

## TAC SCM: Agentes software autónomos en la cadena de suministro\*

Juan José Laviós Villahoz, Ricardo del Olmo Martínez

Dpto. de Ingeniería Civil. Área de Organización de Empresas. Escuela Politécnica Superior. Universidad de Burgos. Av. Cantabria s/n. 09006. Burgos. [jjlavios@ubu.es](mailto:jjlavios@ubu.es), [rdelolmo@ubu.es](mailto:rdelolmo@ubu.es)

### Resumen

*TAC SCM es una competición entre agentes software que se viene desarrollando desde el año 2003. Pretende desarrollar agentes autónomos que participen en una cadena de suministro, funcionando de forma coordinada y simultánea en mercados interrelacionados: los mercados de venta de productos a clientes y los de compra de componentes a proveedores. En este trabajo se realiza un estudio descriptivo de las soluciones desarrolladas por los grupos de investigación que han desarrollado los agentes con las mejores clasificaciones en las últimas ediciones.*

**Palabras clave:** Agentes Autónomos, Cadena de suministro, Comercio Electrónico

### 1. Introducción

TAC SCM es una competición entre agentes software que se viene desarrollando desde el año 2003. Pretende desarrollar agentes autónomos que participen en una cadena de suministro, funcionando de forma coordinada y simultánea en mercados interrelacionados: los mercados de venta de productos a clientes y los de compra de componentes a proveedores. Se trata de conseguir agentes software que desarrollen las mejores estrategias en un mercado electrónico (Sadeh et al. 2003).

En este trabajo se realiza un estudio descriptivo de las soluciones desarrolladas por los grupos de investigación que han desarrollado los agentes con las mejores clasificaciones en las últimas ediciones del TAC SCM. Primero se va a realizar una descripción del Juego destacando los aspectos básicos que lo definen, seguidamente se pasa a analizar los agentes donde se indicará cómo han sido construidos, cómo realizan las funciones básicas como agentes participantes en el juego, así como las herramientas y técnicas que implementan. Por último finalizaremos con las conclusiones.

### 2. Trade Agent Competition (TAC)

La competición se desarrolla del siguiente modo: Seis agentes compiten en un mercado virtual. Cada agente representa un fabricante de ordenadores personales (PC). Todos los agentes tratan de vender sus productos un mercado dividido en tres segmentos (alta, media y baja gama). Las demandas dentro de cada segmento son independientes entre sí. Cada día el agente debe responder a la demanda, fabricar los PC necesarios y adquirir los componentes que se requieran. A este respecto, todos los agentes adquieren a los proveedores los componentes necesarios para construir el PC a través de un mercado competitivo, disponen de la misma capacidad diaria de producción y ésta permanece constante a lo largo de todo el juego. Cada agente va a competir de forma simultánea en el mercado de ordenadores personales como vendedor y en el mercado

---

\* Este trabajo ha sido posible gracias a la financiación del Ministerio de Educación y Ciencia mediante la participación en el proyecto DPI2004-06590

de componentes como cliente El juego tiene una duración de 220 días simulados. El agente ganador es el que obtenga un mayor beneficio monetario al finalizar el juego. El beneficio se mide restando a las ventas realizadas (ofertas que hayan sido aceptadas por parte del cliente y entregadas a tiempo), el coste de las compras de componentes, costes de mantenimiento de inventario por cada unidad de stock almacenada y día, y las posibles penalizaciones que el agente haya podido tener por entregas fuera de plazo de productos a clientes.

Cada día el agente debe enviar ofertas a los clientes como respuesta a las RFQs (request for quotes) generadas por estos, debe realizar el programa de producción y el programa de envíos a clientes y solicitar ofertas a proveedores mediante el envío de RFQs.

