

Algoritmos Evolutivos en la Optimización de Funciones de Evaluación del Juego Mastergoal

Alberto Samaniego¹ y Benjamín Barán^{1,2}

¹Universidad Católica “Ntra. Sra. de la Asunción” - Paraguay

^{1,2}Universidad Nacional de Asunción - Paraguay

alberto_samaniego@uca.edu.py, bbaran@cba.com.py

Resumen. Encontrar una estrategia óptima para un juego puede ser visto como un problema de búsqueda donde, dado el conjunto de todas las estrategias posibles, se trata de encontrar una que venza a todas las demás sin necesidad de explorar exhaustivamente el espacio de soluciones. Los algoritmos evolutivos han probado ser exitosos en este tipo de problemas. En este trabajo se utilizan algoritmos evolutivos y técnicas coevolutivas para ajustar los pesos de las características de la función de evaluación del juego Mastergoal.

Palabras clave: juego de tablero, función de evaluación, algoritmo evolutivo, coevolución.

1. Introducción

Ser un buen jugador de *Ajedrez* u otro juego de tablero similar generalmente es visto como signo de inteligencia y prestigio intelectual. Anteriormente se creía que ésta era una característica exclusiva del ser humano. Sin embargo, con el surgimiento de los computadores, y especialmente con el nacimiento de la Inteligencia Artificial (*IA*), esta creencia se fue dejando de lado.

Los juegos de tablero y otros juegos abstractos son interesantes para su estudio ya que sus dominios son mundos simplificados de problemas de la vida real, los cuales a menudo son difíciles de representar en un computador. Los estados de los juegos de tablero son fáciles de representar, y los jugadores están restringidos, por lo general, a un reducido número de acciones cuyos resultados están definidos por reglas bien precisas [1].

En este trabajo se examina la utilización de los algoritmos evolutivos en el contexto del juego *Mastergoal* [2] y se reportan los resultados obtenidos al aplicar el paradigma coevolutivo para ajustar los pesos de las características de la función de evaluación en los niveles de juego 1 y 3. El *Mastergoal* es un juego de tablero inspirado en el fútbol, similar al *Ajedrez* y las *Damas*, pues consta de casillas y fichas que se mueven dentro de ellas.

2. Mastergoal

Mastergoal [2] es un juego de estrategia inspirado en el fútbol. Es un juego dinámico, de suma cero, e información perfecta y completa. Consiste en dos equipos que están compuestos por fichas que representan a sus jugadores, y una ficha que representa a la pelota. El objetivo de cada equipo es lograr introducir la pelota en el arco del equipo contrario, en otras palabras, marcar un gol. Las partidas se juegan a dos goles, pero para los propósitos de este trabajo, se considera que gana el juego el jugador que marca un gol. Además, si luego de un número máximo de jugadas no se ha marcado un gol, el juego termina en empate.

El juego se desarrolla en un tablero de 13x11 casillas que representa una cancha (más 10 casillas que representan a los arcos). Cada pieza puede moverse en dirección horizontal, vertical o diagonal. Los jugadores pueden moverse 1 ó 2 casillas. La pelota puede moverse de 1 a 4 casillas. La misma, sólo puede moverse cuando hay una cantidad superior de jugadores del equipo de turno alrededor de ella. En ese caso, se dice que dicho equipo está en posesión de la pelota. Cuando se presenta esta configuración, y solamente en este caso, la pelota debe ser movida por el equipo de turno.

La partida transcurre alternando los turnos entre los equipos. En cada turno, el equipo que juega debe mover un jugador una vez y, a continuación, la pelota, en caso que quede en posesión de ella. La pelota puede ser movida hasta cuatro veces en un turno. Al final de cada turno la pelota no debe quedar en posesión de ninguno de los equipos.

