

Sistema de predicción de patrones de juego

Aram B. González y Jorge A. Ramírez Uresti

Departamento de Tecnologías de Información y Computación,
Campus Estado de México del Tecnológico de Monterrey,
Atizapán, Estado de México, México
{a00965707, juresti}@itesm.mx

Resumen Los sistemas multiagentes permiten representar ambientes en los que se requiere la interacción y cooperación de un conjunto de individuos. Mediante el modelado del oponente, se logra producir un concepto de los deseos del rival en el caso de ambientes competitivos. Se presenta el Sistema de Predicción de Patrones de Juego (SPPJ) que permite la predicción del movimiento del balón en posesión del rival basado en un análisis de los juegos previos de los equipos.

Palabras clave: sistemas multiagentes, razonamiento basado en casos, modelado del oponente.

1. Introducción

Se pueden definir a los agentes inteligentes como un sistema computacional que se encuentra en algún ambiente y que es capaz de llevar a cabo acciones autónomas dentro del mismo con el fin de lograr el objetivo para el que fue diseñado como indica [1]. Sin embargo, no existe una definición que abarque todas las características que un agente puede contener ya que pueden diferir en ciertas propiedades dependiendo del problema que atacan y de la implementación específica de su solución.

Una propiedad fundamental en el desarrollo de los sistemas que incorporan agentes es la interacción de los mismos con el medio en que se encuentran debido a que se espera que a cada acción realizada produzca una reacción en el mismo o una salida del sistema. Los agentes, en la mayoría de los casos, no tienen una vista global de su ambiente y tampoco ejercen un control total sobre él. Debido a las características de los sistemas multiagentes, uno de los campos de estudios en que son utilizados es la representación de ambientes competitivos en los cuales se enfrentan dos equipos. Un ejemplo es el uso del fútbol como ambiente. El objetivo fundamental del juego consiste en realizar más anotaciones que el oponente dentro de un campo de juego en base a una serie de reglas predefinidas.

En los ambientes competitivos, los equipos buscan maneras de lograr facilitar la realización de su objetivo primario. Cada equipo puede definir su manera de jugar, puede hacerlo en base a todos los elementos con los que cuenta o en base a un subconjunto de ellos. Los comportamientos derivados de este tipo de procesos

frecuentemente se pueden generalizar en patrones y jugadas que distintivamente se llevan a cabo a lo largo de los juegos.

Una de las mejores maneras para conseguir una ventaja sobre el rival, se basa en conocer de antemano las posibles opciones que frecuentemente utiliza para sobrellevar una situación en particular. Al conocer las acciones de un equipo, se puede determinar un plan que contrarreste sus medidas.

El resultado de predecir el comportamiento y los movimientos basados en un modelo en particular se le denomina Modelado del Oponente (MO). MO no especifica una técnica de aprendizaje automático en particular para llevar a cabo la creación del modelo por lo que los algoritmos y métodos quedan a criterio del investigador y normalmente se elige uno que se adapte a las condiciones del ambiente y del objetivo que se busca realizar.

MO puede ser creado en diversos ambientes aunque el modelo generado tiende a estar fuertemente relacionado con el entorno en el que se desarrolló debido a que generalmente se basa en los objetivos que se buscan conseguir.

En general, para crear un modelo del oponente se emplean técnicas de aprendizaje automático por lo que la cantidad de patrones puede variar pero la tendencia es partir de un número considerable de ejemplos. Se pueden emplear heurísticas para conseguir una base de entrenamiento que permita generalizar el modelo sin ocasionar un sobre-aprendizaje.

Varios estudios hacen uso de MO [2,3,4] para mostrar una mejoría entre un plan de juego básico y el que incorpora esta técnica, mientras que también se llega a su uso en la competencia de Coach de RoboCup [5]. Sin embargo, debido a la cantidad de datos requerida, en muchas pruebas se tiende a emplear un subdominio para llevar a cabo los experimentos. Mientras que los subdominios permiten llevar un mayor control sobre los factores que alteran el ambiente, también se entiende que los modelos generados pueden no llegar a ser suficientes para dominios que abarquen más acciones.

