

UTILIZACION DE LA COMPUTACION EVOLUTIVA Y LA
TEORIA DE JUEGOS PARA LA AUTOMATIZACION EN
SISTEMAS DE SOPORTE A LA NEGOCIACION

HENRY ALBERTO GOMEZ ROCHA
JHON JAIRO ACOSTA PAREJO

UNIVERSIDAD DEL MAGDALENA
FACULTAD DE INGENIERIA
PROGRAMA DE INGENIERIA DE SISTEMAS
SANTA MARTA D.T.C.H.

2006

UTILIZACION DE LA COMPUTACION EVOLUTIVA Y LA
TEORIA DE JUEGOS PARA LA AUTOMATIZACION EN
SISTEMAS DE SOPORTE A LA NEGOCIACION

HENRY ALBERTO GOMEZ ROCHA
JHON JAIRO ACOSTA PAREJO

Memoria de Grado presentado como requisito parcial
para optar al titulo de Ingeniero de Sistemas

Director:

Ph.D (c). OSWALDO VELEZ-LANGS

UNIVERSIDAD DEL MAGDALENA
FACULTAD DE INGENIERIA
PROGRAMA DE INGENIERIA DE SISTEMAS
SANTA MARTA D.T.C.H.

2006

Nota de aceptación:

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Santa Marta D.T.C.H. Octubre de 2006

DEDICADO A:

Los seres más especiales que he tenido, tengo y tendré: mis padres.
Para ellos todo mi respeto y amor.

HENRY

Dios el verdadero creador. A la familia Parejo Castillo en especial a mi madre Ena Alicia Parejo Castillo, mis abuelos Ena Alicia Castillo Acosta, Alejandro Parejo Moreno por ese apoyo incondicional.
A mi hija *Anghely Paola Acosta Durán* y a *Rita Inés Durán de la Hoz* con quienes
deje de compartir tiempo para lograr esta meta.
Sin dejar de lado a mis amigos cercanos quienes siempre estuvieron pendientes de los avances de este trabajo, brindando apoyo, facilitando tareas, razón, motivo y eje (*Jose Diaz, Lenin Martinez, Joham Martinez, Alvaro Campo* y mi compañero de tesis *Henry Gómez*).

JHON

AGRADECIMIENTOS

OSWALDO VELEZ-LANGS, candidato a Doctor en Lenguajes, Sistemas Informáticos e Ingeniería del Software, Profesor Ayudante, Departamento de Arquitectura, Tecnología de Computadores, Ciencia de la Computación e Inteligencia Artificial, Escuela Superior de Ciencias Experimentales y Tecnología, Universidad Rey Juan Carlos; quien manifestó desde el comienzo del proyecto plena confianza en el desarrollo del trabajo, quién nos ha aportado sugerencias oportunas y quién brinda una guía para publicar artículos en eventos de carácter internacional.

RAFAEL VIANA BARCELÓ, Magíster en Ciencias Económicas, Profesor de la Universidad del Magdalena, quién nos facilitó abundante material referente a Teoría de Juegos, quién nos colaboró en la creación de los escenarios con los cuales probamos el sistema desarrollado y quién estableció el contacto con Julián Arévalo Bencardino.

JULIAN ARÉVALO BENCARDINO, candidato a Doctor del Departamento de Economía de la Universidad de Boston, de quién hemos tenido un apoyo incondicional y del cual obtuvimos oportunas sugerencias en el campo de la Teoría de Juegos y Negociación.

A los Doctores **KUO-MING CHAO**, **J-H CHEN** y **SHIH-HUNG WU**, quienes nos han suministrado artículos sobre las temáticas del proyecto y han respondido oportunamente preguntas acerca del trabajo.

RESUMEN

La teoría de juegos nació con el objetivo de analizar el comportamiento de las personas en situaciones con conflictos de intereses, es decir, nos ayuda a crear modelos sobre los cuales razonamos para elegir las estrategias que nos conduzcan a resultados óptimos, donde todos los participantes del proceso ganen. Así mismo, los agentes inteligentes haciendo uso de los algoritmos genéticos son capaces de llegar a acuerdos en procesos de negociación guiados por conceptos cuyo origen es la teoría de juegos. Este trabajo procura utilizar la teoría de juegos y los agentes inteligentes como métodos de análisis de decisión y de apoyo a la decisión de grupos, desplegados sobre la web, con miras a involucrar al público en las decisiones que les afectan, es así como vemos en Internet una oportunidad de acortar la brecha entre regidores y regidos. Podemos resumir el objeto principal que persigue este trabajo como proponer e implementar un esquema de negociación automatizada sobre la Web, el cual esté basado en la computación evolutiva y la teoría de juegos.

Palabras clave: Sistemas MultiAgentes, Agentes Inteligentes, Algoritmos Genéticos, Equilibrio de Nash, Eficiencia de Pareto, TTP (Tercera Parte Confiable), NFD (Algoritmo de Equilibrio sin temor de desviación), INGENIAS, JADE.

ÍNDICE GENERAL

	IV
AGRADECIMIENTOS	v
RESUMEN	vi
INTRODUCCIÓN	1
1. AGENTES INTELIGENTES	4
1.1. Agentes Inteligentes	4
1.1.1. Agentes	5
1.1.2. Agentes inteligentes	6
1.1.3. Ambiente	8
1.1.3.1. Accesibles vs. No Accesibles	9
1.1.3.2. Deterministas vs. No Deterministas	9
1.1.3.3. Episódicos vs. No Episódicos	9
1.1.3.4. Estáticos vs. Dinámicos	10
1.1.3.5. Discretos vs. Continuos	10
1.1.4. Arquitecturas para agentes inteligentes	10
1.1.4.1. Arquitecturas deliberativas	10
1.1.4.2. Arquitecturas reactivas	11
1.1.4.3. Arquitecturas híbridas	11
1.1.5. Tipología de agentes	11
1.1.5.1. Clasificación de los agentes inteligentes	12
1.2. Sistemas Multiagentes	13
1.2.1. Características de un Sistema MultiAgente	14
1.2.1.1. Diseño del Agente	14
1.2.1.2. Entorno	15
1.2.1.3. Percepción	15
1.2.1.4. Control	15

1.2.1.5.	Conocimiento	15
1.2.1.6.	Comunicación	16
2.	TEORÍA DE JUEGOS: CONCEPTOS Y EJEMPLOS	17
2.1.	Elementos de la Teoría de Juegos	18
2.1.1.	Jugadores	18
2.1.2.	Acción	18
2.1.3.	Información	18
2.1.4.	Estrategia	19
2.1.5.	Retribución	19
2.1.6.	Resultado	19
2.1.7.	Equilibrio	19
2.2.	Forma Normal	19
2.3.	Forma Extensiva	20
2.4.	Estrategias Dominantes	20
2.5.	Juegos No Cooperativos	20
2.6.	Juegos Cooperativos	21
2.7.	El Dilema del Prisionero	21
2.8.	Equilibrio de Nash	22
2.9.	Eficiencia de Pareto	23
3.	COMPUTACIÓN EVOLUTIVA	24
3.1.	Algoritmos Genéticos	25
3.1.1.	Estructura de un Algoritmo Genético	26
3.1.2.	Terminología Básica sobre Algoritmos Genéticos	27
3.1.2.1.	Cromosoma	28
3.1.2.2.	Gen	28
3.1.2.3.	Alelo	28
3.1.2.4.	Esquema	28
3.1.2.5.	Espacio de Búsqueda	28
3.1.2.6.	Aptitud (Fitness)	29
3.1.2.7.	Bloque de Construcción	29
3.1.2.8.	Decepción	29
3.1.2.9.	Elitismo	29
3.1.2.10.	Explotación	29
3.1.2.11.	Exploración	30
3.1.2.12.	Genotipo	30
3.1.2.13.	Fenotipo	30

3.1.2.14. Nicho	30
3.1.3. Operadores de un Algoritmo Genético	30
3.1.3.1. Selección	30
3.1.3.2. Cruce	31
3.1.3.3. Mutación	31
3.1.4. Mecanismos de Selección de los Algoritmos Genéticos	34
3.1.4.1. Selección por Ruleta	34
3.1.4.2. Asignación de Aptitud con base en el Ranking	34
3.1.4.3. Selección por Torneo	35
3.1.5. Criterios Necesarios para Implantar un AG	35
3.1.5.1. Criterio de Codificación	35
3.1.5.2. Criterio de Tratamiento de Individuos no Factibles	35
3.1.5.3. Criterio de Inicialización	35
3.1.5.4. Criterio de Parada	35
3.1.5.5. Funciones de Evaluación y Aptitud	36
3.1.5.6. Operadores Genéticos	36
3.1.5.7. Criterios de Selección	36
3.1.5.8. Criterios de Reemplazo	36
3.1.5.9. Parámetros de Funcionamiento	36
4. DEMOCRACIA ELECTRÓNICA	37
4.1. Opciones Tecnológicas para la Democracia Electrónica	38
4.1.1. Herramientas Tecnológicas	39
4.1.2. Beneficios Potenciales	40
4.2. Presupuestos Participativos	41
5. NEGOCIACIÓN AUTOMATIZADA	43
5.1. Negociación	43
5.2. Negociación Automatizada	44
5.3. Componentes de la Negociación Automatizada	44
5.3.1. Protocolos de Negociación	45
5.3.2. Objetos de Negociación	45
5.3.3. Modelo de Toma de Decisiones de los Agentes	45
5.4. Propiedades de un Mecanismo de Negociación	46
5.4.1. Eficiencia	46
5.4.2. Efectividad	46
5.4.3. Función de Utilidad Privada	46
5.4.4. Múltiples Asuntos	46

5.4.5. Simetría	46
5.5. Reseña de la Evolución de la Negociación Automatizada	47
6. DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO	51
6.1. Arquitectura de la Aplicación	51
6.1.1. Ambiente del prototipo de agentes	53
6.2. Características del Sistema MultiAgente	54
6.2.1. Control	54
6.2.2. Conocimiento	54
6.2.3. Comunicación	54
6.3. Explicación Lógica del Prototipo	55
6.3.1. Juego Mediado por una Tercera Parte Confiable (TTP, Trusted Third Party)	60
6.3.2. Mecanismo de Negociación	61
6.3.3. La Tercera Parte Confiable	63
6.3.4. Acción de Comunicación	63
6.3.5. Acción de Comunicación de Garantía	63
6.3.6. Acción de Comunicación de Compensación	63
6.3.7. Protocolo General de Negociación	64
6.3.8. Características del Sistema de Negociación Automatizada	66
6.4. Agentes del Sistema MultiAgente	67
6.4.1. Agente Negociador	67
6.4.2. Agente TTP	69
6.5. Modelos del Sistema MultiAgente	69
6.5.1. Modelos Interacción	70
6.5.2. Modelos de Entorno	72
6.5.3. Modelos de Tareas y Objetivos	75
6.5.4. Casos de Uso	81
7. PRUEBAS DEL SISTEMA	89
7.1. Escenario 1	89
7.2. Escenario 2	92
8. APORTES Y CONTRIBUCIONES	97
8.1. Divulgación de Resultados	97
9. CONCLUSIONES	99
9.1. Trabajo Futuro	99

Bibliografía	101
A. METAMODELOS TRATADOS CON INGENIAS	105
A.1. Metamodelo de Agente	105
A.1.1. Presentación del Modelo	106
A.1.1.1. Control del Agente	106
A.1.1.2. Agentes en Ejecución	107
A.1.1.3. Asociación de Responsabilidades	107
A.1.1.4. Estado Mental	107
A.2. Metamodelos de Objetivos y Tareas	108
A.2.1. Presentación del Metamodelo	108
A.3. Metamodelo de Interacción	110
B. JADE	111
B.1. Clase Agent	113
B.2. Intercomunicación entre Agentes	113
B.3. Tareas de los Agentes (Behaviours)	113

ÍNDICE DE TABLAS

2.1. Matriz de Pagos en el Dilema del Prisionero.	22
6.1. Descripción del Modelo de Agente Negociador.	67
6.2. Descripción del Modelo de Agente TTP.	69
6.3. Descripción del Modelo de Interacción Encontrar Equilibrio.	70
6.4. Descripción del Modelo de Interacción Realimentación Propuestas.	72
6.5. Descripción del Modelo de Entorno Agente TTP.	73
6.6. Descripción del Modelo de Entorno Agente Negociador.	74
6.7. Descripción del Modelo de Tarea y Objetivos Obtener Parámetros de Usuarios.	76
6.8. Descripción del Modelo de Tarea y Objetivos Negociar Valores de Asuntos.	77
6.9. Descripción del Modelo de Tarea y Objetivos Notificar Propuestas.	77
6.10. Descripción del Modelo de Tarea y Objetivos Comprobar Cambios.	78
6.11. Descripción del Modelo de Descomposición de Objetivos.	79
6.12. Descripción del Modelo de Casos de Uso.	83
7.1. Descripción Variables de Negociación Escenario 1.	90
7.2. Configuración del Sistema Escenario 1.	90
7.3. Resultados Primera Etapa.	91
7.4. Resultados Última Etapa.	91
7.5. Matriz después de la Negociación.	92
7.6. Descripción Variables de Negociación Escenario 2.	94
7.7. Configuración del Sistema Escenario 2.	94
7.8. Resultados Primera Etapa.	95
7.9. Resultados Etapa Tres.	95
7.10. Matriz después de la Negociación.	96

ÍNDICE DE FIGURAS

1.1. Abstracción de un agente a partir de su interacción con el medio ambiente o entorno.	5
1.2. Impacto actual de los Agentes Inteligentes. Adaptado de [44].	8
1.3. Una clasificación de los agentes inteligentes. Adaptado de [24].	12
1.4. Robots que juegan fútbol.	13
3.1. Ciclo básico de un AG.	27
3.2. Tipos de operadores de cruce.	32
3.3. Analogía entre la evolución natural y un algoritmo genético.	33
3.4. Selección por Ruleta.	34
6.1. Arquitectura del prototipo desarrollado.	52
6.2. Mecanismo de Negociación Automatizada. Tomado y adaptado de [33].	62
6.3. Modelo de Agente Negociador. Representa el estado mental, objetivos y roles.	67
6.4. Modelo de Agente TTP. Representa los objetivos y roles.	69
6.5. Modelo de Interacción Encontrar Equilibrio.	70
6.6. Modelo de Interacción Realimentación Propuestas.	71
6.7. Modelo de Entorno Agente TTP.	73
6.8. Modelo de Entorno Agente Negociador.	74
6.9. Modelo de Tareas y Objetivos Obtener Parámetros de Usuarios.	76
6.10. Modelo de Tareas y Objetivos Negociar Valores de Asuntos.	76
6.11. Modelo de Tareas y Objetivos Notificar Propuestas.	77
6.12. Modelo de Tareas y Objetivos Comprobar Cambios.	77
6.13. Modelo de Tareas y Objetivos Descomposición de Objetivos.	78
6.14. Modelo de Casos de Uso	82
7.1. Combinaciones Estratégicas del Escenario 1.	92
7.2. Combinaciones Estratégicas del Escenario 2.	96
B.1. Modelo de Plataforma de Agentes usado por JADE.	112

INTRODUCCIÓN

La gran enseñanza de la historia es que los conflictos deben solucionarse mediante el dialogo y la deliberación. Los países desarrollados apuntan a hacer participes cada vez más a los ciudadanos en la toma de decisiones que les afectan y las grandes industrias por su parte tratan de establecer asociaciones con los distribuidores de los bienes que ellas fabrican.

Este cambio de enfoque en la solución de conflictos, ha originado una creciente investigación orientada hacia la creación de un sistema que soporte un proceso de negociación automatizado. Es así como el modelamiento y soporte al problema de la negociación propone un reto muy interesante a los campos de Ciencia de la Decisión, Investigación de Operaciones e Inteligencia Artificial.

La propuesta presentada aquí se enmarca dentro de la denominada: *Democracia Electrónica*, en la cual se pretende implantar un proceso democrático participativo que persigue un acuerdo entre grupos de personas con conflicto de intereses en asuntos concernientes a la sociedad, para lo cual se hace necesario plantear modelos de negociación bajo los cuales se interactúe para encontrar la solución mas óptima, el soporte que se da a dicho proceso esta habilitado por Internet.

El problema surge debido a que dichos modelos por lo general se basan en negociaciones multi-criterio, es decir, con múltiples variables y por tal razón se recae en un proceso complejo, generando un espacio de búsqueda de la solución mas óptima extenso, demandando mucho tiempo e incluso convirtiéndose en un proceso demasiado tedioso. Otro problema que conllevan estas dinámicas es la presencia obligada de los participantes al momento de negociar, promoviendo mayor inversión de tiempo y dinero en el proceso en general. Nuestra propuesta además de abarcar los problemas antes mencionados intenta también fomentar una comunicación más efectiva entre los miembros participantes, explorar los aspectos involucrados en el proceso de negociación de una manera creativa, efectiva y finalmente construir un entendimiento compartido.

Teniendo en cuenta que uno de los puntos centrales para facilitar el trabajo de equipos móviles es la creación y establecimiento de equipos desde agentes autónomos. Nuestro propósito es aplicar una nueva metodología para proveer mecanismos que faciliten la creación de equipos y ayudar a resolver conflictos a través de negociación automatizada. Los mecanismos de negociación serán implementados mediante una combinación de la teoría de juegos y la metodología co-evolutiva.

Para la implementación del sistema propuesto, se utilizó la metodología INGENIAS [7] para el análisis y desarrollo, se aplicaron los meta-modelos para describir los agentes inteligentes necesarios, sus acciones o tareas, sus interacciones, sus metas u objetivos, sus organizaciones, sus percepciones del ambiente o entorno que los rodea y la evolución de su estado mental.

Para implementar el prototipo se utilizó JADE debido a que contiene librerías y clases en JAVA para desarrollar Sistemas MultiAgente y cuenta con una herramienta gráfica que permite monitorear la actividad de los agentes en tiempo real, así mismo, es hoy día una de las arquitecturas de desarrollo más utilizadas y se encuentra en constante evolución.

Este documento se encuentra organizado así: en los tres primeros capítulos tratan los enfoques o teorías que permiten la construcción de un sistema de negociación automatizada sólido y coherente, los cuales están compuestos en su orden por los siguientes temas: agentes inteligentes y sistemas multiagentes, teoría de juegos y computación evolutiva; para tal propósito se abarcaran cada uno de los conceptos que se estiman pertinentes de cada metodología para tener una clara visión del prototipo desarrollado, en el capítulo 4 se realiza una breve descripción del origen de la democracia y la evolución que está sufriendo en la actualidad gracias al surgimiento de nuevas disciplinas y al desarrollo tecnológico; en ese orden de ideas, en el capítulo 5 se explica la negociación automatizada que es en últimas el tema que engloba los temas tratados en los capítulos anteriores, así mismo, en este capítulo se dejará claro los problemas que se presentan al tratar de automatizar un proceso de negociación y en ese sentido como se está enfocando la investigación en este campo.

El capítulo 6 está dedicada al diseño e implementación del prototipo, se describe el diseño del prototipo usando la metodología INGENIAS [7] y la implementación usando la arquitectura JADE. Las pruebas al sistema se detallan en el capítulo 7. Así mismo, el capítulo 8 presenta aportes, contribuciones y divulgación de resultados presentados. El capítulo 9 está dedicado a las conclusiones y recomendaciones, sucesivamente se presentan las fuentes bibliográficas consultadas para la realización de

las diferentes etapas del proyecto de investigación. Además, se incluyen tres anexos compuestos por las especificaciones de la metodología INGENIAS [7], una descripción concisa del API de JADE y un manual del usuario que describe como se puede usar el prototipo desarrollado.

Capítulo 1

AGENTES INTELIGENTES Y SISTEMAS MULTIAGENTES

Los conceptos de Agente Inteligente y de Sistemas MultiAgentes (SMA) se han afianzado como una nueva aproximación para el desarrollo de sistemas que trascienden el ámbito de la Inteligencia Artificial (IA) y encuentra aplicación en muchas áreas de la informática. La tendencia actual de la IA se enfoca en el concepto de agente inteligente.

En ese orden de ideas, un agente inteligente es una entidad que hace lo correcto para cumplir con sus metas, con base en un ambiente en el cual se encuentra situado.

Así mismo, los SMA buscan lograr la cooperación de un conjunto de agentes autónomos para la realización de una tarea. La cooperación depende de las interacciones entre los agentes e incorpora tres elementos: la colaboración, la coordinación y la resolución de conflictos.

En este capítulo se ofrece una explicación detallada de la teoría de agentes que trata de explicar qué es un agente inteligente y las propiedades que incluyen, de igual forma, los ambientes en los cuales los agentes pueden estar situados (Sección 1.1.2 y 1.1.3); también se considerará la arquitectura de agentes que intentan dar metodologías generales para la construcción de estructuras de agentes y las tipologías que los clasifican (Sección 1.1.4 y 1.1.5). Así mismo, se describirán los sistemas multiagentes y sus características (Sección 1.2 y 1.2.1).

1.1. Agentes Inteligentes

En algunos problemas del mundo real se hace necesario debido a la complejidad del sistema y a las múltiples variables que lo afectan, establecer entidades que tengan

la capacidad de actuar por si mismas y ajustarse a las condiciones cambiantes e impredecibles del entorno para sacar el máximo provecho de este. A dichas entidades se les conoce como agentes inteligentes y permiten, así mismo, ser personalizados para realizar tareas de diversa índole.

1.1.1. Agentes

Actualmente existen diversas definiciones de agente originadas desde las diversas disciplinas. Debido a la gran variedad de dominios de aplicación de los agentes, estos han tomado características bien definidas del dominio particular al que se apliquen. Por tal razón, es oportuno dar una definición de agente en forma general:

Un agente es cualquier cosa que puede percibir su entorno a través de sensores y actuar sobre ese entorno a través de efectores [46].

Esto no quiere decir que el agente tenga dominio completo del ambiente, por el contrario en la mayoría de situaciones éste es muy complejo y el agente solo tendrá un control parcial. En la Figura 1.1 se muestra un esquema abstracto de cómo el agente interactúa con el entorno.

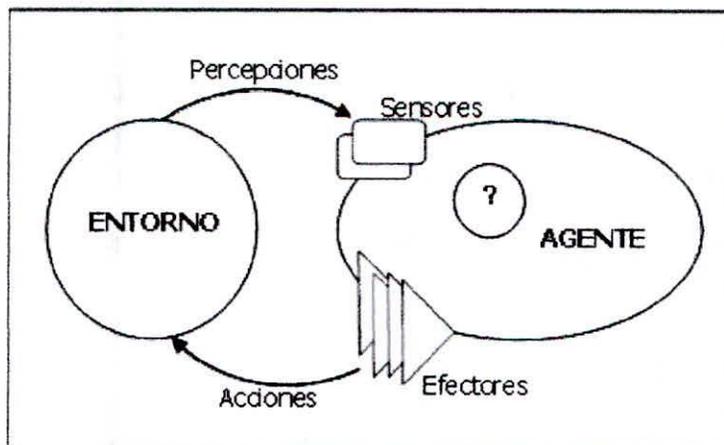


Figura 1.1: Abstracción de un agente a partir de su interacción con el medio ambiente o entorno.

1.1.2. Agentes inteligentes

La Inteligencia Artificial (IA) tiene como objetivo el estudio de las entidades inteligentes, pero a diferencia de la filosofía y la psicología, que tienen también que ver con el estudio de la inteligencia, la IA tiene como meta no únicamente la comprensión de estas entidades inteligentes, sino su construcción. El problema en definir agente inteligente radica en ¿Qué es Inteligencia? en IA. En ese sentido, la inteligencia en los agentes inteligentes está intrínsecamente asociada al comportamiento del agente, es decir, las acciones que un agente debe tomar ante un estado del entorno para cumplir sus objetivos y la capacidad del agente de interactuar con otros.

Una definición consensuada del término agente inteligente en IA puede ser:

Un agente inteligente es un sistema computacional que exhibe persistencia temporal, capaz de actuar de manera autónoma para satisfacer sus objetivos y metas, cuando se encuentra situado en algún medio ambiente [46, 39].

Esta definición puede parecer demasiado general, aún así, provee una abstracción del concepto de agente con base en su presencia y relación con un medio ambiente (Figura 1.1). Stuart Russell y Devika Subramanian [46] encuentran tres ventajas en esta abstracción:

- Nos permite observar las facultades cognitivas de los agentes al servicio de encontrar como hacer lo correcto.
- Permite considerar diferentes tipos de agente, incluyendo aquellos que no se supone tengan tales facultades cognitivas.
- Permite considerar diferentes especificaciones sobre los subsistemas que componen los agentes.

En [53] un agente inteligente es un agente que debe tener un comportamiento autónomo y flexible, donde flexible quiere decir que el agente debe ser reactivo, pro-activo y que tenga habilidades sociales. A continuación se explicarán cada una de las anteriores propiedades.

- Autonomía: Un agente debe funcionar sin la intervención directa de otros agentes o humanos, además debe tener control de sus acciones y su estado interno [53].
- Reactividad: Ya que un agente es capaz de percibir su entorno, éste debe responder con una acción oportuna ante un cambio ocurrido en el ambiente [39].
- Pro-Actividad: Los agentes deben tener un comportamiento orientado a objetivos, por lo tanto ante cualquier estado del ambiente el agente puede tomar la iniciativa de ejecutar acciones para satisfacer sus objetivos [39].
- Habilidad Social: Los agentes deben ser capaces de interactuar con otros agentes, aplicaciones e incluso humanos [39].

Jennings y Wooldridge en [39] distinguen dos nociones de agente: la primera hace referencia a una noción débil que se ajusta a la definición anterior de agente inteligente (igual definición y mismas propiedades), la segunda es una noción más fuerte en la que el agente inteligente debe estar en sistemas de computación (software), esto hace que se incluyan además de las propiedades de la noción débil otras que pueden ser de especial importancia según la aplicación que se le de al agente inteligente. A continuación se enuncian algunas de estas propiedades:

- Movilidad: Es la capacidad que tiene un agente de ir de un sistema a otro a través de una red de datos [53].
- Veracidad: Un agente no comunicará información falsa a otros agentes [53].
- Benevolencia: Un agente no debe tener conflictos con sus objetivos [53]].
- Racionalidad: Un agente siempre debe actuar en procura de sus logros [53] y tratar de maximizar su desempeño de acuerdo a una función de evaluación [39].
- Adaptabilidad: Un agente es capaz de aprender y cambiar su comportamiento [24].

Gracias a todas las características que presentan los agentes son vistos actualmente como una de las más vibrantes e importantes áreas de investigación y desarrollo;

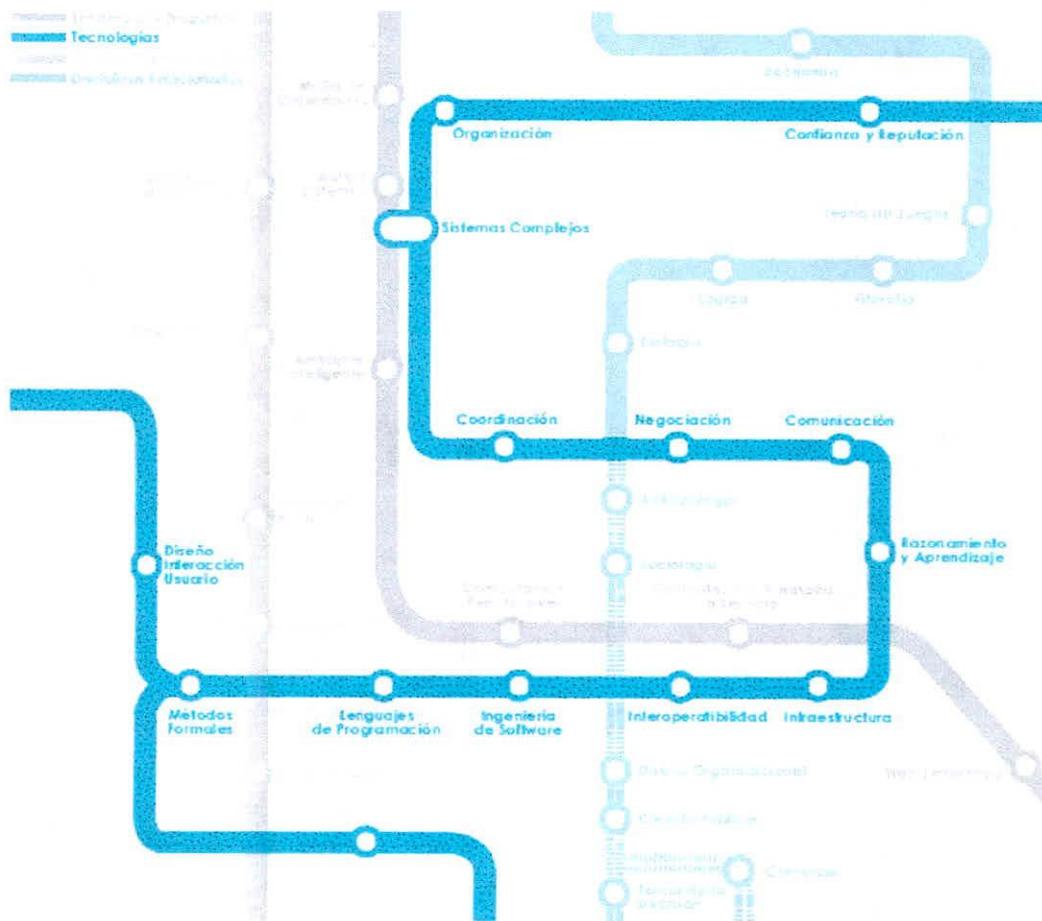


Figura 1.2: Impacto actual de los Agentes Inteligentes. Adaptado de [44].

es por tal razón, que son tenidos como recurso principal para la solución de problemas de diversa índole. La Figura 1.2 muestra las diversas tecnologías, técnicas relacionadas, disciplinas relacionadas, tendencias y programas que están haciendo uso de los agentes inteligentes.

1.1.3. Ambiente

Por ambiente, entendemos el espacio donde un agente, o un grupo de ellos, se encuentran situados. Rodney Brooks [22] argumenta que el medio ambiente por excelencia es el mundo real, y en su propuesta todo agente toma una forma robótica. Por el contrario, Oren Etzioni [18], considera que no es necesario que los agentes tengan implementaciones robóticas porque los ambientes virtuales, como los sistemas operativos y el Web, son igualmente válidos que el mundo real. En este documento asumimos la posición de Etzioni, resaltando que lo importante es que la interacción del agente con su ambiente se de en los términos señalados en nuestra definición de

agente, esto es, en forma autónoma bajo persistencia temporal. Stuart Russell y Peter Norvig [46] señalan que, más allá de esta controversia, es importante identificar que existen diferentes tipos de ambientes.

