

# Modelado de oponentes en paralelo con conocimiento del dominio: búsqueda directa de estrategias en sistemas multiagentes

Roberto Ulloa R.<sup>1</sup>, Elena Gabriela Barrantes S.<sup>1</sup>,

<sup>1</sup> Escuela de Ciencias de la Computación e Informática, Universidad de Costa Rica  
{roberto.ulloa, gabriela.barrantes}@ecci.ucr.ac.cr

**Abstract.** This paper presents a system that infers the different tactics employed by autonomous agents participating in the execution of a multi-player strategy game (The Werewolves of Thiercelieux). One of the agents is a “spy” who participates in the game as a regular player, but at the same time attempts to guess the strategy followed by the others, distributing the inference process in a multi-agent platform it has access to. Each agent inside this platform models the strategy of one player in the real game. This work shows that there are scenarios where strategies can be inferred modeling the behavior of each opponent in parallel. Moreover, it allows to diminish the search by filtering only the necessary information for the inference of each strategy. Finally, it reinforces the point that even partial knowledge of possible strategies (domain knowledge) diminishes significantly the search space and allows faster and more accurate predictions.

**Keywords:** strategy inference, multi-player games, multi-agent systems

## 1 Introducción

Un problema frecuente en el análisis de datos de experimentos en sistemas multiagente es el cómo inferir estrategias a partir de las acciones de los mismos [1]. Ésta ha sido un área de particular interés para distintas disciplinas sobretodo, a partir del surgimiento de la teoría de juegos a finales de los años 70 [2]. No obstante, hasta la fecha, se siguen utilizando escenarios simples para realizar esta inferencia porque el espacio de búsqueda en este problema aumenta exponencialmente en múltiples variables (acciones, estrategias, estados y agentes) [3].

En este trabajo se presenta un modelado de oponentes a través de agentes que infieren las estrategias de los participantes (también agentes) del juego "Los Hombres Lobos de Castronegro" presentado en [4]. El sistema de inferencia modela a cada agente por separado (en paralelo), en lugar de estimar una mejor decisión a partir del estado general del sistema. Con esto se muestra que existen escenarios donde las estrategias individuales son suficientemente independientes y pueden ser inferidas modelando el comportamiento de los oponentes en forma separada. Además, el

sistema de inferencia utiliza conocimiento del dominio para predefinir las posibles estrategias que se pueden encontrar y con esto disminuir el espacio de búsqueda.

En el área de la computación, la inferencia de estrategias es conocida como "búsqueda directa de políticas" y forma parte del área de aprendizaje [3]. En esta área, cualquier mecanismo que permita decidir una acción a partir del estado de un sistema es considerado una estrategia [5]. Para tomar esta acción, la búsqueda directa de políticas se caracteriza por tomar ventaja de la estructura de la solución a diferencia de otros métodos que aprovechan la estructura del problema en sí [6].

El conocimiento de la estructura de la solución del dominio permite establecer meta-estrategias. Básicamente, en lugar de establecer un mapeo directo entre todos los posibles estados del sistema y una acción correspondiente, se establecen reglas (meta-estrategias) que se aplican sobre el estado permitiendo establecer una acción. Por último, se ha visto que el modelado de oponentes y sus estrategias por parte de un agente dentro del sistema es útil para que este último incremente la eficiencia de sus acciones [7], sobre todo si lo que se logra inferir son las meta-estrategias [8], que es parte de lo que retoma este trabajo.

La motivación de realizar el modelado por agentes está basada en que no está claro que la selección individual (o por subgrupos de agentes) de una estrategia en un sistema multiagente produzca un patrón único que pueda llevar a uno de los agentes dentro del sistema a tomar una buena decisión basándose en el espacio de búsqueda como un todo. Aprovechando que implícitamente existe una división del espacio debido al patrón de comportamiento (producido por la existencia de una estrategia) de cada agente, nace la idea de paralelizar la inferencia a través de agentes.

