i>Grade</i>	
		VVL	VL	L	M	H		VH
VVH						Difícil	Difícil	
VH						Difícil	Difícil	
H						Difícil	Difícil	
M	Difícil		Aseq.	Aseq.		Difícil	Difícil	
L	Difícil		Aseq.	Aseq.		Difícil	Difícil	
VL	Difícil		Aseq.	Aseq.		Aseq.	Aseq.	
VVL			Aseq.	Aseq.		Aseq.	Aseq.	
VVVL								

Figura 35. Representación F.A.M. de las reglas (Accesos=H)

La Tabla 18 muestra la clasificación inicial de los desafíos realizada por el profesor, la dificultad asignada por el sistema y el valor numérico de salida, así como la dificultad que el profesor (experto humano) asigna una vez analizados los datos relativos a las respuestas de los alumnos. En este caso hay 5 desafíos que deben ser reclasificados (los desafíos con identificadores 2, 8, 11, 19 y 22). Por otro lado se ve que el profesor y el sistema no coinciden en varios casos en la clasificación final de los desafíos. Se va estudiar a continuación el nivel de acuerdo entre el profesor y otros dos expertos así como el nivel de acuerdo entre el sistema y los expertos humanos para valorar si el sistema experto se puede asimilar a un experto humano más.

Identificador de desafío	Clasificación inicial del profesor	Sistema Experto Genético Difuso (salida numérica)	Sistema Experto Genético Difuso (valor lingüístico)	Experto Humano
1	Fácil	0,41	Fácil	Fácil
2	Difícil	1,01	Asequible	Asequible
3	Difícil	1,56	Difícil tirando a asequible	Difícil
4	Asequible	1,01	Asequible	Asequible
5	Asequible	1	Asequible	Asequible
6	Fácil	0,33	Fácil	Fácil
7	Fácil	0,41	Fácil	Fácil
8	Difícil	1,1	Asequible	Difícil tirando a asequible
9	Difícil	1,72	Difícil	Difícil
10	Fácil	0,53	Fácil tirando a asequible	Fácil
11	Asequible	1,75	Difícil	Difícil
12	Asequible	1	Asequible	Asequible
13	Difícil	1,56	Difícil	Difícil
14	Asequible	1	Asequible	Asequible
15	Fácil	0,41	Fácil	Fácil
16	Asequible	1	Asequible	Asequible
17	Difícil	1,62	Difícil	Difícil
18	Fácil	0,39	Fácil	Fácil
19	Asequible	1,57	Difícil	Difícil
20	Fácil	0,53	Fácil tirando a asequible	Fácil
21	Fácil	1,02	Asequible	Fácil
22	Asequible	1,56	Difícil	Difícil
23	Difícil	1,58	Difícil	Difícil
24	Difícil	1,48	Difícil tirando a asequible	Difícil

Tabla 18. Clasificación de los desafíos de acuerdo al sistema experto y al experto humano

La Tabla 19 recoge las distintas clasificaciones realizadas por el sistema experto y los expertos humanos. Es por tanto la base de datos de validación, punto de partida para crear las tablas de contingencia entre expertos. El nivel de acuerdo o desacuerdo entre los pares de expertos calculado a partir de la medida de kappa ponderada se muestra en la Tabla 21.

En esta tabla se puede observar que el nivel de acuerdo del sistema experto con cada uno de los expertos es un acuerdo casi perfecto según la Tabla 16, especialmente con el experto 3.

Más aún, el nivel de acuerdo del sistema con los expertos ( $P_0=0,840$ ) es superior al consenso existente en el grupo de expertos ( $P_n=0,827$ ) con lo que el índice de Willians es ligeramente superior a 1 ( $I_N=0,015$ ), con lo que se puede considerar al sistema experto como otro experto humano más.

Identificador de desafío	Sistema Experto	Experto 1	Experto 2	Experto 3
1	Fácil	Fácil	Fácil	Fácil
2	Asequible	Asequible	Asequible	Asequible
3	Asequible-Difícil	Difícil	Difícil	Asequible-Difícil
4	Asequible	Asequible	Asequible-Difícil	Asequible-Difícil
5	Asequible	Asequible	Asequible	Asequible
6	Fácil	Fácil	Fácil	Fácil
7	Fácil	Fácil	Fácil	Fácil
8	Asequible	Asequible-Difícil	Asequible	Difícil
9	Difícil	Difícil	Difícil	Difícil
10	Facil-Asequible	Fácil	Fácil-Asequible	Fácil
11	Difícil	Difícil	Difícil	Difícil
12	Asequible	Asequible	Asequible	Asequible
13	Difícil	Difícil	Asequible-Difícil	Difícil
14	Asequible	Asequible	Asequible	Asequible
15	Fácil	Fácil	Fácil	Fácil
16	Asequible	Asequible	Asequible-Difícil	Asequible
17	Difícil	Difícil	Difícil	Difícil
18	Fácil	Fácil	Fácil	Fácil
19	Difícil	Difícil	Difícil	Difícil
20	Fácil-Asequible	Fácil	Fácil-Asequible	Fácil
21	Asequible	Fácil	Fácil-Asequible	Fácil-Asequible
22	Difícil	Difícil	Asequible-Difícil	Difícil
23	Difícil	Difícil	Difícil	Difícil
24	Asequible-Difícil	Difícil	Asequible-Difícil	Asequible-Difícil

Tabla 19. Base de datos de validación para el caso de estudio 2

<b>E2\E1</b>	<b>F</b>	<b>F-A</b>	<b>A</b>	<b>A-D</b>	<b>D</b>	<b>Total</b>
<b>F</b>	5					5
<b>F-A</b>	3					3
<b>A</b>			4	1		5
<b>A-D</b>			2		3	5
<b>D</b>					6	6
<b>Total</b>	8	0	6	1	9	24

<b>E3\E1</b>	<b>F</b>	<b>F-A</b>	<b>A</b>	<b>A-D</b>	<b>D</b>	<b>Total</b>
<b>F</b>	7					7
<b>F-A</b>	1					1
<b>A</b>			5			5
<b>A-D</b>			1		2	3
<b>D</b>				1	7	8
<b>Total</b>	8	0	6	1	9	24

<b>E2\E3</b>	<b>F</b>	<b>F-A</b>	<b>A</b>	<b>A-D</b>	<b>D</b>	<b>Total</b>
<b>F</b>	5					5
<b>F-A</b>	2	1				3
<b>A</b>			4	1	1	6
<b>A-D</b>				2	2	4
<b>D</b>				1	5	6
<b>Total</b>	7	1	4	4	8	24

<b>S1\SE</b>	<b>F</b>	<b>F-A</b>	<b>A</b>	<b>A-D</b>	<b>D</b>	<b>Total</b>
<b>F</b>	5	2	1			8
<b>F-A</b>						0
<b>A</b>			6			6
<b>A-D</b>			1			1
<b>D</b>				2	7	9
<b>Total</b>	5	2	8	2	7	24

<b>S2\SE</b>	<b>F</b>	<b>F-A</b>	<b>A</b>	<b>A-D</b>	<b>D</b>	<b>Total</b>
<b>F</b>	5					5
<b>F-A</b>		2	1			3
<b>A</b>			5			5
<b>A-D</b>			2	1	2	5
<b>D</b>				1	5	6
<b>Total</b>	5	2	8	2	7	24

<b>E3\SE</b>	<b>F</b>	<b>F-A</b>	<b>A</b>	<b>A-D</b>	<b>D</b>	<b>Total</b>
<b>F</b>	5	2				7
<b>F-A</b>			1			1
<b>A</b>			5			5
<b>A-D</b>			1	2		3
<b>D</b>			1		7	8
<b>Total</b>	5	2	8	2	7	24

Tabla 20. Tablas de contingencia de los pares de expertos (caso de estudio 2)

	<b>Experto 1</b>	<b>Experto 2</b>	<b>Experto 3</b>	<b>Sistema Experto</b>
<b>Experto 1</b>	--	0,789	0,886	0,814
<b>Experto 2</b>	0,789	--	0,808	0,848
<b>Experto 3</b>	0,886	0,808	--	0,937
<b>Sistema Experto</b>	0,814	0,848	0,937	--

Tabla 21. Tabla de pares con los valores de Kappa ponderada (caso de estudio 2)



# CAPÍTULO 6

## CONCLUSIONES Y LÍNEAS FUTURAS

En este capítulo se presentan las conclusiones obtenidas en la realización de este trabajo de investigación y se ofrecen unas orientaciones que pueden servir como guía para continuar en un futuro con esta misma línea de investigación y desarrollo.

### 1 Conclusiones

QUESTOURnament es una herramienta telemática que permite el desarrollo de concursos, presentando una serie de desafíos a los alumnos que deben ser resueltos en un tiempo límite. Los desafíos pueden ser propuestos tanto por el profesor como por los alumnos y, en este caso, los propios alumnos deben evaluar a sus compañeros. En todo momento se muestra una tabla de clasificación, construida a partir de la puntuación obtenida por los alumnos tanto al responder desafíos como al proponerlos, lo que marca un fuerte carácter competitivo a la herramienta.

