

TRABAJO PRÁCTICO Nº 1



INTELIGENCIA ARTIFICIAL

PROFESORES:

Gutiérrez, Milagros

Roa, Jorge

GRUPO Nº9

INTEGRANTES:

Alfredo L. Benassi –

alfrebenassi@gmail.com

Maximiliano D'Amato –

maximilianodamato@hotmail.com

Santiago Fernández – santif@gmail.com

Fecha: 18/05/2011

INDICE

RESUMEN	3
INTRODUCCIÓN	3
SOLUCIÓN	6
RESULTADOS	13
CONCLUSIÓN	22
REFERENCIAS	23

RESUMEN:

En el presente trabajo se pretende simular un agente inteligente que se relacione con el mundo que lo rodea. Dicho agente utilizará estrategias de búsqueda convenientes para tomar decisiones sobre las acciones que debe realizar.

Nuestro agente se desenvuelve en un entorno B2B (Business to Business) que intercambia mensajes, actividades y flujos. La detección de cada uno de ellos le permite determinar la acción correspondiente que lo acerque al objetivo en el menor tiempo posible.

INTRODUCCIÓN:

El entorno B2B consiste de un cliente, Computer's Market, y un proveedor, TK Computers, especializados en la fabricación y venta de computadoras. Ambos colaboran para mejorar sus beneficios y disminuir sus costos mediante un pronóstico de demanda y un plan de suministro entre ellos.

Estos procesos consisten de sucesivas negociaciones que implican el envío de mensajes, creando un flujo de información monitoreado por nuestro agente.

Su tarea es percibir estos intercambios, y en base a ellos determinar las acciones correspondientes sobre el ambiente. El objetivo de las negociaciones es llegar a un estado de éxito donde cliente y proveedor acuerden un mutuo abastecimiento de computadoras portátiles.

El proceso de negocio a modelar se puede apreciar en las siguientes figuras:

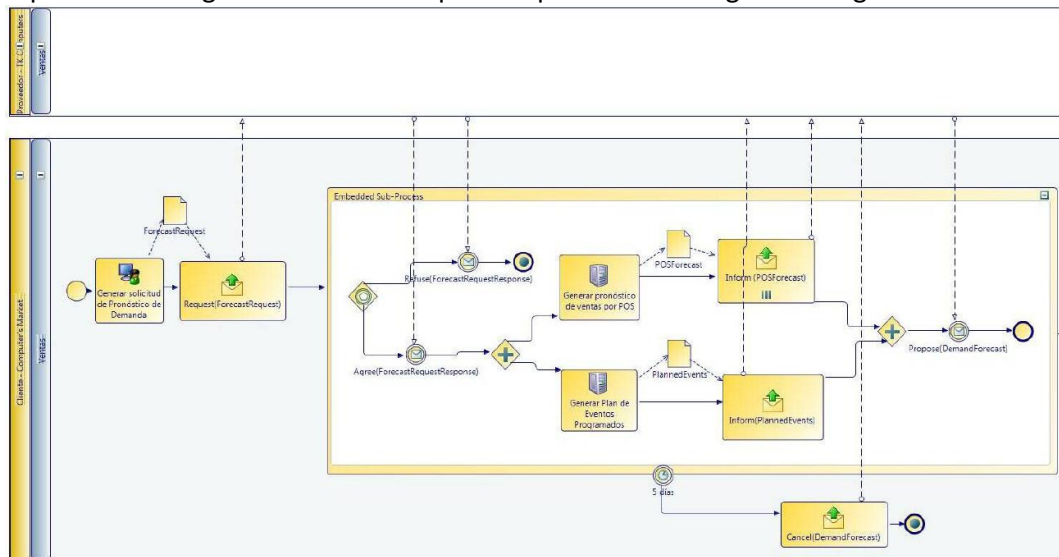


Figura 1. Proceso de negocio del cliente.

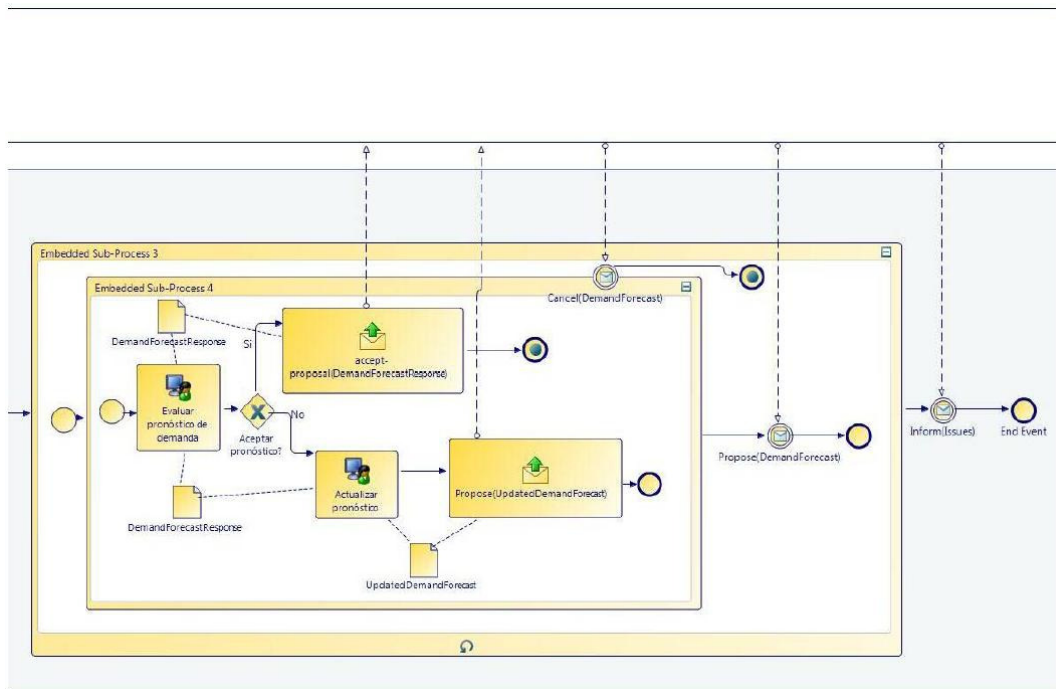


Figura 2. Proceso de negocio del cliente (continuación).

Las configuraciones de escenarios que se van a utilizar para que el agente propuesto resuelva el problema de ejecutar el proceso gestión de pronóstico de demanda colaborativo se pueden observar en las siguientes tablas:

La tabla 1 muestra el tiempo de demora de cada actividad.

