

Construcción de agentes para el TAC SCM Game 2005

Daniel Macías Galindo¹, Fabiola López y López², Darnes Vilariño Ayala²

¹Escuela de Ingeniería en Computación, Universidad Popular Autónoma del Estado de Puebla.

²Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla.
deilan.exe@gmail.com, fabiola.lopez@siu.buap.mx, darnes@cs.buap.mx

Resumen. Con el fin de estudiar mercados dinámicos donde diferentes proveedores y consumidores interactúan, ha sido creada una plataforma que permite la competencia de agentes que trabajan en ambientes de cadenas de producción (The Trading Agent Competition Supply Chain Management Game ó TAC SCM). En ésta competencia, agentes proveedores de computadoras se esmeran en ganar pedidos de clientes y al mismo tiempo en asegurarse que los proveedores les garanticen los componentes básicos que les permitan satisfacer esos pedidos. Al final los agentes ganadores son aquellos que obtengan la máxima ganancia en condiciones cambiantes del mercado. En este artículo se proponen tres agentes con diferentes estrategias capaces de competir en este tipo de juego. Se muestran los resultados de cada agente sobre la plataforma de juego.

1 Introducción

Una cadena de producción se define como el desglose de actividades de flujo y transformación de bienes y productos, desde los componentes básicos hasta la disponibilidad de un producto final en los aparadores [1, 7, 8]. La característica principal de una cadena de producción es que cada actividad puede ser realizada por varias entidades totalmente independientes, situando a una empresa como un eslabón de una cadena. Este esquema implica que mucho del éxito de una empresa para fabricar un bien depende de que tan confiables sean sus proveedores quienes le brindan los componentes básicos para elaborar sus productos. Sólo así, la empresa podrá ofrecer un producto de buena calidad, en el menor tiempo posible, y al costo más bajo. De esta manera se tiene que lograr una administración que genere resultados rápidos, económicos y confiables, que puedan ajustarse a las variaciones en el mercado más complejo, dependiendo de la dimensión del mismo.

Las cadenas de producción observadas en la economía de hoy en día son estáticas, puesto que se basan en relaciones duraderas entre cliente y proveedor. Esta forma de trabajo no se ha cambiado debido a la complejidad de organizar a un amplio sector industrial local, por lo tanto es mucho más complejo organizar un sector económico mundial. La computación promueve implementar tecnologías que permitan prácticas más flexibles y dinámicas, ayudando al cliente a seleccionar al proveedor que le garantice ser la mejor opción, basándose en la información generada por el mercado.

Los agentes de software, por sus características de fácil adaptación al medio ambiente han sido considerados como elementos claves en el modelo de implementación de procesos productivos, que semejen a las cadenas de producción. Un agente es un ente computacional, el cual puede describirse por sus propiedades, que lo diferencian de programas convencionales [2]:

- Es autónomo, porque para actuar no necesita la intervención de un usuario.
- Tiene cierto grado de inteligencia, que va desde reglas hasta motores de aprendizaje que le permiten adaptarse al medio.
- Tiene habilidad social, por lo que puede comunicarse con usuarios, el sistema y otros agentes, según lo requiera.
- Tiene capacidad cooperativa, para realizar tareas complejas en equipo.

Los Laboratorios de Administración de Cadenas de Proveedores de la Universidad de Carnegie Mellon y del Instituto Sueco de Ciencias Computacionales (SICS) han estudiado los problemas presentados en las cadenas de producción desde hace más de seis años [3,4,5]. Para ello se dieron a la tarea de implementar una plataforma que se encarga de simular ambientes de cadenas de producción, donde las condiciones del mercado varían continuamente. Para ello también desarrollaron un conjunto de herramientas que permiten desarrollar agentes que participen en las simulaciones. Es así como nace el TAC (acrónimo de The Trading Agent Competition). Este grupo de investigación realiza cada año un concurso, en el cual invita a universidades y empresas interesadas en el tema de las cadenas de producción, para la implementación de un agente que participe en las simulaciones del TAC SCM (Supply Chain Management).