Dado que algunos autores opinan que el éxito de los nuevos sistemas de aprendizaje depende de la capacidad de adaptar el sistema al conocimiento de los estudiantes, dando lugar a los Sistemas Educativos Inteligentes (Ahmad, Basir, y Hassanein 2004) (Brusilovsky y Peylo

2003), el primer objetivo de la tesis ha consistido *en tratar de identificar las posibilidades de los sistemas adaptativos para potenciar las ventajas y mitigar los inconvenientes que existen en los sistemas de aprendizaje competitivo*, como QUESTOURnament. Para ello primeramente se ha realizado un estudio del estado de arte de los sistemas para el aprendizaje competitivo, identificando sus ventajas e inconvenientes, primera contribución de la presente tesis, cuyos resultados han sido publicados en (E. Verdú et al. 2008a) (Verdu y Lorenzo 2010). Algunas de las ventajas encontradas son la mejora de los resultados académicos, el aumento de la motivación o la protección de la propiedad intelectual. Entre las desventajas se encuentra el aumento de estrés y presión o las diferencias en motivación existentes entre los alumnos según su posición en la clasificación. Este estudio se ha completado con un análisis exhaustivo de la herramienta QUESTOURnament. Se ha demostrado que esta herramienta mejora la motivación, los hábitos de estudio y los resultados académicos de los estudiantes (L.M. Regueras et al. 2009) (E. Verdú et al. 2007). Sin embargo, el uso de QUESTOURnament presenta también algunos aspectos negativos propios de las estrategias de aprendizaje competitivo, como el mencionado estrés y las diferencias de motivación según la actuación en el concurso (E. Verdú et al. 2006) (L.M. Regueras, E. Verdú, M.J. Verdú, M.A. Pérez, et al. 2007).

En busca del objetivo anteriormente mencionado, se realiza un estudio del estado de arte de los sistemas de aprendizaje adaptativo, estudiando sus funcionalidades y soluciones técnicas y analizando su efectividad respecto a los resultados académicos y al aumento del grado de satisfacción de los estudiantes. Tras este análisis se puede concluir que estos sistemas sí mejoran los resultados de aprendizaje de forma significativa y aumentan la satisfacción del alumno. El mencionado estudio, segunda contribución de la presente tesis, ha sido publicado en (E. Verdú, L. M. Regueras, M.J. Verdú, et al. 2008). El trabajo se ha complementado con otro estudio de sistemas competitivos y adaptativos publicado en (Verdu et al. 2010). De dicho estudio se concluye que son muy pocos los sistemas competitivos que introducen mecanismos de adaptación y que la mayoría de los que se encuentran en la literatura, consisten en juegos con distintos niveles de dificultad o sistemas que agrupan alumnos con similares capacidades durante el desarrollo de un juego.

Con el fin de alcanzar el objetivo de *proponer un sistema de aprendizaje adaptativo para la herramienta QUESTOURnament*, a partir de los dos estudios anteriores se han podido identificar las posibilidades de los sistemas adaptativos para potenciar las ventajas y mitigar los inconvenientes existentes en los sistemas de aprendizaje competitivo. Entre las potencialidades de los sistemas adaptativos para su aplicación al aprendizaje competitivo, se han identificado como de especial importancia en este trabajo la posibilidad de formar grupos de alumnos con similares capacidades y la posibilidad de presentar preguntas adecuadas al nivel de conocimiento del grupo, aumentando así su grado de competencia percibida.

La propuesta de sistema QUESTOURnament adaptativo, tercera contribución de la tesis, incluye tres estrategias fundamentales: estimar el nivel de dificultad de las preguntas o desafíos presentados en QUESTOURnament, formar grupos de alumnos cuyos resultados de aprendizaje son similares para que compitan entre sí y seleccionar las preguntas adecuadas para cada uno de los grupos de alumnos. Se propone una arquitectura modular consistente en dos módulos o sistemas principales: el *clusterer* o "agrupador" y el propio sistema QUESTOURnament adaptativo. De esta manera, el *clusterer* puede ser utilizado por otros sistemas distintos que el QUESTOURnament que precisen de la funcionalidad de creación de grupos de alumnos en base a múltiples y diversas propiedades. El *clusterer* es capaz de crear grupos siguiendo diversos métodos de clasificación y tomando como datos los resultados de distintas actividades disponibles en Moodle. En lo referente al sistema competitivo QUESTOURnament, la función del *clusterer* es, por tanto, agrupar alumnos con similares resultados de aprendizaje que

competirán entre sí dentro de una misma liga, mientras que el sistema QUESTOURnament adaptativo tiene como función reclasificar los desafíos de acuerdo a la dificultad que encuentran los alumnos al resolverlos así como presentar preguntas adecuadas a los grupos de alumnos que conforman las diferentes ligas del concurso.

Una vez definidas las características del sistema QUESTOURnament adaptativo, el siguiente objetivo de la tesis consiste en *proponer herramientas y métodos que permitan medir objetivamente el nivel de dificultad de los desafíos del sistema QUESTOURnament*. Este objetivo unido a los dos anteriormente mencionados permite cumplir el objetivo principal de la tesis: *propuesta y validación de métodos basados en técnicas de Inteligencia Artificial para la estimación del nivel de dificultad de los desafíos propuestos en el Entorno On-line de Aprendizaje Competitivo QUESTOURnament, que permita el posterior establecimiento de concursos o itinerarios de aprendizaje para grupos de alumnos según su nivel de conocimiento*.

El estudio del estado de arte de los sistemas de aprendizaje adaptativo ha permitido conocer las soluciones técnicas empleadas en los sistemas adaptativos. En concreto, para la estimación del nivel de dificultad de preguntas se emplea en muchas ocasiones la TRI. Sin embargo, el carácter competitivo de la herramienta QUESTOURnament, sus amplias posibilidades de aplicación (distintos tipos de desafíos, diferentes posibles contextos de aplicación, etc.), y su fuerte dependencia con la variable tiempo, no hace posible el aplicar esta técnica comunmente utilizada en los tests adaptativos informatizados. Existen otras soluciones enfocadas a problemas concretos que también han sido estudiadas y que no permiten su directa aplicación al sistema QUESTOURnament.

Por tanto, se propone una nueva solución, basada en algoritmos genéticos y lógica difusa para alcanzar el objetivo de la tesis. Partiendo de la clasificación inicial del nivel de dificultad que realiza el profesor para cada desafío y tomando como entrada los patrones de respuestas de la forma *<tiempo, nota, accesos >* disponibles en la base de datos de hechos, la solución propuesta optimiza la clasificación de los desafíos de manera automática adaptándola a la dificultad real que encuentran los alumnos a la hora de resolverlos. Esta solución se ha ido perfilando en un proceso cíclico de acuerdo a la metodología definida en (Adrion 1993). Tras diseñar, en una primera fase, un algoritmo genético básico, se comprueba que éste debe ser mejorado para garantizar que encuentra y mantiene soluciones diversas, correspondientes a los distintos grupos de alumnos. Así, el algoritmo genético mejorado con técnicas de diversidad (*sharing* e inyección de cromosomas) y con un algoritmo de búsqueda local (algoritmo de escalada), ofrece para cada nivel de dificultad un conjunto de soluciones (intervalos de *tiempo*, *nota* y *accesos*) que lo caracterizan.

Los resultados del algoritmo genético son la base para definir de forma automática los conjuntos difusos y las reglas que rigen el sistema experto difuso que finalmente determina el nivel de dificultad.

La cuarta contribución de la tesis es, por tanto, un sistema híbrido genético-difuso que permite estimar el nivel de dificultad de las preguntas en función del comportamiento de los alumnos al responderlas y, así, reclasificar los desafíos propuestos en la herramienta QUESTOURnament. El sistema ha sido validado contra un grupo de expertos y se puede afirmar que el sistema ha estimado el nivel de dificultad de los desafíos, tal y como lo haría un experto humano. Esta contribución ha sido parcialmente publicada en (E. Verdú, L.M. Regueras, et al. 2010) (E. Verdú, M.J. Verdú, et al. 2010) y es el primer paso en el proceso para convertir a QUESTOURnament en un sistema adaptativo, dado que una buena estimación del nivel de dificultad del material de aprendizaje permite posteriormente una mejor estimación del nivel de conocimiento de los alumnos.

Por último, cabe indicar que esta solución de sistema híbrido genético-difuso podría ser el punto de partida de soluciones adaptativas para otros sistemas de aprendizaje con características competitivas o, simplemente, para muchos sistemas telemáticos de aprendizaje que permiten gran libertad de uso.