Actividad	Tiempo [hs.]
Generar solicitud de pronóstico de demanda	10-15
Generar pronóstico de ventas por POS	12-96
Generar Plan de Eventos Programados	30-150
Evaluar pronóstico de demanda	10-20
Actualizar pronóstico	2-4

Tabla 1. Tiempos de finalización de ejecución de las actividades del cliente

La tabla 2 muestra el tiempo de demora de cada envío de mensaje.

Mensajes de envío	Tiempo [hs.]
Request(ForecastRequest)	1-3
Inform (POSForecast)	1-2
Inform(PlannedEvents)	2-4
Cancel(DemandForecast)	-
accept-proposal(DemandForecastResponse)	1-3
Propose(UpdatedDemandForecast)	1-2

Tabla 2. Tiempos de ejecución de las actividades del cliente

La tabla 3 muestra el tiempo absoluto en que se produce cada recepción de mensaje.

Mensajes de envío	Tiempo [hs.]
Refuse(ForecastRequestResponse)	2-4
Agree(ForecastRequestResponse)	2-3
Propose(DemandForecast)	1-3
Cancel(DemandForecast)	-
Propose(DemandForecast)	1-2
Inform(Issues)	1-3

Tabla 3. Tiempos de ejecución de las actividades del cliente

En las subsiguientes secciones se expone la solución al anterior escenario y las conclusiones del problema.

SOLUCIÓN:

Definición conceptual del agente:

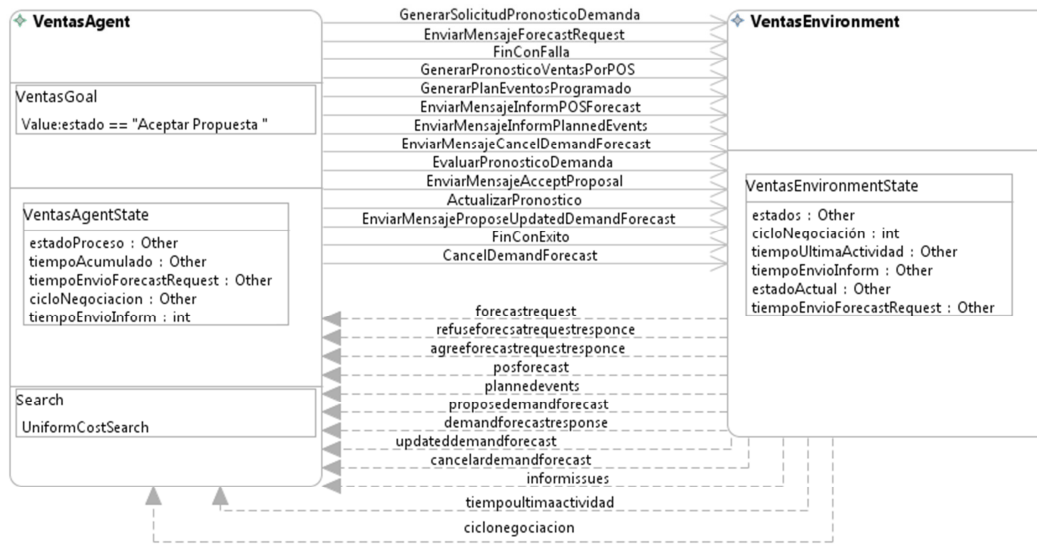


Figura 3. Diagrama IDEM-IA del problema.

En este diagrama se puede ver que el agente, llamado *VentasAgent*, posee las siguientes partes:

VentasAgentState: representa su estado y está compuesto por:

- **estadoProceso**: estado actual del agente.
- **tiempoAcumulado**: cantidad de horas transcurridas.
- **tiempoEnvioForecastRequest**: momento en el que se envía el mensaje `forecastRequest`. Controla que el subproceso primero no exceda los 5 días.
- **cicloNegociacion**: Representa los ciclos de negociación en el cual se encuentra el agente. No puede ser mayor a 5
- **tiempoEnvioInform**: Controla el subproceso cuarto no exceda una semana.

VentasGoal: Representa la meta del agente, en este caso, la aceptación por parte del cliente de la propuesta efectuada por el proveedor.

Search: Representa la búsqueda realizada por el agente. En este caso la de costo uniforme.

Además el ambiente, llamado *VentasEnvironment*, está compuesto por:

VentasEnvironmentState: representa su estado y está compuesto por:

- **estados**: Representa todos los estados posibles
- **estadoActual**: Representa el estado actual del ambiente.
- **tiempoUltimaActividad**: Representa el tiempo requerido para realizar la última actividad.
- **tiempoEnvioForecastRequest**: momento en el que se recibe el mensaje `forecastRequest`. Controla que el subproceso primero no exceda los 5 días.

- cicloNegociacion: Representa los ciclos de negociación en el cual se encuentra el agente. No puede ser mayor a 5
- tiempoEnvioInform: Controla el subproceso cuarto no exceda una semana.

Las acciones están representadas mediante las flechas que se dirigen desde el agente *VentasAgent* hacia el ambiente, *VentasEnvironment*.

Las percepciones son representadas por las flechas punteadas que se dirigen desde *VentasEnvironment* hacia *VentasAgent*.

Definición conceptual del problema de búsqueda: Observando el diagrama del proceso de negocio del cliente, pudimos identificar las percepciones que afectan al agente, las actividades que serán desencadenadas en base a esas percepciones y las realizadas anteriormente (precondiciones).











El estado de nuestro agente está definido de igual manera que el estado actual del entorno, pues lo obtiene de él. Se parte de un estado inicial, llamado *ESTADO_INICIAL*, y el estado final, es *PROCESO_FINALIZADO_EXITO* que se produce cuando se llega a enviar la aceptación del pronóstico propuesto.

La prueba de meta consiste en evaluar si el agente ha llegado al estado óptimo deseado. Hemos definido dicho estado como la aceptación de la propuesta por parte del cliente (*PROCESO_FINALIZADO_EXITO*). Cada vez que el ambiente es modificado por el agente en base a la acción seleccionada, se realiza una prueba de meta para verificar si se ha llegado al estado óptimo deseado.

Los operadores están representados por las actividades internas de la organización:

Se representan las precondiciones y las postcondiciones con el formato Acción-Percepción. En algunos casos la percepción es nula. Y en otros casos se toma como precondición una variable de tiempo.