El MO debe ser al menos capaz de proveer información para que los agentes o miembros de un equipo, puedan anticipar la mayoría de los movimientos del oponente con cierta precisión. Esto debe conservarse inclusive cuando se enfrentan a oponentes que no habían sido presentados o estudiados previamente.

Han existido experimentos en los que se ha intentado mostrar la importancia de crear un buen modelo. Parker et al. [6] decidieron probar dos casos extremos, el primero consistió en actuar como si el oponente fuese a escoger siempre la mejor opción dada una situación. El segundo caso consistió en no tomar en cuenta el conocimiento que se tiene sobre el oponente. Esto fue probado bajo un juego de información incompleta y debido a esto, la suposición de que un oponente pueda tomar siempre la mejor opción es incorrecta. Los resultados generales mostraron este error ya que las pruebas en las que no se tomaba en cuenta al oponente, obtuvieron mejores resultados que las otras. La causa principal que fue atribuida es que reaccionar a la mejor opción posible en este tipo de ambientes es inútil ya que el oponente no conoce todos los factores que la rodean y por lo tanto difícilmente llegará a emplear dicha opción.

La creación de un modelo del oponente en tiempo real es uno de los principales retos existentes dentro del área. Las dos mayores complicaciones que se presentan se encuentran ligadas entre sí. La primera consiste en la cantidad de datos requeridos para crear el modelo, esto se traduce en tener que considerar un número considerable de interacciones que permitan generalizar el modelo. En el caso de un juego de fútbol, la cantidad de interacciones entre jugadores dentro de un juego no es suficiente para poder deducir la estrategia general del equipo [7]. La segunda complicación radica en el tiempo requerido para procesar los datos obtenidos y se deriva de la cantidad de información acumulada. Esto obliga a que se deba tener un conocimiento previo del rival en lugar de comenzar a armar el modelo en tiempo de juego.

Dado que siempre se debe saber reaccionar ante los rivales sin importar si fueron estudiados con anterioridad o si se desconocen por completo, se han hecho recomendaciones al respecto. Una de las posibilidades consiste en tener un plan de contingencia [8] que consiste en tener un modelo lo más genérico posible que abarque solamente información básica. La idea consiste en que esto se emplee como un último recurso cuando no se logra determinar el tipo de rival o simplemente no encaja en alguno de los modelos con los que se cuenta.

En el caso de esta investigación, se decidió que el ambiente a utilizar para el Sistema de Predicción de Patrones de Juego (SPPJ) sería el Simulador 2D de RoboCup. La decisión se basa en que se presenta un ambiente dinámico, con reglas estructuradas, basado en un deporte que es popular en la mayor parte del mundo, se tienen registros de competencias pasadas, se cuenta con los ejecutables para los equipos que ingresan a la competencia y el entorno permite ser modificado si es requerido en su caso [9].

En el fútbol, las estrategias se basan en la posición y posesión del balón debido a que es el factor más determinante en el campo puesto que el objetivo del juego gira a su alrededor. El poder determinar la ruta que el balón seguirá en posesión del rival permite a un equipo reaccionar de modo que pueda detener al oponente en su camino a lograr su objetivo. Las rutas del balón crean así un modelo indirecto del oponente ya que gracias a ellas podemos conocer los patrones de jugadas que emplearán sin tener que considerar todos los movimientos de los jugadores (sólo a los que participan o tienen alguna interacción con el balón). El seguir la ruta del balón también nos muestra los puntos en los que los jugadores deben estar colocados para realizar las jugadas.

Esta investigación está basada en las acciones defensivas de un equipo dentro de un sistema multiagentes que simula un juego de fútbol. La estructura del artículo está organizada de la siguiente forma. La sección 2 comprende creación de la base de conocimiento incluyendo la recolección y limpieza de datos. Las características dentro del desarrollo de un juego son discutidas en la sección 3. La sección 4 contiene los resultados obtenidos empleando el SPPJ. Por último, la sección 5 aporta una serie de conclusiones y trabajo a futuro contemplado.