En la sección 2, se describe el escenario y las herramientas utilizados. Una descripción del sistema de inferencia es presentado en la sección 3. La metodología para probar la inferencia se describe en la sección 4. En la sección 5, se muestran los resultados obtenidos y un análisis de los mismos. Por último, las conclusiones se presentan en la sección 6 y el trabajo futuro en la sección 7.

## **2 Descripción del escenario**

El presente estudio se realizará en un sistema de agentes construido en [4]. El mismo está basado en un juego de estrategia llamado los Hombres Lobo de Castronegro [9]. El sistema cuenta con múltiples tipos de agentes (personajes): ciudadanos, lobos y brujas. Además de ellos, existe un moderador que controla las acciones del juego pero el no es un jugador. El objetivo de cada jugador es sobrevivir al final del juego. Los lobos matan a un ciudadano cada noche. En el día, los ciudadanos intentan adivinar quiénes son los lobos por medio de alguna estrategia. Igualmente, los lobos usan estrategias para que el jugador que muera en el día sea un ciudadano. Las brujas son ciudadanos que tienen poderes especiales que pueden usar en las noches.

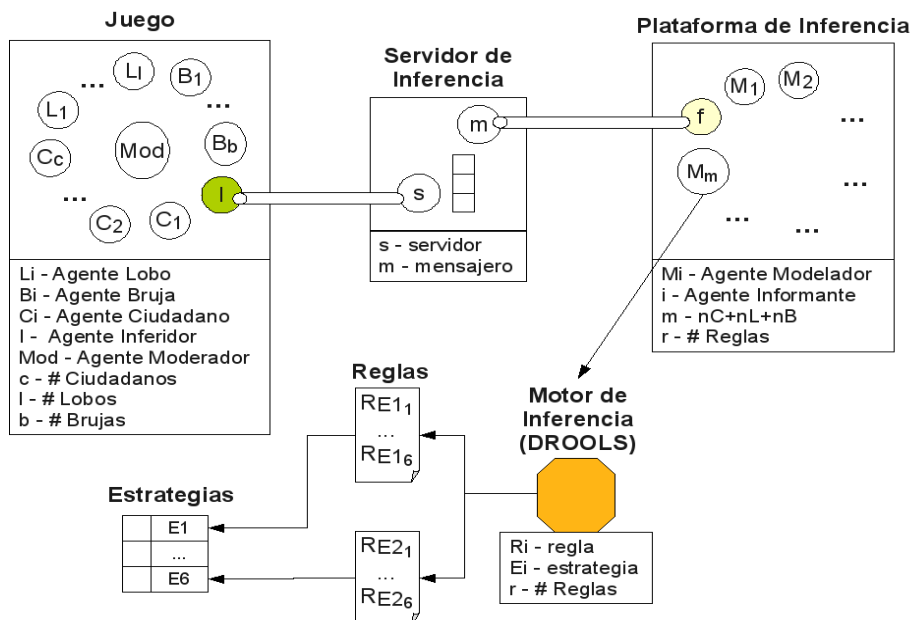
El juego descrito trabaja sobre la plataforma para la construcción de sistemas multiagente JADE [10]. En ella cada agente es implementado como un hilo y puede estar en una máquina diferente. El sistema de inferencia aquí presentado modela cada oponente como un agente de JADE, por lo cual el sistema de inferencia funcionaría en paralelo. La comunicación entre el juego y el sistema de inferencia se hace por medio de RMI de Java [11]. El motor de inferencia fue construido usando Drools [12], un motor de reglas que también trabaja con Java e incluye un *plugin* para Eclipse [13].

### 3 Descripción del sistema de inferencia

Para desarrollar el sistema de inferencia se implementó e incluyó un *agente inferente* que fuera responsable de inferir las estrategias. El mismo participa del juego como si fuera un ciudadano común usando una estrategia *Aleatoria* [4]. La única información que tiene disponible es la misma que tienen los otros ciudadanos. Es decir, que la inferencia va a ser realizada en un dominio parcialmente observable. Al igual que cualquier otro ciudadano, él no conoce la identidad de los otros jugadores (lobo, bruja o ciudadano) ni sus estrategias, sin embargo, puede utilizar la información de la que dispone para adivinarlas. Posteriormente podría usar esa información a su favor.

