

EVOL-PLAYMAC Integración de una Técnica de Computación Evolutiva al Aprendizaje Computacional de un Juego de Tablero entre el Humano y la Máquina

Wilmer Alberto Gil Moreno, Oscar Ortega Lobo y Jorge Orrego Gaviria

UNIVERSIDAD DE ANTIOQUIA, Medellín

wagnm001@hotmail.com ; oortega@udea.edu.co ; jorrego@udea.edu.co

Recibido para revisión May-2006, aceptado Jun-2006, versión final recibida Jun-2006

Resumen: En el campo de la Inteligencia Artificial se han propuesto técnicas en tres tipos básicos de aprendizaje: el aprendizaje supervisado, el aprendizaje no supervisado y el aprendizaje por refuerzo. Las implementaciones que se hacen de estas técnicas están enfocadas en su mayoría a un aprendizaje computacional que poca interacción tiene con el humano; es un aprendizaje que se compara con experiencias anteriores pero no se logra a partir de la propia interacción máquina-humano. Evol-PlayMac surge, entonces, como un sistema que trata en lo posible de solucionar este problema al tratarse de un desarrollo que tiene implementada una técnica de aprendizaje del área de la computación evolutiva, que permite que a través de un juego tradicional de tablero, el parchís, la máquina formule estrategias de juego que cada vez demuestren un mejor desempeño obteniendo un nivel de ganancia mayor y logrando esto a partir de la competencia contra un humano.

1 INTRODUCCIÓN

En la actualidad existen investigadores dedicados a comprender algunas funciones de los humanos, lo que ha permitido el desarrollo de diferentes disciplinas o campos de estudio, como es el caso de la Inteligencia Artificial. La Inteligencia Artificial procura simular comportamientos mentales de procesos de razonamiento, pensamiento, aprendizaje y accionar de los humanos para generar sistemas inteligentes, capaces de recrear tales procesos en las máquinas. Entre los investigadores que desarrollan estos sistemas, se encuentran algunos que se especializan en el área de aprendizaje de máquinas, desarrollando diferentes técnicas de aprendizaje computacional, algunas a partir de enfoques probabilísticos y otras a partir del estudio del ser humano.

Tales técnicas deben ser evaluadas para determinar si éstas realmente simulan el aprendizaje de los humanos o generan características que permitan identificar logros asociados con el aprendizaje. No obstante los métodos desarrollados para la evaluación de dichas técnicas son teóricos o están desarrollados para evaluar una determinada técnica en un ambiente concreto, lo que deja a un lado la interacción máquina-humano, impidiendo que los investigadores comprueben sus técnicas en situaciones reales.

La computación evolutiva ofrece gran variedad de

técnicas cuyas aplicaciones prácticas se diversifican por diferentes temáticas, especialmente aquellas que buscan el óptimo en la solución de un problema. El enfoque que se le ha dado a esta área permite plantear igualmente técnicas enfocadas al aprendizaje computacional, especialmente en la teoría de juegos: aprendizaje de reglas, determinación de estrategias, selección de individuos (jugadores) competentes, etc.

En el presente artículo se pretende analizar un poco estas técnicas de computación evolutiva con el fin de observar las ventajas y desventajas que ofrecen en la solución de un juego y su posible integración en un sistema multiagentes con el fin de establecer una técnica apta para su implementación, cuyo aprendizaje pueda lograrse en parte con la interacción con un humano y un entrenamiento sistematizado y su desempeño pueda ser posteriormente evaluado a partir del juego entre el humano y la máquina, con el fin de aportar en algún grado a la solución del problema de la evaluación de técnicas de aprendizaje computacional en situaciones reales.

2 DEFINICIÓN DEL PROBLEMA

Las técnicas de aprendizaje de máquinas por lo general se evalúan sin tener en cuenta la interacción máquina-humano, lo que no permite obtener un resultado medible del comportamiento de un sistema donde el aprendizaje

se origina a partir de dicha interacción, lo que aleja a los investigadores de la comprobación de sus técnicas en situaciones reales. Sin embargo, se han desarrollado sistemas que de alguna manera tratan de "emular" el aprendizaje computacional a partir del conocimiento humano, estableciendo técnicas comparativas que hacen que los resultados obtenidos a través del aprendizaje computacional sean confrontados con bases de conocimiento previamente alimentadas con experiencias humanas similares.

