Universidad de Granada

INTELIGENCIA COMPUTACIONAL Y JUEGOS APLICADOS A LA ENSEÑANZA

Presentado por

JOSÉ CARPIO CAÑADA

Directores

JUAN JULIÁN MERELO GUERVÓS

VÍCTOR MANUEL RIVAS SANTOS

Firmado: José Carpio Cañada

Noviembre 2015

VISTO BUENO

El **Prof. Dr. D. Juan Julián Merelo Guervós**, Catedrático de Universidad del Departamento de Arquitectura y Tecnología de Computadores de la Universidad de Granada y el profesor **Prof. Dr. D. Víctor Manuel Rivas Santos**, Titular de Universidad del Departamento de Informática de la Universidad de Jaén,

CERTIFICAN:

Que la memoria titulada:

"INTELIGENCIA COMPUTACIONAL Y JUEGOS APLICADOS A LA ENSEÑANZA"

ha sido realizada por **D. José Carpio Cañada** bajo nuestra dirección en el Departamento de Arquitectura y Tecnología de Computadores de la Universidad de Granada para optar al grado de **Doctor en Informática**.

En Granada, a 11 de Noviembre de 2015.

Los Directores de la tesis doctoral:

Fdo. Juan Julián Merelo Guervós
y Víctor Manuel Rivas Santos

DECLARACIÓN

El doctorando José Carpio Cañada y los directores de la tesis Juan Julián Merelo Guervósy Víctor Manuel Rivas Santos garantizamos, al firmar esta tesis doctoral, que el trabajo ha sido realizado por el doctorando bajo la dirección de los directores de la tesis y hasta donde nuestro conocimiento alcanza, en la realización del trabajo, se han respetado los derechos de otros autores a ser citados, cuando se han utilizado sus resultados o publicaciones.

Granada, Noviembre de 2015

José Carpio Cañada	Juan Julián Merelo Guervós y Víctor Manuel Rivas Santos

A mi familia:
Manuel Angel, Matilde, Manuel José y Fernando.

RESUMEN

Las limitaciones de los métodos tradicionales de enseñanza (como por ejemplo, la comunicación unidireccional, las metodologías rígidas, los enfoques orientados a resultados) pueden influenciar de forma negativa en la motivación y en las espectativas de los estudiantes provocando una reducción de los resultados académicos. Con el objetivo de hacer que el proceso de aprendizaje sea motivante esta tesis presenta una metodología que permite mejorar la experiencia de aprendizaje de los alumnos. Como aplicación práctica de la metodología propuesta, se han llevado a cabo varias experiencias reales en asignaturas clasicas de Ingeligencia Artificial en las que algunas sesiones clásicas han sido sustituidas por la participación en competiciones nacionales e internacionales de Ingeligencia Artificial que tenían como objetivo la realización de un agente capaz de competir contra otros adversarios en algún tipo de juego. Se han analizado entre otros elementos el ranking en la competición, la opinión de los estudiantes o el progreso académico con el fin de evaluar la metodología empleada. Hemos comprobado como la experiencia educacional mojora la percepción global de los estudiantes, mejorando incluso sus resultados académicos y sus habilidades personales gracias al aprendizaje a través de la participación en un juego. Por otro lado, aprovechando la motivación extra de la competición se han alcanzado objetivos adicionales (como por ejemplo, el aprendizaje de nuevos lenguajes de programación o nuevas tecnologías). Como conclusión, este paradigma de experiencia real nos ha permitido comprobar que el proceso es más importante que el resultado y que es posible adaptar esta metodología a diferentes escenarios de aprendizaje dentro de una institución.

AGRADECIMIENTOS

TABLA DE CONTENIDOS

LISTA DE FIGURAS

LISTA DE TABLAS

ACRÓNIMOS

CI	Computational Intelligence
API	Application Programming Interface
CPU	Central Processing Unit
EA	Evolutionary Algorithm
EC	Evolutionary Computation
EP	Evolutionary Programming
GA	Genetic Algorithm
GP	Genetic Programming

Parte I

CAPÍTULOS

1

INTRODUCCIÓN

Índice

1.1 INTRODUCCIÓN

La presente tesis doctoral introduce una metodología para mejorar la experiencia tanto del docente como del estudiante en el aprendizaje de la Inteligencia Artificial (IA) a través de la participación en competiciones de IA.

Los métodos didácticos tradicionales a menudo conllevan varios inconvenientes debido a las limitaciones de la enseñanza formal [**?**]. Entre otros, la comunicación unidireccional deja de fomentar la participación activa de los estudiantes, de esta forma los profesores deben realizar un esfuerzo adicional en el camino de alcanzar los objetivos didácticos propuestos. Los errores cometidos por el alumno en el proceso de aprendizaje suelen castigarse desde un enfoque punitivo del método de enseñanza. Además de esto, los calendarios suelen ser rígidos y no siempre se adaptan a las necesidades de los estudiantes con diferentes niveles de conocimientos y habilidades individuales [**?**]. Todo esto deriva en una disminución en la motivación y el interés de los estudiantes, que se agrava en la educación en ingeniería [**?**]. Una metodología más flexible es especialmente adecuada en el caso de materias que incluyan créditos prácticos. En este sentido, los métodos interactivos (por ejemplo: las sesiones de resolución de problemas, las prácticas con ordenador y los juegos) permiten a los profesores conseguir una mayor implicación de los estudiantes en las actividades propuestas por Adams y otros [**?**]. Teniendo todo esto en cuenta, el aprendizaje mediante el juego llega a la escena de la enseñanza como una de las experiencias de aprendizaje más exitosas [**?**].

La educación formal está caracterizada por un modelo sistemático, estructurado y guiado por una serie de directivas curriculares, con frecuencia presentando objetivos, contenidos y metodologías rígidas tanto para los profesores como para los estudiantes [**?**]. Además, el aprendizaje formal no se ajusta a nuestra manera natural de aprender, solo se muestra adecuado para un 18% de los niños de primaria y secundaria (K-12[1]) y un 5.1% de los estudiantes universitarios [**?**]. Con esta configuración, la educación formal no siempre es un buen estímulo a medida que el estudiante demanda mayor naturalidad, flexibilidad e interacción para apoyar su aprendizaje. Además los estudiantes entran en la escena del aprendizaje con diferentes grados de compromiso, habilidades y estilos de didácticos, hecho que afecta a su grado de motivación [**?**]. Como indican las últimas teorías, la

1 K-12 es el término en Estados Unidos y Canadá para los estudiantes de primaria y secundaria, desde 4-6 a 17-19 años

motivación representa un factor clave para aprender y obtener un resultado académico exitoso [**? ? ?**].

En contraposición a lo anterior, la educación informal da a los estudiantes la oportunidad de participar en su aprendizaje de forma proactiva a través metodologías flexibles y con diferentes estilos de aprendizaje [**?**]. Esto amplía las competencias personales más que las desarrolladas en el aprendizaje formal (por ejemplo, el liderazgo, la disciplina, la responsabilidad, el trabajo en equipo, la gestión de conflictos, la planificación, la organización o las relaciones interpersonales). Como consecuencia, es considerada por los estudiantes una metodología más favorable, eficaz y estimulante comparada con una educación formal menos atractiva y eficiente [**?**]. La educación informal y el juego están cambiando el modo en el que pensamos sobre el conocimiento y el aprendizaje, además de la forma en la que estructuramos el trabajo y las ideas. El aprendizaje a través del juego permite al alumnado construir su propio conocimiento, basado en la comprensión de sus propias experiencias, tal y como indican las recientes teorías constructivistas [**?**]. El aprendizaje activo es eficaz para motivar y mejorar el rendimiento de los estudiantes, promoviendo el pensamiento creativo y con diferentes estilos de aprendizaje. El estilo “quinestético" (término que hace referencia al aprendizaje a través de actividades físicas) es el más adoptado en juegos, pero el estilo VARK (visual, auditivo, lectura/escritura y “quinestético") también puede ser utilizado [**?**]. El aprendizaje a través del juego está mejor documentado para niños de primaria y secundaria (K-12) que para universitarios [**?**]. Las ventajas del aprendizaje interactivo para adultos son claras y variadas, especialmente en ingeniería dónde el conocimiento práctico requiere de una interacción directa en los laboratorios además de las lecciones teóricas [**?**].

Desde el año 2010 se utiliza el concepto de gamificación [**?**] para referirnos al uso de juegos en ambientes o entornos no lúdicos [**?**]. También podemos encontrar el término ludificación en el mismo sentido. La gamificación o ludificación en la enseñanza, fomenta de forma activa la creatividad, el desarrollo de estrategias para la resolución de problemas y la autoconfianza para abordar nuevos desafíos [**?**]. Sin embargo, la experiencia no es siempre suficiente para aprender y es necesario incorporar otros aspectos en el proceso [**?**], como pueden ser la observación, el análisis, el pensamiento crítico, la abstracción y las ensayos del conocimiento adquirido en nuevas situaciones. En este contexto, el aprendizaje basado en la competición proporciona un esce-

nario adecuado para proporcionar todos los elementos necesarios para alcanzar un aprendizaje constructivo. Sin embargo, un porcentaje muy bajo de profesores y estudiantes se aprovechan de la gran popularidad de los juegos con fines educativos.

El aprendizaje activo a través de las competiciones se ha probado como un factor motivador permitiendo a los estudiantes adquirir conocimiento por ellos mismos a través de la actividad y el razonamiento [**?**]. Este modo de aprendizaje se caracteriza por una perspectiva centrada en el estudiante dónde el proceso es más importante que el resultado. Por tanto, los profesores se convierten en el medio para guiar a los estudiantes en el proceso de aprendizaje, dónde los alumnos motivados aprenden las materias de la asignatura a través de la resolución de los desafíos planteados [**?**]. Como principal ventaja, los estudiantes responden de manera natural a este tipo de aprendizaje, dónde lo juegos ofrecen un medio para formar y reformar ideas de una forma divertida e interactiva. Como resultado, cuanto más motivado e implicado está el alumno, mayor es el aprendizaje [**?**]. La presente tesis doctoral presenta una metodología para mejorar la experiencia tanto del docente como del estudiante en el aprendizaje de la Inteligencia Artificial (IA) a través de la participación en competiciones de IA.

Aprender jugando en IA

Desde que Alan Turing estableciese el primer juego que podía ser jugado de forma automática por máquinas utilizando algoritmos lógicos, estos han sido utilizados como una metodología de aprendizaje para enseñar diferentes conceptos de IA [**?**]. Esto transformó los juegos en una herramienta potencialmente exitosa utilizada para enseñar una gran variedad de métodos prácticos gracias a su habilidad para motivar a los estudiantes proporcionando espontaneidad, flexibilidad e interactividad para apoyar la experiencia de aprendizaje [**?**]. Los ejemplos más representativos en educación son los juegos de mesa clásicos como Backgammon, utilizado para enseñar métodos de exploración por la técnica de aprendizaje por refuerzo [**?**]; Checkers, utilizado para desarrollar técnicas de resolución de problemas basadas en búsquedas [**?**]; Tic-Tac-Toe, utilizado para Mini-Max y poda Alfa-Beta [**?**]; N-puzle, utilizado en búsqueda en espacio de estados [**?**]; o n-Reinas, utilizado para enseñar problemas de satisfacción de restricciones [**?**], además de otros.

Los profesores han detectado que la motivación de los estudiantes juega un papel clave en el aprendizaje y que es posible

alcanzar los objetivos académicos con éxito a través de los desafíos propuestos en las asignaturas. Por ejemplo, The Open Racing Car Simulator, un framework open source y multiplataforma altamente portable, ha sido utilizado como un juego de coches 3D para enseñar principios mecánicos en la Universidad Northern Illinois [**?**]. Además, diferentes ligas RoboCode se han organizado en la Universidad Nacional de Maynooth, con el objetivo de enseñar lenguajes de programación [**?**]. En estos casos, a los alumnos se les plantea el diseño de agentes inteligentes, llamados robots, para competir unos con otros intentando imitar el comportamiento humano [**?**]. En otros casos, la competición ayuda a descubrir estudiantes con talento y habilidades especiales en las escuelas de ingeniería. Como ejemplo, la competición internacional Facebook Hacker Cup comenzó en 2011 con este propósito, que consiste en resolver un número de problemas basados en algoritmos utilizando cualquier framework o lenguaje de programación [**?**]. Además, la universidad del estado de Wichita ha utilizado Lego Mindstorm para la First Lego League. Esta competición, que también ha sido probada como una útil metodología de enseñanza en estudiantes K-12, ha ayudado a adquirir aptitudes individuales, valores, habilidades y conocimientos que han sido adquiridos de forma natural gracias a la educación informal [**?**].

Con el objetivo de utilizar IA como plataforma de pruebas con el fin de motivar la formación y la investigación en este campo, han surgido diferentes competiciones tanto nacionales como internacionales. Por ejemplo, la Universidad de Stanford utilizó AAAI (Association for the Advancement of AI) General Game Playing como una excelente plataforma de desarrollo para estudiantes durante una competición celebrada en verano [**?**]. Además, la Universidad de Hartfold ha desarrollado y probado un conjunto de proyectos denominados MLExAI (Machine Learning Experience in AI) que pueden ser integrados en cursos introductorios para enseñar IA a través del aprendizaje automático [**?**]. La Universidad de Essex lanzó la liga MS Pac-Man contra Ghost con el fin de enfrentar a robots creados por diferentes competidores que habían sido previamente probados con éxito por profesores y estudiantes en cursos de IA [**?**]. Otra competición que se ha celebrado tradicionalmente en Universidades ha sido el Physical Travelling Salesman Problem, un juego con un único jugador dirigido a resolver problemas de optimización combinatoria con controladores de IA [**?**]; La competición de Carreras de Coches simuladas, un evento que consiste en tres competiciones dónde se aplican técnicas de IA para controlar coches en

un juego de carreras [**?**]; la competición de IA Mario, ha sido un referente utilizado en diferentes competiciones relacionadas con congresos internacionales en educación y/o investigación [**?**]; y la competición Start Craft AI, un juego avanzado de estrategia para los que los robots con IA tienen que abatir a jugadores humanos expertos en tiempo real [**?**], además de otros.

Estos paradigmas representan un escenario dónde la observación, la abstracción de conceptos, el pensamiento crítico, el análisis y el conocimiento adquirido concurren en un proceso educativo de éxito dentro del contexto de una competición. La competición AI Challenge organizada por la Universidad de Waterloo y patrocinada por Google [**?**]. Cada edición ha consistido en un reto diferente y los participantes debían enviar un robot para competir contra robots de otros participantes [**?**]. Los temas en esta serie de competiciones han sido Rock-Paper-Scissors (otoño 2009), Tron Light-Cycles (primavera 2010), Planet Wars (otoño 2010) y Ants (otoño 2011). Aunque la primera edición estuvo basada en un juego muy conocido, las siguientes competiciones se basaron en diseño de juegos totalmente originales. Esto proporcionó un factor de motivación para explorar nuevos enfoques, experimentar con ideas diferentes y finalmente encontrar soluciones a problemas por estudiantes de todo el mundo. Google AI Challenge se distingue por ser una competición internacional con partidas online multijugador tanto para universitarios como para profesionales. Ha sido utilizado para enseñar una variedad de algoritmos de IA (p. ej. algoritmos genéticos, redes neuronales y lógica borrosa), a la vez que se enseñan nuevos lenguajes de programación a través de la implementación de agentes inteligentes.

Competiciones de robótica aplicadas a la IA

Se trata de un claro ejemplo para permitir que los estudiantes ganen interés en la robótica y demostrar que las competiciones de robótica son un buen marco para desarrollar experiencias en el aula. Las competiciones también ofrecen a los estudiantes la oportunidad de encontrase con personas con más experiencia en un determinado campo. Además, mediante este tipo de experiencias de competición, se transmite a los estudiantes la idea de que, con frecuencia, la vida real es diferente a los problemas que se resuelven en la Universidad. Puede ser muy diferente diseñar y programar código para una simulación, que hacerlo para un robot real móvil. Por ejemplo, hay muchos factores adicionales que hay que tener en cuenta, como la carga de la batería o la iluminación del lugar de la competición. Muchas cuestiones sur-

gen durante la experiencia, tanto en relación con el diseño del hardware y como del software.

Casi todos los profesores en el ámbito de la Ingeniería tienen el objetivo de formar a los ingenieros del mañana y, en esa formación, es muy importante acercar a los estudiantes al mundo real. Esta idea se ha plasmado con varias experiencias relevantes que se pueden encontrar en la literatura. Por ejemplo, en Zhongli y otros [**?**], se presenta una plataforma basada en Internet para una competición de fútbol dedicada a robots educacionales. De Vault [**?**], describe una experiencia a través de un curso de robots móviles y la participación en una competición anual. En Grimes [**?**] y otros, se describen los resultados académicos y los conocimientos adquiridos por los estudiantes en una competición que se convierte en una excelente oportunidad de crecimiento educacional. Es este trabajo se describe cómo los estudiantes tienen toda la responsabilidad en la definición de las reglas de la competición, diseñando y construyendo la pista y llevando a cabo la competición. En Berlier [**?**] y otros, se presenta una metodología utilizada para reemplazar el proyecto final por otro dónde los estudiantes construyen un robot basado en micro controlador con el objetivo de participar en una competición. Murphy [**?**], describe una estrategia para integrar una competición de diseño de robots en asignaturas como una forma de mejorar la experiencia de aprendizaje mejorar el desarrollo intelectual. Finalmente en Almeida y otros [**?**] , se presentan las competiciones de robots móviles como un evento muy adecuado para la experimentación, la investigación y el desarrollo en muchas áreas de la educación secundaria y Universitaria.

Descripción de las competiciones

Durante el desarrollo de esta tesis se ha participado en 6 competiciones Cosmobot 2009, CIAR 2010, Ants 2011, First Lego League 2011/2012 y 2012/2013 y Hello World Open 2014. De las seis competiciones en las que se ha participado se han descrito en forma de artículos dos de ellas Cosmobot 2009 (Carpio y otros 2011 [**?**]) y Ants 2011 (Carpio y otros 2014 [**?**] y Carpio y otros 2013 [**?**]). A continuación describiremos brevemente cada una de ellas.

Cosmobot 2009

Cosmobot es la primera de las iniciativas de gamificación en el aula de este trabajo de investigación. Se financia gracias a una ayuda de la Universidad de Huelva para fomentar proyectos de innovación educativa. La competición que se celebró en Madrid

Tabla 1.1
Cosmobot 2009

Ámbito: Nacional
Lugar: CosmoCaixa Madrid
Año: 2009
Modalidades: Robots seguidores de líneas y luchadores de sumo
Número de participantes: 29
Web: http://www.roboticspot.com/especial/cosmobot2009/
Publicación: Carpio y otros 2011 [**?**]

en la sede de CosmoCaixa los días 28 y 29 de marzo de 2009. En esta edición se presentaban dos pruebas diferentes: luchadores de sumo y velocistas seguidores de linea. La prueba de velocistas en la que participamos, consistía en recorrer a máxima velocidad un circuito dibujado con líneas negras sobre fondo blanco que servían de guía a los robots participantes. El robot, utilizando algún tipo de sensor tenía que reconocer estas líneas y seguirlas a la mayor velocidad posible sin llegar a salir de la pista, delimitada por líneas rojas.

Competición de Inteligencia Artificial y Robótica 2010
Competición de Inteligencia Artificial y Robótica 2010 (CIAR 2010), celebrada en la Universidad de Huelva y organizada por varios compañeros y el doctorando pertenecientes a los departamentos de Tecnologías de la Información y de Ingeniería Electrónica, Sistemas Informáticos y Automática de la Universidad de Huelva. Esta competición consta de tres modalidades: diseño de robots, seguidores de línea y carreras de coches en entorno simulado. La modalidad de diseño de robot premiaba la creatividad de los diseños, la originalidad, la funcionalidad y el modo de construcción de los prototiopos. La prueba de seguidores de línea tenía consistía, al igual que en Cosmobot en seguir un circuito dibujado con líneas a la mayor velocidad posible. Y por último la competición de carreras simuladas de coches consistió en la programación de un agente que controlase un coche de carreras en un entorno virtual 3D.

First Lego League
Competición internacional con pruebas de diferentes ámbitos. En este caso participamos en la organización de la prueba provincial de Huelva en la edición 2011/2012 y como colaboradores en la edición 2012/2013. Cada edición se plantea un reto diferente que hay que resolver utilizando una serie de componentes

Tabla 1.2
Competición de Inteligencia Artificial y Robótica 2010

Ámbito: Local
Lugar: Escuela Técnica Superior de Ingeniería, Universidad de Huelva
Año: 2010
Modalidades: Robots seguidores de líneas, carreras
de coches virtuales, diseño de robots.
Número de participantes: 12
Web: http://goo.gl/upakEo

Tabla 1.3
First Lego League

Ámbito: Provincial/Internacional
Lugar: Universidad de Huelva
Año: 2011/2012 y 2012/2013
Modalidades: Diseño de robots para resolver un reto
Número de participantes: 100 (aproximadamente)
Web: http://www.firstlegoleague.es/

de la compañía Lego y su unidad de control Lego Mindstorms. Cada año la temática de la competición es diferente. En el año 2011 la competición tenía el título de "Food Factor´´ y estaba relacionado con la problemática de la alimentación, producción, almacenamiento o distribución de alimentos. La edición 2012 tenía como título "Senior solutions´´ y pretendía motivar a los participantes a reflexionar sobre las necesidades de los mayores y a pensar en posibles soluciones.

AI Challenge Ants 2011

Competición organizada por la Universidad de Waterloo y con el patrocinio de Google. El año 2011 se celebra su cuarta edición siendo las anteriores: Rock Paper Scissors otoño 2009, Tron invierno 2010, Planet Wars otoño 2010, Ants otoño 2011. El reto de la cuarta edición consistía en organizar una comunidad de hormigas con el objetivo de conquistar los hormigueros enemigos. Sobre un mapa se sitúan diferentes comunidades de hormigas, cada una de ellas con un número de hormigueros. En el mapa se distribuye comida de forma aleatoria que las hormigas pueden capturar. Cada vez que una hormiga alcanza la comida, del hormiguero sale una nueva hormiga. De esta forma se

Tabla 1.4
IA Challenge Ants 2011

Ámbito: Internacional
Lugar: Online, sitio web de la organización
Año: 2011
Modalidades: Manejo de una comunidad de agentes robóticos
Número de participantes: 7.897
Web: http://ants.aichallenge.org/
Publicaciones: Carpio y otros 2013 [**?**] y Carpio y otros 2014 [**?**]

consigue que la comunidad de hormigas crezca. Sin embargo el objetivo último de la prueba no es que la comunidad sea muy grande, sino que se lleguen a conquistar los hormigueros enemigos. Gana el equipo que consigue conquistar más hormigueros. En este caso era importante diseñar estrategias de defensa, de ataque, de captura de alimentos y de captura de hormigueros enemigos. Además las estrategias debían ser muy rápidas ya que la ejecución se organizaba en turnos de 1000 ms, lo que requería un gran esfuerzo de optimización de los algoritmos. Cabe destacar de esta competición que en ella participaban estudiantes de todos los rincones del mundo, trabajadores de empresas prestigiosas como Google, y estudiantes de las mejores universidades del mundo Stanford, MIT, EPFL entre otras.

Hello World Open 2014

En esta ocasión el reto consistió en programar un agente robótico capaz de correr en una pista virtual tipo Scalextric, en la que los coches circulan fijos a una línea de la pista. Lo interesante de este reto es que la física cambiaba de un circuito a otro, por lo que era importante intentar descubrir las características de cada pista antes de empezar. Además, se daba la circunstancia de que en el juego simulado, si la velocidad en la curva era demasiado elevada, el coche era expulsado de la pista, de forma que el coche perdía todas sus posibilidades de ganar la carrera. Uno de los retos importantes en este caso, adaptar la velocidad a la máxima posible sin llegar a salir de la pista.

Tabla 1.5
Hello World Open 2014

Ámbito: Internacional
Lugar: Online, sitio web de la organización
Año: 2014
Modalidades: Manejo de una comunidad de agentes robóticos
Número de participantes: 2.520 equipos
Web: https://2014.helloworldopen.com/

Tabla 1.6
Asignaturas por curso y competiciones

Curso	Asignaturas	Competición
2004/2005	PD, IIA	
2005/2006	PD, IA, IAeIC	
2006/2007	PD, IA, IAeIC	
2007/2008	PD, IA, IAeIC	
2008/2009	PD, IAeIC, LIA	Cosmobot 2009 [**?**]
2009/2010	PD, IAeIC, LIA	CIAR 2010
2010/2011	PD, LIA	ANTS 2011 [**? ?**]
2011/2012	IAeIC, LIA, MD	FLL 2011/2012
2012/2013	IAeIC, RC	FLL 2012/2013
2013/2014	RC, MAC	HWO 2014
PD: Programación declarativa IA: Inteligencia Artificial IIA: Introducción a la Inteligencia Artificial LIA: Laboratorio de Inteligencia Artificial IAeIC: IA e Ingeniería del Conocimiento MD: Minería de datos RC: Representación del Conocimiento		

2 OBJETIVOS

Índice

2.1 OBJETIVOS DE ESTA TESIS

El objetivo de esta tesis es crear una metodología que permita mejorar la experiencia docente en el ámbito de la IA/IC. Esta motodología mejora uno de los aspectos imprescindibles en el proceso de aprendizaje como es la motivación. A continuación se describen los sub-objetivos de la tesis:

2.2 OBJETIVOS

Los objetivos que esta tesis quiere validar son los siguientes:

Objetivo 1: Probar que la inclusión de competiciones en el aula puede mejorar la experiencia de aprendizaje de la IA

Las competiciones en IA pueden favorecer el aprendizade de técnicas que tradicionalmente se impartían de una forma teórica con baja implicación de los alumnos.