En la Fig. 1 se muestra la arquitectura del sistema. El *inferente I* se va a encargar de enviar información al **Servidor de Inferencia**, quien coloca el mensaje en una cola. Después, un hilo *mensajero m* va a reenviar ese mensaje al agente *informante f* que se encuentra dentro de la **Plataforma de Inferencia** donde existe un agente *modelador Mm*, trabajando en paralelo, para cada jugador. El agente *informante f* distribuye el mensaje a aquellos agentes modeladores a quienes corresponda según la utilidad de ese mensaje para la inferencia de la estrategia.

Cada agente *modelador Mm* tiene una instancia del **Motor de Inferencia** para descubrir la estrategia del agente a quién está modelando. El motor opera usando reglas de premiación y de castigo sobre las estrategias (seis actualmente). Si una regla de premiación se cumple, la puntuación de la estrategia respectiva aumenta. Si una regla de castigo se cumple, entonces la puntuación disminuye. Si  $K$  es la mayor puntuación obtenida en un ciclo de inferencia, todas aquellas estrategias cuya puntuación sea  $K$  se consideran candidatas.



**Fig. 1. Arquitectura del Sistema de Inferencia.** En la parte superior se representan los tres componentes principales de todo el sistema: a la derecha, el juego junto con un círculo representando cada jugador (lobo L, ciudadano C o Bruja B) y al moderador M; al centro, el servidor de inferencia con un servidor s, un mensajero m y una cola para el control de tráfico; y, a la derecha, el modelo del juego con un círculo representando cada agente modelador M. Cada agente modelador M contiene una instancia del Motor de Inferencia que aplica las reglas establecidas para cada posible estrategia modificando la puntuación de la misma.

Es claro que el esquema de inferencia presentado en este trabajo es muy sencillo, sin embargo, era suficiente un motor de desempeño medio para demostrar que es posible dividir el espacio de búsqueda. Se presume que un motor de inferencia refinado mejoraría significativamente la calidad de los resultados.

## 4 Metodología

Lo que interesa en este artículo es mostrar que la inferencia de estrategias se puede realizar en forma separada por cada agente, por lo que, la calidad del motor de inferencia no es un factor tan importante. Por lo tanto, los experimentos fueron diseñados para medir de forma simple si se logra alcanzar valores aceptables de inferencia, de manera que el agente inferente realmente pudiera sacar provecho de la información. Todos los experimentos descritos son ejecutados en una portátil con un procesador AMD Turion64, 1GB de RAM y Ubuntu 8.04 de sistema operativo.

Primero se van a correr experimentos para determinar el número de turnos promedio que ocupa el sistema para realizar una inferencia razonable y, una vez determinado esto, se procederá a medir la calidad del motor en cuanto a su capacidad

para distinguir estrategias específicas bajo varios escenarios. Para esto, se van a utilizar dos variables de respuesta que medirán la calidad las cuales hemos llamado **precisión** y **correctitud** por su analogía con los verdaderos conceptos. Dichas variables se explicarán a continuación.

**Precisión:** Por la forma en que fue construido el motor de inferencia, en un turno cualquiera, pueden haber varias estrategias candidatas. Se va a considerar que mientras menos estrategias se infieran, más "precisa" es la predicción independientemente de la correctitud. La precisión de la inferencia se calcula como el inverso de la tasa de candidatos con respecto al número máximo de estrategias:

$$Precisión = 1 - \frac{i-1}{6-1} = 1 - \frac{i-1}{5}, \text{ donde } i \text{ son las estrategias inferidas}$$

Desde un punto de vista práctico, una precisión inferior a 0,8 es demasiado poco específico para tomar una acción, y es poco útil. En nuestro caso, una precisión mayor que 0,8 implica que se están presentando menos de dos estrategias posibles en promedio, con lo cuál se podría tomar una mejor decisión final.