2. Creación de la base de conocimiento

Para crear un modelo del oponente útil para el SPPJ, es necesario tomar en cuenta una serie de experiencias previas. Dado el tipo de ambiente y la accesibilidad de las fuentes, se emplearon registros de juegos pasados. Los registros son generados automáticamente por el Simulador 2D de RoboCup cada que un juego es ejecutado y se guarda en un archivo de tipo RCG.

Los archivos RCG son archivos binarios que pueden ser transformados en archivos del tipo XML en los que se guarda toda la información correspondiente a al juego incluyendo los parámetros del servidor, parámetros de los jugadores, estado del juego y las acciones de cada jugador.

Debido a que estamos empleando un modelo del oponente basado en la posición del balón, no toda la información contenida nos es útil por lo que decidimos eliminarla. Este proceso reduce el tamaño de los archivos y resulta en un XML que contiene las acciones de los jugadores, las posiciones de los jugadores, la posición del balón y el estado del juego.

Por cuestiones del tiempo que le toma a los jugadores desplazarse físicamente dentro del terreno de juego y tomando en cuenta el tiempo requerido para procesar las predicciones, se ha decidido que los patrones incluidos dentro de la base de conocimientos deben abarcar al menos 10 pasos en tiempo del servidor. No consideramos otro tipo de restricción al momento.

Para reducir la complejidad del modelo del oponente, se decidió evitar el uso de posiciones absolutas y en su lugar usar zonas que cubren cierto espacio. La división también permite al SPPJ ser tolerante al ruido provocado por el entorno, en este caso el Simulador 2D de RoboCup.

La división en zonas ha sido empleada con anterioridad [10,11], sin embargo, no se menciona una metodología específica que permita la obtención de un tamaño ideal por zona. En general, la división en zonas en este tipo de ambientes se lleva a cabo de manera empírica.

En nuestro caso, la división debía cumplir con que cada región debe ser lo suficientemente grande para reducir la complejidad pero también debe ser suficientemente pequeña para que la predicción continúe siendo relevante. La razón se encuentra en que tener una división muy pequeña daría demasiadas combinaciones para el momento de la búsqueda, desapareciendo así la ventaja de crear una división. Por el contrario, tener zonas de tamaño demasiado grande, nos daría un espacio de búsqueda pequeño pero la precisión en cuanto al lugar en el que el balón se encuentra sería irrelevante para los defensores puesto que abarcaría demasiado espacio.

El resultado de la división fue la creación de una malla de 60 zonas. La división también tiene la ventaja de tomar como puntos las banderas existentes dentro del campo del Simulador 2D de RoboCup. La Figura 1 muestra la división obtenida.

Se ha optado por generar un árbol de búsqueda que es empleado en la fase de juego y es el enlace entre el juego actual y las experiencias previas. El árbol permite reducir el tiempo empleado para comparar la situación actual del juego contra los patrones establecidos. La estructura del árbol está compuesta por una

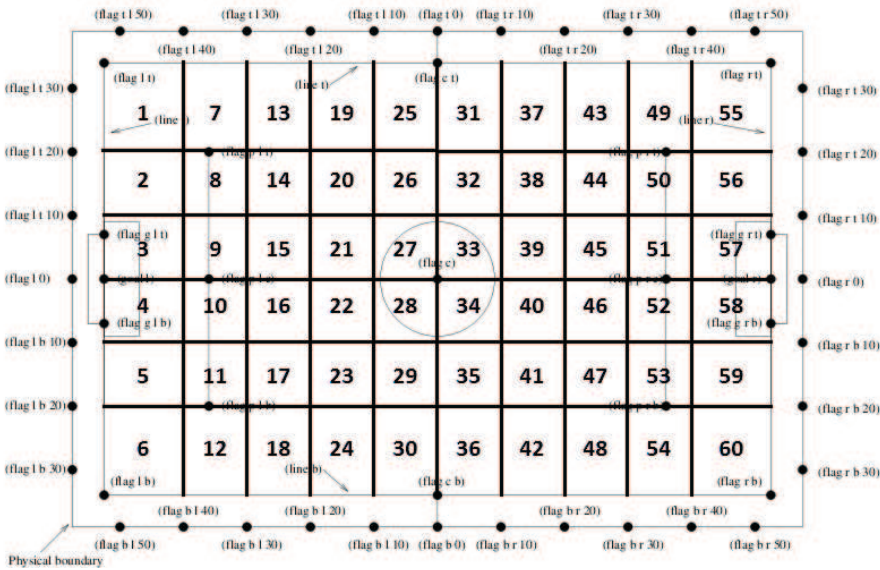


Figura 1. División del campo empleada en el SPPJ.