Este concurso propone la competencia entre seis agentes durante un juego de 55 minutos, donde el agente que obtenga la suma económica más alta es el ganador. Cada agente simula ser el administrador de una fábrica ensambladora de computadoras que debe satisfacer las órdenes de diferentes clientes respecto a equipos de cómputo. Cada agente puede producir 16 tipos de computadora y debe ser capaz de decidir qué órdenes aceptar en base a sus capacidades de producción y al inventario disponible. Los componentes básicos son obtenidos a través de diferentes proveedores los cuales no siempre están disponibles. Cada agente emplea las más diversas estrategias para seleccionar a sus clientes, solicitar componentes a los proveedores y ensamblar los equipos, de tal manera que se tengan mayores ganancias que pérdidas. Por ello, el problema general no necesariamente se puede enmarcar en una de las áreas específicas de la optimización sino que envuelve varias de ellas, que incluyen teoría de juego, programación dinámica y simulación.

El objetivo de este artículo es describir la competencia, los elementos que intervienen en la misma y presentar tres diferentes estrategias que permitan a un agente competir en el TAC SCM 2005. Se presentan los resultados de cada estrategia cuando el agente compitió con los 5 agentes presentados por la plataforma.

El artículo se organiza de la siguiente manera. En la sección 1 se muestra una breve reseña del concurso TAC SCM. La sección 2 profundiza la descripción del juego, así como la información que se envía entre agentes y el servidor. La sección 3, da una pequeña descripción de los paquetes utilizados para implementar un agente que participe en el TAC SCM. La sección 4 presenta la forma en que hemos abordado la solución del problema. La sección 5 contiene una breve descripción de las tres

diferentes estrategias propuestas, las cuales han sido diseñadas e implementadas. Finalmente, en la sección 6, se proporcionan las observaciones y conclusiones con respecto a este trabajo.

2 El TAC SCM

La competencia se lleva a cabo en una plataforma desarrollada en Java que corre en diferentes servidores localizados en Pittsburgh, Pennsylvania y que puede descargarse de Internet para emplearla de manera local. Esta plataforma se encarga de:

- Albergar a los agentes participantes.
- Activar a los agentes al momento de iniciar un juego TAC, y de removerlo una vez que éste termina.
- Proporcionar la información relevante durante el juego.

Entre los datos que la plataforma proporciona, se encuentran los valores de configuración para identificar al agente, y los datos de trabajo para que éste pueda desempeñarse en la simulación, como son la disponibilidad de componentes de los proveedores, los comportamientos de los clientes, la duración de los días TAC (que generalmente es de 15 segundos), etc. Todos estos valores son cargados en el *día de configuración*, o día 0 de la competencia.

De esta manera, los seis agentes que participan en un juego TAC SCM deberán disputarse las solicitudes de los clientes a través de 220 días TAC. Cada día un agente recibe la siguiente información:

- **De los clientes:** Una lista de requerimientos de cotizaciones de computadoras de diferentes tipos (llamada en lo sucesivo cRFQ); así como los pedidos aceptados por los clientes en respuesta a ofertas que les fueron enviadas el día anterior (si éstas fueron hechas, en lo sucesivo se identificarán como cORD). Además, al recibir su mercancía completa, un cliente puede realizar un depósito a la cuenta del agente, por el monto de la compra de los equipos de cómputo (cDEP).
- **De los proveedores:** Respuesta a las solicitudes del día anterior de cotizaciones de los componentes básicos por parte de proveedores (si éstas fueron enviadas, son las sOFF), así como el envío de componentes que fueron ordenados con anterioridad (sDEL).
- **Del banco:** Estado de cuenta.
- **De la fábrica:** Reporte de inventario: la cantidad de componentes básicos, así como de computadoras que ya han sido ensambladas.

Así mismo, en ese día, el agente deberá estar dotado de métodos que le permitan enviar:

- **A los clientes:** Las solicitudes que el agente considere que pueda satisfacer en tiempo y forma (aOFF).
- **A los proveedores:** Las solicitudes de los componentes que necesita para ensamblar los equipos de cómputo pedidos por el cliente (aRFQ), así como las confirmaciones (aORD) para las ofertas hechas por los proveedores. Al recibir el

pedido enviado por el proveedor, hacer un retiro de su cuenta bancaria para pagar a su proveedor (aRET).