Objetivo 2: Proponer una metodología que ayude a los profesores responsables de asignaturas relacionadas con la IA a mejorar la experiencia de aprendizaje de sus alumnos

Describir los recursos necesarios para desarrollar esta metodología de forma práctica.

Objetivo 3: Validar la metodología a través de dos experiencias reales en el aula

Para validar la metodología se han realizado diferentes experiencias en el aula cuyos resultados han sido publicados en diferentes revistas científicas.

Objetivo 4: Validación de la metodología a través de la publicación de un artículo científico con la participación de alumnos

Como objetivo final de la tesis, se ha realizado un trabajo de investigación por parte de alumnos que han trabajado en este nuevo modelo de aprendizaje.

2.3 ESTRUCTURA DE LA TESIS

Este capítulo ofrece una introducción a esta tesis, incluyendo su motivación y las preguntas a abordar.

A continuación se expone la estructura del resto de capítulos:

El primer paso de esta tesis ha consistido en el estudio de las técnicas de IA/IC utilizadas en el ámbito universitario. Fruto de este trabajo inicial se ha publicado el primer trabajo científico en revista internacional [**?**]

El segundo trabajo [**?**], el tercer trabajo [**?**] y el cuarto [**?**].

3

METODOLOGÍA

Índice

3.1 METODOLOGÍA

Con el fin de llevar a cabo este trabajo de investigación, hemos seguido como guías metodológicas principales las buenas prácticas del método científico y, en lo referente a desarrollos, las buenas prácticas de la Ingeniería del Software y la Ingeniería de los Computadores. Teniendo esto como marco general, hemos estructurado nuestro trabajo en las siguientes etapas:

1. Adquisición de conocimientos relacionados con la investigación, el método científico, diseño de experimentos, análisis de datos y resultados en el ámbito de la IA
2. Búsqueda de competiciones que se ajusten al periodo lectivo
3. Breve estudio de las características de la competición
4. Propuesta de la actividad docente de gamificación al alumnado
5. Diseño de la experiencia de gamificación en la enseñanza de la IA en ingeniería con las siguientes etapas:
    1. Diseño de las encuestas para evaluar los diferentes elementos
    2. Planificación de la experiencia docente
6. Recopilación de datos previos a la experiencia
7. Puesta en marcha de la actividad docente
8. Recopilación de datos posteriores a la experiencia
9. Análisis de los datos obtenidos
10. Análisis de las conclusiones
11. Publicación en revistas de impacto de las experiencias

La primera fase de la metodología es el conocer el modo de elaborar un artículo científico en el ámbito de la IA. Para ello, en primer lugar, se realiza una revisión bibliográfica de diferentes técnicas, en nuestro caso las Redes Neuronales y concretamente los Mapas Auto-organizativos (SOM), los algoritmos genéticos y la lógica borrosa. Esta fase es fundamental, ya que sin estos conocimientos no sería posible diseñar la actividad de gamificación en el aula como un experimento científico que nos permita publicar resultados en una revista de prestigio (con clasificación en JCR). Como parte de esta primera etapa se elabora

un trabajo científico en colaboración con otros compañeros en el ámbito de la lógica borrosa y la computación evolutiva [**?**].

Una vez conocido como aplicar el método científico y con la experiencia de una primera publicación, necesitamos adaptar esos conceptos aprendidos al estudio de un experimento en el aula. El método científico como es bien sabido se basa en dos pilares fundamentales: el principio de reproducibilidad y de refutabilidad. Es decir, necesitamos describir la experiencia de forma que pueda ser reproducible y además tenemos que publicar el trabajo realizado de forma que la comunidad científica pueda poner en marcha la experiencia y corroborar o no la certeza de lo expuesto. Al trabajar con estudiantes, a diferencia con la experimentación con un conjunto de datos, la forma de reproducir la experiencia nunca producirá resultados exactamente iguales. Por esta razón, este tipo de experiencias nos obliga a considerar la reproducibilidad desde un punto de vista no tan estricto al que tendríamos considerando por ejemplo un algoritmo aplicado a un conjunto de datos conocido. Superada esta consideración, y con la fuerza de los beneficios observados en los alumnos que participan en competiciones de IA, decidimos dar un tratamiento científico a la experiencia con la esperanza de poder publicar los resultados obtenidos en revistas de prestigio.

Para poder poner en marcha una experiencia de gamificación en el aula necesitamos hacer coincidir el periodo lectivo con alguna de las competiciones relacionadas con la IA. Esto no siempre es posible, por lo que en algunos casos, como sucedió en CIAR 2010, los propios profesores toman parte activa organizando una competición para que los alumnos puedan participar. Esta opción requiere de un gran esfuerzo por parte de los profesores organizadores, por lo que no siempre será una alternativa adecuada. La otra opción más factible, requiere conocer competiciones de IA y para ello es importante disponer de una red de contactos con interés en el mundo de las competiciones en IA. En nuestro caso ha sido esta red de contactos la que nos ha permitido conocer nuevas iniciativas o reediciones de competiciones anteriores que hemos podido encajar dentro del periodo lectivo de las asignaturas de IA impartidas. Es importante destacar en este punto que hay competiciones que se celebran durante varios años seguidos y después dejan de celebrarse por algunos años (como es caso de AI Challenge) o bien otras nuevas surgen (como el caso de Hello World Open Competition que comienza en 2014 y tiene prevista una nueva edición en 2016).

Una vez decidido en que competición participar, el siguiente paso es hacer un pequeño estudio de la competición. Es importante conocer las reglas, lenguajes de programación que podemos utilizar, las fases de la competición, los plazos de inscripción y finalización, recursos necesarios, obligatoriedad o no de desplazarse al lugar del evento, recursos disponibles (programas de ejemplo), foros de discusión, etc. Con toda esta información el profesor debe valorar la viabilidad de la puesta en marcha de la actividad dentro del aula. En este sentido, la experiencia puede ayudarnos a decidir la viabilidad o no de la puesta en marcha de la actividad.

A continuación, el profesor plantea a los alumnos la posibilidad de realizar la actividad de gamificación en el aula. En [**?**] se pone de manifiesto que los resultados son mejores cuando son los alumnos tienen la posibilidad de decidir si participan en la competición. De esta forma los alumnos aceptan el reto propuesto y lo toman como un proyecto personal. El mismo método se aplicó en [**?**] y los resultados en cuanto a motivación e implicación fueron muy positivos.

Una vez aceptado el reto por parte de los alumnos, comienza el proceso de planificación de la actividad por parte del profesor. Con la idea de no alterar demasiado la marcha habitual del curso en cuanto a sus actividades y sus contenidos, por lo general lo que hemos hecho ha sido concentrar el trabajo en un periodo corto. En la experiencia descrita en [**?**], el trabajo se concentra principalmente en la semana previa a la competición. Durante esta semana, los participantes organizaron reuniones de trabajo que ocuparon casi todo un fin de semana. De esta forma los estudiantes aprendieron a organizar bien el tiempo, comprobando cuáles son sus límites y en qué momento es mejor hacer un descanso para que las horas de trabajo vuelvan a ser productivas. Con este método organizativo conseguimos alterar mínimamente la planificación establecida para el curso. Solo se introdujeron algunas sesiones en las que se informó de las reglas de la competición [**? ?**] y algunas sesiones para realizar algunas tareas específicas [**?**] como por ejemplo para el diseño de la placa de circuito impreso PCB o para la fabricación y montaje de la placa de control del robot móvil. En el caso de ANTS 2011 se organizaron un par de sesiones para mostrar cómo crear un agente robótico básico a partir de los ficheros proporcionados por la organización y en el fin de semana anterior se organizó una sesión de trabajo que ocupó casi todo el fin de semana.

Una vez introducido el problema a resolver y puesto que algunos de los conceptos impartidos en la asignatura se podían aplicar al agente robótico, los alumnos encontraban rápidamente la utilidad de lo explicado, como por ejemplo en el caso de encontrar un camino mínimo hacia la comida o el hormiguero enemigo en ANTS 2011 [**?** **?**] con el algoritmo A*. En cursos anteriores, era necesario introducir ejemplos, no siempre cercanos a los intereses de los alumnos, que hacía difícil que ellos pudiesen comprobar la utilidad real de la técnica que se trata de enseñar, lo que derivaba en falta de interés y motivación. Sin embargo, al tratar de resolver un reto planteado los alumnos encuentran rápidamente la aplicación de lo que están aprendiendo y al ver la utilidad adquieren los conocimientos de una forma más rápida y posiblemente más duradera.

Podemos resumir la metodología de la siguiente forma:

- Mínima alteración de la estructura habitual de la asignatura (contenidos y planificación temporal)
- Concentración del trabajo en sesiones de fin de semana
- Aplicación de los conceptos aprendidos en el desarrollo del agente para la competición (A*, algoritmos evolutivos, lógica borrosa, etc.)
- Recopilación de datos de las encuestas de opinión de los alumnos
- Análisis de los resultados

4

DISCUSIÓN Y RESULTADOS

Índice

4.1 DISCUSIÓN Y RESULTADOS

En este punto, debemos tomar perspectiva y analizar lo conseguido en cada una de las experiencias realizadas. En primer lugar empezaremos por el primer trabajo publicado [**?**]. En este trabajo se establecen las bases para elaborar un texto científico. Para el trabajo posterior, no es tan importante lo obtenido para determinar la estructura de las reglas y los parámetros de una función de pertenencia, sino como se deben diseñar los experimentos, evaluar los resultados y extraer las conclusiones sobre los datos obtenidos. Posteriormente, se utilizarán las técnicas de computación evolutiva en el trabajo [**?**] para evolucionar los parámetros de un agente en la competición ANTS 2011, desarrolladas también en este primer trabajo. Este primer trabajo y otras colaboraciones realizadas en los años siguientes establecen las bases para el trabajo de investigación posterior.

Durante los años posteriores, que podemos denominar el periodo de exploración, se buscan diferentes campos a los que poder aplicar las técnicas de IA aprendidas, al mismo tiempo que el trabajo de la enseñanza de la IA proporciona experiencias que posteriormente nos llevará a la puesta en marcha de experiencias de gamificación con el fin de paliar las deficiencias observadas en el proceso de aprendizaje.

El siguiente trabajo [**?**] supone el inicio este apasionante mundo de la puesta en práctica de la gamificación en la enseñanza de la ingeniería, en este caso en la competición de robots seguidores de línea en Cosmobot 2009. Tras desarrollar la experiencia tal y como se describe en el artículo y desde la perspectiva que nos da el tiempo, obtenemos dos resultados muy importantes. El primero es una valoración positiva por parte de los alumnos reflejada en las encuestas realizadas [**?**] y el segundo es la decisión de continuar en esta línea tras los éxitos obtenidos. En esta ocasión la experiencia afecta a un número reducido de alumnos y nos planteamos realizar futuras experiencias que nos permitan abarcar a un mayor número de estudiantes, como así sucedió posteriormente en [**?**]. Un elemento a destacar también de este trabajo es la colaboración en la redacción del artículo de los alumnos que participaron en la competición. De esta forma, los estudiantes también tuvieron la oportunidad de iniciarse en la redacción de textos científicos y ver finalmente publicada su experiencia.

Después de la experiencia de Cosmobot 2009, se ponen en marcha la experiencia de CIAR 2010 que no llega a generar una publicación científica ya que en esta ocasión nos centramos más en la organización del evento y no tanto en la puesta en marcha de experiencia dentro del aula. Sin embargo, esta nueva competición de IA, refuerza la idea de continuar en esta misma línea y nos llevará a realizar la próxima experiencia en el aula en 2011 con la competición ANTS 2011.

ANTS 2011 supone un paso más en la puesta en marcha de experiencias de gamificación en la docencia en ingeniería. En esta ocasión se amplía el número a 19 participantes de dos cursos diferentes (3er y 4º curso de ingeniería informática) y se aplican diferentes metodologías en cada curso, lo cual nos permite comparar resultados de las metodologías aplicadas [**?**]. Además, se analizan datos de cursos anteriores llegando a contabilizar datos de 83 estudiantes. Se analiza el interés y la motivación, la adquisición de conocimientos, el desarrollo de habilidades, la dificultad y cargar de trabajo. En cuanto a los conocimientos adquiridos, todos los alumnos indican tener un mayor nivel de conocimientos después de la experiencia. En la tabla se muestra un notable incremento de las calificaciones, principalmente en alumnos de 4º curso de ingeniería. Los alumnos afirman de forma contundente que esta experiencia permite la adquisición y consolidación de nuevos conceptos teóricos, ofreciéndoles nuevas formas de resolver problemas. Además, los alumnos califican de forma positiva el aprendizaje de nuevos lenguajes de programación. Comprobamos que la percepción de los alumnos de 4º curso es mejor que la de los de 3er curso debido posiblemente a que los primeros tienen mayor conocimiento en materias de IA. En cuanto al interés y la motivación, detectamos un aumento significativo en la opinión de los estudiantes. Esto indica que la competición influencia de forma positiva a los estudiantes en el estudio de las materias del curso. Descubrimos además, que esta percepción no solo afecta al curso en el que se realiza la experiencia, sino que se hace extensiva a la titulación y la Universidad. En cuanto a la percepción sobre la carga de trabajo, detectamos que los alumnos de 3er curso consideran que la actividad es más compleja que los alumnos de 4º curso. Consideramos que esta percepción es debida a que los alumnos de 3er curso estudian por primera vez materias de IA. Sin embargo todos los alumnos valoran positivamente la experiencia dentro del contexto Universitario. Por último, todos los alumnos valoran la experiencia global de forma positiva y satisfactoria.

Otro resultado de la experiencia en ANTS 2011 es el la publicación en formato de artículo científico de un método de ajuste del agente software mediante técnicas de computación evolutiva [**?**]. Este trabajo realizado en colaboración dos alumnos participantes en la experiencia. El trabajo presenta el diseño de un agente para la competición a partir de una combinación de dos comportamientos básicos (Greedy y Lefty) y se utiliza un algoritmo genético GA para hacer un ajuste de los parámetros y así modificar el comportamiento del agente. El agente se prueba en seis mapas diferentes ofrecidos por la organización de la competición y contra otros tres robots creados por otros usuarios para el evento. Los resultados de este trabajo nos indican [**?**] que el ajuste paramétrico de un agente utilizando un algoritmo genético mejora la versión básica, llegando a ganar en ocasiones a competidores diseñados por otros participantes que terminaron en posiciones de cabeza de la competición (993 y 165). La conclusión final del trabajo fue que la optimización de parámetros utilizando un algoritmo genético mejora significativamente las prestaciones del agente en los juegos en tiempo real y que esta técnica puede obtener mejores resultados con buenas estrategias de planificación.

5

CONCLUSIONES

Índice

5.1 CONCLUSIONES

Tras analizar los diferentes objetivos propuestos y los resultados obtenidos, podemos concluir que la metodología propuesta para el uso de la gamificación en el aprendizaje de la IA es adecuada y que obtiene mejoras significativas en aspectos como la motivación y la mejora de los conocimientos adquiridos. Que es posible introducir de forma satisfactoria este tipo de experiencias dentro de los planes de estudios tradicionales y que es posible compatibilizar el enfoque más tradicional de la enseñanza combinándolo con juegos competitivos de forma que mejore la experiencia de aprendizaje de los estudiantes en el ámbito de la IA.

Además de las experiencias descritas en las publicaciones presentadas en esta tesis, se han desarrollado nuevas experiencias que pretenden seguir profundizando en el uso de la gamificación en el aula.

Nuestras líneas de trabajos futuros están orientadas a

- El uso de la gamificación con el fin de realizar de forma colaborativa tareas complejas.
- Ampliar el ámbito de futuros estudios de forma que se puedan implicar diferentes Universidades y obtener así datos más precisos sobre la influencia de este tipo de experiencias.
- Analizar el posible uso de la gamificación para en otros ámbitos distintos a la educación como pueden ser el emprendimiento. En esta línea, para el curso 2015/2016 ya ha comenzado la organización de una competición de vehículos eléctricos solares denominada Desafío Solar Costa de la Luz 2016 que implicará a estudiantes de Ingeniería y de enseñanzas medias.

Parte II

PUBLICACIONES

5.2 EVOLVING TWO-DIMENSIONAL FUZZY SYSTEMS

A continuación se detallan los datos del artículo publicado relacionado con esta sección de la disertación.

Título: **Evolving two-dimensional fuzzy systems**
Revista: **Fuzzy Sets and Systems**, 2003; Factor de impacto: **0,577**;
Autores: Víctor M. Rivas, J.J. Merelo, I. Rojas, G. Romero, P.A. Castillo, **J. Carpio**

Relevancia de la revista:

Nombre de la categoría	Revistas en la categoría	Posición en la categoría	Cuartil en la categoría
COMPUTER SCIENCE, THEORY	70	44	Q3
MATHEMATICS, APPLIED	153	85	Q3
STATISTICS & PROBABILITY	75	42	Q3

ELSEVIER

Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

Fuzzy Sets and Systems 138 (2003) 381–398

FUZZY sets and systems

www.elsevier.com/locate/fss

Evolving two-dimensional fuzzy systems

Víctor M. Rivas[a,*,1], J.J. Merelo[b,1], I. Rojas[b], G. Romero[b,1], P.A. Castillo[b,1], J. Carpio[b,1]

[a] *Dpto. de Informática, Universidad de Jaén, E.P.S., Avda. de Madrid 35, E.23071, Jaén, Spain*
[b] *Dpto. de Arquitectura y Tecnología de Computadores, Universidad de Granada, Spain*

Received 15 September 1999; received in revised form 9 September 2002; accepted 24 September 2002

Abstract

The design of fuzzy logic systems (FLS) generally involves determining the structure of the rules and the parameters of the membership functions. In this paper we present a methodology based on evolutionary computation for simultaneously designing membership functions and appropriate rule sets. This property makes it different from many techniques that address these goals separately with the result of suboptimal solutions because the design elements are mutually dependent. We also apply a new approach in which the evolutionary algorithm is applied directly to a FLS data structure instead of a binary or other codification. Results on function approximation show improvements over other incremental and analytical methods.

Keywords: Fuzzy systems; Genetic algorithms; Evolutionary algorithms; Hybrid methods; Function approximation

1. Introduction

Since the introduction of the basic methods of fuzzy reasoning by Zadeh [23], and the success of their original application to fuzzy control, fuzzy logic and its application to fuzzy control have been widely studied. However, certain important questions still remain open, including: (1) the selection of the fuzzy rule base; (2) the subjective definitions of the membership functions; and (3) the structure of the fuzzy system (number of rules and membership functions).

The transfer function of a fuzzy system is not based on a mathematical model; it is given by the definition of fuzzy rules and fuzzy sets of linguistic variables (for each membership function). The fuzzy rules and the fuzzy sets are designed on the basis of the human operator's experience, decisions

* Corresponding author. Tel.: +34-953-012344; fax: +34-953-002420.
E-mail addresses: vrivas@ujaen.es (V.M. Rivas).
URLs: http://pagina.de/vrivas, http://geneura.ugr.es
[1] GeNeura Team.

0165-0114/03/$ - see front matter
PII: S0165-0114(02)00483-9

and control actions. In conventional expert systems the operator cannot often clearly explain why he/she acts in a certain way. Furthermore, there is no reason to believe that an operator's control is optimal. Then an automatic design method based on a set of examples for the input/output relationship becomes important. Such a set of examples is commonly called the referential data set.

In general, creating a fuzzy logic system (FLS) involves designing the structure of the rules of the system and the parameters of the membership functions. Most techniques deal with these separately, which may result in a suboptimal solution because the design elements are mutually dependent. For example, Genetic Algorithms (GAs) can firstly be used to determine the rules of the system and then, in a second stage, to tune the parameters of the linguistic values, as in [10]. We propose optimizing these parts simultaneously using Evolutionary Computation techniques with a new method slightly different from those presented in [18–20]. This will be discussed later.

The rest of the paper is organized as follows: the next section deals with the current state of the art, including an introduction to Evolutionary Algorithms (EA); Section 3 describes the EA used in this work, including genetic operators, the algorithm itself and the evolutionary computation library, *Evolutionary Objects* [13]. After this, Section 4 describes some experiments and their results; and finally, Section 5 presents some conclusions and future lines of work.

2. State of the art

2.1. Evolutionary algorithms

Evolutionary algorithms (EAs) represent a set of strategies to efficiently search for near optimal solutions in hard to search spaces imitating natural genetics. Depending on the problem, the optimal solution may be one that either maximizes or minimizes a given function. Nevertheless, as minimization problems can be easily changed into maximization ones, EA terminology tends to refer only to the latter.

EAs can be characterized by the following features ([14]):

- A genetic representation for potential solutions to the problem. Solutions are called individuals. However, in this work we apply a new approach in which individuals are not encoded into a chromosome (as they usually are), but in which the EA can directly deal with the solutions as they are, i.e. as FLSs.
- A way to create an initial population of potential solutions. The most common method is by means of a random generator.
- An evaluation function that plays the role of the environment rating solutions in terms of their fitness. In general, the best individuals are those which have the highest fitness.
- Genetic operators are used to manipulate the population's genetic composition. New individuals are created by applying these operators to the previously existing ones. The best individuals should generate more offspring than the rest.
- A set of parameters that provide the initial settings for the algorithm: population size, probabilities employed by the genetic operators, termination conditions, and probably, a set of constraints for individuals.

EAs usually implement three basic operators to manipulate the population's genetic composition: selection, recombination and mutation.

Selection. Selection is the process by which individuals with higher fitness values have a higher probability of being chosen to reproduce, generating and offspring, than individuals with smaller fitness values. Because of this, it is considered a diversity-destruction operator. The most common method used is the weighted roulette selection.

Recombination. Recombination is the process by which one or more new individuals are created using parts from two or more parents. The underlying idea is that the optimal solution is composed of several optimal parts (or *building blocks* [7]), so the massive interchange of information between individuals may lead to better and better new ones. Like selection, recombination operator also decreases diversity.

Mutation. Mutation operators alter, in a random way, the structure or the stored information of the individual to which they are applied. They increase the diversity of population, providing a mechanism to escape from local optima and premature convergence. High rates of these operators allow better exploration of the search space, but make convergence slower and can result in random search.

2.2. Applications of EA to FLS design

The properties of EAs make them a powerful technique for selecting high performance parameters for FLSs. Previous work focused basically on optimizing the FLS parameters and on reducing the number of the rules. In this paper EAs are used to search for an optimized subset of rules (both the number of rules and the rule values) from a given knowledge base to achieve the goal of minimizing the number of rules used while maintaining the FLS performance. EAs will eliminate all unnecessary rules, i.e., those which have no significant contribution to improving system performance.

Recently, EAs have been combined with fuzzy logic and neural networks in the process of designing fuzzy systems. For a good review see [7]. Ishibuchi et al. [9], for instance, propose a hybrid approach where a set of fuzzy rules is first extracted from a trained neural network, and an EA is then used to select a small subset of rules from the extracted rule set. The fitness function of the EA is designed to minimize the number of selected rules and maximize the number of correctly classified examples. For example, Karr and Gentry [11] control the pH of an acid–base system with the fuzzy system's input membership functions manipulated by an EA.

Some other methods [21] combine EA and FLS in order to tune the parameters of the membership functions and outputs of a Takagi–Sugeno fuzzy rule base, and have been used for the approximation of one-input analytical functions.

On the other hand, the codification of rules and membership functions is typically done using vectors (1-D matrices) [2,16,12,8,11]; this is not the most natural way to do it because it keeps "natural" building blocks (for instance, four contiguous cells in the matrix) apart from each other. In this work FLSs are not coded that way, but implemented by 2-D matrices; this device allows us to represent rules with close antecedents together (after all these rules are generally activated at the same time, interfering with each other). This way they can be more easily transmitted to the offspring.

It should also be noted that several researchers have concentrated on using real number coding for chromosomal representation of individuals instead of traditional bit string based coding; and it is reported that for some problems, these techniques outperform the conventional bit string based EAs [3,5,22]. This partly supports our work, since we use real numbers for evolution.

A special mention has to be made to some papers in which EAs are used to model and/or tune complete fuzzy systems. By chronological order, the two first are due to Lian et al. [18,19]. In these works the authors developed a method to tune a neuro-fuzzy controller using genetic algorithms, applied to a set of problems like coupled-tank liquid-level control, unstable plants control and automatic car parking. The main difference in their method and the one presented in this paper is that the size of the controller and the number of parameters is fixed and fitted to the problem being solved at every moment. So this method implies a previous study of the problem to be solved and cannot be applied directly to any kind of problem with two inputs and one output. A second difference is that parameters, that are real values, are represented using bit strings, which is not the natural and logical way it should be done, as was discussed before. A more recent paper is the one by Setnes and Roubos [20], showing how GA can be used to create fuzzy systems applied to modeling and classification problems. Once more, the problem of how to represent the solutions has been solved in an unnatural way, since rule antecedents and consequents are stored sequentially in the chromosome. This adds a new problem, given that the genetic operators can produce solutions that violate the different constraints imposed to both the input space and the output space.

3. The evolutionary algorithm

To program this algorithm we used the **EO** [19] library (*evolutionary objects*) because of the facilities it offers to evolve any object (in the sense of object oriented programming) that can be assigned a cost or fitness function. **EO** is a library that defines the interfaces of several types of evolutionary algorithms, and includes several examples of their use. It is currently programmed in C++ but easily portable to any other object oriented language. It is open-source, and available from http://eodev.sourceforge.net.

The EO library directly evolves classes of objects, so there is no need to code them in a binary chromosome. In this work evolved objects are fuzzy logic systems (FLS) which are implemented as 2-D arrays; thus, some specific operators are needed to mutate and combine them in order to create new FLSs.

The following subsection explain the architectures of both the FLS and the EA that optimizes it.

