Jerarquía de agentes inteligentes para la asignación de tareas mediante aprendizaje y subastas

por

Carlos Abraham Solorio Navarro

Tesis sometida como requerimiento parcial para obtener el grado de

Maestro en Ciencias Computacionales

por el

Instituto Nacional de Astrofísica, Óptica y Electrónica

Febrero 2015

Tonantzintla, Puebla

Supervisor:

Dr. José Enrique Muñoz de Cote Flores Luna

Coordinación de Ciencias Computacionales

INAOE

Resumen

La asignación de tareas es un problema general en el que se busca una relación entre un conjunto de elementos, llamados *recursos*, para poder realizar un conjunto de objetivos llamados *tareas*. Los enfoques de las técnicas de asignación de tareas se clasifican, a grandes rasgos, en centralizados y distribuidos. Los enfoques centralizados tratan de resolver el problema desde un sólo nodo de procesamiento (virtual o real), es decir, nodos capaces de tener el acceso y control total a la información global del ambiente. En contraste, los enfoques distribuidos modelan al problema general en sub-problemas y se resuelven por su propio nodo de procesamiento de forma local, es decir, sólo con la información relativa al sub-problema. Los agentes inteligentes son entidades (sistemas computacionales o robots) capaces de tomar decisiones basándose en estímulos provenientes del ambiente y son usados típicamente para resolver problemas de manera distribuida.

Esta tesis presenta una jerarquía de agentes inteligentes que combina una adaptación de *subastas*, una técnica distribuida escalable muy común en la literatura, y una técnica de aprendizaje por refuerzo para aprender la dinámica de un ambiente desconocido en donde cada decisión puede tener consecuencias importantes a corto, mediano o largo plazo, debido a la llegada de nuevos recursos y tareas a lo largo del tiempo. Las subastas propuestas poseen un esquema de organización diferente a las subastas usadas en la literatura y han sido adaptadas a ambientes dinámicos. La técnica fue aplicada para su experimentación en un juego de estrategia en tiempo real, que es un ambiente dinámico complejo.

Abstract

Task allocation is a general problem where a relationship between a pair of sets, called *tasks* and *resources* is sought. Tipically, resources are assigned to tasks in order to complete them. Task allocation techniques follow, generally, two approaches: centralized and distributed. Centralized approaches try to solve the problem from the perspective of a single processing node (a physical or virtual one), that is, nodes capable of accessing and processing the environment's global information. Meanwhile, distributed approaches model the general problem as a collection of sub-problems and try to solve each one of them independently, in other words, each procressing node uses just the local information or information relative solely to the sub-problem. A common distributed approach is the use of intelligent agents, (software or hardware) entities capable of making choices based on external stimuli (environmental changes, for example).

This thesis introduces an intelligent agents hierarchy that combines an *auction* adaptation, an escalable distributed task allocation technique, and a reinforcement learning algorithm. Reinforcement learning is used to learn the dynamic of an unknown environment, where each choice may have dire consequences in short, medium and even long term, due to the arrival of new tasks and resources. The proposed auction adaptation possess a new organization scheme different from auctions in recent literature and have been tailored for dynamic environments. The proposed technique has been applied, for experimentation purposes, in a very complex dynamic real-time scenario: a real-time strategy game.

Agradecimientos

A todas las personas que me brindaron su apoyo, su confianza y especialmente su paciencia en este largo y duro proceso, lleno de altibajos en las diferentes etapas que lo comprendieron.

A mi madre y hermana - Lucía Silvia Navarro Rivera y Cynthia Karla Solorio Navarro - que, gracias a su incondicional respaldo, duro trabajo, fuerte sentido de responsabilidad, disciplina y ética a lo largo de los años, fui bendecido con una sólida educación, un conjunto irremplazable de valores morales y una invaluable formación académica.

A mi sobrino, Luka Sandoval Solorio, que es una de mis más grandes motivaciones para continuar con el trabajo y la pasión del estudio, pues él y la siguiente generación serán los testigos del mundo que la presente generación estamos construyendo.

A todas las personas que sin tapujos me han brindado su amistad que han estado disponibles para mí por cualquier medio para ofrecerme su apoyo sin importar el día o la hora, me demostraron que las verdaderas amistades son posibles y que una vez alcanzadas, son insustituibles. Todas estas personas marcaron el inicio de una etapa de crecimiento personal que me ayudó de forma inigualable en un momento de mi vida que considero fue especialmente difícil.

A Nayeli que, además de ofrecerme su amistad y una gran cantidad de su

tiempo, ha sido un modelo a seguir por su humildad, amabilidad, disposición, un intachable sentido de justicia y voluntad para hacer el bien público. Me ha mostrado un sin fin de perspectivas que ignoraba y que, sin duda, han reforzado y acelerado el proceso de desarrollo y madurez. Aunque, claramente, todavía queda un largo camino por recorrer. Estoy seguro que, sin importar lo que depare el futuro, su influencia habrá dejado una buena huella.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por su apoyo en el desarrollo de cada etapa de la maestría. A todas las personas que ayudaron a mejorar el trabajo gracias a sus observaciones, críticas y sugerencias, especialmente al doctor José Enrique Muñoz de Cote, que además de dedicarme muchas horas de su tiempo, de enseñarme muchas cosas que ignoraba, tuvo mucha paciencia y me respaldó en cada dificultad que se presentó. Por último, pero no menos importante, a todo el plantel de docentes del Instituto Nacional de Astrofísica, Óptica y Electrónica que tuve oportunidad de conocer y de los que aprendí, explícita o implícitamente valiosas lecciones sobre esfuerzo, pasión, creatividad y humanidad.

A todos, les extiendo mi más profundo y sincero agradecimiento. El entorno de toda una vida es el que forja el carácter de una persona y ustedes han sido elementos fundamentales de ese entorno.

Gracias.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

La asignación de tareas es un problema que surge de forma frecuente en la industria. Por ejemplo, en la farmacéutica, surge en la asignación de recursos (personal, maquinaría, compuestos químicos) al proceso de síntesis, con base a demandas regionales. En las industrias textil, automotriz, aeronáutica, etc. surge de la asignación de personal especializado o multidisciplinario (tejedores, deshebradores, mecánicos, ensambladores, supervisores, etc.) a las distintas etapas de producción. En la industria de transporte terrestre o aéreo, se presenta en la asignación de vehículos, itinerarios y rutas, etc. En general, la asignación de tareas es un problema que se presenta cuando existe alguna clase de restricción, como la escasez de recursos (en comparación con las tareas), el abaratamiento de costos en un proceso industrial, el cumplimiento de objetivos en plazos restringidos de tiempo, etc. El problema, en su definición más básica, consiste en encontrar la forma más eficiente de asignar *recursos* a *tareas*. Sin embargo, en el mundo real normalmente es necesario especificar detalles adicionales, como las capacidades específicas de cada uno de los recursos, los requerimientos específicos de cada tarea, la disponibilidad de los recursos y las tareas a través del tiempo, de qué manera deben ser elegidos los recursos para completar las tareas de la forma más eficiente, etc.

Típicamente, el enfoque para proponer alguna solución al problema de asignación de tareas depende principalmente de las facultades con las que disponga el usuario, es decir, la capacidad de procesamiento, almacenamiento y comunicación,

entre otras, que caracterice al equipo de cómputo con el que se cuente para resolver el problema. Otro factor importante que debe considerarse es la calidad deseada de las asignaciones de tareas. Debido a que el problema pertenece a la clase *NP-duro* [Yedidsion et al., 2011] (o *NP-completo* en su variación como problema de decisión [Garey and Johnson, 1990]) es común tener que ceder en la calidad de las soluciones para, en su lugar, obtener una solución en un tiempo razonable. En escenarios de la vida real, los elementos principales que afectan los requerimientos computacionales de un problema de asignación de tareas son:

- *El tamaño esperado del escenario*, es decir, la cantidad de recursos (como operadores humanos o máquinas, procesadores, unidades de transporte, maquinaría especializada, etc.) y la cantidad de tareas (operación de maquinaria, procesos informáticos, rutas de reparto, fabricación de materiales, etc.) máximos que pueden estar presentes simultáneamente o a lo largo del tiempo.

- *La dinámica del ambiente*, es decir, si los conjuntos de recursos o tareas cambian a través del tiempo. Se le llama *ambientes estáticos* cuando los recursos y tareas no sufren cambios a través del tiempo y *ambientes dinámicos* en el caso contrario. Dependiendo del problema, dentro de los ambientes dinámicos, es posible que puedan surgir patrones en la llegada de los recursos o las tareas, por lo que descubrir dicho patrón puede ser beneficioso para la asignación.

- *La semejanza entre tareas o entre recursos.* En otras palabras, si las tareas comparten requerimientos del mismo tipo entre sí. De igual forma, si los recursos comparten capacidades del mismo tipo entre sí. Cuando esto sucede, se dice que las tareas y los recursos son *homogéneos.* En otro caso, son *heterogéneos.*

- *El tipo de asignaciones* posibles en el problema. Es decir, si una tarea puede ser asignada por uno o más recursos simultáneamente o, en el caso contrario, si un recurso puede encargarse de una o más tareas simultáneamente. Cuando los recursos pueden ocuparse sólo de una tarea y las tareas sólo pueden asignarse

a un recurso, se dice que la asignación es *lineal*, en cualquier otro caso se le llama *asignación no lineal*.

Dependiendo del tamaño y la dinámica del ambiente, pueden elegirse soluciones *centralizadas* o *distribuidas*. La asignación *centralizada* de tareas supone que el conocimiento global del sistema es totalmente accesible desde un único nodo computacional [van der Horst and Noble, 2010]. Por el contrario, la asignación de tareas *distribuida* supone que múltiples nodos computacionales con sus propias capacidades de procesamiento, almacenamiento y comunicación (velocidad, ancho de banda, tasa de transferencia de datos entre nodos) tienen acceso restringido al conocimiento global del sistema. Si se busca asignar recursos a tareas, en donde el número de cualquiera de estos es grande, se suele usar este tipo de técnicas cuyo costo computacional no sufre grandes incrementos con relación al tamaño del ambiente, es decir, son *escalables*.

Los sistemas multiagente son sistemas complejos compuestos de múltiples agentes. El consenso en la literatura afirma que un agente es una entidad capaz de *percibir* su ambiente y *actuar* sobre él [Macarthur, 2011, Vidal, 2010, Vlassis, 2007, Wooldridge, 2009, Wooldridge and Jennings, 1995, van der Hoek and Wooldridge, 2008, Weiss, 2000]. Los múltiples agentes en un sistema pueden percibir y actuar en el ambiente de la misma manera, cuando esto ocurre se le llaman *agentes homogéneos* y en el caso contrario, *agentes heterogéneos*. Debido a la naturaleza de los sistemas multiagente, estos pueden ser utilizados para ofrecer soluciones distribuidas al problema de asignación de tareas. La investigación en el diseño de algoritmos en los sistemas multiagentes se ayuda de la teoría de juegos, economía y biología [Vidal, 2010, Wooldridge, 2009]. Además, también se complementa de la investigación de la inteligencia artificial, planeación, métodos de razonamiento, métodos de búsqueda, aprendizaje computacional, etc.

En las diferentes variaciones del problema de asignación de tareas surgen di-

versas preguntas. Por ejemplo, cuando se habla de una sola asignación, la pregunta común es "*¿qué tipos de recursos y cuántos recursos debo elegir para completar esta tarea de la mejor forma?*". En otras palabras, la cantidad y el tipo de recursos son las variables que influyen de forma más significativa en la calidad de dicha asignación. La ciencia ha tratado de contestar esto de múltiples formas (mediante programación lineal [Zimmermann, 1978], algoritmos evolutivos [Deb and Kalyanmoy, 2001], redes neuronales [Grossberg, 1988, Kosko, 1992], aprendizaje [Kaelbling et al., 1996, Pelikan et al., 2002], métodos iterativos [Marler and Arora, 2004], etc.). Por otro lado, cuando se habla de todo un conjunto de asignaciones en un instante de tiempo (o en ambientes estáticos), la pregunta que surge naturalmente es "*¿qué combinación de asignaciones hay que formar y en qué orden deben elegirse para llevar acabo todas estas tareas?*", "*¿es posible encontrar la solución más eficiente?*". Es decir, en este caso no sólo la consideración de la cantidad y el tipo de los recursos añaden complejidad al ambiente, también lo hacen las asignaciones por sí mismas. En ambientes dinámicos, nuevas preguntas se manifiestan: "*¿cómo cambiará el ambiente?*", "*¿qué decisión será la correcta en este momento?*", "*¿cómo influenciarán en el futuro mis decisiones?*". Estas preguntan han sido respondidas sólo de forma parcial debido a la complejidad del problema ([Cheng et al., 2013, Tsai et al., 2013, Kang et al., 2013]), en algunos casos dándole especial prioridad a averiguar cómo cambia el ambiente [Fogue et al., 2013, Pippin and Christensen, 2011, Pippin and Christensen, 2013]. Sin embargo, cuando se considera un ambiente dinámico, la complejidad del problema aumenta tanto que muchos autores imponen ciertas restricciones al ambiente para simplificar el problema y poder ofrecer soluciones en un tiempo razonable. Esto significa que sólo logran responder a algunas de las cuestiones planteadas aquí. Trabajos como [Celaya and Arronategui, 2013, Huang et al., 2013, Brutschy et al., 2014, Zhang et al., 2010] se centran en cómo tomar decisiones en un momento dado, mientras que otros trabajos también tratan de implementar medidas para enfrentar posibles cambios en el ambiente [Chapman et al., 2010, Tolmidis and Petrou, 2013, van der Horst and Noble, 2010] o en las propias tareas, a causa de las decisiones

tomadas [Schoenig and Pagnucco, 2011, Thomas and Williams, 2009, Tolmidis and Petrou, 2013], aunque relajan ciertas condiciones del ambiente, como la heterogeneidad, asignaciones lineales o el requerimiento de cierto conocimiento *a priori* del ambiente.

Esta tesis se centra en el estudio del problema de asignación de tareas en un ambiente *dinámico*, con tareas y recursos *heterogéneos*, con asignaciones no lineales (aunque restringidas a recursos que sólo puedan hacer una tarea) y además, tratamos de responder todas las preguntas planteadas.