En este sentido han aparecido técnicas basadas en "Sistemas Expertos" [Compton (1995)] en donde el aprendizaje obtenido por la máquina es ingresado en un Sistema Experto previamente alimentado con experiencias anteriores de situaciones similares afrontadas por un humano, y dicho sistema arrojaría como resultado de su proceso la evaluación de la técnica. En esta misma línea se desarrolló tiempo después los "Sistemas basados en conocimiento" [Compton, Preston y Kang (2000)] con el mismo propósito, sólo que la evaluación ya no es realizada por un Sistema Experto sino por un Sistema Basado en Conocimiento (KBS) que posee una colección de parámetros introducidos por la experiencia humana y, a partir de un proceso de comparación, realiza la evaluación. Como una mejora a este tipo de solución, se desarrollaron sistemas que usan métodos de "Evidencia Positiva" [Schwab, Wolfgang y Koychev (2000)].

Dichos Sistemas usan ya no una colección sino una selección de características del comportamiento humano (llamadas "evidencia positiva" dado que son las características que deben cumplirse para asegurar el aprendizaje) y a partir de la comparación se observan sólo los aciertos entre los resultados arrojados por el aprendizaje computacional y las coincidencias con las "evidencias positivas" para así definir la evaluación de la técnica.

Sin embargo, esta evaluación es de alguna manera asincrónica y la interacción máquina-humano no es directa. Para solventar esto, se ha planteado un juego de tablero basado en el Parqués tradicional que permita el juego entre el humano y la máquina de forma tal que teniendo esta última implementada una técnica de la computación evolutiva se genere aprendizaje y pueda ser directamente evaluada en situaciones "reales" y no "ideales" como las generadas computacionalmente.

Lo que se ha realizado y se expresa en este artículo es la experiencia y los resultados obtenidos tras integrar una técnica del área de computación evolutiva al aprendizaje computacional a partir de un juego de tablero entre un humano y una máquina, ofreciendo un Sistema Multiagente capacitado para integrar diversas técnicas de aprendizaje y evaluar su desempeño, aportando a la solución del problema de evaluar técnicas de aprendizaje computacional en situaciones surgidas a partir de la interacción máquina-humano.

3 MÉTODO DE SOLUCIÓN

En el área del aprendizaje computacional se han establecido tres tipos básicos de estrategias o modos de aprendizaje: el aprendizaje supervisado (supervised learning), el aprendizaje no supervisado (unsupervised learning) y el aprendizaje por refuerzo (reinforcement learning). Cada una de estas estrategias o modos de aprendizaje definen diversas técnicas y modos de implementación que reúnen los aspectos más representativos para así lograr nuevas propuestas que combinan tanto la computación evolutiva como el uso de redes neuronales y técnicas bayesianas y estocásticas [Rumanson y Sigurdsson (2003)].

De estos tipos de aprendizaje, el aprendizaje por refuerzo es el que mayor aplicación tiene, por el momento, en la teoría de juegos dado que el conocimiento se adquiere en la medida que se observan y se asimilan situaciones anteriores [Sutton y Barto (2002)]. El aprendizaje por refuerzo es una forma de aprender a partir de aciertos y errores, y de recompensas y castigos [Russell y Norvig (1995)].

Además, un gran número de técnicas de este enfoque han sido previamente ensayadas y corregidas para juegos de tablero [Ghory (2001)] lo que representa un gran avance para la selección de las técnicas que se integrarán en el juego de tablero que se desea desarrollar. El aprendizaje por refuerzo se desprende en parte del aprendizaje supervisado [Barto y Dietterich (2003)], pero el conocimiento se va adquiriendo a través de ensayos y errores que se ven recompensados según el caso en la medida en que los resultados van convirtiéndose en resultados deseados. Su mayor campo de aplicación es en aquellos sistemas en que debe mejorarse el comportamiento para tener un mayor rendimiento, automatizar comportamientos o mejorar (buscar) las estrategias de juego [Kaelbling, Littman y Moore (1996)].

En aras de implementar una técnica de aprendizaje que permitiese que el conocimiento fuese adquiriéndose en el medida que se interactúa con el humano se investigó y seleccionó una técnica del área de la computación evolutiva entre un gran grupo de técnicas enfocadas en estos modos de aprendizaje. ¿Y por qué una técnica de computación evolutiva? Las técnicas de este campo de la informática o de los sistemas ha adquirido un gran avance en los últimos tiempos, de forma que en la actualidad se usa en muchos aspectos y su aplicación en los juegos es muy valiosa. Por ejemplo, se han usado algoritmos genéticos para el reconocimiento de reglas para el campeonato de fútbol de robots "Robocup" [Zagal y Ruiz del Solar (2003)], al igual que la implementación de técnicas adaptativas en sistemas multiagentes [Smith y Taylor (2003)] que adaptan a los "agentes" al ambiente en el que se encuentran inmersos (entiéndase por "agente" a un autómata computacional con caracterís-

ticas de autonomía y proactividad, que buscan o están orientados al cumplimiento de una meta u objetivo, que puede al tiempo estar inmerso en un ambiente con otros agentes y tiene la capacidad de comunicarse con estos; el conjunto de agentes iterando en su mismo ambiente es lo que se conoce como "Sistema Multiagente"). También se ha usado la computación evolutiva en otros sistemas multiagente para el aprendizaje de las reglas y la iteración con otros agentes, detectando a los enemigos y aprendiendo actuar en diversas situaciones; esto último hace parte de técnicas para agentes dinámicos distribuidos en la distinción de ambientes [Van Dyke, von Brueckner y Matthews (2000)].