3.1. The 2-D fuzzy logic system

In this work each FLS is implemented as a two-dimensional matrix storing two different things: (a) the centres of the triangular partition membership functions of two input variables, X and Y, and (b) the values of the output variable, Z. Thus, any 2-D matrix stores both the precedents and the consequents of the FLS rules.

A little more formally, a m by n FLS is implemented by a two-dimensional matrix:

$$M(m+1,n+1)=\begin{vmatrix} - & Y_1 & Y_2 & \cdots & Y_n \\ X_1 & Z_{1,1} & Z_{1,2} & \cdots & Z_{1,n} \\ X_2 & Z_{2,1} & Z_{2,2} & \cdots & Z_{2,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ X_m & Z_{m,1} & Z_{m,2} & \cdots & Z_{m,n} \end{vmatrix}$$

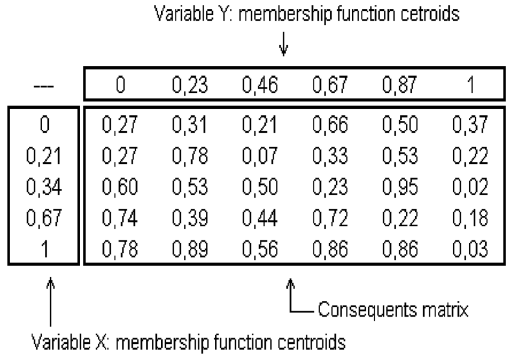


Fig. 1. Example of FLS implementation. In this case it corresponds to a 6×5 FLS.

with $m+1$ rows and $n+1$ columns, being the minimum number of columns and rows equal to 3 (i.e., $m_{\min}=2$ and $n_{\min}=2$.) while the maximum is one of the parameters the evolutionary algorithm must find. In this matrix, m corresponds to the number of membership functions related to input variable X, and n is the number of membership functions related to input variable Y. It must be taken into account that in any given EA generation the population will be composed of FLSs having different numbers of rows and columns.

Fig. 1 shows a typical FLS implemented as a 2-D matrix, where the three different parts in which this matrix can be split have been highlighted using boxes.

Each 2-D matrix implementing a FLS is divided into:

- $M[1,0]$ to $M[m,0]$ (First column). It stores the centres of the triangular partition membership functions of the first input variable, X, except the first cell, $M[0,0]$. Variable X takes values in the range $[x_{\min},x_{\max}]$, so the following must be true at any time:

$$x_{\min}=M[1,0]<M[2,0]<\cdots<M[m,0]=x_{\max} \quad \text{and} \quad m\geqslant 2.$$

- $M[0,1]$ to $M[0,n]$ (First row). As the first column but for the second input variable, Y. Again, the following conditions must be true:

$$y_{\min}=M[0,1]<M[0,2]<\cdots<M[0,n]=y_{\max} \quad \text{and} \quad n\geqslant 2.$$

- $M[1,1]$ to $M[m,n]$ (*Consequents matrix*, i.e, the whole matrix except the first row and the first column). Every cell of this submatrix stores a value for Z, the output variable, where:

$$z_{\min}\leqslant M[i,j]\leqslant z_{\max} \quad \forall i,1\leqslant i\leqslant m \text{ and } \forall j,1\leqslant j\leqslant n.$$

- The cell $M[0,0]$ does not represent anything, thus it is not used.

Using the above implementation, the FLS works using rules with the following form:

$$\text{IF } X \text{ is } M[i,0] \text{ AND } Y \text{ is } M[0,j] \text{ THEN } Z \text{ is } M[i,j]$$

The values $x_{\max}$, $x_{\min}$, $y_{\max}$, $y_{\min}$, $z_{\max}$ and $z_{\min}$ are parameters that must be provided to the algorithm.

3.2. Genetic operators

Using EO library to develop the EA allows us to directly evolve a FLS instead of its bitstring representation. For this reason we are not constrained to use traditional genetic operators. Thus, diversity-generation (mutators) and diversity-destruction (crossover-like or recombination) operators have been designed making use of specific problem knowledge. Moreover, since EO is implemented using C++, an Object Oriented language, operators act over the evolvable objects via their interfaces, and they never modify the objects directly. This property makes unnecessary the use of penalty functions or repair methods to deal with invalid objects generated by the operators: the FLSs themselves control that changes ordered by the operator are carried out in such a way that the resulting objects are always valid.

3.2.1. Recombination

This operator splices values from one matrix into another. Taking into account that these two FLSs do not necessarily have the same dimensions, the recombination cannot be performed in a trivial way.

The operator works as follows:

(1) It randomly chooses a FLS to be modified (we will call it M_r, or *receiver FLS*), and another that will provide the genes to be recombined (the M_d, or *donor FLS*). Only the first one, M_r, will be changed.
(2) It chooses a random block of cells from the consequent matrix of M_r. To specify this block we only need to randomly select two cells corresponding to the top left corner and bottom right corner of the block, respectively. Call these two cells $M_r[r_1, c_1]$ and $M_r[r_2, c_2]$. The values of the cells included in this block will be changed by values coming from M_d.
(3) For each cell $M_r[r_i, c_j]$, where r_i goes from r_1 to r_2, and c_j goes from c_1 to c_2, the operator does the following:
(4) From the first column of the donor, M_d, it selects the cell whose value is closer to $M_r[r_i, 0]$. Design that cell as $M_d[r_d, 0]$.
(5) From the first row of the donor, M_d, it selects the cell whose value is closest to $M_r[0, c_j]$. Design that cell $M_d[0, c_d]$.
(6) Finally, it changes the value stored in $M_r[r_i, c_j]$ by the one stored in $M_d[r_d, c_d]$. As can be seen, the donor FLS, M_d, remains unchanged.

An example of the action of this operator is shown in Fig. 2.

3.2.2. Addition of membership functions (size increment mutators)

There are two independent operators that modify the structure of the FLS they are applied to by adding a membership function to input variables. One of these operators affects he input variable X while the other operates with the input variable Y.

Fig. 3 graphically shows the effect of incrementing the number of rows. In order to do that, once the FLS to be changed has been randomly chosen the operator works as follows:

(1) It inserts an empty row in a random position, different from the first and last ones.
(2) It fills the cells of the new row with random values generated by a Gaussian function centered on the middle point between the corresponding to the preceding row and that corresponding to the following one.

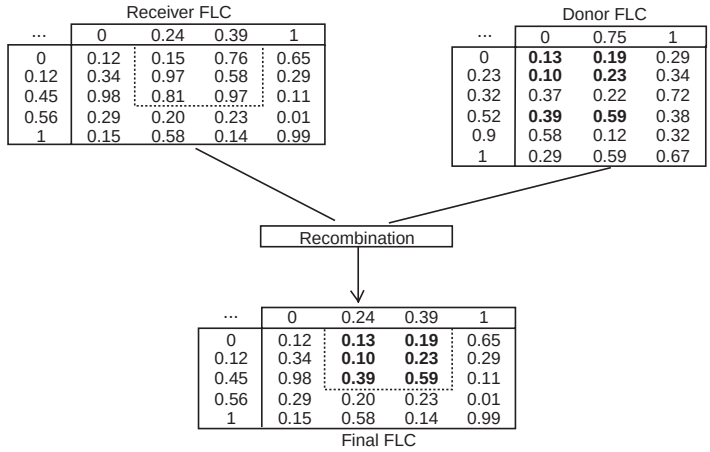


Fig. 2. Recombination operator: the resulting FLS has values in its consequents matrix belonging to both parents.

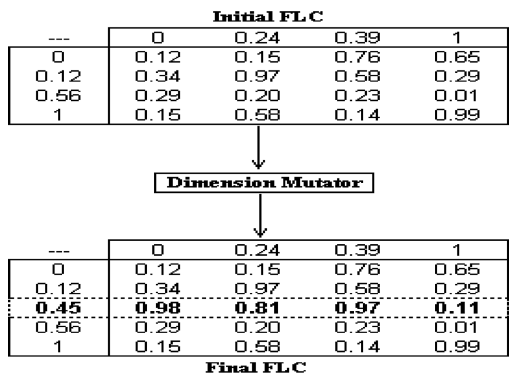


Fig. 3. Size increment mutator: a new row (membership function for X variable) is added to the FLS.

The way these operators work ensures that the resulting FLS is always valid, because the values for the centroid and consequents of the new row are set in a way that results into an ordered FLS (i.e., the value for the new centroid is greater than that of the preceding row and smaller than that of the following row), and thus is valid.

The algorithm for the operator that adds a new column is the same as the one presented above but dealing with columns instead of rows.

3.2.3. Removal of membership functions (size decrement mutators)

There are two operators that, as the previous ones did, also modify the structure of the FLS by decreasing the number of membership functions of input variable X or Y.

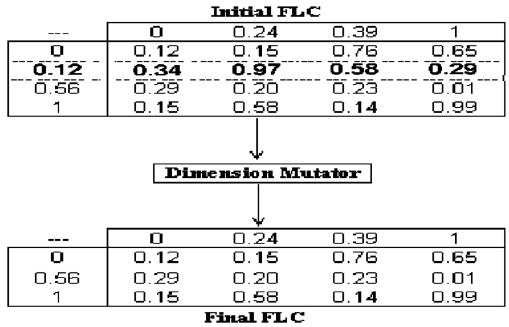


Fig. 4. Size decrement mutator: a row (membership function for the X variable) is deleted from the FLS.

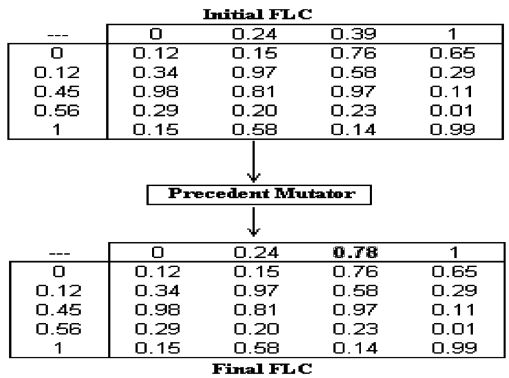


Fig. 5. Precedent mutator: the value for the centroid of the third membership function of input Y has been changed by the operator.

These operators work by choosing a row/column to delete (different from the first and last ones) and removing it (see Fig. 4).

Once again, they ensure that the resulting FLS is valid, given that (a) they cannot delete a row/column of a FLS that has the minimum number allowed (i.e., 3 column by 3 rows); (b) the first and last rows/columns cannot be chosen to be deleted; and (c) obviously, the FLS remains sorted once the row or column has been removed.

3.2.4. Modification of a centroid value (precedent mutators)

These two operators, one for input X and the other for input Y, modify one of the values stored in the FLS first row or first column, respectively.

Once an FLS has been selected, the operator that modifies centroids of variable Y carries out the following operations (see Fig. 5):

(1) It randomly selects a cell in the first row (because we are going to change a centroid of Y). This cell can be neither the first nor the last one.

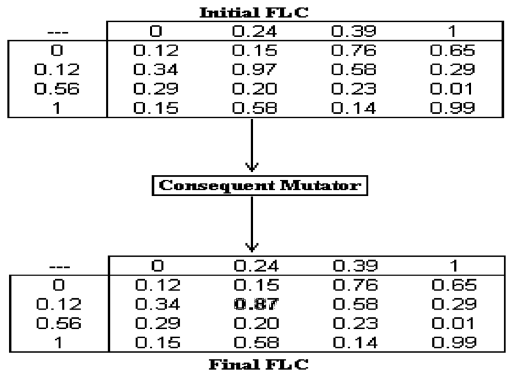


Fig. 6. Consequent mutator: one of the consequents is changed for a new valid value.

(2) It sets the selected cell to a random value, greater than the one in the precedent column and smaller than that in the following column. Once more, a Gaussian function centered on the current value, and with asymmetric widths (distances from current value to previous and next ones, respectively) is used.

The same algorithm, with rows and columns swapped, is used for the operator that changes the values of variable X.

As in the precedent cases these operators ensure that the resulting FLS is valid, because the first and last membership function centroids cannot be changed. Likewise, the way the new value for the cell is set ensures that the FLS remains sorted.

3.2.5. *Modification of a consequent value (consequent mutator)*

This operator works in the following way (see Fig. 6):

(1) It randomly selects a cell in the FLS consequent matrix (i.e., select a random $M[i,j]$, where $1 \leqslant i \leqslant m$ and $1 \leqslant j \leqslant n$).
(2) In that cell it introduces a new random value determined by a Gaussian function centered on the existing value, and varying from $z_{\min}$ to $z_{\max}$.

3.3. *Fitness function*

To calculate the fitness of each individual, a set (namely the *training set*) of input–output pairs from a known function is presented to it (known functions are shown in top left graphics of Figs. 7–10). The fitness assigned to the FLS is the inverse of the Normalized MSE distance from the known correct outputs to the outputs produced by the FLS, calculated using the training set, i.e.:

$$F_j = \frac{N}{\sum_{i=1}^{N} ((z_i - z_{ij})/(z_{\max} - z_{\min}))^2} \tag{1}$$

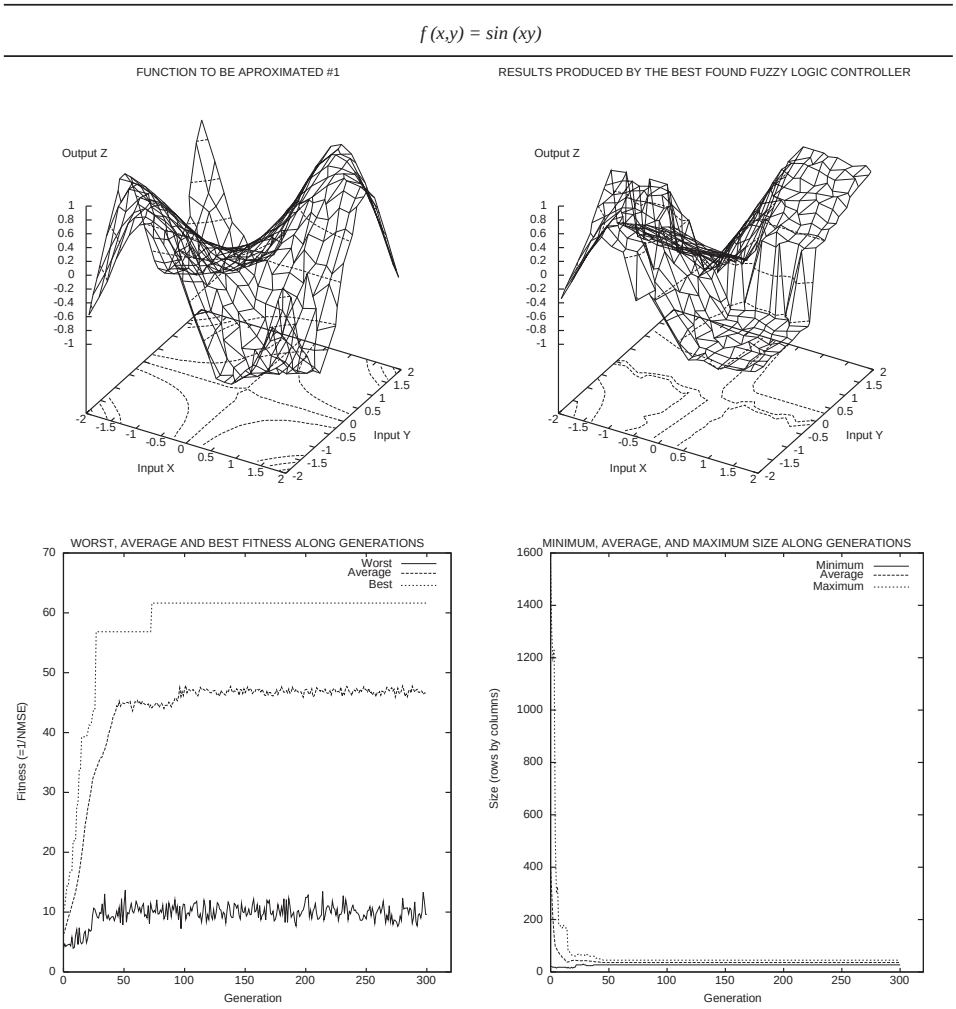


Fig. 7. Result number 1: function to be fitted (top left), solution found by the algorithm (top right), fitness evolution (bottom left), and size evolution (bottom right, logarithmic scale).

where F_j is the fitness for the individual number j, N is the number of samples in the training set, z_i is the correct output, and z_{ij} is the output provided by the FLS.

A special case is when $z_i = z_{ij},\ \forall i,\ 1 \leqslant i \leqslant m$, because F_j should be equal to $N/0$; so in this case, F_j is assigned a very high value.

It should be taken into account that FLS size does not intervene in fitness computation. It is the generalization error that is measured, which should resolve the matter since smaller networks usually generalize better.

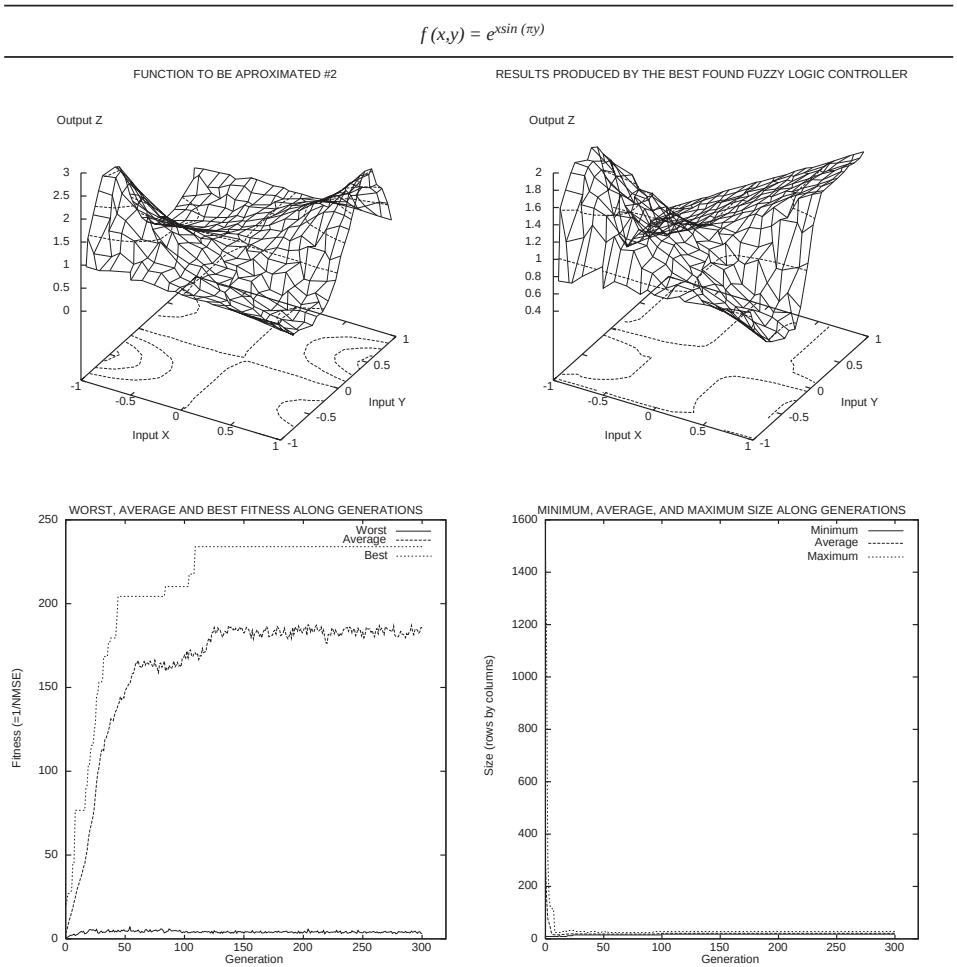


Fig. 8. Result number 2: function to fit (top left), solution found by the algorithm (top right), fitness evolution (bottom left), and size evolution (bottom right, logarithmic scale).

3.4. The algorithm

An evolutionary algorithm is used here consisting of elitist selection [4], fixed population size and the operators described. Here are its main steps:

(1) Create the first population, composed of p randomly generated individuals of random size (with an upper limit in the number of rows and columns, only for this first generation). Set the generation counter to 1.

$$f(x,y) = \frac{40e^{8((x-0.5)^2+(y-0.5)^2}}{e^{8((x-0.2)^2+(y-0.7)^2}+e^{8((x-0.7)^2+(y-0.2)^2}}$$

FUNCTION TO BE APROXIMATED #3 RESULTS PRODUCED BY THE BEST FOUND FUZZY LOGIC CONTROLLER

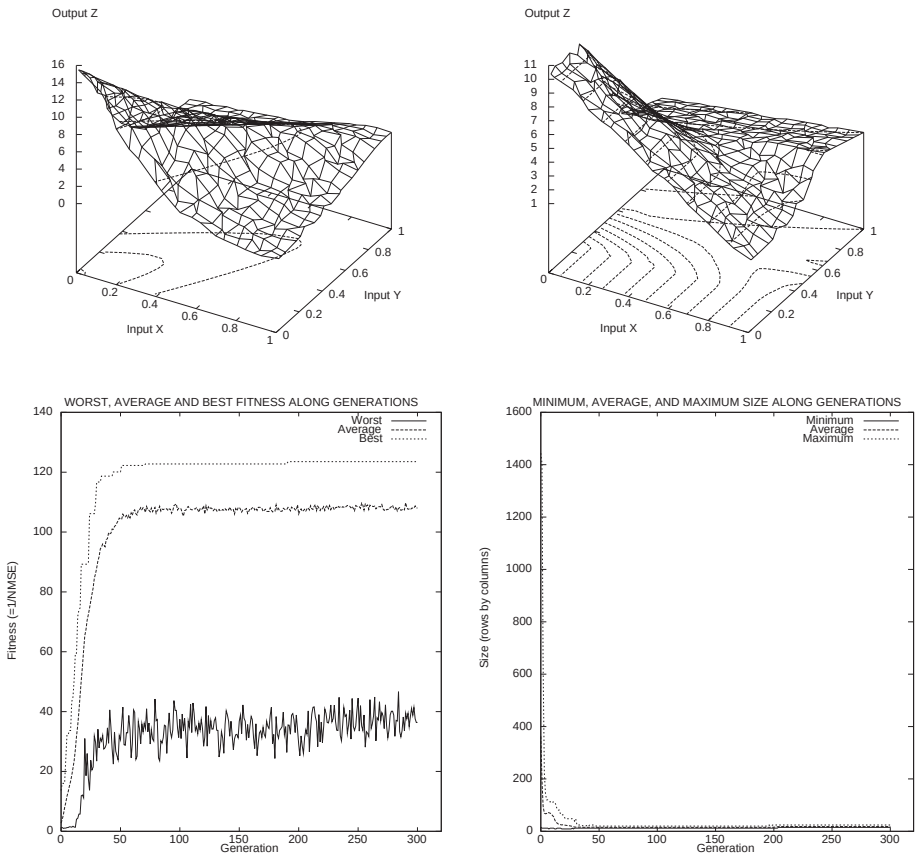


Fig. 9. Result number 3: function to fit (top left), solution found by the algorithm (top right), fitness evolution (bottom left), and size evolution (bottom right, logarithmic scale).

(2) Compute the fitness value of every individual.
(3) Sort the individuals from highest to lowest fitness values.
(4) Select the q best individuals (population elite subset). Delete the remaining $p - q$ individuals.
(5) Generate $p - q$ new FLSs, increasing population size up to p. Every new individual is created by (1) duplicating one individual in the elite subset, and after that, (2) applying one of the operators to the copy. The probability of one individual in the elite subset being selected for reproduction is related to its fitness value.

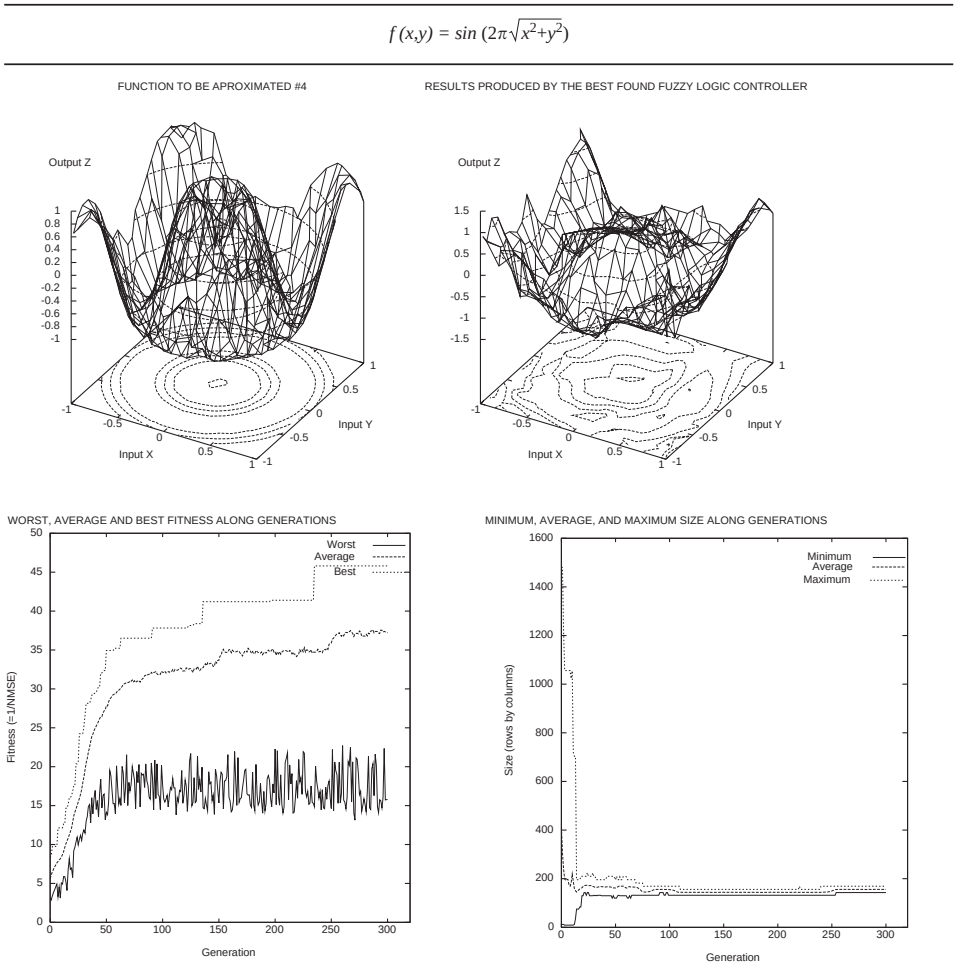


Fig. 10. Result number 4: function to fit (top left), solution found by the algorithm (top right), fitness evolution (bottom left), and size evolution (bottom right, logarithmic scale).