Para tratar de responder estas cuestiones, estudiamos a los sistemas multiagentes compuestos de agentes autónomos, cuya conducta se basa tanto en sus preferencias, es decir, en el objetivo que el diseñador le ha asignado al agente, como en su propia experiencia. Utilizamos la formalización de los *procesos de decisión de Markov* (MDP) para representar el ambiente dinámico de forma que un agente pueda tomar decisiones sobre él, evaluar la calidad de sus decisiones y valorar cuán cerca (o lejos) está de cumplir sus objetivos. Esta representación incluye la definición de un estado; que es la descripción de cómo encuentra el ambiente en un momento dado, las acciones que tiene disponible el agente para interactuar con el ambiente, a qué estado puede llevarlo cada una de estas decisiones y la recompensa que puede obtener llegando a un estado al tomar una cierta decisión. Los múltiples cambios en el ambiente que se ven reflejados en el MDP son la llegada de recursos y tareas, la efectividad inmediata de las asignaciones y la efectividad de las asignaciones a mediano/largo plazo. La efectividad de una asignación puede medirse de diferentes formas, dependiendo de las necesidades del problema a resolver. A mediano/largo plazo, los patrones que pueden surgir en la llegada de tareas y/o recursos son un factor adicional que debe tomarse en cuenta en la toma de decisiones pues influye en la efectividad de las asignaciones.

Cuando la complejidad de un problema crece, un enfoque utilizado es la divi-

sión recursiva del problema principal en sub-problemas cada vez más pequeños hasta que sean lo suficientemente pequeños para ser resueltos en una cantidad razonable de tiempo. Aquí proponemos el uso de esta división del problema y su posterior asignación a múltiples agentes, posiblemente diferentes, en un sistema multiagentes. A este enfoque se le llama *jerarquía multiagente* [Vidal, 2010]. En el contexto del problema de asignación de tareas, las jerarquías multiagente y, en general, la subdivisión de problemas es sumamente útil debido a su reducida complejidad y es utilizada por varios autores [Huang et al., 2013, Tolmidis and Petrou, 2013, Zhang et al., 2010] de distintas formas, incluido nuestro trabajo por la misma razón.

Es importante considerar que en un sistema multiagente el comportamiento de cada agente puede generar conflictos con otros agentes. Esto puede deberse a que cada agente ha sido diseñado para cumplir objetivos diferentes, por lo que las decisiones que cada uno de ellos toman pueden perjudicar directa o indirectamente las decisiones de otros agentes. Por esta razón, es necesario aplicar un esquema de cooperación y coordinación que sincronice las acciones de cada agente, de manera que trabajen cooperativa y eficientemente para resolver un problema específico en conjunto. En este trabajo hemos implementado un esquema de coordinación inspirado en la teoría económica, también llamado un paradigma basado en mercados. La teoría económica orientada a los sistemas multiagentes se centra en los llamados agentes *egoístas* [Wooldridge, 2009], es decir, aquellos que actúan únicamente para maximizar su propia ganancia o sólo consideran sus propias preferencias [Muñoz de Cote, 2008, Weiss, 2000]. Las técnicas inspiradas en mercados más utilizadas son las subastas. Las subastas aplicadas en los sistemas multiagentes funcionan de manera similar a las subastas en la vida real. El *subastador* ofrece un *conjunto de artículos* en una *etapa de anuncio* y establece algún tipo de mecanismo que le permita elegir las *ofertas* que los *postores* u *ofertantes* le proporcionan y que más le convengan. La oferta representa el interés que tienen los postores por los artículos en los que ofertan. En el contexto de la asignación de tareas, un artículo puede representar

tanto un *recurso* como una *tarea* y, por lo tanto, las ofertas representan la utilidad o el costo asociado a completar esa tarea o a utilizar ese recurso. Finalmente, una vez que todas las ofertas se han recibido, el subastador elige qué artículos se darán y a quiénes.

Las subastas son ampliamente utilizadas por su simplicidad, libertad de implementación, escalabilidad y descentralización. Esto le permite al diseñador elegir a los ganadores de la subasta con base, no sólo en las preferencias de los agentes, sino también en lo requerido para llegar a una meta común [Dias et al., 2006]. En ambientes *dinámicos*, el funcionamiento de las subastas requiere modificaciones específicas del ambiente, que le ayuden a lidiar con la incertidumbre procedente de la llegada de nuevos artículos y postores al mercado. Algunas variaciones de subastas que buscan enfrentar los problemas de estar en un ambiente dinámico consisten en aplicar métodos de aprendizaje o de optimización en la formulación de las ofertas [Pippin and Christensen, 2013], pero se aumentan los requerimientos de cómputo por agente y el ambiente no puede ser demasiado grande. Otras técnicas toman en cuenta el tiempo para elegir ganadores o cancelar asignaciones [Nanjanath and Gini, 2010], usan esquemas diferentes para el envío de las ofertas [Schoenig and Pagnucco, 2011], o simplemente se basan en cálculos específicos que dependen del ambiente para formular la oferta de los postores [Thomas and Williams, 2009]. Sin embargo, estos últimos enfoques, aunque rápidos y escalables, proporcionan soluciones que no son óptimas, ni se acercan a serlo.

Nuestra investigación presenta una jerarquía con dos tipos de agentes que perciben y manipulan al ambiente de forma muy diferente entre ellos. Los agentes de un nivel inferior de la jerarquía (que llamaremos *subordinados*) perciben al ambiente como un conjunto de subastas que se anuncian secuencialmente, es decir, una tras otra a criterio del subastador (el agente en el nivel superior) en ciertos intervalos de tiempo. Los postores ofertan por un artículo que representa una tarea o conjuntos de tareas. Por otro lado, los agentes de un nivel superior (que llamaremos *sofisti-*

cados o *líderes*) perciben al ambiente como un conjunto de artículos y un conjunto de posibles compradores. El objetivo de los agentes sofisticados es elegir el artículo más conveniente para abrir una subasta (esto es, una asignación) *en ese momento* y que les dará las mejores ganancias a mediano/largo plazo. Como resultado del trabajo, ofrecemos evidencia de que nuestro enfoque mejora las soluciones obtenidas por otras técnicas basadas en el mercado, en específico las subastas secuenciales [Schoenig and Pagnucco, 2011] que, a pesar de haber sido diseñadas pensando en los ambientes estáticos, también son aptas para ambientes dinámicos y en ambientes con recursos y tareas heterogéneas. Nuestra técnica, además, retiene la escalabilidad de las técnicas distribuidas.

1.1 Objetivo general

- Desarrollar una técnica que sea competitiva frente al estado del arte y sea escalable, para obtener la solución al problema de asignación de tareas en ambientes dinámicos, con tareas y recursos heterogéneos.

1.1.1 Objetivos específicos

- Diseñar funciones de oferta que representen, de la forma más fiel posible, las capacidades de los agentes para la realización de alguna tarea y aplicarla en el mecanismo de subastas.
- Diseñar una jerarquía de agentes inteligentes.
- Simular la técnica propuesta de asignación de tareas en un ambiente dinámico en tiempo real.

1.2 CONTRIBUCIONES

Las contribuciones principales del trabajo son:

- *Dos nuevos tipos de subastas, diseñadas específicamente para ambientes dinámicos. De forma concreta:*
  - *Subastas paralelo-secuenciales (sección 4.4.1). A diferencia de las subastas más utilizadas en estos ambientes (subastas paralelas descritas en la sección 2.3) en donde todos los artículos (en este caso, tareas) se anuncian al mismo tiempo y los n ganadores de los n artículos son seleccionados en el mismo instante de tiempo, las subastas paralelo-secuenciales anuncian un artículo en un instante de tiempo, se aceptan ofertas y se elige a un solo ganador. Posteriormente, se realiza una breve pausa en el anuncio de artículos para que los postores puedan, aunque sea momentáneamente, observar la consecuencia inmediata de la asignación anterior en el mercado. Finalmente se anuncia un nuevo artículo.*
  - *Subastas secuenciales inversas (sección 4.4.2). Complementando al esquema de espera mencionado en las subastas paralelo-secuenciales, esta subasta se inspira en las subastas secuenciales (sección 2.3). Sin embargo, se invierte el rol de recursos y tareas en la subasta: los recursos actúan ahora como subastadores y las tareas como postores. Esto permite que puedan realizarse asignaciones de múltiples recursos a una tarea. Si bien aumenta el número de ofertas que el subastador tiene que recibir, también mejora la calidad de las asignaciones por este mismo motivo, tomando en cuenta que un equipo de agentes puede realizar una tarea más eficientemente que considerando sólo agentes individuales. De particular interés es que este enfoque evita un problema combinatorio.*
- *Un modelo jerárquico de agentes inteligentes para aprender la dinámica de un*

ambiente que cambia a través del tiempo conservando la escalabilidad de una técnica distribuida (sección 4.1).

- *Una representación del ambiente compacta que reduce la complejidad de la exploración en el proceso de aprendizaje (sección 5.2).*
- *Una implementación de la técnica propuesta en un* ***ambiente realista altamente complejo en tiempo real****.*

1.3 ORGANIZACIÓN

Este trabajo está organizado de la siguiente forma: en el capítulo 2 se describe a detalle las bases teóricas sobre las cuales descansa la técnica propuesta. En particular, explicamos el marco general de un sistema multiagente y las influencias de la teoría económica. Posteriormente, se detalla el formalismo en el que descansan las subastas. Finalizamos con la formalización del proceso de aprendizaje para un solo agente y mencionamos los diferentes enfoques en el aprendizaje multiagente. En el capítulo 3 se hace una revisión del trabajo relacionado o la vanguardia del área, analizando las técnicas que cubren los distintos casos de la clasificación en la asignación de tareas, así como las áreas de aplicación en la que yacen los alcances del trabajo. Se hace especial enfoque en las técnicas de asignación de tareas que utilizan paradigmas basados en mercados. En el capítulo 4 se describe detalladamente y se formaliza la técnica propuesta, además se hace mención de todas las consideraciones especiales requeridas para el funcionamiento correcto de la técnica. En el capítulo 5 se describe el ambiente de experimentación, se explica el proceso experimental y las métricas con las que analizamos los resultados. Además, detallamos todos los parámetros, funciones específicas y la configuración del ambiente en cada experimento. Finalmente, mostramos los resultados de los experimentos realizados y una interpretación de los mismos. Para concluir, en el capítulo 6 se recaban las posibi-

lidades, limitaciones y el conocimiento obtenido al realizar este trabajo, además de indicar algunas posibles metas a futuro.

Capítulo 2

Marco teórico

En este capítulo, se definirán los conceptos necesarios para el desarrollo de la técnica propuesta en este trabajo. Gran parte de este capítulo se basa en [Vidal, 2010, Shoham and Leyton-Brown, 2008, Watkins and Dayan, 1992, Muñoz de Cote, 2008, Wooldridge and Jennings, 1995, Wooldridge, 2009, Vlassis, 2007, Weiss, 2000].

2.1 Sistemas multiagente

Para definir lo que es un *sistema multiagente* es necesario definir a un *agente*. Las definiciones son variadas, como puede verse en [Murugesan, 1998, Franklin and Graesser, 1997, Russell and Norvig, 2009, Wooldridge and Jennings, 1995]. Formulando una definición consensuada, un agente es una entidad *autónoma* que obtiene información del *ambiente* en el que se encuentra y toma *decisiones* sobre el mismo con el fin de completar alguna *meta* o actuar en *beneficio* de algo. Las propiedades que describen a un agente, de la forma más general [Wooldridge and Jennings, 1995], son las siguientes:

- Autonomía: los agentes operan sin la intervención directa de humanos u otras entidades, y tienen algún tipo de control sobre sus acciones y su estado interno.
- Habilidad social: los agentes pueden interaccionar (o no) con otros agentes mediante alguna clase de comunicación.

- Reactividad: los agentes perciben su ambiente y responden de forma acorde a los cambios que ocurren en él.

- Pro-actividad: los agentes no simplemente actúan en respuesta a su ambiente, también son capaces de exhibir conductas relacionadas a una meta, al *tomar iniciativa*.

Múltiples autores, a su vez, definen a los sistemas multiagente como sistemas que poseen un conjunto de agentes que pueden o no interactuar entre ellos [Wooldridge, 2009, Vlassis, 2007, Macarthur, 2011]. Los sistemas multiagente que aquí estudiaremos son sistemas complejos compuestos de agentes autónomos que, aunque funcionan sólo con conocimiento local (adquirido por ellos mismos) y disponen de capacidades limitadas, son capaces de realizar comportamientos globales deseados [Vidal, 2010, Vlassis, 2007, Wooldridge, 2009, Macarthur, 2011].

El criterio de los agentes para tomar decisiones se basa fundamentalmente en la noción de *preferencia*. Esto es, la inclinación que tiene cada agente por ejecutar una acción sobre otra, ya sea porque así fue diseñado o porque así lo ha aprendido del ambiente. En la siguiente sección, hablaremos acerca de la forma más común de cuantificar esta preferencia.

2.1.1 PREFERENCIAS DE UN AGENTE

Una suposición que simplifica el concepto de *preferencia de un agente* es que esta se encuentra representada en una *función de utilidad* [Vidal, 2010]. De forma general, esta función relaciona los estados del ambiente en el que se encuentra un agente a un número real que representa su preferencia. Los estados se definen como los estados del mundo que el agente puede percibir. Formalmente, sea S el conjunto de estados percibidos por el agente i, su función de utilidad es de la forma

$$u_i : S \to \mathbb{R} \tag{2.1}$$

Por ejemplo, si un robot posee un sensor que le suministra información binaria, o bien sólo dos posibles valores sobre el estado de iluminación del lugar donde se encuentra, entonces su función de utilidad estaría definida sólo en esos dos estados, indiferentemente de cuán complejo sea realmente el ambiente en donde esté. En resumen, las funciones de utilidad son una forma breve de representar los objetivos de un agente, la cual a su vez dicta su *conducta*.

Dada la función de utilidad de un agente, podemos definir un *ordenamiento de preferencia* sobre los estados del mundo. Al comparar los valores de utilidad de dos estados, es posible determinar cuál de estos es el que prefiere el agente. Formalmente, este ordenamiento posee las siguientes propiedades:

- Reflexivo: $u_i(s) \geq u_i(s)$
- Transitivo: Si $u_i(a) \geq u_i(b)$ y $u_i(b) \geq u_i(c)$, entonces $u_i(a) \geq u_i(c)$.
- Comparable: $\forall a, b$, $u_i(a) \geq u_i(b)$ o $u_i(b) \geq u_i(a)$.