- **A la fábrica:** Las órdenes de ensamblado de equipos (aENS), así como las órdenes de entrega de equipos a sus clientes (fDEL). Estos dos últimos factores son llamados *calendario de producción* y *calendario de entrega* respectivamente. El primero se refiere a que una vez que se tengan los componentes se solicita a la fábrica ensamblar los equipos necesarios para satisfacer una orden, y la segunda a que una vez que se tenga la disponibilidad de la mercancía, se asigne la fecha de entrega para el cliente, la cual debe ser menor o igual a la que se le proporcionó en la cotización. Si la orden no ha sido satisfecha en tiempo, el agente cuenta con cinco días en los cuales obtendrá una penalización del cliente por no entregar la mercancía, antes de que la orden sea cancelada.

En la Figura 1 se muestran de manera resumida las actividades diarias que se realizan en el TAC SCM. Los círculos de color gris contienen mensajes que el agente recibe de las diferentes entidades del juego; los círculos en color blanco contienen los mensajes que el agente envía a estas entidades.

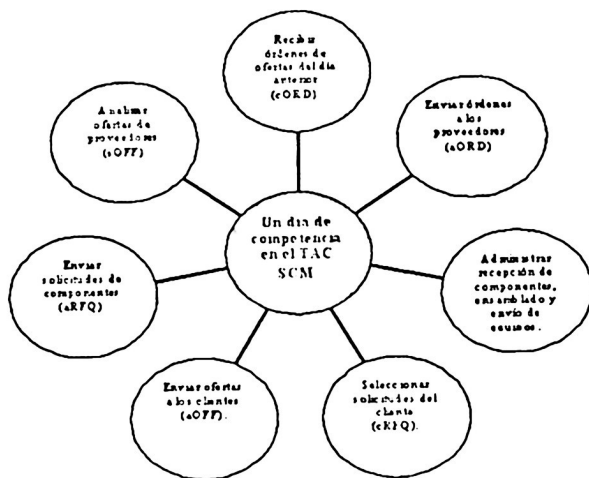


Fig. 1. Actividades que el agente realiza diariamente durante el TAC SCM.

Cada agente tiene su propio almacén de componentes y de equipos, dentro de la fábrica, la cual se dedica al ensamblaje de las computadoras y está limitada por la cantidad de ciclos de trabajo máximos que puede emplear para ensamblar los equipos de cómputo (generalmente 2000 e igualmente dados a conocer en el día 0).

Hay cuatro componentes para tener una PC ensamblada y lista para su venta: CPU, Tarjeta Madre, Memoria y Disco Duro, cada una de ellas son de dos tipos diferentes. Además se disponen de restricciones para el ensamblaje respecto a los CPU y las tarjetas madre, con los cuales se obtienen 16 configuraciones de equipos diferentes.

El juego representa una amplia gama de situaciones de provisión de servicios. El reto del juego está en establecer una estrategia que se ajuste a los cambios en cada simulación, donde el agente demuestre sus habilidades para reaccionar a las variaciones que presentan tanto el posible cliente, como el proveedor, así como adaptarse a las estrategias adoptadas por los otros agentes en competencia.

En la Figura 2, se describe más detalladamente el paso de mensajes de manera ideal en el juego TAC SCM. Los cuadros en negro representan las entidades externas al agente, los cuadros grises al agente y a sus componentes; las flechas indican el paso de mensajes de la entidad emisora a la receptora; los colores de las flechas van de claro a oscuro, entre más claro, es un mensaje que ocurre antes que un mensaje de color oscuro.

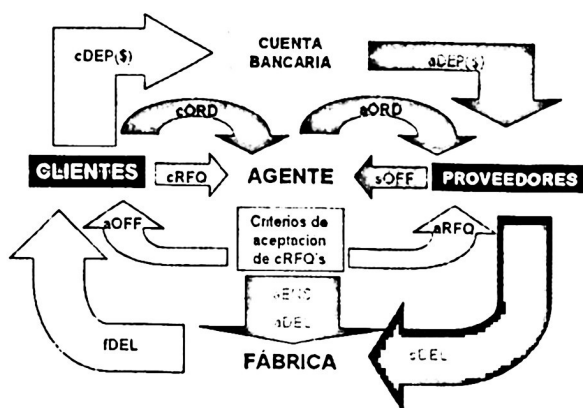


Fig. 2. Diagrama de las actividades que se realizan en el TAC SCM.

Los agentes en este medio ambiente, entonces, deben ser capaces de realizar las siguientes tareas en cada día TAC:

1. Negociar contratos de proveedores.
2. Buscar satisfacer las órdenes del cliente, ofreciéndole los mejores precios en el menor tiempo de entrega.
3. Administrar actividades diarias de ensamblaje y envío dentro de la fábrica.