serie de zonas de interés que indican la colocación inicial tanto del balón como de los jugadores rivales al inicio de una jugada. Como muestra la Figura 2, cada rama del árbol de búsqueda corresponde a un patrón que puede presentarse dentro del juego, empezando por la zona del balón y siguiendo por las zonas en las que se encuentran los jugadores.

3. Fase de juego

Para hacer uso de la base de conocimiento creada en la fase anterior, el SPPJ fue probado en nuevos juegos ante varios rivales. Dentro de esta etapa se debe seguir el proceso completo: obtener la información actual del juego (posiciones del balón y de los rivales), búsqueda de patrones similares dentro de patrones similares dentro de la base de conocimientos (a través del árbol de búsqueda) y obtención de las posibles zonas en las que el balón se encontrará en determinado tiempo.

Para obtener el estado actual del juego se debe obtener la información parcial que cada agente conoce. Sin embargo, los parámetros por defecto del Simulador 2D de RoboCup limitan la comunicación a que cada agente puede escuchar un mensaje aleatorio por ciclo. Debido al enfoque del SPPJ, la forma en cómo se debe lidiar con una comunicación limitada queda fuera del alcance por lo que para efectos de las pruebas, se decidió eliminar dicha limitación.

Aún con los parámetros de comunicación modificados, se debía elegir a quien debería llevar a cabo el proceso de predicción. Los enfoques pensados fueron una decisión distribuida, que cada agente haga el proceso completo de predicción o

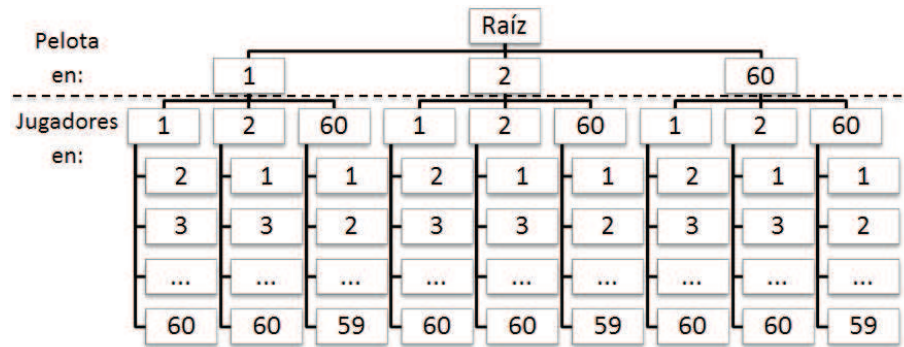


Figura 2. Estructura del árbol de búsqueda.

que un solo agente obtenga la información, complete el proceso y comunique los resultados. Las primeras dos opciones implicaban una gran cantidad de mensajes dentro del servidor y principalmente, los jugadores debían distraerse de su labor en el juego para llevar a cabo el proceso por lo que dejaban pasar sus acciones básicas por estar procesando. Debido a esto, se decidió tener un punto central que llevara a cabo el proceso y que sólo requiriese que la información de los agentes le fuera enviada. Para evitar otra negociación completa para decidir quién sería el agente encargado, se decidió optar por el portero debido a que es el jugador que tiene la menor cantidad de interacciones con el balón y que tiene la mayor visión del campo de juego.

Una vez que el portero recibe toda la información, debe ser limpiada y consolidada en un solo patrón que será usado en el árbol de búsqueda. El resultado de la búsqueda devuelve una serie de identificadores de las jugadas que mayor posibilidad tienen de ser ejecutadas en el momento. La medida de similitud empleada para determinar las mejores posibilidades se basa primero en la diferencia de distancias entre la situación actual y los patrones en la base de conocimiento, como segundo criterio se usa el número de jugadores que participan en la jugada, dando así, mayor relevancia a los patrones que involucren una mayor cantidad de agentes.