La información de la que dispone cada agente es limitada ya que no conoce cuales son los intercambios que se producen en el mercado, sólo conoce aquellos en los que el participa. Cada agente tiene que tomar sus decisiones utilizando la información a la que puede acceder. Cada día el agente recibe por parte de los clientes: todas las Request for Quotes (RFQs) que se han generado en el mercado, las órdenes que ha ganado el propio agente como respuesta a las ofertas enviadas a los clientes el día anterior, así como el precio máximo y el precio mínimo al que han sido intercambiado cada tipo de PC el día anterior. Por parte de los proveedores sólo conoce de forma diaria las ofertas que recibe como respuesta a las RFQs enviadas por él mismo el día anterior

Adicionalmente todos los agentes reciben cada veinte días unos informes que contienen información sobre la suma agregada de unidades de cada componente que han sido enviadas por todos los proveedores y la que han sido ordenadas al conjunto de proveedores durante el periodo. También contiene el precio medio de compra durante el periodo de cada tipo de PC y cada tipo de componente. Acerca de cada proveedor, se indica cual ha sido su capacidad de producción media durante el periodo y de los clientes el volumen agregado de productos solicitados, el volumen agregado de productos ordenados y el precio medio al que se ha adquirido cada tipo de PC.

### **3. Agentes participantes y técnicas utilizadas**

A lo largo de las distintas ediciones los agentes participantes han ido desarrollando distintas estrategias y técnicas, que han ido siendo mejoradas año tras año. Las funciones principales que debe realizar cada agente son la predicción de demanda, determinación de los precios a ofertar, análisis de la oferta de componentes para determinación del momento óptimo de compra, y programación de la producción y de envíos a clientes. Hemos estudiado los agentes finalistas de las últimas ediciones. Agentes como TacTex (Pardoe and Stone 2006), Southampton SCM (He et al. 2006), Deep Maize (Kiekintveld et al. 2006), MinniTAC (Ketter et al. 2004), CMieux (Michael et al. 2006) o Mertacor (Panos et al. 2006).

Todos los agentes estudiados siguen un patrón de funcionamiento en tres fases. En la primera los agentes parten de cero, por lo que necesitan conseguir cuanto antes pedidos de clientes, así como adquirir los componentes necesarios para poder comenzar a atender pedidos. Esta fase abarca los dos o tres primeros días de cada juego. Una segunda fase, que abarca la mayor parte del juego, en la que el agente tiene un comportamiento estable, respondiendo a ofertas de clientes y tratando de obtener las necesidades de componentes que se van generando. En la tercera y última fase, los veinte últimos días del juego, el agente intenta de obtener la máxima liquidez en el banco tratando de deshacerse del stock de producto y componentes que tiene almacenado, ya que el ranking final se establece evaluando en dinero que posee el agente en su cuenta bancaria el último día. Nuestro estudio se va a centrar principalmente en el comportamiento de

los agentes en la segunda fase del juego.

### 3.1. TacTex

Autores: (Pardoe and Stone 2006) . El agente ha sido desarrollado por el grupo de Peter Stone de la University of Texas at Austin.

Descripción general: Este agente se está formado por dos módulos:

- *Supply Manager*: Sus funciones son el lanzamiento de RFQs a proveedores, elegir posteriormente qué ofertas aceptar, así como responder a las RFQs de cliente respondiendo con ofertas.
- *Demand Manager*: Sus funciones son la realización del programa de producción y del programa de envíos de pedidos a clientes.

Previsión demanda: El agente crea un modelo de la demanda. Para ello, realiza el seguimiento de la demanda de cliente y trata de estimar los parámetros subyacentes que la definen. En este caso se conoce cual es la distribución mediante la que se genera la demanda. Se trata de una distribución de Poisson cuya media va variando en el tiempo mediante un factor de tendencia multiplicativo que varía también diariamente. Esto le permite proyectar las necesidades de componentes en un plazo cercano. Utiliza el sistema de predicción de la demanda utilizado por el agente DeepMaize. (Kiekintveld et al. 2004)

Aprovisionamiento: El objetivo es obtener los componentes que proyecta el *Supply Manager* al menor precio posible. Realiza un modelo predictivo del comportamiento de los proveedores que le permite determinar si realiza el pedido de un componente para una determinada fecha en el día actual o si aplaza este pedido a otra fecha en el caso de que prevea que los precios bajarán. Utiliza un algoritmo hill-climbing que trata de encontrar la combinación óptima de pedidos que ha de enviar cada día en función de sus necesidades proyectadas de componentes, su inventario y los precios de los componentes previstos para los próximos días.