3 Paquetes para el desarrollo del agente

La paquetería para implementar un agente capaz de competir en el TAC SCM se encuentra disponible al público en el sitio Web del TAC [6]. A continuación se realiza una breve descripción de lo que se puede encontrar en cada paquete.

3.1. Paquete se.sics.tasim.props

Este paquete es el encargado de recepción-envío de mensajes entre el agente y el servidor de juego, es decir entre el agente y las entidades del juego (clientes, proveedores y la fábrica). Cada mensaje está formado por el nombre del emisor, del receptor y el contenido. El contenido del mensaje describe el propósito de la comunicación. Cabe destacar que toda la comunicación (exceptuando los nombres del emisor y receptor, cuyo formato es de String) se realiza con números enteros, lo cual permite una comunicación cliente-servidor más fluida y directa.

3.2. Paquete se.sics.tasim.aw

El paquete incluye clases básicas para implementar al agente participante en estas simulaciones.

3.3. Paquete se.sics.tasim.tac03.aw

Incluye, principalmente, el visualizador del usuario diseñado desde el TAC SCM 2003, el cual es mostrado en la figura 3.

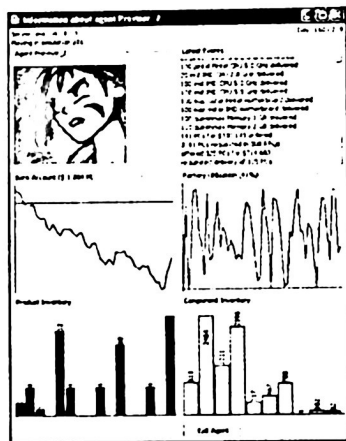


Fig. 3. Visualizador para el usuario del agente Previsor_2.

4 Consideraciones iniciales

La implementación de un agente que se desenvuelva de manera óptima a través de las simulaciones, requiere comenzar a analizar la solución de un problema que puede abordarse usando diferentes estrategias de optimización. Uno de los objetivos de este

proyecto consiste en proveer al agente de una política que le permita obtener el máximo de utilidades en cada juego TAC. En este trabajo se discuten los resultados obtenidos con 3 estrategias que implementan políticas de compra y ventas diferentes, esto nos permitirá confeccionar un agente para la competencia, que pueda de alguna manera cambiar su comportamiento de acuerdo a las condiciones del mercado y de sus ganancias obtenidas.

Los resultados de esta competencia permitirán establecer nuevas relaciones productivas a nivel mundial en la relación entre clientes y proveedores. A continuación se presenta la estructura que conformará a los agentes.

4.1 Diseño del agente

El agente debe enfocarse a tres actividades principalmente en cada uno de los días de la simulación:

- Seleccionar clientes seguros para obtener órdenes de compra.
- Seleccionar sus proveedores para surtir su almacén.
- Administrar las actividades del almacén.

4.1.1. Selección de clientes.

Cada día se recibirá un conjunto de solicitudes de cotizaciones por los clientes (cRFQ). El agente debe ser capaz de seleccionar una población menor de esa lista, considerando que son órdenes que él podrá satisfacer, para lo cual verifica las siguientes condiciones:

- Que los componentes para satisfacer una solicitud estén disponibles en el almacén (dentro de la clase `InventoryStatus` existen métodos para verificar la existencia de componentes y de PCs ensambladas).
- Que los componentes puedan adquirirse en los días sucesivos, preferentemente antes de recibir penalizaciones por entregas retrasadas, si la orden es aceptada (se debe hacer una solicitud, aRFQ, al proveedor correspondiente, apoyado en un método de la clase `SCMAgent`).
- Según se establezca el modelo de solución, se pueden definir uno o más criterios para seleccionar a los clientes; ya sea por fecha de entrega, por precios, cantidad de equipos, o por la combinación de algunos de ellos.

Basándose en esto, el agente debe generar una oferta para cada cliente cuyo pedido puede satisfacer, y al menor precio posible para ganar la orden respecto a los otros agentes, consultado un objeto de la clase `MarketReport` (generada cada 20 días, donde se incluyen el menor y mayor precio ofertado para un tipo de PC). Hay que tomar en cuenta que un cliente aceptará una oferta si ésta viene por la cantidad requerida, si se entrega antes o en la fecha requerida, y si el precio ofrecido es el menor de todas las ofertas hechas por los agentes.