De forma intuitiva, debe ser posible comparar de forma inequívoca dos utilidades, establecer cuál es mayor, menor o igual que otra y que éste ordenamiento pueda aplicarse a las utilidades correspondientes a cualquier estado percibido por un agente.

Una vez definida una función de utilidad para todos los agentes en el sistema, estos pueden tomar decisiones que maximicen su utilidad. Nos referiremos como *egoístas* a los agentes autónomos cuyo objetivo es tomar decisiones que maximicen su propia utilidad.

La *utilidad esperada* considera la incertidumbre ocasionada por el funcionamiento imperfecto de los medios por los cuales el agente recolecta información (sus

sensores, por ejemplo) o por los cuales este participa en el ambiente. Si suponemos que el agente conoce la probabilidad de alcanzar un estado s' al tomar una decisión o *acción* a desde un estado s entonces podemos calcular la utilidad esperada cuando un agente elige a en el estado s. La probabilidad de llegar a un estado s' desde un estado s mediante una acción a está dada por la *función de transición* y se denota por $T(s, a, s')$. Formalmente, la utilidad esperada se define como

$$E\left[u_i, s, a\right] = \sum_{s' \in S} T(s, a, s') u_i\left(s'\right) \tag{2.2}$$

donde S es el conjunto de todos los posibles estados. La utilidad esperada y la función de transición son herramientas útiles que auxilian en el desarrollo de la formulación matemática necesaria para encontrar la conducta óptima que conducirá a los agentes a maximizar su utilidad. Antes de que el agente intente buscar las mejores decisiones, primero es necesario definir el marco sobre el cual el agente pueda responder las siguientes preguntas: "*¿en qué situación me encuentro?, ¿estoy más cerca o más lejos de mi objetivo?, ¿qué puedo hacer ahora?*". Dicho marco se presenta en la siguiente sección.

2.1.2 PROCESO DE DECISIÓN DE MARKOV

Un proceso de decisión de Markov (MDP por sus siglas en inglés) es un modelo de toma de decisiones en un mundo dinámico con incertidumbre. El agente comienza en algún estado s del mundo, elige una acción a y recibe alguna recompensa inmediata. Posteriormente, este efectúa una transición probabilística hacia algún otro estado y el proceso se repite.

Definiremos un MDP a continuación.

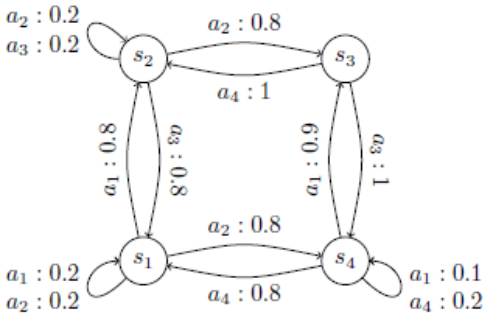
Definición 1. ***(Proceso de Decisión de Markov (MDP))*** *Un MDP es una tupla* $\langle S, A, T, r \rangle$*, donde*

- *S es el conjunto de estados.*
- *A es el conjunto de acciones.*
- *$T : S \times A \times S \to [0, 1] \in \mathbb{R}$ es una función que específica la probabilidad de transición entre estados: $p(s, a, s')$ es la probabilidad de terminar en un estado s' al elegir la acción a en el estado s.*
- *$r : S \times A \to \mathbb{R}$ es la función que proporciona la recompensa inmediata por haber llegado a algún estado $s \in S$ al tomar alguna acción $a \in A$.[Weiss, 2000].*

En un mundo totalmente *determinista*, la función de transición para todo estado s' sería 0, a excepción de uno, para el cual sería 1. Esto significa que en un mundo determinista, la acción del agente tiene un efecto completamente predecible. Para un mundo *no determinista*, este no es el caso, pues la acción del agente lo podría llevar a una variedad de estados, dependiendo del valor de la función de transición. En la figura 2.1 se muestra una representación visual típica de un *MDP*. En esta figura se presenta un MDP con un grafo de cuatro vértices, su función de recompensa r y su función de transición. Las aristas del MDP representan la posibilidad de llegar de un estado a otro. La función r está definida por la acción a que eligió desde un estado anterior para llegar a s. La función de transición T (definida en la sección anterior) muestra la probabilidad de llegar a un estado s_j, tomando una acción a desde un estado s_i.

La conducta de un agente es representada por una *política*, que es una función que mapea estados a acciones. La meta del agente, por lo tanto, es la de encontrar su *política óptima* que es la que maximiza su utilidad esperada, como se espera de un agente *egoísta*.

Esta estrategia se conoce como el principio de la *máxima utilidad esperada*. Recordando la ecuación 2.2, la política óptima de un agente i se puede definir como



(a) MDP con 4 estados.

s	$r(s)$
s_1	0
s_2	0
s_3	1
s_4	0

s_i	a	s_j	$T(s_i, a, s_j)$
s_1	a_1	s_1	0.2
s_1	a_1	s_2	0.8
s_1	a_2	s_1	0.2
s_1	a_2	s_4	0.8
s_2	a_2	s_2	0.2
s_2	a_2	s_3	0.8
s_2	a_3	s_2	0.2
s_2	a_3	s_1	0.8
s_3	a_4	s_2	1
s_3	a_3	s_4	1
s_4	a_1	s_4	0.1
s_4	a_1	s_3	0.9
s_4	a_4	s_4	0.2
s_4	a_4	s_1	0.8

(b) Función de transición y recompensa del *MDP*.

Figura 2.1: Representación visual de un *MDP* [Vidal, 2010].

$$\pi_i^* (s) = \arg\max_{a \in A} E [u_i, s, a] \tag{2.3}$$

Para poder expandir el valor esperado en esa ecuación, se debe determinar cómo lidiar con las recompensas futuras. Para hacerlo, hay que sopesar entre darle más importancia a una sola recompensa alta - esperando que llegue más temprano que tarde - o darle más importancia a cada una de las pequeñas recompensas que se podrían obtener en el camino. Dado que es incierto cuánto hay que esperar para toparse con una recompensa alta, se usan *recompensas descontadas*, que nos permiten reducir el impacto de recompensas que están muy alejadas en el futuro. Esto se logra multiplicando las futuras recompensas del agente por un *factor de descuento*, un número entre 0 y 1 representado por γ.

Si se tiene un agente con una política π, que inicia en un estado s_1, y visita los estados $s_2, s_3, ...$, se dice que su recompensa descontada está dada por

$$\gamma^0 r\left(s_1, a_1\right) + \gamma^1 r\left(s_2, a_2\right) + \gamma^2 r\left(s_3, a_3\right) + \ldots \tag{2.4}$$

Sin embargo, sólo se conoce hasta el momento s_1. El resto de estados dependen de la función de transición T. Es decir, desde el estado s_1, conocemos que la probabilidad de llegar a cualquier otro estado s' dado que tomemos la acción a es $T\left(s_1, a, s'\right)$ pero no sabemos a qué estado específico el agente llegará. Si asumimos que el agente maximiza su utilidad, entonces sabemos que tomará la acción que maximice su utilidad esperada. Así, si expandimos la ecuación 2.3 usando la ecuación 2.2, tenemos:

$$\pi_i^*\left(s\right) = \arg\max_{a \in A} \sum_{s' \in S} T(s, a, s') u_i\left(s'\right) \tag{2.5}$$

donde $u\left(s'\right)$ es la utilidad que esperamos que el agente obtenga al llegar al estado s' y después continuamos obteniendo las recompensas de los estados sucesivos mientras usamos la política π_i^*.

Sabemos que cuando el agente i alcanza s eligiendo a, recibe una recompensa inmediata $r\left(s, a\right)$ y que estando en s, puede elegir su acción basándose en $\pi_i^*\left(s\right)$ y obtener una nueva recompensa descontada por γ. De tal forma, podemos obtener la utilidad real que un agente recibe por estar en un estado s como

$$u\left(s\right) = \max_{a \in A} \left\{ r\left(s, a\right) + \gamma \sum_{s' \in S} T(s, a, s') u_i\left(s'\right) \right\} \tag{2.6}$$

La ecuación 2.6 se conoce como la *ecuación de Bellman*. Esta captura el hecho de que la utilidad de un agente depende, no sólo de sus recompensas inmediatas, sino también de sus futuras recompensas descontadas.

2.1.3 APRENDIZAJE POR REFUERZO

Es común encontrarse en la situación de carecer el conocimiento necesario sobre el ambiente cuando se diseña un sistema multiagente, por lo que el aprendizaje es la única manera de que el agente pueda adquirir lo que necesita saber sobre el ambiente y su dinámica. El aprendizaje por refuerzo sirve para que los agentes aprendan cuando no tienen a su disposición ejemplos válidos de los que puedan aprender, cuando no tienen un modelo de ambiente o cuando no poseen una función de utilidad (ecuación 2.1). Por ejemplo, en un juego de ajedrez donde el agente no tenga conocimiento previo del juego, la única forma de aprender es probando movimientos al azar. Pero sin alguna clase de retro-alimentación sobre lo que es bueno o malo, el agente no tiene bases para decidir qué movimientos son buenos y cuáles son malos. Afortunadamente, esta retro-alimentación en un juego de ajedrez se da cuando el juego termina. En otras palabras, el agente puede saber si lo que hizo estuvo bien o mal, con base en si ganó, perdió o empató el juego. Este tipo de retro-alimentación se llama *recompensa* o *refuerzo*. La tarea del aprendizaje por refuerzo es la de usar estas recompensas para que un agente aprenda a comportarse efectivamente en un ambiente. Los agentes pueden ser aprendices pasivos o activos. Los aprendices pasivos simplemente observan el ambiente y tratan de aprender la utilidad de estar en varios estados. Por otro lado, los aprendices activos deben también *actuar* usando la información aprendida. [Russell and Norvig, 2009]

Q-LEARNING

Esta técnica es una forma de aprendizaje por refuerzo libre de modelo en un MDP [Watkins and Dayan, 1992]. Puede verse también como programación dinámica asíncrona. Este método aprende cuáles son las mejores acciones probando todas las acciones en todos los estados repetidamente considerando las recompensas descontadas a largo plazo. Basándose en la ecuación de Bellman (2.6), *Q-learning* garantiza

aprender una política óptima π^* sin el conocimiento inicial de la función de transición $T(s, a, s')$.

Para una política π, se definen los valores Q o valores de acción como

$$Q^{\pi}(s,a) = r(s,a) + \gamma \sum_{s' \in S} T(s, \pi(s), s') u_i(s') \tag{2.7}$$

En otras palabras, el valor Q es la recompensa descontada esperada por ejecutar una acción a en el estado s siguiendo una política π. El objetivo de esta técnica es estimar los valores Q para una política óptima.

En Q-learning, la experiencia del agente consiste en una secuencia de distintas etapas o *episodios*. En el n-ésimo episodio, el agente:

- Observa su estado actual s_n.
- Elige y realiza una acción a_n.
- Observa el estado siguiente s'_n.
- Recibe una recompensa inmediata r_n.
- Ajusta sus Q_{n-1} valores usando un factor de aprendizaje α_n.

Finalmente, el valor Q en un estado s, eligiendo una acción a está dado por:

$$Q_n(s,a) = \begin{cases} (1-\alpha) Q_{n-1}(s,a) + \alpha [r_n + \gamma V_{n-1}(s'_n)] & \text{si } s = s_n \text{ y } a = a_n \\ Q_{n-1}(s,a) & \text{de otro modo} \end{cases} \tag{2.8}$$

donde

$$V_{n-1}(s'_n) = \max_b \{Q_{n-1}(s'_n, b)\} \tag{2.9}$$

La ecuación 2.9 se refiere a lo mejor que el agente cree que puede hacer desde el estado s'_n en el episodio n. Obviamente, en los primeros episodios del aprendizaje, los valores Q no reflejan precisamente la política óptima que intenta definir implícitamente. Los valores Q iniciales $Q_0(s, a)$ para todos los estados y acciones se asumen dados.

Complejidad del aprendizaje por refuerzo

La medición de la complejidad en las técnicas de aprendizaje por refuerzo se basa fundamentalmente en la medición de la *ejecución de acciones*, y éstas dependen del número de estados. A diferencia de otros métodos de búsqueda donde se tienen nociones del ambiente, las aplicaciones de los algoritmos de aprendizaje por refuerzo se enfocan muchas veces en un ambiente donde no se tiene conocimiento previo alguno, por lo que la única manera de aprender algo sobre él es *explorando*, es decir ejecutar acciones y observar los efectos. En una variedad de problemas donde se aplica aprendizaje por refuerzo, se sabe que el número de ejecuciones de acciones requeridas para alcanzar un objetivo es *exponencial* sobre el tamaño del espacio de estados [Koenig and Simmons, 1993]. Sin embargo, la ejecución se puede limitar a un polinomio aplicando un cambio en la representación de dicho espacio. Es más, si se logra representar un problema de aprendizaje por refuerzo de manera que en el espacio de estados no haya acciones duplicadas, esto es, que no haya acciones cuya transición desde un estado y recompensa sean equivalentes, la complejidad en el peor de los casos está limitada por $\mathcal{O}(n^3)$, donde n denota el número de estados. [Koenig and Simmons, 1993].

EXPLORACIÓN VS. EXPLOTACIÓN

Un problema importante del aprendizaje por refuerzo es la concesión entre explorar el ambiente y explotar el conocimiento que se adquiere [Kaelbling et al., 1996]. El problema surge cuando un agente se enfrenta a la decisión de explotar la información que ha aprendido, pero debe continuar explorando para garantizar que está actuando de manera óptima. En situaciones de aprendizaje multiagente, esta concesión se vuelve incluso más delicada [Claus and Boutilier, 1998]. Estas concesiones se basan comúnmente en la estimación actual de la función de valor y algún componente probabilístico que ocasionalmente selecciona acciones sub-óptimas. Uno de los métodos de exploración más usados en el aprendizaje por refuerzo es ϵ-*greedy*, en donde la mejor acción es elegida con probabilidad $1 - \epsilon$ y con probabilidad ϵ se elige una acción al azar (exploratoria). Inicialmente en el proceso de aprendizaje, la exploración se realiza frecuentemente y con el tiempo esta va disminuyendo. Aunque esta es la estrategia de exploración más común, tiene sus inconvenientes. Es sabido que la exploración $\epsilon - greedy$ puede aprender a un ritmo muy lento en algunos ambientes [Strehl, 2007, Thrun, 1992]. Además, al explorar selecciones equitativamente entre todas las acciones, es igualmente probable que se seleccione tanto la peor como la mejor.