- ✚ **GenerarSolicitudPronosticoDemanda:** Genera la solicitud de pronóstico de demanda del cliente
Precondición: ESTADO_INICIAL.
Postcondición: GENERAR_SOLICITUD_PRONOSTICO_DEMANDA.
- ✚ **EnviarMensajeForecastRequest:** Envía la solicitud de pronóstico de demanda del cliente al proveedor.
Precondición GENERAR_SOLICITUD_PRONOSTICO_DEMANDA-FORECAST_REQUEST
Postcondición: ENVIAR_FORECAST_REQUEST
- ✚ **GenerarPronosticoVentasPorPOS:** Genera los pronósticos de ventas para los puntos de ventas del cliente.
Precondición: ENVIAR_FORECAST_REQUEST-AGREE_FORECAST_REQUEST_RESPONSE
Postcondición: GENERAR_PRONOSTICO_VENTAS_POS.
- ✚ **GenerarPlanEventosProgramado:** Genera el plan de eventos programados del cliente.
Precondición: ENVIAR_FORECAST_REQUEST-AGREE_FORECAST_REQUEST_RESPONSE
Postcondición: GENERAR_PLAN_EVENTOS_PROGRAMADOS.

-  **EnviarMensajeInformPlannedEvents:** Envía el plan de eventos programados al proveedor.
 Precondición: GENERAR_PLAN_EVENTOS_PROGRAMADOS-PLANNED_EVENTS
 Postcondición: ENVIAR_PLAN_EVENTOS_PROGRAMADOS
-  **EnviarMensajeInformPOSForecast:** Envía los pronósticos de ventas de los puntos de ventas del cliente al proveedor.
 Precondición: GENERAR_PLAN_EVENTOS_PROGRAMADOS-POS_FORECAST
 Postcondición: ENVIAR_INFORME_POSFORECAST.
-  **EvaluarPronosticoDemanda:** Evalúa el pronóstico de demanda hecho por el proveedor.
 precondición1: ENVIAR_PLAN_EVENTOS_PROGRAMADOS-PROPOSE_DEMAND_FORECAST_1
 Precondicion2: ENVIAR_INFORME_POSFORECAST - PROPOSE_DEMAND_FORECAST_1
 Poscondicion1: EVALUAR_PRONOSTICO_DEMANDA.
-  **EnviarMensajeAcceptProposal:** Envía la aceptación de la pronóstico de demanda al proveedor.
 Precondición: EVALUAR_PRONOSTICO_DEMANDA-DEMAND_FORECAST_RESPONSE_YES
 Postcondición: ENVIAR_ACEPTACION_PROPOSTA
-  **ActualizarPronostico:** Genera un nuevo pronóstico de demanda actualizado.
 Precondición: EVALUAR_PRONOSTICO_DEMANDA-DEMAND_FORECAST_RESPONSE_NO
 Postcondición: ACTUALIZAR_PRONOSTICO
-  **EnviarMensajeProposeUpdatedDemandForecast:** Envía el pronóstico de demanda actualizado al proveedor.
 Precondición: ACTUALIZAR_PRONOSTICO-UPDATED_DEMAND_FORECAST
 Postcondicion: ENVIAR_PROPOSTA_PRONOSTICO_ACTUALIZADO
-  **FinConExito:** Establece el final con éxito del proceso, es decir se aceptó el pronóstico del proveedor.
 Precondición: ENVIAR_ACEPTACION_PROPOSTA-SUCCESS
 Postcondición: -
-  **FinConFalla:** Establece el final sin éxito del proceso. Puede ser por demora de 5 días en contestar de parte del proveedor, por demora de una semana en la contestación del cliente durante el ciclo de negociación o por la realización de más de 5 ciclos de negociación.
 Precondición1: CANCELAR_DEMAND_FORECAST
 Precondicion2: ENVIAR_CANCELACION_DEMANDA_FORECAST
 Precondicion3: ENVIAR_FORECAST_REQUEST-REFUSE_FORECAST_REQUEST_RESPONSE
 Precondicion4: ENVIAR_PROPOSTA_PRONOSTICO_ACTUALIZADO-INFORM_ISSUES
-  **EnviarMensajeCancelDemandForecast:** Envía una cancelación de parte del cliente al proveedor por superar el plazo de 5 días para contestar al envía de la solicitud de pronóstico de demanda.
-  **CancelDemandForecast:** Envía una cancelación de parte del proveedor al cliente por superar el plazo de 7 días para responder a un ciclo de negociación.

Percepciones:

FORECAST_REQUEST: se percibe como respuesta al operador *GenerarSolicitudPronosticoDemanda*.

AGREE_FORECAST_REQUEST_RESPONSE: se percibe como respuesta al operador **EnviarMensajeForecastRequest**.

POS_FORECAST: se percibe como respuesta al operador **GenerarPronosticoVentasPorPOS**.

PLANNED_EVENTS: se obtiene como percibe al operador **GenerarPlanEventosProgramado**.

DEMAND_FORECAST_RESPONSE_YES: se percibe como respuesta al operador **EvaluarPronosticoDemanda**.

DEMAND_FORECAST_RESPONSE_NO: se percibe como respuesta al operador **EvaluarPronosticoDemanda**.

UPDATED_DEMAND_FORECAST: se percibe como respuesta al operador **ActualizarPronostico**.

PROPOSE_DEMAND_FORECAST_1: se percibe como respuesta al operador **EnviarMensajeInformPlannedEvents o EnviarMensajeInformPOSForecast**.

PROPOSE_DEMAND_FORECAST_2 se percibe como respuesta al operador **EnviarMensajeProposeUpdatedDemandForecast**.

CANCELAR_DEMAND_FORECAST se percibe como respuesta al operador **CancelDemandForecast**.

INFORM_ISSUES se percibe como respuesta al operador **EnviarMensajeProposeUpdatedDemandForecast**. En el caso de que haya más de 5 ciclos de negociación.

REFUSE_FORECAST_REQUEST_RESPONSE se percibe como respuesta de operador **EnviarMensajeForecastRequest**.

SUCCESS se percibe como respuesta de operador **EnviarMensajeAcceptProposal o FinConExito**.

FAIL se percibe como respuesta de operador **FinConFalla**.