**Correctitud:** Se dice que una determinada inferencia para un agente es "correcta" si la estrategia verdadera que usó el agente (conocida por nosotros) forma parte de las opciones de estrategias posibles que indica el motor. Para una precisión de 0,8, o sea, seleccionando a lo sumo dos estrategias, es suficiente con que una inferencia real superara, en promedio, un valor de 33,33% de correctitud debido a que éste valor representa la selección aleatoria (y no inferida) de 2 de las 6 estrategias disponibles. En nuestro caso, decidimos que se iba a exigir que la inferencia fuera superior al 70% para poder afirmar que la inferencia era buena.

A pesar de que la correctitud y la precisión se pudieron haber correlacionado se decidió separarlas para tener más claro lo que estaba sucediendo en el motor.

**Tabla 1. Factores de los experimentos.** En los estudios preliminares (Sección 4.1), los turnos son factor de diseño y en el análisis de las estrategias (Sección 4.2), las estrategias son factor de diseño

Factor	Tipo	Rango	Valor(es) Estudiado(s)	Descripción
Turnos	Diseño / Fijo	1,...,f(x)	2,...,8	Este factor mide la cantidad de turnos que han ocurrido en el momento en que se hace la medición. Un turno cambia cada vez que se completa un día.
Ciudadanos	Fijo	1,...,nc	16	Cantidad de ciudadanos en el sistema
Brujas	Fijo	0,...,nb	3	Cantidad de brujas en el sistema
Lobos	Fijo	1,...,nl	3	Cantidad de lobos en el sistema
Estrategias	Fijo / Diseño	{e1,...,e6}	{e1,...,e6}	Estrategia de cada uno de los agentes: <i>Aleatoria, Siempre al Mismo, Siempre Al De Menos Confianza, Ojo por Ojo, Ataque a Asesinos</i>

Los factores a ser considerados en los experimentos se describen en la Tabla 1. La cantidad de ciudadanos, brujas y lobos fueron seleccionados basados en estudios

previos realizados en [4]. A continuación se describen los estudios preliminares y el análisis de inferencia por estrategia.

#### 4.1 Estudios preliminares

Para estandarizar los experimentos se necesitaba fijar la cantidad de turnos necesaria para realizar una inferencia razonable, definida como una precisión superior a 0.8 y una correctitud superior a 0.7. Para esto, se recolectaron datos de múltiples **iteraciones** del sistema. Observando la tasa de cambio de las variables de respuesta en el tiempo se comprobó que las mismas convergían en menos de 100 iteraciones.

Las estrategia que usa cada agente sigue es seleccionada aleatoriamente al inicio del juego excepto por la estrategia *Todos Al Mismo* que por sus características debe ser asignada arbitrariamente.

#### 4.2 Análisis de inferencia por estrategia

Una vez que se estudió el comportamiento general del sistema y se fijaron valores para la cantidad de turnos e iteraciones, se procedió a recolectar más datos para analizar la calidad del motor de inferencia para cada una de las estrategias. Para esto, se ejecutaron múltiples juegos en los cuales se designó la misma estrategia a todos los agentes participantes. La excepción fue nuevamente la estrategia *Todos Al Mismo*.

Por último, se repitió el experimento del estudio preliminar donde los agentes seleccionaban su estrategia aleatoriamente al inicio del juego. Esta vez, la cantidad de iteraciones fue incrementada en un factor de 6 para mantener la calidad de los datos<sup>1</sup>.

### 5 Resultados

Los resultados se agruparon en las siguientes cuatro categorías: selección de la cantidad de turnos, selección de la cantidad de iteraciones, análisis de la calidad de inferencia por estrategia en un ambiente fijo y en ambientes diversos.