Precio de compra: El agente también genera un *Modelo de proveedores*. Para ello realiza un seguimiento de cada proveedor utilizando toda la información disponible de cada uno como son las órdenes pendientes que tiene y precios que han sido ofrecidos por el proveedor en respuesta a las RFQs enviadas por el agente. Esto permite realizar una estimación de la capacidad disponible futura del proveedor basada en las ofertas recibidas, lo que permitirá predecir el precio que puede ofrecer un proveedor cuando se le envía una RFQ para la fecha de entrega solicitada. El módulo *Offer Acceptance Predictor* permite estimar la probabilidad de que una RFQ sea aceptada en función del precio, basándose en los resultados de ofertas anteriores. Va acumulando el histórico de ofertas realizadas, generando una función de distribución que nos da la probabilidad de que una función sea aceptada. La predicción también tiene en cuenta el número de días que tiene el agente para entregar el pedido desde el momento que se acepta su oferta.

Producción: El agente utiliza una heurística greedy (voraz) consistente en ordena las ofertas aceptadas por su beneficio, obtenido como precio de venta menos coste de reposición de componentes y menos las posibles penalizaciones) y trata de programarlas lo más tarde posible. Para su realización tiene en cuenta tanto las órdenes ya confirmadas como las RFQs que prevé que van a ser aceptadas en base a la previsión de la demanda. El horizonte de planificación que utiliza es de 10 días. El programa de producción se genera en el proceso de elaboración de

ofertas para no vender por encima de la capacidad disponible hora de enviar ofertas.

**Venta.** En el proceso de venta el agente debe seleccionar las RFQs a las que le interesa responder. Para ello, el agente define una función a optimizar: el beneficio durante los 12 días siguientes. Dentro de esta función se consideran tanto las RFQs que han sido enviadas como las que el agente prevé que van a ser enviadas.

**Precio de venta:** El agente utiliza un proceso adaptativo sencillo: comienza realizando el primer día las ofertas utilizando el precio de reserva. Día tras día van rebajando los precios que oferta hasta encontrar el precio que produce un beneficio más alto por PC producido. Se utiliza una función de aproximación de la probabilidad de ganar el beneficio esperado.

### 3.2. Southampton SCM

**Autores.** (He et al. 2006). Este agente ha sido desarrollado por el grupo liderado por el Prof. N. Jennings de la University of Southampton

**Descripción general.** El agente se construye como un sistema compuesto por tres sub-agentes. En este caso utilizamos el término sub-agente en lugar de módulo porque cada una de las partes funciona de modo totalmente autónomo, intercambiando información sobre los pedidos que han sido aceptados y sobre el inventario disponible. Las funciones principales de cada uno de los sub-agentes son las siguientes:

- *Customer Agent:* Recibe las RFQs de clientes, las selecciona y responde realizando una oferta a las RFQs seleccionadas.
- *Factory Agent:* Se encarga de la realización de los programas de producción y de envío. Realiza el control de inventario de componentes y de productos terminados.
- *Component Agent:* Es el agente que crea las RFQs a proveedores en función de la demanda prevista y selecciona las ofertas enviadas por estos. Incorpora un módulo de precisión de la demanda (*Demand Predictor*) y otro de seguimiento de precios de compra de componentes (*Price Tracker*).

**Previsión demanda:** La lleva a cabo el módulo *Demand Predictor del Component Agent*. Realiza una regresión lineal con las RFQs de días anteriores que permite extrapolar valores y predecir el número de RFQs que puede haber en un determinado día. Utiliza un horizonte temporal de 35 días.

**Aprovisionamiento:** Tiene una estrategia de aprovisionamiento mixta. Por una parte sigue una estrategia de largo plazo que consiste en realizar pedidos periódicos de cantidades pequeñas fijas que son realizados al inicio del juego. Con esta estrategia se pretende conseguir los menores precios que se ha observado que se ofertan cuando se compra a largo plazo. La cantidad a adquirir a largo plazo se determina como número de componentes necesario para poder cubrir la demanda mínima prevista por el agente. La estrategia de corto plazo trata de mantener el stock de cada componente por encima de un mínimo teniendo en cuenta la previsión de la demanda y los pedidos ya realizados. Se ayuda de módulos *Demand Predictor* y *Price Tracker*.