El proceso de recuperación de patrones se encuentra basado en el Razonamiento Basado en Casos (RBC). RBC es una técnica de aprendizaje automático que se basa en el pensamiento humano y su forma para reaccionar a situaciones actuales basado en experiencias previas. Este tipo de técnica ha sido empleado previamente en este dominio [10,11]. Al conjuntar los patrones de juego que coinciden con la situación actual del campo de juego, se obtiene la ruta que el balón sigue en cada uno de ellos y se hace una predicción con un muestreo de 5 ciclos. Esto nos deja con una serie de probabilidades distribuidas a lo largo del tiempo sobre la posición del balón, al conjuntarse se puede obtener un patrón de predicción como muestra la Figura 3.

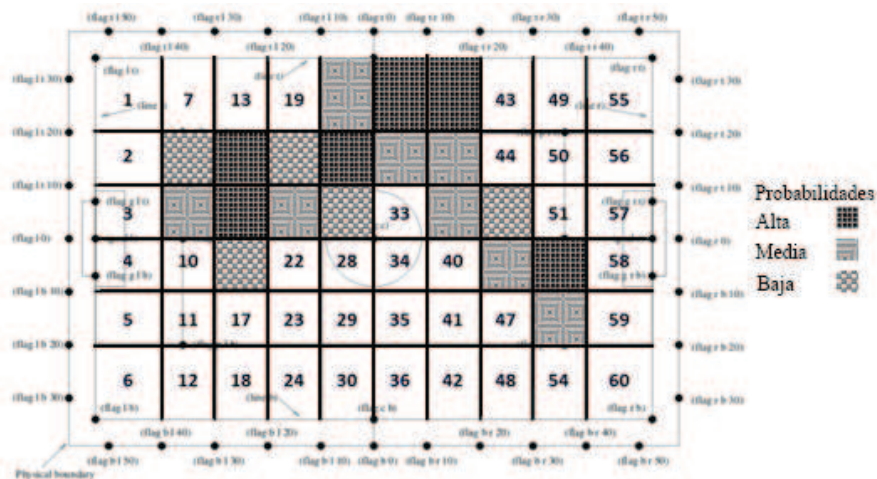


Figura 3. Probabilidad de la presencia del balón a lo largo de una predicción.

Teniendo una predicción con un muestreo de 5 ciclos nos permite determinar las zonas que deben ser cubiertas por nuestros jugadores para evitar el avance del rival y cuales tienen menor prioridad. Siendo un juego de fútbol, las zonas que rodean a la zona predicha también deben ser cubiertas para impedir completamente el ataque del contrario.

Algunas consideraciones tomadas en cuenta dentro del juego al llevar a cabo la predicción son:

- Se debe considerar si es viable mandar a los jugadores a defender una zona de cobertura dependiendo del tiempo pronosticado y la distancia de los jugadores a dicha zona.
- Se debe tener en cuenta que el proceso de predicción toma entre 1 y 2 ciclos para ser completado, sin embargo, en casos extremos en los que no se encuentre un patrón similar se puede tardar hasta 15 ciclos por lo que debe determinarse si se debe continuar la predicción o si es necesario establecer un límite de ciclos.
- En ocasiones en las que el balón se encuentre cercano a la portería, es recomendable deshabilitar la predicción para permitir que el portero se encargue de llevar a cabo sus labores naturales.

Resueltas las condiciones anteriores, el estado natural del juego permite al portero determinar la viabilidad de tomar acciones defensivas, las debe comunicar al resto del equipo y consisten en las zonas y el tiempo en el que un espacio debe ser cubierto puesto que se considera que el balón rondará el lugar.

Los registros generados en las pruebas pueden ser analizados e incluidos en la base de conocimiento. Esto permite al sistema evolucionar y aprender situaciones en las que los rivales han empezado a adaptarse a las condiciones generadas por la predicción.