1.1.3.1. Accesibles vs. No Accesibles

Si un agente puede percibir a través de sus sensores, los estados completos del ambiente donde se encuentra, se dice que el ambiente es accesible. Un ambiente es efectivamente accesible si el agente puede percibir todos aquellos elementos en el ambiente, que son relevantes para su toma de decisiones. Observen que la accesibilidad depende no sólo del ambiente, sino de las capacidades de percepción de un agente. Entre más accesible sea un ambiente, más sencillo será construir agentes en él porque estos tienen acceso a toda la información necesaria para decidir que hacer [46].

1.1.3.2. Deterministas vs. No Deterministas

Si el próximo estado del ambiente es determinado por la acción que ejecuta el agente, se dice que el ambiente es determinista. Si otros factores influyen en el próximo estado del ambiente, éste es no determinista. Si el ambiente es inaccesible, entonces aparecerá como no determinista. Generalmente es mejor considerar el determinismo o no determinismo del ambiente, desde el punto de vista del agente. El no determinismo captura dos nociones importantes: El hecho de que los agentes tienen una esfera de influencia limitada, es decir, en el mejor de los casos tienen un control parcial de su ambiente; y el hecho de que las acciones de un agente pueden fallar y no lograr el resultado deseado por el agente. Por ello, es más sencillo construir agentes en ambientes deterministas [46].

1.1.3.3. Episódicos vs. No Episódicos

En un ambiente episódico, la experiencia de un agente puede evaluarse en rondas de percepción - acción. La calidad de su acción depende del episodio o ronda en turno, esto es, la calidad de la acción en los episodios subsecuentes, no depende de las acciones ocurridas en episodios previos. Dada la persistencia temporal de los agentes, estos tienen que tomar continuamente decisiones locales que tienen consecuencias globales. Los episodios reducen el impacto de estas consecuencias, y por lo tanto es más fácil construir agentes en ambientes episódicos [46].

1.1.3.4. Estáticos vs. Dinámicos

Si el ambiente puede cambiar mientras el agente se encuentra deliberando, se dice que el ambiente es dinámico; de otra forma, el ambiente es estático. Si el ambiente no cambia con el paso del tiempo, pero la evaluación de las acciones del agente si lo hacen, se dice que el ambiente es semi-dinámico. Los ambientes dinámicos tienen dos consecuencias importantes: Un agente debe ejecutar acciones preceptuales, por ejemplo, filtrado de información, porque aún si no ha ejecutado ninguna acción entre los tiempos t_0 y t_1 , el agente no puede asumir que el estado del ambiente sea el mismo en t_0 que en t_1 ; otros procesos en el ambiente pueden interferir con las acciones del agente. Por lo tanto, es más sencillo diseñar agentes en ambientes estáticos [46].

1.1.3.5. Discretos vs. Continuos

Si hay un numero limitado de posibles estados del ambiente, diferenciables y claramente definidos, se dice que el ambiente es discreto; de otra forma se dice que es continuo. Es más fácil construir agentes en ambientes discretos, porque las computadoras también son sistemas discretos y aunque pueden simular sistemas continuos con el grado de precisión deseado, una parte de la información disponible se pierde en al hacer esta aproximación. Por lo tanto, la información que manejan los agentes discretos en ambientes continuos es inherentemente aproximada [46].

1.1.4. Arquitecturas para agentes inteligentes

Una arquitectura permite descomponer un sistema computacional en componentes más pequeños y determinar cómo es la relación entre estos. Una arquitectura para agentes en particular permite descomponer un sistema de agentes, y determinar cómo deben interactuar entre ellos y con el ambiente.

En [53] se distinguen tres arquitecturas diferentes que se clasifican según el modelo de razonamiento que utilizan.

1.1.4.1. Arquitecturas deliberativas

Una arquitectura deliberativa es aquella en la que el comportamiento y conocimiento de los agentes están explícitamente representados mediante un modelo simbólico. El modelo simbólico consta generalmente de un conjunto físicamente realizable de

entidades y estructuras. Además de esto en la arquitectura deliberativa las decisiones de los agentes son hechas mediante razonamiento lógico o pseudo lógico.

1.1.4.2. Arquitecturas reactivas

Una arquitectura reactiva es aquella que no incluye alguna clase central de modelo simbólico del mundo y no usa un razonamiento simbólico complejo. Las acciones que toman sus entidades se basan en una asignación de lectura del medio ambiente con una operación a efectuar.

1.1.4.3. Arquitecturas híbridas

Los investigadores sugieren que ninguna de las dos arquitecturas vistas anteriormente es del todo apropiada para la construcción de agentes, por eso se plantean soluciones de arquitectura híbrida en la cual se integran los dos métodos (deliberativa y reactiva).

Una propuesta de tal arquitectura es la construcción con base a dos subsistemas: uno deliberativo en el cual los planes desarrollados y las decisiones hechas sean planteadas por un conjunto de símbolos y uno reactivo el cual tenga la capacidad de reaccionar ante eventos del entorno sin emplear razonamiento complejo.

1.1.5. Tipología de agentes

La existencia de diferentes tipos de agentes se debe a la gran cantidad de aplicaciones en las que pueden incursionar y en el ambiente en que trabajan. El primer aspecto se refiere a los roles que los agentes pueden tomar al trabajar en diferentes aplicaciones por ejemplo ayudantes personales, agentes móviles, knowbots (agentes de conocimiento), entre otros. El segundo aspecto tiene que ver con la apariencia de los agentes, es decir en donde se sitúan, trabajan o ejecutan sus acciones, por ejemplo los agentes que trabajan en el mundo físico son denominados robots, los agentes que trabajan como un programa de computador son llamados softbots o agentes de software.

Una tipología de agentes se refiere al estudio y clasificación de diferentes tipos de entidades. En primer lugar se verá cómo se pueden clasificar los agentes desde diferentes aspectos y segundo se dará una explicación más detallada sobre algunas clases de agentes de software.

1.1.5.1. Clasificación de los agentes inteligentes

A continuación se muestra la clasificación de los agentes hecha por Nwana en [24].

- Movilidad: Los agentes pueden ser clasificados por su movilidad, aquellos que se pueden mover de un host a otro a través de una red se denomina agentes móviles, aquellos que no tienen esta propiedad son agentes estáticos.
- Arquitectura de construcción: Los agentes se pueden clasificar según los paradigmas de arquitecturas para construcción de agentes que se vieron en la Sección 1.1.4.
- Atributos principales: Los agentes se clasifican de acuerdo a los principales atributos que pueden exhibir. En [24] se define como atributos principales de un agente la autonomía, la adaptabilidad y las habilidades sociales. De esta clasificación se derivan las clases de agentes colaborativos, agentes de interfaz, agentes de aprendizaje colaborativo y agentes verdaderamente inteligentes. En la Figura 1.3 se muestra un diagrama representativo de esta clasificación.

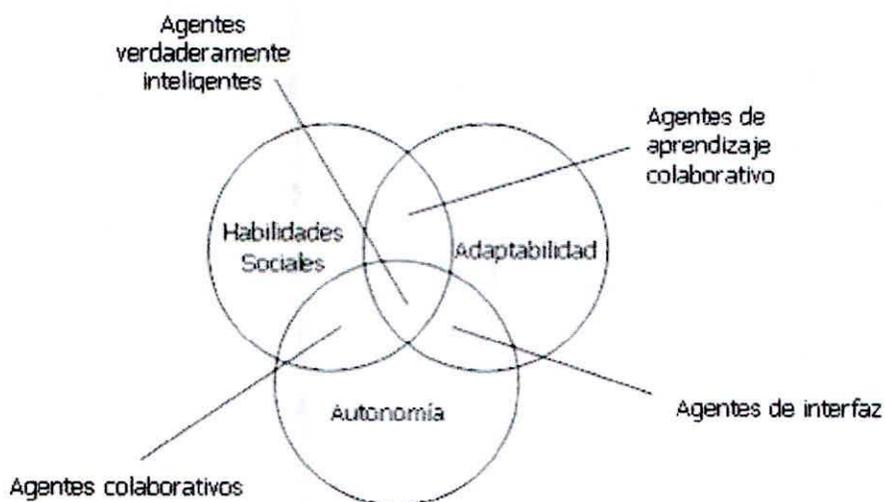


Figura 1.3: Una clasificación de los agentes inteligentes. Adaptado de [24].

- Roles: Los agentes pueden clasificarse algunas veces por sus roles, preferiblemente cuando son especiales, por ejemplo los agentes de información que exploran fuentes de información en busca de resultados para un usuario, otros ejemplos son los agentes de presentación, agentes de prueba, agentes de ayuda, entre otros.

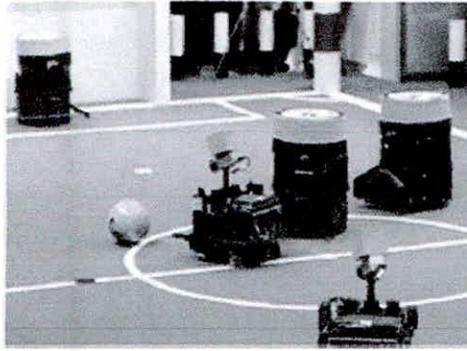


Figura 1.4: Robots que juegan fútbol.

- Clases de agentes inteligentes: Existen diferentes clases de agente inteligentes. En las Secciones 1.3.4.2 y 1.3.4.3 se explicarán dos clases de agentes que se utilizan frecuentemente en sistemas de agentes.

1.2. Sistemas Multiagentes

El enfoque moderno de la IA está centrado alrededor del concepto de agente inteligente. Sin embargo, los agentes pocas veces hacen todo sin la ayuda del sistema. En muchas situaciones ellos coexisten e interactúan con otros agentes en varias formas diferentes.

Ejemplos incluyen agentes software sobre Internet, robots que juegan fútbol (ver Fig. 1.4), y mucho más. A tales sistemas que consisten de un grupo de agentes que pueden potencialmente interactuar entre sí, son denominados como Sistemas MultiAgentes (SMA) y la correspondiente rama de IA que trata con los principios y diseños de SMA es conocida como IA Distribuida (IAD) [17].

Según [41] los SMA son sistemas de computadora distribuida en el cual los diseñadores le atribuyen al componente módulos de autonomía, estado mental y otras características de entidad. Los desarrolladores de software han aplicado SMA para resolver problemas en administración de energía, programación del transporte, y muchas otras tareas.

En el estudio de SMA, incluyendo el campo de IAD y mucha de la actividad actual en “agentes software”, los investigadores apuntan a relacionar el comportamiento agregado del sistema complejo con el comportamiento individual de los agentes integrantes y las propiedades del protocolo de interacción y el entorno. La estructura

para la construcción y análisis de los SMA frecuentemente deriva sobre metáforas —así como también modelos y teorías— desde las ciencias sociales y ecológicas. Tales concepciones sociales son algunas veces aplicadas dentro de un agente para describir su comportamiento en términos de interacciones de sub-agentes, como en la teoría de *Society of Mind* de Minsky [41].

El diseño de SMA típicamente se enfoca sobre el mecanismo de interacción, especificación del lenguaje de comunicación del agente y el protocolo de interacción. El mecanismo de interacción generalmente incluye el medio para implementar decisiones o alcanzar acuerdos como una función de la interacción de los agentes. Dependiendo del contexto, los desarrolladores de un SMA podrían también controlar la configuración de los agentes participantes, la arquitectura del agente inteligente, o aún la implementación de los agentes mismos. En algunos casos, el diseño está basado en principios del mecanismo de interacción y el diseño de los agentes requiere un modelo de las reglas del mecanismo y (algunas veces) modelos de los otros agentes.

1.2.1. Características de un Sistema MultiAgente

Un SMA está compuesto por los siguientes aspectos fundamentales:

1.2.1.1. Diseño del Agente

Es frecuente que se presente el caso de que varios agentes que comprenden un SMA sean diseñados en formas diferentes. Un ejemplo típico son agentes software, que han sido implementados por diferentes personas. En general, las diferencias de diseño podrían involucrar el hardware (por ejemplo robots jugadores de fútbol basados en diferentes plataformas mecánicas), o el software (por ejemplo agentes software corriendo en diferentes sistemas operativos). Frecuentemente se dice que tales agentes son heterogéneos en contraste a los agentes homogéneos que son diseñados en forma idéntica y tienen a primera vista las mismas capacidades. Sin embargo, esta distinción no está bien definida; los agentes que están basados sobre el mismo hardware/software pero implementan diferentes comportamientos pueden también ser llamados heterogéneos. La heterogeneidad de los agentes puede afectar todos los aspectos funcionales de un agente desde la percepción hasta la toma de decisión, mientras que un sistema compuesto por un solo agente, este problema simplemente no existe [17].

1.2.1.2. Entorno

Los agentes tienen que tratar con entornos que pueden ser también estáticos (tiempo invariante) o dinámicos (no estacionarios). Muchas técnicas existentes en IA para un solo agente han sido desarrolladas para entornos estáticos porque estos son más fáciles de manejar y permiten un tratamiento matemático más riguroso. En un SMA, la mera presencia de múltiples agentes convierte el entorno dinámico desde el punto de vista de cada agente. Esto puede frecuentemente ser problemático [17].

1.2.1.3. Percepción

La información colectiva que alcanzan los sensores de los agentes en un SMA es típicamente distribuida: los agentes pueden observar que difiere espacialmente (que aparece en distintos puntos), temporalmente (llegar en tiempos diferentes), o aún semánticamente (requiere diferentes interpretaciones). Esto automáticamente hace al mundo parcialmente observable para cada agente, lo cual tiene varias consecuencias en la toma de decisiones de los agentes. Un problema adicional es la fusión del sensor, que es, como los agentes pueden combinar óptimamente sus percepciones para incrementar su conocimiento colectivo acerca del estado actual [17].

1.2.1.4. Control

Contrario a los sistemas de un solo agente, el control en un SMA es típicamente distribuido (descentralizado). Esto significa que no hay un proceso central que recolecta información desde cada agente y entonces decide que acción debería tomar cada agente. La toma de decisión de cada agente yace en gran medida en el agente mismo. El problema general de la toma de decisión multiagente es el tema de la teoría de juegos. En un SMA cooperativo, la toma de decisión distribuida resulta en un cálculo asíncrono y rápido, pero también tiene la desventaja que el mecanismo de negociación apropiado necesita ser desarrollado por añadidura. La coordinación asegura que la decisión individual del agente resulta en una buena decisión conjunta para el grupo [17].

1.2.1.5. Conocimiento

En un sistema compuesto por un solo agente se asume que el agente conoce sus propias acciones, pero no necesariamente como el mundo es afectado por sus acciones. En un SMA, el nivel de conocimiento de cada agente acerca del estado del

mundo actual puede diferir sustancialmente. Por ejemplo, en un SMA cooperativo que esté formado por dos agentes homogéneos, cada agente puede conocer el conjunto de acciones disponibles del otro agente, ambos agentes pueden saber (por comunicación) su percepción actual, o ellos pueden inferir la intención de cada uno de los otros agentes basados sobre algún conocimiento compartido a priori. Por otro lado, un agente que observa un equipo adversario de agentes normalmente desconocerá sus conjuntos de acciones y sus percepciones actuales y podría también ser incapaz para inferir sus planes.

En general, en un SMA cada agente debería también considerar el conocimiento de cada uno de los otros agentes en su toma de decisión. Un concepto crucial aquí es el de *conocimiento común*, de acuerdo al cual cada agente conoce un hecho, cada agente conoce que otro agente conoce este hecho, y así sucesivamente [17].

1.2.1.6. Comunicación

La interacción está frecuentemente asociada con alguna forma de comunicación. Generalmente la comunicación es vista en un SMA como un proceso bidireccional, donde todos los agentes pueden potencialmente ser emisores y receptores de mensajes. La comunicación puede ser usada en varios casos, por ejemplo, para coordinación entre agentes cooperativos o para negociación entre ellos mismos. Además, la comunicación adicionalmente plantea el problema de que los protocolos de red usados para intercambiar información sean seguros y oportunos, así mismo, que lenguaje los agentes deberían hablar para entenderse (especialmente si ellos son heterogéneos) [17].

Capítulo 2

TEORÍA DE JUEGOS: CONCEPTOS Y EJEMPLOS

La teoría de juegos es un área de trabajo diseñada para analizar la interacción entre diferentes agentes cuyas decisiones de cada uno afectan al otro. En el análisis de un juego teórico, una situación interactiva es descrita como un juego: una descripción de los jugadores (agentes), el curso de las acciones disponibles para ellos y sus preferencias sobre los posibles resultados.

El área de trabajo del juego teórico asume que los jugadores emplean toma de decisión racional, es decir, ellos actúan de tal forma para lograr los resultados que prefieren (von Neumann y Morgenstern 1994). Generalmente, las preferencias son modeladas usando utilidades numéricas y los jugadores asumen maximizar la utilidad esperada [41].

La teoría de juegos analiza, básicamente, la toma de decisiones en términos de construcciones competitivas (juegos no-cooperativos) y coalicionales (juegos cooperativos) abstraídas de los juegos de salón (póquer, bridge, monopolio, etc.), en los cuales dos o más agentes, considerando las acciones de sus oponentes, deben tomar decisiones en el esfuerzo por obtener las máximas ganancias posibles. Esta abstracción ha abierto el espectro de posibilidades de aplicación al mundo real: los “jugadores” pueden ser seres humanos, instituciones, poblaciones de animales, partidos políticos, agentes de un mercado, etc., a la vez que las estrategias pueden ser de muy diversa índole. La teoría de juegos tiene en estos campos una habilidad única: *la de ser un sistema de referencia para el estudio de las interacciones, descrito en términos simples y universales* [31].

En este capítulo se ofrece una explicación concisa de los elementos de la teoría de juegos y las formas como se pueden representar los juegos (Sección 2.1, 2.2 y 2.3), así mismo, se describen las estrategias dominantes, los juegos no-cooperativos y

cooperativos (Sección 2.4, 2.5 y 2.6) ya teniendo claros estos conceptos se procede a tratar temas cruciales para el desarrollo del prototipo de negociación automatizada como el dilema del prisionero, el equilibrio de Nash y la eficiencia de Pareto (Sección 2.7, 2.8 y 2.9).

2.1. Elementos de la Teoría de Juegos

Los elementos presentes en todo juego son: jugadores, acciones, información, estrategias, retribuciones, resultados y equilibrio:

2.1.1. Jugadores

Los individuos que toman las decisiones tratando de obtener el mejor resultado posible, o sea maximizar su utilidad. Se utiliza en algunos juegos la figura de un pseudo jugador, usualmente llamado *entorno*. En realidad la denominación no es correcta pues corresponde en muchas ocasiones a la cuantificación de varios jugadores individuales cuyas reacciones no se pueden modelar en términos sencillos o ni siquiera asumir. Rasmusen [32] también señala a este pseudo jugador como uno que actúa en momentos específicos del juego con específicas probabilidades; lo cual implica conocimiento perfecto de la distribución probabilística (juegos de información completa) o no (juegos de información incompleta). Quedan pendientes, sin embargo, situaciones en las cuales la información es parcial, es decir, no sé cuánto es aquello que no sé.

2.1.2. Acción

Es una de las opciones que cada jugador tiene disponible en un determinado instante para alcanzar el objetivo buscado. El orden del juego determina en qué momento esas acciones están disponibles. Un perfil de acciones es un conjunto de una acción por cada uno de los jugadores del juego.

2.1.3. Información

Es el conocimiento, en un determinado momento, de los valores de las distintas variables (en general, la historia del juego).

2.1.4. Estrategia

Es un conjunto de acciones decididas con antelación y disponibles en cada momento del juego según la información. Un perfil de estrategias es un conjunto de una estrategia por cada uno de los jugadores del juego.

2.1.5. Retribución

Es la utilidad o beneficio que reciben los jugadores al completar el juego y corresponde a la evaluación posterior a la realización de la acción sobre si el objetivo buscado fue alcanzado. También debemos tener en cuenta la utilidad esperada, ya que es ésta en realidad la que motiva la acción.

2.1.6. Resultado

Son las conclusiones que el modelador obtiene una vez que el juego ha concluido.

2.1.7. Equilibrio

Es un perfil de estrategias integrado simultáneamente por la mejor estrategia para cada uno de los jugadores del juego. A menudo existe confusión entre *equilibrio* y *resultado de equilibrio*, y *estrategia* y *acción*. La diferencia, efectivamente, no resulta tan clara porque en algunos casos una estrategia puede estar conformada tan solo por una acción.

2.2. Forma Normal

La forma normal de un juego (también llamada forma estratégica) es la base del tipo de juegos estudiados en la teoría de juegos no cooperativos. Un juego en su forma normal es una lista de todas las estrategias de cada uno de los jugadores, junto con las retribuciones esperadas para cada posible elección de combinación de estrategias (una para cada agente). El resultado es representado por separado para cada jugador por una retribución, la cual es una cifra (conocida como utilidad) que mide cuanto el jugador quiere obtener [52].

2.3. Forma Extensiva

La forma extensiva (conocida como árbol del juego), es más detallada que la forma normal de un juego. Es una descripción completa de cómo es ejecutado el juego a través del tiempo. Esto incluye el orden en cual los jugadores toman acciones, le información que los jugadores tiene en el tiempo que ellos deberían tomar estas acciones y el tiempo en el que cualquier incertidumbre es resuelta. Un juego en la forma extensiva puede ser analizado directamente, o puede ser convertido a su equivalente en la forma normal [52].

2.4. Estrategias Dominantes

La estrategia s_i^* una estrategia dominante si es estrictamente la mejor respuesta para un jugador sin importar la estrategia que los otros jugadores elijan, es decir, le es indiferente la estrategia que ellos elijan; la retribución más alta la siempre la obtendrá con s_i^* . Matemáticamente,

$$\pi_i(s_i^*, s - i) > \pi_i(s'_i, s - i) \forall s - i, \forall s'_i \neq s_i^*. \quad (2.1)$$

2.5. Juegos No Cooperativos

Una gran parte de la teoría de juegos trata con situaciones no cooperativas, en donde cada jugador actúa independientemente. En tal juego, una estrategia S_i para un jugador i especifica la acción que el jugador debe tomar en cualquier estado del juego. Una solución para el juego es una combinación de estrategias S_1, \dots, S_n que satisfacen ciertas condiciones óptimas.

Más abstractamente, una situación es presentada como un juego en forma estratégica, en donde las posibles estrategias para los jugadores son simplemente enumeradas. Cada combinación de estrategias S_1, \dots, S_n conduce a algún resultado, cuyo valor para el jugador i es una retribución $u_i(S_1, \dots, S_n)$. Un juego en forma estratégica de dos jugadores es a menudo representado por una matriz en donde las filas son las estrategias del jugador 1, las columnas son las estrategias del jugador 2, y las entradas de la matriz son los pares de retribución asociados [41].

2.6. Juegos Cooperativos

Un juego cooperativo es una situación derivada de una actividad en la que los elementos o jugadores que intervienen pueden establecer contratos para actuar en forma mancomunada (personas, instituciones, empresas, etc.) para alcanzar un determinado objetivo (ganar una votación, buscar mayores beneficios empresariales, mejorar una gestión, etc.) mediante la colaboración entre ellos.

A diferencia de los denominados juegos no cooperativos – caracterizados por las estrategias que pueden emplear cada uno de los jugadores y una función de pagos asociada a cada jugador, la cual depende de las diferentes estrategias que se empleen –, en un juego cooperativo no es necesario analizar las estrategias de los jugadores; es suficiente conocer los pagos asociados a los resultados del juego [14].

Si la utilidad de los jugadores es transferible, un *juego cooperativo*, es un par (N, v) , donde N es un conjunto finito y v es una función $v : 2^N \rightarrow R$, que verifica $v(\emptyset) = 0$. Los elementos de $N = \{1, 2, \dots, n\}$ se denominan jugadores, los subconjuntos $S \in 2^N$ coaliciones y $v(S)$ es el valor de la coalición S .

Como anota Bilbao [35] el valor de una coalición es, por establecer algún paralelismo, análogo al valor del juego que se determina en los juegos bipersonales y es igual a la cantidad mínima que puede obtener la coalición si todos sus miembros se asocian y juegan en equipo. La función v se denomina habitualmente función característica del juego, siendo identificado – siempre que no haya lugar a confusión – el juego (N, v) mediante su función característica.

2.7. El Dilema del Prisionero

Veamos el siguiente caso. Dos prisioneros A y B acusados de algún delito van a ser interrogados por separado. A cada uno se le preguntará por la culpabilidad del otro. Cada prisionero puede optar por “colaborar” con el otro, asegurando que el compañero se encuentra injustificadamente en la cárcel, o “defraudar”acusándole. Existen por tanto cuatro posibilidades, que ninguno defraude, que lo hagan los dos, que lo haga el primero o el segundo [20].

Cada prisionero recibe un premio en función de la Tabla 2.1. El -1 es una pena de cárcel de 10 años, el 0 es una pena de cárcel de 5 años, el 3 la libertad y el 5 la libertad y una indemnización.

		A	
		No Confiesa	Confiesa
B	No Confiesa	3	5
	Confiesa	-1	0
		5	0

Tabla 2.1: Matriz de Pagos en el Dilema del Prisionero.

Supongamos que somos el jugador B . Si el jugador A defrauda (confiesa) es indiferente lo que hagamos, ya que recibiremos el castigo. Pero si el jugador A coopera con nosotros (no confiesa), el premio recibido es mayor si nosotros a su vez le defraudamos a él. En resumen, haga lo que haga el jugador A , para el jugador B lo mejor es defraudar (confesar) y lo mismo ocurre para el jugador A : haga lo que haga el jugador B , lo mejor es defraudar (confesar). Como podemos observar la estrategia dominante para cada jugador es confesar. Ya que las decisiones son independientes, y dado que el objetivo de cada uno es lograr el máximo beneficio personal, lo racional es defraudar. En ese sentido, como los agentes se comportan racionalmente, ambos recibirán el castigo [45].

2.8. Equilibrio de Nash

Una combinación estratégica (S_i^*, S_{-i}^*) (Si *, S-i *) es un equilibrio de Nash si para cualquier agente que se desvíe de forma unilateral, este obtendrá menos retribución. Matemáticamente:

$$\forall i, \pi_i(S_i^*, S_{-i}^*) > \pi_i(S_i', S_{-i}^*), \forall S_i' \quad (2.2)$$

El concepto central del equilibrio de Nash es recomendar una estrategia a cada jugador que es imposible mejorar de manera unilateral, esto es, dado que los otros jugadores sigan dicha recomendación. Puesto que los otros jugadores son racionales, es razonable para cada jugador esperar que sus oponentes sigan la recomendación igualmente.

2.9. Eficiencia de Pareto

Una combinación estratégica \vec{x}^* es una eficiencia de Pareto si no existe otra combinación estratégica $\vec{x} \in F$ tal que $f_i(\vec{x}) \geq f_i(\vec{x}^*)$ para todo $i = 1, \dots, k$ y para $f_j(\vec{x}) > f_j(\vec{x}^*)$ al menos un j .

En palabras, esta definición dice que \vec{x}^* es una eficiencia de Pareto si no existe una combinación estratégica factible $\vec{x} \in F$ la cual debería incrementar algún criterio sin causar una simultánea disminución en al menos otro criterio. Desafortunadamente, este concepto casi siempre no brinda una única solución, sino más bien un conjunto de soluciones conocido como conjunto de Pareto. Los conjuntos de combinaciones \vec{x}^* correspondientes a la solución incluida en el conjunto de Pareto son llamadas no dominados. La gráfica de la función objetivo cuyas combinaciones estratégicas no son dominadas, corresponden al conjunto de Pareto y son conocidas como la frontera de Pareto.

Capítulo 3

COMPUTACIÓN EVOLUTIVA

En las décadas de 1950 y 1960 varios científicos estudiaron independientemente sistemas evolutivos con la idea de que la evolución podría ser usada como una herramienta de optimización para problemas de ingeniería [16].

En [25] se define la Computación Evolutiva (CE) como un enfoque alternativo para abordar problemas complejos de búsqueda y aprendizaje a través de modelos computacionales de procesos evolutivos. Las implantaciones concretas de tales modelos se conocen como Algoritmos Evolutivos (AE). El propósito genérico de los AE consiste en guiar una búsqueda estocástica haciendo evolucionar a un conjunto de estructuras y seleccionando de modo iterativo las más adecuadas.

Bajo el término de Computación Evolutiva nacieron un amplio conjunto de técnicas de resolución de problemas complejos basadas en la emulación de los procesos naturales de evolución, entre estos AE que surgieron podemos destacar:

- Estrategias Evolutivas (EE): Se hace evolucionar una población de números reales que codifican las posibles soluciones de un problema numérico y los tamaños de salto. La selección es implícita, esto permite tratar la optimización de funciones continuas [11].
- Programas Evolutivos (PE): Se hace evolucionar una población de estructuras de datos sometiénolas a una serie de transformaciones específicas y a un proceso de selección [56].
- Algoritmos Genéticos (AG): Se hace evolucionar una población de enteros binarios sometiénolos a transformaciones unitarias y binarias genéricas y a un proceso de selección; esto permite optimización de funciones discretas y continuas, síntesis de sistemas, afinamiento, prueba, etc. [13].

- Programación Genética (PG): Se evolucionan programas de computador para resolver programas aproximadamente, tales como expresiones ejecutables para predecir series periódicas, etc. [12].

Se debe dejar claro en todo momento como anota Pérez [25] que no se persigue una simulación de los procesos naturales, sino más bien una emulación de dichos procesos. Por eso, un AE será tanto mejor cuanto mejores resultados proporcione en la resolución del problema planteado, independientemente de su fidelidad a la biología. De hecho, la mayoría de algoritmos que se derivan de este enfoque son exageradamente simplistas desde un punto de vista biológico, pero lo suficientemente complejos como para proporcionar robustos y potentes mecanismos de búsqueda.