Además la mayoría de técnicas de la computación evolutiva aplicadas al aprendizaje computacional pueden aplicar como técnicas de aprendizaje por refuerzo dado que su avance se logra a través de la evolución que puede plantearse como una serie de ensayos y errores de generación en generación obteniendo cada vez individuos más aptos.

El sistema (entiéndase por sistema la integración del juego de tablero, la técnica de aprendizaje computacional y la posibilidad de iterar con un humano) fue pensado, diseñado y desarrollado como un sistema multiagente para permitir la escalabilidad posterior del mismo al tiempo que se permite el trabajo modular de aspectos importantes y sincronizados característicos de los juegos.

La decisión de optar por un sistema multiagente se obtuvo tras establecer y estudiar los requerimientos del sistema, entre los que se cuenta que éste debía permitir el juego entre un humano y una máquina, al tiempo que debía tener un pleno control sobre el desarrollo del juego para evitar y anunciar movimientos o estados indebidos, proporcionar la información necesaria para la toma de decisiones tanto de la máquina como del humano, identificar los estados del juego y permitir el aprendizaje del juego, computacionalmente hablando, para poder evaluar posteriormente dicho aprendizaje.

El diseño de dicho sistema, que por lo mencionado se entenderá como un Sistema Multiagente, fue realizado usando el método de desarrollo MASE (*MultiAgent System Engineering*) [Deloach, Wood y Sparkman (2001)] que cubre las fases de análisis y diseño y orienta las primeras etapas de desarrollo apoyado en la herramienta AgentTool. El lenguaje de programación utilizado fue Java, lo que se ajustó a los requerimientos de portabilidad y uso de la programación orientada a objetos sobre la que se basaba el método de desarrollo. La selección se realizó tras el estudio de varios métodos de Sistemas Multiagentes, entre los que se cita GAIA, MassCommandKads y Message.

El uso de este método y el hecho de haber seguido el desarrollo pensado como un sistema multiagente, permitió independizar los módulos de desarrollo del sistema y tratar algunos inconvenientes surgidos (como por ejem-

plo la interfaz de usuario y la técnica de la computación evolutiva a implementar) de forma independiente sin que lo uno interviniere en lo otro. Por otro lado se ha obtenido un sistema capaz de ser complementado o enriquecido con otras técnicas que podrían tratarse como nuevos agentes, que entrarían a participar dentro del mismo sistema haciendo que el proyecto pueda en un futuro lograr evaluar ya no sólo una sino varias técnicas de aprendizaje computacional.

La técnica implementada sigue los parámetros de la "Programación Evolutiva", una de las áreas de la computación evolutiva. Esto se definió dadas las características del juego y del sistema: el juego se desarrolla por estados y puede representarse con una máquina de estados finitos (recuérdese que es una variante del juego del parchís tradicional, adaptado para dos jugadores y con un tablero de juego reducido, conservando el aspecto del "azar" con el uso de dos dados) y el aprendizaje se daría a partir del ensayo y el error ("aprendizaje por refuerzo").

La técnica utiliza una representación de vectores con valores reales y usa la mutación para variar los individuos representados como máquinas de estado finito:

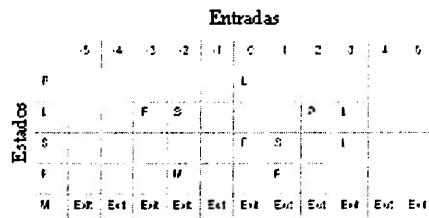


Figura 1: Ejemplo de máquina de estados que representa a cada Individuo (versión simplificada)

Las transiciones y las salidas ya están predefinidas y son inmodificables en el sistema; la técnica se encarga de realizar las transformaciones necesarias en las entradas, representadas como vectores y cuyos valores enteros varían desde -5 hasta 5 calculados a partir de funciones de ganancia:

g_i : Ganancia obtenida en el turno i del juego, que para el caso del equipo 1 (máquina) sería:

$$g_i = posF_{i1}E_1(x_1) + posF_{i2}E_1(x_2) - posF_{i1}E_2 - posF_{i2}E_2$$

donde: $posF_{i\#}$ es la posición de la ficha $\#$ al hacer la jugada, $E\#$ es el número del Equipo (1 ó 2) y $x\#\#$ es la probabilidad de capturar la ficha 1 ó la ficha 2.