(6) Evaluate the new individuals.
(7) Increment the generation counter, and go back to step 3, unless a specified number of generations has been reached.
(8) Finally, the best FLS found is the first individual of the current (last) population (because they are sorted form highest to lowest fitness).

This algorithm is run with the following free parameters:

- Population size.

- Rate of individuals to be removed in every new generation (the *non-elite rate*).
- Application rates of every operator (henceforth, these are normalized to become probabilities).
- Maximum number of rows and columns for the first generation.
- Maximum number of generations.

Every operator has an associated probability of being applied that does not change during the application of the algorithm (although **EO** allows adaptive operator rates). Furthermore, every new individual is created by applying one, and only one, operator to the copy or duplicate of its parent, be it the recombination operator or any of the mutators.

As usual, the recombination operator is applied with a higher rate to allow mixing of building blocks, while mutation-like operators are applied with smaller rates in order to escape from random search. On the other hand, all the mutation-like operators share the same rate in order to balance the increase and decrease of FLS size. Additionally, a termination condition is fixed at a given number of generations to specifically limit the algorithm running time.

4. Experiments and results

The goal of these experiments was to validate the behavior of the operators created. Parameters of the algorithm were fixed to default values, although further study is necessary in order to establish which values are the best. Table 1 the values used for the parameters in the following examples.

In every experiment we attempted to obtain a FLS that estimates a known function. Several functions were used, although only four of them are shown here. For every function the algorithm was run three times with different random seeds in order to get average values and standard deviations. Functions have been taken from [15], chosen for being used in other works to compare results.

One file per function with 400 points was generated to carry out the experiments. Inputs X and Y were given equally spaced values. The resultant values for output Z were modified adding small quantity of random noise. Every time the algorithm was executed 320 randomly chosen points (80% of 400) were used as the training set, while the remaining 80 points were used as *validation set* used to test the generalizing capabilities of the FLS.

Taking into account that the algorithm deals with large matrices storing floating point numbers, it is obviously a very time consuming task. For this reason the number of individuals, generations and initial maximum rows/columns was set to small values allowing us to carry out several experiments

Table 1
Values for parameters

Parameter	Value
Population size	500
Non-elite rate	0.7
Recombination rate	0.1
Dimension mutator rates	0.01
Precedent mutator rates	0.01
Consequent mutator rate	0.01
Number of generations	300
Max. init. number of rows/columns	40

Table 2
Numerical results: fitness, generalization error, and size (rows by columns−1) of the best individual during the different runs of the algorithm for each function

Function ID	Fitness	Generalization error	Size
1	54 (±8)	0.0288 (±0.0007)	34 (±3)
2	197 (±35)	0.006 (±0.003)	41 (±40)
3	114 (±8)	0.0122 (±0.0011)	18.6 (±1.2)
4	34 (±12)	0.046 (±0.012)	117 (±36)

in a relatively short time: every experiment lasted about two hours on a computer with twin Pentium II 450 MHz, and using Linux as the operating system.

Figs. 7–10 show the original functions to be fitted (top left), the best output produced during the three executions of the algorithm in each problem (top right), fitness evolution along generations (bottom left), and size (rows by columns, less 1) evolution along generations (bottom right). Figure headers show the function formulas that generate the original graphics.

Table 2 shows average and standard deviation of final fitness, generalization error, and size (rows by columns, less 1) of the best FLS found (the best individual in the last generation). To calculate the generalization error once the EA has finished a validation set of input–output pairs, different from those used in the EA, was presented to the best FLS. The normalized MSE was computed using the outputs produced by the FLS and the known ones.

The results obtained show that the approach presented in this paper is appropriate to solve the problem. Thus, as usual, the evolutionary algorithm behaves very well in the task of finding good solutions (in these cases, function shapes are quickly learned by the genetic algorithm). However, fitting more precisely is more difficult and it is necessary to increase the number of generations in order to get more accuracy. But on the other hand, adding more generations might not be desirable because generalization might be negatively affected, resulting on the well known problem of overfitting, i.e., points in the training set would be approximated better and better at each generation, but approximation of points in the test set would be worst and worst.

The estimation of centroid values is quite good, as can be seen in the figures by analyzing the contours plotted when output Z is projected over the surface created by inputs X and Y. These contours show how the changes in the shapes of the original functions are reflected in the shape of the results provided by the best FLS. Fitting the values of centroids to the exact values used in the functions makes us face the same problem stated above. It should be done carefully, because once a given generation is reached the individuals would be fitting the errors associated to the points of the training set so that generalization would be performed very badly.

One of the most interesting results provided by this algorithm is related to the complexity of the solutions it finds. As Tables 2 and 3 show the final size of the FLSs did not grow up to the greatest possible value, as it might at first be thought. This is specially relevant because the way the fitness is calculated does not include explicit penalization of large individuals. Fitting the 400 points involved in every experiment can be exactly done if FLSs composed of $20 + 1$ rows and $20 + 1$ columns are used. This would possibly be the solution found if random search algorithms were used. But in our method the search is guided through the use of the different genetic operators together with the method to select individuals to be reproduced and individuals to be removed.

Table 3
Comparison between the algorithm presented in this paper and some others from differents sources. Columns show, for each function and algorithm, the normalized mean square error and number of fuzzy rules. For the column related to Cherkassky no information about size is reported

Function ID	Proposed algorithm		Pomares [15]		Rojas [17]		Cherkassky [1]
	NMSE	Size	NMSE	Size	NMSE	Size	
1	**0.0288**	34	0.047	36	0.089	64	0.033
2	**0.006**	34	0.008	48	—	—	0.016
3	**0.0122**	18	—	—	—	—	—
4	**0.046**	117	0.031	64	—	—	0.106

Table 3 compares the results obtained by the algorithm proposed in this paper with the results taken from Pomares [15], who developed an automatic method for fuzzy system design. Pomares' algorithm is based on mathematical analysis and uses gradient methods when any value has to be optimized. More precisely, Pomares' algorithm is divided into four steps: the first one calculates the optima consequents for the fuzzy rules for a fixed number of membership functions. The second step optimizes the central values of the membership functions and the consequent values at the same time. The third step analyzes the surface provided by the generalization error in order to determine in which variables is necessary to increment the number of membership functions, trying to improve the normalized mean squared error. On the last step, from the various configurations found in the process, the one which represents the best trade-off between accuracy and complexity is chosen as the final fuzzy system. On the other hand, the work by Cherkassky [1] corresponds to the method named *constrained topological mapping*, in which self-organizing maps were used to divide the input domain into different areas not connected with each other. Finally, Rojas [17] developed an algorithm for function approximation that optimized both the number of rules and the rules themselves. To do this, the rule consequents were determined by weighing the output provided by each input data with the degree of activation of every rule. The number of membership functions was optimized by minimizing an index that determined when two of those membership should be joint into only one.

Some conclusions can be extracted when analyzing Table 3. Firstly, the number of rules automatically found by the algorithm proposed here is very often less than the number of rules found by the rest of algorithms, and this is achieved without imposing any restriction with respect to the size when the algorithm generates new individuals. Secondly, the FLSs created by the our algorithm perform better than the others. This is specially important taking into account that no local method is used to improve the results found by the EA. Finally, the method can be applied to any kind of function even when its exact shape is unknown, since it only needs a set of inputs–outputs pairs without considering if the function to be approximated is continuously defined.

5. Conclusions and future work

The algorithm presented here takes advantage of the fact that the object being evolved is itself the solution to the problem, not a representation. This has allowed us to design new operators that

include specific problem knowledge; for this very fact, every time an operator is applied the new individual it produces is always valid, and penalty functions and repair methods need not to be used.

The evolutionary algorithm performs well in the task of learning the shape of the function to be fitted (this is, the task of approximating the solution to the optimum one). This means that membership function centroids are well estimated and even trends in the rule consequents are detected, although exact values are more difficult to find. All this is done while keeping a small number of membership functions; thus, the fitness function used is easy to calculate, and good enough to ensure that good results are found without requiring any method that penalizes big FLSs.

The research presented in this paper will continue along the following lines:

- To use some kind of local searching in order to tune up the solutions found by the EA, getting more accurate values for consequents.
- To find some other way to compute the fitness in order to significantly reduce the time needed to run the algorithm.
- To test variations on current operators, especially in recombination, to allow the interchange of full rows/columns, and, in general, the interchange of cells from the first row and first column. This is difficult since every individual has a different size.
- To generalize the method so that any number of input variables can be used. This represents a serious challenge because a new representation and a different way to handle the individuals as well as new operators would be needed.

Acknowledgements

This work has been partially supported by the FEDER I+D project 1FD97-0439-TEL1, and by the CICYT projects TIC97-1149 and CICYT, TIC99-0550 (PUFO).

We would also thank the anonymous referees who reviewed this work. Their comments have helped to significantly improve the paper.

References

[1] V. Cherkassky, D. Gehring, F. Mulier, Comparison of adaptative methods for non-parametric regression analysis, IEEE Trans. Neural Networks 7 (4) (1996) 969–984.
[2] O. Cordón, M.J. del Jesús, F. Herrera, E. López, Selecting fuzzy rule-based classification systems with specific reasoning methods using genetic algorithms, in: Proc. Internat. Fuzzy Systems Assoc. (IFSA), Prague, Czech Republic, June 1997, pp. 424–429.
[3] L. Davis, The Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York, 1991.
[4] K.A. De Jong, An analysis of the behavior of a class of genetic adaptative systems, Dissertation Abstract International, 1975.
[5] L.J. Eshelman, J.D. Shaffer, Real-coded genetic algorithms and interval-schema, in: Foundation of Genetic Algorithms, Vol. 2, Morgan Kaufmann Publishers, Los, Atlos, CA, 1993, pp. 187–202.
[6] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, MA, USA, 1989.
[7] F. Herrera, M. Lozano, J.L. Verdegay, Genetic Algorithms and Soft Computing, Physica-Verlag, Heidelberg, 1996.
[8] A. Homaifar, E. McCormick, Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms, IEEE Trans. Fuzzy Systems 3 (2) (1995) 129–139.

[9] H. Ishibuchi, T. Murata, I.B. Türksen, A genetic-algorithm-based approach to the selection of linguistic classification rules, in: Proc. EUFIT'95, Aachen, Germany, 1995, pp. 1414–1419.
[10] C. Karr, Applying genetics to fuzzy logic, AI Expert 6 (1991) 26–33.
[11] C. Karr, E. Gentry, Fuzzy control of ph using genetic algorithms, IEEE Trans. Fuzzy Systems 1 (1993) 46–53.
[12] D. Kim, C. Kim, Forecasting time series with genetic fuzzy predictor ensemble, IEEE Trans. Fuzzy Systems 5 (4) (1997) 523–535.
[13] J.J. Merelo, J. Carpio, P.A. Castillo, V. Rivas, G. Romero, Evolving Objects, in: Proc. Internat. Workshop on Evolutionary Computation (IWEC'2000), State Key Laboratory of Software Engineering, Wuhan University, 2000, pp. 202–208.
[14] Z. Michalewicz, Genetic Algorithms+Data Structures = Evolution programs, 3rd Edition, Springer, New York, USA, 1999.
[15] H. Pomares, Nueva metodología para el diseño automático de sistemas difusos, Ph.D. Thesis, University of Granada, Spain, 1999.
[16] I. Rojas, J.J. Merelo, J.L. Bernier, A. Prieto, A new approach to fuzzy controller designing and coding via genetic algorithms, in: Proc. 6th IEEE Internat. Conf. on Fuzzy Systems, Vol. II, Barcelona, Spain, 1997, IEEE Service Center, NJ, USA, pp. 1505–1510.
[17] I. Rojas, H. Pomares, J. Ortega, A. Prieto, Self-organized fuzzy system generation from training examples, IEEE Trans. Fuzzy Systems 8 (1) (2000) 23–36.
[18] T.L. Seng, M. Bin Khalid, R. Yusof, Tuning of a neuro-fuzzy controller by genetic algorithms with and application to a coupled-tank liquid-level control system, Eng. Appl. Artif. Intell. 11 (1998) 517–529.
[19] T.L. Seng, M. Bin Khalid, R. Yusof, Tuning of a neuro-fuzzy controller by genetic algorithm, IEEE Trans. Systems, Man, Cybernet. 29 (2) (1999) 226–236.
[20] M. Setnes, H. Roubos, Ga-fuzzy modeling and classification: complexity and performance, IEEE Trans. Fuzzy Systems 8 (2000) 509–522.
[21] P. Siarry, F. Guely, A genetic algorithm for optimizing Takagi-Sugeno fuzzy rule bases, Fuzzy Sets and Systems 99 (1998) 37–47.
[22] A. Wright, Genetic algorithms for real parameter optimization, in: Foundation of Genetic Algorithms, Morgan Kaufmann Publishers, Los Altos, CA, 1991, pp. 205–218.
[23] L.A. Zadeh, Fuzzy sets, Inform. Control 8 (1965) 338–353.

5.3 FROM CLASSROOM TO MOBILE ROBOTS COMPETITION ARENA: ...

A continuación se detallan los datos del artículo publicado relacionado con esta sección de la disertación.

Título: **From Classroom to Mobile Robots Competition Arena: An Experience on Artificial Intelligence Teaching**
Revista: **The International journal of engineering education**, 2011; Factor de impacto: **0,418**;
Autores: **José Carpio Cañada**, T. J. Mateo Sanguino, S. Alcocer, A. Borrego, A. Isidro, A. Palanco, J.M. Rodríguez

Relevancia de la revista:

Nombre de la categoría	Revistas en la categoría	Posición en la categoría	Cuartil en la categoría
EDUCATION, SCIENTIFIC DISCIPLINES	33	24	Q4
ENGINEERING, MULTIDISCIPLINARY	90	60	Q3

IJEE 2469 PROOFS

International Journal of Engineering Education Vol. 27, No. 4, pp. 1–8, 2011
Printed in Great Britain

0949-149X/91 $3.00+0.00

Classroom to Mobile Robots Competition Arena: An Experience on Artificial Intelligence Teaching*

J. CARPIO CAÑADA,[1] T. J. MATEO SANGUINO,[2] S. ALCOCER,[1,2] A. BORREGO,[1,2] A. ISIDRO,[1,2] A. PALANCO[1,2] and J. M. RODRÍGUEZ[1,2]

[1] Dpto. de Tec. de la Información, Sistemas Informáticos y Automática, Universidad de Huelva, Ctra. Huelva-La Rábida s/n, 21819 Palos de la Frontera (Huelva), Spain. E-mail: jose.carpio@dti.uhu.es

[2] Dpto. de Ing. Electrónica, Sistemas Informáticos y Automática, Universidad de Huelva, Ctra. Huelva-La Rábida s/n, 21819 Palos de la Frontera (Huelva), Spain.

This paper presents an educational experience developed in the fourth year of Computer Engineering degree at Huelva University (Spain). To make Artificial Intelligent (AI) learning processes more captivating, a new educational project was incorporated into classical teaching of Artificial Intelligence and Knowledge Engineering subject. In this paper, we present the experience fulfilled with a group of college students. Here it is related how they changed for some days their classroom lessons for the robotic competition arena. With this project we have extended regular classroom lessons with additional work that could be useful and cannot be provided by traditional practical lessons, the real life experience. As a real example about how the work was accomplished we describe the mechanical construction of the mobile robots as well as the software development process.

Keywords: artificial intelligence; engineering education; gaming; robotic competition

1. Introduction

This project was a perfect example to make students gain some interest in robotics, and robotic competitions are a good framework to develop classroom experiences. Competitions also offer the students the opportunity of meeting more experienced people on this field. It also helps students to realize that frequently, real life is different from the problems the students solve at the university. It is completely different to design or program code for a computer simulation and to do it for a real mobile robot.

For example, there are many factors that have to be taken into consideration, like the mobile robot's battery charge or the light at the competition hall. In our case, it gave us some valuable pieces of advice; some of them about hardware design and others about how to design our software.

The competition game consisted in developing a mobile robot able to follow a line as fast as possible in a simple track. But, which is the meaning of these terms?

- 'A robot is a virtual or mechanical agent. In practice, it is usually an electromechanical system which, by its appearance or motion, conveys a sense realized on its own'.
- 'A robotic competition is an event where robots have to accomplish a given task'.

These two definitions—given by Wikipedia—describe the concepts of a robotic competition. A robotic contest is important for AI because two robots with no human help or guidance have to fulfill a given task faster or better than the rest of the competitors. In general terms, in a basic AI lab practice, it is enough if the robot accomplishes the task; no matter how long it takes to do it. In a robotic competition it is not enough; the behavior of the robots must also be changed and improved so it can beat the rest of the competitors.

There are many already developed robotic platforms as Pololu, e-puck [1] or Khepera [2] but we have developed the complete platform and control software for this hardware. So, students have developed new skills designing and building Printed Circuit Boards (PCBs), working with electronic components or designing embedded software. Almost all the technical teaching have the goal to form tomorrow's engineers and this experience brings class closer to real-life frameworks. For example, in Zhongli et al. [3] it is presented an Internet-based platform for a soccer competition devoted to robotics education. To facilitate the students' learning when participating in the robotics competition, the supporting hardware and software kits have also been developed. DeVault [4] describes an engineering experience through the implementation of a mobile robotics course and the participation in an annual robot contest. In Grimes et al. [5] the educational outcomes are described and student know-how to make of a competition an excellent opportunity for educational growth. In this paper is illustrated how the students have full responsibility for defining the competition rules, designing, constructing the course and carrying out the competition. In Berlier at al. [6] it is presented a meth-

* Accepted 15 October 2010.

Table 1. main features of some educational experiences on robotic competitions

Reference	Education Level	Kind of Competition	Programming Language	Field
[3]	Secondary	Robot soccer	Icon-based instructions	Robotics
[4]	University	Sumo wrestling	C	Engineering Physics
[5]	University	Ping-pong balls	C	Mechanical & Computer Engineering
[6]	University	Line following, maze-navigation & drawing	Assembly & C	Micro-computers
[7]	University	Robot soccer	C++	Robotics & Computer Vision
[8]	University & High School	Chessboard & robot soccer	GrafCet	Engineering & Computer Science
Onubot	University	Line following	Java	Artificial Intelligent

odology used to replace the final project with a robotics project where students build a microcontroller-based robot with the ultimate goal of competing. Murphy [7] describes a strategy for integrating robot design competitions into courses in order to maximize learning experience and promote intellectual development. Finally, in Almeida et al. [8] mobile robot competitions are presented as events well suited to experimentation, research and development in many areas concerning both High School and University.

The present paper is organized in the following way. The 'General Overview' section presents the 'OnuBot' teamwork and how the motivation to compete comes from the classroom to mobile robots competition arena. The following section 'Project Development' explains the task development and how the work was accomplished. The 'Competition' section briefly discusses the results of this game. The 'Experience in Teaching' section puts into practice the methodology developed by this teamwork so far, an evaluation questionnaire is presented with such purpose. Finally, this paper contributes with some conclusions to the work carried out; it provides new expectations and offers this experience to the engineering community at its social website.

In order to highlight the contributions of this teaching innovation project, different features and properties are compared to some of the aforementioned educational experiences (see Table 1).

2. General overview

We have used mobile robots during the last five years as a means of teaching main mobile agents aspects. In the first years students showed an excellent motivation. Nevertheless, this interest gradually decreased in the following years. These students were doing the fourth year of 'Computer Engineering' degree at Huelva University and this project was developed in the 'Artificial Intelligence and Knowledge Engineering' subject.

The experience started in the academic course 2008/2009 and was mostly carried out by students. This experience allowed us to obtain a teaching innovation project awarded by the University of Huelva. This project let the teachers set up two different students' teams and two different mobile agents—Mini-Z and Iwaver 01—with the goal of participating in a national robotic competition called 'Cosmobot 2009' (see Fig. 1). The competition was held by 24 teams from different national universities. The game consisted of developing two mobile robots able to follow a line as fast as possible in a simple closed track.

This collaborative experience is not only understood as a serial of practical sessions. Otherwise it comprises the possibility of discovering AI techniques with the aim of a robotic competition. In this project we have extended regular classroom lessons with additional work that would be useful and may not be provided by many traditional practical lessons. Such tasks—approximately 600 hours—consisted in working with electronic components, develop PCBs, designing embedded software, modelling the systems by using Unified Model Language (UML) diagrams or testing AI algorithms (see

Fig. 1. The Onubot teamwork in the competition arena during Cosmobot.

Table 2. List of tasks and hours spent on the project

Item	Hours	Persons	Task
1	2	7	Briefing & work organization
2	3	6	Working with robot: task planning, interfaces, programming & debugging
3	3.5	3	Working with robot: track design & reactive algorithm
4	4.5	5	Schematic & layout design with Eagle
5	4	1	Mapping & testing track algorithm
6	3	6	Printing board, develop & etching
7	4.5	2	Reactive algorithm & adjust parameters
8	3	2	Welding, drilling & assembling PCB
9	2	3	Reactive algorithm & optimization for straight lines
10	2	1	Working with robot: testing algorithm in circuit with predominance of curves
11	5	2	Testing with robot: creation of new circuits & parameter settings
12	6	1	Working with 2nd robot: assembly, testing & new servomotors
13	2	2	Research on CMUcam
14	5	4	UML design: tasks & processes to perform
15	3	1	UML specification: classes, attributes & methods
16	9	7	Code debugging and circuit preparation
17	12	4	Improving code
18	3	3	Algorithm advances with Iwaver & Mini-Z robotic vehicles
18	40	7	Cosmobot 2009 competition
20	10	2	Updating webpage
21	6.5	1	Microcontroller research
22	4.5	1	Documentation of the directed academic work

Table 2). So, working in group has also been important to accomplish the project.

3. Project development

3.1 Rules of the competition

The first task we had to do, once the working team had been formed, was to study the rules of the competition which had been previously published at the 'Cosmobot 2009' official website. This information included all the requirements to be fulfilled by the participants and their robots. To sum up, the most relevant aspects were the following:

- Every team should be formed by four members with one representative.
- Every team could have more than one mobile robot.
- The track was closed, white and delimited in both sides by black insulating tape lines of 2 cm wide. The whole track was 1550 cm wide. At the same time, there was a margin of 15 cm in each side delimited by a red insulating tape.
- The mobile robot should not ever reach those lines in its routes because it would be disqualified.
- The bends were made of pieces of circumference whose radius was always over 40 cm.

In addition, there were some possible eventualities contemplated as, for example, the fact that the track had little irregularities or that the lighting condition was not defined beforehand. Finally, it is important to point out the limitations imposed on the vehicles. They must have a maximum dimension of 20 cm wide $\times$ 30 cm long $\times$ 13 cm high, apart from being completely autonomous. It was completely forbidden the existence of any remote control element.

3.2 Brainstorm meetings

Once all the rules of the competition were known, we met in order to prepare the work to develop. On the one hand, the leader suggested the teamwork the strategy of buying two specific mobile robots which had a reduced size and low cost. On the other hand, we had to make our own PCBs where we had to insert the chosen microcontroller, sensors and other electronic devices. Once the PCBs had been made it would be joined to the chassis of the vehicles and the appropriate connections would be done.

The robotic vehicles—Mini-z and Iwaver 01—both had very similar qualities and their dimensions where approximately 12 cm long $\times$ 7 cm wide $\times$ 4 cm high (see Fig. 2). The chosen microcontroller was

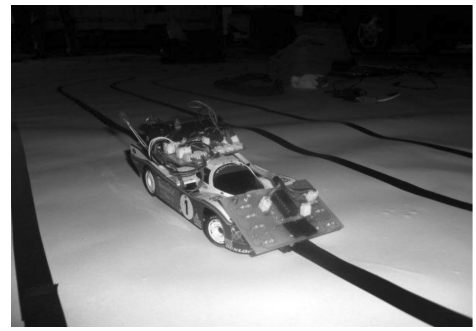


Fig. 2. Iwaver 01 robot of the Onubot teamwork in the competition arena during Cosmobot.

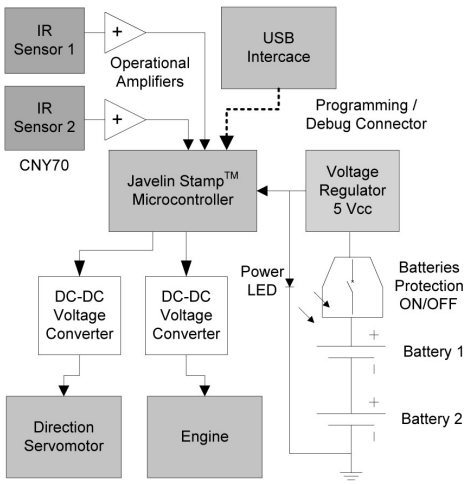


Fig. 3. Outline of the electronic aboard the robotic vehicles.