Como anota [3] en su perspectiva histórica y comparación de estos paradigmas, las tres principales líneas de investigación - algoritmos genéticos, estrategias evolutivas y programación evolutiva - comparten muchas similitudes. Cada una mantiene una población de soluciones de prueba, imponiendo cambios aleatorios a estas soluciones e incorporando selección para determinar cuales soluciones se mantendrán en generaciones futuras. Fogel [3] también anota que los AG enfatizan en los modelos de operadores genéticos como se observaron en la naturaleza, tales como el cruce, inversión y mutación; y aplica estos a cromosomas elegidos. Mientras que las EE y los PE hacen énfasis en la mutación para la reproducción que mantiene la conexión del comportamiento entre cada padre y sus hijos.

En lo sucesivo, este capítulo, se limitará al análisis de los AG, para tal fin se expondrá la estructura básica de los AG y la terminología básica (Sección 3.1.1 y 3.1.2), de igual forma, se describirá los operadores que utilizan los AG, los mecanismos de selección de los AG y los criterios que se deben tener en cuenta para implantar un AG (Sección 3.1.3, 3.1.4 y 3.1.5).

3.1. Algoritmos Genéticos

Los Algoritmos Genéticos son métodos estocásticos de búsqueda ciega (no disponen de ningún conocimiento específico del problema) de soluciones cuasi-óptimas [25]. En este sentido, un AG permite a una población compuesta de muchos individuos evolucionar bajo unas reglas específicas de selección que maximizan la “aptitud” (i.e., minimizan la función de costo) [38].

Algunas de las ventajas de los AG incluyen:

- Optimiza con variables discretas o continuas.
- No requieren información derivada.
- Trata con un gran número de variables.
- Se adapta bien a computadores en paralelo.
- Optimiza variables con costo de superficie altamente complejo (ellos pueden saltar fuera de un mínimo local).
- Proveen una lista de variables óptimas, no solo una única solución.
- Pueden codificar las variables así que la optimización es hecha con variables codificadas, y
- Trabajan con datos generados numéricamente, datos experimentales, o funciones analíticas.

Estas ventajas son intrigantes y producen resultados asombrosos cuando la técnica tradicional de optimización falla pobremente [38]. Es conveniente anotar que la principal innovación de los AG en el dominio de los métodos de búsqueda es la adición de un mecanismo de selección de soluciones. Recuérdese que la selección tiene dos vertientes: a corto plazo los mejores tienen más posibilidades de sobrevivir y a largo plazo los mejores tienen más posibilidades de tener descendencia. A causa de esto, el mecanismo de selección se desdobra en dos: uno elige los elementos que se van a transformar posteriormente, este es el operador de selección a secas; el otro operador elige los elementos que van a sobrevivir, por lo que se le llama reemplazo [25].

3.1.1. Estructura de un Algoritmo Genético

La estructura genérica del bucle básico de un AG se sintetiza en la siguiente figura:

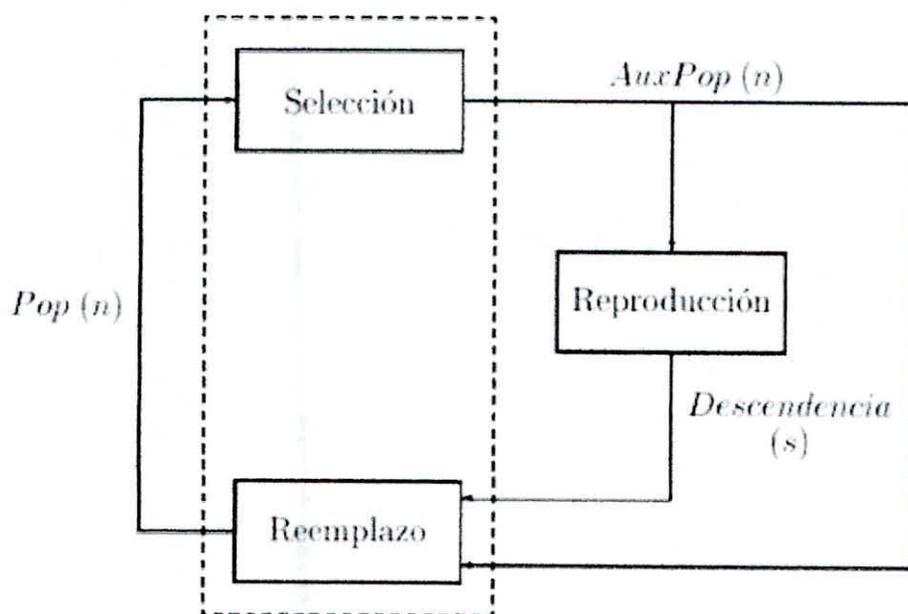


Figura 3.1: Ciclo básico de un AG.

Como puede observarse en la figura 3.1, una población Pop , que consta de n miembros se somete a un proceso de selección para constituir una población intermedia $AuxPop$ de n criadores. De dicha población intermedia se extrae un grupo reducido de individuos llamados progenitores que son los que efectivamente se van a reproducir. Sirviéndose de los operadores genéticos (Sección 3.1.3), los progenitores son sometidos a ciertas transformaciones de alteración y recombinación en la fase de reproducción en virtud de las cuales se generan s nuevos individuos que constituyen la *Descendencia*. Para formar la nueva población $Pop[t + 1]$ se deben seleccionar n supervivientes de entre los $n + s$ de la población auxiliar y la descendencia, eso se hace en la fase de reemplazo. Como ya se comentó, la selección (recuadro rayado) se hace en dos etapas con la idea de emular las dos vertientes del Principio de Selección Natural: selección de criadores o ‘selección’ a secas y selección de supervivientes para la próxima generación o ‘reemplazo’.

3.1.2. Terminología Básica sobre Algoritmos Genéticos

Alguna de la terminología básica usada por la comunidad de investigadores en AG es la siguiente [2]:

3.1.2.1. Cromosoma

Es una estructura de datos que contiene un arreglo de parámetros asignados, o genes. Este arreglo puede ser almacenado, por ejemplo, como un conjunto de bits (representación binaria) o como un arreglo de enteros (representación en punto flotante o cifras reales) que representan un número en punto flotante. Este cromosoma es análogo a los cromosomas en base 4 presentes en nuestro propio ADN. Normalmente, en la comunidad de AG, es asumido el modelo monosomático de una célula (un cromosoma individual). Sin embargo, el modelo diploide también ha sido usado en el pasado.

3.1.2.2. Gen

Es una subdivisión de un cromosoma que usualmente codifica el valor de un solo parámetro.

3.1.2.3. Alelo

Es el valor de un gen. Por ejemplo, para una representación binaria cada gen puede tener un valor de 0 o 1, y para una representación en punto flotante, cada gen puede tener un valor desde 0 a 9.

3.1.2.4. Esquema

Es un patrón de valores de gen en un cromosoma, el cual puede incluir estados "que no le importan" (representados por un * como símbolo). De esa manera en un cromosoma binario, cada esquema puede ser especificado por un arreglo de la misma longitud del cromosoma, con cada carácter siendo uno de 0, 1 o *. Un cromosoma particular se dice "contiene" un esquema particular si corresponde a ese esquema (e.g. el cromosoma 01101 corresponde al esquema *1*0*).

3.1.2.5. Espacio de Búsqueda

Si la solución de un problema puede ser representado por un conjunto de N valores reales, entonces el trabajo de encontrar esta solución puede ser concebida como una búsqueda en un espacio de N-dimensiones. Esta región es simplemente conocida como el espacio de búsqueda del problema.

3.1.2.6. Aptitud (Fitness)

Es el valor de rendimiento de un individuo (i.e., que tan bien resuelve una cierta tarea). Una función de aptitud es una representación de los cromosomas en una población para su correspondiente valor de aptitud.

3.1.2.7. Bloque de Construcción

Es una pequeña parte, un grupo fuertemente aglomerado el cual tiene que co-evolucionar de tal forma que su introducción dentro de cualquier cromosoma posibilitará incrementar la aptitud de ese cromosoma.

3.1.2.8. Decepción

Es una condición bajo la cual la combinación de un buen bloque de construcción conduce a reducir la aptitud, más que incrementar la aptitud. Esta condición fue propuesta por Goldberg [4] como una razón para el fallo de los AG sobre ciertas tareas.

3.1.2.9. Elitismo

Es un mecanismo que asegura que los cromosomas miembros altamente adecuados de una población pasaran a la siguiente generación sin ser alterados por cualquier operador genético. El uso del elitismo garantiza que la máxima aptitud de la población nunca disminuye de una generación a la próxima, y generalmente produce una convergencia más rápida de la población.

3.1.2.10. Explotación

Es el proceso de usar información recogida de puntos previamente visitados en el espacio de búsqueda para determinar cual lugar podría ser beneficioso visitar la próxima vez. El ascenso a la colina es un ejemplo de explotación, porque investiga los puntos adyacentes en el espacio de búsqueda, y se mueve en la dirección que da los más grandes incrementos en la aptitud. La técnica de explotación es buena para encontrar mínimos locales (o máximos). Los AG usan el cruzamiento como un mecanismo de explotación.

3.1.2.11. Exploración

Es el proceso de visitar completamente nuevas regiones de un espacio de búsqueda para ver si cualquier cosa prometedora puede ser encontrada allí. Diferentes explotaciones, involucran en la exploración saltos dentro de regiones desconocidas. La búsqueda aleatoria es un ejemplo de exploración. Los problemas que tienen muchos mínimos locales (o máximos) pueden algunas veces solo ser resueltos usando técnicas de exploración como la búsqueda aleatoria. Los AG usan la mutación como un mecanismo de exploración.

3.1.2.12. Genotipo

Representa una solución potencial a un problema y es básicamente el arreglo de valores elegidos por el usuario, también llamado cromosoma.

3.1.2.13. Fenotipo

Es el contenido de un cromosoma particular, definido externamente por el usuario.

3.1.2.14. Nicho

Es un grupo de individuos los cuales tienen aptitudes similares. Normalmente en optimización multiobjetivo y multimodal, una técnica llamada repartición es usada para reducir la aptitud de estos individuos que están en el mismo nicho, para prevenir que la población converja a una sola solución, de esta manera se pueden formar subpoblaciones estables, cada una correspondiendo a una diferente objetivo o pico (en un problema de optimización multimodal) de la función.

3.1.3. Operadores de un Algoritmo Genético

La forma más simple de un AG involucra tres tipos de operadores: selección, cruce y mutación [16].

3.1.3.1. Selección

Este operador selecciona un cromosoma en la población para reproducción. El cromosoma más apropiado, es la mayoría de las veces el más probable a ser elegido para la reproducción [16].

3.1.3.2. Cruce

Este operador selecciona genes desde un cromosoma padre y crea un nuevo hijo. La forma más simple para hacer esto es elegir aleatoriamente algún punto de cruce y todo antes de este punto es copiado desde el primer padre y entonces, todo después de este punto de cruce es copiado desde el segundo padre. Un solo punto de cruce es ilustrado como sigue (| es el punto de cruce) [1].

CromosomaA : 11011|00100110110

CromosomaB : 10011|11000011110

HijoA : 11011|11000011110

HijoB : 10011|00100110110

Como ilustra la Figura 3.2, existen varias técnicas de cruce. En un cruce uniforme, los bits son aleatoriamente copiados desde el primer al segundo padre. Un cruce específico para un problema específico puede mejorar el desempeño del AG.

3.1.3.3. Mutación

Después del operador de cruce, la mutación tiene lugar. La mutación cambia aleatoriamente el nuevo hijo. Para codificación binaria, la mutación es ejecutada cambiando unos pocos bits elegidos aleatoriamente desde 1 a 0 o desde 0 a 1. La mutación depende de la codificación así como también el cruce. Por ejemplo, cuando se codifica permutaciones, la mutación podría ser intercambiar dos genes. Una operación de simple del operador mutación es ilustrada a continuación (- es el bit mutado) [1].

CromosomaA : 1101111000011110

CromosomaB : 1101100100110110

HijoA : 1100111000011110

HijoB : 1101101100110110

La Figura 3.3 muestra la analogía entre la evolución biológica y un GA binario. Ambas comienzan con una población de miembros aleatorios. Cada fila de números binarios representa características seleccionadas de uno de los perros en la población.

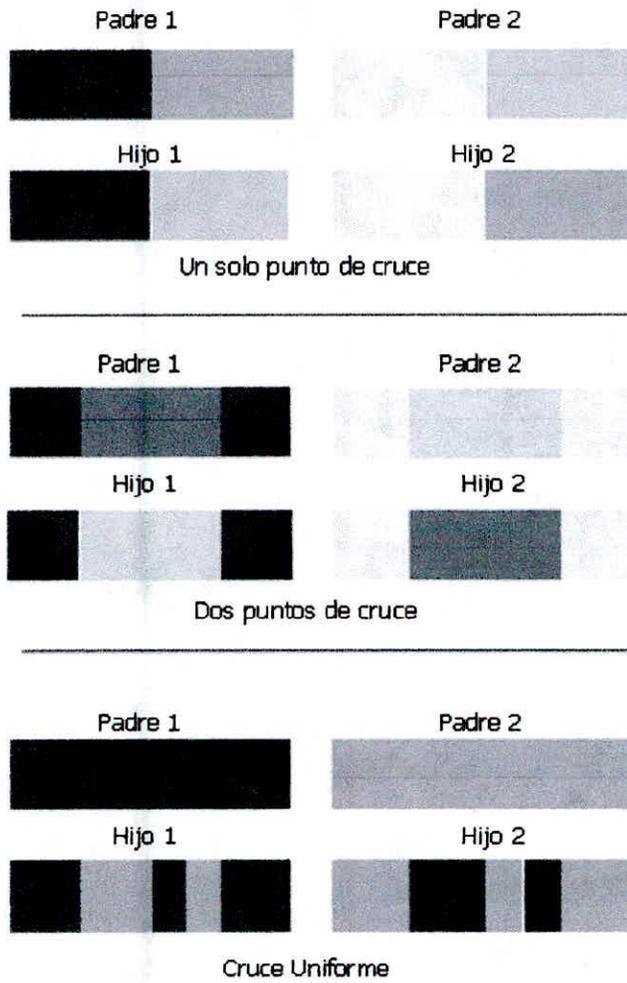


Figura 3.2: Tipos de operadores de cruce.

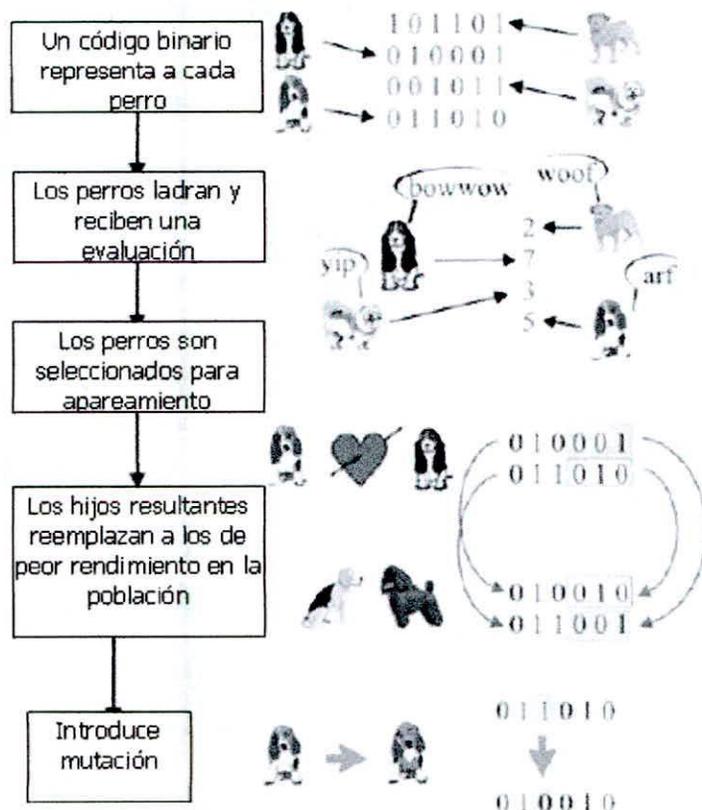


Figura 3.3: Analogía entre la evolución natural y un algoritmo genético.

Rasgos asociados con ladrido fuerte son codificados en la secuencia binaria asociada con estos perros. Si se prueba criar perros con un ladrido más fuerte, entonces solo unos cuantos de los perros con ladridos más fuerte, (en este caso, los cuatro más fuerte) se mantienen para reproducción. Debería existir la forma de determinar el ladrido más fuerte, se les podría tomar una audición mientras el volumen de su ladrido es medido.

Los perros con ladrido fuerte reciben un costo bajo. De esta población de ladrido fuerte, dos perros son seleccionados aleatoriamente para crear dos nuevos cachorros. Los cachorros tienen una alta probabilidad de ser perros con ladrido fuerte por que sus padres tienen genes que los hacen perros con ladrido fuerte. La nueva secuencia binaria de los cachorros contiene porciones de secuencia binaria de ambos parientes de ambos padres.

Estos nuevos cachorros reemplazarán dos perros que son descartados por no tener un ladrido suficientemente fuerte. Suficientes cachorros son generados para que la población vuelva a su tamaño original. Iterar sobre este proceso induce a perros con

ladrido fuerte. Este proceso de optimización natural puede ser aplicado a objetos inanimados igualmente.

3.1.4. Mecanismos de Selección de los Algoritmos Genéticos

A continuación se presentan algunos de los más populares mecanismos de selección [1]:

3.1.4.1. Selección por Ruleta

La más simple de los mecanismos de selección es la selección por ruleta, también llamada muestreo estocástico con reemplazo. Esta técnica es análoga a una rueda de ruleta con cada parte proporcional en tamaño a la aptitud. Los individuos son equiparados a segmentos contiguos de una línea tal que cada segmento individual es igual en tamaño a su aptitud. Un número aleatorio es generado y el individuo cuyo segmento abarque el número aleatorio es seleccionado. El proceso es repetido hasta que el número deseado de individuos es obtenido. Como ilustra la Figura 3.4, el cromosoma 1 tiene la más alta probabilidad de ser elegido puesto que tiene la más alta aptitud.

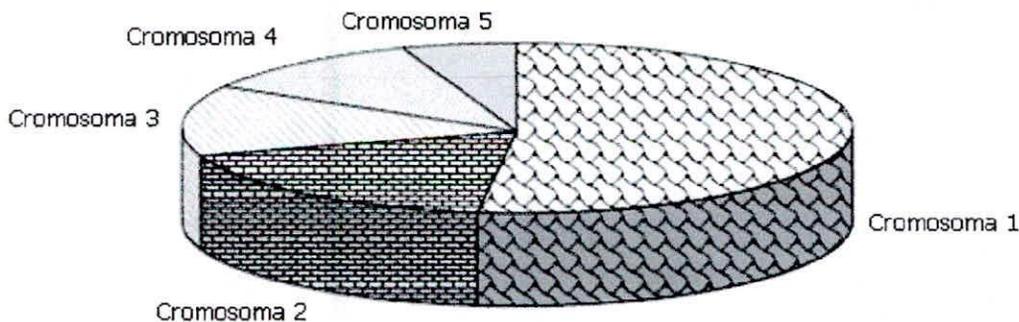


Figura 3.4: Selección por Ruleta.

3.1.4.2. Asignación de Aptitud con base en el Ranking

En este mecanismo de selección la población es ordenada según el valor del objetivo. La aptitud asignada para cada individuo depende de la posición del valor objetivo en el ranking individual. El ranking introduce una escala uniforme a través de la población.

3.1.4.3. Selección por Torneo

este tipo de selección, un número de individuos son elegidos aleatoriamente desde la población y el mejor individuo de este grupo es seleccionado como el padre. Este proceso es repetido hasta que hay suficientes individuos para elegir. El tamaño del torneo es dependerá frecuentemente del problema, tamaño de la población, etc. El parámetro para la selección por torneo es el tamaño del torneo. Los valores del tamaño del torneo van desde dos hasta la totalidad de los individuos de la población.

3.1.5. Criterios Necesarios para Implantar un AG

Después de definir la mayor parte de conceptos y técnicas que utilizan los AG, es necesario definir de modo inequívoco los criterios que se deben tener en cuenta a la hora de implantar un AG [25].

3.1.5.1. Criterio de Codificación

Dado que los AG operan exclusivamente con cadenas binarias (representación) se debe especificar el procedimiento (codificación) con el que se hace corresponder cada punto del dominio del problema con una cadena, o con la terminología anterior, el mecanismo del paso del fenotipo a los genotipos.

3.1.5.2. Criterio de Tratamiento de Individuos no Factibles

No siempre es posible establecer una correspondencia punto por punto entre el dominio de un problema y el conjunto de las cadenas binarias de tamaño usadas para resolverlo, i.e., el espacio de búsqueda. Como consecuencia, no todas las cadenas codifican elementos válidos del espacio de búsqueda y se deben habilitar procedimientos útiles para distinguirlas.

3.1.5.3. Criterio de Inicialización

Se refiere a cómo se debe construir la población inicial con la que se arranca el bucle básico del AG.

3.1.5.4. Criterio de Parada

Se deben concretar las condiciones con las que se considera que el AG ha dado una solución aceptable o, en su defecto, ha fracasado en la búsqueda y no tiene sentido continuar.

3.1.5.5. Funciones de Evaluación y Aptitud

Se debe concretar la función de evaluación más apropiada para el problema, así como la función de aptitud que utilizará el AG para resolverlo.

3.1.5.6. Operadores Genéticos

Todos AG hace uso de al menos tres operadores genéticos; no obstante ellos no son los únicos posibles y además admiten variaciones.

3.1.5.7. Criterios de Selección

La selección debe dirigir el proceso de búsqueda a favor de los miembros más aptos, pero existen muchas maneras de llevar esto a cabo.

3.1.5.8. Criterios de Reemplazo

Los criterios con se seleccionan los criadores no necesariamente han de ser los mismos con que se seleccionan los supervivientes, de ahí la necesidad de especificarlos por separado.

3.1.5.9. Parámetros de Funcionamiento

Un AG necesita que se le proporcionen ciertos parámetros de funcionamiento, tales como el tamaño de la población, las probabilidades de aplicación de los operadores genéticos, la precisión de la codificación, las tolerancias de la convergencia, etc.

Capítulo 4

DEMOCRACIA ELECTRÓNICA

La democracia se desarrolló primero en la antigua Atenas, como resultado de las reformas de Cleístenes en los años 507-508 a.C. Sobre esta forma temprana de gobierno hay muchos conceptos erróneos. Según el estándar democrático actual, la democracia ateniense era, al mismo tiempo, limitada y extensiva. Limitada en el sentido de que estaba restringida a una élite masculina: las mujeres, los esclavos y aquellos sin suficientes propiedades quedaban excluidos; para votar, los individuos tenían que ser ciudadanos de Atenas y debían haber nacido en matrimonio legítimo. A causa de éstas y otras restricciones, se estima que sólo el 15% de la población estaba legitimada para participar en un proceso democrático. Era extensiva en el sentido de que se suponía que todos los ciudadanos cualificados debían participar.

El concepto de democracia directa, donde el ciudadano está continuamente implicado de modo explícito en el proceso de toma de decisiones, resulta muy atractivo para algunos de los postulantes de la democracia electrónica. Sin embargo, las demandas de recursos y logística de la primitiva democracia ateniense, se han considerado excesivas a la hora de aplicarse a gran escala. Se sugiere ahora que la tecnología hace viable la democracia directa a mayor escala y esto ha revitalizado el debate sobre tal democracia, como prueba el creciente interés en muchos lugares del orbe por los, denominados, presupuestos participativos.

Probablemente, la definición más citada y sucinta de lo que es la democracia puede hallarse en el discurso de Lincoln en Gettysburg, cuando la definió como el gobierno del pueblo, por el pueblo y para el pueblo”, estando la clave en la frase “por el pueblo”[34]. El definir qué significa esto y cómo puede realizarse es un problema que ha atraído a muchos filósofos. Sin ahondar demasiado en este debate, ver Sartori [6] para más detalles, las características de la democracia podrían resumirse en:

- Autoridad (kratos) emanada del pueblo (demos).

- Representación
- Consulta.
- Participación.
- Responsabilidad ante el pueblo.

Una distinción crítica, que tiene una importancia clave en el debate sobre la democracia electrónica es, sin embargo, la que se da entre democracia directa y democracia representativa. De hecho, como señala Budge [10] no se trata de una dicotomía, sino de un espectro. Sin embargo, por las razones de dificultades en los transportes y las comunicaciones, la forma dominante de democracia es la del modelo representativo, con enormes variantes, si se comparan, las estructuras democráticas, de, por ejemplo, Estados Unidos, Francia, Japón o España, existiendo variaciones significativas en el equilibrio de poder entre la autoridad local y la central.

La democracia directa parte de supuestos diferentes que incluyen:

- La implicación de la comunidad en la formulación de políticas y legislación.
- La toma de decisiones en comunidad.
- El derecho al debate público.
- La mediación.
- La protección de las minorías.

Los problemas logísticos han sido el principal motivo por el que la democracia representativa es casi universal. En la práctica, la democracia directa, en los últimos tiempos, se ha limitado a algunos comicios o votaciones populares. Hoy en día se afirma que la democracia electrónica puede cambiar esta situación y contribuir a que la democracia sea más efectiva.

4.1. Opciones Tecnológicas para la Democracia Electrónica

Antes de analizar el tema específico de la democracia electrónica en que se desea aplicar el prototipo desarrollado, revisaremos diversas cuestiones de carácter tecnológico..

4.1.1. Herramientas Tecnológicas

Las actuales (TIC) proporcionan numerosas herramientas con el potencial de enriquecer la democracia, entre las que se incluyen, no sólo tecnologías relacionadas con Internet, sino también relacionadas con otros campos, como la Inteligencia Artificial o de seguridad informática. Algunas de las aplicaciones, potenciales o actuales, de apoyo a la democracia serían:

- La votación electrónica.
- El voto en línea.
- Los comicios en línea.
- El registro en línea de votantes.
- La difusión de reuniones de representantes.
- La disponibilidad de bases de datos online.
- La publicación de información de todo tipo en el web.
- Foros de discusión política con funcionarios y políticos.
- El lobby electrónico.
- Las campañas electrónicas.
- Las iniciativas en línea de ciudadanos.
- La encuesta electrónica.
- Las consultas en línea.
- La toma de decisiones en línea.
- La legislación directa por los ciudadanos.

Algunas de estas posibilidades son más ambiciosas que otras. Algunas de ellas se refieren van en la línea de hacer más eficientes los procedimientos tradicionales.

Podemos de hecho, clasificarlas, según la siguiente distinción tomada de Scott [15]:

- Tecnologías que automatizan o facilitan los procesos. Esto incluye el voto electrónico, y ciertas formas de voto en línea. Tales tecnologías proporcionan una versión más efectiva y eficiente que la práctica existente.
- Tecnologías que suministran una mejor información, que incluye desde la publicación de minutas o la transmisión de reuniones por videoconferencia, hasta paseos virtuales por proyectos de planes de desarrollo.
- Tecnologías que transforman, de algún modo, el proceso democrático. Tales tecnologías incluyen consultas en línea, toma de decisiones en línea y discusiones públicas generales sobre las políticas legislativas.
- Tecnologías que cambian la escala. Un buen ejemplo serían los referendos en línea. Muchos países realizan referendos ocasionalmente. La capacidad de los comicios en línea para reducir los costes de los comicios habituales podrían tener un impacto significativo en los procesos democráticos.

Es una opinión generalizada que las tecnologías que automatizan e informan son importantes y útiles, sin embargo, son de menor interés que aquéllas con un potencial para cambiar la naturaleza de la democracia.

4.1.2. Beneficios Potenciales

Los beneficios potenciales de la naciente democracia electrónica han sido ampliamente pregonados. Entre ellos se destacan:

- Legitimación.
- Acercar las decisiones a la gente.
- Decisiones públicas, tomadas públicamente.
- Eliminación de la apatía.
- Maximización del potencial del ciudadano.

También se pueden identificar algunos peligros potenciales como la incapacidad de los ciudadanos de hacer juicios correctos en temas complejos, el debilitamiento de la autoridad central poniendo en peligro a las minorías, etc. Otros críticos avisan sobre los riesgos que la cultura de “hazlo rápidamente” de Internet puede imponer sobre el valor de la deliberación tranquila en el proceso de toma de decisiones.

4.2. Presupuestos Participativos

Un campo potencialmente fructífero de aplicación de las ideas anteriores es el de los presupuestos participativos, que son un intento de dar la palabra y la decisión a los ciudadanos en las cuentas públicas haciéndolos partícipes de la elaboración y aprobación de parte del presupuesto. Esto haría de la democracia mucho más que la elección de representantes cada cuatro años. Se trata de un modelo de gestión presupuestario basado en el diálogo y la participación ciudadana, distinto al modelo representativo actual. Así, los ciudadanos participan en decisiones cómo en qué se prefiere gastar el dinero disponible, tomando decisiones en cuestiones como el transporte, la salud pública y la asistencia social, la educación, la cultura, el ocio y entretenimiento, el desarrollo económico, los impuestos y el desarrollo urbanístico.

Hay que tener en cuenta que este problema podría modelizarse matemáticamente como un problema de asignación de recursos limitados que maximicen la utilidad (multiobjetivo) de los ciudadanos, pero requiere una elaborada solución que, en particular, conlleva conocer las preferencias de grupo de los ciudadanos, para lo cual las tecnologías de la información tendrían mucho que aportar.

Para que los ciudadanos informados participen en la toma de decisiones presupuestarias, se podrían usar sistemas informáticos como el descrito que ayuden a modelizar sus preferencias a través de asambleas virtuales en donde los ciudadanos expusiesen sus preferencias sobre cómo gastar el presupuesto y se consensuasen preferencias de grupo ayudados por tales sistemas, que ayudarían también a escoger el presupuesto que mejor se ajusta a las preferencias consensuadas por todos, pues esta tarea sería muy difícil de lograr por las partes implicadas a la hora de decidir el mejor presupuesto, ya que sus preferencias no tienen por qué coincidir con las del grupo.

Los presupuestos participativos para determinar el gasto público hasta ahora no aplican estas técnicas que aportan racionalidad en las preferencias de grupo. De hecho, cuando se habla de “Presupuestos Participativos” se está entendiendo un proceso en que se elaboran los presupuestos mediante la participación ciudadana que se asienta en sucesivas tandas de Asambleas, que van determinando las prioridades, de donde salen delegados que consensúan una propuesta de gasto para toda la ciudad. Quizá en un futuro, el debate mediante presupuestos participativos incluirá, también, los capítulos de ingresos, de forma que se cierre una propuesta completa de presupuesto.