4.1.2. Negociación con proveedores.

Para ello se realizará una planeación de compras, en la cual se deben tomar en cuenta los siguientes factores:

- Los pedidos de los clientes que están pendientes. (Dependiendo del momento en que se haga la compra, sería un objeto de la clase **RFQBundle** o de **OrderBundle**).
- Los pedidos de componentes básicos a los proveedores que aún no han sido satisfechos,
- El gasto que el agente realizará para adquirir estos componentes, (se puede analizar un objeto de la clase **BankStatus**).

Hay que aclarar que esta planeación tiene que contemplar el panorama futuro, y predecir los comportamientos del mercado. Entonces la planeación de compras consiste en que el agente:

- Decide qué comprar (según órdenes pendientes, almacén y estado de su cuenta).
- Decide a quién comprar (según la factibilidad de que el proveedor pueda satisfacer la orden, que se adquiere analizando la preferencia que tiene cada proveedor respecto hacia el agente).
- Decide en qué momento comprar (si lo hace antes de recibir las confirmaciones del cliente, o en el momento de recibirlas).

4.1.3. Administración del almacén.

El agente debe estar capacitado para manejar sus propias actividades de ensamblaje, las cuales:

- Dependen de los componentes en existencia en almacén (hay un método en la clase **SCMAgent**, llamado *addProductionSchedule*, donde se hace este análisis y si pueden armarse la cantidad especificada de PCs, ésta orden se añade al ensamblado del día siguiente en la fábrica).
- Deben satisfacer las órdenes de compra ya hechas de los clientes al agente (igualmente, con el método *addDeliverySchedule*, se analiza la posibilidad de que un pedido completo sea enviado al cliente al día siguiente de que se da la orden).

4.1.4. Administración de la cuenta bancaria.

Con el banco solamente se tendrá la interacción para conocer el estado de cuenta, y saber si el agente es capaz de comprar más componentes, y al final, si es el ganador de la ronda.

5 Estrategias implementadas

Hasta el momento se han implementado tres agentes, con dos estrategias completamente diferentes, con los cuales se ha intentado conocer el funcionamiento de la plataforma, aprovechar la información que se envía en los mensajes y agregar

más situaciones que no han sido contempladas. Todo esto con la intención de tener un agente fuerte, versátil e inteligente para la competencia mundial.

A continuación se discutirán brevemente estas estrategias, y se hará una comparativa de los resultados obtenidos con cada agente.

5.1 Agente Loco_Avorazado

Este agente es la primera implementación realizada para comenzar a entender el funcionamiento de la plataforma. Su funcionamiento se puede resumir en los siguientes puntos:

- Selecciona a los clientes tomando el día de entrega de la orden; éste debe ser igual a ocho días después del día TAC actual.
- Oferta las solicitudes de los clientes al 95% del precio sugerido por el mismo cliente.
- Solicita los componentes el mismo día que envía las ofertas a los clientes.
- Selecciona a su proveedor de manera aleatoria.

RESULTADOS:

- Toma el 10% de las solicitudes totales enviadas cada día.
- Se utiliza en promedio el 24% de la fábrica.
- El cliente prefiere a este agente en el 96% de las ofertas hechas.
- Ganancias: de \$4 M a \$8.2 M.

Result for game 197@localhost played at 2005-03-25 01:23:00

Played with TAC SCM 04 rules (February 2004 specification)

Player	Revenue	Interest	Costs				Margin 1	Margin 2	Result
			Material	Storage	Penalty				
Dummy	82 785 964	1 058 250	57 931 535	231 890	373 619	1%	30%	31%	25 307 170
Dummy-2	84 562 422	1 065 005	59 447 116	239 713	762 266	1%	29%	30%	25 178 332
Dummy-4	86 274 490	1 064 240	61 070 658	247 744	963 993	2%	29%	29%	25 056 335
Dummy-3	80 504 809	1 034 900	56 238 763	228 033	793 775	1%	30%	30%	24 279 138
Dummy-5	82 253 921	928 198	58 402 467	247 182	3 006 001	5%	29%	26%	21 526 469
Loco_Avorazado	42 706 142	319 508	30 791 591	96 723	4 359 310	12%	28%	18%	7 778 026

Download game data [here](#)

Player	Orders	Utilization	Deliveries (on time/late/missed)	DPerf
Dummy	3846	49%	3715 / 131 / 0	97%
Dummy-2	3910	50%	3629 / 281 / 0	93%
Dummy-4	4029	51%	3738 / 291 / 0	93%
Dummy-3	3797	48%	3533 / 264 / 0	93%
Dummy-5	3789	49%	3107 / 682 / 0	82%
Loco_Avorazado	2210	27%	1711 / 354 / 145	77%

Fig. 4. Resultados del agente Loco_Avorazado.