Diseño de la solución:

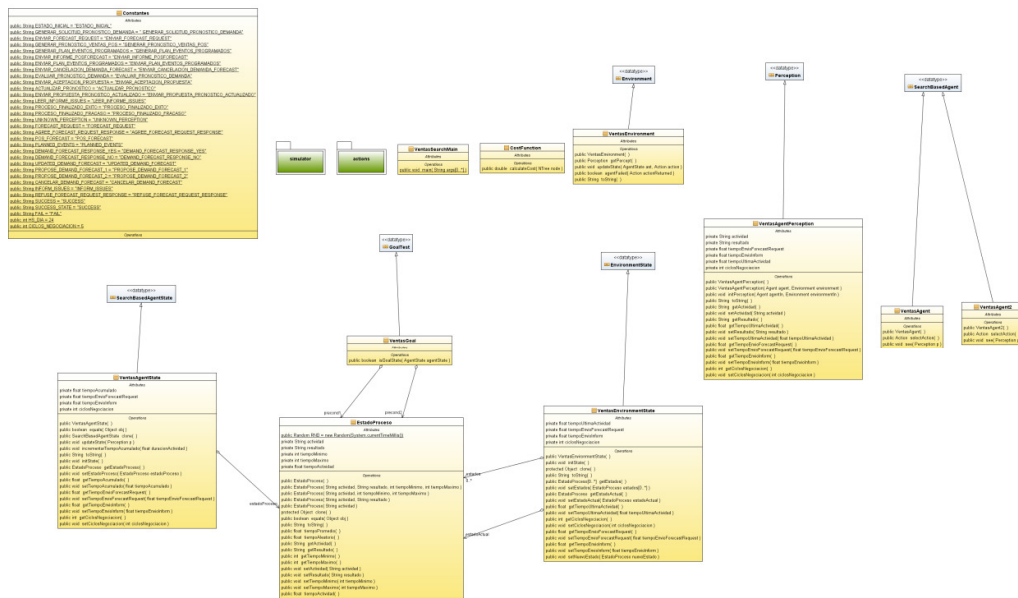


Figura 4. Diagrama de Clases de las extensiones de FAIA (Parte 1)

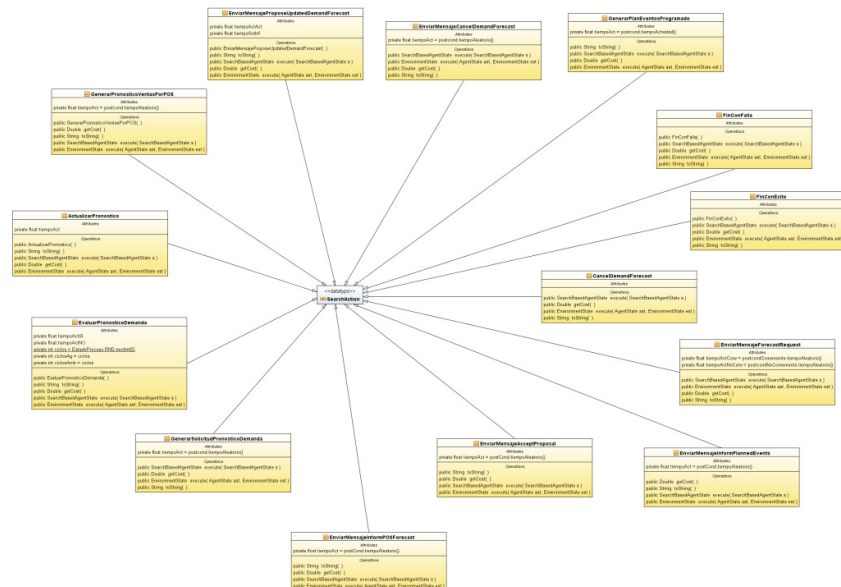


Figura 5. Diagrama de Clases de las extensiones de FAIA (Parte 2)

Implementación de la solución:

Antes de seleccionar una estrategia de búsqueda debemos definir hasta donde llega el dominio de nuestro problema. Ya que no poseemos más información que la proporcionada por la definición del mismo, sabemos que necesitaremos estrategias de búsqueda no informada. De todas ellas nos interesan las completas y óptimas. La

completitud es deseable para garantizar que se encuentre una solución cuando exista por lo menos una. La optimalidad garantizará que nuestro agente encuentre la solución de mejor calidad. Para nuestro escenario, la solución de mejor calidad será aquella que demande menos tiempo.

Búsqueda primero en anchura:

Esta es una estrategia sencilla en la que se expande primero el nodo raíz, a continuación se expanden todos los sucesores del nodo raíz, después sus sucesores, etc. En general se expanden todos los nodos a una profundidad en el árbol de búsqueda antes de expandir cualquier nodo del próximo nivel.

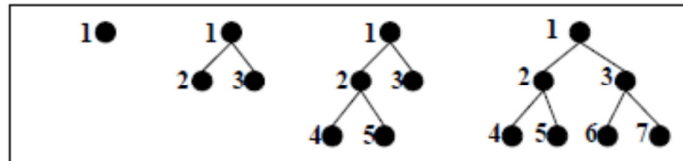


Figura 4. Ejemplo genérico búsqueda primero en anchura.

Se puede observar que la estrategia es completa si el nodo objetivo más superficial está a una cierta profundidad finita d (como en nuestro caso), pues lo encontrará después de expandir los nodos más superficiales. También se puede decir que es óptima si se considera que el costo del camino es una función no decreciente de la profundidad del nodo.

Considerando b : factor de ramificación.

d : profundidad de solución.

Podemos decir que el tiempo requerido para llegar al objetivo está dado por la siguiente ecuación:

$$1 + b + b^2 + b^3 + \dots + b^d$$

Y que se requiere almacenar todos los nodos en memoria, generando una cantidad de nodos dada por la siguiente ecuación:

$$b + b^2 + b^3 + \dots + b^d + (b^{d+1} - b) = O(b^{d+1}).$$

En base a estas propiedades podemos decir que el método de primero en anchura no es recomendable para problemas de gran magnitud, ya que requiere mucha memoria y tiempo para cumplir la meta.

Después de analizar detenidamente nuestro caso de estudio, concluimos que era más bien lineal y pequeño. Por lo que la generación y expansión de los nodos para un árbol de búsqueda no consumiría demasiado tiempo y memoria. Por lo antes mencionado, podemos decir que es un método de búsqueda apropiado para este caso de estudio.

Búsqueda de costo uniforme: La búsqueda de costo uniforme expande el nodo n con el camino de costo más pequeño. Notemos que si todos los costos son iguales, es idéntico a la búsqueda primero en anchura. Esta estrategia no se preocupa por el número de pasos que tiene un camino, pero sí sobre su costo total. Podemos garantizar completitud si el costo de cada paso es igual o mayor a alguna constante positiva pequeña. El costo de un camino siempre aumenta cuando vamos por él, es fácil de ver que el algoritmo expande nodos que incrementan el costo del camino. Por lo tanto, el primer nodo objetivo seleccionado para la expansión es la solución óptima.

Al no tener que expandir todos los nodos se mejoran considerablemente el tiempo de respuesta y la utilización de memoria en comparación con la búsqueda primero en anchura.

Esta estrategia es aplicable a nuestro caso de estudio ya que podemos asignar como costo de cada camino el tiempo que demanda la realización de cada actividad, considerando el tiempo acumulado del proceso como función de costo de cada nodo del árbol de búsqueda. Asegurándose que el costo de cualquier camino siempre aumentará de estado a estado.