Al terminar un juego la ganancia total se calcula por la fórmula:

$$G = \sum i = 1, \dots, N, \quad 1 \in Z$$

donde, N es el número de turnos del jugador en todo el juego.

El individuo con mayor ganancia se considera entre los más aptos y por ende se conserva a través del almacenamiento de su máquina de estados.

4 DISEÑO EXPERIMENTAL

Con la implementación del sistema se buscó obtener un avance en la ganancia lograda por la máquina y conseguir un nivel más alto de juegos ganados por la máquina en sus enfrentamientos directos contra un humano. De allí que se haya planteado como hipótesis de estudio que la máquina logrará un nivel de juegos ganados similar al del jugador humano a través de la repetición de varias series de juego contra dicho jugador.

La implementación de la técnica en el sistema se hizo asignándole a un agente, llamado el "agente jugador", la capacidad de identificar los diferentes estados del juego (que son informados constantemente por otro agente llamado "agente controlador" y cuyas instrucciones son traducidas al humano por medio de un "agente de interfaz") y a partir de estos él evalúa sus posibles acciones, calcula la ganancia esperada por cada una de sus posibles decisiones (como cada jugador cuenta con dos fichas, la decisión a tomar es qué ficha se va a jugar con qué dado, pues también son dos dados) y a partir de la representación del individuo define el salto de estado a dar. Al final del juego se compara la ganancia final obtenida, se realiza o se genera un nuevo individuo realizando mutaciones y se simula de nuevo el juego.

El entrenamiento de la máquina se logró haciendo que ésta repitiera automáticamente el mismo juego diez veces lo que permitió hacer una comparación más acertada del desempeño obtenido por cada individuo tras su mutación. A este entrenamiento del mismo juego durante diez veces se le denomina "serie".

La prueba por serie de juegos se realiza con un conjunto de 5 individuos. Es decir, cinco individuos diferentes se someten al mismo juego contra el humano y se repite nuevamente este juego por nueve ocasiones.

Los resultados favorables o que producen mayor ganancia han permanecido como individuos sobrevivientes y que siguen poco a poco evolucionando; si la descendencia no es favorable se descarta esa nueva serie de individuos.

El resultado obtenido tras la primera serie de 10 juegos se muestra en la Figura 2. Allí se observan los resultados obtenidos por el "individuo estocástico" (agente que usa una técnica de ganancia calculada de manera probabilística), el mejor individuo evolutivo de los 5 probados y el resultado promedio de los 5 individuos evolutivos.

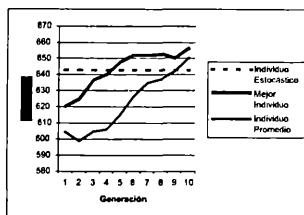


Figura 2: Resultado de la primera serie de Juegos

Los sobrevivientes de una serie de 10 juegos son analizados y evolucionados nuevamente en una nueva secuencia del juego (variación en los resultados de los dados generados tras una partida contra un humano) obteniendo nuevos individuos y descartando algunos que habían permanecido en juegos anteriores.

La evaluación se realiza a partir del alcance de la función objetivo que es maximizar la ganancia final.

5 RESULTADOS

Inicialmente los individuos eran comparados con resultados previamente obtenidos usando una técnica estocástica que definía la jugada a seguir a partir del cálculo de la mayor ganancia esperada, obteniendo resultados desfavorables pero que poco a poco fueron alcanzando la media. Sin embargo, el número de juegos que tuvo que realizarse para esto correspondió a una serie de 5 generaciones para el primer individuo que la alcanzó (una generación permanece en cada serie de juego y evoluciona para la serie siguiente), aunque se obtuvo un promedio de 10 generaciones para los demás individuos. Luego el avance obtenido siguió siendo significativo hasta la 35 generación y de allí en adelante comenzó a mantener un nivel de estabilidad, aunque tuvo unas leves caídas al cambiar la secuencia de los juegos cada 10 generaciones (ver Figura 3). Sin embargo, se ha establecido que los individuos resultantes han definido unas máquinas de estados que plantean estrategias de juego competitivas.