El *Mastergoal* consta de tres niveles de complejidad creciente, además de algunas variantes dentro de los mismos niveles. Los niveles se diferencian principalmente por la cantidad de fichas por equipo, y la complejidad de las reglas. En el nivel uno los equipos están compuestos por una sola ficha, en el nivel dos cada equipo tiene dos fichas, mientras que en el nivel tres hay cinco fichas por cada equipo, siendo una de ellas un jugador especial que representa al arquero. Una variante del nivel tres (nivel tres "2 + 1") consiste en las mismas reglas, pero con sólo tres fichas por equipo (una de ellas, el arquero).

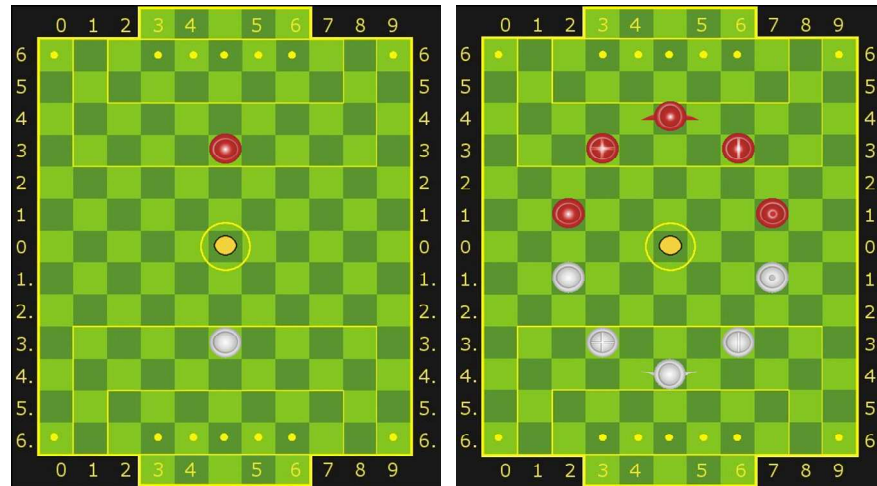


Figura 1. Configuración inicial del tablero de Mastergoal en nivel 1 (izquierda) y nivel 3 (derecha).

La complejidad del juego *Mastergoal* y el hecho de que prácticamente no existen trabajos previos de investigación sobre el mismo en el campo de la *IA*, hacen del mismo un candidato ideal como plataforma de prueba para distintas técnicas de *IA*. La tabla 1 muestra una comparativa de la complejidad del *Mastergoal*, en sus distintos niveles, con algunos de los juegos de tablero más populares. Nótese que el *Mastergoal*, a partir del nivel 2, ya supera en complejidad al *Ajedrez* y está sólo por debajo del *Go* [1], que continúa siendo el gran desafío para los investigadores de *IA*. Cabe destacar que, en ciertas ocasiones, el número de jugadas posibles para un estado en *Mastergoal* (en los niveles “2 + 1” y 3) supera ampliamente al promedio del *Go*, llegando incluso a más de 1000 jugadas.

Tabla 1. Comparación de la complejidad del Mastergoal con otros juegos.

Juego	Tamaño del Tablero	Factor de Ramificación Promedio
Tatetí	9	5
Othello	64	8
MGoal 1	153	20
Ajedrez	64	36
MGoal 2	153	46
MGoal “2+1”	153	78
MGoal 3	153	132
Go	361	250

3. Funciones de Evaluación

Una *función de evaluación* es una función que recibe un estado y devuelve un valor numérico indicando la utilidad esperada del mismo para un jugador [1]. Utilizando esta función, el jugador de turno evaluaría cada uno de los sucesores del estado actual, y elegiría la acción que lleve al estado con mayor valor.

Una función de evaluación E se dice que es perfecta dados dos estados s y s' , $E(s) > E(s')$ si y sólo si la utilidad obtenida si se juega de manera perfecta hasta el final del juego partiendo del estado s , es mayor que la obtenida si se juega a partir de s' .