2.1.4 APRENDIZAJE MULTIAGENTE

En la sección anterior de aprendizaje por refuerzo se hace la suposición que *un solo* agente se encuentra manipulando el ambiente. Pero este modelo es insuficiente cuando se quiere modelar un sistema multiagente, por lo que es necesario realizar modificaciones al formalismo base, el proceso de decisión de Markov (MDP) definido en la sección 2.1.2. Posteriormente a la modificación del formalismo, hay trabajos que sugieren principios de diseño para mejorar el rendimiento de estos algoritmos de aprendizaje multiagente.

Hay varias maneras de transformar un MDP a un MDP multiagente. La manera más fácil es que cada agente ignore totalmente la existencia de otros agentes. La segunda manera más sencilla consiste en colocar los efectos del resto de agentes en la función de transición. Es decir, asumir que los demás agentes no existen realmente como entidades sino como partes del ambiente. Esta técnica puede funcionar para casos sencillos donde los agentes no estén cambiando su conducta debido a que la función de transición en un MDP debe ser fija. Desafortunadamente es muy común que los agentes cambien sus políticas a través del tiempo, ya sea por su propio aprendizaje o por la intervención del usuario.

Un mejor método es extender la definición del MDP para incluir múltiples agentes, todos capaces de tomar decisiones en cada instante de tiempo. Así, en lugar de tener una función de transición $T(s, a, s')$, tendríamos una función de transición $T(s, \overrightarrow{a}, s')$, donde $\overrightarrow{a}$ es un vector del mismo tamaño que el número de agentes, donde cada elemento es la acción (o un símbolo que represente la inacción) de un agente. También es necesario redefinir la función de recompensa de manera que cada agente reciba una cantidad proporcional a su contribución.

En este trabajo, jes se tomó la decisión de que los agentes percibieran a los demás agentes como parte del ambiente, por lo que no ahondaremos más en esta sección.

2.2 TEORÍA DE JUEGOS

En los sistemas multiagentes que aquí estudiamos, se tiene un conjunto de agentes autónomos en los que cada uno realiza sus propias acciones usando la información que tengan disponible, cualquiera que sea. Debido a que otros agentes también están realizando acciones, cada uno debe tomar en consideración estas acciones para decidir qué hacer para maximizar su propia utilidad. De esta manera, lo que uno haga depende de lo que otros hagan y viceversa. El agente debe decidir

qué hacer cuando su decisión depende de la decisión de otros. Este tipo de problemas son muy comunes en diferentes campos, desde la *economía*, ciencia política, psicología, la biología hasta las *ciencias de la computación* [Vidal, 2010, Shoham and Leyton-Brown, 2008]. Por este motivo, un conjunto de herramientas matemáticas ha sido desarrollado y refinado a través de los años para modelar y resolver estos problemas. Este conjunto es conocido como teoría de juegos.

Es típico clasificar estos juegos como *juegos cooperativos* y *juegos no cooperativos*. Los juegos no cooperativos son aquellos en los que el conjunto de acciones preferibles de los agentes puede estar en conflicto con las de otros. Es decir, una decisión que sea buena para un agente no necesariamente tiene que repercutir positivamente en otro. Hay que remarcar que el hecho de que sean juegos no-cooperativos no significa que la cooperación no pueda existir. Por otro lado, los juegos cooperativos, o también llamados *juego de coaliciones*, se denominan así debido a que los agentes deben cooperar entre ellos para formar coaliciones y un agente no puede decidir unilateralmente unirse con otros agentes. Estos deben buscar formas para formar grupos o coaliciones entre ellos y cada coalición recibe alguna utilidad. El problema que enfrentan los agentes es decidir cómo dividirse entre ellos en sub-grupos de manera que cada sub-grupo tenga el conjunto de habilidades necesarias para tener éxito en el ambiente. De forma similar, un grupo de agentes con diferentes habilidades debe decidir cómo dividirse a sí mismo de manera que pueda manejar el mayor número de tareas posible de la manera más eficiente.

En este trabajo usamos la teoría de juegos como la base sobre la que recae una de las partes que utiliza la técnica propuesta: las subastas. Para poder formalizar de manera correcta este marco de trabajo es necesario definir primero los escenarios conflictivos entre agentes egoístas y las estrategias para resolver dichos conflictos (juegos en forma normal y extensiva). Posteriormente describimos los escenarios donde no se tenga toda la información perteneciente a las preferencias de los agentes (juegos bayesianos). Veremos también el diseño de sistemas multiagentes desde la

perspectiva del diseñador (decisión social). Luego introduciremos la base teórica para construir protocolos de diseño de sistemas multiagente que tomen en cuenta tanto al diseñador como a los agentes, sin que sea necesario requerir de forma explícita el conocimiento privado de estos (diseño de mecanismos) y finalmente, definimos a las subastas mediante estos protocolos. Esta sección se basa en gran parte en [Vidal, 2010, Shoham and Leyton-Brown, 2008].

2.2.1 JUEGOS EN FORMA NORMAL

Las interacciones estratégicas entre los agentes usando la teoría de juegos se representan de forma canónica, mediante un juego en *forma normal*, también llamado *forma estratégica* [Shoham and Leyton-Brown, 2008].

Definición 2. ***(Juego en forma normal)*** *Un juego en forma normal se define como una tupla* $\langle N, A, u \rangle$ *donde:*

- N *es un conjunto finito de* n *jugadores, indizados por* i*.*
- $A = A_1 \times \ldots \times A_n$ *donde* A_i *es un conjunto finito de acciones* disponibles *para el jugador* i*. Cada vector* $a = (a_1, \ldots, a_n) \in A$ *es llamado un perfil de acción.*
- $u = (u_1, \ldots, u_n)$ *donde* $u_i : A \to \mathbb{R}$ *es una función real de utilidad (o paga) para el jugador* i*.*

Es importante recordar que la función de utilidad (ecuación 2.1) definida en la sección 2.1.1 (y todas las funciones derivadas de esta) se refiere a una valuación que el agente calcula con base en el estado del ambiente en el que cree que se encuentra. La función de utilidad definida en esta sección, a diferencia de la anterior, se refiere al cálculo de algún valor basado en el conjunto de acciones que pueda realizar un jugador en el ambiente en algún momento. A partir de esta sección, utilizaremos las palabras *agente* y *jugador* de forma intercambiable.

Una forma natural de representar juegos en la forma normal es mediante una matriz n-dimensional. Por esta razón, los juegos en forma normal también son llamados *juegos de matriz*. Como ejemplo incluimos a uno de los juegos más comunes: *el Dilema del Prisionero* [Vidal, 2010]. Se describen normalmente a dos sospechosos A y B arrestados por la policía. La policía no tiene evidencias suficientes para establecer una condena y, habiéndolos separado, les ofrecen el mismo trato individualmente: si uno de ellos testifica para acusar al otro sospechoso y el otro no dice nada, el cómplice que no coopera recibe la sentencia completa de 10 años, mientras que el traidor sale libre. Si ambos no cooperan, la policía sólo puede sentenciarlos por 6 meses con un cargo menor. Si ambos se traicionan, cada uno recibe una sentencia de 2 años cada uno. La representación matricial de este juego se puede observar en la figura 2.2, donde las opciones de un agente se representan por los renglones y las opciones del segundo agente se representan por las columnas. Aunque en este caso, en cada celda está el resultado de las opciones que podrían tomar los agentes, lo que se acostumbra es incluir en cada casilla o celda las utilidades de los agentes al haber elegido las opciones correspondientes con los renglones y las columnas.

Gracias a las herramientas que nos brinda el formalismo de los juegos en forma normal, podemos analizar las conductas de los jugadores en un juego y, a su vez, definir juegos más complejos como los juegos en su forma extensiva, juegos repetidos, juegos bayesianos, etc. Estos juegos nos permiten construir de forma incremental la base teórica necesaria para una de las técnicas utilizadas en este trabajo: las subastas, mezclando el formalismo de los juegos bayesianos y el diseño de mecanismos.

Hemos definido hasta ahora las acciones disponibles para cada jugador en un juego pero no hemos definido sus conjuntos de *estrategias* o *decisiones* disponibles. Es posible que una estrategia sea seleccionar una sola acción y jugar con ella. Esta estrategia se llama *estrategia pura* y usaremos la notación desarrollada para las acciones para representarla. Una *decisión* de estrategias puras para cada agente se le llama *perfil de estrategias puras*. Los jugadores también pueden seguir otro tipo

		A	
		No coopera	Traiciona
B	No coopera	Ambos son sentenciados 6 meses	B es sentenciado 10 años A se va libre
	Traiciona	A es sentenciado 10 años B se va libre	Ambos son sentenciados 2 años

Figura 2.2: Matriz que representa el juego del *dilema del prisionero* en su forma normal.

(menos obvio) de estrategias: seleccionar al azar sobre el conjunto de acciones disponibles, de acuerdo a alguna distribución de probabilidad. Tal estrategia es llamada una *estrategia mixta* y se denota como S_i para un jugador i.

Ahora que se han definido qué son los juegos en forma normal y qué estrategias están disponibles para los jugadores en ellos, la pregunta es cómo razonar sobre tales juegos. En la teoría de decisiones para agentes individuales, la noción clave es aquella de una estrategia óptima, es decir, una estrategia que maximiza la paga o utilidad esperada para un ambiente dado. La situación para el caso de un solo agente puede estar llena de incertidumbre, principalmente debido a la naturaleza estocástica del ambiente. Sin embargo, cuando están presentes múltiples agentes en el ambiente, la situación es mucho más compleja, debido a que cada uno busca maximizar su propia utilidad, por lo que la noción de una estrategia óptima para un agente carece de sentido debido a que la mejor estrategia depende de las decisiones de los demás [Shoham and Leyton-Brown, 2008].

La teoría de juegos lidia con este problema, identificando ciertos subconjuntos de resultados, llamados *conceptos de solución*. Los dos conceptos de solución fundamentales son la *optimalidad Pareto* y el *equilibrio de Nash*. Tener presentes estos conceptos al momento de diseñar mecanismos nos permite tener la perspectiva de los agentes, haciendo al mecanismo más justo para los participantes, en caso de que eso se requiera.

OPTIMALIDAD PARETO

Desde el punto de vista de un observador externo, se podría preguntar si algunos resultados del juego pueden ser mejores que otros. Esta pregunta es complicada porque no hay manera de saber si los intereses de un agente son más importantes que otros en todos los casos. Por ejemplo, en un ambiente donde los agentes expresen su utilidad en dinero, pero con diferentes monedas y uno no tiene información alguna sobre los diferentes tipos de cambio. Aun cuando no sea posible saber qué resultado es mejor que otro, es decir, si 10 unidades de la moneda A valen más que 5 de una moneda B, existen situaciones en las que podemos decir con seguridad si un resultado es mejor que otro. Por ejemplo, es mejor tener 10 unidades de la moneda A y 3 de la moneda B que tener 9 unidades de la moneda A y 3 de la moneda B, sin importar el tipo de cambio. Formalizamos esta idea en la siguiente definición.

Definición 3. ***(Dominación Pareto)*** *Un perfil de estrategias s Pareto-domina a un perfil de estrategias s' si para todo $i \in N$, $u_i(s) \geq u_i(s')$, y $\exists j \in N$, tal que $u_j(s) > u_j(s')$.*

En otras palabras, en un perfil de estrategias Pareto-dominado, algún jugador ($\exists j \in N$) puede mejorar su utilidad ($u_j(s) > u_j(s')$) sin hacer que la utilidad de todos los demás jugadores empeore ($\forall i \in N,\ u_i(s) \geq u_i(s')$). Hay que observar que definimos la dominación Pareto sobre perfiles de estrategia, no sólo perfiles de acciones. La dominación Pareto nos da un ordenamiento parcial sobre perfiles de estrategia. Es decir, no siempre podemos identificar un único *mejor* resultado, pero podríamos tener un conjunto de óptimos no comparables entre ellos.

Definición 4. ***(Optimalidad Pareto)*** *Un perfil de estrategias s es Pareto-óptimo o estrictamente Pareto-eficiente, si no existe otro perfil de estrategias $s' \in S$ que Pareto-domine a s.*

EQUILIBRIO DE NASH

Ahora veremos a los juegos desde la perspectiva de un agente, en lugar de la perspectiva de un observador externo. Esto nos llevará al concepto de solución más importante, el *equilibrio de Nash.* La primera observación es: si un agente supiera cómo los demás agentes van a jugar, su problema estratégico sería muy simple, ya que se reduciría al problema de elegir la acción que maximice su utilidad. Formalmente, definimos a $s_{-i} = (s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n)$ como un perfil de estrategias s sin la estrategia del agente i. Así podemos escribir $s = (s_i, s_{-i})$. Si todos los agentes, a excepción de i al que denotamos $-i$, se comprometiera a jugar el perfil de estrategias s_{-i}, un agente i que maximice su utilidad, se enfrentaría con el problema de determinar su *mejor respuesta*.

Definición 5. ***(Mejor respuesta)*** *La mejor respuesta de un jugador i al perfil de estrategias s_{-i} es una estrategia mixta $s_i^* \in S$ tal que $u_i(s_i^*, s_{-i}) \geq u_i(s_i, s_{-i})$ para todas las estrategias $s_i \in S_i$.*

En general, un agente no puede saber las estrategias que otros jugadores usarán. Por lo que la noción de la mejor respuesta no es un concepto de solución. Sin embargo es un concepto necesario para definir el equilibrio de Nash.

Definición 6. ***(Equilibrio de Nash)*** *Un perfil de estrategias $s = (s_1, \ldots, s_n)$ es un equilibrio de Nash si, $\forall i$, s_i es la mejor respuesta a s_{-i}.*

En otras palabras, dado un perfil de estrategias s que sea un equilibro de Nash, a ningún agente le sería beneficioso cambiar su estrategia si supiera qué estrategias están siguiendo los demás.