## 2 Futuras Líneas de Investigación

Como principales trabajos futuros se propone:

- Estudio de la posible inclusión de más parámetros relacionados con el comportamiento y los perfiles de los estudiantes en los patrones de respuesta de la base de hechos del sistema.
- Profundización en la problemática de los patrones temporales de respuesta. Dado que el comportamiento del alumno no es predecible y no se puede conocer el tiempo que exactamente ha dedicado a resolver un desafío, se debe depurar este aspecto y estudiar si el empleo de técnicas como las de búsqueda y análisis de patrones secuenciales (*mining sequential patterns*) podrían posibilitar una solución más precisa. Ejemplos de estas técnicas se pueden encontrar en (C. -C. Yu y Y. -L. Chen 2005) (Y. Huang, L. Zhang, y P. Zhang 2008) y, aplicadas en concreto al análisis de interacción en sistemas e-learning, en (Pahl 2004) (W. Wang et al. 2004). Por otro lado, en el sistema actual se podría introducir un factor de confianza para gestionar estos datos de la base de hechos que no son totalmente fiables, además del factor de confianza de las reglas existente.
- Optimización del algoritmo genético incorporando, por ejemplo, nuevos operadores de cruce y mutación diseñados específicamente para el presente problema.
- Analizar cómo aplicar las distintas dificultades de los desafíos en la estimación del nivel de conocimiento del alumno.
- Estudio de la posible aplicación de los resultados del generador del modelo difuso a la clasificación y el modelado de estudiantes. Las pruebas realizadas muestran que un nivel de dificultad puede estar caracterizado por diferentes conjuntos difusos. Como se adelantaba en el apartado 4.1.3 del capítulo 4, que muestra cómo el algoritmo genético diverso encuentra varios juegos de intervalos para caracterizar los distintos niveles de dificultad, esto puede deberse a los diferentes comportamientos de los estudiantes cuando abordan un problema, así como a su personalidad y su nivel de conocimiento. Por ello, sería interesante estudiar la posibilidad de usar los conjuntos difusos resultantes para detectar y clasificar grupos de estudiantes en función de sus niveles de conocimiento u otras características.
- Validación del sistema propuesto en otros entornos. Hasta el momento el sistema híbrido genético difuso que permite reclasificar las preguntas ha sido probado con datos de asignaturas de titulaciones de ingeniería. Dado que el sistema QUESTOURnament permite el desarrollo de concursos de cualquier área de conocimiento, puede ser conveniente probar el sistema con datos relativos a asignaturas de otras áreas. Por otro lado, tal y como afirma (O'Keefe, Balci, y E.P. Smith 1986), un prototipo con un rendimiento medio y muestras de que la aproximación básica es correcta, es aceptable para ser utilizado en entornos reales. Por tanto, no siendo QUESTOURnament una aplicación crítica (como las del campo de la medicina, por ejemplo), es posible emplear el sistema global QUESTOURnament adaptativo en cursos reales y validarlo según se utiliza. Así se puede conseguir

recopilar más datos y poder analizar mejor los resultados. Experimentar con el sistema en cursos reales ayudará sin duda a ajustarlo y mejorarlo y permitirá una evaluación completa del mismo, en la que el usuario final juega un papel importante.



## BIBLIOGRAFÍA

- Adrion, W.R. 1993. Research methodology in software engineering: summary of the Dagstuhl workshop on future directions on software engineering. *Software Engineering Notes* 18, n°. 1: 36-37.
- Ahmad, A.-R., O. Basir, y K. Hassanein. 2004. Adaptive User Interfaces for Intelligent E-Learning: Issues and Trends. En *Proceedings of the International Conference on Electronic Business (ICEB'04)*.
- Albacete, P.L., y K.A. VanLehn. 2000. Evaluating the Effectiveness of a Cognitive Tutor for Fundamental Physics Concepts. En *Proceedings of the 22nd Annual Meeting of the Cognitive Science Society*, 25-30.
- Amabile, T.M. 1982. Children's Artistic Creativity: detrimental effects of competition in a field setting. *Personality and Social Psychology Bulletin* 8, n°. 3: 573-578.
- Andersen, E.B. 1997. The rating scale model. En *Handbook of Modern Item Response Theory*, 67-84. New York: Springer-Verlag.
- Anderson, J.R. 2006. On Cooperative and Competitive Learning in the Management Classroom. *Mountain Plains Journal of Business and Economics, Pedagogy* 7: 1-10.
- Bäck, Thomas, David B. Fogel, Darrell Whitley, y Peter J. Angeline. 2000. Mutation operators. En *Evolutionary Computation 1. Basic Algorithms and Operators*, ed. Thomas Bäck, David B. Fogel, y Zbigniew Michalewicz, 237-255. Bristol and

- Philadelphia: Institute of Physics Publishing.
- Barbero, M.I. 1999. Desarrollos recientes de los modelos psicométricos de la teoría de respuesta a los ítems. *Psicothema* 11, nº. 1: 195-210.
- Becker, K. 2001. Teaching with games: The minesweeper and asteroids experience. *J. Computing Small Colleges* 17, nº. 2: 23-33.
- Bendavid, A. 2008. Comparison of classification accuracy using Cohen's Weighted Kappa. *Expert Systems with Applications* 34, nº. 2: 825-832. doi:10.1016/j.eswa.2006.10.022.
- Biswas, R. 1995. An application of fuzzy sets in students' evaluation. *Fuzzy Sets and Systems* 74: 187-194.
- Bonwell, C.C., y J.A. Eison. 1991. *Active Learning: Creating Excitement in the Classroom*, ASHEERIC Higher Education Report. Washington, DC: George Washington University.
- Booker, L.B., D.B. Fogel, D. Whitley, P.J. Angeline, y A.E. Eiben. 2000. Recombination. En *Evolutionary Computation 1. Basic Algorithms and Operators*, ed. T. Bäck, D.B. Fogel, y Z. Michalewicz, 256-307. Bristol and Philadelphia: Institute of Physics Publishing.
- Brenner, H., y U. Kliebsch. 1996. Dependence of Weighted Kappa Coefficients on the Number of Categories. *Epidemiology* 7, nº. 2: 199-202.
- Brugos, J.A.L., A. Neira, V. García, y A. Alguero. 1998. Methodology to DMITS: Distributed Multimedia Intelligent Tutoring Systems. En *Proceedings on the 4th EATA International Conference on Networking Entities, NETIES'98 (Networking for the Millennium)*. Leeds, West Yorkshire, UK. <http://www.lmu.ac.uk/ies/conferences/brugos.html>.
- Brule, J.F. 1985. Fuzzy Systems - A Tutorial. <http://www.austinlinks.com/Fuzzy/tutorial.html>.
- Brusilovsky, P. 1998. Methods and Techniques of Adaptive Hypermedia, Adaptive Hypertext and Hypermedia. En *Adaptive Hypertext and Hypermedia*, 1-43. London: Kluwer Academic Publishers.
- Brusilovsky, P., y C. Peylo. 2003. Adaptive and Intelligent Web-based Educational Systems. *International Journal of Artificial Intelligence in Education* 13: 156-169.
- Bryndum, S., y J.A. Montes. 2005. La motivación en los entornos telemáticos. *RED Revista de Educación a Distancia*, nº. 13. <http://www.um.es/ead/red/13>.
- Bures, E.M., P.C Abrami, y C. Amundsen. 2000. Student motivation to learn via computer conferencing. *Research in Higher Education* 41, nº. 5: 593-621.
- Burke, E.K., S. Gustafson, y G. Kendall. 2004. Diversity in Genetic Programming: An Analysis of Measures and Correlation With Fitness. *IEEE Transactions on Evolutionary Computation* 8, nº. 1 (2): 47-62. doi:10.1109/TEVC.2003.819263.
- Capuano, N., M. Gaeta, A. Micarelli, y E. Sangineto. 2003. An Intelligent Web Tutoring System for Learning Personalization and Semantic Web Compatibility. En *Proceedings of the 11th International PEG Conference "Powerful ICT for Teaching and Learning"*. San Petersburgo, Rusia.
- Coe, R. 2002. It's the Effect Size, Stupid: What effect size is and why it is important. En *Annual Conference of the British Educational Research Association*.
- Cohen, J. 1992. A power primer. *Psychological Bulletin*.