Sin sistemas de ayuda a la decisión en grupos y basándose en métodos aún intuitivos

existen experiencias en presupuestos participativos. Una ciudad pionera es Porto Alegre, en donde, desde 1989, sus ciudadanos han comenzado a decidir el destino de sus fondos públicos. En cada una de las 16 asambleas ciudadanas iniciales se discute y se fijan las prioridades presupuestarias y sus delegados las comunican posteriormente a las autoridades municipales. Estos delegados elegidos por los ciudadanos para velar por el cumplimiento de sus prioridades presupuestarias tienen voz y voto en la elaboración y aprobación del presupuesto de Porto Alegre.

La ONU ha calificado como 'buena práctica de gestión urbana' la articulación de 'Presupuestos participativos' en el ámbito local y ha recomendado su aplicación en la Asamblea de Ciudades y Autoridades Locales celebrada en Estambul en 1996. Este instrumento ha demostrado su efectividad en más de cien municipios brasileños y en grandes ciudades como Porto Alegre o Sao Paulo. Así mismo, la Federación Española de Municipios participará en el Programa URB-AL de la UE para el estudio y seguimiento de los 'Presupuestos participativos' en Europa y América Latina durante los próximos dos años.

Capítulo 5

NEGOCIACIÓN AUTOMATIZADA

Si consideramos el término negociación en un sentido amplio, podemos decir que pasamos la vida negociando; es decir, llegando a acuerdos con otras personas para solucionar los conflictos que van surgiendo en la convivencia. Puede aplicarse en ámbitos muy diversos, desde los conflictos surgidos entre dos o más personas a nivel de amistades, compañeros, familia, etc.; hasta los conflictos entre grupos más amplios (e.g. sindicatos y empresarios), e incluso a los conflictos entre comunidades o naciones.

La negociación se basa en la idea de que aunque las partes implicadas difieran en sus intereses, comparten un interés común por encontrar una solución negociada. Suele usarse cuando los objetivos o intereses de las personas en conflicto son mutuamente excluyentes, por lo que no se pueden hallar soluciones en las que todos ganen, sino que hay que negociar; es decir, ceder por ambas partes, para llegar a algún tipo de acuerdo [5].

La negociación a sido reconocida por mucho tiempo como un tópico central en SMA [23, 55]. Mucho de este interés ha surgido a través de la posibilidad de automatizar escenarios de negociación, en los cuales agentes inteligentes negocien por un bien o servicio en nombre de un usuario final [48].

5.1. Negociación

Muchas definiciones de negociación han sido ofrecidas, por diferentes investigadores, dependiendo de los propósitos de su investigación [55, 26]. Por ejemplo:

- Gulliver [9] define negociación como un tipo de proceso de resolución de problemas, en el cual las personas intentan encontrar una decisión conjunta sobre temas de interés común en situaciones donde ellos están en desacuerdo y en conflicto.
- Rosenschein y Zlotkin [55] definen negociación como una forma de proceso de toma de decisión donde dos o más partes conjuntamente buscan en un espacio de posible solución con el objetivo de encontrar un consenso (trato).

La principal diferencia entre estas definiciones es que el antiguo énfasis se centró en resolver conflictos y desacuerdos, mientras el posterior pensamiento sobre negociación es que puede ser una metodología para mejorar la cooperación. No necesariamente existe conflicto entre los participantes de la negociación [30].

5.2. Negociación Automatizada

Es el proceso por medio del cual se da soporte a una parte central en la teoría de SMA; la interacción. Esta ocurre por la inter-dependencia que inevitablemente existe entre los agentes y se manifiesta en diversas formas, incluyendo la cooperación y la coordinación. La negociación es quizás el mecanismo más fundamental y poderoso para manejar las interacciones en tiempo de ejecución. La negociación intenta apuntar a cooperar y coordinar (tanto en agentes humanos como artificiales) y es requerida cuando los agentes son auto interesados así como cuando son cooperativos.

Para que un agente influencie a un conocido, el conocido necesita ser convencido de que debería actuar de una forma particular. El medio para lograr este estado es hacer una propuesta, una opción de canje, ofrecer concesiones y (esperanzadoramente) llegar a un mutuo acuerdo. En resumen, negociar [47].

5.3. Componentes de la Negociación Automatizada

La Teoría de Negociación cubre un amplio rango de fenómenos y abarca múltiples enfoques (desde IA, psicología social y teoría de juegos) a pesar de esta variedad, los investigadores en negociación automatizada consideran tres grandes tópicos:

5.3.1. Protocolos de Negociación

El conjunto de reglas que gobiernan la interacción. Esto cubre el permisible tipo de participantes (los negociadores y algunas relevantes terceras partes), el estado de la negociación (aceptando propuestas, negociación clausurada) los eventos que causan cambios en los estados de negociación (no mas oferentes, propuestas aceptadas) y la validación de acciones de los participantes en estados particulares (en el cual los mensajes pueden ser enviados por cualquiera, a cualquiera, en cualquier etapa) [47].

5.3.2. Objetos de Negociación

El rango de asuntos sobre el cual el acuerdo debe ser cumplido, en caso extremo, el objeto puede contener un simple asunto (semejante como un precio), mientras en otro caso puede cubrir centenares de asuntos (relacionado a precio, calidad, tiempos, sanciones, términos y condiciones, etc). Ortogonal a la estructura del acuerdo, y determinado por el protocolo de negociación, es el asunto del tipo de operación que puede ser ejecutado en este. En el caso simple, la estructura y el contenido del acuerdo son inalterados y los participantes cada uno puede aceptar o rechazar esto (tomar o dejar esta oferta). En el próximo nivel, los participantes tienen la flexibilidad de cambiar los valores de los asuntos del objeto de negociación (ellos pueden hacer un contador de propuestas para asegurar el acuerdo mas apropiado a los objetivo de su negociación). Finalmente a los participantes se les puede permitir alterar dinámicamente (agregando o removiendo asuntos) la estructura del objeto de negociación (un vendedor de carros puede ofrecer por un año seguro gratis para rematar el negocio) [47].

5.3.3. Modelo de Toma de Decisiones de los Agentes

El mecanismo de toma de decisiones de los participantes empleados actúa en línea con el protocolo de negociación en orden para lograr sus objetivos. La complejidad del modelo, así como también el rango de decisiones que tiene que tomar, son influenciados por el protocolo en su lugar, la naturaleza del objeto de negociación y el rango de operaciones que pueden ser ejecutadas sobre este [47].

5.4. Propiedades de un Mecanismo de Negociación

Existen algunas propiedades que son generalmente consideradas por un mecanismo de negociación automatizado. Las más importantes son tratadas a continuación.

5.4.1. Eficiencia

El sistema asegura que los agentes participantes alcanzan un acuerdo en el tiempo más corto, dado que exista alguna ruta [37].

5.4.2. Efectividad

Si existe un acuerdo entre los agentes entonces debería estar basado sobre un equilibrio estable al finalizar la negociación. Esto asegurará que los agentes no se desviarán de su acuerdo [37].

5.4.3. Función de Utilidad Privada

La función de utilidad es usada para determinar que tan buena es una oferta o una contra oferta. Esta provee información esencial para que los agentes decidan si una oferta debería ser aceptada o no. Esto es importante para mantener esta función privada para el agente [37].

5.4.4. Múltiples Asuntos

Una negociación automatizada debería permitir a los agentes expresar los asuntos que ellos quieren negociar. Múltiples asuntos generalmente forma un largo y complejo espacio de búsqueda. Por tal razón un algoritmo de búsqueda efectivo es requerido para explorar el espacio de búsqueda [37].

5.4.5. Simetría

Normalmente en escenario de negociación automatizada, ningún agente tiene completo control del juego. Por consiguiente se prefieren mecanismos que garanticen simetría en términos del poder de los agentes [43].

5.5. Reseña de la Evolución de la Negociación Automatizada

Como la ineficiencia ha sido ampliamente reconocida como un problema común en negociación, muchos proyectos de investigación han sido conducidos para mejorar la eficiencia en negociación usando tecnología computacional.

La tecnología de SMA da una oportunidad para superar el problema de ineficiencia en negociación. Los agentes en SMA actúan colectivamente como una sociedad y ellos colaboran (o cooperan) para lograr sus propios objetivos individuales así como también los objetivos globales de la sociedad a la cual ellos pertenecen.

Además, teniendo en cuenta la definición sobre negociación de Rosenschein y Zlotkin, en la comunidad de IAD se gestaron algunos proyectos de investigación en los cuales, como anota Peyman [19] el principal asunto en la investigación de SMA es como modelar las interacciones entre los agentes. El modelo de negociación ha surgido como un candidato apropiado para resolver este problema de interacción debido a su naturaleza descentralizada, énfasis sobre selección mutua de una acción y la prevalencia de la negociación en sistemas sociales reales.

Incluso muchas teorías importantes y principios han sido desarrollados para explicar diferentes aspectos de negociación, tales como la metodología organizacional, la metodología estructural, la metodología de la teoría de juegos, la metodología de la teoría económica, la metodología de la teoría del comportamiento y la metodología de experimentación y simulación.

En ese sentido, una perspectiva útil en negociación es ver esta como una búsqueda distribuida en un espacio de potenciales acuerdos [47]. Y teniendo en cuenta como apunta Ficici [8] que inspirados por el principio de selección natural, los algoritmos co-evolutivos son métodos de búsqueda en los cuales el proceso de adaptación mutua ocurre entre los agentes que interactúan estratégicamente. Los resultados de la interacción revelan una estructura de recompensas que guían la evolución hacia el descubrimiento de comportamientos cada vez más adaptativos. De esa manera, los algoritmos co-evolutivos son frecuentemente usados para buscar el comportamiento óptimo de un agente en dominios de interacción estratégica.

De igual forma, cabe destacar como señala Robert Axelrod [21] que en entornos complejos, los individuos no son completamente capaces de analizar la situación y

calcular su estrategia optima; ellos pueden haber esperado adaptar su estrategia a través del tiempo basados en que ha sido efectivo y que no. Una útil analogía para el proceso de adaptación es la evolución biológica. En evolución, las estrategias que han sido relativamente eficaces en una población llegan a ser más difundidas y las estrategias que han sido menos efectivas llegan a ser menos comunes en la población.

La evolución biológica ha sido altamente eficiente en el descubrimiento de métodos efectivos y complejos de adaptación para muchas situaciones ambientales. Esto es logrado por reproducción diferencial de los individuos más exitosos. El proceso evolutivo también requiere que las características exitosas sean heredadas a través de un mecanismo genético que permite en algunas ocasiones que nuevas estrategias sean descubiertas [21].

Por otro lado la negociación y la discusión de precios fueron estudiadas en la literatura de la teoría de juego antes de que los sistemas multi-agentes surgieran como una disciplina de investigación e incluso antes de la llegada de las primeras computadoras digitales.

Determinando un particular escenario de negociación que involucre agentes automatizados, las técnicas de teoría de juegos pueden ser aplicadas a dos problemas claves:

1. El diseño de un apropiado protocolo que gobernará las interacciones entre los participantes de la negociación [40].
2. El diseño de una estrategia particular (los modelos de toma de decisión de los agentes) que los agentes individualmente pueden usar mientras negocian, un agente pretenderá usar una estrategia que maximice su propia ganancia [40].

Incluso trabajos anteriores mostraron que los agentes racionales son capaces de coordinar (ordenar actividades relacionadas para ser ejecutadas en una forma coherente) y cooperar con un mecanismo de negociación de juegos teóricos, aun sin comunicación [21, 36]. Sin embargo, hay tres diferentes fallas en la toma de decisiones de juegos teóricos sin comunicación. La primera es el problema del dilema del prisionero (Sección 2.7), los agentes entrarán en un resultado deficiente bajo la suposición de racionalidad. El segundo, una clase de juegos que no brindan ningún punto de equilibrio causarán un ciclo de razonamiento sin limite [42]. El tercero, cuando hay más de un equilibrio, es difícil seleccionar el mejor y permanecer equilibrado.

Para escapar del problema del dilema del prisionero y de aquellos juegos que no brindan puntos de equilibrio, se introdujo la capacidad de pedir garantía como el

primer tipo de acción de comunicación dentro de la convencional teoría de juegos [50], pero los juegos sin equilibrio permanecían sin resolver. Para lo cual se introdujo un segundo tipo de acción de comunicación, ofreciendo compensación, dentro de la teoría de juegos. Esta acción de comunicación necesita una tercera parte confiable en el juego para coordinar los dos agentes involucrados. La tercera parte confiable juega el rol de un banquero o un mediador [51].

A pesar de los avances en negociación basados en teoría de juegos, todavía existía un problema y es que la información completa de la retribución no puede estar disponible durante la negociación en la mayoría de los dominios de aplicación práctica, es decir, los agentes solo pueden conocer su propia retribución y no pueden conocer la retribución de su contraparte.

En el trabajo de Soo y Wu [54] se extiende el mecanismo de negociación para tratar con juegos de información incompleta. Durante el proceso de negociación, cada agente hace sugerencias sobre como se puede encontrar un estado de equilibrio mientras maximiza su propia retribución. Ya que la información es incompleta, se le llama al equilibrio alcanzado equilibrio NFD (No Fear of Deviation, sin temor de desviación) [54].

Muchos investigadores en su búsqueda por una solución más óptima dentro de los mecanismos de negociación automatizada han optado exclusivamente cada uno por una metodología co-evolutiva tal como los algoritmos genéticos (AGs) o por la teoría de juegos. Mientras cada metodología tiene sus propios méritos ambas sufren de serias limitaciones.

En la metodología de los AGs las soluciones óptimas globales son difíciles de encontrar. En la teoría de juegos la complejidad del espacio de búsqueda para muchos dominios de aplicación práctica atribuye la generación de la matriz de pagos durante la negociación. Por tal razón, Chen y Chao [33] han optado por una combinación de la teoría de juegos y una metodología co-evolutiva para el desarrollo de mecanismo de negociación automatizada.

Este esquema involucra un proceso que itera sobre la generación de un conjunto de estrategias por medio de la metodología co-evolutiva, la codificación de estas estrategias dentro de una matriz de pagos y el razonamiento sobre la matriz por la teoría de juegos para encontrar un punto óptimo. El proceso termina cuando la teoría de juegos encuentra un punto óptimo que satisface a ambos agentes. La principal ventaja de este sistema es que los agentes, sin el conocimiento de las estrategias de

los otros agentes, acuerdan una solución optimizada que está conforme al equilibrio de Nash y la eficiencia de Pareto [33].

Debido a que el desarrollo de un proyecto que soporte Democracia electrónica demanda mucha tecnología, tiempo y está compuesto por varios módulos, el desarrollo se limita a trabajar sobre el módulo de negociación, el cual se encarga de guiar la obtención de un acuerdo entre los participantes del proceso democrático.

El trabajo persigue aplicar una nueva metodología para proveer mecanismos que faciliten la creación de equipos y ayudar a resolver conflictos a través de negociación automatizada, para tal fin se a desarrollado un sistema prototipo que automatiza el proceso de negociación partiendo del trabajo que realizó la simbiosis entre la teoría de juegos y los algoritmos genéticos, agregándole la capacidad de que sea un sistema independiente del objeto de negociación, ya que soportará el logro de un acuerdo entre los participantes de un proceso de democracia electrónica sin un asunto de negociación definido (Ej.: elaboración de presupuestos, políticas para prevención de desastres, proyectos a realizar, etc).

Capítulo 6

DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO

El objetivo principal del prototipo implementado es lograr un acuerdo acorde al equilibrio de Nash y la eficiencia de Pareto. Para tal fin se ha desarrollado un esquema de negociación automatizada que soporta en presupuestos participativos (un área dentro de la Democracia Electrónica) la asignación de recursos a proyectos mediante la interacción entre agentes haciendo uso de la computación evolutiva y la teoría de juegos.

Para desarrollar el diseño del prototipo del SMA se a utilizado la metodología de INGENIAS [7], así mismo, todo el desarrollo se a realizado en JAVA y el API de construcción de SMA propuesto por JADE.

6.1. Arquitectura de la Aplicación

El prototipo del sistema de negociación automatizada realizado en este trabajo de investigación se compone de: una interfaz gráfica, una BD con información referente al proceso de negociación, dos agentes negociadores, un AG seleccionador de valores de los conjuntos de asuntos, un AG seleccionador de las estrategias de negociación, un algoritmo NFD y un agente TTP. En la figura 4 se puede observar una representación de la arquitectura del sistema.

La siguiente es una descripción concisa de la función que cumple cada parte del sistema:

- Interfaz gráfica: permite al usuario interactuar con el prototipo desarrollado, suministrándole información relacionada con el proceso de negociación que se

pretende realizar. Así mismo, facilita posteriormente a la autenticación en el sistema, la personalización del agente que lo representará en el proceso de negociación.

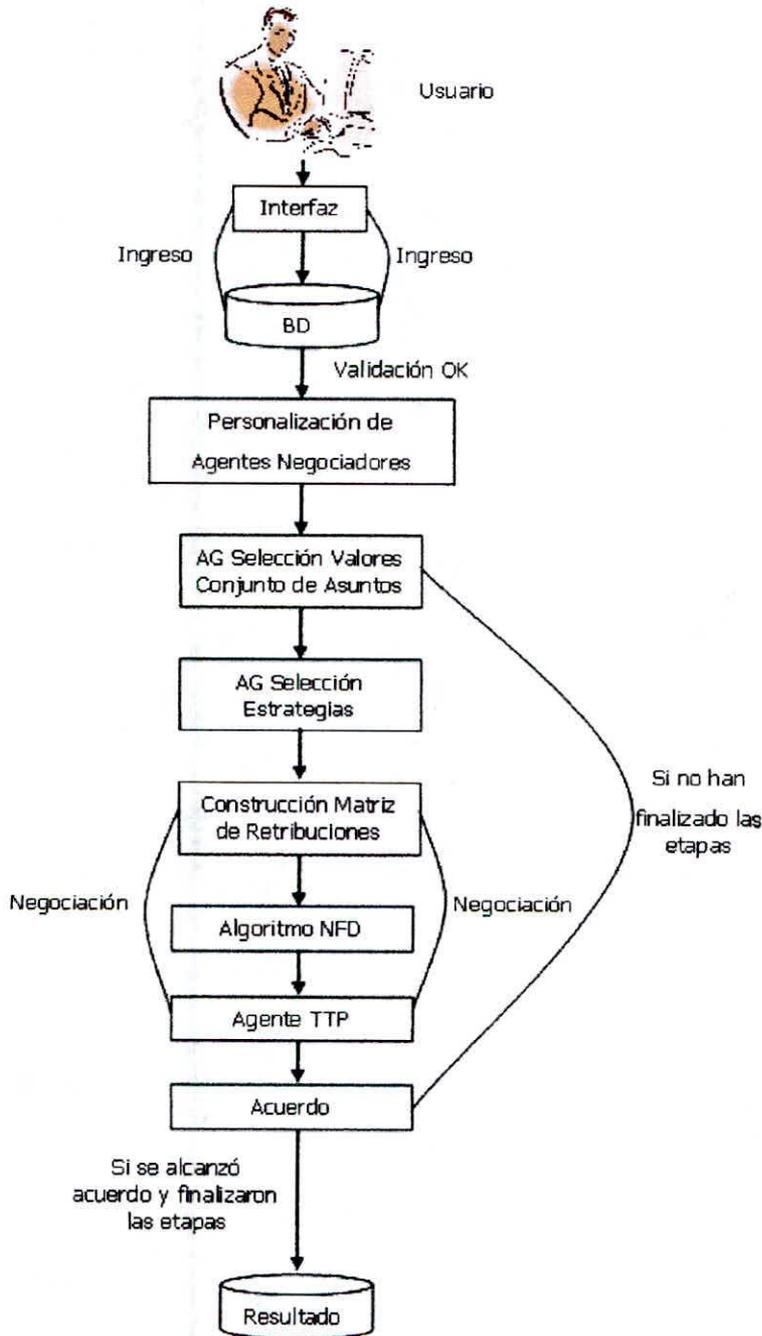


Figura 6.1: Arquitectura del prototipo desarrollado.

- Base de datos de información: su función es almacenar la información relativa a los usuarios, por ejemplo: nombre, identificación, máxima y mínima utilidad

esperada, etc.

- Agentes negociadores: son los encargados de representar al usuario en el proceso de negociación y su objetivo principal es maximizar la utilidad que espera alcanzar el usuario.
- AG Seleccionador de Valores de los Conjuntos de Asuntos: la función de utilidad que ingresa el usuario al prototipo está compuesta por una serie de variables que representan asuntos bien definidos en el proceso de negociación; la tarea de este AG consiste en maximizar cada uno de estos asuntos haciendo uso del protocolo de comunicación del sistema y estableciendo grados de satisfacción para cada asunto.
- AG Seleccionador Estrategias Negociación: la función de este AG es generar estrategias con base a los valores de los conjuntos de asuntos facilitados por el AG Seleccionador de Valores de los Conjuntos de Asuntos, es decir, optimiza las estrategias para que la matriz de retribuciones que se construya permita alcanzar un acuerdo más rápido puesto que se a generado con base a grados de satisfacción.
- Algoritmo NFD: este algoritmo se encarga de guiar el proceso de negociación haciendo uso del agente TTP y de las acciones de comunicación de garantía y compensación, su objetivo es lograr un acuerdo acorde al equilibrio de Nash y la eficiencia de Pareto.
- Agente TTP: la labor de este agente consiste en generar confianza en el proceso de negociación, debido fundamentalmente a que garantiza que todos los acuerdos que se hagan durante el proceso se cumplan.

6.1.1. Ambiente del prototipo de agentes

El ambiente en el que se encuentran los agentes está determinado por la matriz de retribución, los agentes negociadores y el agente TTP. Este reúne las siguientes características:

- Es no accesible ya que los agentes negociadores tienen acceso solamente a las retribuciones que le competen, es decir, desconocen la matriz de retribución de la otra parte negociadora.

- Es no determinista debido a que no se garantiza que el estado final del ambiente sea de acuerdo gracias a las acciones realizadas por los agentes, es decir, podríamos tener un estado final del ambiente en acuerdo o no acuerdo.
- Es no episódico ya que no existe un número discreto de episodios sobre los cuales los agentes deban percibir y actuar.
- Es estático ya que el ambiente solo sufre cambios debido a los efectos directos de los agentes.
- Es discreto ya que el número de percepciones y acciones (comportamientos) de cada agente esta fijado desde la creación del sistema de agentes.

6.2. Características del Sistema MultiAgente

En las siguientes secciones se explicará como es control, el conocimiento y la comunicación del SMA.

6.2.1. Control

El control en el SMA desarrollado es descentralizado. Cada agente está en capacidad de tomar sus propias decisiones libremente con base en las percepciones y gracias a la coordinación subyacente se asegura que la decisión individual del agente resulta en una buena decisión conjunta para el grupo.

6.2.2. Conocimiento

El conocimiento en el SMA desarrollado es parcial para los agentes negociadores, debido a que a estos solamente tienen acceso a su matriz de retribución, en tanto, que el agente TTP tiene un conocimiento total del ambiente. El modelo está planteado así para reflejar los procesos de negociaciones del mundo real en donde las partes involucradas en proceso de negociación desconocen lo que espera ganar la contraparte en el proceso.

6.2.3. Comunicación

Los agentes negociadores hacen uso de los mensajes para realizar ofertas y contraofertas, las cuales son enviadas al agente TTP que es el que mediador en el

proceso de negociación. Los mensajes como tal contienen propuestas que conducen el proceso de negociación a un acuerdo, en caso de que este exista.

6.3. Explicación Lógica del Prototipo

El sistema de negociación automatizada está compuesto por dos clases de agentes: Agente Negociador y Agente TTP. El Agente Negociador, es el que representa al usuario en el proceso de negociación automatizada del sistema, además de negociar, su objetivo es lograr un acuerdo con el mayor grado de utilidad posible. El Agente TTP, es el encargado de conducir el proceso de negociación automatizada y en ese sentido hace llegar las propuestas entre los agentes negociadores, ejecuta las acciones de comunicación (garantía y/o compensación) y hace cumplir los compromisos pactados.

El protocolo de negociación y la estrategia son dos importantes constituyentes en el mecanismo utilizado, los cuales suministran a los agentes negociadores las reglas de interacciones y directivas para la comunicación durante la negociación. Ellos también determinan cuando el proceso de negociación termina.

Básicamente el prototipo de negociación propuesto funciona así: El administrador del sistema ingresará información relevante para la ejecución del software, por ejemplo: las etapas de negociación que se llevarán a cabo, el número de iteraciones que se ejecutarán en cada etapa, etc. Posteriormente cada uno de los Usuarios del Sistema al interactuar con el software, personaliza al agente que lo representará en la negociación. Cada agente negociador en el sistema incorpora tres componentes principales: dos Algoritmos Genéticos y un Algoritmo de Equilibrio NFD.

El empleo de los algoritmos genéticos es para explorar el espacio de posibles acuerdos y el uso del algoritmo NFD es para distribuir la retribución y encontrar un punto óptimo. Los tres componentes intercambian información internamente por medio de una matriz de retribuciones, generada por la metodología co-evolutiva. Cada agente negociador implementa un ciclo sobre el cual genera y selecciona estrategias, codifica la matriz de retribuciones y determina un punto optimizado. El punto óptimo puede ser usado como un punto de referencia para que los agentes negociadores exploren más el espacio de posibles acuerdos. Además, los agentes negociadores intercambian información externamente por medio de un protocolo. A continuación, se describirá en detalle como funciona el prototipo desarrollado.

Se asume que dos agentes negociadores (denotados como Ag_1 y Ag_2) tienen su propia función de utilidad, $Ag_i u()$ y $i=1,2$. Estas son usadas para evaluar la oferta propuesta por su contraparte y son usadas como la base para una contra oferta.

La negociación comienza cuando uno de los agentes, por ejemplo Ag_1 , hace la primera oferta (δ). La oferta incluye un número de asuntos, $\delta^{(1)} = \{i_1^{(1)}, i_2^{(1)}, \dots, i_k^{(1)}\}$. Los dos agentes tienen el mismo conjunto de asuntos sobre los cuales ellos negocian. El agente Ag_2 evalúa la oferta con su función de utilidad $u_2(\delta^1)$ para obtener la utilidad $Ag_2 u$.

Por lo tanto: $u_i : Ag_i u(\delta) \rightarrow Ag_i \mu$ e $i = 1, 2$. Hay que tener en cuenta, que en el mundo real los agentes tienen diferentes funciones de utilidad (i.e. $Ag_1 u(\delta) \neq Ag_2 u(\delta)$).

Una vez el agente Ag_2 recibe la oferta, la evalúa, hace una contraoferta y responde con un grado de satisfacción con respecto a la oferta para el Ag_1 . El grado de satisfacción es calculado como:

$$Ag_i Score(\delta) = (Ag_i u(\delta) - Ag_i \mu_{min}) / (Ag_i \mu_{max} - Ag_i \mu_{min}) \quad (6.1)$$

donde $Ag_i \mu_{min}$ es la utilidad más baja que se espera ganar (como umbral) y $Ag_i \mu_{max}$ es la utilidad más alta esperada, es en otras palabras, la utilidad ideal para Ag_i , donde $i = \{1, 2\}$ y cada agente participante debería tener estos dos valores; note que esta formula asume que:

$$Ag_i \mu_{min} < Ag_i u(\delta) < Ag_i \mu_{max} \quad (6.2)$$

El grado de satisfacción suministra información esencial para la evaluación de las ofertas y contraofertas y en resumen acorta el proceso de negociación.

Cada agente también tiene un conjunto de estrategias de negociación para dirigir como los agentes hacen ofertas y contraofertas. La tarea de la estrategia de negociación (denotada como S) es generar una serie de ofertas, δ^i considerando el grado de satisfacción del oponente, la utilidad objetivo y el número de restricciones de ofertas alternas.

Una estrategia de negociación S deriva desde una función (denotada como $\zeta()$) que varia la utilidad objetivo que el agente intenta obtener sobre la secuencia de ofertas.

La función $\zeta()$, adoptada de [49, 19], es definida como sigue:

$$\zeta(t, \beta, k^{(s)}) = k^{(i)} + (1 - k^{(i)})(t/t_{max})^{(1/\beta)}, \quad (6.3)$$

donde t es un entero $1 \leq t \leq t_{max}$.

y

$$Ag_i \mu_t = Ag_i \mu_{min} + \zeta(t, \beta, k^{(i)})(Ag_i \mu_{max} - Ag_i \mu_{min}) \quad (6.4)$$

Los dos valores de $k^{(i)}$ son el conjunto $k^{(s)}$ el cual es un valor arbitrario, $0 \leq k^{(s)} \leq 1$, representando el inicio de la mínima utilidad esperada.

La función $\zeta(t, \beta, k^{(s)})$ depende de los valores de β que son capaces de generar diferentes estrategias y t es un entero y $1 \leq t \leq t_{max}$, de esta manera una estrategia de negociación específica S es derivada de: $\zeta(t, \beta, k^{(s)}) \rightarrow S$. La estrategia de negociación S contiene una serie de utilidades denotada como μ las cuales contienen un conjunto ordenado de utilidades con un rango $(\mu_1 \pm \varepsilon, \mu_2 \pm \varepsilon, \dots, \mu_t \pm \varepsilon, \dots, \mu_{t_{max}} \pm \varepsilon)$ que el agente espera obtener sobre t_{max} veces y ε es una constante que representa un nivel de tolerancia.

El agente solo negocia sobre los asuntos I y μ no es revelada a su contraparte. El valor actual de μ es usado como la utilidad objetivo de una búsqueda para encontrar una posible combinación de valores de asuntos los cuales produzcan μ . Se denota esta función de búsqueda como $GAs(I, \mu)$, I son los asuntos de negociación y μ es la utilidad objetivo.

Los vectores, $\delta^{(i)}_j$, son conjuntos de valores para los asuntos I que produce el valor de utilidad μ_i mediante $GAs(I, \mu_i)$ donde $\delta^{(i)}_j = \{(\delta^{(i)}_j)_1, (\delta^{(i)}_j)_2, (\delta^{(i)}_j)_3, \dots, (\delta^{(i)}_j)_n\}$, e $I \neq \{\}$. Los vectores $\delta^{(i)}_j$ satisfacen la condición $|Ag_k u(\delta^{(i)}_j) - \mu_i| \leq \varepsilon$ (donde k puede ser 1 o 2).