**Precio de compra.** El módulo *Price Tracker* realiza un modelo sobre los precios futuros que van a ofrecer proveedores. Para ello realizad diariamente una serie test de mercado enviando RFQs que le sirven para conocer la tendencia que van a seguir los precios los próximos días, de modo

que pueda estimar para cada componente cuál es el momento óptimo para realizar un pedido en función de las necesidades que ha previsto.

**Producción.** El agente produce bajo pedido y mantiene unos determinados niveles de stock de componentes. Para realizar la programación de producción sigue una heurística basada en reglas de forma que da preferencia a aquellos pedidos que deban ser servidos de forma inmediata, después a los pedidos con retraso y por último a los pedidos con fecha de entrega posterior. En el caso de que después de programar los pedidos exista capacidad de producción libre, entonces produce PC de cada tipo hasta que llega a un determinado nivel de stock.

**Venta.** Ordena las RFQs recibidas por su beneficio potencial. El beneficio potencial de cada RFQ lo obtiene considerando el Precio de reserva indicado, el precio base de los componentes utilizados y un factor que tiene en cuenta la penalización por retraso. Posteriormente a la hora de seleccionar a qué RFQs enviar ofertas comprueba primero si dispone de producto terminado para poder atender el pedido y en caso contrario si dispone de componentes y capacidad de fabricación suficiente para poder producirlo a tiempo. Si no se dan ninguna de estas dos condiciones, no se envía la oferta, pasando a la siguiente RFQ y repitiendo el proceso hasta que se evalúan todas las RFQs recibidas un determinado día.

**Precio de venta.** Emplea técnicas fuzzy para permitir que el agente adapte sus precios diariamente de modo que pueda utilizar su capacidad al completo e intentando obtener el máximo beneficio. En la última etapa del juego el agente modifica su comportamiento de forma que trata de deshacerse de del inventario de componentes y productos terminado del que dispone.

### **3.3. Deep Maize**

**Autores.** (Kiekintveld et al. 2006). Este agente ha sido desarrollado por el grupo liderado por Michael Wellman de la University of Michigan.

**Descripción general.** Basa su funcionamiento en dos niveles de toma de decisión: Un nivel estratégico, centralizado que trata de realizar un programa de producción a largo plazo óptimo, y un segundo nivel donde se toman las decisiones específicas de cada mercado. El nivel estratégico establece una zona de operación de referencia que la que el agente se debe mantener. Dentro de esta zona existe un equilibrio entre lo que el agente compra y lo que el agente vende. Sus planes a largo plazo buscan que el agente se mueva dentro de esta zona. Fuera de ella el agente se encontraría bien con exceso de inventario o bien con inventario insuficiente para cubrir sus compromisos. El agente implementa un sistema de control que hace que modifica el funcionamiento del agente en caso de que se salga de la zona de referencia.

**Previsión demanda:** A la hora de realizar la previsión de la demanda utiliza un modelo de red bayesiana de forma que le permite proyectar la demanda subyacente y las variables de tendencia en cada uno de los tres segmentos de mercado. Este modelo ha sido seguido por otros agentes como TacTex o MinneTac.

**Aprovisionamiento.** Genera las RFQs en dos fases: Primero establece las RFQs potenciales que tendría que enviar para cubrir las necesidades de los días futuros. Posteriormente, y dado que existe un límite de envío de cinco RFQs de cada tipo de componente, utiliza un algoritmo hill-climbing para reducir el número de RFQs al máximo permitido.

**Precio de compra.** Realiza una estimación de los precios de compra de componentes utilizando una media ponderada de los precios ofrecido en días anteriores. Cada oferta recibida representa

un punto en la curva. Los puntos intermedios se calculan mediante interpolación lineal, dando más peso a los valores más recientes. Así mismo, utiliza los informes periódicos de mercado para mejorar la estimación de la capacidad de los proveedores para cada componente.