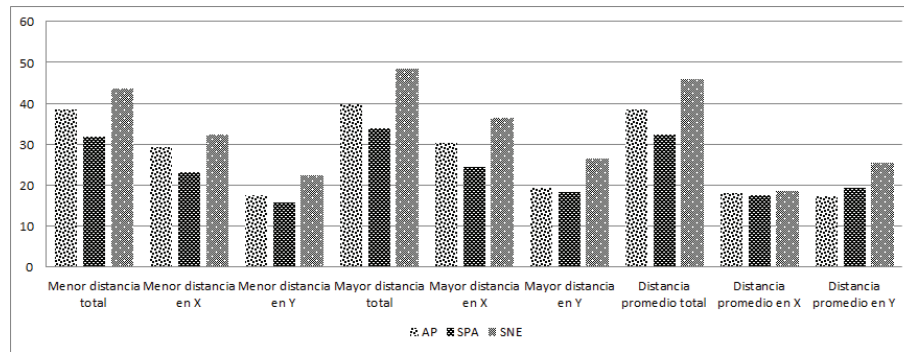


Figura 4. Distancias para: AP – Analizados Previamente; SPA – Sin Previo Análisis; SNE – Sin Nuestro Equipo.

4. Resultados

Se realizaron tres tipos de pruebas: nuestro equipo contra equipos previamente analizados e incluidos en la base de conocimiento, nuestro equipo contra equipos no incluidos en la base de conocimiento y juegos en los que nuestro equipo no participó. El primer tipo de pruebas se usó para probar la confiabilidad del SPPJ frente a equipos conocidos. La segunda muestra la confiabilidad sobre situaciones desconocidas para el sistema. El tercer tipo es una prueba general para mostrar la utilidad de la aplicación fuera de las estrategias empleadas por un solo equipo. Se ejecutaron 27 pruebas de las cuales 12 fueron del primer tipo, 3 sobre el segundo y 12 más sobre el tercero.

En general, el SPPJ logró obtener una precisión en la predicción de la posición del balón de más del 80 % en un rango aceptable, siendo éste de una zona que rodea la posición real del balón. Esto permite definir las zonas de cobertura en un tiempo en el que puedan ser ocupadas por nuestros jugadores para que el ataque sea detenido y el balón recuperado.

Los resultados que involucran la distancia entre la predicción y la realidad se muestran en la Figura 4. Según lo obtenido, tomar las distancias promedio en X y en Y brindan las mejores posiciones para determinar la situación actual.

La Figura 5 muestra el porcentaje de resultados que son útiles para la predicción empleando sólo la distancia promedio en X y la distancia promedio en Y. En la mayoría de los casos excede al 80 % por lo que se puede tener cierta fiabilidad a la hora de llevar a cabo las acciones defensivas.

Analizando los resultados mostrados, pudimos observar las siguientes situaciones:

- Existen equipos que su comportamiento no es consistente con lo obtenido en la base de conocimiento por lo que la predicción de su movimiento se dificulta.
- En las predicciones, existen situaciones en las que el balón no llega a las zonas en las que se suponía debía pasar debido a situaciones que se presentaron

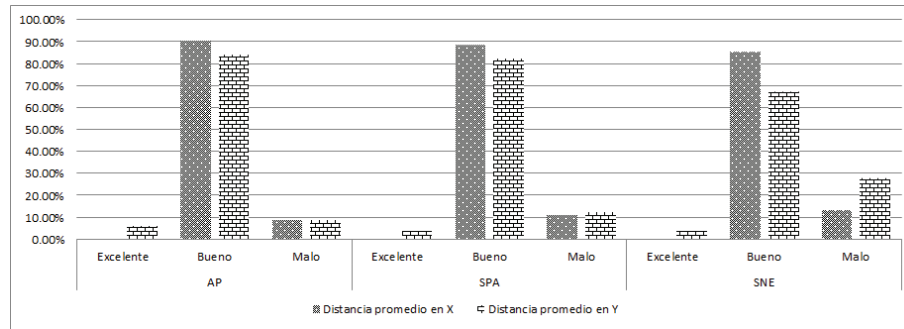


Figura 5. Porcentaje de utilidad en los resultados para: AP – Analizados Previamente; SPA – Sin Previo Análisis; SNE – Sin Nuestro Equipo.

en la ruta. Por ejemplo en algunos casos se cambia la ruta al presentarse un jugador de nuestro equipo como obstáculo y otras en las que nuestro equipo logra recuperar el balón.