El paso inicial para la función $GAs()$ es buscar la posible combinación de valores de asuntos para formar la población inicial que podría producir el objetivo μ . Dado que todos los $\delta^{(i)}_j$ tienen un desempeño similar, solo uno de estos $j = j_0$ debería ser seleccionado para lograr $|Ag_k u(\delta^{(i)}_j) - \mu_i| \leq \varepsilon$. Esto es usado para hacer una propuesta en un paso posterior para que el algoritmo que se basa en la teoría de juegos razone con ella. A continuación se describe el procedimiento de selección.

Se toma un conjunto de elementos $\delta^{(i)}_j (j = 1, 2, \dots, m_i)$ y se denotan como $\Delta^{(i)}$. Estos serán ofrecidos secuencialmente a los otros agentes para obtener la realimentación

desde su contraparte, el grado de satisfacción para $\delta^{(i)}_j$. La definición de grado de satisfacción es un medio para medir que tan bien el $\delta^{(i)}_j$ representa a su contraparte. Una vez todos los $\delta^{(i)}_j$ han obtenido el grado de satisfacción en la población, la función $AG(I, \mu_i)$ es utilizada para producir un único nuevo hijo $\delta^{(i)}$, los dos $\delta^{(i)}_j$ en la población inicial con el más alto grado de satisfacción son elegidos como padres. Un nuevo hijo viable $\delta^{(i)}_j$ debe satisfacer la condición $\mu_i - \varepsilon \leq Ag_k u(\delta^{(i)}_j) \leq \mu_i + \varepsilon$ y $\delta^{(i)}_j \notin \Delta^{(i)}$. Si la condición no es cierta, el hijo generado es abandonado y los padres intentarán generar un nuevo hijo hasta que el estado de la condición presentada sea satisfecho.

La nueva oferta viable $\delta^{(i)}$ es hecha por el agente como una propuesta a su contraparte y la contraparte dará la realimentación con el grado de satisfacción. Si el grado de satisfacción para $\delta^{(i)}$ es más alto que cualquiera de sus padres, el padre con el menor grado de satisfacción será reemplazado con $\delta^{(i)}$ para producir el próximo hijo. Sino es así, los padres tratarán de generar otro hijo viable. El resultado para μ_i es un solo vector que alcanza la utilidad objetivo y tiene la más alta puntuación de estas ofertas clasificadas por la contraparte. Este vector se conoce como $\delta^{(i)}_{mejor}$.

Al finalizar el proceso, la estrategia de negociación S es un conjunto de valores que contienen un subconjunto ordenado de valores: utilidad μ_i , cada una asociada con $\delta^{(i)}_{mejor}$ y su grado de satisfacción $0 < t \leq t_{max}$, es decir:

$$S = \{ \{ \mu_1, \delta^{(1)}_{mejor}, Ag_1 Grado(\delta^{(1)}_{mejor}) \}, \{ \mu_2, \delta^{(2)}_{mejor}, Ag_2 Grado(\delta^{(2)}_{mejor}) \}, \dots, \{ \mu_{t_{max}}, \delta^{(t_{max})}_{mejor}, Ag_{t_{max}} Grado(\delta^{(t_{max})}_{mejor}) \} \}. \quad (6.5)$$

Hasta el momento, se a descrito como es formada una estrategia de negociación para un agente dada la opción de β y k^s . A continuación detallaremos como es el proceso para evolucionar las estrategias de negociación.

Para evaluar completamente una estrategia de negociación se examina la suma, para cada μ_i , del grado de satisfacción del agente mismo y de su contraparte. Posteriormente se encuentra el máximo de esta suma y a este resultado se le llama $SGrado = MAX(Ag_1 Grado(\delta^{(i)}_{mejor})) + Ag_2 Grado(\delta^{(i)}_{mejor})$ y $0 < i \leq t_{max}$.

Lo anterior es el punto de vista de la propia estrategia de negociación de un agente. Sin embargo, al mismo tiempo la contraparte está haciendo contraofertas así como también provee grados de satisfacción para cualquier oferta hecha. Esto significa

que cada agente puede llevar un proceso de evaluación de estrategias en paralelo alimentándose ambos del mismo valor de t_{max} .

Cada uno de los dos agentes elige aleatoriamente, dos valores para β , estos valores son usados para definir estrategias para cada agente. El primer agente tiene estrategias $S^{(1)}_j (j = 1, 2)$ y el segundo agente tiene $S^{(2)}_j (j = 1, 2)$. El rango de valores para β en la selección aleatoria es asignado a priori y es tomado para que sea el mismo en ambos agentes. Dos rondas de evaluación toman lugar usando el mecanismo previamente descrito. Así $S^{(1)}_1$ es jugado contra $S^{(2)}_1$ y $S^{(1)}_2$ es jugado contra $S^{(2)}_2$. Cada estrategia obtiene un grado, $SGrado$, para su agente. Para ambos agentes la estrategia de negociación inicial $S^{(i)}_1$, serán los padres que generaran dos nuevas estrategias $S^{(i)}_3, S^{(i)}_4$. El agente usará la nueva estrategia de negociación para jugar con su contraparte para obtener un valor de $SGrado$. Si cualquiera de $SGrado^{(i)}_3$ o $SGrado^{(i)}_4$ es más alto que $SGrado^{(i)}_1$ o $SGrado^{(i)}_2$, entonces $S^{(i)}_1$ o $S^{(i)}_2$ será reemplazado con $S^{(i)}_3$ o $S^{(i)}_4$ como nuevo padre para generar estrategias. Este proceso será iterado q generaciones de evolución. Lo que significa que $2 \times q$ estrategias de negociación han sido generadas y jugadas por cada agente (como es de suponer, ambos agentes usan el mismo valor de q). El próximo paso es construir la matriz de retribución.

Las interacciones antes mencionadas entre los agentes han sido un proceso de búsqueda y no ha sido un proceso para alcanzar un acuerdo. Cada agente selecciona su mejor estrategia g (la g es el valor más alto de $SGrado$ y $g < 2 \times q$). Para cada uno de las $g \times g$ combinaciones de los dos agentes negociadores. Con una estrategia elegida la secuencia $\delta^{(i)}_{mejor}$ para esa estrategia es usada. Para una combinación dada una conclusión será alcanzada y la retribución para cada agente es calculada. Ahora es posible formar una matriz de retribución Γ . Cada celda en la tabla Γ contiene dos valores de utilidad $Ag_1u(\delta)$ y $Ag_2u(\delta)$ para Ag_1 y Ag_2 respectivamente, y δ es su acuerdo.

Una vez la matriz Γ es formada, el método *TTP* es usado para razonar sobre Γ usando sus dos acciones de comunicación: garantía (G) y compensación (C). El proceso alcanza un equilibrio *NFD* el cual satisface el criterio de eficiencia de Pareto y equilibrio de Nash.

En la primera etapa de la formulación de estrategias el valor de k^s a sido fijado. Cada agente asume tener un par de valores, $Ag_i\mu_{max}$ y $Ag_i\mu_{min}$, que especifican el rango de utilidades que se pueden considerar. k^s define un sub-rango de estas utilidades y tiene el efecto de incrementar la mínima utilidad esperada. El valor más grande de

k^s . es el más alto umbral. Esto permite suponer razonablemente que los valores más grandes de k^s incrementan el mutuo beneficio potencial de los dos agentes.

Los eventuales valores de las retribuciones en el acuerdo *NFD* son nfd^1 y nfd^2 . Estos son usados para modificar la k^i en Ag_1 y Ag_2 respectivamente. En lugar de cambiar un solo valor común para k^s , los nuevos k^i son un conjunto igual a $(nfd^i - Ag_i\mu_{min}) / (Ag_i\mu_{max} - Ag_i\mu_{min})$. Esto permite una nueva dirección en la búsqueda para mejorar el trato. El *NFD* se convierte un punto de referencia para que ambos agentes exploren mucho más el espacio de búsqueda. Todo el proceso descrito desde generar estrategias de negociación hasta producir un *NFD* es una etapa de negociación.

Los agentes modifican su valor de k^i y guardan la estrategia de negociación para explorar más el espacio de búsqueda en la próxima etapa de negociación. Si ambos agentes no encuentran una mejor solución en etapas posteriores del *NFD* alcanzado, la retribución en el *NFD* es la menor cantidad que ellos pueden obtener.

El proceso de negociación termina cuando una de las siguientes condiciones es satisfecha.

- El número de etapas de negociación es alcanzado independientemente de que exista o no acuerdo.
- Existe un acuerdo sobre el trato δ el cual es derivado desde la retribución acordada en el equilibrio *NFD*. Como el sistema puede no estar en capacidad de encontrar una retribución exactamente equivalente a la asociada con el equilibrio *NFD*, el trato δ que genere valores aproximados es aceptado.

Posteriormente se mostrará el resultado del proceso de negociación automatizada a todos los usuarios del sistema, ver figura 6.2.

6.3.1. Juego Mediado por una Tercera Parte Confiable (TTP, Trusted Third Party)

Un juego mediado por un TTP puede ser definido como una tupla $\langle A, S, P, MN \rangle$. La notación $A = \{P, Q\}$ es un conjunto de agentes, S es el conjunto de estrategias fijada para cada agente en A , $S = \{S_P, S_Q\}$, $S_P = P_1, P_2, \dots, P_n$, $S_Q = \{Q_1, Q_2, \dots, Q_n\}$,

$P = \{Payoff_P, Payoff_Q\}$ son las funciones de retribución que mapean cada combinación estratégica a un valor de retribución para los agentes P y Q respectivamente $Payoff_P : S_P \times S_Q \rightarrow R$, $Payoff_Q : S_P \times S_Q \rightarrow R$, y MN es mecanismo de negociación. La función de retribución de cada agente sobre cada combinación de estrategias forma una matriz de juego. Los agentes pueden usar el mecanismo de negociación MN para alterar la matriz de juegos y encontrar un equilibrio aceptable.

Se usa el juego mediado por la tercera parte confiable para modelar la coordinación multiagente. El objetivo es encontrar un compromiso estable y justo. Este propósito puede ser logrado mediante el MN .

6.3.2. Mecanismo de Negociación

El mecanismo de negociación MN se define como una tupla $\langle TTP, AC, PN \rangle$. Donde el TTP es un agente tercera parte confiable, AC es el conjunto de acciones de comunicación que un agente puede usar para negociar y PN es el protocolo de negociación. El conjunto AC incluye acciones de comunicación de garantía y compensación, i.e., $AC = \{ACG, ACC\}$. La retribución de cada agente sobre cada combinación de estrategias será cambiada si el agente usas estas dos acciones de comunicación. El protocolo de negociación es un conjunto de procedimientos para que los agentes hagan o acepten ofertas o realicen contraofertas. El protocolo ayuda a los agentes a lograr un estado particular que es estable, justo y aceptable.

El mecanismo de negociación MN es una parte externa del juego mediado por una tercera parte confiable. Bajo situaciones diferentes, el mecanismo puede necesitar ser reconstruido. El mecanismo incluye dos tipos de acciones de comunicación. Las dos acciones proveen una forma para negociar retribuciones y por consiguiente cambiar la matriz de juego desde un juego difícil a uno aceptable. El protocolo general de negociación es un ciclo de aceptar ofertar o hacer contraofertas.

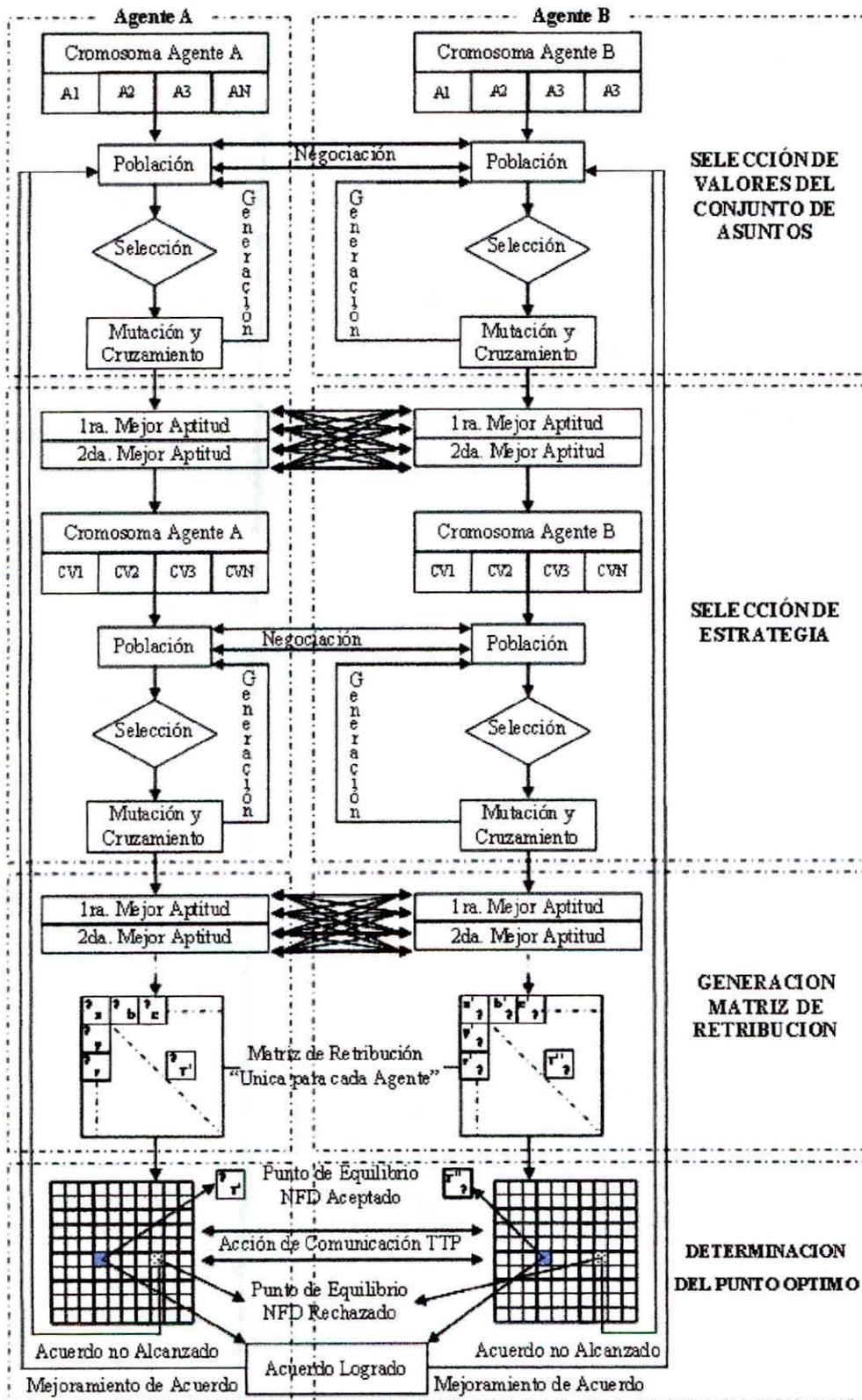


Figura 6.2: Mecanismo de Negociación Automatizada. Tomado y adaptado de [33].

6.3.3. La Tercera Parte Confiable

La tercera parte confiable *TTP* es una agente independiente del dominio que no se involucra en la matriz del juego y en quien todos los agentes en el juego confían. La tercera parte confiable debe tener la habilidad de ejecutar las acciones de comunicación necesarias tales como garantía y compensación. La tercera parte confiable tiene la habilidad de observar el comportamiento de los agentes y detectar si los agentes mantienen su compromiso.

6.3.4. Acción de Comunicación

La acción de comunicación $AC = \{ACG, ACC\}$ incluye las acciones de comunicación de garantía y compensación que pueden alterar la matriz de juego, de esta manera el juego puede ser cambiado desde un juego difícil a un juego con un equilibrio aceptable. Las dos acciones de comunicación, garantía y compensación, pueden ayudar a coordinar los agentes inteligentes.

6.3.5. Acción de Comunicación de Garantía

La acción de comunicación de garantía $ACG = S^*, G$ incluye una estrategia particular S^* y una garantía G . El efecto de que el agente P garantice que solo juega S_P^* es: Para todas las estrategias del S_Q agente Q y todas las estrategias del agente P excepto S_P^* , la *Retribución* $P(S'_P, S_Q)$ se convierte en *Retribución* $(S'_P, S_Q) - G$, y la *Retribución* $Q(S'_P, S_Q)$ se convierte en *Retribución* $(S'_P, S_Q) + G$, donde G es la cantidad de garantía que es incluida en la oferta que lleva la acción de comunicación de garantía durante el proceso de negociación.

6.3.6. Acción de Comunicación de Compensación

La acción de comunicación de compensación $ACC = S^*, C$ incluye una estrategia particular S^* y una compensación C . El efecto de la acción de comunicación de compensación a causa de que el agente Q juegue S_Q^* es: Para las estrategias S_Q^* del agente Q y todas las estrategias S'_P del agente P , la *Retribución* $P(S'_P, S_Q)$ se convierte *Retribución* $P(S'_P, S_Q^*) - C$ y la *Retribución* $Q(S'_P, S_Q^*)$ se convierte en *Retribución* $Q(S'_P, S_Q^*) + C$. Donde C es la cantidad de compensación que es incluida en la oferta que lleva la acción de comunicación de compensación durante el proceso de negociación.

6.3.7. Protocolo General de Negociación

Se describe el protocolo general de negociación PN. Se detalla como realizar ofertas, rechazar y hacer contraofertas. El protocolo es simétrico, i.e., los agentes lo usan para hacer una oferta y el otro agente puede también aceptarla o hacer una contraoferta en el mismo protocolo. Podrían existir muchas formas diferentes de hacer una oferta y diferentes criterios de aceptar una oferta.

El procedimiento se presenta en dos etapas, la primera es un procedimiento general de negociación y la segunda es cada sub-procedimiento. Note que ambos agentes usan el mismo procedimiento y la garantía y compensación deben ser depositadas en la tercera parte confiable aparte de los otros agentes. El resultado es independiente de cual agente inicia la negociación [51].

Procedimiento $P1$ (Procedimiento general de negociación).

1. Construcción de la matriz de juego.
2. Ordenar las retribuciones en la matriz de juego y formar un conjunto ordenado P de retribuciones.
3. Ciclo de ofertas, contra ofertas, negociación.
 - a) El agente i ofrece una combinación estratégica particular. $P2$
 - b) El agente $-i$ acepta la oferta o propone una contra oferta. $P3$
 - c) La negociación finaliza sobre un resultado acordado o no existen más ofertas (falla).

Fin del Procedimiento $P1$.

La etapa 2 detalla sub-procedimiento:

Procedimiento $P2$ (El sub-procedimiento de hacer una oferta).

1. Selecciona la combinación estratégica (S_i^*, S_{-i}^*) que da la más alta retribución al agente i desde el conjunto P .
2. Calcula la cantidad de garantía que hace (S_i^*, S_{-i}^*) un NFD para el agente i por el Procedimiento $P4$. El procedimiento encuentra la cantidad de garantía que debe pedir al agente $-i$ para jugar (S_i^*, S_{-i}^*) y para que el mismo agente i no se desvíe desde (S_i^*, S_{-i}^*) . Entonces se la envía al agente $-i$.

3. Si (S_i^*, S_{-i}^*) ha sido ofrecida antes, entonces la mínima cantidad de compensación adicional para $-i$ se obtiene por medio del procedimiento $P5$, para persuadir al agente $-i$ a aceptar (S_i^*, S_{-i}^*) . Si (S_i^*, S_{-i}^*) aún da la más alta retribución para el agente i , entonces envía la oferta al agente $-i$, de otra forma borra (S_i^*, S_{-i}^*) de P y va a 1 y vuelve a seleccionar otra combinación estratégica.

Fin del Procedimiento $P2$.

Procedimiento $P3$ (El sub-procedimiento de aceptar una oferta o hacer una contra oferta).

Seleccione la combinación estratégica (S'_i, S'_{-i}) que da la más alta retribución al agente i desde el conjunto P . Si la retribución de (S_i^*, S_{-i}^*) es menor que la retribución de (S'_i, S'_{-i}) , rechace (S_i^*, S_{-i}^*) , haga otra oferta por medio del procedimiento $P2$.

Fin del Procedimiento $P3$.

Acorde al procedimiento $P3$, los agentes deberían aceptar solo sugerencias que dan la más alta posible retribución.

Procedimiento $P4$ (El sub-procedimiento of calcular la cantidad de garantía).

Para un resultado particular (S_i^*, S_{-i}^*) , compare la retribución del agente i en todas las otras estrategias S'_i en la matriz de juego, encuentre la máxima distancia $d1 = \max P_i(S'_i, S_{-i}^*) - P_i(S_i^*, S_{-i}^*), \forall S'_i \neq S_i^*$. Si $d1$ es mayor que 0, la garantía que el agente i debe depositar es $d1 + \delta$ de otra forma no hay necesidad para el agente $-i$ deposite cualquier garantía. Donde la garantía marginal δ es una cantidad pequeña de retribución que hace que el agente cambie el orden dominante. El mismo procedimiento puede encontrar la cantidad de garantía para el agente $-i$. Para un resultado final (S_i^*, S_{-i}^*) compare la retribución del agente $-i$ en todas las otras estrategias (S'_{-i}) en la matriz de juego, encuentre la máxima distancia $d2 = \max P_{-i}(S_i^*, S'_{-i}) - P_{-i}(S_i^*, S_{-i}^*), \forall S'_{-i} \neq S_{-i}^*$. Si $d2$ es más grande que 0, la garantía que el agente $-i$ debería depositar es $d2 + \delta$ de otra forma no hay necesidad para que el agente i deposite cualquier garantía.

Fin del Procedimiento $P4$.

La cantidad de garantía es la mínima cantidad de retribución lo suficientemente elevada para prevenir la desviación.

Procedimiento $P5$ (El sub-procedimiento de calcular la cantidad de compensación).

Se necesita una cantidad mínima de retribución θ . Si la retribución no tiene una cantidad básica, la mitad de la mínima diferencia entre dos retribuciones de una combinación estratégica puede ser tratada como θ .

Fin del Procedimiento P5.

6.3.8. Características del Sistema de Negociación Automatizada

El sistema de negociación posee las siguientes características:

- Los agentes solamente tienen acceso cada uno a su matriz de retribuciones y función de utilidad, no tienen conocimiento de las retribuciones y función de utilidad de los otros agentes.
- Ya que el proceso co-evolutivo es responsable de encontrar altas retribuciones para los agentes negociadores buscando en el espacio de potenciales acuerdo, el trato o acuerdo no puede ser hecho en este nivel. Esto asegura que el espacio de potenciales acuerdos es explorado completamente y los agentes actúan de una forma racional, por ejemplo, no aceptar una oferta que es menor a una anterior.
- El algoritmo de equilibrio NFD es aplicado a la matriz de retribución, cuando es requerido, obtener un equilibrio de Nash que también es eficiencia de Pareto.
- Si el equilibrio NFD no satisface los requerimientos de los agentes, la fase co-evolutiva es revisitada para seleccionar una nueva estrategia.
- Una vez los agentes acuerdan sobre las retribuciones producidas por el algoritmo NFD, es necesario determinar las estrategias que conducen a estas retribuciones. Para este fin el proceso del GA es invocado con la función de adaptabilidad ajustada para reflejar el nuevo equilibrio NFD. Como el GA no puede ser capaz de encontrar una correspondencia exacta a la retribución asociada con el equilibrio NFD, una solución aproximada es aceptable.
- Las acciones de comunicación de Compensación y Garantía son incorporadas para permitirle a los agentes reevaluar las retribuciones en la matriz en su búsqueda de un equilibrio.
- El sistema se detiene cuando un acuerdo mutuamente aceptable es logrado siguiendo un nuevo equilibrio o cuando los recursos de los agentes se hayan agotado.

6.4. Agentes del Sistema MultiAgente

El propósito del prototipo del sistema multiagente desarrollado es soportar un proceso de negociación automatizada. Al final del proceso cada agente negociador participante deberá arrojar un resultado en caso de haber logrado un acuerdo, acorde al equilibrio de Nash y la eficiencia de Pareto.

A continuación se describirán los agentes utilizados en el prototipo, para tal fin se enunciará al agente en cuestión, se mostrará un diagrama que describe su comportamiento y por medio de una tabla se realizará la descripción del agente. Todo esto gracias a la metodología de desarrollo de SMA INGENIAS [7].

6.4.1. Agente Negociador

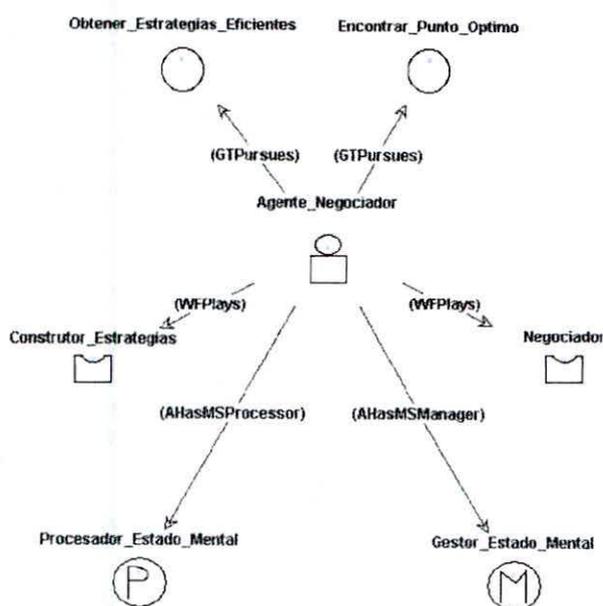


Figura 6.3: Modelo de Agente Negociador. Representa el estado mental, objetivos y roles.

Tabla 6.1: Descripción del Modelo de Agente Negociador.

NOMBRE	TIPO	DESCRIPCIÓN
Procesador Estado Mental	MentalState Processor	El Procesador del Estado Mental se encarga de la toma de decisiones del agente y de su control. Esas tareas se realizan con las siguientes funciones: evaluación de propuestas, evaluación de acuerdo y todas las funciones que se realizan en la teoría de juegos.

Continúa en la siguiente página...

... Viene de la página anterior

NOMBRE	TIPO	DESCRIPCIÓN
Constructor Estrategias	Role	Mediante este rol el Agente Negociador asume construir estrategias eficientes para comenzar el proceso de negociación, la clave de este rol radica en que dichas estrategias buscan que el proceso de negociación converja rápidamente, debido a que dichas estrategias son creadas con base en grados de satisfacción suministrados por los agentes participantes en el proceso de negociación.
Negociador	Role	Este rol lo asume el Agente Negociador basado en el perfil del usuario para representarlo en las diferentes etapas del proceso de negociación.
Agente Negociador	Agent	Un agente Negociador representa al usuario en el sistema. Se encarga de sustituirle en el proceso de negociación. Además de negociar, su objetivo es lograr un acuerdo con el mayor grado de utilidad posible.
Obtener Estrategias Eficientes	Goal	El propósito de este objetivo es sacar el máximo provecho a los dos Algoritmos Genéticos con los que cuenta el prototipo de negociación automatizada. Para tal fin se hace uso de la comunicación entre agentes y se permite antes de comenzar el proceso de negociación, establecer grados de satisfacción que permitirán generar utilidades para la matriz de retribución más cercanas a lo que espera cada participante en el proceso de negociación.
Gestor Estado Mental	MentalState Manager	Las operaciones básicas del Gestor del Estado Mental son: creación, construcción, modificación y monitorización del conocimiento del agente. Es decir, la evolución del estado mental. Entonces se puede decir que funciones como el Algoritmo Genético, el Envío de Propuestas, La Evaluación de Propuestas hacen parte del Gestor del Estado Mental.
Encontrar Punto Óptimo	Goal	El objetivo principal del Agente Negociador es encontrar un punto de equilibrio, que a la vez le genere la retribución mas alta, teniendo en cuenta el punto de vista del los demás Agentes Negociadores. Esto lo logra buscando en el espacio de posibles acuerdos (espacio de búsqueda) haciendo uso de las acciones de comunicación de garantía y compensación y razonando gracias a los conceptos de teoría de juegos.

6.4.2. Agente TTP

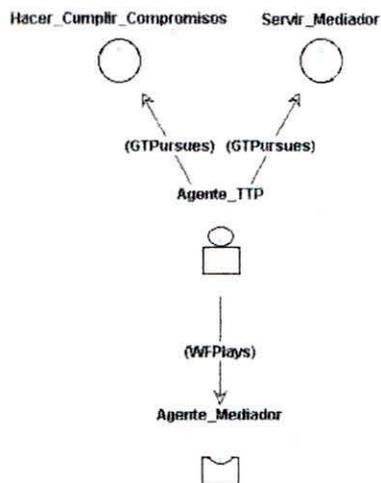


Figura 6.4: Modelo de Agente TTP. Representa los objetivos y roles.

Tabla 6.2: Descripción del Modelo de Agente TTP.

NOMBRE	TIPO	DESCRIPCIÓN
Agente Mediador	Role	Este rol lo asume el Agente TTP (Tercera Parte Confiable), básicamente consiste en servir de intermediario entre los Agentes Negociadores y tratar de que se lleven a cabo los compromisos adquiridos entre estos.
Servir Mediador	Goal	Este objetivo consiste en hacer llegar las garantías o compensaciones entre los Agentes Negociadores, es decir, servir de mediador.
Agente TTP	Agent	El Agente TTP (Tercera Parte Confiable) es el encargado de conducir el proceso de negociación, hacer llegar las propuestas entre los agentes negociadores, ejecutar las acciones de comunicación (garantía y/o compensación), hacer cumplir las obligaciones. Este es un agente mediador en el proceso de negociación.
Hacer Cumplir Compromisos	Goal	Uno de los objetivos principales del Agente TTP, consiste en darle confianza a los Agentes Negociadores en que los compromisos pactados entre ellos si se cumplirán.

6.5. Modelos del Sistema MultiAgente

Ya se ha podido observar los modelos de los agentes como unidad, a continuación se presentaran los modelos que describen como los agentes se desenvuelven en el ambiente.

6.5.1. Modelos Interacción

Se han identificado dos interacciones bien definidas en el sistema desarrollado, estas interacciones permiten a los agentes lograr sus objetivos de una manera eficiente. Así mismo, las interacciones identificadas hacen uso de los mensajes para comunicar los agentes negociadores y al agente TTP respectivamente. Las figuras 6.5 y 6.6, presentan los diagramas de los modelos de interacción.