**Producción.** Cada día el agente realiza una proyección del programa de producción con un horizonte de varios días. Este le sirve para calcular el valor marginal de los PCs, el valor marginal de los componentes y el programa de producción y envío definitivos. Esto proporciona límites en los precios que el agente puede aceptar para cantidades dadas y fechas de entrega en los mercados de componentes y producto final. Para realizar la proyección utiliza un algoritmo Greedy (voraz) para aproximarse a un programa de producción óptimo, de forma que los recursos (ciclos y componentes) se reservan primero para las órdenes de clientes pendientes. Se programan cantidades adicionales hasta obtener el máximo margen por ciclo. Los beneficios marginales y coste de reemplazo se actualizan en cada momento. El programa de producción se cierra cuando el margen cae por debajo de un determinado nivel. El agente utiliza el primer día de la proyección del programa de producción como programa de producción para el día siguiente, utilizando una heurística donde se ordenan las órdenes primero por fecha de entrega y en segundo lugar por precio base por ciclo de producción.

**Venta.** Selección de RFQs A partir de los datos obtenidos en el programa de producción trata de obtener el conjunto de RFQs que optimice el beneficio esperado. Para ello evalúa la probabilidad de obtener una RFQ, el valor del cada PC la oferta que realiza y la cantidad de PC requerida en cada RFQ.

**Precio de venta.** El agente trata de estimar la probabilidad de ganar una RFQ con un determinado precio de oferta, dado un determinado nivel de ofertas. Para ello utiliza el método *Nearest neighbors learning* utilizando tanto datos de otros juegos como los generados en el desarrollo del juego actual. Las características que utiliza a la hora de buscar situaciones similares incluyen indicadores de la actividad reciente del mercado, las estimaciones de la demanda y de la capacidad de los clientes y estimaciones sobre el estado agregado de los inventarios en toda la cadena de suministro.

### 3.4. MinniTAC

**Autores.** (Ketter et al. 2004) El agente ha sido desarrollado por el grupo liderado por John Collins y Maria Gini de la University of Minnesota

**Descripción general.** Está formado por siete módulos: *Repository*, *Communications*, *Oracle*, *Procurement*, *Production*, *Shipping* y *Sales*. El módulo *Repository* es el elemento unificador. Es el único visible para los demás. Sirve como base de datos interna. Notifica a los otros componentes los cambios en el estado del sistema. El módulo *Communications* gestiona la comunicación con el servidor del juego. Al inicio realiza la adquisición de los parámetros del juego. Diariamente recibe y envía los datos al servidor del juego. El módulo *Oracle* lleva a cabo la previsión de la demanda. El módulo *Procurement* se encarga de realizar el lanzamiento de RFQs a proveedores y el control de los inventarios. El Módulo *Production* realiza el programa de producción y el módulo *Shipping* los envíos. Por último el módulo *Sales* realiza ofertas como respuesta a las RFQ de cliente. (Ketter et al. 2006)

**Previsión demanda.** Utiliza una Red Bayesiana para estimar los parámetros que generan la demanda. Utiliza un sistema basado en el que utiliza el agente Deep Maize. (Kiekintveld et al. 2004).

Aprovisionamiento y venta. Mediante técnicas estadísticas (algoritmo EM) trata de estimar en qué régimen se encuentra cada mercado (saturado, equilibrado y de escasez). Modela el sistema como un proceso de markov, lo que le permite no sólo caracterizar en qué régimen está, sino también estimar la probabilidad de que se produzca un cambio de régimen en el mercado. Esto supone una ventaja para el agente al anticiparse a los cambios que se pudieran producir en el mercado, lo cual tiene ventajas a la hora de planificar compra de componentes, inventarios a mantener, o decisiones sobre aplazamiento o anticipación de venta de productos para aprovechar los mejores precios.

### 3.5. CMieux

Autores. (Michael et al. 2006) Desarrollado por el grupo e-Supply Chain Management Lab. de la Carnegie Mellon University

Descripción general. El agente está formado por cinco módulos: Bidding Module, Procurement Module, Scheduling module, Strategy module y Forecasting module.