Todos los conceptos discutidos en esta sección son útiles para analizar el comportamiento de los agentes en un juego dado, es decir, cuando las reglas de un juego ya han sido establecidas (las acciones disponibles para cada jugador, etc.). Sin em-

bargo, es posible analizar el juego desde una perspectiva externa, la del diseñador. Este enfoque se detalla en las siguientes secciones.

2.2.2 Decisión Social

En nuestro trabajo, las reglas sobre las cuales los agentes subordinados expresan sus preferencias sobre alguna tarea en particular y el criterio para reunir todas esas preferencias para elegir finalmente un ganador, se basa en la teoría de la decisión social y el diseño de mecanismos. En las secciones anteriores, se ha adoptado la perspectiva del agente, es decir, nos hemos centrado en cómo un agente debería actuar en una situación dada, en lo que el agente cree o quiere. Ahora adoptamos la *perspectiva del diseñador*, donde nos centramos en las reglas que deberían ser puestas por la entidad encargada de orquestar a los agentes.

Un ejemplo sobre la perspectiva del diseñador es una votación. ¿Cómo debería una entidad central reunir las preferencias de diferentes agentes de manera que refleje mejor los deseos de la población como un conjunto? La votación es sólo un caso especial de los problemas de *decisión social*. La decisión social es una teoría motivacional pero no es estratégica, es decir, los agentes tienen preferencias pero no tratan de manipular el resultado a su favor.

A continuación enunciamos un modelo formal para la decisión social.

Sea $N = \{1, 2, \ldots, n\}$ un conjunto de agentes y O un conjunto finito de resultados (por ejemplo, candidatos en el caso de una votación). Definimos una notación de *preferencia*. Para cualquier $o_1, o_2 \in O$, denotamos como:

- $o_1 \succeq o_2$, a que un agente prefiere *débilmente* o_1 a o_2.
- $o_1 \sim o_2$, a que un agente le es indiferente o_1 u o_2.
- $o_1 \succ o_2$, a que un agente prefiere *estrictamente* o_1 a o_2.

Estas preferencias tienen la propiedad de transitividad, es decir, si $o_1 \succeq o_2$ y $o_2 \succeq o_3$, entonces $o_1 \succeq o_3$. Por la relación de preferencia, podemos inducir un *ordenamiento* de preferencia sobre el conjunto de los resultados O. Con L denotamos al conjunto de todos los ordenamientos posibles sobre O. Los ordenamientos de preferencias de cada agente lo denotamos como un elemento $l \in L$. Ahora definimos varias *funciones sociales* que tienen como objetivo *reunir estas preferencias*.

Definición 7. ***(Función de decisión social)*** *Una función de decisión social (sobre N y O) es una función $C : L^n \to O$, donde $L^n = \underbrace{L \times L \times \ldots L}_{n \text{ veces}}$.*

Definición 8. ***(Función de bienestar social)*** *Una función de bienestar social (sobre N y O) es una función $W : L^n \to L$.*

Además, definimos una función de *bienestar social*, similar a las funciones de decisiones sociales, pero producen objetos más complejos: ordenamientos sobre un conjunto de resultados. Se dice función de *bienestar social* porque el resultado (el ordenamiento de alternativas) de esta función reúne y refleja las preferencias de todos los agentes. Las funciones sociales no sólo surgen en las votaciones, también están presentes cuando se quiere formar una lista de clasificación o *rankings* sobre objetos o los mismos agentes, basándose en alguna métrica como reputación, popularidad, etc. Aunado a los conceptos de solución en la sección de los juegos en forma normal (2.2.1), la teoría de decisión social le permite al diseñador implementar mecanismos que beneficien a los agentes que participan en él (mediante el análisis de sus estrategias), y que lo acerquen al objetivo global de un sistema multiagente.

2.2.3 Diseño de Mecanismos

En la sección anterior, afirmamos que la teoría de la decisión social no es estratégica, es decir, toma las preferencias de los agentes e investiga maneras de cómo reunir esas preferencias. Usualmente, esas preferencias no se saben. Lo que se

tiene es que los agentes declaran unas preferencias, pero se desconoce si son *honestas* o no. Asumiendo que los agentes son *egoístas*, en general no revelan sus preferencias verdaderas. El *diseño de mecanismos* es la versión estratégica de la teoría de la decisión social, que añade a la suposición que los agentes se comportarán de manera que maximicen sus utilidades individuales. Por ejemplo, en una votación, los agentes puede que no voten por su verdadera preferencia.

Como ejemplo, consideremos que estamos cuidando a cuatro niños: Guillermo, Laura, Víctor y Raymundo. Nuestro plan es llevarlos a realizar alguna de sus actividades favoritas: (a) ir al parque, (b) jugar baloncesto, (c) dar un paseo en auto. Para elegir la actividad, les pedimos a los niños que voten por su favorita, desempatando alfabéticamente ((a) sobre (b), etc.). Consideremos el caso en que las preferencias *verdaderas* de los niños son las siguientes:

- Guillermo: $b \succ a \succ c$
- Laura: $b \succ a \succ c$
- Víctor: $a \succ c \succ b$
- Raymundo: $c \succ a \succ b$

Guillermo, Laura y Víctor son honestos, pero no es el caso en Raymundo. Él prefiere las actividades más sedentarias (de ahí sus preferencias), pero conoce bien a sus amigos, por lo que también sabe qué actividades van a votar. Por lo tanto, sabe que si vota (c), ganará (b). Así que prefiere votar por (a) para que empate con (b) y aquella gane después del desempate. ¿Hay alguna manera de evitar la manipulación por parte de Raymundo?

Aquí es donde entra el diseño de mecanismos. Ocasionalmente se le llama de manera coloquial *teoría de juegos inversa*. Recordando que la teoría de juegos se enfoca en la predicción o prescripción del comportamiento de los agentes que participan

en una interacción, en contraste, el diseño de mecanismos comienza asumiendo un conjunto de comportamientos deseados por parte de los agentes y nos preguntamos qué interacción estratégica es necesaria para conseguir esos comportamientos. La propiedad más fundamental de un diseño de mecanismos es que el resultado que surge del juego cuando los agentes interaccionan en él sea consistente con una función social dada. La función social indica el resultado que queremos obtener. Funciones sociales comunes buscan el resultado que maximiza el valor que todos los agentes obtienen o que minimiza la diferencia entre valores obtenidos por cada par de agentes (con el fin de buscar mayor igualdad entre estos).

En el ejemplo del cuidado de niños de la sección anterior, el mecanismo se refiere al establecimiento de dos reglas: *cada niño vota por una actividad* y *la actividad seleccionada es aquella con más votos*, desempatando de forma alfabética". Es un mecanismo bien formado porque especifica las acciones disponibles para cada niño y el resultado depende de las decisiones que se han tomado. La función social corresponde a "la actividad seleccionada es aquella que sea la decisión más votada por la mayor cantidad de niños, desempatando de forma alfabética". Sin embargo este mecanismo no tiene la propiedad de ser *directo* debido a que no incentiva a que los agentes revelen sus verdaderas preferencias (como en el caso de Raymundo). Un *mecanismo directo* es aquel en el que la mejor acción disponible para cada agente es la de anunciar su información *privada*.

Los esquemas de subastas cuyo objetivo sea maximizar la ganancia de todos los agentes es un ejemplo de *mecanismo directo*, pues incentiva a los agentes a decir su oferta *real*. Regresando al ejemplo del cuidado de los niños, un mecanismo directo pediría la actividad favorita y la actividad menos favorita de cada niño, y la función social usada sería una *función de bienestar social*, donde se haga una clasificación sumando los votos positivos (actividad favorita) y restando los negativos (actividad menos favorita) a cada actividad, desempatando alfabéticamente. Eso incentivaría a cada niño a declarar sus preferencias reales.

2.3 SUBASTAS

El escenario de *subastas* es muy importante debido a su uso extendido en la vida real (en aspectos corporativos, gubernamentales, de consumo, etc.) y porque ofrecen un marco teórico general para entender el problema de *asignación de tareas* entre un grupo de agentes autónomos egoístas. Hay que recalcar que el problema general de la asignación de tareas no se limita a la asignación entre un conjunto de agentes egoístas, sino que abarca una variedad muy grande de ambientes y cuya formalización se hace en la sección 2.4.

Una subasta - siendo esta una importante aplicación del diseño de mecanismos - proporciona una solución general al problema de asignación de tareas. Formalmente hablando, una subasta es cualquier protocolo que le permita a los agentes (referidos en este contexto como *compradores* o *postores*) indicar el interés que tienen por una o más tareas y posteriormente usar estas indicaciones para realizar asignaciones y establecer pagos. A la entidad que coordina este protocolo le llamaremos *subastador* o *vendedor*.

Dado que la subasta ofrece una solución importante al problema del diseño de mecanismos (sección 2.2.3), hay una gran variedad de tipos de subastas. Las subastas pueden variar tanto en la cantidad de artículos que se subastan, como en la proporción de agentes que hay de un lado u otro del mercado (subastadores/postores), por ejemplo:

- **Subastas inglesas.** Es el tipo más común de subastas, en donde se establece un precio inicial para el artículo a subastar y los agentes tienen la opción de *anunciar públicamente* su oferta, la cual debe ser mayor a la última oferta anunciada, usualmente por un incremento mínimo colocado por el subastador.
- **Subastas japonesas.** Utilizan la misma naturaleza ascendiente de las ofertas

que en las subastas inglesas, sin embargo el procedimiento es diferente. El subastador establece un precio inicial para el artículo y los postores anuncian si siguen dentro de la subasta. Posteriormente, el subastador anuncia precios ascendentes para el artículo y los compradores deciden, en cada anuncio, si permanecen o se retiran. La subasta termina cuando sólo quede un postor.

- **Subastas holandesas**. Estas empiezan con el subastador estableciendo un precio alto y sucesivamente anunciando precios menores. El ganador es el primer postor que anuncie su decisión de comprar el artículo.

- **Subastas de oferta o sobre sellado.** A diferencia del resto de subastas, en este tipo se envía la oferta de forma secreta o «sellada» al subastador y el precio más alto que este reciba determina el ganador. El precio, sin embargo, que debe pagar para comprar el bien depende del tipo de subasta de oferta sellada. En la subasta de *primer precio*, el ganador paga su oferta. Análogamente, en la subasta de *segundo precio*, el ganador paga la segunda oferta más alta y así sucesivamente.

- **Subastas de posición.** En esta subasta, se le pide a los postores que califiquen su preferencia usando un número real y gana aquel con una preferencia mayor. El precio es determinado por algún otro factor dependiente del dominio de la subasta.

- **Subastas paralelas.** Estas subastas se aplican cuando se tiene un conjunto de artículos. Los postores formulan una oferta por y para cada artículo, y la envían al subastador. Las ofertas de cada artículo no influyen en las ofertas de los demás. Al final, el subastador anuncia los ganadores de todos los artículos al mismo tiempo. Ningún postor puede ganar más de un artículo.

- **Subastas combinatorias.** Nuevamente en el caso de que haya un conjunto de artículos para la venta. En estas subastas, el subastador permite que se hagan ofertas por artículos individuales o por conjunto de artículos. La oferta por un

conjunto de artículos no tiene por qué ser la suma de las ofertas individuales de esos artículos. De esta manera, es posible que tanto el vendedor como los compradores puedan encontrar mejores tratos que no podrían encontrar en otras subastas. Sin embargo, el análisis de las ofertas y la búsqueda de los mejores tratos se vuelven análisis demasiados complejos mientras haya más artículos y más compradores. La inclusión de un solo artículo o comprador duplica la cantidad de ofertas que hay que analizar. Por este motivo, determinar a los ganadores es un problema *NP-completo* [Vidal, 2010, Shoham and Leyton-Brown, 2008]. Hay maneras de encontrar los ganadores de forma óptima si el conjunto de ofertas cumple ciertas condiciones, pero siempre suponiendo que el conjunto de artículos y agentes permanece *estático* [Shoham and Leyton-Brown, 2008].

- **Subastas secuenciales.** Esta subasta está inspirada en las dos anteriores. Los postores hacen su oferta por el primer artículo que anuncia el subastador. Luego este anuncia el ganador, se realiza la asignación y se subasta un nuevo artículo. Sin embargo, y a diferencia de las subastas paralelas, los vendedores que ya han ganado algún artículo, pueden seguir participando.

- **Subastas inversas.** En las subastas comunes, se supone que el mercado está formado por un vendedor o subastador y un conjunto de compradores o postores. En el caso de las subastas inversas, se tiene lo contrario: un comprador y varios vendedores. Por lo que los vendedores ahora son los que funcionan como postores, y el comprador se encarga de elegir la oferta más baja.

En resumen, una subasta es un protocolo para negociaciones que comprende de tres tipos básicos de reglas:

- Reglas de licitación: ¿Cómo se formulan las ofertas? ¿Quién las hace?, ¿cuándo se hacen?, ¿qué pueden expresar?

- Reglas de liquidación: ¿Cuándo ocurren los intercambios?, ¿qué intercambios ocurren como resultado de la subasta?

- Reglas de información: ¿Quién sabe «qué» y «cuándo» con respecto al estado de la negociación?

Formalmente, definimos a las subastas a continuación.