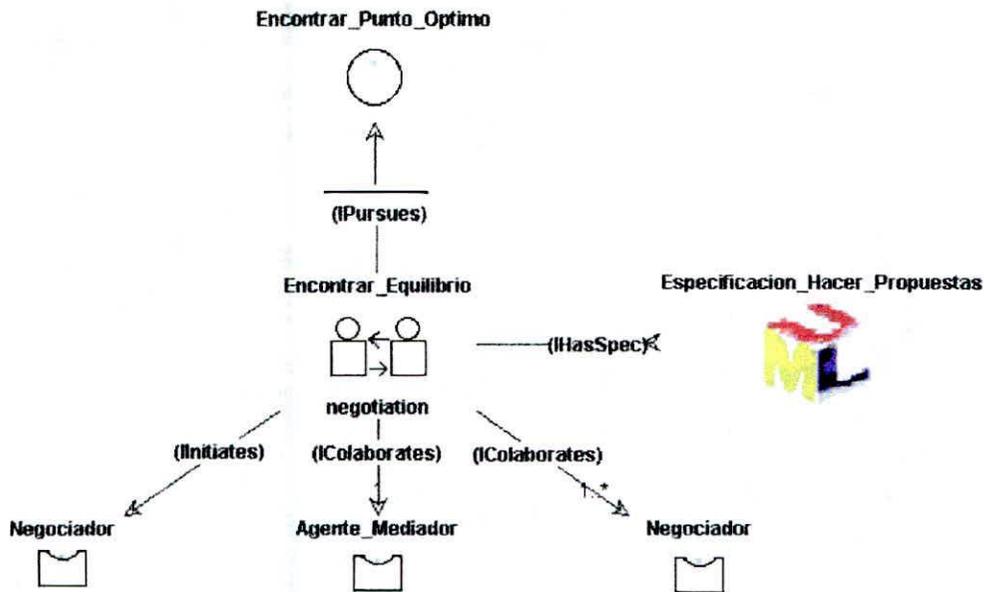


Figura 6.5: Modelo de Interacción Encontrar Equilibrio.

Tabla 6.3: Descripción del Modelo de Interacción Encontrar Equilibrio.

NOMBRE	TIPO	DESCRIPCIÓN
Especificación Hacer Propuestas	UML Specification	Ver Modelo de Colaboración.
Encontrar Equilibrio	Interaction	Esta interacción es el núcleo del prototipo de negociación automatizada, ya que por medio de ella se logra que los Agentes Negociadores por medio del Agente TTP, alcancen un equilibrio en caso de que este exista el espacio de potenciales acuerdos.
Negociador	Role	Este rol lo asume el Agente Negociador basado en el perfil del usuario para representarlo en las diferentes etapas del proceso de negociación.
Encontrar Punto Optimo	Goal	El objetivo principal del Agente Negociador es encontrar un punto de equilibrio, que a la vez le genere la retribución más alta, teniendo en cuenta el punto de vista de los demás Agentes Negociadores. Esto lo logra buscando en el espacio de posibles acuerdos (espacio de búsqueda) haciendo uso de las acciones de comunicación de garantía y compensación y razonando gracias a los conceptos de teoría de juegos.
Agente Mediador	Role	Este rol lo asume el Agente TTP (Tercera Parte Confiable), básicamente consiste en servir de intermediario entre los Agentes Negociadores y tratar de que se lleven a cabo los compromisos adquiridos entre estos.

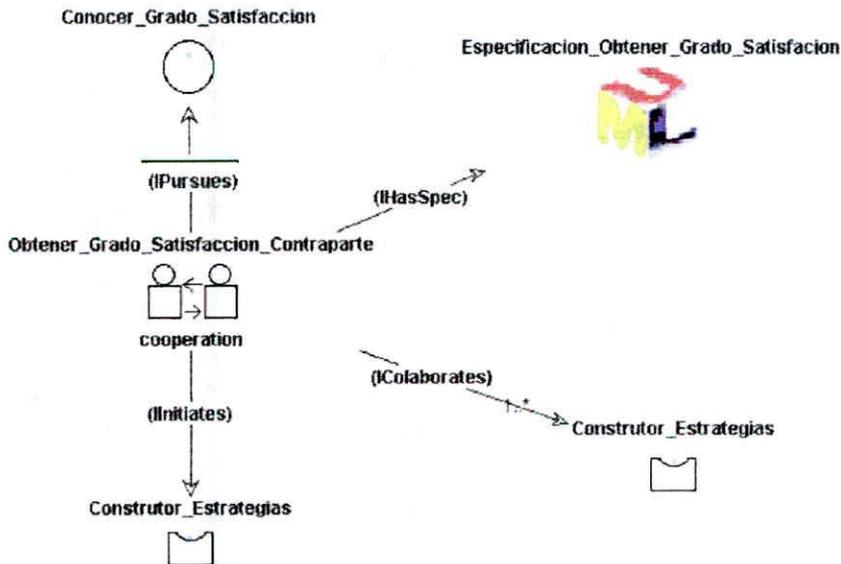


Figura 6.6: Modelo de Interacción Realimentación Propuestas.

Tabla 6.4: Descripción del Modelo de Interacción Realimentación Propuestas.

NOMBRE	TIPO	DESCRIPCIÓN
Constructor Estrategias	Role	Mediante este rol el Agente Negociador asume construir estrategias eficientes para comenzar el proceso de negociación, la clave de este rol radica en que dichas estrategias buscan que el proceso de negociación converja rápidamente, debido a que dichas estrategias son creadas con base en grados de satisfacción suministrados por los agentes participantes en el proceso de negociación.
Especificación Obtener Grado Satisfacción	UML Specification	Este diagrama de especificación nos sirve para detallar la forma en que los Agentes Negociadores realizan el intercambio de mensajes, de esta forma obtienen información adicional sobre las propuestas que están construyendo y recibir las opiniones de los demás Agentes Negociadores en el punto en el que a cada quien le compete.
Obtener Grado Satisfacción Contraparte	Interaction	Por medio de las interacciones entre Agentes Negociadores, cada Agente Negociador puede obtener información adicional de su contraparte; acerca de las propuestas que va creando el AG. Además, esta información adicional acortara el proceso de negociación que se realizará más adelante.
Conocer Grado Satisfacción	Goal	El objetivo en este modelo de interacción, consiste en obtener el grado de satisfacción de cada una de las propuestas que el AG va elaborando.

6.5.2. Modelos de Entorno

A continuación se presentan modelos acerca de como los agentes perciben y modifican el entorno, con base en el prototipo implementado se detalla el modelo de entorno tanto para los Agentes Negociadores como para el Agente TTP. Las figuras 6.7 y 6.8, presentan los diagramas de los modelos de entorno.

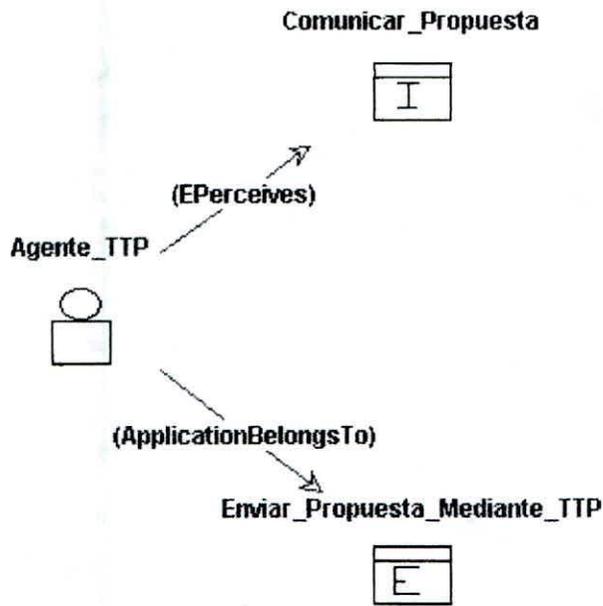


Figura 6.7: Modelo de Entorno Agente TTP.

Tabla 6.5: Descripción del Modelo de Entorno Agente TTP.

NOMBRE	TIPO	DESCRIPCIÓN
Comunicar Propuesta	Internal Application	Este procedimiento le permite al Agente TTP comunicar la propuesta que recibe de parte de un Agente Negociador para otro en específico.
Agente TTP	Agent	El Agente TTP (Tercera Parte Confiable) es el encargado de conducir el proceso de negociación, hacer llegar las propuestas entre los agentes negociadores, ejecutar las acciones de comunicación (garantía y/o compensación), hacer cumplir las obligaciones. Este es un agente mediador en el proceso de negociación.
Enviar Propuesta Mediante TTP	Environment Application	Todos los Agentes Negociadores envían sus propuestas, las cuales además contienen la cantidad de compensación o garantía calculada por cada agente, entre si por medio del Agente Mediador, utilizando esta función.

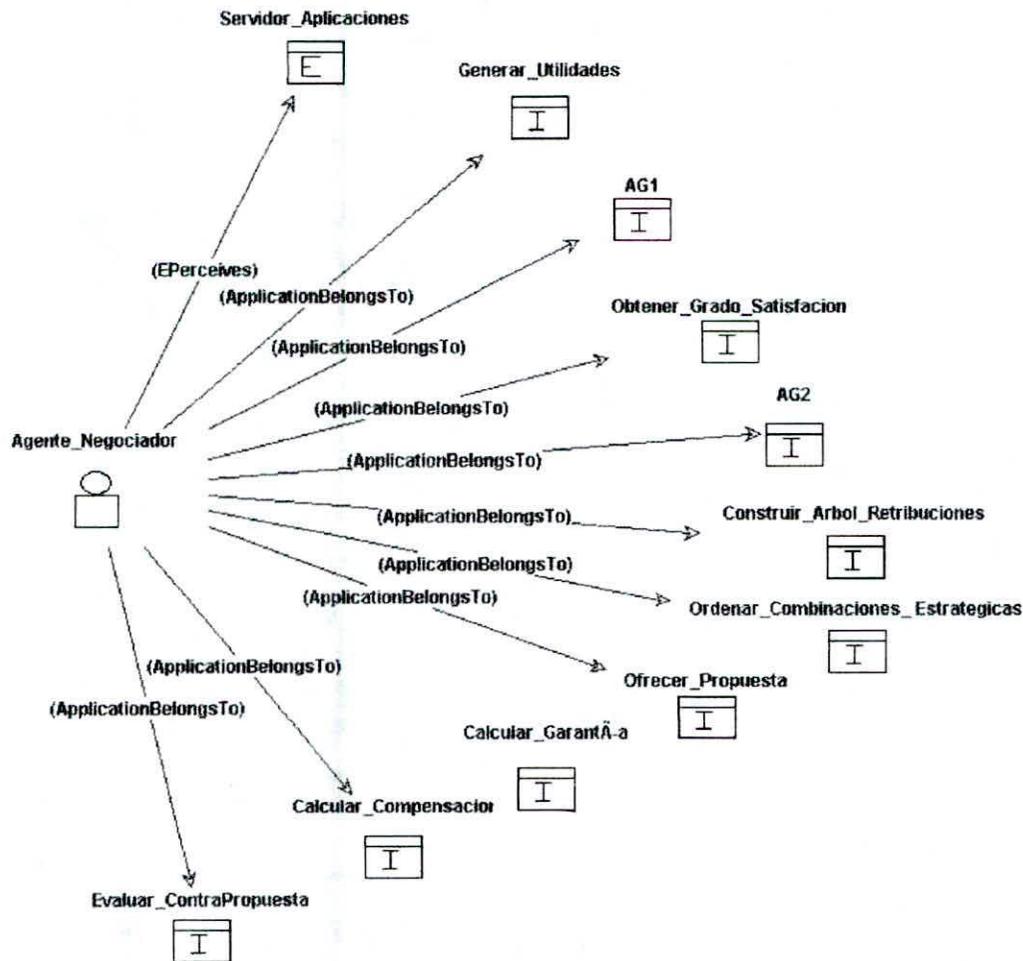


Figura 6.8: Modelo de Entorno Agente Negociador.

Tabla 6.6: Descripción del Modelo de Entorno Agente Negociador.

NOMBRE	TIPO	DESCRIPCIÓN
Calcular Garantía	Internal Application	Este procedimiento permite a los agentes establecer que cantidad de garantía pueden ofrecer a su contraparte.
Construir Matriz Retribuciones	Internal Application	Todo Agente Negociador genera una matriz de retribución después de construir y perfeccionar las estrategias de negociación.
Agente Negociador	Agent	Un agente Negociador representa al usuario en el sistema. Se encarga de sustituirle en el proceso de negociación. Además de negociar, su objetivo es lograr un acuerdo con el mayor grado de utilidad posible.

Continúa en la siguiente página...

... Viene de la página anterior

NOMBRE	TIPO	DESCRIPCIÓN
AG1	Internal Application	El primer algoritmo genético (AG1) se encargara de generar utilidades con base en los asuntos que se trataran en el proceso de negociación, teniendo en cuenta el grado de satisfacción para cada conjunto de asunto en específico suministrado por la contraparte.
Ordenar Combinaciones Estratégicas	Internal Application	Mediante este procedimiento se ordena en un arreglo las combinaciones estratégicas las combinaciones estratégicas que brindan mayor retribución a un Agente Negociador en específico. Dicho orden es de mayor a menor y le servirá al Agente Negociador como toma de decisión para la aceptación de ofertas, así como también la elaboración de contraofertas.
Servidor Aplicaciones	Environment Application	Por medio de esta aplicación cada agente captura de la base de datos lo necesario para representar a un usuario del sistema.
Evaluar Contrapropuesta	Internal Application	Mediante este procedimiento el Agente Negociador evalúa una oferta que le fue realizada por otro Agente Negociador. Gracias a este procedimiento el agente podrá ofrecer una respuesta ya sea aceptando la propuesta o rechazándola para lo cual ofrecerá una contraoferta.
Obtener Grado Satisfacción	Internal Application	Gracias a este procedimiento se acerca más el modelo a la realidad, debido a que los agentes se pueden comunicar y establecer grados de satisfacción para los conjuntos de asuntos que ellos trataran (AG1), así como también, el grado de satisfacción para el conjunto de estrategias que serán en ultimas las que llenaran la matriz de retribución.
AG2	Internal Application	El segundo algoritmo genético generara estrategias (conjunto de utilidades) con base en los resultados arrojados por el primer algoritmo genético, buscando mejorarlas para encontrar resultados más acordes a las necesidades de los Agentes Negociadores.
Generar Utilidades	Internal Application	Este procedimiento permite generar las utilidades con las cuales se establecerá el proceso de negociación.
Calcular Compensación	Internal Application	Este procedimiento permite a los agentes establecer que cantidad de compensación necesitan para jugar determinada estrategia.
Ofrecer Propuesta	Internal Application	Este procedimiento le permite establecer comunicación al Agente Negociador con los otros Agentes Negociadores para darle paso a la negociación.

6.5.3. Modelos de Tareas y Objetivos

A continuación se presentan modelos sobre las tareas que realizan los agentes negociadores y el agente TTP para alcanzar los distintos objetivos que conducen a una

meta final. lograr un acuerdo entre las partes que conforman el proceso de negociación. Las siguientes figuras, presentan los diagramas de los modelos de objetivos y tareas con sus respectivas tablas de descripción.

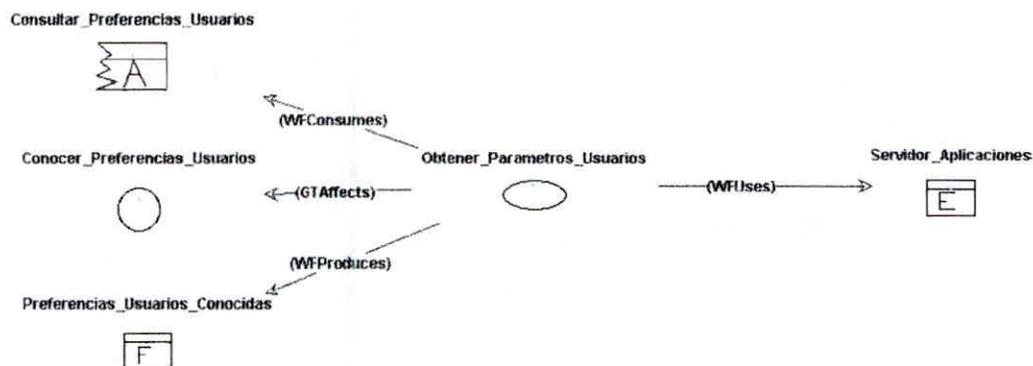


Figura 6.9: Modelo de Tareas y Objetivos Obtener Parámetros de Usuarios.

Tabla 6.7: Descripción del Modelo de Tarea y Objetivos Obtener Parámetros de Usuarios.

NOMBRE	TIPO	DESCRIPCIÓN
Conocer Preferencias Usuarios	Goal	Mediante este objetivo se define si ya se puede comenzar la personalización del Agente Negociador que representará al usuario en el proceso de negociación con base en los parámetros establecidos por este.
Preferencias Usuarios Conocidas	Fact	Con este hecho el Agente Negociador muestra que ya esta listo (tiene toda la información necesaria) para entrar en la fase de construcción de estrategias.
Servidor Aplicaciones	Environment Application	Por medio de esta aplicación cada agente captura de la base de datos lo necesario para representar a un usuario del sistema.
Obtener Parámetros Usuarios	Task	Esta tarea les permite a los Usuarios entrar en el sistema y posteriormente personalizar el agente.
Consultar Preferencias Usuarios	Application Event	Por medio de este evento se le notifica al Agente Negociador que ya puede consultar toda la información referente al usuario que representara en el proceso de negociación.

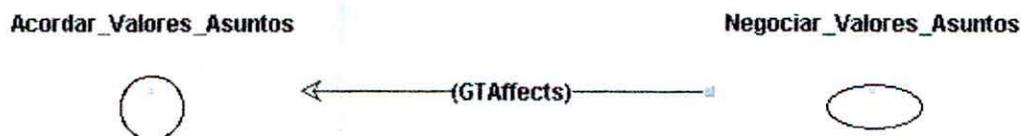


Figura 6.10: Modelo de Tareas y Objetivos Negociar Valores de Asuntos.

Tabla 6.8: Descripción del Modelo de Tarea y Objetivos Negociar Valores de Asuntos.

NOMBRE	TIPO	DESCRIPCIÓN
Negociar Valores Asuntos	Task	Antes de comenzar el proceso de negociación y de generar la matriz de retribución sobre la cual se negociará, los Agentes Negociadores se comunican entre si para acordar el valor esperado en el conjunto de asuntos de parte y parte.
Acordar Valores Asuntos	Goal	Gracias a este propósito, se puede esperar que el primer Algoritmo Genético (AG1) genere conjuntos de asuntos los cuales repercutirán en utilidades esperadas acordes al grado de satisfacción brindado por la contraparte, lo cual, en resumidas cuentas repercutirá en un proceso de negociación que converge mas rápido en caso de existir un acuerdo.

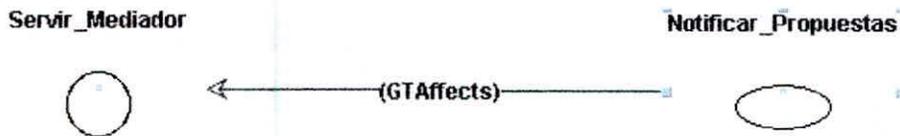


Figura 6.11: Modelo de Tareas y Objetivos Notificar Propuestas.

Tabla 6.9: Descripción del Modelo de Tarea y Objetivos Notificar Propuestas.

NOMBRE	TIPO	DESCRIPCIÓN
Notificar Propuestas	Task	Mediante esta tarea se pretende que los Agentes Negociadores notifiquen al Agente TTP que propuesta aceptan o en su defecto rechazan.
Servir Mediador	Goal	Este objetivo consiste en hacer llegar las garantías o compensaciones entre los Agentes Negociadores, es decir, servir de mediador.

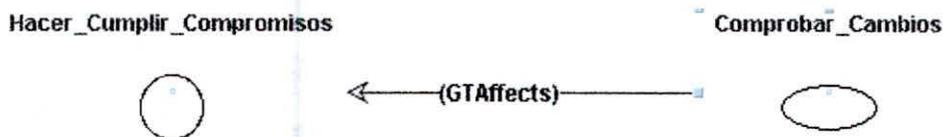


Figura 6.12: Modelo de Tareas y Objetivos Comprobar Cambios.

Tabla 6.10: Descripción del Modelo de Tarea y Objetivos Comprobar Cambios.

NOMBRE	TIPO	DESCRIPCIÓN
Comprobar Cambios	Task	Como el agente TTP es el agente que garantiza la confianza en el proceso de negociación se cerciora de que en realidad los Agentes Negociadores realizan los cambios en su respectiva matriz de retribución, para garantizar la transparencia del proceso de negociación automatizado el Agente TTP también guarda copia de todos los cambios realizados por los Agentes Negociadores.
Hacer Cumplir Compromisos	Goal	Uno de los objetivos principales del Agente TTP, consiste en darle confianza a los Agentes Negociadores en que los compromisos pactados entre ellos si se cumplirán.

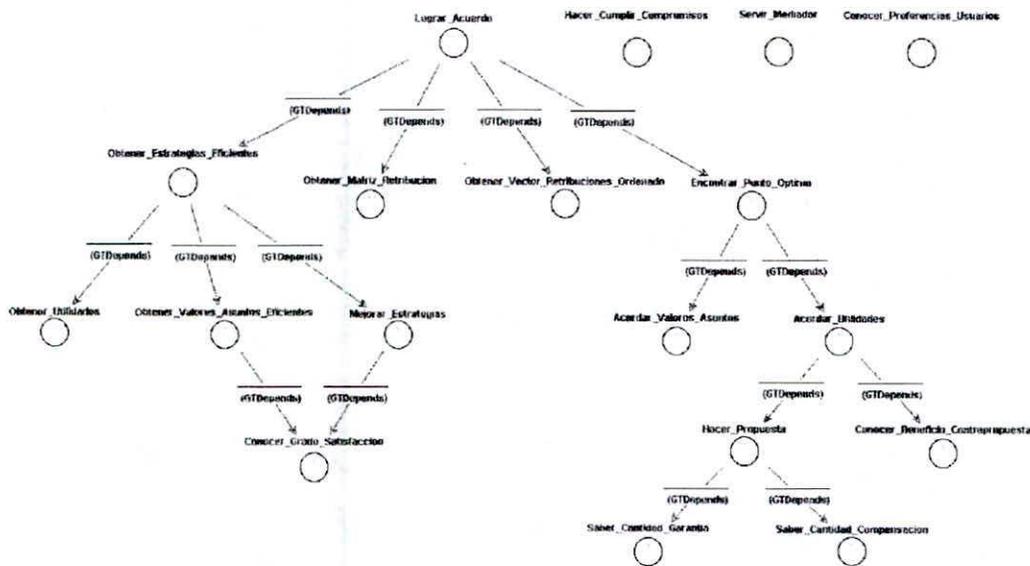


Figura 6.13: Modelo de Tareas y Objetivos Descomposición de Objetivos.

Tabla 6.11: Descripción del Modelo de Descomposición de Objetivos.

NOMBRE	TIPO	DESCRIPCIÓN
Conocer Beneficio Contrapropuesta	Goal	Por medio de este objetivo los agentes negociadores pueden conocer el beneficio que les brinda una oferta hecha por otro agente.
Obtener Valores Asuntos Eficientes	Goal	Este objetivo le permite a los Agentes Negociadores hacer uso de la comunicación y del primer algoritmo genético antes de comenzar el proceso de negociación para obtener grados de satisfacción del conjunto de asuntos que conforman las variables de la función de utilidad, esto con el fin de que dichos valores sean acordes a lo que espera la contraparte negociadora, lo cual repercutirá en un acortamiento del proceso de negociación, una vez este comience.
Obtener Vector Retribuciones Ordenado	Goal	Mediante este objetivo se logra ordenar un vector con un tamaño previamente definido el cual tendrá el conjunto de retribuciones que ofrecen un mayor beneficio al Agente Negociador, dicho orden como es de esperar será de mayor a menor. Esto se realiza con el fin de permitirle al Agente Negociador una vez comenzado el proceso de negociación, iniciar ofreciendo la utilidad que le ofrece mayor beneficio en caso de que su propuesta no sea aceptada, pasa a la siguiente que es la segunda utilidad que arroja mayor beneficio y así sucesivamente.
Conocer Grado Satisfacción	Goal	El objetivo en este modelo de interacción, consiste en obtener el grado de satisfacción de cada una de las propuestas que el AG va elaborando.
Hacer Propuesta	Goal	Mediante este objetivo el Agente Negociador se comunican a su contraparte la estrategia que desea jugar o en su defecto la estrategia que desea que juegue.
Hacer Cumplir Compromisos	Goal	Uno de los objetivos principales del Agente TTP, consiste en darle confianza a los Agentes Negociadores en que los compromisos pactados entre ellos si se cumplirán.
Encontrar Punto Óptimo	Goal	El objetivo principal del Agente Negociador es encontrar un punto de equilibrio, que a la vez le genere la retribución mas alta, teniendo en cuenta el punto de vista del los demás Agentes Negociadores. Esto lo logra buscando en el espacio de posibles acuerdos (espacio de búsqueda) haciendo uso de las acciones de comunicación de garantía y compensación y razonando gracias a los conceptos de teoría de juegos.

Continúa en la siguiente página...

... Viene de la página anterior

NOMBRE	TIPO	DESCRIPCIÓN
Saber Cantidad Garantía	Goal	Este objetivo permite calcular la cantidad de garantía que se ofrecerá en la propuesta.
Obtener Estrategias Eficientes	Goal	El propósito de este objetivo es sacar el máximo provecho a los dos Algoritmos Genéticos con los que cuenta el prototipo de negociación automatizada. Para tal fin se hace uso de la comunicación entre agentes y se permite antes de comenzar el proceso de negociación, establecer grados de satisfacción que permitirán generar utilidades para la matriz de retribución más cercanas a lo que espera cada participante en el proceso de negociación.
Lograr Acuerdo	Goal	El objetivo general consiste en lograr un acuerdo entre los agentes negociadores, el cual este acorde al concepto de eficiencia de Pareto y equilibrio de Nash.
Obtener Utilidades	Goal	Mediante este objetivo se obtendrán las utilidades que conformaran el espacio de búsqueda del proceso de negociación automatizado.
Saber Cantidad Compensación	Goal	Este objetivo permite calcular la cantidad de compensación que se ofrecerá en la propuesta.
Servir Mediador	Goal	Este objetivo consiste en hacer llegar las garantías o compensaciones entre los Agentes Negociadores, es decir, servir de mediador.
Acordar Valores Asuntos	Goal	Gracias a este propósito, se puede esperar que el primer Algoritmo Genético (AG1) genere conjuntos de asuntos los cuales repercutirán en utilidades esperadas acordes al grado de satisfacción brindado por la contraparte, lo cual, en resumidas cuentas repercutirá en un proceso de negociación que converge mas rápido en caso de existir un acuerdo.
Obtener Matriz Retribución	Goal	Este objetivo le permite al segundo algoritmo genético crear la matriz de retribución para cada agente participante en el proceso de negociación, la cual conformará el espacio de búsqueda de posibles acuerdos.
Acordar Utilidades	Goal	Este objetivo le permite al agente ir modificando la matriz de restricciones para acordar al final un equilibrio.
Mejorar Estrategias	Goal	Este objetivo le permite a los Agentes Negociadores hacer uso de la comunicación y del segundo algoritmo genético antes de comenzar el proceso de negociación para obtener grados de satisfacción del conjunto de utilidades que conforman las estrategias, esto con el fin de que dichos valores sean acordes a lo que espera la contraparte negociadora, lo cual repercutirá en un acortamiento del proceso de negociación, una vez este comience.

Continúa en la siguiente página...

... Viene de la página anterior

NOMBRE	TIPO	DESCRIPCIÓN
Conocer Preferencias Usuarios	Goal	Mediante este objetivo se define si ya se puede comenzar la personalización del Agente Negociador que representará al usuario en el proceso de negociación con base en los parámetros establecidos por este.

6.5.4. Casos de Uso

En esta sección se presenta un diseño del prototipo del sistema implementado desde el punto de vista de aplicación y no como un sistema multiagente. Para tal fin, se mostrará el diagrama de casos de uso, construido con INGENIAS.