- Conejo, R., E. Guzmán, E. Millán, M. Trella, J.L. Pérez-de-la-Cruz, y A. Ríos. 2004. SIETTE: A Web-Based Tool for Adaptive Testing. *International Journal of Artificial Intelligence in Education*, n°. 14: 29-61.
- Conlan, O. 2005. The Multi-Model, Metadata driven approach to Personalised eLearning Services. PhD Thesis, Dublín, Irlanda: University of Dublin, Trinity College.
- Conti, R., M.A. Collins, y M.L. Picariello. 2001. The impact of competition on intrinsic motivation and creativity: considering gender, gender segregation and gender role orientation. *Personality and Individual Differences* 30: 1273-1289.
- Cordón, O. 2004. Ten years of genetic fuzzy systems: current framework and new trends. *Fuzzy Sets and Systems* 141, n°. 1 (1): 5-31. doi:10.1016/S0165-0114(03)00111-8.
- Cordón, O., M.J. del Jesús, y F. Herrera. 1998. Genetic learning of fuzzy rule-based classification systems cooperating with fuzzy reasoning methods. *International Journal of Intelligent Systems* 13: 1025-1053.
- Chaiyaratana, N., T. Piroonratana, y N. Sangkawelert. 2006. Effects of diversity control in single-objective and multi-objective genetic algorithms. *Journal of Heuristics* 13, n°. 1 (12): 1-34. doi:10.1007/s10732-006-9003-1.
- Chang, L.J., J.C. Yang, F.Y. Yu, y T.W. Chan. 2003. Development and Evaluation of Multiple Competitive Activities in a Synchronous Quiz Game System. *Journal of Innovations in Education and Training International* 40, n°. 1: 16-26.
- Chang, P.-C., W.-H. Huang, y C.-J. Ting. 2010. Dynamic diversity control in genetic algorithm for mining unsearched solution space in TSP problems. *Expert Systems with Applications* 37, n°. 3 (3): 1863-1878. doi:10.1016/j.eswa.2009.07.066.
- Chang, W.-C., y H.-C. Yang. 2009. Applying IRT to Estimate Learning Ability and K-means Clustering in Web based Learning. *Journal of Software* 4, n°. 2: 167-174.
- Chang, Y.-H., T.-Y. Lu, y R.-J. Fang. 2007. An Adaptive E-Learning System Based on Intelligent Agents. En *Proceedings of the 6th WSEAS International Conference on Applied Computer Science*, 200-205.
- Chen, C. 2008. Intelligent web-based learning system with personalized learning path guidance. *Computers & Education* 51, n°. 2: 787-814. doi:10.1016/j.compedu.2007.08.004.
- Chen, C.-M., H.-M. Lee, y Y.-H. Chen. 2005. Personalized e-learning system using Item Response Theory. *Computers & Education* 44, n°. 3: 237-255.
- Chen, G., C. Chang, y C. Wang. 2008. Ubiquitous learning website: Scaffold learners by mobile devices with information-aware techniques. *Computers & Education* 50, n°. 1: 77-90. doi:10.1016/j.compedu.2006.03.004.
- Chen, Y.-F., S.-B. Chang, C.-C. Liu, T.-W. Chan, M.-H. Yu, y Y.-C. Lu. 2005. Elementary Science Classroom Learning with Wireless Response Devices - Implementing Active and Experiential Learning. En *IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE'05)*, 96-103. Tokushima, Japan. doi:10.1109/WMTE.2005.22. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1579244>.
- Chen, Z.-H., y T.-W. Chan. 2008. Learning by Substitutive Competition: Nurturing My-Pet for Game Competition Based on Open Learner Model. En *Proceedings of*

- the Second IEEE International Conference on Digital Games and Intelligent Toys Based Education (DIGITEL 2008)*, 124-131.
- Chou, C.-Y., T.-W. Chanb, y C.-J. Lin. 2003. Redefining the learning companion: the past, present, and future of educational agents. *Computers & Education* 40, n° 3: 255-269. doi:10.1016/S0360-1315(02)00130-6.
- Chu, H., T. Chen, C. Lin, M. Liao, y Y. Chen. 2009. Development of an adaptive learning case recommendation approach for problem-based e-learning on mathematics teaching for students with mild disabilities. *Expert Systems with Applications* 36, n° 3: 5456-5468. doi:10.1016/j.eswa.2008.06.140.
- Chu, K., M. Chang, y Y. Hsia. 2004. Stimulating students to learn with accuracy counter based on competitive learning. En *Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT'04)*, 786-788. IEEE Computer Society.
- Dadone, P. 2001. Design Optimization of Fuzzy Logic Systems Electrical and Computer Engineering, Dissertation PhD. Blacksburg, Virginia, U.S.A.: Department of Electrical and Computer Engineering, Faculty of the Virginia Polytechnic Institute and State University. <http://scholar.lib.vt.edu/theses/available/etd-05262001-035320>.
- Dagger, D., O. Conlan, y V. Wade. 2005. Fundamental Requirements of Personalised eLearning Development Environments. En *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, ed. G. Richards, 2746-2754. Chesapeake, VA: AACE.
- Dale, E. 1969. *Audio-Visual Methods in Teaching*. New York: The Dryden Press.
- Deb, K. 2001. *Multi-objective optimization using evolutionary algorithms*. Chichester: John Wiley & Sons.
- Diaz-Gomez, P.A., y D.F. Hougen. 2006. Pseudo Random Generation of the Initial Population in Genetic Algorithms: an Experiment. En . University of Oklahoma Computer Science Graduate Student Association.
- Fan, W., E.A. Fox, P. Pathak, y H. Wu. 2004. The Effects of Fitness Functions on Genetic Programming-Based Ranking Discovery for Web Search. *Journal of the American Society for Information Science and Technology* 55, n° 7: 628-636.
- Fasli, M., y M. Michalakopoulos. 2005. Supporting Active Learning through Game-Like Exercises. En *Fifth IEEE International Conference on Advanced Learning Technologies (ICALT'05)*, 730-734. Kaohsiung, Taiwan. doi:10.1109/ICALT.2005.247. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1508802>.
- Felder, R.M., G.N. Felder, y E.J. Dietz. 1998. A longitudinal study of engineering student performance and retention. V. Comparisons with traditionally-taught students. *Journal of Engineering Education* 87, n° 4: 469-480.
- Franken, R.E., y D.J. Brown. 1995. Why do people like competition? The motivation for winning, putting forth effort, improving one's performance, performing well, being instrumental, and expressing forceful/aggressive behavior. *Personality and Individual Differences* 19, n° 2: 175-184.
- Frees, S., y G.D. Kessler. 2004. Developing collaborative tools to promote communication and active learning in academia. En *Proceedings of the 34th Annual Conference Frontiers in Education (FIE'04)*, 3:S3B/20 - S3B/25.

- Piscataway, NJ: IEEE.
- Frías Navarro, M.D., J.P. Llobell, y J.F. García Pérez. 2000. Tamaño del efecto del tratamiento y significación estadística. *Psicothema* 12, n°. 2: 236-240.
- Fuentes, C., M.J. Carrión, C. Arana, J.G. Boticario, C. Barrera, O.C. Santos, J.A. Schmidt, et al. 2005. *aLFanet D66 – Evaluation Results, Project Deliverable Report*. Project Deliverable Report. [http://adenu.ia.uned.es/alfanet/reports/ALFANET\\_D66.pdf](http://adenu.ia.uned.es/alfanet/reports/ALFANET_D66.pdf).
- Gen, M. 1997. *Genetic algorithms and engineering design*. New York: Wiley.
- Gonzalez, C., J. Burguillo, y M. Llamas. 2006. A Qualitative Comparison of Techniques for Student Modeling in Intelligent Tutoring Systems. En *Proceedings. Frontiers in Education. 36th Annual Conference*, 13-18. San Diego, CA, USA. doi:10.1109/FIE.2006.322537. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4117043>.
- Gündoğan, K.K., B. Alataş, y A. Karci. 2004. Mining Classification Rules by Using Genetic Algorithms with Non-random Initial Population and Uniform Operator. *Turkish Journal of Electrical Engineering & Computer Sciences* 12, n°. 1: 43-52.
- Guzman, E., R. Conejo, y J.L. Perez-de-la-Cruz. 2007. Improving Student Performance Using Self-Assessment Tests. *IEEE Intelligent Systems* 22, n°. 4: 46-52. doi:10.1109/MIS.2007.71.
- Hatzilygeroudis, I., C. Giannoulis, y C. Koutsojannis. 2005. Combining Expert Systems and Adaptive Hypermedia Technologies in a Web Based Educational System. En *Fifth IEEE International Conference on Advanced Learning Technologies (ICALT'05)*, 249-253. Kaohsiung, Taiwan. doi:10.1109/ICALT.2005.86. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1508666>.
- Hernandez, D., J. Asensio, y Y. Dimitriadis. 2006. Collaborative Learning Strategies and Scenario-based Activities for Understanding Network Protocols. En *Proceedings. Frontiers in Education. 36th Annual Conference*, 19-24. San Diego, CA, USA: IEEE. doi:10.1109/FIE.2006.322584. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4117080>.
- Hibou, M., y J.-M. Labat. 2004. Embedded Bayesian network student models. En *Information Technology Based Proceedings of the Fifth International Conference on Higher Education and Training, 2004. ITHET 2004.*, 468-472. Istanbul, Turkey. doi:10.1109/ITHET.2004.1358218. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1358218>.
- Hiroyasu, T., K. Kobayashi, M. Nishioka, y M. Miki. 2008. Diversity Maintenance Mechanism for Multi-Objective Genetic Algorithms using Clustering and Network Inversion. En *Parallel Problem Solving from Nature – PPSN 2008*, ed. G. Rudolph, T. Jansen, S. Lucas, C. Poloni, y N. Beume, 5199:722-732. Berlin, Heidelberg: Springer Berlin Heidelberg. <http://www.springerlink.com/index/10.1007/978-3-540-87700-4>.
- Hislop, G.W. 1999. Anytime, Anyplace Learning in an Online Graduate Professional Degree Program. *Group Decision and Negotiation* 8, n°. 5: 385-390.
- Huang, Y., L. Zhang, y P. Zhang. 2008. A Framework for Mining Sequential Patterns from Spatio-Temporal Event Data Sets. *IEEE Transactions on Knowledge and Data Engineering* 20, n°. 4: 433-448. doi:10.1109/TKDE.2007.190712.