Algunos aspectos importantes a tener en cuenta para diseñar una buena función de evaluación son [1]:

- La función de evaluación debería ordenar los estados terminales del mismo modo que la función de utilidad verdadera.
- El cálculo del valor de un estado no debe llevar demasiado tiempo.
- Para estados no terminales, la función de evaluación debería estar fuertemente relacionada con las posibilidades de ganar.

La mayoría de las funciones de evaluación calculan el valor de un estado s basándose en el valor numérico $f_i(s)$ de ciertas características del mismo [1, 3]. Por ejemplo, el valor material de las piezas en el tablero, o el número de peones capturados en el *Ajedrez*. Algunas de las características más importantes utilizadas en la función de evaluación del *Mastergoal* son las que se muestran en la tabla 2.

Tabla 2. Características más importantes de la función de evaluación del Mastergoal.

Característica	Descripción	Característica	Descripción
avance_balon	Adelantamiento de la pelota.	dist_balon_op	Distancia de la pelota al oponente.
dist_balon_jug	Distancia de la pelota al jugador de turno.	avance_op	Adelantamiento del oponente.
avance_jug	Adelantamiento del jugador de turno.	dist_col_balon_op	Distancia del oponente a la columna de la pelota.
dist_col_balon_jug	Distancia del jugador de turno a la columna de la pelota.	dif_dist_balon	Diferencia entre la distancias del oponente y del jugador a la pelota.

A cada característica f_i se puede asociar un peso w_i de manera a que el resultado de la función de evaluación sea una suma ponderada de todas las características. Una función de evaluación con n características podría expresarse entonces como:

$$E(s) = \sum_{i=1}^n w_i * f_i(s) \quad (1)$$

Una de las estrategias más utilizadas en los agentes de juegos de tablero es la estrategia *minimax* combinada con una función de evaluación [1, 4]. Éste es el enfoque adoptado en este trabajo. Se utiliza una variante del algoritmo minimax junto con una función de evaluación para hacer jugar *Mastergoal* al computador.

Sin embargo, el problema central radica en encontrar la relación óptima entre las características de la función de evaluación de manera a que ésta se aproxime lo máximo posible a una función de evaluación perfecta. Surge entonces la pregunta: ¿cómo se podría hacer para encontrar el valor óptimo de los pesos de dichas características? Es ahí donde entran en juego los algoritmos evolutivos. Hallar el valor óptimo de los pesos de la función de evaluación del juego *Mastergoal* podría ser también considerado como un problema de búsqueda donde, dado el conjunto de todas las combinaciones posibles de los valores de los pesos, se quiere encontrar una combinación tal que la función de evaluación con dichos pesos venza a todas las demás (o al menos que no pierda), sin necesidad de explorar exhaustivamente el espacio de búsqueda. Para lograr dicho propósito se sigue el *paradigma coevolutivo* [4, 6].

4. Coevolución

4.1 Paradigma Coevolutivo

La coevolución es el proceso por el cual dos o más especies u organismos ejercen presión de selección mutua, que resulta en adaptaciones específicas recíprocas. En la evolución natural, la aptitud (o *fitness*) de un individuo es determinada con respecto a sus competidores y colaboradores, así como al medio ambiente.

Haciendo una analogía entre los juegos y la evolución natural surgen interesantes posibilidades en aplicar técnicas evolutivas a los mismos [4]. La evolución natural puede ser vista como una competencia donde las recompensas de los organismos más aptos, que se desempeñan bien en el juego de la vida, son la supervivencia y la propagación de su material genético a sus descendientes [5].

Se podría considerar entonces a cada función de evaluación como un individuo, cuyo fitness sería medido mediante la competencia contra otros individuos de la misma u otra población. Los mejores individuos sobrevivirían y generarían nuevos individuos que reemplazarían a los de fitness más bajo. De esta forma, la población iría evolucionando, obteniéndose cada vez mejores soluciones. Utilizando este paradigma se han obtenido estrategias expertas de juegos sin necesidad de la intervención humana [5].