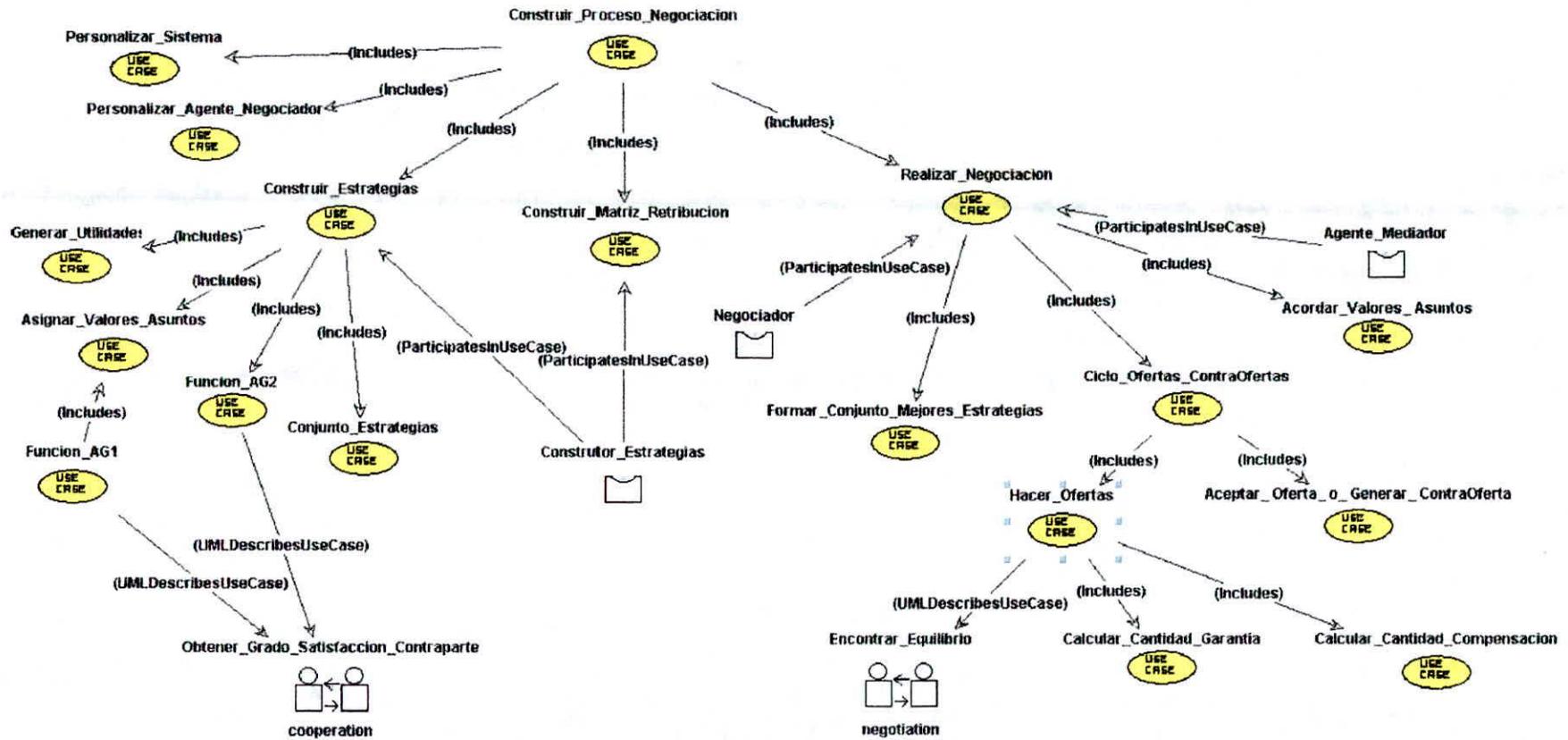


Figura 6.14: Modelo de Casos de Uso

Tabla 6.12: Descripción del Modelo de Casos de Uso.

NOMBRE	TIPO	DESCRIPCIÓN
Asignar Valores Asuntos	Text Use Case	Los Agentes solamente negocian sobre los asuntos I y el actual valor de Au es usado como utilidad esperada para buscar una posible combinación de valores de asuntos los cuales produzcan Au. Nosotros denotamos esta función de búsqueda como $GAs(I,Au)$. Además, I es igual al vector $d(i)$ el cual es un conjunto de valores para los asuntos I. Los vectores $d(i)$ satisfacen la condición $-Agku(d(i))-Aui - i = E$ ($Agku$ corresponde a la función de utilidad de cada agente). Debido a que el algoritmo genético no puede generar valores exactos a Aui , debemos tener un intervalo permisible y esto se logra gracias a $[Au-E,Au+E]$ (donde E es una constante inicializada al momento de personalizar al Agente Negociador).
Función AG1	Text Use Case	Esta función se encarga de generar valores para cada una de las variables que se tienen en cuenta en la negociación, luego la evalúa gracias a la función de utilidad $Agku(d(i))$ (donde $d(i)$, es el conjunto de variables de la negociación) y si cumple con la condición, es decir, $-Agku(d(i))-Aui - i = E$ entonces la envía como propuesta a los demás Agentes Negociadores para recibir su grado de satisfacción. Dicho grado de satisfacción se toma como el fitness de la propuesta y al final de la función $AGs(I,Au)$ el $d(i)_{best}$ (el mejor conjunto de valores para las variables de la negociación) asociado a la utilidad Aui , es la propuesta con el grado de satisfacción mas alto.
Ciclo Ofertas Contraofertas	Text Use Case	Mediante este ciclo se da comienzo a la negociación en sí, en el los Agentes Negociadores se entablaran comunicación por medio del Agente TTP y enviando ofertas y contraofertas según sea el caso.
Calcular Cantidad Compensación	Text Use Case	El calculo de la compensación sirve a los Agentes Negociadores para incentivar que un agente que juegue una estrategia deseada.
Conjunto Estrategias	Text Use Case	Al final del proceso Construir Estrategias, la estrategia de negociación S es un conjunto de valores que contiene un subconjunto ordenado de valores: utilidad Aui , cada una con su mejor vector $d(i)_{best}$ asociado y su grado de satisfacción. i.e. $S = Au1, d(1)_{best}, AgiScore(d(1)_{best}), Au2, d(2)_{best}, AgiScore(d(2)_{best}), \dots, Autmax, d(tmax)_{best}, AgiScore(d(tmax)_{best})$.

Continúa en la siguiente página...

... Viene de la página anterior

NOMBRE	TIPO	DESCRIPCIÓN
Acordar Valores Asuntos	Text Use Case	Antes de comenzar la negociación y se hace uso de la comunicación y el primer algoritmo genético para acordar los valores de los asuntos, que no son más que el conjunto de variables de la función de utilidad que el usuario a suministrado. Este proceso se realiza con el fin de que dichos valores permitan generar utilidades más acorde a lo que cada parte espera; con esto se busca acortar el proceso de negociación, puesto que este alcanzará un acuerdo en menor tiempo.
Personalizar Sistema	Text Use Case	El administrador del sistema se encarga de configurar el sistema antes de que los Agentes Negociadores y el Agente TTP comiencen el proceso de negociación, incluso puede ser un proceso paralelo al de Personalizar Agente Negociador. El administrador interactúa con el sistema ingresando datos pertinentes para este como por ejemplo: el número de agentes negociadores, el tiempo que durara o las etapas que se llevaran a cabo y el número de iteraciones que se ejecutaran en cada etapa, etc. Al finalizar todas las etapas de negociación, el administrador se encargara de mostrar el resultado a todos los usuarios del sistema.
Formar Conjunto Mejores Estrategias	Text Use Case	El conjunto de mejores estrategias se forma para guiar al Agente Negociador en el proceso de negociación, para tal fin el Agente hará uso de dicho conjunto para negociar estrategias que vayan en un orden descendente, es decir, de mayor a menor.
Aceptar Oferta o Generar Contraoferta	Text Use Case	Cada vez que el un Agente Negociador se comunica con otro agente, este último está obligado a aceptar la propuesta que se le a enviado o en su defecto rechazarla, realizando una contraoferta. El paso de turno no está permitido en el proceso de negociación.
Constructor Estrategias	Role	Mediante este rol el Agente Negociador asume construir estrategias eficientes para comenzar el proceso de negociación, la clave de este rol radica en que dichas estrategias buscan que el proceso de negociación converja rápidamente, debido a que dichas estrategias son creadas con base en grados de satisfacción suministrados por los agentes participantes en el proceso de negociación.
Negociador	Role	Este rol lo asume el Agente Negociador basado en el perfil del usuario para representarlo en las diferentes etapas del proceso de negociación.
Función AG2	Text Use Case	Esta función da inicio al segundo algoritmo genético. El cual mejorará el conjunto de utilidades que representan una estrategia.

Continúa en la siguiente página ...

... Viene de la página anterior

NOMBRE	TIPO	DESCRIPCIÓN
Hacer Ofertas	Text Use Case	Mediante el cálculo de la garantía y compensación los Agentes Negociadores realizan ofertas a su contraparte para encontrar un equilibrio.
Construir Proceso Negociación	Text Use Case	Cada uno de los Usuarios del Sistema al interactuar con el software, personaliza al agente que lo representara en la negociación, al mismo tiempo el administrador del sistema ingresara información relevante para la ejecución del software como por ejemplo: las etapas de negociación que se llevaran a cabo, el número de iteraciones que se ejecutaran en cada etapa, etc. Luego cada agente negociador construirá sus estrategias de negociación, las codificara en una matriz de retribución y entre todos los agentes negociadores trataran de encontrar un punto óptimo común, gracias al razonamiento sobre estas matrices por medio del enfoque de la teoría de juegos. Además, este punto óptimo común servirá como guía para realizar mas exploración sobre el espacio de posibles acuerdos debido a que existe la posibilidad de mejorar dicho punto y finalmente terminara su ejecución el sistema cuando se cumpla el numero de etapas de negociación, en lo posible, con un acuerdo entre todos los agente negociadores acorde a la eficiencia de Pareto y al equilibrio de Nash, mostrándole el resultado del proceso de negociación automatizada a todos los usuarios del sistema.
Encontrar Equilibrio	Interaction	Esta interacción es el núcleo del prototipo de negociación automatizada, ya que por medio de ella se logra que los Agentes Negociadores por medio del Agente TTP, alcancen un equilibrio en caso de que este exista el espacio de potenciales acuerdos.
Construir Estrategias	Text Use Case	Todo agente generara una utilidad esperada para cada iteración que se lleve a cabo dentro de cada etapa, la cual será enviada al AG. El AG se encarga de generar valores para los asuntos a tratar escogiendo la combinación de valores que se aproxime mas a la utilidad esperada en ese momento. Esta combinación de valores le permitirá al agente realizar una oferta a los demás agentes participantes en el proceso de negociación; recibiendo de estos un grado de satisfacción y una contraoferta. Al final del proceso, la estrategia de negociación es un conjunto de valores que contiene un subconjunto ordenado de utilidad, combinación estratégica y grado de satisfacción.

Continúa en la siguiente página...

... Viene de la página anterior

NOMBRE	TIPO	DESCRIPCIÓN
Agente Mediador	Role	Este rol lo asume el Agente TTP (Tercera Parte Confiable), básicamente consiste en servir de intermediario entre los Agentes Negociadores y tratar de que se lleven a cabo los compromisos adquiridos entre estos.
Construir Matriz Retribución	Text Use Case	Después de perfeccionar las estrategias los Agentes Negociadores construirán la matriz de retribución, en nuestro caso esto equivale a construir el árbol del juego, seleccionando las mejores estrategias (el numero de estrategias para generar la matriz es establecido previamente por el administrador del sistema), esto permitirá sentar la bases para comenzar el proceso de negociación.
Realizar Negociación	Text Use Case	Una vez construida la matriz de retribución entramos en un ciclo de negociación que culminara cuando se cumple el numero de encuentros estipulados o se logre un acuerdo, en dicho ciclo se llevaran a cabo las siguientes actividades: el mecanismo TTPMG -juego Mediado por Tercera Parte Confiable- (Trusted Third Party Mediated Game, por sus siglas en inglés) es usado para razonar sobre la matriz utilizando sus dos acciones de comunicación, las cuales son Garantía y Compensación, este proceso comienza cuando un Agente Negociador hace una sugerencia, para formar un punto de equilibrio aceptable usando las acciones citadas basadas sobre la suposición de mutua racionalidad, el Agente Tercera Parte Confiable tiene la habilidad de observar el comportamiento de los Agentes Negociadores y detectar si estos agentes mantienen sus compromisos. Luego el otro agente decidirá si acepta esto o no bajo algún criterio. Si existe una mejor sugerencia, rechaza la oferta y responde con una contraoferta.

Continúa en la siguiente página. . .

... Viene de la página anterior

NOMBRE	TIPO	DESCRIPCIÓN
Realizar Negociación	Text Use Case	Al finalizar los encuentros (ciclos de negociación) se debe contar o no con un equilibrio NFD -Sin Temor a Desviación- (No Fear of Deviation, por sus siglas en inglés) en caso de existir un equilibrio este se encuentra acorde al criterio de eficiencia de Pareto, por el contrario, en caso de que no exista un acuerdo entre dos o mas estrategias, un valor por omisión, el cual es una utilidad negativa, será asignada a los agentes. Esta última actividad se realiza de acuerdo a la teoría de negociación de Nash. Bajo el principio de la metodología co-evolutivo la cual asume que la existencia de un equilibrio de Nash puede ser refinado, teniendo el NFD alcanzado comienzan nuevamente los procesos de: Construir Estrategias, Perfeccionar Estrategias, Construir Matriz de Retribución, Realizar Negociación. Todo este proceso finaliza cuando se cumplen todas las etapas de negociación, cabe destacar, que si se ha alcanzado un NFD y en etapas posteriores no se mejora ese NFD quiere decir que esta es la mejor retribución que se puede obtener.
Obtener Grado Satisfacción Contraparte	Interaction	Por medio de las interacciones entre Agentes Negociadores, cada Agente Negociador puede obtener información adicional de su contraparte; acerca de las propuestas que va creando el AG. Además, esta información adicional acortara el proceso de negociación que se realizara mas adelante.
Personalizar Agente Negociador	Text Use Case	El usuario estará en capacidad de definir el perfil del agente puesto que este es el que lo representara en el proceso de negociación. Como tal podrá especificar el mínimo valor de utilidad esperada, la máxima ganancia esperada, la tasa de cruzamiento, la tasa de mutación, el numero de mutaciones.
Calcular Cantidad Garantía	Text Use Case	La garantía se puede dar y recibir, en este sentido, es vital en el proceso de negociación puesto que lo dinamiza con el fin de establecer compromisos bien definidos entre los Agentes Negociadores.
Generar Utilidades	Text Use Case	El primer paso para crear las estrategias es generar las utilidades, ya que estas se envían al AG para que este genere valores para los asuntos a tratar en la negociación. Al generar las utilidades se tiene en cuenta la siguiente ecuación $F(t, B, k(s)) = k(i) + (1 - k(i))(t/tmax)^{1/B}$, en donde los valores de k(s) son asignados a los k(i) y este valor de k(s) es arbitrario entre [0,1] en la primera etapa de negociación en las posteriores etapas k(s) depende de los equilibrios NFD que se acuerden. Además, el valor de B es escogido aleatoriamente entre un conjunto de valores previamente establecidos y los valores de t y tmax dependen del número de iteraciones establecidas para cada etapa al personalizar el sistema por parte del administrador del sistema.

Continúa en la siguiente página...

... Viene de la página anterior

NOMBRE	TIPO	DESCRIPCIÓN
Generar Utilidades	Text Use Case	Las utilidades generadas en cada iteración de cada etapa de negociación dependen de $AgiUmin + F(t,B,k(i))$ ($AgiUmin$ y $AgiUmax$ son valores asignados por el usuario del sistema en el momento de personalizar el Agente Negociador que lo representara.

Capítulo 7

PRUEBAS DEL SISTEMA

Para evaluar la eficacia del Sistema de Negociación Automatizada se realizaron pruebas con base en los siguiente escenarios.

7.1. Escenario 1

Este escenario está compuesto por un Empleador (denotado como Ag_1) y un representante de un grupo de empleados que para este caso lo designaremos como Empleado (denotado como Ag_2). Los agentes tratarán de acordar un punto bajo los siguientes criterios:

X_1 = Subsidio de Educación

X_2 = Porcentaje sobre la Plusvalía

La función de utilidad para el Ag_1 estará definida como:

$$f(Ag_1) = (1850000 - X_1) * (1 - X_2) \quad (7.1)$$

Y para el Ag_2 tenemos:

$$f(Ag_2) = 100000 + X_1 + (1850000 - X_1) * X_2 \quad (7.2)$$

La Tabla 7.1 presenta los mínimos y máximos de las funciones de utilidad y las variables comunes al proceso de negociación.

VARIABLE	MÍNIMO	MÁXIMO
X_1 para Ag_1	400000	550000
X_2 para Ag_1	0.0	0.1
X_1 para Ag_2	400000	700000
X_2 para Ag_2	0.05	0.15
$f(Ag_1)$	1150000	1450000
$f(Ag_2)$	572500	972500

Tabla 7.1: Descripción Variables de Negociación Escenario 1.

El sistema ha sido configurado como se detalla en la Tabla 7.2, observe que el valor del *Quantum* corresponde al valor que los agentes ofrecerán haciendo uso de la *Acción de Comunicación de Compensación*, este será igual al *Quantum* multiplicado por el máximo valor esperado de la función de utilidad de cada agente. Así mismo, el *Nivel de Tolerancia* especificará el rango que al que debe limitarse el Algoritmo Genético una vez concluye la negociación, es decir, ajustar los valores de los conjuntos de asuntos para que reflejen los valores acordados por medio del Algoritmo NFD.

CONFIGURACIÓN ALGORITMO GENÉTICO 1		
Tamaño Cromosoma	12	
Tamaño Población	10	
Número Generaciones	10	
Selección	Tipo	por Torneo
Tasa Cruce	Tipo	Multipunto
	Probabilidad	30 %
	Puntos	2
Tasa Mutación	Tipo	Aleatoria
	Probabilidad	2 %
CONFIGURACIÓN ALGORITMO GENÉTICO 2		
Tamaño Población	10	
Número Generaciones	5	
Selección	Tipo	por Torneo
Tasa Cruce	Tipo	Multipunto
	Probabilidad	30 %
	Puntos	2
Tasa Mutación	Tipo	Aleatoria
	Probabilidad	2 %
VALORES COMUNES AL PROCESO DE NEGOCIACIÓN		
Número Estrategias	5	
Número Etapas	5	
Quantum	0.001	
Nivel de Tolerancia	0.01	

Tabla 7.2: Configuración del Sistema Escenario 1.

La Tabla 7.3 muestra la matriz de negociación una vez finalizada la primera etapa. Observe que el punto de acuerdo se presenta en cifras color azul.

		EMPLEADOR				
		Ag_1	Ag_1	Ag_1	Ag_1	Ag_1
E	Ag_2	997730	1051330	1105167	1135388	1212443
		952269	787664	790531	939557	746979
M	Ag_2	1267876	1312262	1317700	1294552	1317984
		793128	637737	638541	760988	632696
P	Ag_2	1159468	1205917	1212159	1188973	1212356
		844832	737840	737840	811512	737908
L	Ag_2	1033874	1106902	1161918	1174896	1212443
		791180	737556	737594	775103	737556
E	Ag_2	1203020	1212062	1212091	1189011	1212443
		737556	737556	737643	760988	737556

Tabla 7.3: Resultados Primera Etapa.

La Tabla 7.4 presenta resultados de la cuarta etapa.

		EMPLEADOR				
		Ag_1	Ag_1	Ag_1	Ag_1	Ag_1
E	Ag_2	987022	1043067	1099208	1132154	1210761
		962977	791072	794377	948399	748997
M	Ag_2	1269131	1315300	1320965	1298270	1321376
		796727	634699	635471	762032	628910
P	Ag_2	1155622	1204323	1210629	1188066	1211072
		850791	739238	739370	814518	738828
L	Ag_2	1024705	1100868	1158587	1173223	1211072
		794740	738828	738828	776776	739370
E	Ag_2	1201002	1210884	1211171	1187524	1211072
		739238	738828	738927	762032	738927

Tabla 7.4: Resultados Última Etapa.

La Tabla 7.5 muestra la matriz una vez finalizado el proceso de negociación.

		EMPLEADOR				
		Ag_1	Ag_1	Ag_1	Ag_1	Ag_1
E	Ag_2	1083910	1164086	1244067	1321907	1403547
		866089	620665	625232	814211	561295
M	Ag_2	1063623	1128702	1136694	1135788	1137836
		1051624	821297	822439	501418	813305
P	Ag_2	1059056	1127560	1136694	1135788	1137836
		971643	813305	813305	888922	812163
L	Ag_2	872124	980745	1063124	1114120	1137836
		891756	812163	812163	835879	812163
E	Ag_2	1122993	1136694	1137836	1135788	1137836
		812163	812163	812163	814211	812163
A	Ag_2					
D	Ag_2					
O	Ag_2					

Tabla 7.5: Matriz después de la Negociación.

Como se puede observar el acuerdo alcanzado en la Tabla 7.5 mejora sustancialmente los acuerdos alcanzados en las etapas representadas por las Tablas 7.4 y 7.3.

La Figura 7.1 muestra las distintas combinaciones estratégicas de las Tablas anteriormente presentadas, observe que los puntos con color rojo, amarillo y verde son los respectivos *NFD*, en donde el punto verde es el que representa el acuerdo final.

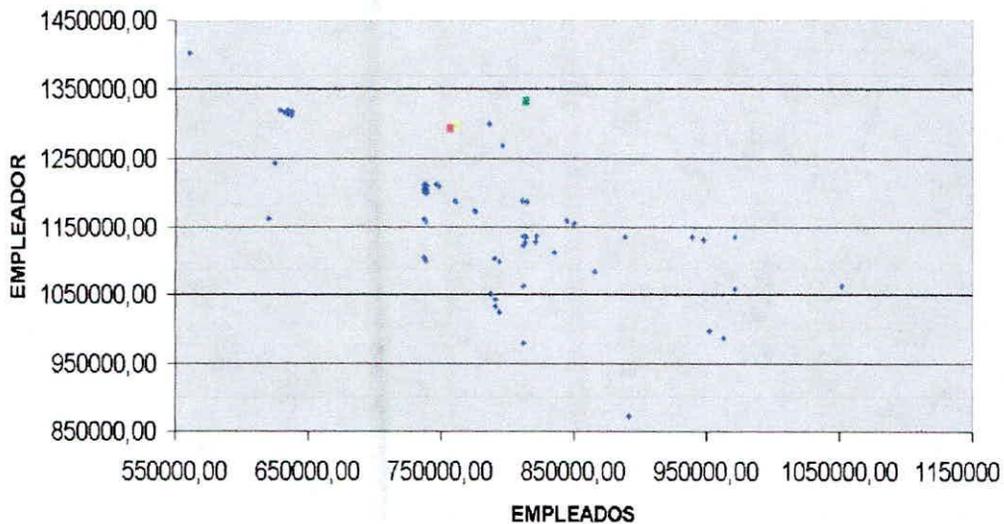


Figura 7.1: Combinaciones Estratégicas del Escenario 1.

7.2. Escenario 2

Presentamos un proceso de negociación entre reinsertados y gobierno, algunos de los datos con los cuales se elaboró el escenario fueron tomados del “Programa para

la Reincorporación a la Vida Civil (PRVC)” del “Ministerio del Interior y de Justicia (MIJ)”, la negociación será llevada por un representante del gobierno, el cual estará denotado por Ag_1 y un representante del grupo al margen de la ley denotado como Ag_2 . Los agentes tratarán de acordar un punto bajo los siguientes criterios:

$$X_1 = \text{Salario Derivado del Gobierno}$$

$$X_2 = \text{Tiempo en meses por el cual se devengará el salario}$$

El Ag_1 tiene las siguientes constantes:

$$\text{Salario por Informalidad} = 400000$$

$$\text{Costos Ingresos por Actividad Delictiva} = 500000$$

Así mismo, tenemos que para el Ag_1 :

$$\text{Ingreso} = X_1 + \text{Salario por Informalidad}$$

La función de utilidad para el Ag_1 estará definida como:

$$f(Ag_1) = (\text{Ingreso} - \text{Costos Ingresos por Actividad Delictiva}) * X_2 \quad (7.3)$$

El Ag_2 tiene las siguientes constantes:

$$\text{Educacion} = 416000$$

$$\text{Salud} = 50000$$

El gobierno lo que tiene es una función de costo puesto que en el periodo de estudio no tiene ingresos derivados de la actividad de los reinsertados (impuestos, incremento indirecto de la economía). La función para el Ag_2 estará definida como:

$$f(Ag_2) = (X_1 + \text{Educacion} + \text{Salud}) * X_2 \quad (7.4)$$

La Tabla 7.6 presenta los mínimos y máximos de las funciones de utilidad y las variables comunes al proceso de negociación.

VARIABLE	MÍNIMO	MÁXIMO
X_1 para Ag_1	350000	500000
X_2 para Ag_1	18	24
X_1 para Ag_2	400000	600000
X_2 para Ag_2	18	36
$f(Ag_1)$	14688000	23184000
$f(Ag_2)$	5400000	18000000

Tabla 7.6: Descripción Variables de Negociación Escenario 2.

El sistema ha sido configurado como se detalla en la Tabla 7.7:

CONFIGURACIÓN ALGORITMO GENÉTICO 1		
Tamaño Cromosoma	13	
Tamaño Población	10	
Número Generaciones	10	
Selección	Tipo	por Torneo
Tasa Cruce	Tipo	Multipunto
	Probabilidad	30 %
	Puntos	2
Tasa Mutación	Tipo	Aleatoria
	Probabilidad	2 %
CONFIGURACIÓN ALGORITMO GENÉTICO 2		
Tamaño Población	10	
Número Generaciones	7	
Selección	Tipo	por Torneo
Tasa Cruce	Tipo	Multipunto
	Probabilidad	30 %
	Puntos	2
Tasa Mutación	Tipo	Aleatoria
	Probabilidad	2 %
VALORES COMUNES AL PROCESO DE NEGOCIACIÓN		
Número Estrategias	5	
Número Etapas	4	
Quantum	0.001	
Nivel de Tolerancia	0.1	

Tabla 7.7: Configuración del Sistema Escenario 2.

La Tabla 7.8 nos muestra la matriz de negociación una vez finalizada la primera etapa, la cual representa el primer acuerdo dentro del proceso de negociación. Observe que el punto de acuerdo se presenta en cifras color azul.

REINSERTADO						
		<i>Ag₁</i>	<i>Ag₁</i>	<i>Ag₁</i>	<i>Ag₁</i>	<i>Ag₁</i>
G O B I E R N O	<i>Ag₂</i>	14244148	11572110	6643685	4031237	8881332
		20802273	23198652	19636041	19674250	22672132
	<i>Ag₂</i>	20079290	17570516	12244519	11366493	14003988
		14197855	14510607	13898948	12974060	10851199
	<i>Ag₂</i>	17690675	15215146	11431388	11294311	12160269
		14325210	15492876	14823062	13687001	10851199
<i>Ag₂</i>	15637468	13040800	11294311	11292944	11421597	
	13436960	14835358	14308223	12767077	10847009	
<i>Ag₂</i>	14244148	11291351	11291123	11298637	11294994	
	10962614	12614425	15903609	10862242	10851199	

Tabla 7.8: Resultados Primera Etapa.

La Tabla 7.9 presenta un acuerdo logrado en la etapa tres.

REINSERTADO						
		<i>Ag₁</i>	<i>Ag₁</i>	<i>Ag₁</i>	<i>Ag₁</i>	<i>Ag₁</i>
G O B I E R N O	<i>Ag₂</i>	15009000	12871000	9042512	7007977	10776705
		21405722	20866497	20298398	20190000	21589480
	<i>Ag₂</i>	19551000	17544979	13404865	12721070	14772683
		15648614	17536690	16873593	16534565	13567400
	<i>Ag₂</i>	17683000	15715149	12763423	12664371	12664371
		1697360	18038755	17364159	14689935	14101457
<i>Ag₂</i>	16093000	14025130	12662322	12658679	12658679	
	15621244	16463111	15481548	14319939	16146375	
<i>Ag₂</i>	15009000	12666000	12659590	12660728	12660728	
	15130451	16897753	13718003	14387271	13578466	

Tabla 7.9: Resultados Etapa Tres.

La Tabla 7.10 muestra la matriz una vez finalizado el proceso de negociación.

		REINSERTADO				
		Ag_1	Ag_1	Ag_1	Ag_1	Ag_1
G O B I E R N O	Ag_2	16761210	15709354	13799601	12778804	14661461
		22378000	21840000	21565000	24074037	22039593
	Ag_2	19015935	18033984	15966204	15622827	16651138
		19201901	19632899	19001908	18497749	18931707
	Ag_2	18088269	17120436	15649240	15600512	15937969
		19402075	19948132	20598592	18752254	18217696
Ag_2	17291988	16271100	15596413	15600512	15644914	
	19566318	20136216	19719745	20248246	18216261	
Ag_2	16761210	15593681		15601878	15600512	
	18243722	18579768	18286030	18219950	20979768	

Tabla 7.10: Matriz después de la Negociación.

Como se puede observar el acuerdo alcanzado en la Tabla 7.10 mejora sustancialmente los acuerdos alcanzados en las etapas representadas por las Tablas 7.9 y 7.8. Observe que el acuerdo logrado es el punto donde se alcanzan las más altas retribuciones acordes a la eficiencia de Pareto y al equilibrio de Nash.

La Figura 7.2 muestra las distintas combinaciones estratégicas de las Tablas anteriormente presentadas, observe que los puntos con color rojo, amarillo y verde son los respectivos NFD , en donde el punto verde es el que representa el acuerdo final.

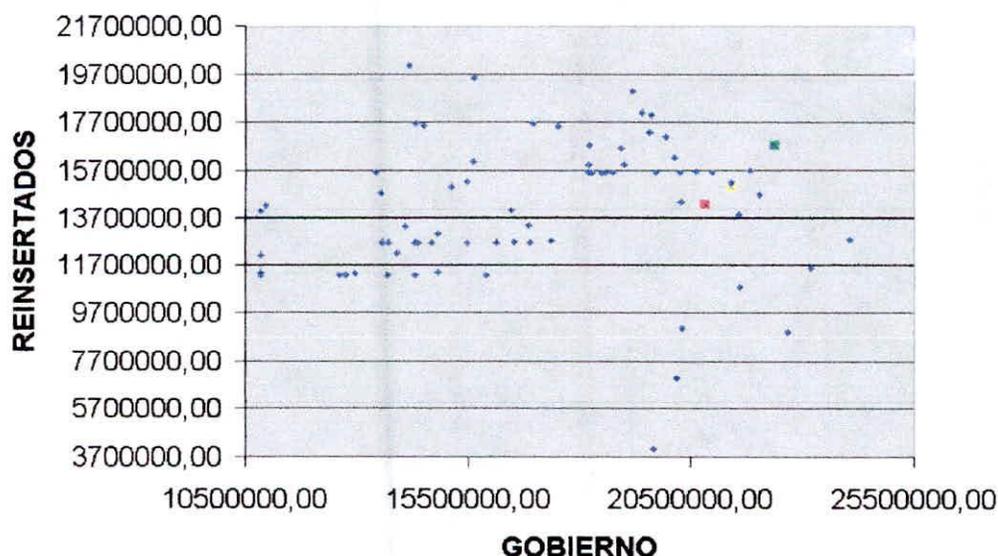


Figura 7.2: Combinaciones Estratégicas del Escenario 2.

Capítulo 8

APORTES Y CONTRIBUCIONES

El principal aporte del trabajo desarrollado es el enfoque sobre el cual se realizó el sistema, es decir, Democracia Electrónica, apuntamos a facilitar una herramienta de negociación al público en general. Aprovechar conceptos de Computación Evolutiva, Teoría de Juegos y Sistemas MultiAgentes y dirigirlos hacia la solución de problemas de negociación del mundo real es la piedra angular del proyecto. Reconocer las fortalezas y debilidades de cada una de las disciplinas citadas y poner al servicio de la comunidad académica colombiana y mundial un sistema que guíe un proceso de negociación que permita alcanzar un acuerdo acorde al equilibrio de Nash y la eficiencia de Pareto es un logro alcanzado.