RESULTADOS:

Los resultados obtenidos se basaron en un escenario en cual el agente llega al éxito en el primer ciclo de negociación para ambas búsquedas, pero con tiempos aleatorios distintos para las actividades y los mensajes, a continuación se muestran los resultados obtenidos:

Búsqueda de costo uniforme:

En esta corrida el agente logra el objetivo generando un árbol de búsqueda de ocho nodos llevándole un tiempo acumulado de 53 horas. A continuación se detallan los resultados obtenidos para de cada secuencia:

--- Search Based Simulator ---

Sending perception to agent...

Perception: Percepción: [actividad: ESTADO_INICIAL,resultado: null,
tiempoEnvioForecastRequest: 0.0, ciclosNegociacion: 0, tiempoEnvioInform: 0.0,
tiempoUltimaActividad: 0.0]

Agent State: VentasAgentState: [EstadoProceso: [Actividad: ESTADO_INICIALResultado: null],
tiempoAcumulado: 0.0, tiempoEnvioForecastRequest: 0.0, tiempoEnvioInform: 0.0,
ciclosNegociacion: 0]

Environment: VentasEnvironmentState: [EstadoProceso: [Actividad: ESTADO_INICIALResultado:
null],ciclosNegociacion: 0,tiempoUltimaActividad: 0.0,tiempoEnvioForecastRequest:
0.0,tiempoEnvioInform: 0.0,ciclosNegociacion: 0]

Asking the agent for an action...

Action returned: GenerarSolicitudPronosticoDemanda

Sending perception to agent...

Perception: Percepción: [actividad:
GENERAR_SOLICITUD_PRONOSTICO_DEMANDA,resultado: FORECAST_REQUEST,
tiempoEnvioForecastRequest: 0.0, ciclosNegociacion: 0, tiempoEnvioInform: 0.0,
tiempoUltimaActividad: 10.0]

Agent State: VentasAgentState: [EstadoProceso: [Actividad:
GENERAR_SOLICITUD_PRONOSTICO_DEMANDAResultado: FORECAST_REQUEST],
tiempoAcumulado: 10.0, tiempoEnvioForecastRequest: 0.0, tiempoEnvioInform: 0.0,
ciclosNegociacion: 0]

Environment: VentasEnvironmentState: [EstadoProceso: [Actividad:
GENERAR_SOLICITUD_PRONOSTICO_DEMANDAResultado:
FORECAST_REQUEST],ciclosNegociacion: 0,tiempoUltimaActividad:
10.0,tiempoEnvioForecastRequest: 0.0,tiempoEnvioInform: 0.0,ciclosNegociacion: 0]

Asking the agent for an action...

Action returned: EnviarMensajeForecastRequest

Sending perception to agent...

Perception: Percepción: [actividad: ENVIAR_FORECAST_REQUEST,resultado:
AGREE_FORECAST_REQUEST_RESPONSE, tiempoEnvioForecastRequest: 0.0,
ciclosNegociacion: 0, tiempoEnvioInform: 0.0, tiempoUltimaActividad: 3.0]

Agent State: VentasAgentState: [EstadoProceso: [Actividad:
ENVIAR_FORECAST_REQUESTResultado: AGREE_FORECAST_REQUEST_RESPONSE],
tiempoAcumulado: 13.0, tiempoEnvioForecastRequest: 0.0, tiempoEnvioInform: 0.0,
ciclosNegociacion: 0]

Environment: VentasEnvironmentState: [EstadoProceso: [Actividad:
ENVIAR_FORECAST_REQUESTResultado:
AGREE_FORECAST_REQUEST_RESPONSE],ciclosNegociacion: 0,tiempoUltimaActividad:
3.0,tiempoEnvioForecastRequest: 13.0,tiempoEnvioInform: 0.0,ciclosNegociacion: 0]
Asking the agent for an action...
Action returned: GenerarPronosticoVentasPorPOS

Sending perception to agent...
Perception: Percepción: [actividad: GENERAR_PRONOSTICO_VENTAS_POS,resultado:
POS_FORECAST, tiempoEnvioForecastRequest: 0.0, ciclosNegociacion: 0, tiempoEnvioInform:
0.0, tiempoUltimaActividad: 27.0]
Agent State: VentasAgentState: [EstadoProceso: [Actividad:
GENERAR_PRONOSTICO_VENTAS_POSResultado: POS_FORECAST], tiempoAcumulado:
40.0, tiempoEnvioForecastRequest: 0.0, tiempoEnvioInform: 0.0, ciclosNegociacion: 0]
Environment: VentasEnvironmentState: [EstadoProceso: [Actividad:
GENERAR_PRONOSTICO_VENTAS_POSResultado: POS_FORECAST],ciclosNegociacion:
0,tiempoUltimaActividad: 27.0,tiempoEnvioForecastRequest: 13.0,tiempoEnvioInform:
0.0,ciclosNegociacion: 0]
Asking the agent for an action...
Action returned: EnviarMensajeInformPOSForecast

Sending perception to agent...
Perception: Percepción: [actividad: ENVIAR_INFORME_POSFORECAST,resultado:
PROPOSE_DEMAND_FORECAST_1, tiempoEnvioForecastRequest: 0.0, ciclosNegociacion: 0,
tiempoEnvioInform: 0.0, tiempoUltimaActividad: 1.0]
Agent State: VentasAgentState: [EstadoProceso: [Actividad:
ENVIAR_INFORME_POSFORECASTResultado: PROPOSE_DEMAND_FORECAST_1],
tiempoAcumulado: 41.0, tiempoEnvioForecastRequest: 0.0, tiempoEnvioInform: 0.0,
ciclosNegociacion: 0]
Environment: VentasEnvironmentState: [EstadoProceso: [Actividad:
ENVIAR_INFORME_POSFORECASTResultado:
PROPOSE_DEMAND_FORECAST_1],ciclosNegociacion: 0,tiempoUltimaActividad:
1.0,tiempoEnvioForecastRequest: 13.0,tiempoEnvioInform: 0.0,ciclosNegociacion: 0]
Asking the agent for an action...
Action returned: EvaluarPronosticoDemanda

Sending perception to agent...
Perception: Percepción: [actividad: EVALUAR_PRONOSTICO_DEMANDA,resultado:
DEMAND_FORECAST_RESPONSE_YES, tiempoEnvioForecastRequest: 0.0, ciclosNegociacion:
1, tiempoEnvioInform: 0.0, tiempoUltimaActividad: 10.0]
Agent State: VentasAgentState: [EstadoProceso: [Actividad:
EVALUAR_PRONOSTICO_DEMANDAResultado: DEMAND_FORECAST_RESPONSE_YES],
tiempoAcumulado: 51.0, tiempoEnvioForecastRequest: 0.0, tiempoEnvioInform: 0.0,
ciclosNegociacion: 1]
Environment: VentasEnvironmentState: [EstadoProceso: [Actividad:
EVALUAR_PRONOSTICO_DEMANDAResultado:
DEMAND_FORECAST_RESPONSE_YES],ciclosNegociacion: 1,tiempoUltimaActividad:
10.0,tiempoEnvioForecastRequest: 0.0,tiempoEnvioInform: 0.0,ciclosNegociacion: 1]
Asking the agent for an action...
Action returned: EnviarMensajeAcceptProposal

Sending perception to agent...