Javelin Stamp™ from Parallax enterprise (see Fig. 3). This election was due to the fact that we were already familiar with the use of this microcontroller in the practices of 'Artificial Intelligence and Knowledge Engineering' subject. In these practices we got in touch with the robots programming and with external elements interaction. During the first meeting of the teamwork we considered the possibility of using the Parallax version of CMUcam camera as a complement of the car. The rules allowed it and at the beginning, we thought that it would be useful to give a computer feedback vision to the vehicle. We guessed that the robotic vehicles could see the bend before arriving and react with more efficiency. Finally, this option was ruled out because of several reasons. First, the camera's weight decreased the speed and stability of the vehicle as well as its autonomy. Second, the processing capacity required by this camera and the microcontroller became inadequate.

Inspired by the mobile robots used in our practices, we decided to include two sensors in the central part at the front of the vehicle. We also established that the vehicle would go over the line, although it was allowed to cover the white track space. We left the possibility to add two extra sensors, one in each extreme of the front side.

3.3 PCB layout and assembling process

Once the working process became clear, we started to meet again during the following days, this time in order to make the PCB design by means of EAGLE Layout Editor©. The electronic circuit was already designed so we printed it in transparencies. Taking advantages of the University's facilities we could use a laboratory to make the PCBs. The process had the following steps:

- Designing and cutting the light photosensitive bakelite with the intended size.
- Sticking tightly, by means of sticky tape, the piece of the PCB and the printed sheet. After this, putting them into the isolation machine. This machine is like a photocopier with a special light and, in fact, it carries out a similar function applying an ultraviolet light to the whole PCB. The light intensity excites the PCB's surfaces except the drawn mask that will be the copper traces of the future circuit.
- When it became photosensitive, the board was carefully introduced into a photo-sensitive dissolution of a specific product to reveal it.
- Etching all the photosensitive PCB area leaving only the fiberglass base with the printed circuit intact.

The assembly of the hardware was the following task. Once we already had the PCBs, we only had to drill, sold and assemble the set of resistances, capacitors and undoubtedly the microcontroller.

3.4 Software specification and requirements

Due to the short time available to build and program the robots, it was necessary to distribute the different task between the people in the project, so we could finish on time. To achieve this goal we modelled the system using UML diagrams.

We decided to clarify the way to develop the desired work in a common and efficient way. In that sense, we had to carry out a detailed analysis of all the existing variables related to the competition. Accordingly, the variables were classified taking into account if they were related to the robotic vehicle or the track. These variables are shown in Table 3. It can be observed that light intensity affects the measurement of the sensors. On the other hand, when the battery charge got low, the speed and acceleration got down. Regarding the speed, we had to decide if put the cars at high speed or medium speed. With high speed, the cars sometimes went off

Table 3. Classification of variables related to the mobile robots and the track of the competition arena

Variables	Related with
Light intensity	Sensor measurements
Battery charge	Speed & acceleration
Maximum speed	Risk level
Turning angle	Motion in a curve or in a straight line
Number of sensors	Accuracy
Physical limits	Behavior of the robotic vehicle
Stretches position	Anticipation to the next event
Friction	Behavior of the robotic vehicle
Pothole	Disorientation

the road so we had to decide what level of risk we should accept. It was also important to know if the cars were in a curve or in a straight line because the turning angle to correct the car should be different. Taking into account our experience, the number of sensors was an important lack of our robotic vehicles since they only had two sensors. We discovered that it could not have enough accuracy; nevertheless, it was enough for our design. It was also very important to know other physical limitations like size of the wheels, position of axis direction, weight, stability and specially the hit resistance. According to the variables related to the track of the competition arena, stretches position may be the most important variable. We programmed the car with a map of the circuit so it could always be in the best position and anticipate the next event before it came. With this, we hoped the car braked before the curve and accelerated when the straight line was beginning. With relation to friction, it was important to know that the results obtained in our tests could differ significantly from the results obtained in the official circuit due to material construction. Another quite important aspect was the pothole. Both our test circuits and the official one contained undulations. These strains were mainly caused by the tension of the tape that formed the path. Our small robotic cars were negatively influenced by it.

We concluded that if we knew the features of the track, we could set a clear analysis and the mobile robot could react before. By means of that, we could do diagrams about the program structure in UML format. Besides, it would be necessary to program every section of the track in a single way, so that we could finally combine all of them. Thus, we realized that students required an additional effort to provide a high reliability to the design of mobile agents. Usually this aspect is not crucial for general purpose software developed on AI. That is, when something wrong happens the computer engineer usually shows 'windows alerts'. On the contrary, in a competition a fault means the end of the game for competitors; no faults or errors are allowed. Indeed, this aspect is difficult to learn outside of a real problem.

3.5 Firsts tests at the laboratory

The program was developed in Java language but avoiding the use of classes and minimizing the calls to external functions to reduce the computational cost (see Fig. 4). We realized that we based our strategy in a good software design, but our basic hardware did not let us make a working program that could satisfy the competition requirements. It was difficult to fulfill the task with our designed hardware because we designed a simple and fast system with only two light sensors with digital signal processing. When we tried to make a high speed control with only two sensors, we comprehended that environmental light conditions and circuit surface would cause wrong measurements. Thus, at high speeds these fails cause the mobile robots loose the line and go out from the circuit. Consequently, the team would be withdrawn from the competition.

In the following stage, we thought it would be a good idea to make a copy of the competition track, so we started working on that. One of the problems consisted on the material to support the track; getting a plastic sheet was too expensive. As a consequence, we used a great toll of paper as the base. To do this, we firstly added several sheets in order to get the desired dimensions. Secondly, by means of a tape, we composed the straight lines and the curves all the way long. We already had our track; thanks to it we could perform our first reliable test.

```
// */
public static void wait_start(){
  System.out.println("pass1");
  while(CPU.readPin(SWICTH) != ON){}
    System.out.println("pass2");
  while(CPU.readPin(SWICTH) == ON){}
    System.out.println("pass3");
}

  public static void main (){

  int MAXIMUM_TURN = GREAT_TURN[2];
  int MINIMUM_TURN = LITTLE_TURN[0];
  int MAXIMUM_SPEED = HIGH_SPEED[1];
  int MINIMUM_SPEED = LOW_SPEED[1];
  int direction=1; //1 right, -1 left
  int speed = MINIMUM_SPEED;
  int turn = 0;

//  wait_start();

CPU.writePin(LED1,true);
CPU.writePin(LED2,true);
  while(true){
    servo_mot.update(neutral_engine+speed, low_mot);
    servo_dir.update(neutral_direction + (turn * direction), low_dir);

    if (CPU.readPin(SID) == INSIDE && CPU.readPin(SDD) ==
    INSIDE){
      speed = HIGH_SPEED;
      turn = 0;

    }else if(CPU.readPin(SID) == INSIDE && CPU.readPin(SDD) ==
    OUTSIDE){
      turn = MINIMUM_TURN;
      direction = 1;

    }else if(CPU.readPin(SID) == OUTSIDE && CPU.readPin(SDD)
    == INSIDE){
      turn = MINIMUM_TURN;
      direction = -1;

    }else if(CPU.readPin(SID) == OUTSIDE && CPU.readPin(SDD)
    == OUTSIDE){
      turn = MAXIMUM_TURN;
      speed = MINIMUM_SPEED;
    }
  }
}
}
```

Fig. 4. Example of code implemented in the mobile robots.

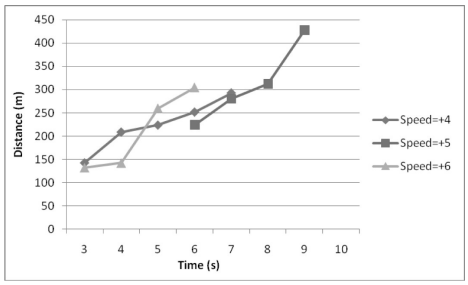


Fig. 5. Speed variations with different AI algorithms applied to the vehicles.

At that moment we could see the true necessity to improve the speed and the efficiency of the robotic vehicle. Moreover, we could face the problem of the distortions on the track beforehand. These were mainly caused by the tension produced, not only by the insulating tape but also by the paper fragility. This situation would be reflected on the true track. That is the reason why all participants had the same problems like us.

The software development continued but it did not live up to our expectation. We understood the necessity of having more sensors for a better accuracy measurement that would influence the behavior with more precision. In addition, we had difficulties with the batteries, so the battery overuse by the car's engines negatively influenced on acceleration and braking cycles. These and other reasons made us decide to load a more primitive algorithm into the robot only few days before the competition. This algorithm did not perform complicated intelligent analysis; it simply read the sensors, being the engine speeds and servomotors' direction constant. The result was not very fast but at least it did not come off the bends and allowed a higher working autonomy. In Fig. 5 it can be seen these tests realized with algorithms tuned at different speeds. The lines show distance travelled by the robotic vehicle (expressed in cm) versus time (represented in seconds).

4. Competition

We went to 'Cosmobot 2009' with the proof material including our track. We were surprised due to the fact that any participant did not carry anything similar and they only had little stretches over different surfaces to calibrate the sensors' sensibility. This made us reflect upon the greater importance of pure mechanics and electronics versus the computer analysis we mainly had done. All competitors tried out their vehicles in our track and we apprehended the invested effort had been worthy: most vehicles came off or made maneuvers that would make them fail. We could notice some similar models of robotic vehicles like our ones with very specialized features prepared for this competition. As we could check later these competitors kept on participating with all their vehicles.

The competition took place in a very short time and in a very good sporting atmosphere. Its dynamics was also specified in the rules. It would consist of a first validation lap in which all the robot vehicles had to prove their capacity to cover the track without coming off the time limit. Later, each vehicle performed individually a round of three tries. The goal was to complete at least two laps in the shortest possible time. Once all the vehicles attained the classification, approximately half of them went to the following step: the car chase. In this step, two vehicles placed in opposite points of the track started at the same time. The race finished when one of the two mobile robots reached the other and the winner was the one which had done it twice in at least two big tries.

We did not have any problem at passing the validation lap and the classification later but the quality of our opponents exceeded our robotic systems. Our teams took 41.87s to make 2 laps with an average speed of 0.54 m/s. The best competitor took 14.93 s to make the same laps with an average speed of 1.52 m/s, taking 102.60 s the worst runner with an average speed of 0.22 m/s. This made us finish in a middle position. When the competition finished the winners where congratulated and agreed to be interviewed by us. It was very interesting to share experiences with different competitors who kindly explained the bases of their designs. The experience has been quite pleasant and we consider that in this first attempt we have learnt a lot of things about working in group, especially when this work is orientated towards a competition task.

5. Experience in teaching

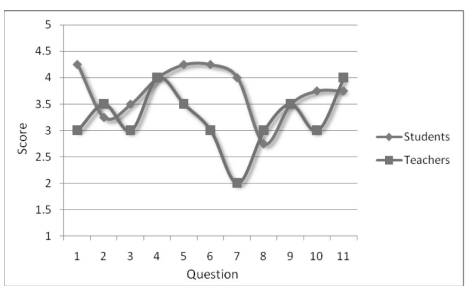
With the aim of addressing innovative teaching and learning methods related to this experience, we have evaluated the students' opinion and their implications for Engineering Education. A statistical study has been carried out on two teamworks of students and teachers during 2009 (see Table 4) with a score ranging between 1 (completely disagree) and 5 (completely agree). The questionnaire includes aspects referred to how the educational gaming has improved teaching-learning practices in university education. Two groups of users (6 students and 2 professionals) have been considered.

Questions 1 to 11 describe the level of knowledge acquired on several technical fields like electronics, sensors and microcontrollers as well as transverse knowledge like project management, software pro-

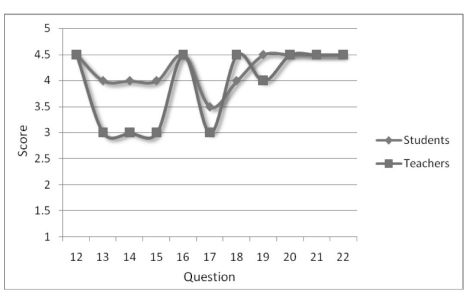
Table 4. Evaluation questionnaire of the experience in teaching

Question	Description	Teachers	Students	Deviation
1	Acquired knowledge on electronic	3	4.25	0.76
2	Acquired knowledge on project management	3.5	3.25	0.76
3	Acquired knowledge on software programming	3	3.5	0.28
4	Acquired knowledge on modelling system	4	4	0
5	Acquired knowledge on designing system	3.5	4.25	0.5
6	Acquired knowledge on developing system	3	4.25	0.76
7	Acquired knowledge on sensors and actuators	4	4	1.25
8	Acquired knowledge on computer languages	3	2.75	0.76
9	Acquired knowledge on microcontrollers	3.5	3.5	0.5
10	Acquired knowledge on 'Artificial Intelligent' subject	3	3.75	0.5
11	Acquired knowledge on testing phase	4	3.75	0.28
12	Acquired knowledge on working competition	4.5	4.5	0
13	Acquired knowledge on writing documentation	3	4	0.76
14	Acquired knowledge on working within a team	4.5	4	0.76
15	Project organization	3	4	0.76
16	Resources available	3	4.5	0
17	Similar known experiences	3	3.5	0.57
18	Other national robotic groups known	4.5	4	1.04
19	Motivation in the study of robotic agents	4	4.5	0.57
20	Motivation in the study of 'Artificial Intelligent' subject	4.5	4.5	0.5
21	Evaluation of the competition experience	4.5	4.5	0
22	Global evaluation of this teaching innovation project	4.5	4.5	0

gramming, and modelling, designing and developing of systems (see fig. 6a). In these items it can be observed that students' rates are higher with those questions related to hardware development (questions 1, 6 and 7) and lower with those ones related to software programming (questions 3, 6 and 8). The low rate in question 8 indicates that students already have a thorough knowledge of programming languages when they get to fourth course in Computer Engineering. On the other hand, question 10 proves that AI knowledge has been strengthened.



(a)



(b)

Fig. 6. Average score of students and teaching professionals.

Figure 6b shows how the students' opinion agrees with the opinion of teachers. Most of the questions favorably scored are related to the working competition and the social development of the person (question 12). Perhaps, the most important interpretation is that questions 19 and 20 stand out the motivation and its implications for both 'Artificial Intelligent' subject and robotic agents. In this sense, students and teachers highlight the success of the competition experience being rather high (questions 18, 21 and 22). It is still too early to obtain concluding results; although the developed analysis leads the professionals to conclude that with the educational experience we obtained several additional targets that helped students to develop their skills. In general terms, it has been a great learning experience, we have learned from the difficulties and even when the results were not the expected ones, all the acquired knowledge was perfectly worthy.

6. Conclusions

Student motivation is an essential issue in a learning process and it implies a challenge for educators who want to preserve and increase this educational aspect. We have used mobile robots during the last five years as a means of teaching mainly mobile agents aspects. In the first years students showed excellent motivation. Nevertheless, this interest has gradually decreased in the subsequent years. In order to make more captivating the AI learning

process, we decided to incorporate a new practical session. That session comprises the possibility to share discovered AI techniques in a robotic competition. In this paper, we present the experience fulfilled with a group of students at university who changed for some days their classroom lessons for the robotic competition arena. We describe the mobiles robots construction and software development process including modelling, design, and implementation and testing phases.

We conclude that participating in robotics competitions gives university students a broader vision of AI subject, extra motivation and the possibility to share knowledge with more experienced students belonging to other universities. In order to evaluate the teamwork's activities, we present a pedagogical survey that emphasizes the importance of working in group with a real robot designed for a competition. Our experience will always be a good start point to develop a similar project with future promotions. Furthermore, the possitive experience has brought up so much expectatives that teachers and students have confirmed to compete this year again. Besides, the students evidence the intention to accomplish future works in AI mobile agents' domain. Videos, links and further information about this work are available at the Onubot teamwork's site: www.facebook.com/pages/Onubot/103904463217.

Acknowledgements—We are grateful to the Teaching Innovation Service of the University of Huelva that promoted this project (PIE084/2009). We would also like to thank the Department of Electronic Engineering, Computer Systems and Automatics for providing its laboratory for manufacturing the PCBs.

References

1. F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J. Zufferey, D. Floreano, A. Martinoli and P. Gonçalves, The e-puck, a Robot Designed for Education in Engineering, *Proc. 9th Conference on Autonomous Robot Systems and Competitions*, **1**(1), 2009, pp. 59–65.
2. F. Mondada, E. Franzi and P. Ienne, Mobile robot miniaturization: A tool for investigation in control algorithms, *Proc. Third International Symposium on Simulation on Experimental Robotics (ISER-93)*, **200**, (1993) pp. 501–513.
3. Zhongli Wang; Yafang Liu; Dongxiao Wang; Qingyun Li; Tai Chen; Liu, Y. H.; Yunde Jia, Internet Based Robot Competition and Education, *IEEE International Conf. Robotics and Biomimetics (ROBIO 2007)*, (2007) pp. 285–290.
4. J. E. DeVault, A competition-motivated, interdisciplinary design experience, *Proc. Frontiers Educ. Conf. (1998 FIE)*, (1998) pp. 460– 465.
5. J. Grimes and J. Seng, Robotics Competition: Providing Structure, Flexibility, and an Extensive Learning Experience, *38th ASEE/IEEE Frontiers in Education Conference*, (2008) pp. 1–5.
6. J. A. Berlier and J. M. McCollum, The robot competition: A recipe for success in undergraduate microcomputers courses, *IEEE International Conf. Microelectronic Systems Education (MSE '09),* (2009) pp. 126—129.
7. R. R. Murphy, A Strategy for Integrating Robot Design Competitions into Courses in Order to Maximize Learning Experience and Promote Intellectual Development, *IEEE Robotics & Automation Magazine*, **8**(2), 2001, pp. 44–45.
8. L. Almeida, J. Azevedo, C. Cardeira, P. Costa, P. Fonseca, P. Lima, F. Ribeiro and V. Santos, Mobile Robot Competitions: Fostering Advances in Research, Development and Education in Robotics, *CONTROLO'2000 the 4th Portuguese Conf. Automatic Control*, (2000) pp. 592–597.

José Carpio Cañada is Computer Engineer. Since 2004, he has worked as a full-time Associate Teacher in the Department of Information Technology at the University of Huelva (Spain). His research lines are focused on robotics and artificial intelligence.

Tomás de J. Mateo Sanguino is Electronic Engineer, Master in University Teaching and Doctor. From 1998 to 2004, he was granted a scholarship at the National Institute of Aerospace Technology (INTA) and worked as a hired engineer at the Spanish National Research Council (CSIC). Since 2004, he has worked as a full-time Associate Teacher in the Department of Electronic Engineering, Computer Systems and Automatics at the University of Huelva (Spain). Besides, he currently works as an instructor in the CCNA program at the CISCO Networking Academy. His research lines are focused on robotics and engineering education.

S. Alcocer Vázquez, A. Borrego Delgado, A. Isidro de la Cruz, A. Palanco Salguero and J.M. Rodríguez González are students of Computer Engineering at University of Huelva (E.P.S. La Rábida).

5.4 EVOLVING THE STRATEGIES OF AGENTS FOR THE ANTS GAME

A continuación se detallan los datos del artículo publicado relacionado con esta sección de la disertación.

Título: **Evolving the Strategies of Agents for the ANTS Game**
Congreso: **International Work-Conference on Artificial Neural Networks (IWANN)**, 2013
Autores: **José Carpio**, Pablo García-Sánchez, Antonio Miguel Mora, Juan Julián Merelo Guervós, Jesús Caraballo, Fermín Vaz, Carlos Cotta

Relevancia del congreso atendiendo al "Computer Science Conference Ranking"

Nombre del Área	Posición en la categoría
Artificial Intelligence and Related Subjects	Rank3

Evolving the Strategies of Agents for the ANTS Game

José Carpio[1], Pablo García-Sánchez[2], Antonio M. Mora[2], Juan Julián Merelo[2], Jesús Caraballo[1], Fermín Vaz[1], and Carlos Cotta[3]

[1] Dept. of Computer Science, University of Huelva, Spain
[2] Dept. of Computer Architecture and Technology, University of Granada, Spain
[3] Dept. of Computer Languages and Computer Sciences, Málaga, Spain
jose.carpio@dti.uhu.es

Abstract. This work studies the performance and the results of the application of Evolutionary Algorithms (EAs) for evolving the decision engine of a program, called in this context *agent*, which controls the player's behaviour in an real-time strategy game (RTS). This game was chosen for the Google Artificial Intelligence Challenge in 2011, and simulates battles between teams of ants in different types of maps or mazes. According to the championship rules the agents cannot save information from one game to the next, which makes impossible to implement an EA 'inside' the agent, i.e. on game time (or on-line), that is why in this paper we have evolved this engine off-line by means of an EA, used for tuning a set of constants, weights and probabilities which direct the rules. This evolved agent has fought against other successful bots which finished in higher positions in the competition final rank. The results show that, although the best agents are difficult to beat, our simple agent tuned with an EA can outperform agents which have finished 1000 positions above the untrained version.

1 Introduction

Real-Time Strategy (RTS) games are a sub-genre of strategy-based videogames in which the contenders control a set of units and structures that are distributed in a playing arena. The game objective is normally eliminating all the enemy units. It is usually possible to create additional units and structures during the course of the game, at a cost in resources. Another usual feature is their real time nature, so the player is not required to wait for the results of other players' moves as in turn-based games. Starcraft™, Warcraft™ and Age of Empires™ are some examples of RTS games.

The 2011 edition of the Google AI Challenge [5] was conducted with an RTS game named ANTS, in which the players control a set of ants that must 'fight' against the colonies of the rest of players in a grid with labyrinthine paths. The ants must gather food for generating new individuals and get an advance over the rivals. The fighting between ants is solved following some rules, but as a thumb rule, the higher number of ants are grouped, the easier will be to win a fight.

I. Rojas, G. Joya, and J. Cabestany (Eds.): IWANN 2013, Part II, LNCS 7903, pp. 324–333, 2013.

Thus, this is a RTS where the AI must be implemented at both commented levels: on the one hand, the ants must be grouped and specialized (explorers, fighters, gatherers), on the other hand each individual should have a particular behaviour to get a global emergent behaviour.

As a first approximation, a behavioural engine (for both levels) was designed by defining a set of states and rules guided by several parameters. This agent participated in the contest and finished in position 2076.

Then the initial engine has been improved by means of a Evolutionary Algorithms (EAs)[2]. They are a class of probabilistic search and optimisation algorithms inspired in darwinistic evolution theory. There are some types, including the extended Genetic Algorithms (GAs)[4], but the main features are common to all of them: a population of possible solutions (individuals) of the target problem, a selection method that favours better solutions and a set of evolutionary operators that act upon the selected solutions. After an initial population is created (usually randomly), the selection mechanism and the operators (crossover, mutation, etc) are successively applied to the individuals in order to create new populations that replace the older one. The candidates compete using their fitness (quality of adaptation). This process guarantees that the average quality of the individuals tends to increase with the number of generations. Eventually, depending on the type of problem and on the efficiency of the EA, the optimal solution may be found.

To conduct the evolution (in the evaluation step), every candidate agent in the population has fought against three different enemies (in two different approaches): a deterministic agent who finished in rank 993, and two very competitive agents which got position 1 and 165.

According to the results the agent has performed quite good, and has been able to beat bots which finished almost 1000 positions better than it in the competition.

2 State of the Art

AI in games has become the most interesting element in actual games from the player's point of view, once the technical components (graphics and sound) have reached almost an upper bound. They mostly request opponents exhibiting intelligent behaviour, or just better human-like behaviours [9].

Researchers have also found it an interesting area from the early nineties, so this scope has presented an exponential grown in several videogames and fields, mainly starting with the improvement of FPS Bot's AI, the most prolific type of game [8,12], and following with several games such as Super Mario [19], Pac-Man [10] or Car Racing Games [14], to cite a few.

The RTS games research area presents an emergent component [18] as a consequence of the commented two level AIs (units and global controllers). RTS games usually correspond to vast search spaces that traditional artificial intelligence techniques fail to play at a human level. As a mean to address it, authors in [15] proposed to extract behavioural knowledge from expert demonstrations which could be used to achieve specific goals. There are many research

problems involving the AI for RTSs, including: planning with uncertainty or incomplete information, learning, opponent modelling, or spatial and temporal reasoning [1].

However, most of the RTS games in industry are basically controlled by a fixed script (i.e. a pre-established behaviour independent of inputs) that has been previously programmed, so they are predictable for the player some combats later. Falke et al. [3] tried to improve the user's gaming experience by means of a learning classifier system that can provide dynamically-changing strategies that respond to the user's strategies.

Evolutionary Algorithms (EAs), have been widely used in this field [16,7], but they are not frequently used on-line (in real-time) due to the high computational cost they require. In fact, the most successful proposals for using EAs in games corresponds to off-line applications [17], that is, the EA works previously the game is executed (played), and the results or improvements can be used later during the real-time game. Through off-line evolutionary learning, the quality of bots' intelligence in commercial games can be improved, and this has been proven to be more effective than opponent-based scripts. For instance, in [11] an agent trained with an EA to play in the previous Google AI Challenge is presented.

In the present work, EAs are also used, and an off-line Genetic Algorithm (GA) is applied to improve a parametrised behaviour model (set of rules), inside a RTS named ANTS.

3 The Google AI Challenge

This section describes the game scenario where the bots will play. The ANTS game was used as base for the Google AI Challenge 2011 (GAIC)[1] [6]. An ANTS match takes place on a map (see Figure 1) that contains several anthills. The game involves managing the ant community in order to attack (and destroy) the maximum number of enemy hills. Initially, game players have one or more hills and each hill releases the first ant. Then, the bot has to control it in order to reach food and generate another ant. Game is based on a turn system (1000 turns in official games). For each turn, participants have a limited time to develop a strategy with the ant community, i.e. decide the set of simple steps (just one cell in one direction) that every ants must perform. Before turn time-over, the bot should return a witness indicating that tasks have been finished. If the witness is not sent before time-over, the player receives the 'timeout' signal. This signal carries penalty points and the inability to make more movements until game finish. However, this does not entail game disqualification.