- Hung, J.C., L.J. Lin, W.-C. Chang, T.K. Shih, H.-H. Hsu, H.-B. Chang, H.-P. Chang, y K.-H. Huang. 2004. A cognition assessment authoring system for e-learning. En *24th International Conference on Distributed Computing Systems Workshops, 2004. Proceedings.*, 262-267. Hachioji, Tokyo, Japan. doi:10.1109/ICDCSW.2004.1284041.  
<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1284041>.
- Hwang, G. 2003. A conceptual map model for developing intelligent tutoring systems. *Computers & Education* 40, n°. 3: 217-235. doi:10.1016/S0360-1315(02)00121-5.
- Hwang, G., P. Yin, T. Wang, J. Tseng, y G. Hwang. 2008. An enhanced genetic approach to optimizing auto-reply accuracy of an e-learning system. *Computers & Education* 51, n°. 1 (8): 337-353. doi:10.1016/j.compedu.2007.05.014.
- Hwang, G.-L. 2007. Gray Forecast Approach for Developing Distance Learning and Diagnostic Systems. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)* 37, n°. 1: 98-108. doi:10.1109/TSMCC.2006.876062.
- Hwang, H.-S. 1998. Control Strategy for Optimal Compromise between Trip Time and Energy Consumption in a High-Speed Railway. *IEEE Transactions on Systems, Man and Cybernetics* 28, n°. 6: 791-802.
- Jantzen, J. 1998. Design Of Fuzzy Controllers. Tech. report no 98-E 864 (design). Technical University of Denmark, Department of Automation.
- del Jesus, M.J., P. Gonzalez, F. Herrera, y M. Mesonero. 2007. Evolutionary Fuzzy Rule Induction Process for Subgroup Discovery: A Case Study in Marketing. *IEEE Transactions on Fuzzy Systems* 15, n°. 4 (8): 578-592. doi:10.1109/TFUZZ.2006.890662.
- Johnson, D.P., y R.T. Johnson. 1999. *Learning together and alone: cooperative, competitive and individualistic learning*. 5° ed. Boston: Allyn & Bacon.
- Jong, B.-S., T.-Y. Chan, Y.-L. Wu, y T.-W. Lin. 2006. Applying the Adaptive Learning Material Producing Strategy to Group Learning. En *Technologies for E-Learning and Digital Entertainment*, ed. Zhigeng Pan, Ruth Aylett, Holger Diener, Xiaogang Jin, Stefan Göbel, y Li Li, 3942:39-49. Lecture Notes in Computer Science. Berlin/Heidelberg: Springer-Verlag. <http://www.springerlink.com/index/10.1007/11736639>.
- Kavcic, A. 2004. Fuzzy User Modeling for Adaptation in Educational Hypermedia. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)* 34, n°. 4: 439-449. doi:10.1109/TSMCC.2004.833294.
- Kobayashi, K., T. Hiroyasu, y M. Miki. 2007. Mechanism of Multi-Objective Genetic Algorithm for Maintaining the Solution Diversity Using Neural Network. En *Evolutionary Multi-Criterion Optimization*, ed. Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, y Tadahiko Murata, 4403:216-226. Berlin, Heidelberg: Springer Berlin Heidelberg. <http://www.springerlink.com/index/10.1007/978-3-540-70928-2>.
- Koedinger, K.R., J.R. Anderson, W.H. Hadley, y M.A. Mark. 1997. Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education* 8, n°. 1: 30-43.
- Koutsojannis, C., G. Beligiannis, I. Hatzilygeroudis, C. Papavlasopoulos, y J. Prentzas.

2007. Using a hybrid AI approach for exercise difficulty level adaptation. *International Journal of Continuing Engineering Education and Life-Long Learning* 17, n.º. 4: 256-272. doi:10.1504/IJCEELL.2007.015042.
- Kuh, G.D. 2001. Assessing what really matters to student learning: Inside the National Survey of Student Engagement. *Change* 33, n.º. 3: 10-17.
- Kunichika, H., M. Urushima, T. Hirashima, y A. Takeuchi. 2002. A computational method of complexity of questions on contents of English sentences and its evaluation. En *International Conference on Computers in Education, 2002. Proceedings.*, 97-101. Auckland, New Zealand. doi:10.1109/CIE.2002.1185873. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1185873>.
- Lanzilotti, R., y T. Roselli. 2007. An Experimental Evaluation of Logiocando, an Intelligent Tutoring Hypermedia System. *International Journal of Artificial Intelligence in Education* 17, n.º. 1: 41-56.
- Lavrač, N., P. Flach, y B. Zupan. 1999. Rule Evaluation Measures: A Unifying View. En *Inductive Logic Programming LNAI*, ed. S. Dzeroski y P. Flach, 1634:174-185. Berlin Heidelberg: Springer.
- Lavrač, N., B. Kavšek, P. Flach, y L. Todorovski. 2004. Subgroup Discovery with CN2-SD. *Journal of Machine Learning Research* 5: 153-188.
- Lee, M. 2001. Profiling students' adaptation styles in Web-based learning. *Computers & Education* 36, n.º. 2: 121-132. doi:10.1016/S0360-1315(00)00046-4.
- Leung, K.S., y W. Lam. 1998. Fuzzy Concepts in Expert Systems. *Computer* 21, n.º. 9: 43-56.
- Levy, Y. 2007. Comparing dropouts and persistence in e-learning courses. *Computers & Education* 48, n.º. 2: 185-204. doi:10.1016/j.compedu.2004.12.004.
- Lilley, M., T. Barker, y C. Britton. 2004. The development and evaluation of a software prototype for computer-adaptive testing. *Computers & Education* 43, n.º. 1: 109-123. doi:10.1016/j.compedu.2003.12.008.
- van der Linde, G. 2005. The Perception of Business Students At PUCMM Of The Use Of Collaborative Learning Using The BSCW As A Tool. En *2005 6th International Conference on Information Technology Based Higher Education and Training*, F2D-10-F2D-15. Santo Domingo, dominican Republic. doi:10.1109/ITHET.2005.1560280. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1560280>.
- van der Linden, Wim J. 2007. A Hierarchical Framework for Modeling Speed and Accuracy on Test Items. *Psychometrika* 72, n.º. 3 (8): 287-308. doi:10.1007/s11336-006-1478-z.
- Liu, H., H. Shi, Y. Shang, y S.-S. Chen. 2004. Student modeling with timed assessment information. En *Proceedings of the 2nd International Conference on Information Technology: Research and Education (ITRE 2004)*, 121-125.
- Lo, J.-J., H.-M. Wang, y S.-W. Yeh. 2004. Effects of confidence scores and remedial instruction on prepositions learning in adaptive hypermedia. *Computers & Education* 42, n.º. 1: 45-63. doi:10.1016/S0360-1315(03)00064-2.
- López-Pujalte, Cristina, Vicente P. Guerrero-Bote, y Félix de Moya-Anegón. 2003. Order-based fitness functions for genetic algorithms applied to relevance feedback. *Journal of the American Society for Information Science and Technology* 54, n.º. 2 (1): 152-160. doi:10.1002/asi.10179.