Previsión demanda. La previsión de demanda se divide en dos: previsión de la demanda de clientes y previsión del precio. La previsión la demanda trata de estimar los parámetros de media y tendencia del número de RFQs que se generan diariamente en cada segmento de demanda. La previsión del precio se realiza utilizando una regresión por mínimos cuadrados sobre los precios que ha tenido cada producto los días pasados.

Aprovisionamiento. En su diseño se ha buscado expresamente la capacidad de adaptación a las condiciones cambiantes del mercado. Su objetivo es tratar de aprovechar los días en los que cada proveedor ofrece precios más bajos. El agente, por tanto, primero trata de identificar estas oportunidades para después construir una combinación de RFQs que tenga en cuenta tanto las necesidades reales de componentes como las oportunidades a aprovechar por el agente. De forma continua actualiza sus predicciones tanto para la disponibilidad como para los precios de los componentes. El agente es capaz de anticipar posibles rupturas de stock futuras y aprovechar las fluctuaciones de los precios de los distintos componentes.

Precio de compra. Utiliza un algoritmo de búsqueda que optimiza la utilidad de los componentes a adquirir teniendo en cuenta la demanda objetivo a conseguir para los productos terminados, costes de mantenimiento de inventario y costes de oportunidad por no poder construir un PC por falta de algún componente.

Producción. El *Scheduling Module* mantiene un programa de producción con un horizonte de varios días. Este programa incluye tanto unidades ya comprometidas, como las previsiones, así como los niveles de inventario de componentes proyectados en dicho horizonte. El módulo de programación utiliza una heurística que tiene en cuenta tanto el tiempo restante hasta la fecha de entrega del pedido, como la penalización asociada.

Venta. Selección de RFQs. Obtiene del Strategic Module el objetivo de recursos que debe vender. Lo debe vender al máximo precio posible. Mantiene para cada producto una distribución de la probabilidad de ganar una RFQ de un determinado tipo de PC y a un precio dado utilizando un modelo de regresión de árbol de distribución, que toma las RFQs de juegos anteriores.

Precio de venta. El Strategic module estima cual es la cuota de mercado objetivo de forma que se optimice el beneficio total obtenido. Se realiza una previsión de los precios utilizando la técnica árbol de regresión que permite realizar la previsión del precio ofertado a partir del histórico de

ofertas de precios que se han recibido de un determinado proveedor para cada componente.

### 3.6. Mertacor

Autores. Agente desarrollado por el grupo liderado por P.A. Mitkas de la Aristotle University of Thessaloniki (Panos et al. 2006)

Descripción general. El agente está formado por cuatro módulos:

- Bidder: para responder a las RFQ de clientes.
- Scheduler: Programación de las producción y envíos.
- Inventory Manager: gestión del inventario de componentes.
- Procurer: Envío de RFQs y órdenes de compra a los proveedores.

Previsión demanda. No realiza una previsión de la demanda. En este sentido se trata de un agente reactivo. Se sigue un de producción bajo pedido: ensambla los PCs cuando recibe el pedido definitivo del cliente y para los componentes trabaja contra stock.

Aprovisionamiento: Tiene una estrategia mixta. Establece tres tipos de RFQs: *anticipadas*, el agente lanza RFQs con mucha antelación para conseguir mejores precios suponiendo una demanda mínima del componente; *Normales*, para satisfacer los niveles de reorden y permitir ofertar a las RFQs de cliente; y *Críticas*, para satisfacer las órdenes de cliente para poder completar un pedido en el caso que se haya quedado sin uno de los componentes necesario para realizarlo.

Producción: El agente fabrica contra pedido. Realiza un programa de producción con un horizonte de varios días mediante el Factory Simulator: En él incluye las RFQs de las que puede obtener beneficio, que le ha enviado el módulo *Bidder*. Se consideran tanto las RFQs ya aceptadas como las que todavía no lo están. Reserva una parte de su capacidad para poder programar posibles futuras RFQs que generen más beneficio.