If the player has accumulated enough points before 'timeout', she could win. For each captured hill, the player receives two points and if one of our hills is captured, she misses a point.

There are two strong constraints (set by the competition rules) which determine the possible methods to apply to design a bot: a simulated turn takes *just*

[1] `http://ants.aichallenge.org/`

one second, and the bot is *not allowed to store any kind of information between games* about its former actions, about the opponent's actions or about the state of the game (i.e., the game's map).

Thus, if desired, it is mandatory to perform an off-line (not during the match) fine-tuning or adaptation in order to improve an agent's behaviour. In this work, an evolutionary algorithm has been applied. Therefore, the goal in this paper is to design a bot/agent and improve it using an extra GA layer that consider a set of representative maps and enemies to train and adapt the bot for being more competitive, in order to fight the enemy, conquer its anthills, and finally win the game.

4 Algorithm and Experimental Setup

In this section the strategy to evolve is presented. A Genetic Algorithm (GA) is used to improve parameters of a basic agent. In order to improve the agent two different type of fitness functions and six different maps have been used.

4.1 Behavioural Rules and Parameters

The basic behaviour of our bot is mainly based in a Greedy strategy to prioritize multiple tasks entrusted to the ants:

```
IF enemy hill in sight
   attack the hill
ELSE IF food in sight
        pick up the food
      ELSE IF enemy ants in sight
             attack the ants
           ELSE IF non-explored zone in sight
                 explore the area randomly
```

The second part of the strategy, is a *lefty movement*, i.e. follow a straight line until water/obstacle is found, and then, walks to the left bordering it.

In order to perform a parameter optimization using genetic algorithms, we have defined a set in the above specified bot's rules. They are:

- *food_distance*: Maximum distance to go get food, i.e. ants ignores food that is at a distance greater than this value.
- *time_remaining*: Margin time we have for one turn to finish without a 'time-out penalty'. Higher values indicate that more actions are performed, but as previously explained, the player receives a penalty.
- *distance_my_ant_attack* and *distance_hill_attack*: These parameters are used to determine the attack priority. Distance_my_ant_attack means that we have one ant partner close enough to take advantage when attacking enemy ants. In this situation, the distance_hill_attack is taking into account in order to change ant objective. If another enemy ant is close to our hill, our ant give priority to this more dangerous situation for our interest. In this case an ant is sacrificed to keep alive our anthill.

– *turns_lefty*: Maximum number of consecutive turns in which an ant lefty strategy can be used. After that number of turns, ants community change to Greedy strategy.

4.2 Genetic Algorithm

A GA has been used to evolve the previously presented parameters. Thus each individual in the population is represented by an array of integers, where each number indicates the value of one of the parameters previously explained.

The *fitness function*, which determines the individual's adaptation to the environment, is based on launching a game against several opponents, in a certain number of turns and a specific map. The score for the agent after that game will determine the degree of kindness and individual adaptation to the problem we want to solve, knowing the individual that maximizes the score. Two different fitness functions have been studied:

– Basic fitness: it only considers the score obtained by our agent in the battle.
– Hierarchical fitness: the fitness is a tuple of the following elements in order: My score, enemy's score (negative), number of my own ants and number of enemy's ants (negative). A lexicographical order is applied to compare two individuals.

The considered operators have been:

– *selection*: choose half of population with individuals who obtained the highest scores in the games for improving the convergence component.
– *crossover*: multi-point crossover has been performed, mixing some parts of the parents to create the offspring.
– *mutation*: changes parameter values in an individual randomly (inside a range) with certain probability.

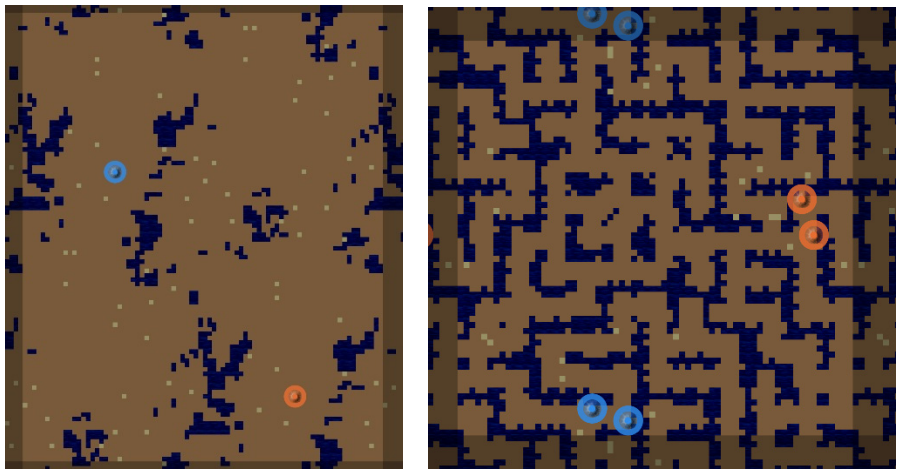
In order to achieve evolution it has been added an extra layer to the game implementation that allows us to store best individuals (set of parameters), and let to evolve the population in future generations.

4.3 Experimental Setup

Six maps have been considered in order to perform the bot evolution. All of them are provided by the competition organizers in a tools package. Three maps are mazes with different level of difficulty and the rest are open walking areas. Figure 1 shows two examples of different type of maps. The circles mark hills positions with one colour for each team/player. The blue areas represent water that ants cannot cross, nor walk on it, small points represent food and the rest are land where ants can move. Some other relevant information about maps is detailed in Table 1.

Table 1. Maps

	Name	Type	#competitors	Rows	Cols	#Hills
map1	random_walk_p02_01	Open	2	100	80	1
map2	random_walk_p02_05	Open	2	52	70	1
map3	maze_p02_05	Maze	2	66	66	2
map4	maze_p02_34	Maze	2	108	138	1
map5	maze_p02_42	Maze	2	72	126	2
map6	cell_maze_p02_10	Open maze	2	42	142	2



(a) Map 1: open map (b) Map 3: maze/labyrinth type

Fig. 1. Two different example maps considered in the experiments

The experiments conducted try to analyze the performance of the implemented approaches (GA + fitness function) in each of the six maps. Both have considered 64 individuals in the population, a crossover rate equal to 0.3, a mutation rate of 0.1 and a stop criterion set to 20 generations. Every agent is evolved in the six maps 10 times in order to get a reliable fitness value; i.e trying to avoid the 'noisy nature' [13] of game playing as a valuation function for an individual when the opponent is non-deterministic. The reason is the same agent (individual) could be valued as very good or very bad depending on the combat result, which in turn depends on the enemy's actions and the game events.

5 Results and Analysis

Firstly it is important to notice that all the selected competitors which have been considered as opponents in the evolution got higher final rankings than our bot, who finished in rank 2076. They are a deterministic agent who finished in rank 993, and two very competitive and non-deterministic agents which got position

Table 2. Results of ten battles between the evolved bot (using two fitness functions) and three different opponents with higher final ranks in the Google AI 2011 Competition. The scores, number of own and enemy's ants and the average number of turns to finish the match are presented, along with the standard deviation in each case.

	maxScore	**maxMyAnts**	**maxEnemyAnts**	**meanTurns**
Basic fitness vs. Bot993.				
map1	3,00 ± 0,00	84,08 ± 43,82	66,50 ± 49,20	416,87 ± 125,94
map2	3,00 ± 0,00	68,08 ± 39,10	60,67 ± 34,92	425,64 ± 90,26
map3	6,00 ± 0,00	39,91 ± 15,65	186,91 ± 63,74	318,28 ± 124,63
map4	1,00 ± 0,00	8,64 ± 0,67	12,00 ± 0,00	150,00 ± 0,00
map5	5,42 ± 0,51	36,00 ± 27,24	228,75 ± 89,91	428,93 ± 189,53
map6	6,00 ± 0,00	46,25 ± 59,21	111,25 ± 20,82	221,18 ± 104,35
Hierarchical fitness vs. Bot993.				
map1	3,00 ± 0,00	154,56 ± 28,84	2,67 ± 1,50	481,33 ± 48,26
map2	3,00 ± 0,00	97,67 ± 37,83	3,00 ± 2,18	486,78 ± 79,99
map3	6,00 ± 0,00	45,00 ± 8,85	118,33 ± 19,49	266,00 ± 55,57
map4	1,00 ± 0,00	9,22 ± 0,44	12,00 ± 0,00	150,00 ± 0,00
map5	4,67 ± 0,50	73,78 ± 71,10	226,89 ± 57,61	706,78 ± 262,88
map6	5,00 ± 1,15	104,11 ± 107,52	77,89 ± 58,10	519,44 ± 262,51
Hierarchical fitness vs. Bot165.				
map1	0,00 ± 0,00	33,58 ± 2,97	101,17 ± 7,83	183,42 ± 7,29
map2	0,17 ± 0,39	31,08 ± 8,54	122,00 ± 49,41	221,17 ± 86,06
map3	0,00 ± 0,00	35,33 ± 9,72	98,83 ± 10,99	186,50 ± 9,26
map4	0,00 ± 0,00	34,75 ± 9,75	99,17 ± 10,96	184,92 ± 9,11
map5	0,00 ± 0,00	32,50 ± 10,51	101,75 ± 12,19	186,25 ± 9,18
map6	0,00 ± 0,00	31,50 ± 10,91	103,10 ± 12,80	188,00 ± 9,08
Hierarchical fitness vs. Bot1.				
map1	0,00 ± 0,00	31,00 ± 34,00	109,00 ± 95,00	185,00 ± 198,00
map2	0,00 ± 0,00	17,00 ± 23,00	119,00 ± 132,00	156,00 ± 175,00
map3	0,00 ± 0,00	16,00 ± 17,00	118,00 ± 147,00	160,00 ± 186,00
map4	0,00 ± 0,00	14,00 ± 17,00	130,00 ± 120,00	166,00 ± 160,00
map5	0,00 ± 0,00	20,00 ± 31,00	112,00 ± 108,00	149,00 ± 147,00
map6	0,00 ± 0,00	21,00 ± 19,00	127,00 ± 131,00	172,00 ± 171,00

165 and the winner of the competition. Table 2 shows the obtained results in ten combats performed once the evolution has been completed.

It could be noticed the small standard deviation present in most of the results, due to the small variations in the combat scores. It is zero in many cases because there are very few possible values (i.e. in maps with only two hills, max_score will be 0, 1 or 3 points). In addition, when a bot is good, it wins most of times and the other way round. Thus in the evolutionary process after 20 generations the system evolves always to reach max score.

For the same reason it can be seen in the table that our bot can not beat those in positions 165 and 1, since they are much more sophisticated in its defined behavioural engine. However, the evolution of the agent gets higher number of own ants and decreases the number of enemy ants.

Moreover, our evolved agent wins on all maps to the robot that ended in ranking 993, more than 1000 positions above the initial version (without optimization). The number of ants is the main difference between basic fitness and hierarchical fitness, and this feature allows to use more effective attack techniques. In maps 5 and 6, the score is lower than the obtained with basic fitness in some cases. However, the number of own ants doubles those obtained with a basic fitness. This invites us to improve strategies in such type of maps to achieve a better use of the large community of generated ants.

6 Conclusions and Future Work

This paper presents the design of an agent (bot) that plays in the RTS ANTS game proposed for the Google AI Challenge 2011. Starting with a combination of two basic behaviours (Lefty and Greedy) and a set of parameters, an Evolutionary Algorithm (EA) is used to fine-tune them and thus modify the agent's behaviour.

This bot is evolved in six maps provided by Google, and fighting three different bots that participated in the contest: those who finished in positions 993, 165 and the winner. Two different fitness functions have been tested: a basic function that only takes into account the final score (the number of conquered anthills in a run), and a hierarchical fitness, where the number of player's ants, turns, and enemy ants are also used to compare individuals.

Results show that, even evolving the parameters of two simple strategies, the agent is capable to win harder opponents. On the other hand, the same strategy is not affective against a medium-ranked bot, so it is clear that the enemy behaviour affects to the off-line training algorithms with an specific strategy. However genetic optimization is enough to beat a competitor who is above more than 1000 positions in the ranking.

We conclude that parameters optimization using EA significantly improves agent performance in RTS games and this technique would obtain better results combined with good planning strategies.

For future work, new combination of strategies will be studied and more different fitness funtions will be analysed: for example, combining all maps in each fitness calculation. Because the stochastic behaviour of some robots also affects the fitness, an study of how this fitness is affected during the algorithm run will be performed. As demonstrated, the behaviour of the enemies is also a very important key to analyse for designing a all-terrain bot: an agent should adapt to these different behaviours. Also, using a quick map analysis in each turn to set the parameters obtained in this work could be studied to adapt the agent accordingly. A map analysis could be performed, for example, counting the number of direction changes in a period of time. If many direction changes occurs by collisions with walls, means that bots are fighting in a map with maze pattern. Once map type has been detected, bot can choose suitable parameter group for the map. The combination of the Greedy and Lefty actions also will be studied in other RTS games, as the previous Google AI Contest games.

Acknowledgements. This work has been funded in part by projects P08-TIC-03903 (Andalusian Regional Government), and TIN2011-28627-C04-02 (Spanish Ministry of Science and Innovation), and by the FPU Grant 2009-2942.

References

1. Buro, M.: Call for AI research in RTS games. In: Proc. AAAI workshop on Challenges in Game AI, pp. 139–141 (2004)
2. Eiben, A., Smith, J.E.: What is an evolutionary algorithm? In: Rozenberg, G. (ed.) Introduction to Evolutionary Computing, pp. 15–35. Addison Wesley (2005)
3. Falke-II, W., Ross, P.: Dynamic strategies in a real-time strategy game. In: Cantú-Paz, E., et al. (eds.) GECCO 2003. LNCS, vol. 2724, pp. 1920–1921. Springer, Heidelberg (2003)
4. Goldberg, D.E.: Genetic Algorithms in search, optimization and machine learning. Addison Wesley (1989)
5. Google. Google AI Challenge 2011: ANTS (2011), `http://aichallenge.org/`
6. Holdum, K.H., Kaysø-Rørdam, C., Østergaard, C.: Google ai challenge 2011: Ants. Jørgen Villadsen, 11 (2011)
7. Jang, S.H., Yoon, J.W., Cho, S.B.: Optimal strategy selection of non-player character on real time strategy game using a speciated evolutionary algorithm. In: IEEE Symposium on Computational Intelligence and Games, CIG 2009, pp. 75–79. IEEE (2009)
8. Laird, J.E.: Using a computer game to develop advanced ai. Computer 7(34), 70–75 (2001)
9. Lidén, L.: Artificial stupidity: The art of intentional mistakes. AI Game Programming Wisdom 2, 41–48 (2004)
10. Martín, E., Martínez, M., Recio, G., Saez, Y.: Pac-mAnt: Optimization based on ant colonies applied to developing an agent for ms. pac-man. In: 2010 IEEE Conference on Computational Intelligence and Games, CIG 2010, pp. 458–464 (2010)
11. Mora, A.M., Fernández-Ares, A., Merelo-Guervós, J.-J., García-Sánchez, P.: Dealing with noisy fitness in the design of a RTS game bot. In: Di Chio, C., et al. (eds.) EvoApplications 2012. LNCS, vol. 7248, pp. 234–244. Springer, Heidelberg (2012)
12. Mora, A.M., Moreno, M.A., Merelo, J.J., Castillo, P.A., García-Arenas, M.I., Laredo, J.L.J.: Evolving the cooperative behaviour in Unreal^{TM} bots. In: Proc. 2010 IEEE Conference on Computational Intelligence and Games, CIG 2010, pp. 241–248 (2010)
13. Mora, A.M., Fernández-Ares, A., Merelo Guervós, J.J., García-Sánchez, P., Fernandes, C.M.: Effect of noisy fitness in real-time strategy games player behaviour optimisation using evolutionary algorithms. J. Comput. Sci. Technol. 27(5), 1007–1023 (2012)
14. Onieva, E., Pelta, D.A., Alonso, J., Milans, V., Prez, J.: A modular parametric architecture for the torcs racing engine. In: Proc. 2009 IEEE Symposium on Computational Intelligence and Games, CIG 2009, pp. 256–262 (2009)
15. Ontañón, S., Mishra, K., Sugandh, N., Ram, A.: Case-based planning and execution for real-time strategy games. In: Weber, R.O., Richter, M.M. (eds.) ICCBR 2007. LNCS (LNAI), vol. 4626, pp. 164–178. Springer, Heidelberg (2007)

16. Ponsen, M., Munoz-Avila, H., Spronck, P., Aha, D.W.: Automatically generating game tactics through evolutionary learning. AI Magazine 27(3), 75–84 (2006)
17. Spronck, P., Sprinkhuizen-Kuyper, I., Postma, E.: Improving opponent intelligence through offline evolutionary learning. International Journal of Intelligent Games & Simulation 2(1), 20–27 (2003)
18. Sweetser, P.: Emergence in games. Game Development (2008)
19. Togelius, J., Karakovskiy, S., Koutnik, J., Schmidhuber, J.: Super mario evolution. In: IEEE Symposium on Computational Intelligence and Games, CIG 2009, pp. 156–161 (2009)

5.5 OPEN CLASSROOM: ENHANCING STUDENT ACHIEVEMENT ON ARTIFICIAL ...

A continuación se detallan los datos del artículo publicado relacionado con esta sección de la disertación.

Título: **Open classroom: enhancing student achievement on artificial intelligence through an international online competition**
Revista: **Fuzzy Sets and Systems**, 2003; Factor de impacto: **1,023**;
Autores: J. Carpio Cañada, T.J. Mateo Sanguino, J.J. Merelo Guervós, V.M. Rivas Santos

Relevancia de la revista:

Nombre de la categoría	Revistas en la categoría	Posición en la categoría	Cuartil en la categoría
EDUCATION & EDUCATIONAL RESEARCH	219	61	Q2

Original article

doi: 10.1111/jcal.12075

Journal of Computer Assisted Learning

Open classroom: enhancing student achievement on artificial intelligence through an international online competition

J. Carpio Cañada,* T.J. Mateo Sanguino,† J.J. Merelo Guervós‡ & V.M. Rivas Santos§

*Department of Information Technologies, University of Huelva, Spain
†Department of Electronic Engineering, Computer Systems and Automatics, University of Huelva, Spain
‡Department of Computer Architecture and Technologies, University of Granada, Spain
§Department of Computer Science, University of Jaén, Spain

Abstract Limitations of formal learning (e.g., one-way communication, rigid methodology, results-oriented approach) can significantly influence the motivation and expectation of students, thus resulting in an academic progress reduction. In order to make learning processes more playful and motivating, this paper presents a new educational experience developed by two groups of Computer Science students at the University of Huelva (Spain). As a result, an authentic real experience was incorporated into the classical teaching of Artificial Intelligence courses where classroom sessions were changed during some days for an international online competition. A comprehensive study considering the competition ranking, the students' opinion and their academic progress was analysed to assess the followed methodology. We found out that the educational experience improved the students' motivation, thereby enhancing their academic performance and personal skills as a result of learning through play. Moreover, additional teaching goals (e.g., learning new programming languages or increasing exam attendance) were obtained because of the positive motivation experienced by the competition. As a conclusion, this paradigm of real-life experience – not otherwise provided by traditional practical lessons – allowed us to ascertain that the process is more important than the outcome, which could be adapted to different teaching scenarios within an institution.

Keywords artificial intelligence, engineering education, online competition, teaching innovation.

Introduction

Traditional classrooms often involve several drawbacks due to limitations of formal learning (Novosadova *et al.*, 2007). Among others, one-way communication fails to encourage the students' proactive participation, additional effort is required by teachers to become aware of student's understanding of problems. Failures are mostly ascribed to learners by a punitive methodology, and rigid timing only adapts to students considering no individual skills or abilities (Dib, 1988). This results in a decrease in students' motivation and interest to study, which is compounded in engineering education (Van Kollenburg & van Schenk Brill, 2009). In effect, practical learning is specially required to be applied by a flexible knowledge rather than a conventional one. In this regard, interactive methods (e.g., problem-solving sessions, computer-based practices,

Accepted: 18 May 2014
Correspondence: Tomás de J. Mateo Sanguino, Universidad de Huelva (ETSI), Dpto. de Ing. Electrónica, Sistemas Informáticos y Automática, Ctra. Huelva-La Rábida S/N, 21819 Palos de la Frontera (Huelva), Spain. Email: tomas.mateo@diesia.uhu.es

gaming) allow teachers to engage students when they are actively working with educational resources (Adams, Hill, & Slater, 2000). Bearing this in mind, learning through play comes to the teaching scene as one of the most successful learning experiences (Veganzones *et al.*, 2011).

To make learning of Artificial Intelligence (AI) courses more captivating, an educational project based on learning through play has been conducted throughout the classical teaching of a Computer Science degree. In the past, we successfully used intelligent agents and mobile robots as a means of attractively teaching an AI course (Carpio Cañada *et al.*, 2011). In that pilot project, a group of college students changed their classroom lessons for the robotic competition arena for a few days. From the analysis, we found that participating in a robotic competition gave students a broader vision of AI concepts, extra motivation, and the possibility to share knowledge with more experienced students from other universities. Nevertheless, the motivation to study gradually decreased in the following years – turned into lower academic results – as no new teaching experiences were accomplished from 2008/2009.

In order to address this problem, we have incorporated learning through play into traditional classrooms to provide students with an authentic life experience. Considering this aim, we started a new educational activity during two different AI courses. It consisted of participating in a computer-based competition, called *Google AI Challenge*, with two groups of students from the University of Huelva (UHU) during a whole semester. This provided a new scenario with additional settings such as the absence of a physical meeting point and the participation in a worldwide environment. Thereby, this paper describes the followed methodology and reports the students' reactions in their involvement in the international competition. Thus, benefits of learning through play and extra effort required by both teacher and students are provided through this experience. As a result, the *Google AI Challenge* has contributed to significantly improve the achievement and skills of the students, while consolidating theoretical concepts.

The research question this paper aimed to examine was: what are the implications of learning through play for students in Computer Science? It had three main objectives: (1) to explore the importance of playful learning as a motivating factor for the university academic performance; (2) to situate an international online contest within teaching methodology and educational goals; (3) to undertake primary research about the educational experience outcomes using the students' feedback and their academic progress. Thus, the paper is organized according to the following sections: literature review, educational objectives, developed didactic methodology, competition development, educational experience assessment, and conclusions and recommendations.

Literature review

This section provides a theoretical framework on the importance of motivation for the academic achievement and surveys the state of the art of AI competitions in educational contexts.

Motivation and gaming theory

Formal education is characterized by a systematic learning model structured and conducted according to a set of curricular directives, often presenting fairly rigid objectives, contents and methodologies to both teachers and learners (Dib, 1988). Moreover, formal learning represents no natural way of human learning, only comprising between 18.5% and 5.1% for K12 (Kindergarten through Twelve) and graduate students, respectively (Banks *et al.*, 2007). In this setting, formal education will not always stimulate students as they demand higher naturalness, flexibility and interactivity to support their learning experience. In addition, students come to the learning scene with different commitment, ability and learning styles, thus distinctively influencing their degree of motivation (Kirkland & O'Riordan, 2013). As theories state, motivation represents a key factor to learn and attain a successful academic achievement (Amrai, Motlagh, Zalani, & Parhon, 2011; Maclellan, 2005; Williams & Williams, 2011).

Informal education refers to the real-life experience, whereby individual aptitudes, values, skills and knowledge are naturally acquired from the daily practice (Novosadova *et al.*, 2007). Informal education gives students the opportunity to engage in their learning processes by proactively participating through flexible methodologies and different learning styles (Chen &

Bryer, 2012). This broadens personal competencies more so than the ones developed by formal learning (e.g., leadership, discipline, responsibility, teamwork, conflict management, planning, organizing, interpersonal relationships). As a consequence, it is felt by learners as a more favourable, effective and stimulating methodology compared to a largely inefficient and unappealing formal education (Schulz, 2008).

Informal education and play are changing both the way we think about knowledge and learning, as well as the manner in which we structure work and ideas. Learning through play enables learners to construct their own knowledge based on the understanding of their personal experiences, as the educational constructivist theory states (Gagnon & Collay, 2006). Active learning is effective in motivating and improving student achievement by promoting creative thinking and multi-style learning approaches. Kinesthetic is the learning style best adopted by playful methods, but other VARK (visual, aural, read/write and kinesthetic) approaches can also be incorporated (Cannon & Newble, 2000). Learning through play is currently better documented for K12 than for undergraduates (Rice, 2009). The advantages of interactive learning for adults are clear and varied, especially in engineering education where practical knowledge requires direct interaction with phenomena rather than theoretical lessons (Rieber, 2001).

Teaching through play fosters active creativity, development of problem-solving strategies and self-confidence to try new challenges (Lester & Russell, 2008). However, experience is not always enough to achieve learning and some other aspects must be introduced in the educational process (Bolton, 2010). These are observation, analysis, critical reflection, abstraction of concepts and testing of acquired knowledge in new situations. In this context, competition-based learning represents a suitable scenario to provide all components required to achieve constructive learning. Nevertheless, a low percentage of teachers and students take advantage of the high popularity of games for educational purposes.

Active learning through competitions has proven to be a captivating learning factor by enabling students to attain knowledge for themselves through activity and reasoning (Carpio Cañada *et al.*, 2011). This learning approach is characterized by a student-centred perspective where the process is more important than the outcome. Thus, teachers become the means to guide through the learning process, while motivated students learn about a course through problem-solving challenges (Hmelo-Silver, 2004). As benefits, students naturally respond to this type of learning, while games offer a medium to form and reform ideas in a fun and interactive way. As a result, the more motivated and engaged the students are, the more learning occurs (Squire & Jenkins, 2003).