#### 5.1 Evaluación de la calidad de la inferencia en los turnos del juego

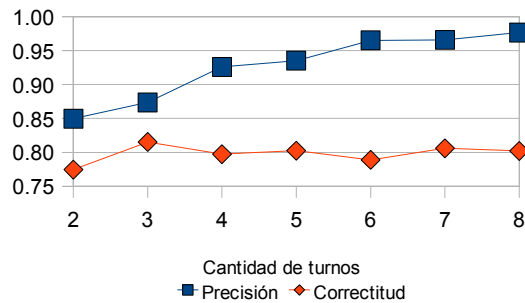
La Fig. 2 muestra un análisis por turnos de los dos valores de respuesta. Se puede observar como la precisión va aumentando conforme se acerca el final del juego mientras la correctitud se mantiene. Esto indica que conforme los turnos avanzan la inferencia logra descartar más estrategias pero parece ser que pocas de ellas son reinsertadas dentro de las posibilidades de un agente. Esto puede deberse a que la

---

<sup>1</sup> El incremento en un factor de 6 se debe a que se estaban estudiando 6 estrategias, entonces, en promedio, se iban a tener 1/6 del total de datos para cada una cuando la selección se hacía aleatoriamente entre ellas.

premiación y el castigo en las reglas no tiene pesos asociados. Se presume que es posible mejorar el desempeño dando más importancia a ciertas reglas de premiación.

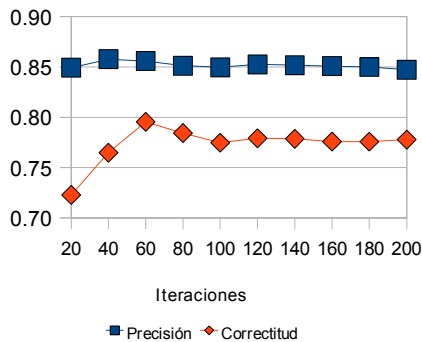
Lo más importante es que se encontró que el sistema alcanzaba los valores de correctitud (0,7) y precisión (0,8) definidos en la metodología desde los primeros dos turnos. Esto indica que la inferencia es realizada rápidamente mientras que el número de jugadores aún es grande (los jugadores van muriendo en cada turno).



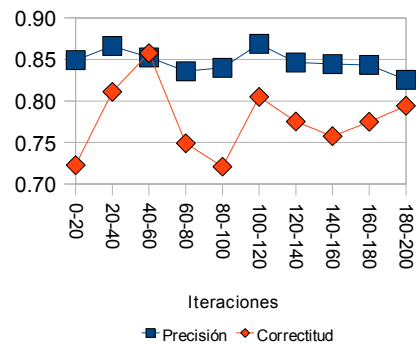
**Fig. 2. Precisión y correctitud del motor de inferencia.** Se consideran 8 turnos consecutivos en 100 iteraciones del juego.

### 5.2 Evaluación y selección de la cantidad de iteraciones

La cantidad de iteraciones fue fijada arbitrariamente en 100 para observar cómo se comportaban las variables de respuesta con respecto a los turnos únicamente. Sin embargo, había que estudiar si ese valor fijado arbitrariamente en 100 iteraciones era suficiente para 2 turnos, cantidad seleccionada en la sección anterior.



**Fig. 3. Convergencia de precisión y correctitud para dos turnos.** Mediciones tomadas cada 20 iteraciones.



**Fig. 4. Variación entre la precisión y la correctitud para dos turnos.** Datos obtenidos con 10 muestras independientes de 20 iteraciones.

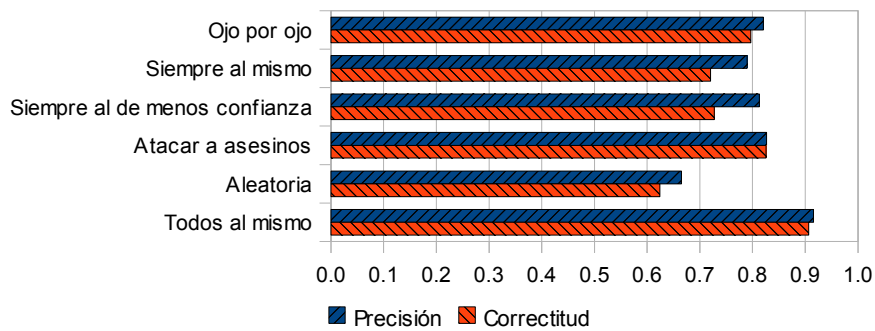
En la Fig. 3 se puede observar que en alrededor de 80 iteraciones la precisión del sistema converge hacia 0,85 y 0,77 para la correctitud. En la Fig. 4 se muestra que la variación entre grupos independientes de 20 iteraciones es relativamente poca