Venta. Selección de RFQs. El módulo *Bidder* selecciona entre las RFQs que ha recibido, aquellas que le proporcionan un beneficio potencial.

Precio de venta. El precio de venta se estima mediante un modelo que trata de obtener el precio ganador para una RFQ basándose en el histórico de datos. Para ello utiliza el entorno WEKA de minería de datos.

## 4. Conclusiones

A partir del análisis de los agentes participantes en el TAC SCM podemos destacar algunos aspectos positivos de esta competición:

*Oportunidad de desarrollo de agentes autónomos.* Esta competición ofrece una oportunidad de estudio del comportamiento adaptativo de los agentes. Uno de los puntos interesantes en esta competición es que en cada juego la demanda de los clientes y la oferta de los proveedores, variables generadas por el servidor del juego, es diferente cada vez. Pero, además, los agentes que compiten son distintos y desconocidos a priori para el agente que participa. Una de las características más importantes que hemos encontrado en los agentes situados en las mejores



posiciones es su capacidad de adaptación a las condiciones de cada juego.

*Aplicación de técnicas de diversos campos de conocimiento.* La competición posibilita su utilización como benchmarking de técnicas pertenecientes a diversos campos de conocimiento como la gestión de la cadena de suministro, investigación de operaciones, inteligencia artificial.

*Laboratorio experimental.* Dado que somos capaces de controlar las condiciones en las que compiten los agentes participantes, como la presión de la demanda, nivel de oferta o los agentes que participan, podemos utilizar el diseño realizado para el juego como laboratorio experimental que nos permita obtener conclusiones válidas para futuras ediciones de la competición y extrapolables al mundo real.

En el análisis de los agentes participantes también hemos encontrado algunos aspectos negativos que se deberían tener en cuenta en próximas ediciones:

*Posible tendencia a la uniformidad en las soluciones después de varias ediciones.* Hemos observado convergencia de algunos agentes en la resolución de determinados problemas como el modelo de gestión de stock o las técnicas de previsión de la demanda, donde varios agentes utilizan la misma técnica. Esto puede llevar al agotamiento de la propia competición. Proponemos como solución para incentivar la investigación realizar cambios en algunos de los aspectos que parezcan ya suficientemente explotados.

*Soluciones muy específicas.* En muchos de los casos las soluciones que se buscan son muy específicas del juego. Se debería incentivar que los agentes buscaran soluciones más generales, abiertas a otras posibles condiciones más allá del ámbito de la competición.

## Referencias

He, M. H., A. Rogers, et al. (2006). Designing and evaluating an adaptive trading agent for supply chain management. *Agent-Mediated Electronic Commerce: Designing Trading Agents and Mechanisms* Vol. 3937 No., pp. 140-156.

Ketter, W., J. Collins, et al. (2006). Identifying and forecasting economic regimes in TAC SCM. *Agent-Mediated Electronic Commerce: Designing Trading Agents and Mechanisms* Vol. 3937 No., pp. 113-125.

Ketter, W., E. Kryzhnyaya, et al. (2004). MinneTAC sales strategies for supply chain TAC. *Third International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2004*, New York.

Kiekintveld, C., J. Miller, et al. (2006). Controlling a supply chain agent using value-based decomposition. *EC '06: Proceedings of the 7th ACM conference on Electronic commerce*, ACM.

Kiekintveld, C., M. P. Wellman, et al. (2004). Distributed feedback control for decision making on supply chains. *Fourteenth International Conference on Automated Planning and Scheduling*.

Michael, B., S. Alberto, et al. (2006). CMieux: adaptive strategies for competitive supply chain trading. *SIGecom Exch.* Vol. 6 No. 1, pp. 1-10.

Panos, T., K. Dionisis, et al. (2006). Mertacor: a successful autonomous trading agent.

Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems. Hakodate, Japan, ACM Press.

Pardoe, D. and P. Stone (2006). Predictive Planning for Supply Chain Management International Conference on Automated Planning and Scheduling, Cumbria (UK).

Sadeh, N., R. Arunachalam, et al. (2003). TAC-03 - A supply-chain trading competition. *Ai Magazine* Vol. 24 No. 1, pp. 92-94.