5. Mastergoal como un Problema Coevolutivo

5.1 Aspectos a Tener en Cuenta

La coevolución, en su forma más simple no siempre produce los resultados deseados. Una implementación efectiva de la coevolución competitiva debe resultar en un progreso continuo en la aptitud de los individuos. Para ello, los nuevos individuos que surgen deben ser capaces de vencer a todos los anteriores. Sin tal garantía, el proceso puede quedarse estancado en un ciclo de estrategias pobres (óptimos locales) que se

vencen entre sí. Para evitar este problema, se utilizó la técnica denominada *Salón de la Fama* (o *Hall of Fame*) [6], que consiste en mantener un historial de los mejores individuos de cada generación, de manera a que puedan ser utilizados para evaluar a los nuevos individuos. De esta forma, para asignar el fitness a un individuo, éste debe competir no sólo contra los oponentes actuales, sino también contra algunos oponentes del salón de la fama.

5.2 Aspectos de la Implementación

El programa utilizado en este trabajo fue codificado enteramente en C++, utilizando la librería de algoritmos genéticos GALib¹ [7], la cual provee de clases y métodos estándares para la representación de los individuos y operadores de selección, cruzamiento y mutación.

5.3 Representación de los Individuos

Se decidió representar a los cromosomas con un vector de números reales, donde cada elemento corresponde al peso de una característica determinada. Los valores de dichos pesos están comprendidos dentro de un rango [-100; 100]. La elección de esta representación se debió a que los únicos elementos que varían en los individuos con la evolución son los pesos, mientras que las características utilizadas son siempre las mismas. Para la función de evaluación del nivel 1 se utilizaron ocho características, mientras que para la función de evaluación del nivel 3 se utilizó un total de veinticuatro.

$$C \equiv [w_1, w_2, \dots, w_n]$$

Figura 2. Cromosoma como un vector de n números reales.

5.4 Función de Fitness

Para evaluar el fitness de los individuos se utilizó un *torneo muestral* [6], donde cada individuo de la población actual (población de *huéspedes*) compite contra todos los individuos de una muestra más pequeña (población de *parásitos*).

El fitness de los individuos fue calculado utilizando la siguiente fórmula basada en el método *Competitive Fitness Sharing* [6]:

$$f(h) = \sum_{i=1}^k \frac{1}{N_{gXi}} + \sum_{i=1}^t \frac{e}{g^*(N_{gYi} + N_{eYi})} \quad (2)$$

donde el huésped h vence a los parásitos $p_{X1}, p_{X2}, \dots, p_{Xk}$ y empata con los parásitos $p_{Y1}, p_{Y2}, \dots, p_{Yt}$. Por su parte, N_{gXi} es el número de huéspedes que vencen a p_{Xi} mientras

¹ Desarrollada por Matthew Wall en el Instituto Tecnológico de Massachussets.

que N_{gY_i} y N_{eY_i} son el número de huéspedes que vencen y empatan con el parásito p_{Y_i} , respectivamente. Además, por cada partida ganada un huésped recibe g puntos, y por cada empate e puntos (en este trabajo en particular se utilizó $g=3$ y $e=1$).

En cada iteración, la mitad de los individuos de la población de muestra (parásitos) eran seleccionados de la población anterior de huéspedes basándose el método de *Shared Sampling* [6], mientras que la otra mitad era escogida de manera aleatoria del salón de la fama.

Si bien, en la coevolución generalmente se tiene a dos poblaciones evolucionando simultáneamente, cuando la relación es simétrica (como en el caso del *Mastergoal*), en el sentido que el jugador que empieza no tiene ninguna ventaja sobre el otro ni viceversa, evolucionar una sola población es suficiente [4].