Definición 9. ***(Subasta) [Shoham and Leyton-Brown, 2008]*** *Una subasta es un mecanismo (para un escenario de juego Bayesiano $\langle N, O, \Theta, p, u\rangle$), una tupla $\langle A, \S, \wp\rangle$, donde*

- *N es un conjunto de agentes.*

- *$O = X \times \mathbb{R}^n$ es el conjunto de posibles resultados que se forma por el conjunto de decisiones X (las formas de asignar el artículo) y todas las posibles maneras de cobrarle a los agentes.*

- *$\Theta = \Theta_1 \times \ldots \Theta_n$, donde Θ_i es el tipo del agente i. Es decir, toda la información privada de ese agente, como el conocimiento de su propia función de pago, la noción que tiene de los pagos de otros agentes, etc.*

- *$p : \Theta \to [0, 1]$ que es una función de distribución de probabilidad sobre los tipos.*

- *Una función de utilidad $u_i : O \times \Theta \to \mathbb{R}$, donde el agente i del tipo $\theta \in \Theta$ manifieste su preferencia por una asignación de algún artículo $x \in X$. A esta preferencia en las subastas, se le acostumbra llamar **valuación** en lugar de utilidad.*

- *$A = A_1 \times \ldots \times A_n$, donde A_i es el conjunto de acciones disponibles para el agente $i \in N$.*

- *Una función de decisión $\S$ que seleccione uno de los resultados, dadas las acciones del agente.*

- *Una función de pago $\wp$ que determine lo que cada agente debe pagar dadas las acciones de todos los agentes.*

Por ejemplo, en una subasta de sobre sellado, cada conjunto A_i es un intervalo de $\mathbb{R}$. Es decir, la acción de un agente es la declaración de una oferta entre un valor mínimo y un máximo. Como en todos los problemas de diseño de mecanismos, las funciones de decisión y de pago dependen del objetivo de la subasta, como una asignación eficiente o maximizar la ganancia.

2.4 Formalización de la asignación de tareas

En esta sección formalizamos el problema de asignación de forma general, es decir, fuera del contexto de la teoría de juegos. Al problema, en el área de optimización combinatoria, se le conoce como *problema de asignación generalizada* [Cohen et al., 2006] y en el área de robótica se le conoce como *problema de asignación de tareas* [Dias et al., 2006].

La definición del problema general de asignación de tareas se detalla a continuación.

Definición 10. ***(Problema de asignación generalizada)*** *Hay un número n de* agentes *y un número m de* tareas*. Cualquier agente puede ser* asignado *para realizar cualquier tarea, incurriendo en un* costo y ganancia*, que pueden variar dependiendo de la asignación agente-tarea. Además, cada agente tiene un* presupuesto *y la suma de los costos de las tareas asignados a él no puede superar dicho* presupuesto*. Se requiere encontrar un conjunto de asignaciones de tal manera que todos los agentes no excedan su presupuesto y la ganancia total de las asignaciones se maximice.[Cohen et al., 2006]*

Esta definición general del problema no cubre dos casos: La posibilidad de que el ambiente es dinámico, por lo que toma como suposición que todos los recursos y tareas se conocen desde el principio, y además restringe las asignaciones a parejas de sólo un recurso y una tarea. Sin embargo, la variación del problema en el que nos centramos en esta tesis se define a continuación.

Definición 11. ***(Problema de asignación de tareas en ambientes dinámicos)*** *Un problema de asignación de tareas se define como una tupla* $\langle X, \kappa, R, T, \phi \rangle$*, donde,*

- $X = \mathbb{Z}^+$ *es el conjunto que representa los instantes de tiempo, indizados por* x*.*
- $\kappa : 2^N \times 2^T \to \mathbb{R}$ *es una función costo que mapea una asignación entre recursos y tareas a un costo.*
- $R = R_1 \times \ldots \times R_{|X|}$*, donde* R_x *es un conjunto de* recursos *en un instante de tiempo* x*.*
- $T = T_1 \times \ldots \times T_{|X|}$*, donde* T_x *es el conjunto de* tareas *en un instante de tiempo* x*.*
- $\phi : 2^R_x \to 2^T_x$ *es una función asignación que mapea un subconjunto de recursos a un subconjunto de tareas en un instante de tiempo* x*.*

El problema consiste en minimizar

$$\sum_{x \in X} \sum_{r \in 2^{R_x}} \kappa\left(r, \phi\left(r\right)\right) \tag{2.10}$$

2.4.1 Taxonomía del área de asignación de tareas

Hay distintas clasificaciones de los problemas de asignación de tareas. La primera y más común, definida formalmente en [Gerkey, 2003] (representada en la figura 2.3), define tres ejes por los que puede describirse un problema de asignación. La terminología utilizada por el autor hace referencia únicamente a *robots*, pero puede reemplazarse sin ningún tipo de modificación formal por el concepto de *recurso* o *agente*. En esta sección utilizaremos el concepto de *utilidad* un poco diferente. Con utilidad nos referimos (en esta sección únicamente) a la ganancia desde el punto de

vista del diseñador (a diferencia de la utilidad definida en secciones anteriores) que se obtiene al asignar un recurso a una tarea.

- Capacidades de recursos: recursos mono-tareas (MoT) vs recursos multi-tareas (MuT). El eje de capacidades de recursos se refiere a la capacidad de un recurso para realizar una o más tareas *simultáneamente*.
- Requerimientos de tareas: tareas que requieren un solo agente (SA) vs tareas que requieren múltiples agentes (MA). Este eje se refiere a la cantidad de recursos que una tarea requiere para ser completada de manera óptima.
- Temporalidad: asignaciones instantáneas (AsI) vs asignaciones extendidas en el tiempo (AsT). La temporalidad hace referencia a dos cosas: a la disponibilidad de información pasada para realizar la asignación en un instante de tiempo y a la secuencia de ejecución de múltiples tareas, posiblemente interrelacionadas.

Las restricciones en los problemas de asignación de tareas son funciones posiblemente arbitrarias que restringen el espacio de soluciones y se clasifican comúnmente de la siguiente manera [Gerkey, 2003, Korsah et al., 2013].

- Restricciones de capacidad. Estas determinan qué tipo de agente es capaz de realizar alguna tarea específica, además de señalar cuántas tareas puede realizar un agente simultáneamente.
- Restricciones de simultaneidad. Estas indican cuándo un conjunto determinado de tareas debe ser asignado y realizado al mismo tiempo o, por el contrario, evitar que un conjunto de tareas sea ejecutado simultáneamente.
- Restricciones de precedencia: Siendo estas las más comunes en esquemas de planificación (*scheduling*), indican el orden en el que las tareas deben ser realizadas.

Gerkey indica que esta clasificación excluye la consideración de utilidades inter-relacionadas y las restricciones en las tareas, es decir, supone independencia entre utilidades y entre tareas. Las utilidades inter-relacionadas son aquellas en las que el cálculo de la utilidad para un agente, depende de las decisiones anteriores que ha tomado él mismo u otros agentes. El trabajo en [Korsah et al., 2013] propone una clasificación que complementa a [Gerkey, 2003] e incluye los casos no considerados. En nuestra investigación, proponemos lidiar con estas utilidades inter-relacionadas de forma implícita, mediante el aprendizaje sobre la creación de mercados y las asignaciones que resultan de estos. Por esta razón, detallamos brevemente la clasificación por el grado de inter-dependencias que propone [Korsah et al., 2013].

- Sin dependencias. Problemas que tienen utilidades independientes. Esto sucede cuando sólo es necesario saber las capacidades del agente para resolver alguna tarea y las relaciones entre estos conjunto se restringe a uno a uno.

- Dependencias internas. Problemas para los cuales, la utilidad de la asignación de un agente a una tarea depende de las tareas que ese mismo agente *esté realizando o haya realizado*. Estas dependencias surgen en problemas cuando existen plazos o tiempos límites en los que se tienen que cumplir tareas o cuando tenemos recursos capaces de realizar simultáneamente varias tareas.

- Dependencias cruzadas. Problemas para los cuales, la utilidad de un agente para una tarea depende de las tareas que otros agentes estén realizando, hayan realizado o *realizarán*. Este tipo de dependencias se manifiesta en problemas donde existan restricciones de precedencia, proximidad o simultaneidad o en donde tareas multi-recurso exijan la formación de coaliciones.

- Dependencias complejas. Problemas para los cuales existen dependencias internas y externas. Estas dependencias se dan en problemas donde surja la pregunta *¿cuál subconjunto de tareas debe asignarse?*, en adición a los problemas planificación y asignación comunes.

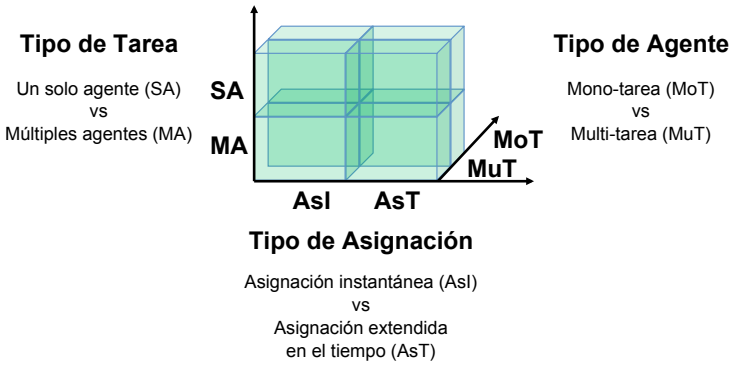


Figura 2.3: Representación visual de la taxonomía propuesta por [Gerkey, 2003]

2.4.2 Complejidad del problema de asignación

El problema de asignación de tareas está estrechamente ligado al *problema de asignación cuadrática* [Garey and Johnson, 1990] y al *problema generalizado de asignación* (este, a su vez, surgió como la generalización del problema de la mochila [Chekuri and Khanna, 2000]), por lo que pertenece a la categoría de problemas *NP-duro* [Yedidsion et al., 2011] (y *NP-completo en su variación como problema de decisión*). También pertenece a la clase de problemas *APX-duro* [Chekuri and Khanna, 2000]. La clase de problemas *APX-duro* se refiere al conjunto de problemas de *optimización* para el cual, si $P \neq \text{NP}$, no existe un algoritmo de aproximación en tiempo polinomial [Papadimitriou and Yannakakis, 1991] que garantice una solución cercana a la óptima por una cota fija. Sin embargo, [Shmoys and Tardos, 1993] afirman que existen aproximaciones en tiempo para algunas variaciones del problema, en específico, para el caso de tareas y recursos homogéneos. La complejidad del problema de asignación depende principalmente del tipo de asignaciones permitidas en el escenario analizado y en el número de recursos y tareas. Un cálculo específico del problema que estamos enfrentando se realiza en el capítulo 4.

Capítulo 3

Trabajo relacionado

En esta tesis se propone una técnica para resolver el problema de asignación de tareas. Algunos problemas que pueden ser vistos como una instancia del problema de asignación de tareas son: planeación, exploración, balance de carga en sistemas multi-procesador, escenarios de rescate, etc.

El problema de asignación de tareas puede ser clasificado de varias maneras: por su variabilidad en el tiempo (ambientes *dinámicos* y *estáticos*), por el tipo de relaciones entre recursos y tareas [Gerkey, 2003]: relaciones *uno a uno, uno a muchos, muchos a muchos*, por la complejidad en las *dependencias* entre tareas [Korsah et al., 2013], por su enfoque en el modelado del problema [van der Horst and Noble, 2010]: de forma *centralizada, distribuida, o híbrida*, etc. Siendo esta última la clasificación más común.

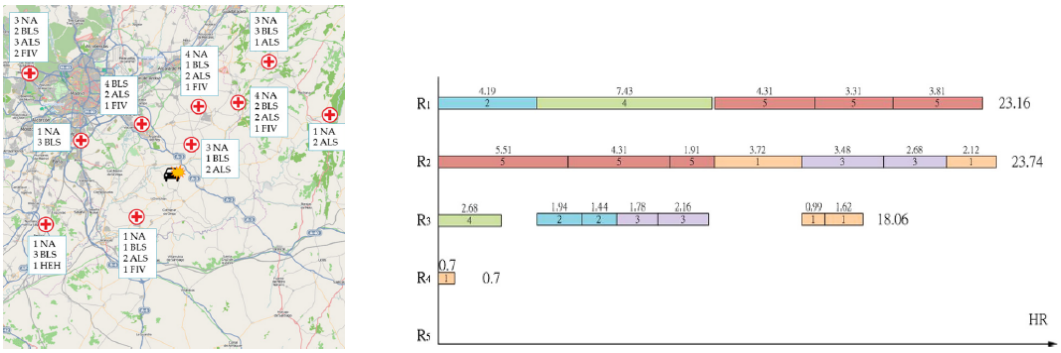
3.1 Asignación de tareas centralizada

Los enfoques centralizados toman ventaja de la disponibilidad del conocimiento global del sistema a un único nodo de procesamiento, por tanto, las soluciones óptimas son cada vez más difíciles de encontrar mientras el tamaño del espacio de soluciones crece. Es por esto que las técnicas centralizadas son elegidas en problemas donde los conjuntos de tareas y recursos son pequeños y sufren poco cambio.

Trabajos como [Fogue et al., 2013, Huang et al., 2013, Tsai et al., 2013] modelan un problema de asignación de tareas como un problema de optimización multiobjetivo y encuentran soluciones mediante algoritmos genéticos o evolutivos. En general, los algoritmos evolutivos consisten en la generación inicial de soluciones y su mejora iterativa mediante operadores de *cruza*, *mutación* y *selección de sobrevivientes* entre generación y generación de soluciones. De igual forma, esta solución requiere formas de representar las soluciones y de evaluar su calidad. De manera específica, [Fogue et al., 2013] modela un escenario de asignación de recursos sanitarios (vehículos médicos de emergencia equipados para tratar heridas específicas) a accidentes de tránsito en un ambiente urbano. El problema que los autores tratan de resolver es asignar los recursos sanitarios disponibles para minimizar varios objetivos, entre los cuales se encuentran: el daño que reciben las personas relacionadas en un accidente vial, el costo de la operación de rescate, el uso excesivo de los recursos. Las asignaciones encontradas pueden resultar en un solo recurso o conjunto de recursos, con las capacidades de enfrentar los requerimientos específicos de un accidente. En este trabajo, las soluciones son representadas como vectores que señalan el conjunto de recursos asignados a una sola tarea. Debido a la formulación del problema, este enfoque centralizado no puede enfrentarse a la situación de varios accidente simultáneos. Esta es una situación similar a utilizar subastas de un solo artículo en nuestra técnica, por lo que el enfoque que nos proporciona este trabajo nos podría ser de utilidad en niveles inferiores de la jerarquía propuesta. La carencia remarcada puede ser subsanada por el aprendizaje por refuerzo del nivel superior. De manera similar a [Fogue et al., 2013], [Kang et al., 2013] plantea un algoritmo genético de optimización para resolver el problema de asignación en un instante de tiempo para varios recursos y tareas. A diferencia del trabajo anterior, este se enfoca en la fiabilidad del sistema, no en la eficiencia.