Cabe destacar que aunque el propósito del sistema es ser útil en Democracia Electrónica su aplicabilidad no se restringe a dicho campo, cualquier negociación entre dos agentes es posible llevarla haciendo uso del Sistema de Negociación Sistemática. Gracias a que el sistema se encuentra montado sobre internet se proporciona su acceso a nivel global.

8.1. Divulgación de Resultados

Resultados parciales de este proyecto han sido presentados en los siguientes eventos:

- XI Escuela Internacional de Verano en Investigación de Operaciones, celebrada del 24 al 29 de julio en Villa de Leyva - Colombia. Título del artículo: *Un Enfoque Mixto de Computación Evolutiva y Teoría de Juegos para la Automatización en Sistemas de Soporte a la Negociación*[28].
- Congreso de Inteligencia Computacional 2005, celebrado en Montería Colombia del 10 al 12 de agosto del 2005. Título del artículo: *Un Enfoque Mixto de Computación Evolutiva y Teoría de Juegos para la Automatización en Sistemas de Soporte a la Negociación*[29].
- V Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento, celebradas en Puebla, Mexico, del 1 al 3 de Febrero de 2006, ISBN 970-94770-0-5. Título del artículo: *Computación Evolutiva y Teoría de Juegos*:

Un Híbrido Para La Automatización En Sistemas De Soporte A La Negociación[27].

Capítulo 9

CONCLUSIONES

El trabajo presentado es un análisis sobre la negociación automatizada, desde dos perspectivas fundamentales, teoría de juegos y la metodología co-evolutiva. El primer tema que se tiene en cuenta desde la teoría de juegos es el equilibrio de Nash. Aunque una solución de negociación acorde a Nash ofrece una interesante y útil perspectiva sobre el comportamiento de la negociación, falla para negociar con juegos donde se traten múltiples asuntos o se utilicen estrategias con información incompleta. La teoría de juegos puede tratar múltiples temas, pero dentro de un espacio de búsqueda limitado y con información completa. La adopción de una metodología co-evolutiva para negociación automatizada permite a los usuarios ignorar mucha de las suposiciones requeridas por la metodología de teoría de juegos.

La condición requerida por las metodologías de AG y teoría de juegos cuando ellas operan aisladamente restringe su rango de aplicación. El sistema de negociación ofrece una simbiosis entre una metodología de AG y una metodología de teoría de juegos que supera sus limitaciones inherentes. La suposición de la teoría de juegos de la existencia de una matriz de retribución con el propósito de alcanzar un equilibrio de Nash es frecuentemente difícil de cumplir.

La metodología de AG es, por otra parte, capaz de generar un conjunto de estrategias efectivas y relacionar retribuciones para proveerlas a la metodología de teoría de juegos. La metodología de AG requiere, sin embargo, la presencia de un equilibrio de Nash. En cambio, la metodología de teoría de juegos puede determinar un equilibrio sin temor a desviación desde la matriz de retribución y así poder proveer un punto focal para la metodología de AG, la cual permite refinar el equilibrio. Algunas de las ventajas que conlleva esta combinación es que es posible para los agentes negociantes llegar a un acuerdo sin conocer las retribuciones de cada uno de los agentes participantes en la negociación. La transparencia ofrecida por el sistema facilita el establecimiento de conexiones y la creación de equipos.

9.1. Trabajo Futuro

Nuestra investigación futura estará dirigida hacia negociaciones que involucren más de dos agentes, automatizar este tipo de negociaciones implica tratar con Juegos Cooperativos o Coalicionales y por ende se hace necesario apuntar a una serie de

nuevos conceptos que garanticen que el acuerdo logrado es el que mejor beneficia a los agentes que participan en un proceso de negociación determinado.

Bibliografía

- [1] ABRAHAM Ajith. *Evolutionary computation*. 2005.
- [2] COELLO COELLO Carlos Artemio. *An Empirical Study of Evolutionary Techniques for Multiobjective Optimization in Engineering Design*. Tesis PhD, Tulane University. [on-line], 1996. <http://delta.cs.cinvestav.mx/~ccoello/index.html>.
- [3] FOGEL D.B. *Evolutionary Computation*. IEEE Press, 1995.
- [4] GOLDBERG D.E. *Genetic Algorithms in Search Optimization and Machine Learning*. Addison Wesley, 1989.
- [5] ROCA Elia. *Las negociaciones*. 2004. <http://www.cop.es/colegiados/PV00520>.
- [6] SARTORI G. *Qué es la democracia*. Taurus, 2002.
- [7] GRASIA. *INGENIAS Development Kit: Tutorial and Manual*. Universidad Complutense de Madrid. [on-line], 2004. <http://ingenias.sourceforge.net>.
- [8] FICICI Sevan Gregory. *Solution Concepts in Coevolutionary Algorithms*. Tesis PhD, School of Arts and Sciences Brandeis University, 2004.
- [9] GULLIVER P. H. *Disputes and Negotiation: A Cross-Culture Perspective*. Academia Press Inc, 1979.
- [10] BUDGE I. *The New Democracy*. Policy Press, 1996.
- [11] RECHENBERG I. *Cybernetic Solution Path of an Experimental Problem*. Royal Aircraft Establishment, 1965.
- [12] KOZA J. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.
- [13] HOLLAND J.H. *Adaptation in Natural and Artificial Systems*. MIT Press, 1975.
- [14] FERNANDEZ Julio. *Complejidad y Algoritmos en Juegos Cooperativos*. Universidad de Sevilla, 2000.
- [15] SCOTT MORTON M. *The Corporation of the 90's*. Oxford U.P, 1991.
- [16] MITCHELL Melanie. *An Introduction to Genetic Algorithms*. The MIT Press, 1998.

- [17] VLASSIS Nikos. *A Concise Introduction to Multiagent Systems and Distributed AI*. University of Amsterdam. [on-line], 2003. <http://www.science.uva.nl/vlassis/cimasdai/>.
- [18] ETZIONI O. *Intelligence without robots*. 1993.
- [19] FARATIN Peyman. *Automated Service Negotiation Between Autonomous Computational Agents*. Tesis PhD, University of London, 2000.
- [20] AXELROD R. *The Evolution of Cooperation*. Basics Books Inc, 1984.
- [21] AXELROD R. *The Evolution of Strategies in the Iterated Prisoner's Dilemma, Genetic Algorithms and Simulated Annealing*. Lawrence Davis (ed.), 1987.
- [22] BROOKS R.A. *Cambrian Intelligence: the Early History of the New AI*. MIT Press, 1999.
- [23] KRAUS S. *Negotiation and cooperation in multi-agent environments*. 1997.
- [24] NWANA Hyacinth S. *Software agents: A overview*. 1996.
- [25] PEREZ Anselmo Serrada. *Una Introducción a la Computación Evolutiva*. 1996.
- [26] ZARTMAN I. W. *The 50 % Solution*. Vail-Ballou Press, 1981.
- [27] GOMEZ ROCHA Henry y ACOSTA PAREJO Jhon y VELEZ-LANGS Oswaldo. *Computación evolutiva y teoría de juegos: Un híbrido para la automatización en sistemas de soporte a la negociación*. 2005.
- [28] GOMEZ ROCHA Henry y ACOSTA PAREJO Jhon y VELEZ-LANGS Oswaldo. *Un enfoque mixto de computación evolutiva y teoría de juegos para la automatización en sistemas de soporte a la negociación*. 2005.
- [29] GOMEZ ROCHA Henry y ACOSTA PAREJO Jhon y VELEZ-LANGS Oswaldo. *Un enfoque mixto de computación evolutiva y teoría de juegos para la automatización en sistemas de soporte a la negociación*. 2005.
- [30] REN Z. y ANUMBA C. J. y UGWU O. O. *Negotiation in a multi-agent system for construction claims negotiation*. 2002.
- [31] MONSALVE Sergio y ARÉVALO Julián. *Un curso de teoría de juegos clásica*. Universidad Externado de Colombia, 2005.
- [32] RASMUSEN Eric y BLACKWELL Basil. *Games and Information: An Introduction to Game Theory*. [on-line], 2004.
- [33] CHEN Jen-Hsiang y CHAO Kuo-Ming y ANANE Rachid y GODWIN Nick. *Architecture of an agent-based negotiation mechanism*. 2002.
- [34] RIOS INSUA David y FERNANDEZ Eugenio y RIOS Jesús. *Más allá del gobierno electrónico: hacia la democracia electrónica*. 2004.
- [35] BILBAO J.M. y FERNANDEZ Julio. *Avances en teoría de juegos con aplicaciones económicas y sociales*. 2001.

- [36] GENESERETH M.R. y GINSBERG M.L. y ROSENSCHEIN J.S. Cooperation without communication. 1986.
- [37] CHEN Jen-Hsiang y GODWIN Nick y CHAO Kuo-Ming y SOO V-W. A multiple-stage cooperative negotiation. 2004.
- [38] HAUPT Randy y HAUPT Sue Ellen. *Practical Genetic Algorithms*. Wiley-Interscience, 2004.
- [39] WOOLDRIDGE Michael y JENNINGS Nicholas. Intelligent agents: Theory and practice. 1995.
- [40] PARSONS S. y JENNINGS P. y FARATIN A. R. y LOMUSCIO S. *Automated Negotiation: prospects, methods y challenges*. Int Journal of Group Decision and Negotiation, 2000.
- [41] WILSON Robert A. y KEIL Frank C. *The MIT encyclopedia of the cognitive sciences*. The MIT Press, 1999.
- [42] DURFEE E.H. y LEE J. y GMYTRASIEWICZ P.J. Overeager reciprocal rationality and mixed strategy equilibria. 1993.
- [43] JENNINGS Nicholas y LOMUSCIO Alessio R. y WOOLDRIDGE Michael. A classification scheme for negotiation in electronic commerce. 2000.
- [44] LUCK Michael y MCBURNEY Peter y SHEHORY Onn y WILLMOTT Steven. Agent technology roadmap. 2005. <http://www.agentlink.org>.
- [45] VON NEUMANN Jhon y MORGENSTERN Oskar. *The Theory of Games and Economic Behavior*. Princeton University Press, 1944.
- [46] RUSSELL Stuart J. y NORVIG Peter. *Artificial Intelligence: a Modern Approach*. Prentice Hall, 2003.
- [47] JENNINGS Nicholas y PARSONS S. y SIERRA C. y FARATIN P. Automated negotiation. 2000. <http://www.ecs.soton.ac.uk/~nrj/so-neg.html>.
- [48] NORIEGA P. y SIERRA C. *Agent Mediated Electronic Commerce (LNAI Volume 1571)*. Springer-Verlag, 1999.
- [49] MATOS N. y SIERRA C. y JENNINGS Nicholas. Determining successful negotiation strategies: an evolutionary approach. 1998.
- [50] WU Shih-Hung. y SOO Von-Wun. Escape from a prisoners'dilemma by communication with a trusted third party. 1998.
- [51] WU Shih-Hung. y SOO Von-Wun. Game theoretic reasoning in multi-agent coordination by negotiation with a trusted third party. 1999.
- [52] TUROCY Theodore L. y VON STENGEL Bernhard. *Game Theory*. Research Report, 2001.
- [53] JENNINGS Nicholas y WOOLDRIDGE Michael. Agent theories, architectures and languages: A survey. 1995.

- [54] SOO V-W y WU V-W. Negotiation without knowing other agents payoffs in the trusted third-party mediated-game. 2000.
- [55] ROSENSCHEIN J. y ZLOTKIN G. *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. MIT Press, 1994.
- [56] MICHALEWICZ Zbignew. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer - Verlag, 1994.

Apéndice A

METAMODELOS TRATADOS CON INGENIAS

A.1. Metamodelo de Agente

Se usa para describir agentes particulares sin entrar en la descripción de cómo es la interacción con otros agentes. El modelo se centra en la funcionalidad del agente, y el diseño de su control, de esta manera proporciona información sobre:

- Responsabilidades (funcionalidad): son las tareas o acciones que debe ejecutar y los objetivos que se compromete alcanzar.
- Para alcanzar un objetivo se debe desarrollar un conjunto de tareas, estas se determinan en INGENIAS como la funcionalidad (capacidad) y las propiedades (comportamiento), esto se encapsula en un rol al cual se le pueden asociar uno o varios estados mentales. El rol puede ser concedido a un actor.
- Dentro de los roles se encuentran los servicios definidos como un conjunto de funciones que se ofrecen a otros agentes.
- Comportamiento: en INGENIAS se toma el control como mecanismos que aseguran la ejecución de las tareas dentro de los parámetros acordados. Su entrada es un conjunto de datos que se denomina estado mental.

Los dos aspectos anteriores son importantes para la construcción de agentes ya que se centran individualmente en satisfacer necesidades para las que fueron diseñados.

El comportamiento del agente se desarrolla con base al tipo de control, a la especificación y a la evolución del estado mental.

Las responsabilidades del agente se desarrollan teniendo en cuenta la asociación de tareas, objetivos y los roles del agente.

El Meta-modelo debe tener en cuenta como expresar el estado mental y su evolución, este se representa mediante creencias, compromisos y deseos. Teniendo en cuenta un control que debe tener en cuenta las siguientes funciones:

- Establecer una manera de gestionar el estado mental creando, destruyendo o modificando las entidades que lo componen.
- Determinar el mecanismo de decisión que se maneja mapeando reglas.

A.1.1. Presentación del Modelo

Un agente es una entidad autónoma que se caracteriza por tener propósitos y una identidad única. Heredando de entidad única la capacidad de perseguir objetivos, que se satisfacen mediante la asociación con roles y tareas. En conclusión existen asociaciones entre el agente, sus objetivos y tareas desempeñadas. La primera asociación del agente con los objetivos, se hereda de la entidad autónoma. El meta modelo se explica en tres partes:

A.1.1.1. Control del Agente

Muestra como aparece el control al nivel conceptual y su relación con el estado mental del agente, conllevando a un paradigma en el cual un agente juega cierto rol que lo lleva a perseguir objetivos los cuales se pueden alcanzar mediante la modificación del ambiente por medio de las tareas, se identifican una modificación cuando ocurren evidencias, estas se gestionan mediante el gestor del estado mental y se procesan mediante el procesador de estado mental para producir decisiones en el agente.

- Los requisitos que debe satisfacer el gestor y el procesador se define bajo tres herramientas. Relaciones entre las tareas y entidades mentales: son instancias de la meta relación GTAfecta que representan como se afecta el estado mental del agente ante la ejecución de una tarea.
- Relaciones entre agentes y objetivos, sirve para colocar el objetivo y el agente en el contexto de la organización se hereda una de estas relaciones de la entidad autónoma y otras a través del estado mental. Se instancia mediante la relación WFPercigue que define el flujo de trabajo y justifica por que se ejecutan las tareas en la organización.
- Asociación de estados intermedios a agentes en ejecución: determina como se referencia las instancias de los agentes al nivel de meta modelo.

A.1.1.2. Agentes en Ejecución

Es necesarios para describir la evolución del estado mental en el Meta- modelo se introduce como tema de consulta en la entidad autónoma. además de ser utilizado para describir quien participa en las relaciones y que roles desempeña. La consulta de entidad autónoma se especializa en las consultas de propiedades de los agentes (en general) y para los agentes concretos.

Por cada asociación en la que participe la entidad de agente debe poder participar en la consulta del agente. Si el agente en ejecución es de tipo concreto, se utiliza la Meta- relación a instancia de, se puede hacer referencia a otras cualidades mediante la propiedad de descripción de agente.

La entidad de consulta autónoma se representa por un símbolo dentro de paréntesis y se etiqueta con el nombre del agente o con una expresión única de ser un agente concreto, si es para consulta de los agentes se etiqueta con una expresión que defina este conjunto.

A.1.1.3. Asociación de Responsabilidades

En esta parte los roles se usan para asignar responsabilidades ya que mediante estos se puede hacer una abstracción de un conjunto de funciones y así tener un estado, de la misma manera este debe ser llevado a cabo por una entidad no abstracta por ejemplo un agente.

La relación en el Meta modelo se realiza mediante el WFJuega, quien indica que el agente adquiere todos los objetivos, responsabilidades y capacidades asociadas a su rol.

A.1.1.4. Estado Mental

Se define como agregación de entidades mentales tales como objetivos, creencias y compromisos, adicionalmente los hechos y eventos. De esta forma se admitan múltiples instancias del estado mental relacionado con el agente.

El estado mental puede asociarse de dos maneras: una directa esta se realiza cuando se crea el agente, y este partirá de ese estado; la segunda es indirecta cuando define que el agente tiene estados intermedios. En el Meta-Modelo para descubrir la segunda clase de asociación se utilizan las entidades consulta de entidad autónoma. Para los estados Intermedios sugiere agregar toda la información adicional que se pueda acerca de lo se espera de las entidades mentales. En el Meta-modelo se define un estado mental condicionado. Las entidades de control son los objetivos y los compromisos. Los objetivos tienen asociado un estado que debe estar en una lista de posibles estados, que se satisfacen cuando sucede cierta evidencia. Si esta evidencia no se produce durante un estado prudencial, el objetivo debe estar atento capturando pruebas y evidencias para determinar en algún momento si la evidencia se produce. Lo anterior sugiere que un objetivo debe tener un ciclo de vida.

A.2. Metamodelos de Objetivos y Tareas

Tiene como objetivo definir las acciones identificadas en los modelos de organización, interacción y de agentes, para determinar cuales son las consecuencias y razones para ejecutar tales tareas.

Se asocian los objetivos con las tareas, ya que un agente tiende a enfocar sus acciones siguiendo un plan que se lleva acabo por medio de tareas, también ayudan a la evaluación del estado mental de sus responsables. El metamodelo maneja las tareas dándole dos enfoques, a saber:

- Las tareas como transformadoras del estado global del sistema en donde se conciben como pre-condiciones y post -condiciones.
 - Las pre-condiciones se identifican con la parte izquierda de una regla y cuyos términos son entidades de conocimiento.
 - Las post condiciones se consideran como la integración de tareas en los flujos de trabajo. Se modelan como la producción o modificación de las entidades mentales y como la utilización de elementos del entorno.
- Las tareas como procesos.

Para simplificar el metamodelo se restringe una tarea a lo que puede hacer omitiendo el estado de la misma.

Los objetivos se consideran en el metamodelo como entes auto representativos que guían el comportamiento del agente, se debe permitir que los objetivos se relacionen con el conjunto de elementos al cual representa.

A.2.1. Presentación del Metamodelo

Se sa para expresar la motivación que hay para realizar las tareas y que opciones de ejecución que se le presentan al agente en un momento determinado.

El Metamodelo refleja la relación de os objetivos con los agentes, roles y organizaciones.

- Una entidad autónoma al igual que un agente persigue objetivos (GTPersigue).
- Los roles se asocian con los objetivos mediante el flujo de trabajo, la asociación se hace mediante la relación WFPersigue.
- Los agentes se pueden asociar con los objetivos y los roles.
- Las tareas se asocian con los objetivos por medio de la meta - relación GTAfecta. En La ejecución de una tarea afecta a la entidad mental que se asocia con el objetivo.

- Si se decide modificar el estado mental de otros agentes mediante la tarea, este se modela haciendo que la tarea produzca una interacción con el otro agente.

Hay tres formas de modificar el estado mental: destruyéndolas (GTDestruye), modificándolas (GTModificada) y creándolas (GTCrea) , todas viene de un patrón de estado mental para indicar como sucede la modificación.

- La relación GTModifica se usa para los objetivos, como se satisfacen o no, para las metarelaciones GTSatisface y GTFallas respectivamente, se tiene en cuenta dos enfoques para decir que hacer cuando se satisface o fracasa la consecución del objetivo.
 - El primero delega esta función al gestor y procesador del estado mental.
 - El segundo define nuevas tareas para tratar con objetivos satisfechos o fracasados

En este modelo se han incorporado metarelaciones para la descomposición de tareas y objetivos cuando estos son muy complejos, WFDescompone y GTDescompone respectivamente. La descomposición de objetivos da lugar a dos nuevas relaciones.

- GTDepende dando por entendido que cuando los subobjetivos se cumplen el objetivo padre también, si falla un sub-objetivo el objetivo padre falla.
- GTDepende O: da por entendido que si un objetivo se alcanza, entonces el objetivo padre también, si todos los objetivos fallan el objetivo padre también lo hace.

Las precondiciones de las tareas se representan mediante WFConsume, Wfusa y GTAfecta y las postcondiciones por WFProduce y GTAfecta.

- WFConsume indica si se desea ejecutar la tarea deben existir determinadas entidades mentales.
- Wfusa señala que se necesitan algunos recursos del metamodelo del entorno.
- GTAfecta informa que se actúa sobre una entidad mental.
- WFProduce indica la creación de nuevas entidades mentales o recursos.
- GTAfecta señala cambios en las entidades mentales.

Las tareas no se ejecutan sino se satisfacen las precondiciones.

A.3. Metamodelo de Interacción

Las interacciones identifican las dependencias entre los componentes del sistema, también desarrollan especificaciones sobre el comportamiento de estos componentes y la funcionalidad de cada uno de ellos.

En INGENIAS las interacciones se estudian desde diferentes puntos de vista como son: la naturaleza de la interacción, la ejecución, la representación y el contexto, todas estas estudiadas en el diseño e integradas en el proceso de desarrollo.

Las interacciones además determinan el comportamiento de los agentes mostrando cual es su reacción cuando actúan sobre ellos. Ya que el comportamiento está definido en función de los objetivos del agente, se afirma que las interacciones están relacionadas con los objetivos y las tareas.

Cada una de las interacciones entre agentes debe contemplar lo siguiente:

- *Los actores que participan.* Un actor debe demostrar el por qué esta participando en la interacción.
- *La definición de unidades de interacción.* Una unidad que represente como debe procesarse la interacción en el receptor, por ejemplo paso de mensaje.
- *Orden sobre las unidades de interacción.* Las unidades de interacción se deben organizar siguiendo un protocolo estándar..
- *Acciones ejecutadas en la interacción.* Criterios para decidir el momento en que se debe ejecutar una tarea y cuales son las consecuencia de la ejecución de esta..
- *Definiciones del contexto de la interacción.* Consiste en detallar lo que ocurre en el sistema cuando la interacción se inicia, se desarrolla y concluye. Con este detalle de hecho, se proporciona información sobre los actores que participan, los motivos que impulsan a hacerlo, los objetivos que persigue la interacción y los estados mentales que se requieren durante la ejecución de la interacción.
- *Un modelo de control.* El control asegura que la interacción se desarrolle según los parámetros con las que fue definida.

Para comenzar con el análisis de las interacciones se debe identificar las que son relevantes para el funcionamiento del sistema, estas aparecen cuando se consideran las relaciones entre los componentes del sistema o cuando se trata explicar la ejecución de algunas tareas.

Apéndice B

JADE

El framework JADE (Java Agent DEvelopment) permite desarrollar SMA en conjunto con especificaciones FIPA. Han salido varias versiones desde su lanzamiento en 1999, la versión actual es la tercera que fue desarrollada por TILAb S.p.A al igual que las versiones liberadas en el 2001 y 2002.

La librería de desarrollo es un software gratuito que puede ser redistribuido o modificado bajo los términos de Licencia Pública GNU versión 2.1.

JADE es un software para permitir el desarrollo de SMA y especificaciones conforme el estándar FIPA para agentes inteligentes. El framework incluye dos productos: una plataforma de agente FIPA y varios paquetes para desarrollar agentes en Java. JADE está compuesto por los siguientes paquetes:

jade.core: implementa el kernel del sistema. Este incluye la clase Agente que puede ser extendida por las aplicaciones que se hagan. También está la clase Behaviour contenida en el sub-paquete *jade.core.behaviours* que implementa las tareas o intenciones del agente.

jade.content: contiene un conjunto de clases que soportan las definiciones del usuario en cuanto a las ontologías y a los lenguajes.

jade.lang.acl: este subpaquete contiene la especificación y funcionalidad del Lenguaje de Comunicación de Agentes ACL.

jade.domain: contiene todas las clases que representan al Agent Management, a los agentes AMS y DF.

jade.gui: este paquete contiene un conjunto de clases genéricas para crear y editar agentes con interfaz gráfica.

jade.mtp: contiene una interfaz Java para cada implementación del protocolo de transporte de mensajes, de tal manera que se pueda integrar con el framework JADE.

jade.proto: es el paquete que contiene clases para modelar los protocolos de interacción más comunes (fipa-request, fipa-query, fipa-contract-net, fipa-subscribe, entre otros descritos por FIPA).

jade.wrapper: contiene la envoltura de las funcionalidades de JADE que se pueden usar en aplicaciones externas.

El modelo de plataforma de agentes que tiene JADE, es el modelo estándar según las especificaciones de la FIPA, cuya estructura se puede ver en la figura B.1, y cuyos componentes son:

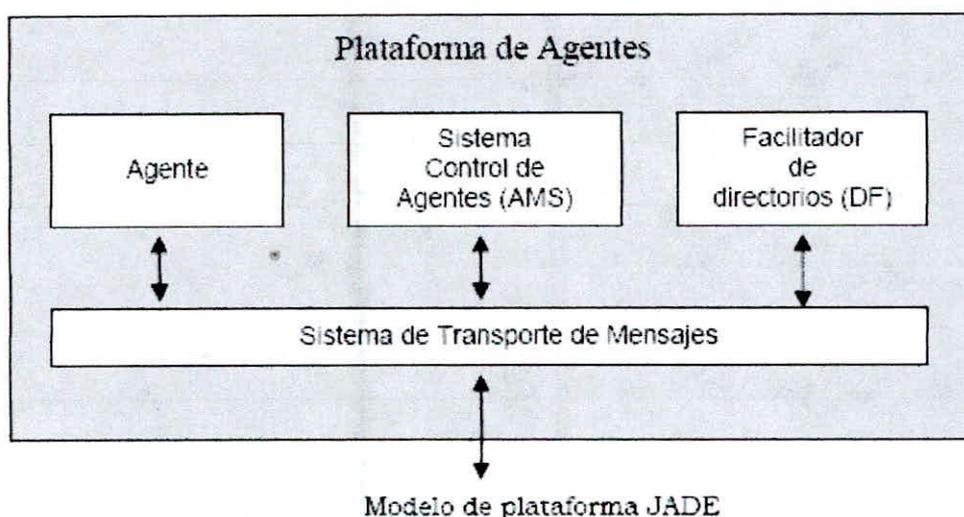


Figura B.1: Modelo de Plataforma de Agentes usado por JADE.

- El **Agent Management System (AMS)** : es el agente encargado de supervisar cualquier acceso y utilización de la plataforma de agentes. Tan solo existe un AMS en cada plataforma, y este es el encargado de proporcionar los servicios de directorio y de control del ciclo de vida, manteniendo un directorio de identificadores y de estados de agentes (AID). Cada agente debe registrarse en un AMS para obtener un identificador AID valido..
- El **Directory Facilitator (DF)** : es el agente que proporciona el servicio de directorio dentro de la plataforma..
- El **Message Transport System (MTS)** : también llamado Agent Communication Channel (ACC), es el componente software que controla todos los intercambios de mensajes dentro de la plataforma, incluyendo aquellos mensajes dirigidos o recibidos desde plataformas remotas.

JADE cumple totalmente con esta especificación de arquitectura, y cuando una plataforma JADE se ejecuta, el AMS y el DF se crean inmediatamente y se configura el módulo ACC para permitir la comunicación mediante mensajes. La plataforma puede ser distribuida en varios "hosts", aunque solo una aplicación Java y por tanto solo una JVM (Java Virtual Machine) se ejecuta en cada "host", así cada JVM es un "contenedor", que proporciona un entorno completo para la ejecución de agentes, permitiendo ejecuciones concurrentes dentro del mismo "host". El "contenedor" principal de la plataforma es aquel en el que reside el AMS y el DF, y al que se conectan el resto de contenedores creados.

B.1. Clase Agent

Desde el punto de vista del programador, un agente JADE es simplemente una instancia de una clase Java una extensión de la clase Agent. Esto le asigna la herencia de una serie de características destinadas a conseguir las interacciones básicas con la plataforma de agentes (registro, configuración, etc.) y un conjunto básico de métodos que implementa las tareas del agente. El modelo computacional de un agente JADE es un modelo multitarea en donde las tareas se ejecutan de forma concurrente.

B.2. Intercomunicación entre Agentes

La clase Agent también proporciona un conjunto de métodos para la intercomunicación entre agentes. Esta se produce mediante el paso de mensajes asíncronos en donde se envían objetos de la clase ACLMessage. El método Agent.send() es el que permite enviar un ACLMessage y el parámetro receiver del objeto ACLMessage es el que contiene la identificación del agente o agentes a quién será enviado dicho mensaje. Este método es totalmente transparente al lugar en donde reside el agente (ya sea local o remoto), y es la plataforma la que se encarga de seleccionar los mecanismos de transporte y direccionamiento apropiados.

B.3. Tareas de los Agentes (Behaviours)

Cuando se quiera implementar tareas específicas para los agentes, se deben definir una o más subclases Behaviour, instanciándolas y añadiendo los objetos creados a la lista de tareas del agente.

La clase Agent proporciona dos métodos:

addBehaviour(Behaviour) y ***removeBehaviour(Behaviour)***, que permiten gestionar la cola de tareas de un agente específico añadiendo o eliminando tareas de ella. Obsérvese que las tareas y subtareas se pueden añadir en el momento en que sean necesarias y no solo dentro del método Agent.setup().

El proceso de ejecución de una tarea comienza cuando el planificador da permiso para que se inicie, entonces se ejecuta el método `action()`. Cuando este método devuelve el control al planificador, ya sea porque el planificador "round-robin" se lo quite o porque ha terminado su trabajo, se ejecuta el método `done()` para chequear si la tarea se ha completado con éxito. Si es así, la tarea se elimina de la cola, sino se mantendrá a la espera hasta que el planificador le vuelva a dar permiso para continuar su ejecución.

Con el objeto de evitar una espera de mensajes activa y como consecuencia un desaprovechamiento del tiempo de la CPU, cada tarea puede bloquear su ejecución mediante el método `block()` que coloca a la tarea en una cola de tareas bloqueadas hasta que se reciba un mensaje¹.

¹Tomado del Proyecto @LIS -UTEM - Tutorial Jade