Perception: Percepción: [actividad: ENVIAR_ACEPTACION_PROPUESTA,resultado: SUCCESS, tiempoEnvioForecastRequest: 0.0, ciclosNegociacion: 1, tiempoEnvioInform: 0.0, tiempoUltimaActividad: 2.0]
Agent State: VentasAgentState: [EstadoProceso: [Actividad: ENVIAR_ACEPTACION_PROPUESTAResultado: SUCCESS], tiempoAcumulado: 53.0, tiempoEnvioForecastRequest: 0.0, tiempoEnvioInform: 0.0, ciclosNegociacion: 1]
Environment: VentasEnvironmentState: [EstadoProceso: [Actividad: ENVIAR_ACEPTACION_PROPUESTAResultado: SUCCESS],ciclosNegociacion: 1,tiempoUltimaActividad: 2.0,tiempoEnvioForecastRequest: 0.0,tiempoEnvioInform: 0.0,ciclosNegociacion: 1]
Asking the agent for an action..
Action returned: FinConExito

Sending perception to agent...
Perception: Percepción: [actividad: PROCESO_FINALIZADO_EXITO,resultado: SUCCESS, tiempoEnvioForecastRequest: 0.0, ciclosNegociacion: 1, tiempoEnvioInform: 0.0, tiempoUltimaActividad: 0.0]
Agent State: VentasAgentState: [EstadoProceso: [Actividad: PROCESO_FINALIZADO_EXITOResultado: SUCCESS], tiempoAcumulado: 53.0, tiempoEnvioForecastRequest: 0.0, tiempoEnvioInform: 0.0, ciclosNegociacion: 1]
Environment: VentasEnvironmentState: [EstadoProceso: [Actividad: PROCESO_FINALIZADO_EXITOResultado: SUCCESS],ciclosNegociacion: 1,tiempoUltimaActividad: 0.0,tiempoEnvioForecastRequest: 0.0,tiempoEnvioInform: 0.0,ciclosNegociacion: 1]
Asking the agent for an action..
Agent has reached the goal!

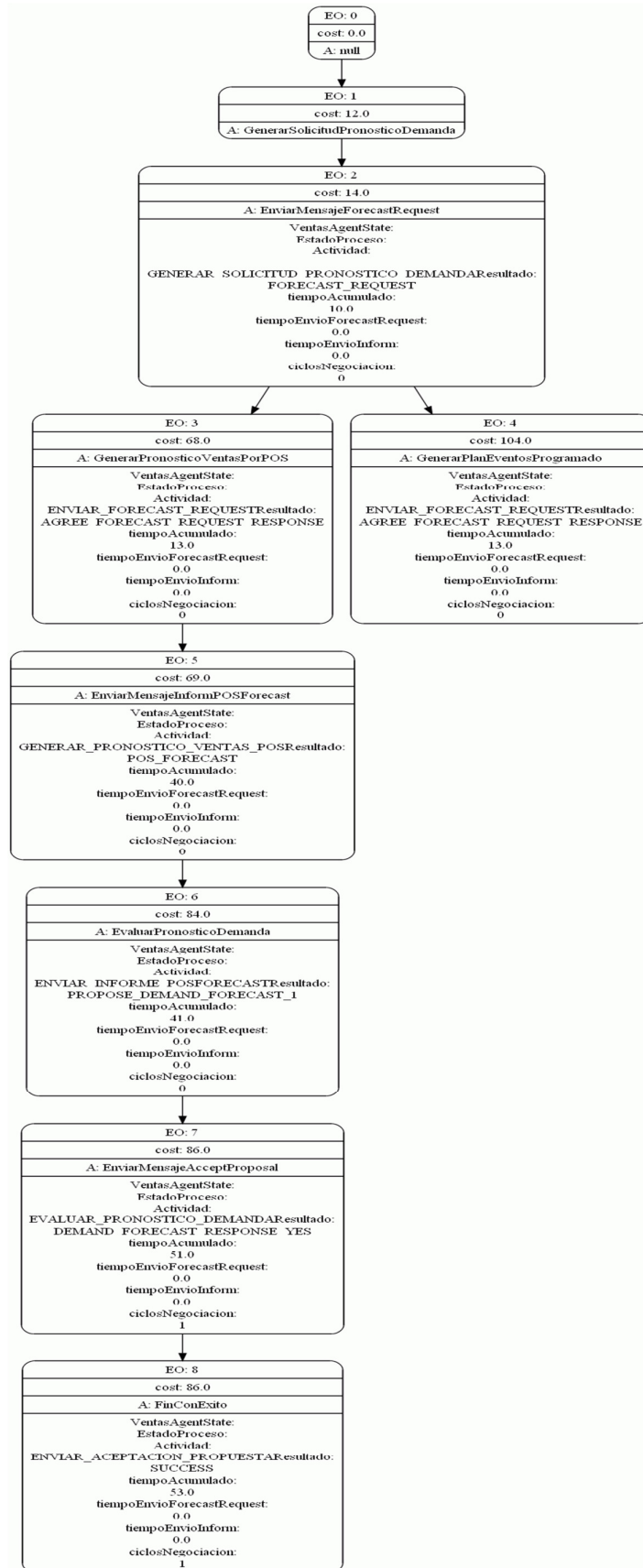


Figura 5. Árbol de búsqueda elaborado por el agente

Búsqueda Primero en Anchura:

En esta corrida el agente logra el objetivo generando un árbol de búsqueda de doce nodos llevándole un tiempo acumulado de 63 horas. A continuación se detallan los resultados obtenidos para de cada secuencia y el árbol de búsqueda generado por el agente:

--- Search Based Simulator ---

Sending perception to agent...