( $\pm 0.08$ ). Aún así, y para garantizar la calidad de los datos, se fijó la cantidad de iteraciones en 120 para los próximos experimentos.

### 5.3 Análisis por estrategia

El siguiente paso en la metodología consistió analizar la inferencia de cada estrategia por separado para lo cual se fijó la misma estrategia para cada agente. En la Fig. 5 se muestran los resultados de este experimento. En promedio, la correctitud para 120 iteraciones se mantuvo (pasó de 0,7791 a 0,7695 para 120 iteraciones en el primer y segundo experimento respectivamente) pero la precisión disminuyó (de 0,8525 a 0,8073).

Se puede observar que la inferencia de la estrategia *Aleatoria* fue la más baja de todas. En general, inferir una estrategia aleatoria es complicado justamente porque no hay un patrón definido. Por otro lado, la correctitud y precisión de la estrategia *Todos Al Mismo* es muy buena. La razón es que es una estrategia colectiva y esto la hace más detectable debido a que hay todo un subgrupo de agentes comportándose coherentemente.



**Fig. 5. Eficiencia individual de la inferencia de cada estrategia.** Se muestra la precisión y correctitud de la inferencia cada estrategia en 120 en donde todos los agentes están usando la misma estrategia

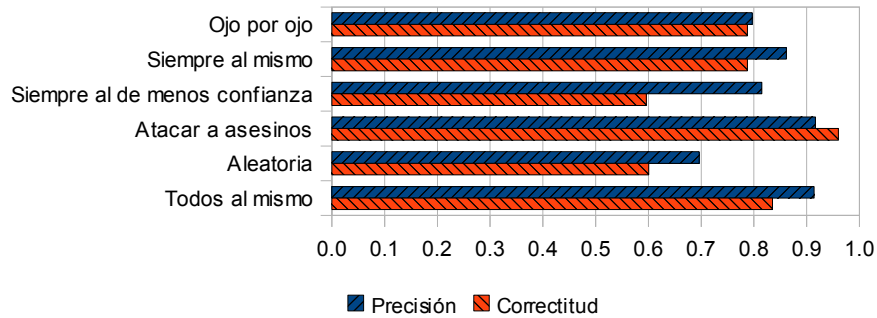
### 5.4 Análisis por estrategia en distintos escenarios

Un último experimento consistió en que los agentes seleccionaran nuevamente su estrategia de manera aleatoria pero, ésta vez, aumentando las iteraciones en un factor de 6, es decir, 720 iteraciones. Esta selección aleatorio de estrategias genera un escenario distinto en cada iteración. El objetivo aquí es justamente analizar los resultados de la inferencia de estrategias en diversos escenarios para saber si la inferencia es consistente.

Ya se había visto en el experimento anterior que la correctitud se había mantenido y la precisión había disminuido. Por esta razón, se se espera que en este experimento la correctitud se mantuviera y la precisión mejorara, cosa que efectivamente sucedió



(la precisión fue de 0,8311 y la correctitud de 0,7602) con lo cual se corroboró aún más la validez de la muestra.



**Fig. 6. Eficiencia individual de la inferencia de cada estrategia en distintos escenarios.** Se muestra la precisión y correctitud de la inferencia cada estrategia en 720 iteraciones en donde todos los agentes seleccionan una estrategia al azar

Analizando la inferencia de las estrategias en este escenario se encontró que no hay cambios drásticos en las respuestas. Sin embargo, sí hay algunos cambios significativos. La Fig. 6 resalta la mejora en la inferencia de *Atacar a Asesinos* y llama la atención las pérdidas de correctitud en *Todos Al Mismo* y *Siempre Al De Menos Confianza*. Se está trabajando en un análisis sobre las causas de estos cambios.