- Lowen, R. 1995. Ordering Fuzzy Real Quantities. En *Fuzzy Logic and Intelligent Systems*, ed. H. Li y M.M. Gupta, 69-83. Massachusetts: Kluwer Academic Publishers.
- Lozano, M, F Herrera, y J Cano. 2008. Replacement strategies to preserve useful diversity in steady-state genetic algorithms. *Information Sciences* 178, nº. 23 (12): 4421-4433. doi:10.1016/j.ins.2008.07.031.
- Mahfoud, S.W. 2000. Niching Methods. En *Evolutionary computation 2, Advanced Algorithms and Operators*, ed. Thomas Bäck, David B. Fogel, y Zbigniew Michalewicz, 87-92. Bristol and Philadelphia: Institute of Physics Publishing.
- Mangina, E, y J Kilbride. 2008. Utilizing vector space models for user modeling within e-learning environments☆. *Computers & Education* 51, nº. 2: 493-505. doi:10.1016/j.compedu.2007.06.008.
- Martinez-Mones, A., E. Gomez-Sanchez, Y.A. Dimitriadis, I.M. Jorriñ-Abellan, B. Rubia-Avi, y G. Vega-Gorgojo. 2005. Multiple Case Studies to Enhance Project-Based Learning in a Computer Architecture Course. *IEEE Transactions on Education* 48, nº. 3: 482-489. doi:10.1109/TE.2005.849754.
- MathWorks. 2000. Fuzzy Logic Toolbox. User's Guide.
- Mayo, M., y A. Mitrovic. 2001. Optimising ITS Behaviour with Bayesian Networks and Decision Theory. *International Journal of Artificial Intelligence in Education* 12: 124-153.
- Mehlenbacher, B., C.R. Miller, D. Covington, y J.S. Larsen. 2000. Active and interactive learning online: a comparison of Web-based and conventional writing classes. *IEEE Transactions on Professional Communication* 43, nº. 2: 166-184. doi:10.1109/47.843644.
- Mengshoel, Ole J., y David E. Goldberg. 2008. The Crowding Approach to Niching in Genetic Algorithms. *Evolutionary Computation* 16, nº. 3: 315-354.
- Mickle, M.H., L. Shuman, y M. Spring. 2004. Active learning courses on the cutting edge of technology. En *Proceedings of the 34th Annual Conference Frontiers in Education (FIE'04)*, 1:T2F/19 – T2F/23. Piscataway, NJ: IEEE.
- Michalewicz, Zbigniew. 1996. *Genetic algorithms + data structures = evolution programs*. 3º ed. Berlin New York: Springer-Verlag.
- Minaei-Bidgoli, B., D.A. Kashy, G. Kortemeyer, y W.F. Punch. 2003. Predicting student performance: an application of data mining methods with an educational web-based system. En *33rd Annual Frontiers in Education, 2003. FIE 2003.*, T2A\_13-T2A\_18. Westminster, Colorado, USA. doi:10.1109/FIE.2003.1263284. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1263284>.
- Mitchell, M. 2001. *An introduction to genetic algorithms*. 7º ed. Cambridge Mass. [u.a.].
- Mitchell, T. 1997. *Machine learning*. International ed. New York: McGraw-Hill.
- Mitra, S., R.E. Lopez-Herrejon, D. Zimmaro, M. Johnson, y M. Schulman. 2005. An Assessment of the Effectiveness of Interactive Technology in an Introductory Programming Course for Non-Majors. En *Proceedings Frontiers in Education 35th Annual Conference*, S3C-17-S3C-22. Indianapolis, IN, USA. doi:10.1109/FIE.2005.1612265.

- <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1612265>.
- Mödritscher, F. 2007. Implementation and Evaluation of Pedagogical Strategies in Adaptive E-Learning Environments. Graz, Austria: Graz University of Technology.
- Moreno, L., C. Gonzalez, I. Castilla, E.J. Gonzalez, y J. Sigut. 2007. Use of Constructivism and Collaborative Teaching in an ILP Processors Course. *IEEE Transactions on Education* 50, n° 2: 101-111. doi:10.1109/TE.2006.886461.
- Morozov, M., A. Tanakov, A. Gerasimov, D. Bystrov, y E. Cvirco. 2004. Virtual chemistry laboratory for school education. En *IEEE International Conference on Advanced Learning Technologies, 2004. Proceedings.*, 605-608. Joensuu, Finland. doi:10.1109/ICALT.2004.1357486. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1357486>.
- Mosqueira-Rey, E., y V. Moret-Bonillo. 2000. Validation of intelligent systems: a critical study and a tool. *Expert Systems with Applications* 18, n° 1: 1-16.
- Mosqueira-Rey, E., V. Moret-Bonillo, y Á. Fernández-Leal. 2008. An expert system to achieve fuzzy interpretations of validation data. *Expert Systems with Applications* 35, n° 4: 2089-2106.
- Muñiz, J. 1990. *Teoría de Respuesta a los Ítems, Un nuevo enfoque en la evolución psicológica y educativa*. Madrid: Ediciones Pirámide.
- Muñoz, F., y A. Ortigosa. 2006. Using Adaptive Hypermedia to Support Diversity in Secondary Schools. En *Proceedings of the 6th International Conference on Advanced Learning Technologies*, 1055-1059.
- Negnevitsky, Michael. 2005. *Artificial intelligence : a guide to intelligent systems*. 2° ed. Harlow England New York: Addison-Wesley.
- Nirmalakhandan, N. 2007. Computerized adaptive tutorials to improve and assess problem-solving skills. *Computers & Education* 49, n° 4: 1321-1329. doi:10.1016/j.compedu.2006.02.007.
- Noguez, J., y L.E. Sucar. 2005. A Probabilistic Relational Student Model for Virtual Laboratories. En *Sixth Mexican International Conference on Computer Science (ENC'05)*, 2-9. Puebla, Mexico. doi:10.1109/ENC.2005.7. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1592194>.
- Nouh, Y., P. Karthikeyani, y R. Nadarajan. 2006. Intelligent Tutoring System-Bayesian Student Model. En *Proceedings of the 1st International Conference on Digital Information Management*, 257-262.
- Oakley, B., R.M. Felder, R. Brent, y I. Elhajj. 2004. Turning Student Groups into Effective Teams. *Journal of Student Centered Learning* 2, n° 1: 9-34.
- Oakley, B.A., D.M. Hanna, Z. Kuzmyn, y R.M. Felder. 2007. Best Practices Involving Teamwork in the Classroom: Results From a Survey of 6435 Engineering Student Respondents. *IEEE Transactions on Education* 50, n° 3: 266-272. doi:10.1109/TE.2007.901982.
- Oh, K.-W., y A. Kandel. 1991. Coimplication and its application to fuzzy expert systems. *Information Sciences* 56, n° 1: 59-73. doi:10.1016/0020-0255(91)90024-O.
- O'Keefe, R.M., O. Balci, y E.P. Smith. 1986. *Validation of Expert System Performance*. Technical Report TR-86-37, Computer Science. Virginia Polytechnic Institute and State University. <http://eprints.cs.vt.edu/archive/00000043/>.

- Olea, J., y V. Ponsoda. 2003. *Tests Adaptativos Informatizados*. Madrid: UNED Ediciones.
- Own, Zangyuan. 2006. The Application of an Adaptive Web-Based Learning Environment on Oxidation–Reduction Reactions. *International Journal of Science and Mathematics Education* 4, n°. 1: 73-96. doi:10.1007/s10763-004-8129-6.
- Pahl, Claus. 2004. Data mining technology for the evaluation of learning content interaction. *International Journal on E-learning (IJEL)*.
- Petr, D.W. 2005. Experience with a Multiple-Choice Audience Response System in an Engineering Classroom. En *Proceedings Frontiers in Education 35th Annual Conference*, S3G-1-S3G-6. Indianapolis, IN, USA. doi:10.1109/FIE.2005.1612282. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1612282>.
- Philpot, T.A., R.H. Hall, N. Hubing, y R.E. Flori. 2005. Using games to teach statics calculation procedures: Application and assessment. *Computer Applications in Engineering Education* 13, n°. 3: 222-232.
- Prince, M. 2004. Does Active Learning Work? A Review of the Research. *Journal of Engineering Education* 93, n°. 3: 223-231.
- Prinsen, F., M. Volman, J. Terwel, y P. Vandeneeden. 2009. Effects on participation of an experimental CSCL-programme to support elaboration: Do all students benefit? *Computers & Education* 52, n°. 1: 113-125. doi:10.1016/j.compedu.2008.07.001.
- Pró, L., M.E. Ruiz, y C. Yañez. 2005. Diseño e implementación de un modelo de aprendizaje mediante algoritmos genéticos. *Revista de Investigación de Sistemas e Informática (RISI)* 2, n°. 3: 21-29.
- Qian Hao, y Tao Lu. 2009. Context modeling and reasoning based on certainty factor. En *2009 Asia-Pacific Conference on Computational Intelligence and Industrial Applications (PACIIA)*, 38-41. Wuhan, China. doi:10.1109/PACIIA.2009.5406578. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5406578>.
- Reeve, J., y E.L. Deci. 1996. Elements of the Competitive Situation that Affect Intrinsic Motivation. *Personality and Social Psychology Bulletin* 22, n°. 1: 24-33.
- Reguera, P., J.J. Fuertes, M. Domínguez, y B.A. León de la Barra. 2005. Active learning in control education. En *Proceedings of the 44th IEEE Conference on Decision and Control and the European Control Conference 2005 (CDC-ECC '05)*, 2721- 2726. IEEE.
- Regueras, L.M., E. Verdú, M.F. Muñoz, M.A. Perez, J.P. de Castro, y M.J. Verdú. 2009. Effects of Competitive E-Learning Tools on Higher Education Students: A Case Study. *IEEE Transactions on Education* 52, n°. 2: 279-285. doi:10.1109/TE.2008.928198.
- Regueras, L.M., E. Verdú, M.A. Pérez, J.P. de Castro, y M.J. Verdú. 2008. E-learning 2.0: The Learning Community. En *Advances in E-Learning: Experiences and Methodologies*, 213-231. Idea Group Inc.
- Regueras, L.M., E. Verdú, M.J. Verdú, J.P. de Castro, y M.A. Pérez. 2007. The Impact of Competitive E-learning Tools on Motivation and Outcomes of Higher Education Students. En *Proceedings of the Computer Assisted Learning*