Perception: Percepción: [actividad: ESTADO_INICIAL,resultado: null,
tiempoEnvioForecastRequest: 0.0, ciclosNegociacion: 0, tiempoEnvioInform: 0.0,
tiempoUltimaActividad: 0.0]

Agent State: VentasAgentState: [EstadoProceso: [Actividad: ESTADO_INICIALResultado: null],
tiempoAcumulado: 0.0, tiempoEnvioForecastRequest: 0.0, tiempoEnvioInform: 0.0,
ciclosNegociacion: 0]

Environment: VentasEnvironmentState: [EstadoProceso: [Actividad: ESTADO_INICIALResultado:
null],ciclosNegociacion: 0,tiempoUltimaActividad: 0.0,tiempoEnvioForecastRequest:
0.0,tiempoEnvioInform: 0.0,ciclosNegociacion: 0]

Asking the agent for an action...

Action returned: GenerarSolicitudPronosticoDemanda

Sending perception to agent...

Perception: Percepción: [actividad:
GENERAR_SOLICITUD_PRONOSTICO_DEMANDA,resultado: FORECAST_REQUEST,
tiempoEnvioForecastRequest: 0.0, ciclosNegociacion: 0, tiempoEnvioInform: 0.0,
tiempoUltimaActividad: 12.0]

Agent State: VentasAgentState: [EstadoProceso: [Actividad:
GENERAR_SOLICITUD_PRONOSTICO_DEMANDAResultado: FORECAST_REQUEST],
tiempoAcumulado: 12.0, tiempoEnvioForecastRequest: 0.0, tiempoEnvioInform: 0.0,
ciclosNegociacion: 0]

Environment: VentasEnvironmentState: [EstadoProceso: [Actividad:
GENERAR_SOLICITUD_PRONOSTICO_DEMANDAResultado:
FORECAST_REQUEST],ciclosNegociacion: 0,tiempoUltimaActividad:
12.0,tiempoEnvioForecastRequest: 0.0,tiempoEnvioInform: 0.0,ciclosNegociacion: 0]

Asking the agent for an action...

Action returned: EnviarMensajeForecastRequest

Sending perception to agent...

Perception: Percepción: [actividad: ENVIAR_FORECAST_REQUEST,resultado:
AGREE_FORECAST_REQUEST_RESPONSE, tiempoEnvioForecastRequest: 0.0,
ciclosNegociacion: 0, tiempoEnvioInform: 0.0, tiempoUltimaActividad: 3.0]

Agent State: VentasAgentState: [EstadoProceso: [Actividad:
ENVIAR_FORECAST_REQUESTResultado: AGREE_FORECAST_REQUEST_RESPONSE],
tiempoAcumulado: 15.0, tiempoEnvioForecastRequest: 0.0, tiempoEnvioInform: 0.0,
ciclosNegociacion: 0]

Environment: VentasEnvironmentState: [EstadoProceso: [Actividad:
ENVIAR_FORECAST_REQUESTResultado:
AGREE_FORECAST_REQUEST_RESPONSE],ciclosNegociacion: 0,tiempoUltimaActividad:
3.0,tiempoEnvioForecastRequest: 15.0,tiempoEnvioInform: 0.0,ciclosNegociacion: 0]

Asking the agent for an action...
Action returned: GenerarPronosticoVentasPorPOS

Sending perception to agent...
Perception: Percepción: [actividad: GENERAR_PRONOSTICO_VENTAS_POS,resultado:
POS_FORECAST, tiempoEnvioForecastRequest: 0.0, ciclosNegociacion: 0, tiempoEnvioInform:
0.0, tiempoUltimaActividad: 28.0]
Agent State: VentasAgentState: [EstadoProceso: [Actividad:
GENERAR_PRONOSTICO_VENTAS_POSResultado: POS_FORECAST], tiempoAcumulado:
43.0, tiempoEnvioForecastRequest: 0.0, tiempoEnvioInform: 0.0, ciclosNegociacion: 0]
Environment: VentasEnvironmentState: [EstadoProceso: [Actividad:
GENERAR_PRONOSTICO_VENTAS_POSResultado: POS_FORECAST],ciclosNegociacion:
0,tiempoUltimaActividad: 28.0,tiempoEnvioForecastRequest: 15.0,tiempoEnvioInform:
0.0,ciclosNegociacion: 0]
Asking the agent for an action...
Action returned: EnviarMensajeInformPOSForecast

Sending perception to agent...
Perception: Percepción: [actividad: ENVIAR_INFORME_POSFORECAST,resultado:
PROPOSE_DEMAND_FORECAST_1, tiempoEnvioForecastRequest: 0.0, ciclosNegociacion: 0,
tiempoEnvioInform: 0.0, tiempoUltimaActividad: 2.0]
Agent State: VentasAgentState: [EstadoProceso: [Actividad:
ENVIAR_INFORME_POSFORECASTResultado: PROPOSE_DEMAND_FORECAST_1],
tiempoAcumulado: 45.0, tiempoEnvioForecastRequest: 0.0, tiempoEnvioInform: 0.0,
ciclosNegociacion: 0]
Environment: VentasEnvironmentState: [EstadoProceso: [Actividad:
ENVIAR_INFORME_POSFORECASTResultado:
PROPOSE_DEMAND_FORECAST_1],ciclosNegociacion: 0,tiempoUltimaActividad:
2.0,tiempoEnvioForecastRequest: 15.0,tiempoEnvioInform: 0.0,ciclosNegociacion: 0]
Asking the agent for an action...
Action returned: EvaluarPronosticoDemanda

Sending perception to agent...
Perception: Percepción: [actividad: EVALUAR_PRONOSTICO_DEMANDA,resultado:
DEMAND_FORECAST_RESPONSE_YES, tiempoEnvioForecastRequest: 0.0, ciclosNegociacion:
1, tiempoEnvioInform: 0.0, tiempoUltimaActividad: 15.0]
Agent State: VentasAgentState: [EstadoProceso: [Actividad:
EVALUAR_PRONOSTICO_DEMANDAResultado: DEMAND_FORECAST_RESPONSE_YES],
tiempoAcumulado: 60.0, tiempoEnvioForecastRequest: 0.0, tiempoEnvioInform: 0.0,
ciclosNegociacion: 1]
Environment: VentasEnvironmentState: [EstadoProceso: [Actividad:
EVALUAR_PRONOSTICO_DEMANDAResultado:
DEMAND_FORECAST_RESPONSE_YES],ciclosNegociacion: 1,tiempoUltimaActividad:
15.0,tiempoEnvioForecastRequest: 0.0,tiempoEnvioInform: 0.0,ciclosNegociacion: 1]
Asking the agent for an action...
Action returned: EnviarMensajeAcceptProposal

Sending perception to agent...
Perception: Percepción: [actividad: ENVIAR_ACEPTACION_PROPUUESTA,resultado: SUCCESS,
tiempoEnvioForecastRequest: 0.0, ciclosNegociacion: 1, tiempoEnvioInform: 0.0,
tiempoUltimaActividad: 3.0]