Resultados Generales

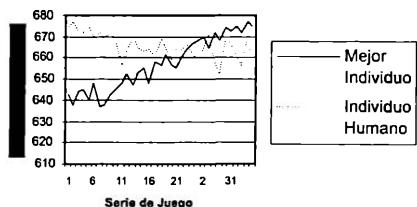


Figura 3: Resultados promedio de las 35 primeras series de Juego

La Figura 4 detalla el número promedio de juegos ganados tanto por la máquina como por el jugador humano en las primeras 10 series o generaciones de juego, también denominadas "Épocas". Es un resultado promedio por serie ya que se tomó el número de juegos ganados por cada uno en cada serie y se dividió por 10 que es el número de repeticiones por serie:

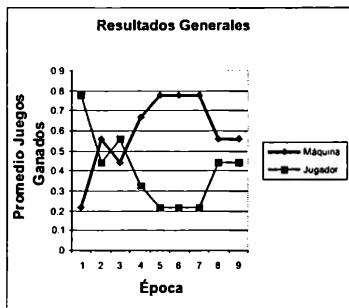


Figura 4: Resultados promedio de juegos ganados en 9 épocas consecutivas de juego

6 CONCLUSIONES

La elección del desarrollo de un sistema multiagente para dar solución al problema de evaluar técnicas de aprendizaje computacional resulta en cierto modo acertada dado que hace que un sistema desarrollado pueda en algún momento ser probado con diversas técnicas, dada su flexibilidad y escalabilidad. En el sistema desarrollado se ha implementado una técnica basada en la computación evolutiva, propia del área de la programación evolutiva, que se adapta a las características del juego gracias a la representación del mismo como una máquina de estados, y ha mostrado un avance o progreso en el aprendizaje logrado aunque cada vez que se genera una nueva secuencia de juegos (variación en los resultados predefinidos de los dados) hay una leve desmejora de los resultados previos pero rápidamente se logra una recuperación de los mismos en cada generación.

El hecho de permitir la interacción máquina - humano e introducir el elemento de azar ha hecho que el aprendizaje se desarrolle de alguna manera bajo condiciones "reales de juego", lo que garantiza que el aprendizaje o avance obtenido con la técnica implementada en las estrategias de juego sea confiable y, de alguna manera, comparable con el aprendizaje humano que constantemente se enfrenta a situaciones similares.

REFERENCIAS

- Barto, A. y Dietterich, T. (2003), 'Reinforcement learning and its relationship to supervised learning', En línea: <<http://web.engr.oregonstate.edu/~tgd/publications/Barto-Dietterich-03.pdf>> C11/05.
- Compton, P. (1995), Simulating expertise, Technical report, School of Computer Science and Engineering, University of New South Wales, Sydney-Australia.
- Compton, P., Preston, P. y Kang, B. (2000), The use of simulated expert in evaluating knowledge acquisition, Technical report, School of Computer Science and Engineering, University of New South Wales, Sydney-Australia.
- Deloach, S., Wood, M. y Sparkman, C. (2001), 'Multiagent system engineering', *International Journal of Software Engineering and Knowledge Engineering* 11(3), 231-258.
- Ghory, I. (2004), Reinforcement learning in board games, Technical report, University of Bristol.
- Kaelbling, L., Littman, M. y Moore, A. (1996), 'Reinforcement learning: A survey', En línea: <<http://www.cs.cmu.edu/afs/cs/project/jair/pub/volume4/kaelbling96a.pdf>> C11/05.
- Runarson, T. y Sigurdsson, S. (2003), 'The learning methodology', En línea: <<http://cerium.raunvis.hi.is/~tpr/courseware/svm/notes/chapter1.pdf>> C11/05.
- Russell, S. y Norvig, P. (1995), *Inteligencia Artificial, un enfoque moderno*, Trad. por R.B.G., Prentice Hall Hispanoamericana, México.
- Schwab, I., Wolfgang, P. y Koychev, I. (2000), *Learning to Recommend from Positive Evidence*, GMD-FIT.MMK, Sankt Augustin-Germany.
- Smith, R. y Taylor, N. (2003), A framework for evolutionary computation in agent-based system, Technical report, The ICSC, University the West of England Bristol, UK.
- Sutton, R. y Barto, A. (2002), *Reinforcement Learning*, The MIT Press, London.
- Van Dyke, H., van Brueckner, J. y Matthews, R. (2000). *Distinguishing Environmental and Agent Dynamics: A Case Study in Abstraction and Alternate Modeling Technologies*, ERIM, USA.
- Zagal, J. y Ruiz del Solar, J. (2003), Adaptación evolutiva de un sistema visual de reconocimiento de objetos para el campeonato de fútbol robótico nobocup, Technical report, Departamento de ingeniería Eléctrica, Universidad de Chile.