Por otro lado, la técnica *IDEA* [Tsai et al., 2013], proporciona como resultado un esquema de planificación para asignar múltiples procesadores (recursos) a múlti-

ples tareas en un escenario de cómputo en la nube, en donde el conjunto de recursos disponibles permanece fijo y la asignación es uno a uno. A diferencia de [Fogue et al., 2013], las soluciones representan una secuencia de sub-tareas (que componen una tarea) asignadas a un recurso y el problema de optimización multi-objetivo consiste en encontrar el mínimo del costo de renta por recurso, tiempo de procesamiento, uso de espacio de almacenamiento, etc. Este enfoque, a diferencia del anterior, realiza un trabajo similar a la etapa de aprendizaje de nuestra técnica. Pero a diferencia de esta, no es escalable ni considera ambientes dinámicos. La diferencia en las soluciones generadas por las técnicas recién discutidas se muestra en la figura 3.1. En el escenario de simulación que se presenta en la figura 3.1(a), la ejecución completa de la técnica regresa como resultado el mejor destino para el/los vehículos de emergencia disponibles y la asignación se hace de forma inmediata. Para asignar a toda la flotilla de vehículos, es necesario ejecutar la técnica repetidas veces. En contraste, en el escenario de simulación que se presenta en la figura 3.1(b) se observa que la aplicación de la técnica correspondiente tiene como resultado la planificación de múltiples tareas para cada uno de los recursos disponibles.



(a) Escenario de simulación en [Fogue et al., 2013].

(b) Escenario de simulación en [Tsai et al., 2013].

Figura 3.1: Diferencia entre problemas de asignación: (a) asignación instantánea y (b) planificación.

La arquitectura propuesta por [Huang et al., 2013] utiliza de igual manera un algoritmo genético para asignar recursos de procesamiento a un conjunto de tareas

en un escenario de cómputo de nube. Dicho proceso de asignación pasa por dos etapas, la primera es la predicción de la *cantidad* de recursos que necesitará una tarea y posteriormente la asignación de los recursos físicos restringidos a la cantidad suministrada por la fase previa. La etapa de predicción, sin embargo, se basa en un procesamiento de clasificación supervisada. Esto significa que requiere entrenamiento, es decir, saber de antemano los requerimientos físicos de un número de procesos conocidos. Este inconveniente hace que la técnica no sea aplicable en ambientes desconocidos, a diferencia de nuestra técnica que, a pesar de requerir cierto conocimiento del dominio en la formulación de las subastas, la etapa de aprendizaje compensa dicho desconocimiento. Otros enfoques centralizados [Cheng et al., 2013, Celaya and Arronategui, 2013] modelan un problema de asignación de tareas en un sistema multi-procesador restringido a recursos *homogéneos*. De manera más específica, [Cheng et al., 2013] resuelve el modelo usando programación lineal y lo restringe a ambientes estáticos. [Celaya and Arronategui, 2013] utiliza algoritmos de ruteo similares a los usados para la distribución de paquetes en redes para ambientes dinámicos, lo que produce que la calidad de las asignaciones no sean óptimas, ni tengan alguna garantía sobre la calidad de sus asignaciones. Sin embargo, suministra soluciones de manera muy eficiente.

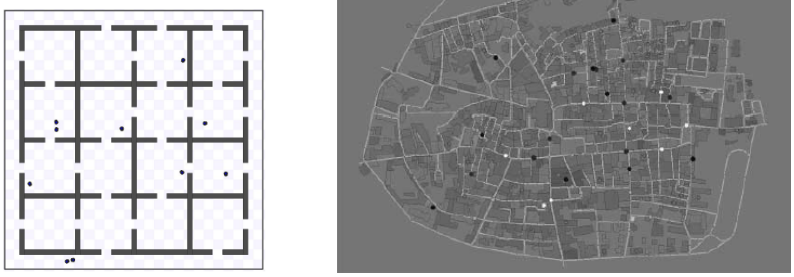
3.2 Asignación de tareas distribuida e híbrida

Los enfoques distribuidos se caracterizan por la ausencia de un coordinador o nodo que englobe toda la información del sistema. Estos enfoques son llamados frecuentemente como la solución para grandes ambientes dinámicos pues están pensados para problemas que crecen indefinidamente, a diferencia de los problemas en los que se aplican los enfoques centralizados. Debido a la creciente complejidad de los sistemas distribuidos modernos, ha crecido el interés por los enfoques distribuidos ya que prometen robustez y *escalabilidad*. La inteligencia artificial distribuida busca

resolver estos problemas mediante agentes (sección 2.1), ya sea mediante agentes autónomos auto-motivados que buscan su propio beneficio o grupos de agentes que cooperan para realizar una tarea en común, o bien, enfoques híbridos. Uno de los escenarios de experimentación más comunes para la asignación de tareas distribuida es la exploración (figura 3.2, donde se muestra escenarios de experimentación en (a) un escenario de malla, y en (b) un escenario realista). El objetivo de un problema de exploración, como su nombre lo dice, es explorar un conjunto de locaciones (o la mayor parte posible de estas) de un lugar, dado un conjunto de exploradores (o robots móviles). Además, estos robots deben desplazarse lo menos posible (o lograr algún otro objetivo relacionado). Un problema de exploración puede modelarse como un problema básico de asignación de tareas, donde cada recurso es un robot móvil, cada tarea es una locación específica del lugar a explorar y el costo de una asignación entre un robot y una tarea es la distancia que tiene que recorrer el robot para llegar a esta.

En la técnica propuesta por [Chapman et al., 2010], se dota a cada agente con la capacidad de descubrir su propia función de utilidad en un ambiente dinámico, pero requiere información previa sobre los requerimientos de una tarea con un tipo de recurso en específico, por lo que el modelo se restringe a recursos homogéneos. [Brutschy et al., 2014] proponen una técnica basada en el paradigma de enjambre o *auto-organizativo*, en donde no es necesaria la comunicación entre agentes y cada agente toma sus propias decisiones de acuerdo a la percepción propia del rendimiento del enjambre (todo el grupo de agentes realizando la tarea) y a los cambios que sufre el ambiente. Aunque esta técnica funciona en grandes ambientes dinámicos, requiere de gran conocimiento del ambiente en el que se implementa, por lo que su eficacia se encuentra limitada en ambientes simples con pocos tipos de tareas y recursos.

Los enfoques híbridos, es decir, aquellos que combinan aspectos centralizados y distribuidos limitan el crecimiento del espacio de soluciones al dividir el problema en sub-problemas y aplicar técnicas centralizadas en cada uno de estos como



(a) Escenario de malla o cuadrícula para navegación de robots.[Heap and Pagnucco, 2012a]

(b) Escenario realista equivalente.

Figura 3.2: Escenarios típicos de experimentación en asignación de tareas distribuida. [Chapman et al., 2010]

en [Tolmidis and Petrou, 2013, Zhang et al., 2009, Zhang et al., 2010]. Por este motivo, estas técnicas califican como jerárquicas. En el caso de [Tolmidis and Petrou, 2013], se propone una técnica que comienza con la agrupación de los recursos. Dicha agrupación es determinada previamente por el usuario. La etapa inicial del algoritmo consiste en la etapa de sensado, donde se busca en el ambiente alguna tarea. Si alguna tarea es descubierta por algún grupo, esta se procesa por el coordinador del grupo que la descubrió y la anuncia a los agentes del grupo capaces de realizarla. Acto seguido, el coordinador abre una subasta donde los agentes previamente anunciados podrán ofertar para realizar la asignación. Finalmente se formula la oferta de cada agente mediante un algoritmo genético para resolver un problema de optimización multi-objetivo que tome en cuenta las capacidades de cada robot y las particularidades de las demandas de cada tarea. Aunque esta técnica admite recursos y tareas heterogéneas en un ambiente dinámico, la búsqueda de asignaciones es computacionalmente cara y, a pesar de ser un enfoque híbrido, su eficacia se encuentra limitada por la exigencia de las capacidades computacionales en cada agente y en que las subastas se limitan a grupos, por lo que si hay recursos mejor

equipados para realizar una tarea en un grupo que no la descubrió, estos simplemente se desaprovechan. En [Zhang et al., 2009, Zhang et al., 2010], por otro lado, se aplica una técnica propia del área de visión por computadora - utilizada en segmentación y rastreo de movimientos - modelando las tareas como un grafo. Como procesamiento inicial, la técnica conecta las tareas representadas por nodos mediante alguna característica de similaridad. Posteriormente, hace agrupaciones entre tareas y asignaciones entre estos grupos y recursos inicialmente al azar. La búsqueda de asignaciones óptimas se realiza refinando de manera iterativa la asignación inicial. Esto se hace mediante la transferencia de nodos entre agrupaciones o mediante intercambios entre grupos completos asignados a agentes diferentes. Después de cada cambio, los agentes calculan el costo de realizar la nueva asignación. Si el costo es mejor, se acepta el cambio. De otra forma, se proponen otros cambios. Inicialmente, se le asigna el rol de coordinador a un solo agente y el proceso de refinación iterativa se realiza totalmente por este. Cuando una tarea nueva es descubierta por un agente, el rol de coordinador se le otorga a este y el proceso de refinación iterativa se hace con la información correspondiente a las tareas asignadas a los agentes cercanos. Debido a que la complejidad computacional del algoritmo depende del número de agentes y que el proceso para proponer cambios al grafo que representa las tareas es computacionalmente caro, la técnica sólo es conveniente en escenarios donde haya pocos agentes y de grandes capacidades.

3.3 Paradigmas basados en mercados

Los paradigmas basados en mercados son comunes tanto en enfoques centralizados como distribuidos pero en ocasiones se suele ignorar esta distinción debido a que la transformación de un enfoque a otro se realiza fácilmente al permitir que los agentes sean capaces de crear y participar simultáneamente en mercados [Zhang et al., 2010]. Los paradigmas basados en mercados, en particular las subastas, se

ajustan de manera más natural a ambientes estáticos pues las ofertas de un agente se basan en el interés que tiene en ciertos artículos sobre todos los demás. Un ambiente dinámico dificulta esta consideración, debido a que que los compradores no suelen tomar en cuenta futuros artículos (es decir, que no estén en el mercado en el momento de la subasta) cuando realizan una oferta. Los ambientes dinámicos también impiden que la mayoría de las técnicas basadas en las subastas sean capaces de considerar las consecuencias de asignaciones a mediano/largo plazo. La calidad de las asignaciones obtenidas por técnicas basadas en subastas depende de varios factores, entre ellos: la calidad del cálculo de la oferta de los agentes que participan en una subasta, el protocolo de envío de ofertas (de agentes individuales o grupos de ellos, hacia un solo artículo o conjunto de artículos), la determinación del ganador de la subasta, etc. De forma específica, el protocolo del envío de las ofertas determina la fidelidad con la que se representa el espacio de búsqueda en una subasta definido por las funciones de oferta y se refiere a la elección entre subastas paralelas, combinatorias, secuenciales, etc. Una de las propuestas más utilizadas que se encuentra en un término medio entre las subastas paralelas y las subastas combinatorias, son las subastas secuenciales. Es utilizada en muchos trabajos como [Zheng et al., 2006, Zheng and Koenig, 2010, Heap and Pagnucco, 2012b, Heap and Pagnucco, 2012a, Heap, 2013]. En trabajos como [Leme et al., 2012, Said, 2011, Schoenig and Pagnucco, 2011], se ofrece evidencia tanto teórica como experimental que apoya la idea de que las subastas secuenciales (sección 2.3) tienen resultados tan buenos como las subastas paralelas de primer y segundo precio, y que aplicadas a ambientes dinámicos tienen resultados comparables a ambientes estáticos. Aprovechamos esta evidencia para aplicar una variación de subastas que están inspiradas en las subastas secuenciales, y su vez, comparar el rendimiento de nuestra técnica con estas subastas.

Trabajos como [Heap and Pagnucco, 2012b, Heap and Pagnucco, 2012a] modelan un problema de exploración como uno de asignación de tareas, en donde forman grupos de tareas mediante el algoritmo de agrupamiento *K-Means* y se subastan de

forma similar a las subastas secuenciales [Schoenig and Pagnucco, 2011]. Sin embargo, por estar inspirado en un problema de exploración, el modelo se restringe a tareas y recursos homogéneos. Aunque [Choi et al., 2009, Binetti et al., 2013] también modelan el problema de asignación con base en un escenario de exploración, el uso de las subastas es muy diferente. En ambos casos, cada agente crea una subasta para cada tarea y todos los agentes participan en todas las subastas. Para elegir las asignaciones, se realiza un proceso de consenso tomando en consideración todas las listas de ganadores y ofertas que posee cada agente. Esto significa que, para llegar a consensos reales, el ambiente debe ser estático. En el caso de [Pippin and Christensen, 2011], para el mismo escenario de exploración, la formulación de la oferta incluye una estimación probabilista sobre la veracidad de la información que arrojan los sensores de cada robot. De manera similar, para mejorar la calidad de las formulaciones de las ofertas, [Pippin and Christensen, 2013] proponen aplicar aprendizaje por refuerzo a cada uno de los agentes para descubrir un factor de costo que modifique la oferta que el agente proporciona. En [van der Horst and Noble, 2010], se implementan subastas paralelas para resolver el problema de asignación de tareas en exploración espacial con el requerimiento específico de bajo costo en comunicación entre agentes. [Amador et al., 2014] proponen una técnica que realiza asignaciones de manera justa, es decir, donde los recursos tengan asignaciones con un costo similar entre ellos, tomando como suposición que los recursos son homogéneos. Para esto, modela un problema de exploración como un mercado en donde los lugares a explorar son representados como bienes y los robots son compradores. El objetivo de la técnica es equilibrar la demanda de bienes, dada la capacidad de comprar de los agentes. [Thomas and Williams, 2009] proponen un modelo de subastas inspirado en [Schoenig and Pagnucco, 2011] pero requiere conocimiento *a priori* del ambiente como parte de la definición del problema. En esta técnica, cada agente busca la tarea en la que mejor se desempeña y sólo participa en esa subasta. Al hacer esto, aunque el procesamiento por agente permanece uniforme, se disminuye la carga de comunicación. Además propone tres formas de calcular la oferta por agente, tomando en

cuenta las capacidades excedentes de los recursos como penalización. En [Nanjanath and Gini, 2010], se dota a las subastas secuenciales con un tiempo de vigencia a las asignaciones realizadas, por lo que si un agente es incapaz de completar su tarea asignada, esta volverá a ser subastada.