5.2 Agente Previsor_2

- Selecciona clientes respecto a una función entre precio sugerido y cantidad de equipos a comprar.
- Su inventario se obtiene *a priori*, entonces tiene que asignar de manera óptima tanto éste como su trabajo en la fábrica: por lo que se crea un problema de optimización: maximizar ganancias, sujetándose a la cantidad de componentes en inventario y ciclos de trabajo libres en la fábrica.
- Funciona con un algoritmo selectivo, que ordena los pedidos y decide qué órdenes pueden ser satisfechas con nuestras limitantes (inventario y fábrica).
- La producción de los equipos de cómputo se realiza inmediatamente que se envían las ofertas al cliente.

RESULTADOS:

- Aceptación del cliente del 99%, debido al precio ofertado del 92%.
- Existe un remanente en inventario tanto de componentes como de equipos no ensamblados.
- Los equipos que no se entregan en tiempo, nunca son enviados a sus clientes (posiblemente debido a un error en el esquema de análisis de entregas).
- Ganancias variables, desde saldo negativo hasta \$14 M.
- Su falta de constancia no es suficiente para competir contra los *Dummies* (agentes demo que corren en la plataforma).

Result for game 136@localhost played at 2005-03-02 05:47:00

Played with TAC SCM 04 rules (February 2004 specification).

Player	Revenue	Interest	Costs				Margin 1	Margin 2	Result
			Material	Storage	Penalty				
Dummy-2	74 073 709	657 047	55 017 982	198 405	1 801 001	3%	25%	24%	17 713 368
Dummy-3	71 691 196	657 370	52 661 800	200 297	2 556 928	5%	26%	24%	16 929 541
Dummy	72 753 713	640 361	53 898 947	206 941	2 359 649	4%	26%	23%	16 928 537
Dummy-4	68 487 339	603 112	49 963 677	190 860	2 102 179	4%	27%	25%	16 833 735
Dummy-5	71 563 973	589 644	52 468 135	205 140	3 463 654	6%	26%	22%	16 016 688
Previsor_2	90 263 967	104 893	72 671 940	904 350	12 700 665	15% 18%		5%	4 091 905

Download game data [here](#)

Player	Orders	Utilization	Deliveries (on time/late/missed)	DPerf
Dummy-2	3718	44%		3194 / 520 / 4 86%
Dummy-3	3534	43%		2868 / 664 / 2 81%
Dummy	3701	43%		3127 / 533 / 41 84%
Dummy-4	3489	41%		3050 / 417 / 22 87%
Dummy-5	3617	43%		2918 / 682 / 17 81%
Previsor_2	2834	54%		2246 / 0 / 588 79%

Fig. 5. Resultados del agente Previsor_2.

5.3 Agente Previsor_3

- Su operación es similar a Previsor_2.
- Implementa la compra de los componentes de acuerdo al inventario y a cuotas máximas y mínimas, así como la siguiente regla: tomando en cuenta 4 diferentes procesadores, se compran menos procesadores que los demás componentes.
- Con esto ajusta sus compras, quedando con un inventario todavía menor al final del juego que Previsor_2.
- Limita compra de componentes y ofertas en el momento adecuado antes del final del juego.

RESULTADOS:

- Ganancias de \$29 a \$39 M, primer lugar.
- Todos los pedidos se entregan en tiempo y forma, no hay penalizaciones.
- Remanente de inventario en componentes, más no en equipos ensamblados.
- Precios ofertados al cliente, fijos.