Agent State: VentasAgentState: [EstadoProceso: [Actividad:
ENVIAR_ACEPTACION_PROPUESEResultado: SUCCESS], tiempoAcumulado: 63.0,
tiempoEnvioForecastRequest: 0.0, tiempoEnvioInform: 0.0, ciclosNegociacion: 1]
Environment: VentasEnvironmentState: [EstadoProceso: [Actividad:
ENVIAR_ACEPTACION_PROPUESEResultado: SUCCESS],ciclosNegociacion:
1,tiempoUltimaActividad: 3.0,tiempoEnvioForecastRequest: 0.0,tiempoEnvioInform:
0.0,ciclosNegociacion: 1]
Asking the agent for an action...
Action returned: FinConExito

Sending perception to agent...
Perception: Percepción: [actividad: PROCESO_FINALIZADO_EXITO,resultado: SUCCESS,
tiempoEnvioForecastRequest: 0.0, ciclosNegociacion: 1, tiempoEnvioInform: 0.0,
tiempoUltimaActividad: 0.0]
Agent State: VentasAgentState: [EstadoProceso: [Actividad:
PROCESO_FINALIZADO_EXITOResultado: SUCCESS], tiempoAcumulado: 63.0,
tiempoEnvioForecastRequest: 0.0, tiempoEnvioInform: 0.0, ciclosNegociacion: 1]
Environment: VentasEnvironmentState: [EstadoProceso: [Actividad:
PROCESO_FINALIZADO_EXITOResultado: SUCCESS],ciclosNegociacion:
1,tiempoUltimaActividad: 0.0,tiempoEnvioForecastRequest: 0.0,tiempoEnvioInform:
0.0,ciclosNegociacion: 1]
Asking the agent for an action...
Agent has reached the goal!

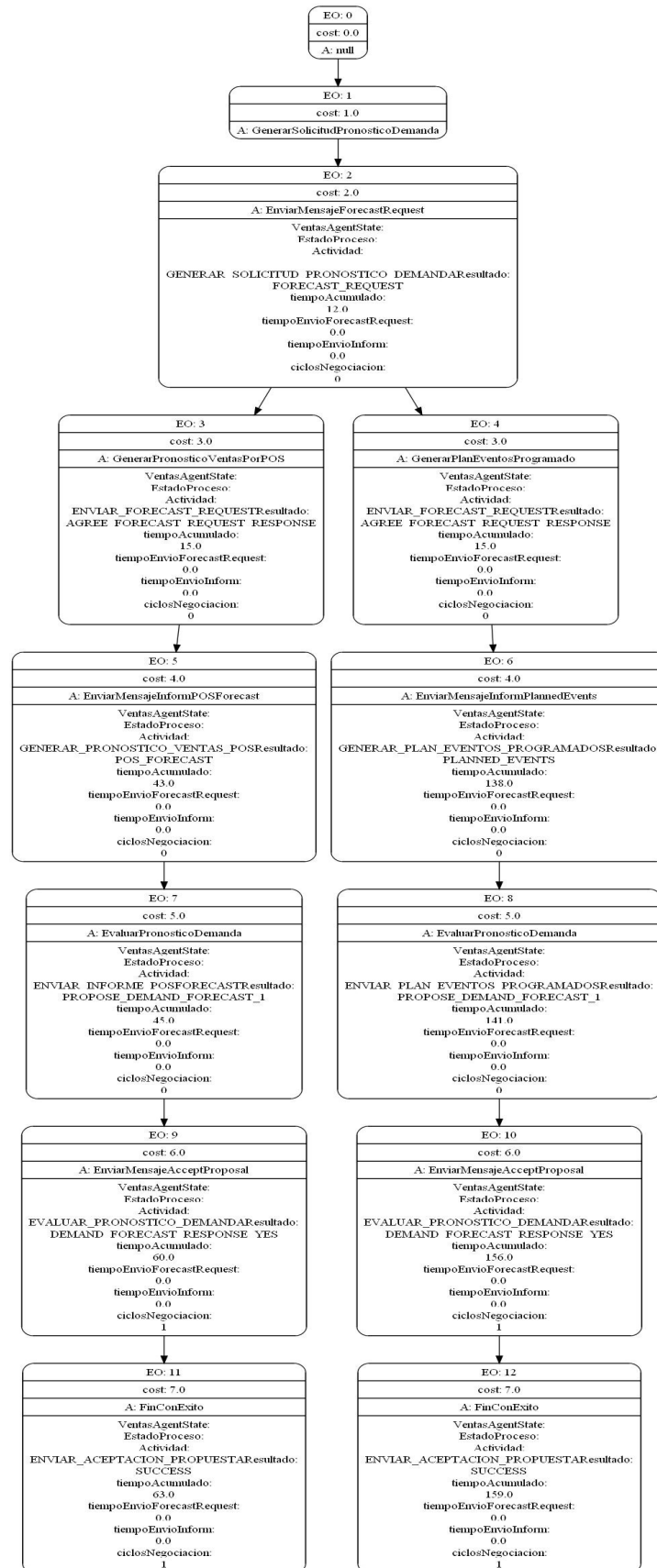


Figura 6. Árbol generado por el agente con una búsqueda primero en anchura.

En ambos casos se ha obtenido el resultado deseado ya que el agente logra cumplir con la prueba de meta. Sin embargo hay diferencias significativas entre las dos estrategias de búsquedas. Como se puede apreciar tal cual se explica en la teoría, la búsqueda de Costo uniforme llega al estado objetivo en menor tiempo que la de Primero en anchura. El consumo de memoria y tiempo en la primera simulación es menor ya que genera ocho nodos en el árbol de búsqueda contra doce de la segunda simulación.

Se concluye que para nuestro caso de estudio es más eficiente la estrategia de Costo uniforme corroborando lo planteado al principio.

CONCLUSIÓN:

En base al trabajo realizado se puede concluir que las técnicas de búsquedas seleccionadas han logrado solucionar el problema planteado ya que el agente logra el resultado esperado en el menor tiempo posible. Sin embargo, se puede observar que en este caso, en el cual tiene sentido asignar un costo diferente a cada acción, relacionado con el tiempo en que se efectúa, y considerar el tiempo acumulado del proceso como algo significativo la búsqueda de Costo Uniforme es más adecuada para la resolución de este proceso B2B ya que consume menos espacio en memoria, demanda menos tiempo para llegar al objetivo, y dada las características del problema es completa y óptima.

REFERENCIAS:

1 - Russell, Norvig: Inteligencia artificial: un enfoque moderno 2da. Edición, PEARSON EDUCACIÓN. S.A., Madrid. 2004;