Learning through play in AI

Since *Alan Turing* first established that games could be automatically played by machines using logical algorithms, these have been used as a teaching methodology to train different AI concepts (Turing, 1950). This turned games into potentially successful tools used to teach a wide variety of practical methods because of their ability to stimulate students, providing spontaneity, flexibility and interactivity to support their learning experiences (Moursund, 2007). The more representative examples in education are classic board games as *Backgammon*, used to teach exploring methods by reinforcement learning algorithms (Moursund, 2006); *Checkers*, used to develop search-based problem-solving techniques (Sturtevant, 2008); *Tic-Tac-Toe*, used for min-max and alpha-beta pruning (Michulke & Schiffel, 2011); *N-puzzle*, used for state-based search (Markov, Russell, Neller, & Zlatareva, 2006); or *n-Queens*, used to teach constraint satisfaction problems (Letavec & Ruggiero, 2002), among others.

Teachers have found that the students' motivation plays a key factor in learning and attaining successful academic achievement by the challenges proposed within courses. For example, *The Open Racing Car Simulator* – an open source and highly portable multi-platform framework – has been used as ordinary three-dimensional (3D) car game for the teaching of mechanical principles at the Northern Illinois University (Coller, 2009). Furthermore, several *RoboCode* leagues have been organized in the National University of Maynooth with the aim of teaching programming languages (O'Kelly & Gibson, 2006). In them, students are challenged with the design of intelligent agents – called bots – to compete ones against others trying to mimic human behaviour (Eisenstein, 2003). In other cases, competitions help to discover talented and

skilled students from engineering schools. As an example, the *Facebook Hacker Cup* international competition has been proposed since 2011 with this purpose, which consists of solving a number of algorithmic-based problem statements using any programming framework or language (Forišek, 2013). In addition, the Wichita State University has actively used *Lego Mindstorm* for the *First Lego League*. This competition has also proven to be a useful teaching methodology for K12 students whereby individual aptitudes, values, skills and knowledge have been naturally acquired according to informal education (Whitman & Witherspoon, 2003).

With the aim of using AI systems as testing platforms to promote both education and research in this field, several national and international contests have recently appeared. For example, the Stanford University used the *AAAI (Association for the Advancement of AI) General Game Playing* as an excellent development framework for students during a summer competition (Genesereth, Love, & Pell, 2005). Furthermore, the University of Hartford has developed and tested a suite of projects – called *MLExAI* (Machine Learning Experiences in AI) – that can be closely integrated into introductory courses to teach AI through machine learning (Neller, Russell, & Markov, 2008). The University of Essex started the *Ms Pac-Man vs. Ghost League* to compete against bots submitted by other competitors, which was previously tested with success by teachers and learners on AI courses (Szita & Lorincz, 2007). Other recent game competitions regularly held by universities are the *Physical Travelling Salesman Problem*, a single-player game aimed at solving combinatorial optimization problems with AI controllers (Perez, Rohlfshagen, & Lucas, 2012); the *Simulated Car Racing Championship*, an event consisting of three competitions where computational intelligence techniques were applied to car controllers for a racing game (Loiacono *et al.*, 2010); the *Mario AI Championship*, a benchmark used in several competitions related to international conferences on research and/or education (Karakovskiy & Togelius, 2012); and the *StarCraft AI Competition*, an advanced strategy game for which AI-based bots had to beat expert human players in real time (Togelius *et al.*, 2010), among others. These paradigms represent a scenario where observation, abstraction of concepts, critical thinking, analysis and acquired knowledge concur into a successful educational process with the aim of a competition.

In this context, the *Google AI Challenge* appeared as a biannual online contest initially organized in 2009 by the University of Waterloo and sponsored by Google (Savchuk, 2012). A different game is chosen every year and contestants shall submit specialized bots to play against other competing bots (Perick, St-Pierre, Maes, & Ernst, 2012). The topics in these series of competitions have been *Rock-Paper-Scissors* (2009/Fall), *Tron Light-Cycles* (2010/Spring), *Planet Wars* (2010/Fall) and *Ants* (2011/Fall). Although the first edition was based on a widely known game, the following competitions pursued the design of completely original games to try new challenges. This provided a captivating factor to explore new approaches, experiment with different ideas and ultimately find solutions to problems by worldwide students.

In order to focus the framework of this educational project, Table 1 shows an overview of the different competitions involved in the aforementioned educational experiences. The *Google AI Challenge* is distinguished for being an international contest – played online in multiplayer mode – for both university and professional levels. It has been used herein to teach a variety of AI topics (e.g., genetic algorithms, neural networks and fuzzy logic), while illustrating new ways of teaching programming languages through the implementation of intelligent agents.

Educational objectives

One of the pedagogical goals intended with this project has been to improve the motivation and interest of our students to study. In a previous experience, the teachers realized how the fact of changing – for a few days – the traditional classroom for a robotic competition hall influenced learning (Carpio Cañada *et al.*, 2011). In the case of the *Google AI Challenge*, new parameters as the lack of having a real space to play the competition – in contrast with virtual – were added. That is, only a classroom with computers, the Internet and no more special needs were required. From our experience, the lack of a meeting point did not limit the implementation of this teaching project despite being conducted through a virtual environment. Moreover, the *Google AI Challenge* offered no monetary reward to

Table 1. Main Features of Challenges Used as Educational Methodology by Some Authors

Challenge	Competition context	Teaching level	Game mode	Educational goal	Used methodology	Programming language	Project year
FLL[1]	National/physical	K12 students	Team players	Real-world basics related to the sciences	Mechatronic design and robot programming	NXT-G, Robolab	2003
RoboCode League[2]	International/Web server	University	Multiplayer	Language programming	Intelligent agents	Java™, C#, VB, .NET	2003
Pac-Man vs. Ghost League[3]	International/Web server	University	Multiplayer	Artificial intelligence	Reinforcement learning	Java™	2007
MLExAI[4]	Multi-institutional/ laboratory	University	Single player	Data mining, neural networks and machine learning	Web recommender and classification, pattern recognition, data mining, games	Java™	2008
TORCS[5]	Laboratory/Client-server	University	Multiplayer	Mechanics	Dynamic systems and control	C++	2009
Mario AI Championship[6]	International/Web server	University	Single player	AI techniques	Reinforcement learning	Java™ , C++ and Python	2009
Simulated Car Racing[7]	International/Client-server	University	Team players	AI techniques	Intelligent agents	C++	2009
StarCraft AI Competition[8]	International/Web server	Professional	Single/multiplayer	Advanced AI techniques	Planning, data mining, machine learning, case-based reasoning	C++	2009
Facebook Hacker Cup[9]	International/Web server	Professional	Two Players	Identify top engineering talents	Algorithmic-based problem statements	Any	2011
Google AI Challenge[10]	International/Web server	University	Multi Player	AI techniques	Intelligent agents	Any	2011
Physical Travelling Salesman[11]	International/Web server	University	Single player	Combinatorial optimization problems	Intelligent agents	Java™	2012

Note. List of the authors' names is given by the superscripts of the challenges: [1](Whitman & Witherspoon, 2003); [2](O'Kelly & Gibson, 2006) & (Eisenstein, 2003); [3](Szita & Lorincz, 2007); [4](Neller *et al.*, 2008); [5](Coller, 2009); [6](Karakovskiy & Togelius, 2012); [7](Loiacono *et al.*, 2010); [8](Togelius *et al.*, 2010); [9](Forišek, 2013); [10](Savchuk, 2012); [11](Perez *et al.*, 2012).

contestants so, unlike other competitions, this was not a factor influencing the students' participation.

Furthermore, we found that an international event was a much more appealing factor for the students' participation. That is, the students felt this experience as an opportunity to measure themselves against students from other universities, many of them prestigious ones. Thus, the students were able to test – without having to physically move to the competition site – the knowledge acquired during their study years within the context of a competition. This way, the students discovered through play that they could solve challenges in the same way than students from influential universities and professionals from all around the world. Consequently, their self-esteem to try new activities was reinforced and caused positive changes in the perception of their abilities (see *Educational Experience Assessment*).

Other educational goals proposed in this project were to promote teamwork and information sharing. In order to achieve these goals, the students were encouraged to work in groups, discuss solutions together and share information about their programs. The teacher allowed this scenario as long as the students implemented their own solutions. This was made possible by primarily using the forum of the competition's website. However, this is not limiting and other electronic resources predominantly used by universities (e.g., blogs, chats, Moodle) can be used by teachers to accomplish similar experiences (Martín-Blas & Serrano-Fernández, 2009). In previous editions of the *Google AI Challenge*, the participants shared information from the very beginning of the competition, thus facilitating the creation of high-quality intelligent agents. However, such spirit was not achieved in the *Google AI Challenge 2011* from the beginning. As an example, the post published by *a1k0n* – winner of the *Tron Light-Cycles* edition – called attention to this circumstance (Sloane, 2011). The message sent to the organizer's forum began with the following paragraph:

> I miss the collaborative nature of the Tron contest where everyone basically revealed their strategy in the forum and generated better ideas. Everyone's been much more tight-lipped since then. So I'm going to reveal mine here and now.

This message claimed the collaborative spirit of the competition and determined the beginning of the collaboration between contestants. Thus, our students found that – regardless of the position obtained in the final classification – information sharing and teamwork were essential to carry out their works.

Developed didactic methodology

The pilot experience started in the academic year of 2011/2012 during which students attended *Artificial Intelligence Laboratory* (AIL) or *Artificial Intelligence and Knowledge Engineering* (AIKE) courses; both in the 3rd and 4th years of their Computer Science degree at the University of Huelva (UHU). The AIL course was optional in contrast with the AIKE course, which was mandatory for the students. This different nature provided an ideal scenario for testing various educational goals addressed with the same experience. Furthermore, the *Ants* game for the *Google AI Challenge* was used as a novel educational methodology, thus providing teachers valuable information to meet new challenges on AI courses.

To conduct this educational project, we extended regular classroom lessons with additional work performed by both teachers and students. It gave our students the possibility to freely discover AI techniques with the aim of competing internationally. Hence, this educational experience was not only felt by the students as a series of practice sessions in lab. Table 2 shows the work carried out by the teacher and students to adapt the *Google AI Challenge* into the programs of AI courses. The two courses have been offered annually maintaining the same structure and length since 2004/2005.

The AIL course comprised a total of 120 h of student work divided into 50 and 70 h of classroom and non-classroom instructions, respectively. The hands-on components usually consisted of recognizing data structures, AI techniques (e.g., evolutionary algorithms, fuzzy logic, neural networks), and learning both programming languages and common tools used on AI. Moreover, the AIKE course comprised a total of 290 h of student work divided into 130 and 160 h of classroom and non-classroom instructions, respectively. The hands-on activities are designed to build intelligent systems for the automatic demonstration of theorems, implement search and planning algorithms, and coordinate intelligent agents in lab practices. In summary, the differences in the courses' structure are

Table 2. Integration of the Educational Experience in the Teaching of Artificial Intelligence Courses

ID	AIL program	AIKE program	Dedication to the competition	Date	Competition status	Teacher work	Student work
1	Overview of AI	Overview of AI	1 session	25 October 2011	Competition starts	Description of challenge goals	Website overview, user account set-up, tools and forum use
2	Genetic algorithms	Statistics, uncertainty and Bayesian networks		–	–	Traditional classroom	Quickstart guide and first basic work entry
3	–	Machine learning		–	–	–	Learn new programming languages (optional)
4	–	Logic and planning		–	–	–	Implement intelligent agents
5	Neural networks	Markov decision processes and reinforcement learning	1 weekend	18 December 2011	First phase	Customize strategies	Testing and debugging own AI algorithms
6	–	Hidden Markov models and filters		–	–	Traditional class work	Share information between students
7	–	Adversarial and advanced planning		–	–	–	Upload bots to online contest
8	Fuzzy logic	Image processing and computer vision	1 session	24 December 2011	Final competition	Bots competition	Get final ranking
9	–	Robotics and robot motion planning	1 session	26 December 2011	Offline competition	Knowledge and skills	Feedback on AI algorithms and intelligent agents
10	–	Natural language processing and information retrieval	1 session	9 January 2012	–	Student achievement	Opinion survey and experience discussion

AI = Artificial Intelligence; AIKE = Artificial Intelligence and Knowledge Engineering; AIL = Artificial Intelligence Laboratory.

that AIL was aimed at introducing AI concepts in a completely practical way by programming AI techniques and algorithms. On the contrary, AIKE – of a more advanced level than the previous one – intended to give students greater theoretical and practical knowledge on AI, thereby including other fields such as robotics and computer vision.

The teaching methodology was as follows. The challenge goals were introduced by the teacher in a first introductory session of 1.5 h (ID1) 2 months before the end of the competition (see Table 2). Then, the students learned to use the competition website, activate their user accounts and sent their first basic entries to the virtual organizer's platform (ID2). During the following preparatory weeks, the teacher explained the game operation and the students were guided on the implementation of their intelligent agents (ID3–ID4).

The *Google AI Challenge* nature allowed our students to start writing simple codes for their bots without high programming skills. Therefore, the students were able to write their algorithms in different languages (e.g., C++, Python, Java™) using a starter kit available and send them to an application program interface built by the competition organizer. Since most of the students had no previous knowledge of Python programming – one of the educational goals intended herein – they were encouraged to learn some basic language within the context of the competition. Although learning a new language was an additional effort, it allowed the students to acquire new knowledge without affecting their course performance (see *Validation as Educational Experience*).

The first phase of the competition lasted a week (ID5–ID7). For this purpose, a programming marathon was organized by the teacher and students during a weekend. The aim was for students in 3rd and 4th years to meet in a common place to share ideas about their programs (i.e., the classroom). During this phase, the tasks consisted of designing strategies and testing AI algorithms while the teacher followed up their works. During this phase, the students were qualified to make changes in their bots and upload new versions to the virtual platform of the competition. The score was reset to the end of the ranking with each new version. However, this action was not penalized by the organizer as the virtual platform was designed to quickly promote skillful intelligent agents, thus fostering the students' critical reasoning about the construction of their algorithms. During this stage, sharing information between the students to learn the techniques used by their classmates became essential in the progress and quality of the algorithms. From our experience, this short period of time was the most productive of all the time spent on programming. As an example, some students worked up to 30 h from 16:00 h on Friday to 06:00 h on Monday.

Once the final phase of the competition began, it was not possible to upload new versions of bots to the competition's platform (ID8). The waiting time between games was 1–4 h. Hence, participants could only monitor their matches against other players while the problem-solving strategies and educational goals learned were discussed. As a result of the motivation provided by the *Google AI Challenge*, most of the students improved their algorithms and intelligent agents even in the days after the competition. This was possible since the virtual platform of the *Google AI Challenge* was available offline (ID9). Finally, students were asked to ascertain the impact of this educational experience on teaching in engineering (ID10). In summary, the cost of putting into practice this experience required an average time of 24 and 20 h per student in each course, and a total of 22 and 32 h by the teacher, respectively (see Figure 1).

Competition development

The *Google AI Challenge* was held by 7897 contestants from 116 countries. The *Ants* game was used as the basis for the 2011 edition. The strategy of this game consisted of managing an ant colony in order to fight against other colonies for domination. The game took place on a map where participants initially had one or more anthills. The purpose of an anthill was to generate ants, which should be controlled by each bot. Participants had to perform the movements they deemed appropriate through a turn-based system. Actions to get points by bots were to explore the map, attack enemy hills, gather food, avoid collisions and to not block their own anthills. The rules stipulated that each participant had to give a token to the game server, thus indicating the end of movements before the turn expired. If the token was not submitted on time, the player received penalty points and was prevented from making movements in the remaining turns. However, this did not imply the disqualification and a player

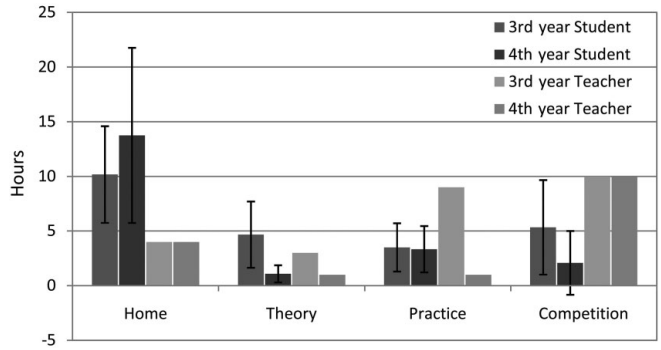


Figure 1 Times Devoted to the Project Development Depending on the Role of the Person

could even win the game if enough points were accumulated (further goals and rules of the *Google AI Challenge* are available at http://aichallenge.org).

Matches were played on the organizer's server during the challenge and contestants were able to play them on the website after each round. Thus, our students were able to upload different versions of their programs to test ideas and improve their intelligent agents. In the final phase, entries were closed, the classification was restored and the latest version of each bot played several matches to determine the final classification (Savchuk, 2012). During the competition, the user ranking was continuously updated through TrueSkill™ (Herbrich, Minka, & Graepel, 2007). This tool represents a Bayesian classification algorithm developed by Microsoft Research, which allows you to obtain a ranking based on the skill of each intelligent agent. Henceforward, the skills were tracked by the system after each game to determine the individual abilities of each player over other contestants.

There were no restrictions on the techniques used and any AI algorithm learned in the course could be used to promote the students' creativity. Thereupon, the students in 3rd and 4th years followed two different methodologies for developing their intelligent agents. As a starting point, the students in 3rd year studied some sample bots provided by the organization. After making and testing the initial versions, the students found that the best performance was obtained by combining two basic bots – named *Lefty* and *Hunter* – instead of using an intelligent agent as the single best solution. Thus, the proposed strategy consisted of alternating the two bots in a series of turns, each one with a different behaviour. Figure 2 shows the basic structure of the algorithm used in the competition. The variable *ArraySchedule* sets the number of turns for bots, which allowed controlling the bots' strategies to combat more efficiently. In kind, little variations in the cycles influenced the expansion rate of the ants over the map. This working methodology was used by 90% of our students.

The methodology, although also available for the students in 4th year, was discarded as they preferred to implement their own bots because of their higher knowledge on AI techniques learned. As an example, the techniques taught during the months prior to the competition were the tree search, graph search, breadth-first search, uniform-cost search and A* search, among others. However, the latter was able to find the minimum cost path between two points (Hart, Nilsson, & Raphael, 1968). Consequently, the design of the intelligent agents mostly consisted of searching techniques based on the A* algorithm to compute optimal routes between the ants and targets (Cowley, 2012). In order to illustrate a paradigm of learning through play, Figure 3 shows a match between three

```
function ChooseStrategy()
  initialize ArraySchedule
  for i=0 to size (ArraySchedule)
    if ArraySchedule[i] == 'h' then
      HunterBotMode()
    else if ArraySchedule[i] == 'l' then
      LeftyBotMode()
    end if
  end for
end function
```

Figure 2 Example or Programming Code for Intelligent Agents

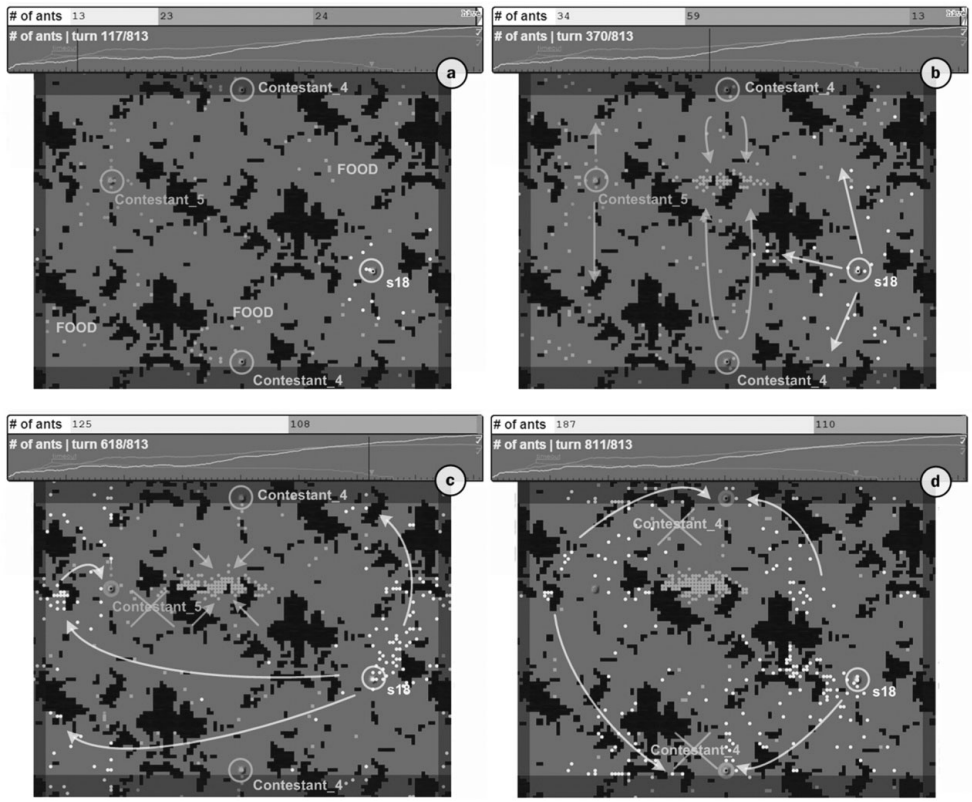


Figure 3 Evolution of Strategies for a Three-Player Match: (a) Location of the Ant's Communities at Turn 117/813, (b) Turn 370/813, (c) Turn 618/813 and (d) Turn 811/813

competitors during 813 turns: the teacher (grey colour) and two contestants (orange and blue). As shown, different behaviours and movements experienced by the bots can be seen. The image sequence displays how the evolution of grey ants (named *s*18) shows higher dispersion than the other communities as a result of an exploration strategy to look for food and enemy anthills. Thus, the students learned that the success of an intelligent agent is determined by a trade-off among several fitnesses; that is, to collect food and attack enemy anthills (brown pixels and circles coloured in Figure 3).

An analysis about the influence of the *Google AI Challenge* on the students' motivation to achieve additional goals – such as learning new programming language – was carried out. Therefore, a *t*-test was applied to over 300 contestants worldwide considering one nominal variable (i.e., programming language) and three measurement variables (program version, skill and bots' position). The null hypothesis was that the mean measurements between two categories of programming language (i.e., C++, Java™ and Python) were the same. Table 3 shows a comparative summary of the official ranking of our students regarding the winners of the competition.

Precisely, we have found out significant differences between the number of versions and the programming language adopted by the contestants. The results of the *t*-test reject the null hypothesis when Python is considered ($p = 0.147$ for C++ vs. Java™, $p = 0.031$ for C++ vs. Python, $p < 0.001$ for Java™ vs. Python). Besides, the bots with better skills were programmed by

Table 3. Comparison of the Final Ranking of the *Google AI Challenge 2011*

Rank	Username	Role/level	Language	Versions	Games played	Skill
1	*c*1	2nd year student	Java™	3	169	90.68
2	*c*2	Professional	Java™	15	170	89.98
3	*c*3	Professional	C++	5	171	87.37
472	*s*1	4th year student	C++	18	171	65.25
1514	*s*2	4th year student	Python	17	55	51.57
1516	*s*11	3rd year student	Python	13	54	51.54
1812	*s*18	Teacher	Python	41	53	49.06
1873	*s*12	3rd year student	Python	26	51	48.62
2076	*s*13	3rd year student	Python	8	49	47.26
2079	*s*3	4th year student	Python	10	39	47.25
2194	*s*14	3rd year student	Python	3	43	46.53
2254	*s*15	3rd year student	Python	14	39	46.23
2296	*s*4	4th year student	Python	3	38	45.96
2323	*s*16	3rd year student	Python	10	38	45.80
2414	*s*17	3rd year student	Python	4	49	45.31
2647	*s*5	4th year student	Python	1	41	43.79
4139	*s*6	4th year student	Python	29	17	39.99
4450	*s*7	4th year student	Python	1	19	39.56
5157	*s*8	4th year student	Python	21	16	38.29
5265	*s*9	4th year Student	C++	6	17	38.14
6126	*s*10	4th year student	Python	3	13	36.93

contestants who chose C++/Java™ as programming language instead of Python ($p = 0.716$ for C++ vs. Java™, $p < 0.001$ for C++ vs. Python, $p < 0.001$ for Java™ vs. Python). As an example, the winners' bots were programmed in C++/Java™ by contestants – named *c*1, *c*2 and *c*3 – who participated in previous editions of the *Google AI Challenge*, some of them professional programmers (Lichtenberger, 2011; Voronyuk, 2011). On the contrary, the students from the UHU – with positions from 472 to 6126 – mostly used Python as preferred programming language. In fact, the results of the *t*-test reject the null hypothesis when Python is considered again ($p = 0.513$ for C++ vs. Java™, $p < 0.001$ for C++ vs. Python, $p < 0.001$ for Java™ vs. Python).

As a conclusion, the analysis suggests that C++/Java™, more efficient and robust, is preferred by more experienced users. However, Python, easier and faster to implement, is preferred by many other users, mostly beginners. Although the number of versions required by participants was influenced by the programming language, their final position did not depend on the versions or languages, but the skills achieved by the intelligent agents. As a result, decisions on language had no influence on the students' motivation since the programming language was felt as part of the learning process (see *Educational Experience Assessment*). For these reasons, we considered the *Google AI Challenge* as an appropriate learning experience to try artificial intelligence and additional educational goals. In effect, teachers can encourage their students to learn advanced concepts on Computer Science through play regardless of the programming knowledge.

Educational experience assessment

This section presents the results of a comprehensive study on the applied methodology considering two different areas. Firstly, the students' opinion regarding the educational experience is analysed by means of a questionnaire elaborated by our multidisciplinary research team. Secondly, the students' academic progress is compared over three academic years.