## 6 Conclusiones

Se concluye de este trabajo que:

- Es posible hacer un modelado de oponentes en paralelo. Es decir, existen escenarios con estrategias suficientemente independientes entre ellas para hacer un modelado de cada oponente de forma individual. Además, esto reduce el espacio de búsqueda por agente.
- Hasta ahora se han buscado muchas soluciones generales al problema de aprendizaje de agentes. Sin embargo, teniendo un conocimiento del dominio (lo cual es casi siempre posible en un sistema real) se pueden plantear algunas reglas simples que lleven a conocer sobre los otros.

## 7 Trabajo futuro

El motor de inferencia utilizado en este trabajo es simple, sin embargo, sería interesante probar el sistema con otros tipos de motores de inferencia. Ahora bien, si

se quiere aprovechar la estructura en paralelo cualquiera de ellos debe enfocarse en modelar los oponentes.

Una línea de investigación que está siendo explorada en este momento es el uso de la información obtenida por el agente inferente para causar un impacto en el sistema, con el objeto de mostrar el posible peligro de este tipo de agentes. El presente trabajo presenta únicamente la inferencia de las estrategias pero en ningún momento utiliza esta información.

Por último, se está trabajando en la introducción de ruido para perturbar el aprendizaje del agente observador. Este trabajo se considera muy importante porque la mayoría (si no es que todos) de sistemas multiagente reales usan estrategias estáticas y, por lo tanto, si son adivinadas pueden representar un potencial peligro en un ambiente competitivo.

## Referencias

1. Duffy, J., Engle-Warnick, J.: Using Symbolic Regression to Infer Strategies from Experimental Data. In: Proceedings of the Fifth International Conference: Computing in Economics and Finance. Boston College, MA, USA (1999)
2. Engle-Warnick, J., Ruffle B.J.: The Strategies Behind their Actions: A New Method to Infer Repeated-Game Strategies and an Application to Buyer Behavior. Departmental Working Papers 2005-04. Department of Economics, McGill University (2006)
3. Busoniu, L., Babuska, R., De Schutter, B.: Comprehensive Survey of Multiagent Reinforcement Learning. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 38, pp. 156—172 (2008)
4. Ulloa, R., Barrantes, E.G.: Construyendo un escenario para la simulación de ataques en sistemas de agentes. In: Memorias de las Jornadas Chilenas de Computación 2008. Punta Arenas, Chile (2008)
5. Peshkin, L.M.: Reinforcement learning by policy search. Doctoral Thesis, Brown University (2002)
6. Rosenstein, M.T., Barto, A.G.: Robot weightlifting by direct policy search. In: Proceedings of the Seventeenth international Joint Conference on Artificial Intelligence, Vol. 2, pp. 839—846. Seattle, Washington, USA (2001)
7. Lazaric, A., Quaresimale, M., Restelli, M.: On the usefulness of opponent modeling: the Kuhn Poker case study. International Foundation for Autonomous Agents and Multiagent Systems, pp. 1345--1348. Estoril, Portugal (2008)
8. Makino, T., Aihara, K.: Multi-agent reinforcement learning algorithm to handle beliefs of other agents' policies and embedded beliefs. In: Proceedings of the Fifth international Joint Conference on Autonomous Agents and Multiagent Systems AAMAS'06, pp. 789—791. Hakodate, Japan (2006)
9. Les Loups-Garous de Thiercelieux, <http://lesloupsgarous.free.fr/>
10. Jade - Java Agent DEvelopment Framework, <http://jade.tilab.com/>
11. Remote Method Invocation Home, <http://java.sun.com/javase/technologies/core/basic/rmi/>
12. Drools, <http://www.jboss.org/drools/>
13. Eclipse, <http://www.eclipse.org/>