- (CAL'07) – 'Development, Disruption and Debate'. Dublín, Irlanda.
- Regueras, L.M., E. Verdú, M.J. Verdú, M.A. Pérez, y J.P. de Castro. 2007. E-learning Strategies to Support Databases Courses: a Case Study. En *Post-proceedings of the International Conference on Technology, Training and Communication - Extended Papers*. <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-361/paper19.pdf>.
- Regueras, L.M., E. Verdú, M.J. Verdú, M.A. Pérez, J.P. de Castro, y M.F. Muñoz. 2008. Motivating Students through On-Line Competition: An Analysis of Satisfaction and Learning Styles. En *Advances in Web Based Learning - ICWL 2008*, ed. Frederick Li, Jianmin Zhao, Timothy K. Shih, Rynson Lau, Qing Li, y Dennis McLeod, 5145:167-177. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg. <http://www.springerlink.com/index/10.1007/978-3-540-85033-5>.
- Riekert, M., K. M. Malan, y A.P. Engelbrect. 2009. Adaptive Genetic Programming for dynamic classification problems. En *2009 IEEE Congress on Evolutionary Computation*, 674-681. Trondheim, Norway, 5. doi:10.1109/CEC.2009.4983010. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4983010>.
- Rodríguez, M.C., R. Varela, J. Palma Méndez, y R. Marín Morales. 2008. Técnicas basadas en búsquedas heurísticas. En *Inteligencia Artificial: Técnicas, Métodos y Aplicaciones*. Aravaca: McGraw-Hill Interamericana de España.
- Rodríguez, S., I. Cheng, y A. Basu. 2007. Multimedia games for learning and testing physics. En *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME 2007)*, 1838-1841.
- Romero, C, P Gonzalez, S Ventura, M Deljesus, y F Herrera. 2009. Evolutionary algorithms for subgroup discovery in e-learning: A practical application using Moodle data. *Expert Systems with Applications* 36, n°. 2 (3): 1632-1644. doi:10.1016/j.eswa.2007.11.026.
- Romero, C., S. Ventura, C. de Castro, W. Hall, y N.M. Hong. 2002. Using Genetic Algorithms for Data Mining in Webbased Educational Hypermedia Systems. En *Proceedings of AH2002 workshop Adaptive Systems for Web-based Education*. <http://www.lsi.us.es/redmidas/Capitulos/LMD23.pdf>.
- Romero, C., S. Ventura, y E. Garcia. 2008. Data mining in course management systems: Moodle case study and tutorial. *Computers & Education* 51, n°. 1: 368-384. doi:10.1016/j.compedu.2007.05.016.
- Ronald, Simon. 1995. Preventing Diversity Loss in a Routing Genetic Algorithm with Hash Tagging. *Complexity International* 2. [http://www.complexity.org.au/ci/vol02/sr\\_hash/](http://www.complexity.org.au/ci/vol02/sr_hash/).
- Roskam, E. E. 1997. Models for Speed and Time-Limit Tests. En *Handbook of Modern Item Response Theory*, ed. W. J. van der Linden y R. K. Hambleton, 188-207. New York: Springer-Verlag.
- Rudolph, G., y J. Ziegenhirt. 2000. Computation time of evolutionary operators. En *Evolutionary computation 2, Advanced Algorithms and Operators*, ed. Thomas Bäck, David B. Fogel, y Zbigniew Michalewicz, 247-252. Bristol and Philadelphia: Institute of Physics Publishing.
- Samejina, F. 1997. Graded Response Model. En *Handbook of Modern Item Response Theory*, ed. W. J. van der Linden y R. K. Hambleton, 85-100. New York:

- Springer-Verlag.
- Santos, O. C., J.G. Boticario, y C. Barrera. 2004. The Standards-based Architecture of the Adaptive Learning Environment aLFanet. *WSEAS Transactions on Computers*.
- Sareni, B., y L. Krahenbuhl. 1998. Fitness sharing and niching methods revisited. *IEEE Transactions on Evolutionary Computation* 2, n°. 3 (Septiembre): 97-106. doi:10.1109/4235.735432.
- Semerci, Ç. 2004. The influence of fuzzy logic theory on students' achievement. *The Turkish Online Journal of Educational Technology – TOJET* 3, n°. 2: 56-61.
- Shifrony, E., y D. Ginat. 1997. Simulation game for teaching communication protocols. En *Proceedings of the 28th ACM Computer Science Education Symposium, SIGCSE*, 184-188. ACM Press.
- Shimodaira, H. 2001. A diversity-control-oriented genetic algorithm (DCGA): performance in function optimization. En *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, 1:44-51. Seoul, South Korea. doi:10.1109/CEC.2001.934369. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=934369>.
- Siddiqui, A, M Khan, y S Akhtar. 2008. Supply chain simulator: A scenario-based educational tool to enhance student learning. *Computers & Education* 51, n°. 1: 252-261. doi:10.1016/j.compedu.2007.05.008.
- Smith, L., y S. Mann. 2002. Playing the game: A model for gameness in interactive game based learning. En *Proceedings of the 15th Annual NACCOQ*, 397-402. Hamilton New Zealand.
- Smoline, D. 2008. Some problems of computer-aided testing and “interview-like tests”. *Computers & Education* 51, n°. 2: 743-756. doi:10.1016/j.compedu.2007.07.008.
- Springer, L., M.E. Stanne, y S. Donovan. 1999. Effects of Small-Group Learning on Undergraduates in Science, Mathematics, Engineering and Technology: A Meta-Analysis. *Review of Educational Research* 69, n°. 1: 21-51.
- Striegel, A., y D.T. Rover. 2002. Problem-based learning in an introductory computer engineering course. En *32nd Annual Frontiers in Education*, F1G-7-F1G-12. Boston, MA, USA. doi:10.1109/FIE.2002.1158138. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1158138>.
- Sultan, A.B.M, R. Mahmood, M.N. Sulaiman, y M.R. Abu Bakar. 2006. Maintaining Diversity for Genetic Algorithm: A Case of Timetabling Problem. *Jurnal Teknologi* 44: 123-130.
- Suraweera, P., y A. Mitrovic. 2004. An Intelligent Tutoring System for Entity Relationship Modelling. *International Journal Artificial Intelligent in Education* 14, n°. 3: 475-417.
- Tai, D.W.-S., T.-A. Tsai, y F.M.-C. Chen. 2001. Performance Study on Learning Chinese Keyboarding Skills Using the Adaptive Learning System. *Global Journal of Engineering Education* 5, n°. 2: 153-161.
- Taplin, John H.E., y Min Qiu. 2001. Using GA to Optimise the Selection and Scheduling of Road Projects. En *The Practical Handbook of Genetic Algorithms Applications*, ed. Lance Chambers. 2° ed. Chapman & Hall/CRC.
- Terenzini, P.T., A.F. Cabrera, C.L. Colbeck, J.M. Parente, y S.A. Bjorklund. 2001.

- Collaborative Learning vs. Lecture/Discussion: Students' Reported Learning Gains. *Journal of Engineering Education* 90, n°. 1: 123-130.
- Thalheimer, W., y S. Cook. 2002. How to calculate effect sizes from published research: A simplified methodology. *Work-Learning Research*. [http://work-learning.com/effect\\_sizes.htm](http://work-learning.com/effect_sizes.htm).
- Timmerman, B., y R. Lingard. 2003. Assessment of active learning with upper division computer science students. En *33rd Annual Frontiers in Education, 2003. FIE 2003.*, S1D\_7-S1D\_12. Westminster, Colorado, USA. doi:10.1109/FIE.2003.1265914. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1265914>.
- Titcomb, S.L., R.M. Foote, y H.J. Carpenter. 1994. A model for a successful high school engineering design competition. En *Proceedings of 1994 IEEE Frontiers in Education Conference - FIE '94*, 138-141. San Jose, CA, USA. doi:10.1109/FIE.1994.580489. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=580489>.
- Tomescu, B. 2001. On the use of fuzzy logic to control paralleled DC-DC converters. Dissertation PhD, Blacksburg, Virginia, U.S.A.: Department of Electrical and Computer Engineering, Faculty of the Virginia Polytechnic Institute and State University. <http://scholar.lib.vt.edu/theses/available/etd-10252001-143729>.
- Tseng, J, H Chu, G Hwang, y C Tsai. 2008. Development of an adaptive learning system with two sources of personalization information☆. *Computers & Education* 51, n°. 2: 776-786. doi:10.1016/j.compedu.2007.08.002.
- Tseng, S., P. Sue, J. Su, J. Weng, y W. Tsai. 2007. A new approach for constructing the concept map. *Computers & Education* 49, n°. 3: 691-707. doi:10.1016/j.compedu.2005.11.020.
- Van Rosmalen, P., H. Vogten, R. Van Es, H. Passier, P. Poelmans, y R. Koper. 2006. Authoring a full life cycle model in standards-based, adaptive e-learning. *Educational Technology & Society* 9, n°. 1: 72-83.
- VanLehn, K., C. Lynch, K. Schulze, J.A. Shapiro, R. Shelby, L. Taylor, D. Treacy, A. Weinstein, y M. Wintersgill. 2005. The Andes Physics Tutoring System: Lessons Learned. *International Journal of Artificial Intelligence in Education* 15, n°. 3: 678-685.
- Vansteenkiste, M., y E.L. Deci. 2003. Competitively Contingent Rewards and Intrinsic Motivation: Can Losers Remain Motivated? *Motivation and Emotion* 27, n°. 4: 273-299.
- Verdu, E., J. P. de Castro, M. J. Verdú, M. A. Pérez, y L. M. Regueras. 2010. The Next Generation of Competitive E-learning: The Adaptive Competitive Systems. En *A New Learning Paradigm: Competitions supported by Technology*, ed. E. Verdú, R. Lorenzo, M. A. Revilla, y L. M. Regueras, 41-62. Madrid: Sello Editorial.
- Verdu, E., y R. Lorenzo. 2010. Effective Strategies for Learning: Collaboration and Competition. En *A new learning paradigm : competition supported by technology*, ed. E. Verdú, R. Lorenzo, M. A. Revilla, y L. M. Regueras, 11-40. 1º ed. Madrid: Sello Editorial.
- Verdú, E., L. M Regueras, M.J. Verdú, M. A. Pérez, y J. P de Castro. 2006. Improving the Higher Education through Technology-based Active Methodologies: A Case