Evaluation of the students' opinion

A statistical study has been carried out on the students of 3rd and 4th years, respectively (see Table 4). All participants were asked – at the end of the experience – to complete a questionnaire based on a five-level Likert scale (1 = strongly disagreed; 5 = strongly agreed). This consisted of covering four analysis areas with the aim of exploring the difference in self-reporting between these student groups regarding the experience.

Table 4. Evaluation Questionnaire of the Students' Self-Reporting about the Educational Experience

Question	Knowledge	Average value	*SD*
1	Previous level on AI	1.89	±0.32
2	Final level on AI	3.42	±0.45
3	The experience enables the consolidation of theoretical concepts on AI	3.73	±0.49
4	The experience allows new theoretical concepts on AI to be acquired	3.89	±0.43
5	The experience allows to discover new own ways to solve problems	3.73	±0.40
6	The experience allows new theoretical concepts on language programming to be acquired	3.47	±0.56
7	My ability to apply knowledge in practical and real problems after the challenge is positive	3.22	±0.43
	Interest/motivation		
8	My general assessment for the course before the experience is positive	3.10	±0.40
9	My general assessment for the course after the experience is positive	4.05	±0.35
10	The general assessment for my degree before the experience is positive	3.36	±0.41
11	The general assessment for my degree after the experience is positive	3.84	±0.44
12	The general assessment for my university before the experience is positive	3.15	±0.34
13	The general assessment for my university after the experience is positive	3.42	±0.34
14	My general assessment for the teacher before the experience is positive	3.73	±0.32
15	My general assessment for the teacher after the experience is positive	3.89	±0.32
16	The mark obtained in the challenge influences learning on AI	2.00	±0.42
17	The mark obtained in the challenge influences interest and motivation	3.00	±0.64
18	Competing in a national context promotes motivation and interest	4.21	±0.42
19	Competing in an international context promotes motivation and interest	4.21	±0.42
20	Programming through the play promotes motivation and interest	4.42	±0.25
	Personal skills		
21	The need to travel abroad to further my education before the challenge is positive	3.63	±0.53
22	The need to travel abroad to further my education after the challenge is positive	3.89	±0.55
23	The value of sharing information before the challenge is positive	3.42	±0.45
24	The value of sharing information after the challenge is positive	4.00	±0.37
25	The challenge has served to better understand personal skills	3.47	±0.60
26	The experience allows knowledge on work organization to be acquired	3.10	±0.46
27	The experience allows knowledge on cooperation and teamwork to be acquired	3.21	±0.54
	Human workload/difficulty		
28	The difficulty and workload of this practice/experience is high	3.68	±0.44
29	This practice/experience is feasible for implementation in the university context	4.15	±0.47
30	The general assessment on development and organization of this practice is positive	3.15	±0.44
31	My working capacity before the challenge is positive	3.42	±0.38
32	My working capacity after the challenge is positive	3.78	±0.42
33	My comprehension before the challenge is positive	3.52	±0.48
34	My comprehension after the challenge is positive	3.78	±0.42
35	My general assessment for this practice/experience is positive	3.78	±0.48

AI = Artificial Intelligence.

That is, knowledge acquisition (Q1–Q7), interest/motivation (Q8–Q20), skills development (Q21–Q27) and human workload/difficulty (Q28–Q34). With this purpose, a total of 19 volunteers participated (7 and 12 students for each course, respectively). In general, the results showed the highest differences mainly in the areas of interest/motivation and human workload/difficulty (see Figure 4).

Regarding the area of knowledge acquired on AI, all the students claimed to have a greater level after the educational experience (Q1 vs. Q2). Subsequently, we found a significant increase with respect to the final

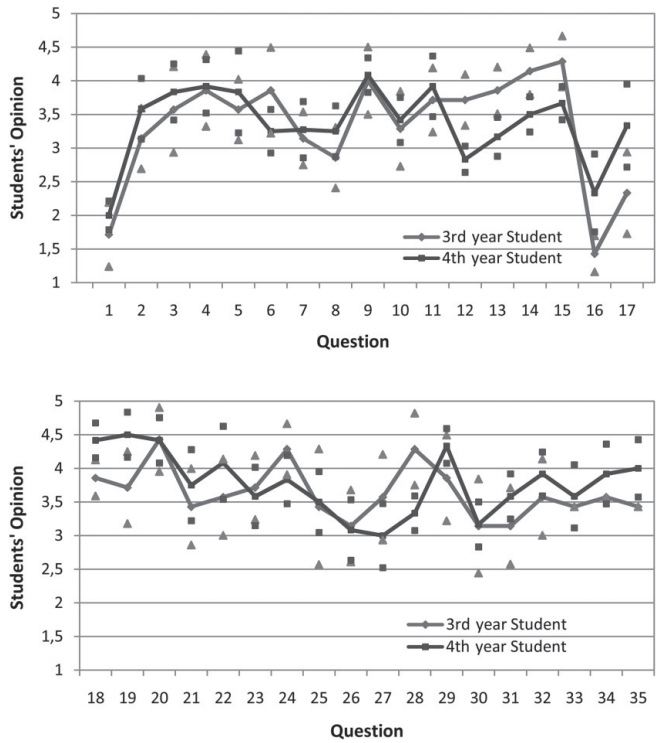


Figure 4 (a) Average Score and (b) Standard Deviation for the Students' Opinion Based on a Five-Level Likert Scale

knowledge on AI acquired by the students as shown in Table 5 (for Wilcoxon signed-rank test, $p < 0.001$). The students strongly agreed that this experience enabled the consolidation and acquisition of new theoretical concepts, allowing them to discover new ways to solve problems (Q3–Q5). Besides, the learning of new theoretical concepts on language programming was well rated by the students (Q6). However, the average score was higher for those students belonging to the 3rd course, which is consistent with the fact that students in the 4th course were more experienced on this matter (see Figure 4).

Table 5. Wilcoxon Signed-Rank Test for Questions Measuring the Same Construct Before and After the Educational Experience

Before	After	p value	Significant
Q1	Q2	$p < 0.001$	✓
Q8	Q9	$p = 0.003$	✓
Q10	Q11	$p = 0.007$	✓
Q12	Q13	$p = 0.043$	✓
Q14	Q15	$p = 0.224$	–
Q21	Q22	$p = 0.043$	✓
Q23	Q24	$p = 0.027$	✓
Q31	Q32	$p = 0.011$	✓
Q33	Q34	$p = 0.043$	✓

Regarding the area of interest/motivation, we found a significant increase in the students' opinion about the courses involved in the experience (for Q8 vs. Q9, $p = 0.003$). This suggests that the contest positively influenced the students' feelings about their courses. We discovered that it is also applicable when the students were asked about the general perception of both their degree and university (for Q10 vs. Q11, $p = 0.007$; for Q12 vs. Q13, $p = 0.043$), respectively. Correspondingly, we found that answers from Q12 to Q15 were

more positive for 3rd year students, although very positive in general. Resultantly, 4th year students opined that areas like motivation, interest and AI learning were less influenced as a consequence of the mark obtained in the challenge (Q16, Q7). We suspect that the reason may be that 4th year students had higher expectations as a consequence of joining the international contest than 3rd year students (Q18, Q19). However, both groups of students agreed very similarly that learning through play promoted the motivation and interest in the same level (Q20).

Regarding the area of personal skills development, we noted in all the students a positive change of mind on the need to travel abroad to complete their training after participating in the international competition (for Q21 vs. Q22, $p = 0.043$). Specifically, 4th year students realized the need to go abroad to complete their education to a greater extent than 3rd year students (Q22). As another implication for education, the need for sharing information was highly valued by the students in general as a means to share experiences and provide feedback of their knowledge (for Q23 vs. Q24, $p = 0.027$). Characteristically, 3rd year students found the need to share information after the challenge more valuable than 4th year students (Q23, Q24). These suggest that both beginner and experienced students differently appreciated this form of learning due to their limitations and knowledge of the matter.

Regarding the human workload/difficulty, we found that the students considered the level of difficulty and workload of the practice/experience as medium-high. Respectively, 3rd year students felt that the difficulty and workload was higher compared with 4th year students (Q28). We believe that the reason is because 3rd year students enrolled for the first time in an AI course as opposed to the more experienced 4th graders. This suggests that the assessment that the students made about the implementation of an educational experience was proportional to the degree of the practice's difficulty. Nonetheless, the feasibility to implement this experience in the university context was highly rated in general (Q29). Moreover, we found significant differences about the working capacity developed by the students before and after the challenge (for Q31 vs. Q32, $p = 0.011$), thereby resulting in a comprehensive improvement due to the educational experience (for Q33 vs. Q34, $p = 0.043$). Results are validated in Q35, where the general students' opinion about the educational experience was given as fairly positive and satisfying.

Evaluation of academic results

In order to evaluate the educational impact of this teaching experience, a statistical study considering 83 students along three academic courses has been made (see Figure 5). On the one hand, Figure 6a shows a comparison on the average grade of the students (being A = 8–10, B = 7–7.9, C = 6–6.9, D = 5–5.9, F = 0–4.9 points, respectively). The grades were very similar each course for 3rd year students, being the grade in the last course – where the experience was carried out – slightly higher than the previous ones (7.8, 8.0 and 8.1). By contrast, the average grade for 4th year students in 2011/2012 showed a significant increase regarding the previous courses (5.45, 4.42 and 7.16). On the other hand, Figure 6b shows a comparison on the percentage of students who did not attend

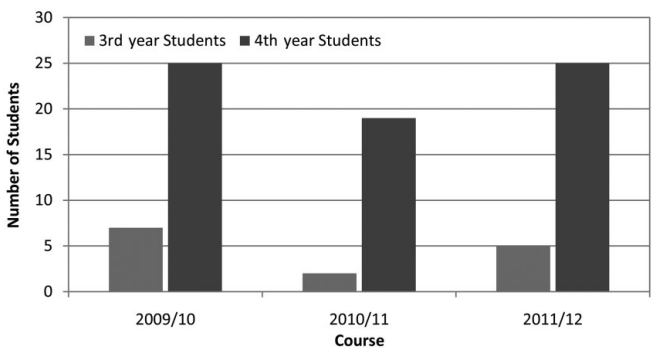


Figure 5 Distribution of the Students Enrolled during the Three Academic Years

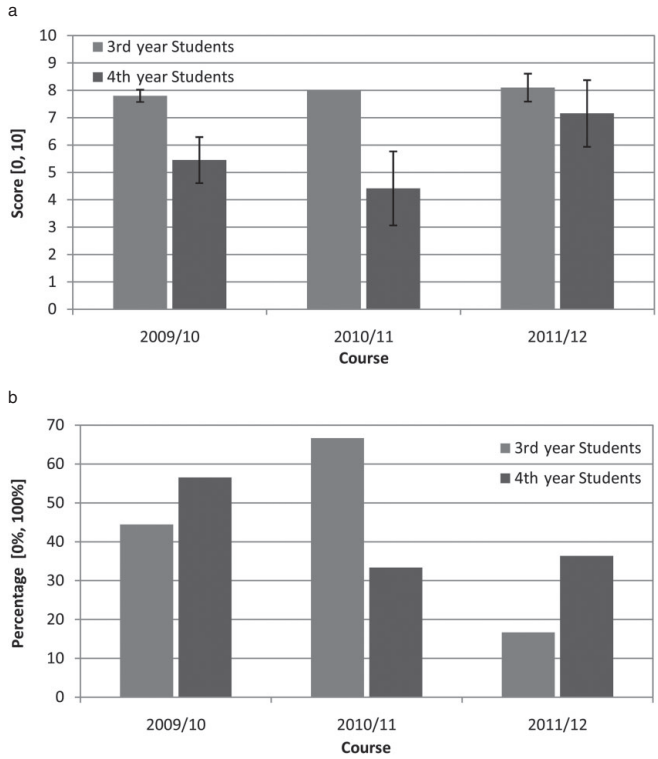


Figure 6 Statistics for 3rd and 4th Year Students: (a) Academic Scores for Students Who Attended the Exams and (b) Non-Attendance for the Overall of Students

examination sessions. In the case of 3rd year students, the number of students not attending exams was drastically reduced in 2011/2012 compared to the previous years (44.44%, 67% and 17% over the total of students, respectively). In the case of 4th year students, the trend during the year 2009/10 remained similar compared to the previous year, and significantly improved compared to the year 2019/2010 (56.52%, 33.33% and 36.4% over the total of students, respectively).

These results may be influenced by a large number of variables, observable or not, although the educational context during the courses' development may provide a better understanding of the students' attitudes. As for the difference in attendance in each course, we point out the optional nature of the 3rd year course as the main possible cause in contrast with the 4th year course, which was mandatory for the students. Moreover, although both courses were affected by different changes of tutor, we believe that this factor could have influenced 3rd year students to a greater extent since introductory courses on complex concepts could be more sensitive to these changes. By contrast, 4th year students were more experienced, had a greater number of teaching hours and thereby could be less responsive to changes of tutor. Regarding the increase in the students' ratings, both courses were structured according to the educational system previous to the European Credit Transfer and Accumulation System. As a principal disadvantage, this educational system – introduced in our country in 1983 – does not take into consideration the development of alternative activities in the traditional classroom or the hours of work–study that students should devote to overcome their studies, which is closer to formal learning. On the contrary, the European Higher Education Area (EHEA) is characterized by a student-centred perspective consistent with constructivist principles, which comes in purposes of our educational experience. Therefore, we believe that

the students' grades might have changed more significantly in the 4th year course due to this experience. The reason is because the 3rd year course is eminently practical (i.e., lab-based learning) and the 4th year course is structured in theoretical and practical hours, thus having the most noticeable impact on the practical experience.

Even though the findings cannot be generalized, we believe that a real experience such as the *Google AI Challenge* may positively influence the motivation of beginner students in optional courses – with special focus on practical training in lab – who are more susceptible to changes in teaching (e.g., tutors). Moreover, the learning experience may significantly improve the academic performance of learners in more advanced courses that demand practical study without affecting the levels of theoretical knowledge, which comes in direction of the EHEA. These results suggest that it has been possible to successfully incorporate the *Google AI Challenge* in our teaching system without compromising the educational goals of the courses involved. This may validate the implementation of new educational experiences by both teachers and students, making courses more appealing, as well as improving student achievement.

Conclusions and recommendations

This paper presents a new teaching experience in Computer Science with the aim of improving the academic achievement of students by increasing their motivation and interest. As a result, the authors implemented the idea of a computer-based competition, the *Google AI Challenge*, into students' AI courses.

In order to address the implications of learning through play for engineering education, we examined the importance of an interactive approach towards learning. In order to fulfill this objective, students' opinion considering knowledge, interest/motivation, personal skills and human workload/difficulty was analysed. According to the results, the advantages of participating in a computer-based competition enabled students to consolidate theoretical concepts, improve perception on courses, promote motivation and interest, and broaden personal skills (e.g., cooperation, teamwork, organization, value of information sharing).

The integration of challenge-based interactive learning into AI courses provided our students, unlike other approaches, all the components required to achieve constructive learning (i.e., observation, analysis, criticism, abstraction). Among the positive aspects, learning through play further allows you to apply multi-style approaches (as, e.g., VARK), which helps to stimulate, engage and captivate students, thus responding to the natural human learning process. Without a doubt, when teaching engineering degrees, learning is strongly facilitated by interactive approaches.

In the case of teaching advanced concepts or new programming languages, competition through game generates an added motivation for students. Furthermore, this type of interest is a valuable path towards broadening knowledge.

Playing through international competition gave learners the chance to measure themselves against others, to reinforce their self-esteem and to enrich their knowledge by fostering proactive participation. This provided a real-life experience not always provided by formal education. In addition, playing in an online contest provided teachers additional settings to challenge the students (e.g., absence of a physical meeting point or monetary rewards), thus allowing the learning process to be more important than the outcome. In effect, the implementation of teaching experiences through play (as the *Google AI Challenge*) was possible by using a virtual working environment, even without rewards.

In order to evaluate the academic achievement, a statistical study over three academic years was carried out. The results showed that students increased their average grades after the experience. Moreover, the results point out that interactive learning approaches are highly recommended to decrease the number of students not attending exams because of the positive motivation felt with competitions. Despite requiring a minimal extra work by teachers and students, the successful incorporation of computer-based competitions – as the methodology followed herein – is possible without compromising the educational goals. That is, learning through play can satisfy expectations for adults and improve traditional teaching methodologies in higher education.

The educational experience was conducted around engineering education with students from different backgrounds and motivations (i.e., volunteers and non-volunteer students from two course levels). While the results are applicable to a wide range of disciplines, it

is not clear how it relates to the non-technical studies. The research did not try to examine these implications and future efforts could be addressed in this line of study. Nonetheless, the findings presented herein will help to enable a useful comparison between disciplines.

Acknowledgements

The present work was supported by the national project TIN2011–28627-C04-02 and P08-TIC-03903 awarded by the Andalusian Regional Government. We would like to express our very great appreciation to Dr. E. Gualda Caballero, Dr. A.M. Mora García and P. García Sánchez for their valuable and constructive suggestions that helped improve this research work.

References

Adams, J., Hill, P., & Slater, T. (2000). *Instructional guide for MSU faculty*. Bozeman, MT: Montana State University.

Amrai, K., Motlagh, S. E., Zalani, H. A., & Parhon, H. (2011). The relationship between academic motivation and academic achievement students. *Procedia Social and Behavioral Sciences*, *15*, 399–402.

Banks, J., Au, K., Ball, A., Bell, P., Gordon, E., Gutierrez, K., . . . Zhou, M. (2007). *Learning in and out of school in diverse environments*. Technical Report: LIFE Center, University of Washington, Seattle, WA.

Bolton, G. (2010). *Reflective practice: Writing and professional development* (3rd ed.). London: Sage Publications Ltd.

Cannon, R., & Newble, D. (2000). *Handbook for teachers in universities and colleges: A guide to improving teaching methods*. New York, NY: Routledge.

Carpio Cañada, J., Mateo Sanguino, T. J., Alcocer, S., Borrego, A., Isidro, A., Palanco, A., & Rodríguez, J. M. (2011). From classroom to mobile robots competition arena: An experience on artificial intelligence teaching. *International Journal of Engineering Education*, *27*(4), 813–820.

Chen, B., & Bryer, T. (2012). Investigating instructional strategies for using social media in formal and informal learning. *The International Review of Research in Open and Distance Learning*, *13*(1), 87–104.

Coller, B. D. (2009). Video game-based education in mechanical engineering. A look of student engagement. *International Journal of Engineering Education*, *25*(2), 308–317.

Cowley, E. (2012). Ant Bot: Submission to the Google AI Challenge. *Conference on Computer-Human Interaction (CHI)*, pp. 1–3.

Dib, C. Z. (1988). Formal, non-formal and informal education: Concepts/applicability. *AIP Conference Proceedings*, *173*, 300–315.

Eisenstein, J. (2003). *Evolving RoboCode tank fighters*. Technical Report: Massachusetts Institute of Technology. Retrieved Nov 7, 2013 from http://dspace.mit.edu/handle/1721.1/30431

Forišek, M. (2013). Pushing the boundary of programming contests. *Olympiads in Informatics*, *7*, 23–35.

Gagnon, G. W., & Collay, M. (2006). *Constructivist learning design: Key questions for teaching to standards*. London: Corwin Press.

Genesereth, M., Love, N., & Pell, B. (2005). General game playing: Overview of the AAAI competition. *AI Magazine*, *26*, 62–72.

Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, *4*(2), 100–107.

Herbrich, R., Minka, T., & Graepel, T. (2007). TrueSkill™: A Bayesian skill rating system. *Advances in Neural Information Processing Systems*, *20*, 569–576.

Hmelo-Silver, C. E. (2004). Problem-based learning: What and how do students learn? *Educational Psychology Review*, *16*(3), 235–266.

Karakovskiy, S., & Togelius, J. (2012). The Mario AI benchmark and competitions. *IEEE Transactions on Computational Intelligence and AI in Games*, *4*(1), 55–67.

Kirkland, D., & O'Riordan, F. (2013). Games as an engaging teaching and learning technique: Learning or playing? *International Conference on Engaging Pedagogy (ICEP)*.

Lester, S., & Russell, W. (2008). *Play for a change: Play, policy and practice. A review of contemporary perspectives*. London: National Children's Bureau Enterprises Ltd.

Letavec, C., & Ruggiero, J. (2002). The n-Queens problem. *INFORMS Trans. Education*, *2*(3), 101–103.

Lichtenberger, M. (2011). *AI Challenge 2011 (Ants) post mortem by xathis*. Technical Report. Retrieved Nov 7, 2013 from http://xathis.com/posts/ai-challenge-2011-ants.html

Loiacono, D., Lanzi, P. L., Togelius, J., Onieva, E., Pelta, D. A., Butz, M. V., . . . Quadflieg, J. (2010). The 2009 simulated car racing championship. *IEEE Transactions on Computational Intelligence and AI in Games*, *2*(2), 131–147.

Maclellan, E. (2005). Academic achievement: The role of praise in motivating students. *Active Learning in Higher Education*, *6*(3), 194–206.

Markov, Z., Russell, I., Neller, T., & Zlatareva, N. (2006). Pedagogical possibilities for the N-puzzle problem. *36th Annual Frontiers in Education Conference*, pp. 1–6.

Martín-Blas, T., & Serrano-Fernández, A. (2009). The role of new technologies in the learning process: Moodle as a teaching tool in Physics. *Computers & Education*, *52*, 35–44.

Michulke, D., & Schiffel, S. (2011). Distance features for general game playing. *IJCAI-11 Workshop on General Game Playing*, pp. 7–14.

Moursund, D. G. (2006). *Brief introduction to educational implications of artificial intelligence*. Retrieved Nov 7, 2013 from http://darkwing.uoregon.edu/~moursund/Books/AIBook/index.htm

Moursund, D. G. (2007). *Introduction to using games in education: A guide for teachers and parents*. Retrieved Nov 7, 2013 from http://uoregon.edu/~moursund/Books/Games/games.html

Neller, T. W., Russell, I., & Markov, Z. (2008). Throw down an AI Challenge. *Proceedings of AAAI Spring Symposium: Using AI to Motivate Greater Participation in Computer Science*, pp. 67–73.

Novosadova, M., Selen, G., Piskunowicz, A., Mousa, S. H. N., Suoheimo, S., Radinja, T., . . . Reuter, P. (2007). The impact of non formal education on young people and society. In M. Nomikou (Ed.), *Non formal education book* (pp. 1–58). Belgium: AEGEE-Europe.

O'Kelly, J., & Gibson, J. P. (2006). RoboCode & problem-based learning: A non-prescriptive approach to teaching programming. *Proceedings of the 11th Annual SIGCSE Conference Innovation and Technology in Computer Science Education*, 38(3), pp. 217–221.

Perez, D., Rohlfshagen, P., & Lucas, S. M. (2012). 'The physical travelling salesman problem: WCCI 2012 competition'. *IEEE Congress on Evolutionary Computation*, *1*, 1–8.

Perick, P., St-Pierre, D. L., Maes, F., & Ernst, D. (2012). Comparison of different selection strategies in Monte-Carlo tree search for the game of tron. *IEEE Conference on Computational Intelligence and Games*, *1*, 242–249.

Rice, L. (2009). Playful Learning. *Journal for Education in the Built Environment*, *4*(2), 94–108.

Rieber, L. P. (2001). *Designing learning environments that excite serious play*. Technical Report: The University of Melbourne.

Savchuk, O. (2012). *Analysis of algorithms and strategies in the Google AI Challenge 2011* (Bachelor Thesis, Technische Universität Darmstadt).

Schulz, B. (2008). The importance of soft skills: Education beyond academic knowledge. *Journal of Language and Communication*, *2*(1), 146–154.

Sloane, A. (2011). *Implementing combat with random sampling*. Technical Report. Retrieved Nov 7, 2013 from http://forums.aichallenge.org/viewtopic.php?f=24&t=2044

Squire, K., & Jenkins, H. (2003). Harnessing the power of games in education. *Insight (American Society of Ophthalmic Registered Nurses)*, *3*(1), 7–33.

Sturtevant, N. R. (2008). An analysis of UCT in multiplayer games. *Lecture Notes in Computer Science*, *5131*, 37–49.

Szita, I., & Lorincz, A. (2007). Learning to play using low-complexity rule-based policies. *Journal of Artificial Intelligence Research*, *30*(1), 659–684.

Togelius, J., Preuss, M., Beume, N., Wessing, S., Hagelbäck, J., & Yannakakis, G. N. (2010). Multiobjective exploration of the StarCraft map space. *IEEE Conference on Computational Intelligence and Games*, *1*, 265–272.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind; A Quarterly Review of Psychology and Philosophy*, *59*, 433–460.

Van Kollenburg, P. A. M., & van Schenk Brill, D. (2009). Motivating students in engineering & ICT education. *SEFI 2009 Annual Conference*, pp. 1–14.

Veganzones, C., Martinez, S., Arribas, J. R., Diaz, M. E., Ramirez, D., Blazquez, F., & Platero, C. (2011). A learning through play approach to the development and assessment of general competences in electrical engineering based on a student competition. *International Journal of Engineering Education*, *27*, 831–837.

Voronyuk, E. (2011). *GreenTea's 2nd place entry postmortem translated by Hand*. Technical Report. Retrieved Nov 7, 2013 from http://trevoroakes.com/blog/2011/12/23/greenteas-2nd-place-entry-postmortem-translated-by-hand/

Whitman, L. E., & Witherspoon, T. L. (2003). Using LEGOS to interest high school students and improve K12 stem education. *33rd ASEE/IEEE Frontiers in Education Conference*, pp. F3A6–F3A10.

Williams, K. C., & Williams, C. C. (2011). Five key ingredients for improving student motivation. *Research in Higher Education Journal*, *12*, 121–123.

Parte III

APÉNDICE

NOTES