Result for game 181@localhost played at 2005-03-11 03:02:00

Played with TAC SCM 04 rules (February 2004 specification)

Player	Revenue	Interest	Costs			Margin 1	Margin 2	Result
			Material	Storage	Penalty			
Previsor_3	127 286 487	1 120 193	91 002 270	183 919	0	0% 28%	29%	37 220 491
Dummy-3	70 341 194	807 564	48 507 512	140 282	1 607 982	3% 31%	30%	20 892 982
Dummy	70 174 845	816 435	48 173 347	139 362	1 864 515	4% 31%	30%	20 814 056
Dummy-5	70 010 335	805 223	48 171 366	138 945	1 728 061	3% 31%	30%	20 777 186
Dummy-4	71 904 724	775 918	49 765 980	143 578	2 119 062	4% 31%	29%	20 652 022
Dummy-2	67 717 514	757 742	46 409 527	133 241	1 412 388	3% 31%	30%	20 520 100

Download game data [here](#)

Player	Orders	Utilization	Deliveries (on time/late/missed)	DPerf
Previsor_3	3530	78%	3530 / 0 / 0	100%
Dummy-3	3626	42%	3174 / 452 / 0	88%
Dummy	3623	42%	3056 / 567 / 0	84%
Dummy-5	3617	41%	3107 / 510 / 0	86%
Dummy-4	3750	43%	3205 / 526 / 19	85%
Dummy-2	3523	40%	3094 / 419 / 10	88%

Fig. 6. Resultados del agente Previsor_3.

6 Conclusiones

Para analizar el comportamiento de las estrategias propuestas, se lanzaron los 3 agentes de manera local a competir con los *dummies* (agentes de prueba) que posee la

plataforma. Los resultados de cada uno de los juegos se muestran en las figuras 4, 5 y 6. Se comprobó que la estrategia de compra y venta que se implementó en Loco_Avorazado no permite ganancias sustanciales en el mercado cambiante, por su parte para encontrar una solución del modelo propuesto en Previsor, se utilizó un algoritmo de Monte Carlo, sin embargo no siempre se encontró la solución en el tiempo adecuado (antes de 15 segundos), por lo que se perdieron un buen número de clientes y se decidió por la estrategia selectiva implementada en Previsor_2.

En Previsor_3 se trató de eliminar los problemas presentados con los otros agentes (Loco_Avorazado y Previsor_2), esto permitió lanzar al agente al primer lugar en la competencia contra los *Dummies*. Sin embargo aún se tienen problemas en el manejo del inventario respecto a computadoras ensambladas y a la cantidad de componentes de determinados tipos, que aún quedan en el almacén, esto permitió el diseño de una nueva estrategia Previsor_Avorazado que se encuentra en fase de implementación.

Próximamente la plataforma en Internet lanzará los agentes ganadores del año anterior, con lo que pretendemos probar todas nuestras estrategias y finalmente proveer al agente de diferentes mecanismos que le permitan modificar su comportamiento de acuerdo al mercado cambiante y al número de ordenes que logre ganar.

Los resultados han sido satisfactorios, de agentes perdedores logramos diseñar un agente que siempre ocupe el primer lugar. Hasta el momento no se han incluido técnicas fuertes de optimización, Previsor_Avorazado ya considera técnicas de juego con lo que ya estaremos en condiciones de enfrentarnos a los agentes ganadores del año anterior. Por primera vez México participará en dicha competencia mundial.

Referencias

1. Hellriegel, Jackson y Slocum. "Administración: Un enfoque basado en competencias", Editorial Thomson, 9a Edición, pp. 58-59
2. H. Nwana, Software Agents: An Overview, The Knowledge Engineering Review, Volume 11, Number 3, pp. 205-244. 1996.
3. N. M. Sadeh, R. Arunachalam, J. Eriksson, N. Finne and S. Janson, "TAC'03: A Supply Chain Trading Competition", AI Magazine, Spring 2003.
4. Arunachalam, R., Sadeh, N., Eriksson, J., Finne, N., and Janson, S., "Design of the Supply Chain Trading Competition", IJCAI-03 Workshop on "Trading Agent Design and Analysis", Mexico, August 2003.
5. Arunachalam, R., Sadeh, N., Eriksson, J., Finne, N., and Janson, S., "The TAC Supply Chain Management Game", 2003 CMU-CS-03-184 tech. Report
6. Sitio web de la Trading Agent Competition: <http://www.sics.se/tac/page.php?id=1>
7. Gómez Escobar, Ignacio; "La logística en el mercadeo", ABC Formación: http://www.abcformacion.com/contenidos/logistica_0006.htm
8. López, Carlos; "Administración de la cadena de abastecimiento": <http://www.gestiopolis.com/canales/gerencial/articulos/26/aca.htm>