5.5 Operadores de Selección, Cruzamiento y Mutación

Los operadores de selección, cruzamiento y mutación utilizados fueron los provistos por la librería GALib. El operador de selección utilizado sigue el método *Roulette Wheel* [8], donde los individuos son seleccionados de manera proporcional a su fitness. El operador de cruzamiento es el de *Cruzamiento Uniforme (Uniform Crossover)* [8], donde cada elemento del nuevo vector, es heredado de uno de los padres con probabilidad 0,5. El operador de mutación cambia los valores de los genes con una probabilidad de 0,2; el gen mutado es reemplazado por un nuevo valor de acuerdo con una distribución gaussiana.

6. Resultados Experimentales

A continuación se detallan los resultados obtenidos. En ambos casos la profundidad de búsqueda del algoritmo minimax fue 1. Se utilizó elitismo, permitiendo pasar a la siguiente población sin ser alterados a los mejores 3 individuos de cada generación.

Las partidas se jugaron con la siguiente regla: ganaba la partida el primero en anotar un gol; y, si luego de un número máximo de movidas (40 para el nivel 1 y 60 para el nivel 3) no se marcaba ningún gol, el juego terminaba en empate. Luego de cada iteración, el mejor de la generación era guardado en el salón de la fama (si es que todavía no se encontraba allí). Al final de todas las iteraciones se realizó un torneo de todos contra todos entre los individuos del salón de la fama de manera a poder verificar el progreso de los mismos.

Para el nivel 1 se utilizó un tamaño de población de 1000 huéspedes y 100 parásitos, y se iteró un total de 200 generaciones. Para el nivel 3 se utilizó un tamaño de población de 200 huéspedes y 20 parásitos, y se iteró un total de 300 generaciones.

La figura 3 muestra la cantidad de partidas ganadas en el torneo de todos contra todos en los individuos de nivel 1. La figura 4 muestra el mismo progreso en los individuos de nivel 3. En ambos casos se puede notar que mejora el nivel de juego de los individuos a medida que transcurren las generaciones. Se observa un mayor aprendizaje hacia el comienzo de la evolución.

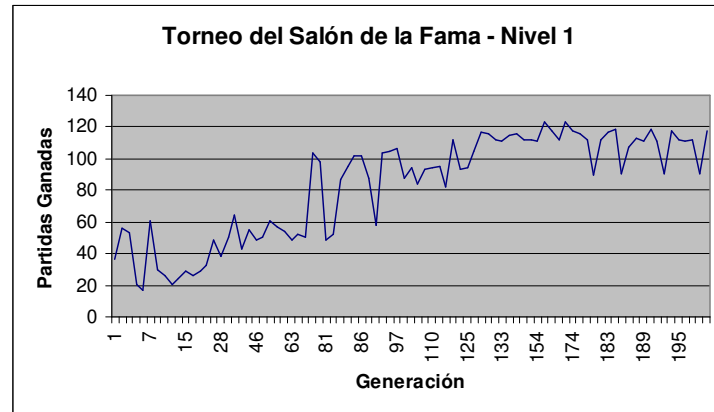


Figura 3. Resultados del torneo todos contra todos entre los individuos del salón de la fama en nivel 1.

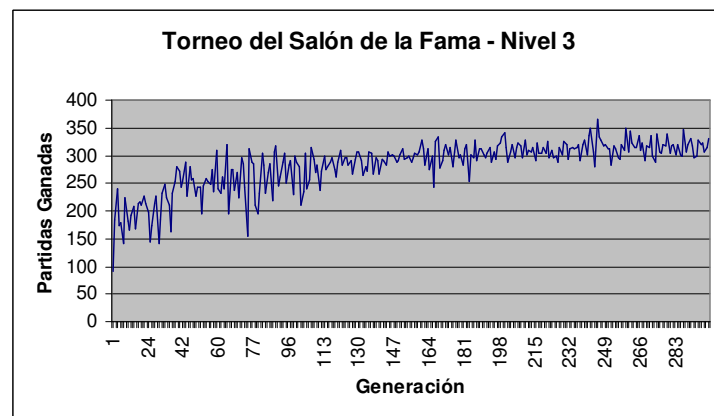


Figura 4. Resultados del torneo todos contra todos entre los individuos del salón de la fama en nivel 3.