- Study. *WSEAS Transactions on Advances in Engineering Education* 3, nº. 7: 649-656.
- Verdú, E., L. M. Regueras, M. J. Verdú, J. P. de Castro, y M.A. Pérez. 2008a. Telematic Environments and Competition Based Methodologies: an Approach to Active Learning. En *Advances in E-Learning: Experiences and Methodologies*, ed. F.J. García-Peñalvo, 232-249. Idea Group Inc.
- Verdú, E., L. M. Regueras, M. J. Verdú, J. P. de Castro, y M.A. Pérez. 2008b. Application of Intelligent Adaptive Systems in a Competitive Learning Environment. En *Proceedings of the IADIS International Conference E-learning 2008*, 2:273-275.
- Verdú, E., L. M. Regueras, M.J. Verdú, J. P. de Castro, y M. A. Pérez. 2008. An analysis of the Research on Adaptive Learning: The Next Generation of e-Learning. *WSEAS Transactions on Information Science & Applications* 5, nº. 6: 859-868.
- Verdú, E., L. M. Regueras, M.J. Verdú, M. A. Pérez, y J. P. de Castro. 2007. Aplicación de un Sistema Telemático de Aprendizaje Activo y Competitivo en el Área de Ingeniería Telemática. En *VI Jornadas de Ingeniería Telemática, JITEL 2007*, ed. L. Fuentes, J. López, y P. Merino, 245-252. Universidad de Málaga.
- Verdú, E., L.M. Regueras, M.J. Verdú, y J.P. de Castro. 2010. Estimating the Difficulty Level of the Challenges Proposed in a Competitive e-Learning Environment. En *The Twenty Third International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems. Part I.*, ed. N. García-Pedrajas, 6096:225-234. Lecture Notes in Artificial Intelligence. Springer.
- Verdú, E., M. J. Verdú, L. M. Regueras, y J. P. de Castro. 2010. A Diversity-Enhanced Genetic Algorithm to Characterize the Questions of a Competitive e-Learning System. En *Proc. of the Tenth IEEE International Conference on Advanced Learning Technologies (ICALT 2010)*. IEEE Computer Society Conference Publishing Services.
- Viera, A.J., y J.M. Garrett. 2005. Understanding interobserver agreement: the kappa statistic. *Family Medicine* 37, nº. 5: 360-363.
- Vogel, J.J., A. Greenwood-Ericksen, J. Cannon-Bowers, y C.A. Bowers. 2006. Using Virtual Reality with and without Gaming Attributes for Academic Achievement. *Journal of Research on Technology in Education* 39, nº. 1: 105-118.
- Vozár, Oto, y Mária Bieliková. 2008. Adaptive Test Question Selection for Web-Based Educational System. En *2008 Third International Workshop on Semantic Media Adaptation and Personalization*, 164-169. Prague, Czech Republic. doi:10.1109/SMAP.2008.15.  
<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4724867>.
- Wang, H, A Parrish, R Smith, y S Vrbsky. 2005. Improved variable and value ranking techniques for mining categorical traffic accident data. *Expert Systems with Applications* 29, nº. 4: 795-806. doi:10.1016/j.eswa.2005.06.007.
- Wang, T.-H. 2008. Web-based quiz-game-like formative assessment: Development and evaluation. *Computers & Education* 51, nº. 3: 1247-1263. doi:10.1016/j.compedu.2007.11.011.
- Wang, W., J.-F. Weng, J.-M. Su, y S.-S. Tseng. 2004. Learning Portfolio Analysis and Mining in SCORM Compliant Environment. En *34th ASEE/IEEE Frontiers in Education*

- Conference*, T2C-17 - T2C-24. Savannah, GA: IEEE.
- Watson, K. 1995. Utilization of active and cooperative learning in EE courses: three classes and the results. En *Proceedings Frontiers in Education 1995 25th Annual Conference. Engineering Education for the 21st Century*, 3c2.1-3c2.6. Atlanta, GA, USA. doi:10.1109/FIE.1995.483137. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=483137>.
- Webb, Geoffrey I., y Songmao Zhang. 2005. K-Optimal Rule Discovery. *Data Mining and Knowledge Discovery* 10, n°. 1 (1): 39-79. doi:10.1007/s10618-005-0255-4.
- Wei, F.-H., L.-Y. Lee, y G.-D. Chen. 2004. Supporting adaptive mentor by student preference within context of problem-solving learning. En *IEEE International Conference on Advanced Learning Technologies, 2004. Proceedings.*, 789-791. Joensuu, Finland. doi:10.1109/ICALT.2004.1357658. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1357658>.
- Wen, J., S. Wang, S. Cheng, Q.H. Wu, y D.W. Shimmin. 1998. Measurement based power system load modeling using a population diversity genetic algorithm. En *POWERCON '98. 1998 International Conference on Power System Technology. Proceedings (Cat. No.98EX151)*, 771-775. Beijing, China. doi:10.1109/ICPST.1998.729070. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=729070>.
- White, S., J. Carter, S. Jamieson, N. Efford, y T. Jenkins. 2007. TOPS - Collaboration and competition to stretch our most able programming novices. En *2007 37th annual frontiers in education conference - global engineering: knowledge without borders, opportunities without passports*, S2J-20-S2J-24. Milwaukee, WI, USA. doi:10.1109/FIE.2007.4417998. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4417998>.
- Wright, A.H. 1991. Genetic Algorithms for Real Parameter Optimization. En *Foundations of Genetic Algorithms*, ed. G.J.E. Rawlins, 205-218. San Mateo, CA: Morgan Kaufmann Publishers.
- Wu, W.M.C., H.N.H. Cheng, M.-C. Chiang, Y.-C. Deng, C.-Y. Chou, C.-C. Tsai, y T.-W. Chan. 2007. AnswerMatching: A Competitive Learning Game with Uneven Chance Tactic. En *2007 First IEEE International Workshop on Digital Game and Intelligent Toy Enhanced Learning (DIGITEL'07)*, 89-98. Jhongli City, Taiwan. doi:10.1109/DIGITEL.2007.11. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4148836>.
- Yacef, K. 2005. The Logic-ITA in the classroom: a medium scale experiment. *International Journal on Artificial Intelligence in Education*, n°. 15: 41-60.
- Yu, C. -C., y Y. -L. Chen. 2005. Mining Sequential Patterns from Multidimensional Sequence Data. *IEEE Transactions on Knowledge and Data Engineering* 17, n°. 1: 136-140.
- Zadeh, L.A. 1965. Fuzzy Sets. *Information and Control* 8, n°. 3: 338-353.
- Zadeh, L.A. 1983. The role of fuzzy logic in the management of uncertainty in experts systems. *Fuzzy Sets and Systems* 11: 199-227.
- Zakrzewska, D. 2008. Cluster Analysis for Users' Modeling in Intelligent E-Learning Systems. En *New Frontiers in Applied Artificial Intelligence*, 5027:209-214. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer.
- Zhou, F., J. Jiao, D. Schaefer, y S. Chen. 2009. Rough Set Based Rule Mining for

Affective Design. En . Stanford, California.

Zhu, K.Q. 2003. A diversity-controlling adaptive genetic algorithm for the vehicle routing problem with time Windows. En *15th IEEE International Conference on Tools with Artificial Intelligence*, 176-183. Los Alamitos, CA: IEEE Computer Society.

Zimmerman, H.-J. 1991. *Fuzzy Set theory and its Applications*. 2º ed. New York: Kluwer Academic Publishers.