- Algunos equipos cambian sus estrategias y tácticas dependiendo del resultado del juego y del tiempo que queda por jugar.

Aun teniendo en cuenta estas situaciones, se considera que las predicciones tienen un buen rango de precisión y se logran llevar a cabo en un tiempo suficiente para reaccionar.

5. Conclusiones y trabajo a futuro

En este artículo, se presentó el SPPJ, un sistema que permite obtener información de los registros de partidos del Simulador 2D de RoboCup y conjunta en una base de conocimientos los patrones de estrategias usadas por oponentes. Se genera un árbol de búsqueda que indexa todos los posibles casos para poder ser comparados en tiempo real. En tiempo de juego, se usa la información correspondiente a la posición de los jugadores y el balón, se compara con los registros guardados y se genera una predicción de la posible ruta que el balón seguirá. Todo el proceso se lleva a cabo en un tiempo muy corto para que pueda emplearse en tiempo real aun en el ambiente dinámico en el que fue probado.

El modelo discutido obtiene una precisión de resultados útiles de más del 80 % en las pruebas. Dadas las consideraciones sobre el ambiente, podemos concluir que es un porcentaje muy aceptable.

Se considera que el modelo presentado puede ser llevado a uso en otro tipo de dominios que no requieran una competición directa, dominios en los que se requiera obtener un patrón claro y un objetivo a predecir. También se requiere tener un registro de experiencias previas para poder llevar a cabo la comparación.

Las mejoras y el trabajo a futuro incluyen:

- Encontrar la división óptima de las zonas.

- Incorporación de pesos en la fase de aprendizaje.
- Emplear distintas técnicas de aprendizaje supervisado.

Referencias

1. Wooldridge, M.: An introduction to multiagent systems. Wiley (2008)
2. Del Giudice, A., Gmytrasiewicz, P., Bryan, J.: Towards strategic kriegspiel play with opponent modeling. In: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2. AAMAS '09, Richland, SC, International Foundation for Autonomous Agents and Multiagent Systems (2009) 1265–1266
3. Laviers, K., Sukthankar, G., Molineaux, M., Aha, D.W.: Improving offensive performance through opponent modeling. In: Proceedings of Artificial Intelligence for Interactive Digital Entertainment Conference (AIIDE). (2009) 58–63
4. Richards, M., Amir, E.: Opponent modeling in scrabble. In: Proceedings of the 20th international joint conference on Artificial intelligence. IJCAI'07, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (2007) 1482–1487
5. Kuhlmann, G., Knox, W.B., Stone, P.: Know thine enemy: a champion robocup coach agent. In: proceedings of the 21st national conference on Artificial intelligence - Volume 2. AAAI'06, AAAI Press (2006) 1463–1468
6. Parker, A., Nau, D., Subrahmanian, V.S.: Overconfidence or paranoia? search in imperfect-information games. In: proceedings of the 21st national conference on Artificial intelligence - Volume 2. AAAI'06, AAAI Press (2006) 1045–1050
7. Stone, P., Riley, P., Veloso, M.: Defining and using ideal teammate and opponent models. In: Proceedings of the Twelfth Annual Conference on Innovative Applications of Artificial Intelligence. (2000)
8. McCracken, P., Bowling, M.: Safe strategies for agent modelling in games. In: AAAI Fall Symposium on Artificial Multi-agent Learning. (2004) 103–110
9. RoboCup: Robocup website (2009)
10. Arias, M.: Definición de una arquitectura para la generación y selección de estrategias en equipos de fútbol robótico. Master's thesis, Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Estado de México (2008)
11. Berger, R., Burkhard, H.D.: At humboldt team description 2007. Lecture Notes in Computer Science (RoboCup 2007) (2008)