Los mejores individuos fueron encontrados hacia las últimas generaciones. El mejor individuo del nivel 1 fue el de la generación 169, con 123 partidas ganadas, 12 empates y 33 partidas perdidas de un total de 168 partidas jugadas. El mejor del nivel 3 fue el de la generación 245, con 365 victorias, 4 empates, y 207 derrotas de un total de 576 partidas jugadas.

Otro análisis que se puede hacer es el de la relevancia de las características utilizadas en las funciones de evaluación. La figura 5 ilustra los valores de los pesos asignados a cada característica en los mejores individuos del nivel 1. Se puede ver que la característica con mayor relevancia es la distancia del jugador a la pelota (*dist_balon_jug*), la cual debe ser minimizada pues tiene un valor negativo. Le sigue el adelantamiento de la pelota (*avance_balon*), el cual debe ser maximizado pues el

valor del peso es positivo; esto resulta lógico ya que lo que se busca es marcar un gol. Otra de las características relevantes es el adelantamiento del oponente (*avance_op*) el cual debe ser minimizado. Se ve también que la estrategia da un peso positivo a la característica *dist_col_balón_jug*, indicando que prefiere no mantener al jugador en la misma columna de la pelota. Al mismo tiempo, la estrategia considera conveniente mantener la pelota alejada del oponente, pues *dist_balón_op* tiene un valor positivo. La mejor estrategia surgida en el nivel 3 dio mayor importancia a maximizar el adelantamiento de la pelota, minimizar la distancia de la pelota a los jugadores del equipo propio, y maximizar la distancia de la pelota a los jugadores del equipo contrario, en ese orden.

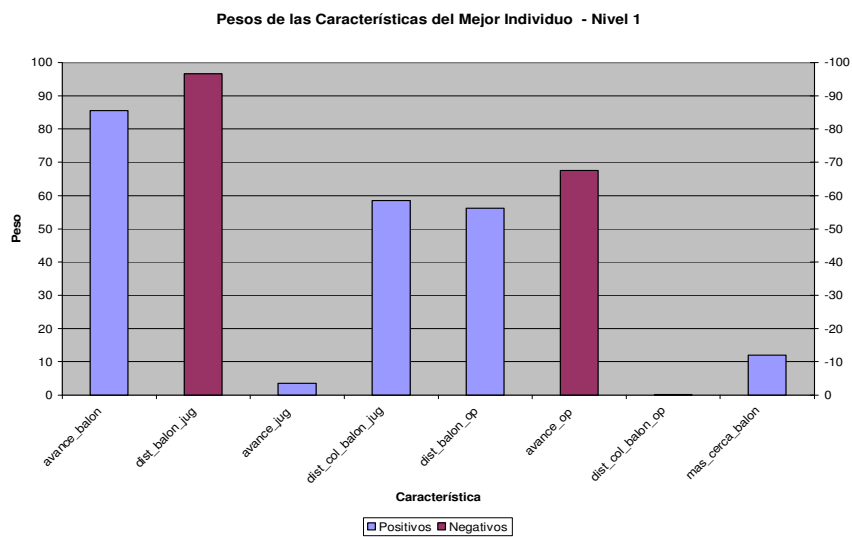


Figura 5. Valores de los pesos del mejor individuo del salón de la fama en nivel 1.

7. Conclusiones y Trabajos Futuros

El aporte principal de este trabajo fue utilizar al *Mastergoal* como plataforma de prueba para explorar la capacidad de los algoritmos evolutivos de encontrar buenas ponderaciones para las características de funciones de evaluación siguiendo el paradigma coevolutivo, al punto de resultar competitivo incluso al jugar con personas expertas en el juego. De hecho, al jugar con diferentes personas, se comprobó que el factor principal para determinar la competitividad de esta propuesta es la profundidad de la búsqueda al momento del juego y no el esquema de entrenamiento del programa implementado que resulta suficientemente bueno, aunque no se lo haya derivado con el mismo nivel de profundidad en las búsquedas.

En el caso ideal, la función de evaluación debería converger hasta llegar a ser perfecta. Sin embargo, se debe tener en cuenta que las presunciones hechas en el planteamiento del problema podrían limitar la evolución. Por ejemplo, la relación entre las

características de la función de evaluación óptima podría no ser lineal. O, se podrían estar omitiendo características importantes que sirvan para poder jugar de manera perfecta. Se puede suponer entonces, que la solución a la que llegarían los experimentos corridos en este trabajo luego de un número muy grande de iteraciones, sería el óptimo global, no necesariamente entre las todas soluciones posibles, pero sí en el subconjunto de las soluciones que sean combinaciones lineales de las características utilizadas. Si bien, las soluciones encontradas no son óptimas en el sentido que no hay una estrategia resultante que venza siempre a todas las demás, los resultados obtenidos son alentadores ya que muestran un claro progreso de los individuos hacia mejores soluciones.

Otra pregunta que surge es si las soluciones obtenidas al evolucionar los pesos con profundidad de búsqueda 1 en el algoritmo minimax, serían también buenas soluciones para las búsquedas minimax con mayor profundidad. Un trabajo futuro podría consistir en realizar comparativas entre las soluciones obtenidas con distintas profundidades de búsqueda en la evolución. Cabe destacar la necesidad de la paralelización del algoritmo evolutivo para poder correrlo con mayor profundidad de búsqueda minimax en un tiempo razonable (debido al gran factor de ramificación del *Mastergoal*).

El modelo coevolutivo es un paradigma emergente y tiene interesantes aplicaciones en el área de la teoría de juegos. Las técnicas aplicadas para desarrollar estrategias de juegos de *Mastergoal* podrían ser utilizadas de forma similar para encontrar estrategias a otros juegos, incluyendo a los que representen casos concretos del mundo real.

Por último, la utilización del *Mastergoal* en el estudio de la IA no se limitaría sólo a los algoritmos evolutivos, sino que también podría ser un instrumento útil para el estudio de varias técnicas de IA como la búsqueda selectiva, aprendizaje por refuerzo, redes neuronales y la construcción de libros de apertura, por citar algunas. Este trabajo pretende introducir al *Mastergoal* como herramienta de prueba en el campo de la IA, y abrir así nuevas puertas a futuras líneas de investigación.

Referencias

1. Russell, S. J.; Norvig, P. *Inteligencia Artificial. Un Enfoque Moderno*. Segunda Ed. Pearson Educación, S.A. Madrid, pp. 181--212, 2004.
2. Mastergoal, un Juego Colectivo de Estrategia. <http://www.mastergoal.com/>.
3. Smed, Jouni; Hakonen Harri. *Algorithms and Networking for Computer Games*. John Wiley & Sons. Chichester, UK, pp. 73--91 2006.
4. Anthony, Lisa. *Evolving Board Evaluation Functions for a Complex Strategy Game*. Master's thesis, Department of Computer Science, Drexel University. 2002.
5. Lucas, Simon M.; Kendall, Graham. *Evolutionary Computation and Games*. IEEE Computational Intelligence Magazine. Febrero, pp. 10--18 2006.
6. Rosin, Christopher Darrel. *Coevolutionary Search Among Adversaries*. PhD thesis, University of California, San Diego. 1997.
7. Wal, Matthew. *GALib: A C++ Library of Genetic Algorithm Components. Ver 2.4*. Mechanical Engineering Department, Massachusetts Institute of Technology. Agosto, 2006.
8. Mitchell, Melanie. *An Introduction to Genetic Algorithms*. The MIT Press, pp. 27--60, 116--132, 1998.