3.4 Comparación entre técnicas

En la tabla 3.1 se presentan las características de los ambientes en los que se desempeña el trabajo relacionado en el problema de asignación de tareas y la técnica propuesta en este trabajo, con el objetivo de distinguir fácilmente qué tipo de soluciones ofrece cada técnica y con qué técnicas es útil compararnos. Hemos definido en las columnas las diferentes particularidades que enfrenta u ofrece cada técnica de asignación de tareas. Específicamente:

- Distribuido/Centralizado. Especifica si la técnica pertenece a un esquema distribuido/híbrido o a uno centralizado.
- Recursos heterogéneos. Indica si la técnica, en su definición, considera a los recursos como entidades con características diferentes entre ellos o como un conjunto de entidades sin distinción.
- Tareas heterogéneas. Indica si la técnica, en su definición, considera a las tareas como elementos con requerimientos diferentes entre ellos o como un conjunto de entidades sin distinción.
- Asignaciones secuenciales o extendidas en el tiempo (AsT). Especifica si el criterio para generar asignaciones toma en cuenta las consecuencias a futuro a corto, mediano o largo plazo
- Ambientes dinámicos. Especifica si la técnica funciona en un escenario donde el conjunto de tareas o recursos cambie con el tiempo.

- Tareas de múltiples agentes (MA). Especifica si la técnica puede formar grupos de recursos para resolver una tarea.

- Conocimiento previo (CP). Especifica si la técnica **no** tiene como requisito el conocimiento de un experto del ambiente para poder ser implementado efectivamente.

Podemos observar que sólo 3 técnicas pueden implementarse en un ambiente dinámico, con tareas y recursos heterogéneos, sin embargo, 2 de estas 3 técnicas ([Tolmidis and Petrou, 2013, Thomas and Williams, 2009]) requieren de conocimiento *a priori* del ambiente para ser aplicado y que bajo la circunstancia de estar un ambiente desconocido, se imposibilita su implementación. Además, en el caso de [Tolmidis and Petrou, 2013], se requiere que el usuario suministre una partición sobre el grupo de agentes y un conjunto de objetivos específicos (en forma de tareas) que el agente tiene que realizar. En el caso de [Thomas and Williams, 2009], la definición del problema requiere especificar un conjunto de habilidades y su costo correspondiente *por tarea*, lo que significa que hay que detallar de antemano las relaciones entre recursos y tareas. La técnica propuesta carece de este requerimiento, por lo que se puede aplicar a un ambiente donde no se tenga conocimiento alguno de la efectividad de ciertos recursos sobre las tareas. [Schoenig and Pagnucco, 2011] tampoco tiene este requerimiento, por lo que esta es la técnica ideal con la que comparar los resultados de la técnica propuesta.

3.5 RESUMEN

En esta sección hemos hablado de las técnicas centralizadas y distribuidas propuestas en los últimos años para resolver el problema de la asignación de tareas y, en especial, hemos discutido sus capacidades, identificando sus debilidades y fortalezas. Los esquemas centralizados de los que hemos hablado, en su mayoría, parten de la

Técnica	Dist.	Rec. Het.	Tar. Het.	AsT.	Din.	MA	CP
[Fogue et al., 2013]		X	X			X	
[Tsai et al., 2013]		X	X	X		X	
[Huang et al., 2013]		X	X	X		X	
[Cheng et al., 2013]			X	X			
[Celaya and Arronategui, 2013]			X	X	X	X	
[Kang et al., 2013]			X	X		X	
[Chapman et al., 2010]	X		X	X	X	X	
[Binetti et al., 2013]	X	X	X			X	
[Choi et al., 2009]	X	X	X			X	
[Brutschy et al., 2014]	X		X		X	X	
[Tolmidis and Petrou, 2013]	X	X	X	X	X	X	
[Zhang et al., 2010]	X	X	X			X	X
[Pippin and Christensen, 2011]	X	X	X				X
[Pippin and Christensen, 2013]	X	X	X				X
[Heap and Pagnucco, 2012a]	X		X			X	
[Nanjanath and Gini, 2010]	X		X		X		
[Thomas and Williams, 2009]	X	X	X	X	X	X	
[Schoenig and Pagnucco, 2011]	X	X	X	X	X	X	X
Propuesta	X	X	X	X	X	X	X

Dist: Distribuido/Centralizado.

Rec. Het: Recursos Heterogéneos.

Tar. Het: Tareas Heterogéneas.

AsT: Asignación secuencial o extendida en el tiempo.

Din: Ambientes dinámicos.

MA: Asignación de múltiples recursos a una tarea.

CP: Conocimiento previo del ambiente.

Tabla 3.1: Tabla comparativa del trabajo relacionado

suposición de que el ambiente es estático. Debido a esto, los trabajos mencionados ignoran dos fenómenos: el crecimiento que puede llegar a tener un ambiente (descartando el requerimiento de escalabilidad) y la influencia positiva o negativa de las asignaciones anteriores en asignaciones con recursos nuevos, es decir, no se garantiza una solución óptima en ambientes dinámicos utilizando estas técnicas. Sin embargo, en ambientes pequeños, muchas de estas técnicas sí garantizan la optimalidad de las soluciones que generan [Fogue et al., 2013, Kang et al., 2013, Huang et al., 2013, Tsai et al., 2013]) e incluso algunas podrían reemplazar a las subastas en nuestra técnica o ayudarlas cuando se realizan las ofertas. Las técnicas distribuidas, por otra parte, consideran en su mayoría ambientes dinámicos. A pesar de que están diseñadas para enfrentar a un ambiente de naturaleza cambiante, estas técnicas se basan más en reactividad que en planeación o previsión. Es decir, estas técnicas no suelen tomar en cuenta las consecuencias de las decisiones a mediano o largo plazo para realizar asignaciones, más bien reaccionan a los cambios del ambiente. Esta carencia es compensada en algunas técnicas (como en [Schoenig and Pagnucco, 2011, Thomas and Williams, 2009, Tolmidis and Petrou, 2013]) al calcular la costo conjunto de realizar una tarea más en el futuro inmediato. La técnica propuesta se distingue del trabajo relacionado porque comparte el enfoque de la planeación de las técnicas centralizadas usando un enfoque distribuido, pero manteniendo la escalabilidad de estas técnicas.

Capítulo 4

Jerarquía de agentes inteligentes usando aprendizaje

Este trabajo propone una técnica que implementa un sistema multiagente jerárquico para resolver el problema de asignación de tareas en un ambiente dinámico. La aplicación específica de esta técnica ha sido en un escenario de combate en tiempo real, donde cada jugador posee un conjunto de unidades que poseen habilidades específicas con el fin de destruir las unidades enemigas, muy similar al ajedrez pero sin restricción de movimientos por turnos o de algún tablero con posiciones predeterminadas.

4.1 Descripción general

El trabajo actual implementa un sistema jerarquizado como lo muestra la figura 4.1. La técnica busca asignar un conjunto *variable* de recursos *heterogéneos*, a un conjunto *variable* de tareas *heterogéneas*. El nivel inferior de la jerarquía corresponde a agentes (o recursos y también nos referiremos a ellos como *agentes subordinados*) que compiten directamente con otros agentes del mismo nivel en mercados, específicamente, subastas abiertas, coordinadas y cerradas por agentes del nivel superior o de mayor jerarquía (y que llamamos *sofisticados o* líderes). Dependiendo del tipo de subasta aplicada, los agentes pueden tener el papel de los *compradores* o de los

vendedores en la subasta. El modelo de la jerarquía es recursiva, es decir, los agentes líderes también pueden competir contra otros agentes en mercados abiertos por agentes del nivel inmediato superior y así sucesivamente. Debido a que el ambiente es dinámico, es decir, los conjuntos de recursos y tareas cambian a través del tiempo, asumimos que las tareas y los recursos pueden llegar al ambiente en cualquier momento, por lo que es importante considerar posponer la ejecución de ciertas tareas o evitar el uso de ciertos recursos con el fin de usarlos en situaciones importantes futuras.

Cuando el número de tareas y/o recursos crece, estos conjuntos pueden ser particionados y aplicar el procedimiento previamente explicado en cada par de particiones (recursos y tareas). Estas particiones deben limitarse en tamaño para poder aprovechar las características de la jerarquía propuesta. Es decir, crear un par de particiones nuevas (de recursos y tareas) con sus propios agentes subordinados y líder. Cuando tanto el tamaño de todas las particiones como el número de particiones presentes han alcanzado el límite establecido, los agentes líderes toman el papel de subordinados y pueden participar en subastas que un nuevo agente líder crea para ellos. Este procedimiento se puede observar con más claridad en la figura 4.1. El proceso de particionado depende del dominio del problema. Algunos criterios para particionar pueden ser la ubicación, en el caso de ambientes donde los recursos y las tareas ocupen un lugar físico. En otros casos, como en el cómputo masivo con decenas, centenas o miles de procesadores, otros criterios como la velocidad de comunicación entre los procesadores pueden ser considerados.

Los agentes capaces de abrir mercados tienen como tarea principal seleccionar el artículo que será subastado, con el fin de aumentar el pago que recibirá de la subasta. Esto puede darse porque el valor que los compradores le dan al artículo puede cambiar mientras transcurre el tiempo. En el contexto del problema de asignación de tareas, un artículo es una *tarea* o *conjunto de tareas* para ser asignado a un *recurso* o *conjunto de recursos*. El criterio para seleccionar una tarea viene del

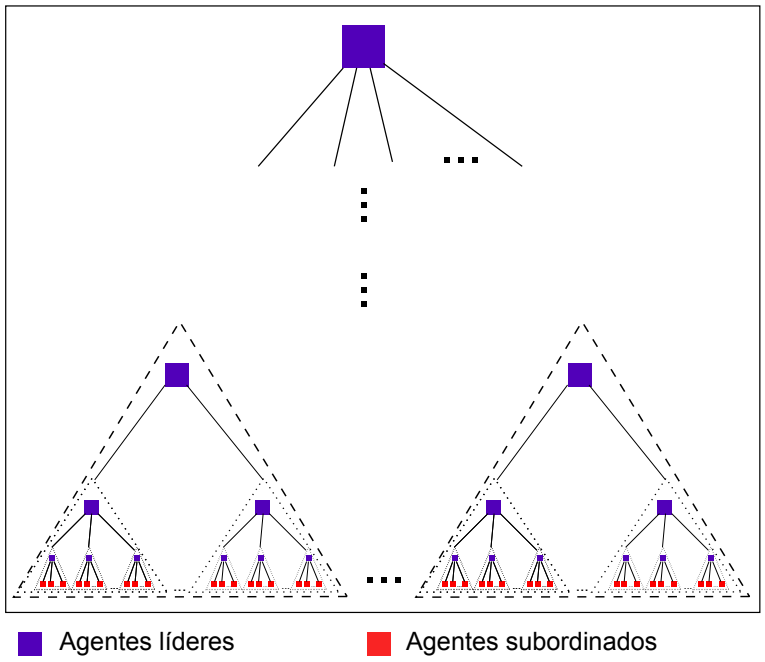


Figura 4.1: Jerarquía de agentes inteligentes propuesta.

conocimiento de la dinámica del dominio. Sin embargo, siendo un ambiente desconocido, no se tiene conocimiento alguno sobre él, por lo que es necesario aplicar una técnica de aprendizaje por refuerzo para obtenerlo. En el caso específico de este trabajo, elegimos a *Q-Learning* ([Watkins and Dayan, 1992]). Posterior a la selección de la tarea, el agente abre una subasta en la que todos los agentes del nivel inferior pueden participar. La asignación entre el ganador o ganadores de la subasta y la tarea seleccionada es realizada por el agente del nivel superior. Es decir, el agente de mayor jerarquía anuncia al ganador o a los ganadores y estos ejecutan la tarea. Acto seguido, el agente de mayor jerarquía observa las consecuencias de esta asignación, es decir, observa si esta asignación afecta positiva o negativamente a todo el grupo y concreta este resultado (llamado recompensa en el contexto del aprendizaje por refuerzo) en el momento en que la asignación deja de existir (por la terminación de la tarea o la pérdida del recurso). Podemos decir que una tarea es la mejor, en el momento de la selección, cuando se sabe que la asignación resultante que incluye esa tarea influye positivamente (a todos los compradores) más que cualquier otra tarea.

La figura 4.2 describe en forma visual la técnica propuesta. La parte inicial del diagrama, etiquetada como *Estado Actual*, hace referencia a la etapa en la que un agente líder observa al ambiente y se dispone de tomar una decisión. Inicialmente, el conocimiento que el agente líder posee sobre el ambiente es reducido, por lo que requiere *explorarlo* para aprender de él, es decir, debe elegir una acción disponible al azar para observar posteriormente el impacto de ésta. En el contexto de la técnica, un líder tiene el papel de abrir una subasta, por lo que su decisión o acción es elegir la tarea a subastar. La siguiente parte del diagrama consiste de la participación de los agentes subordinados en la subasta recién creada y de la posterior selección de un ganador, que será asignado para realizar la tarea subastada. La etapa final consiste en la observación, por parte del agente líder, del impacto de esta elección. Dicho impacto no es inmediato, por lo que hay que decidir por cuánto tiempo en el futuro se deben observar las consecuencias. En nuestra propuesta, hemos elegido observar las consecuencias hasta que la tarea haya sido completada o el recurso se vuelva incapaz de completarla por cualquier motivo.

4.2 Consideraciones especiales

Mientras el tamaño del ambiente y la heterogeneidad aumentan, la calidad de las decisiones en etapas tempranas se vuelve más relevante. Esto obedece a que una asignación de baja calidad (es decir, una asignación que afecte el poder adquisitivo de los compradores en el mercado) afectará negativamente a todas las subastas futuras. El espacio de búsqueda también se ve afectado por lo anterior. Cuando las tareas o los recursos son homogéneos, la calidad de las asignaciones es equivalente entre ellas si no hay algún otro factor, independiente de su tipo, que influya en estas. Es decir, en dominios donde los recursos o tareas ocupan un punto o región en un espacio multidimensional (como un conjunto de robots en un área de exploración de dos o tres dimensiones), la calidad de las asignaciones se ve afectada por la posición,

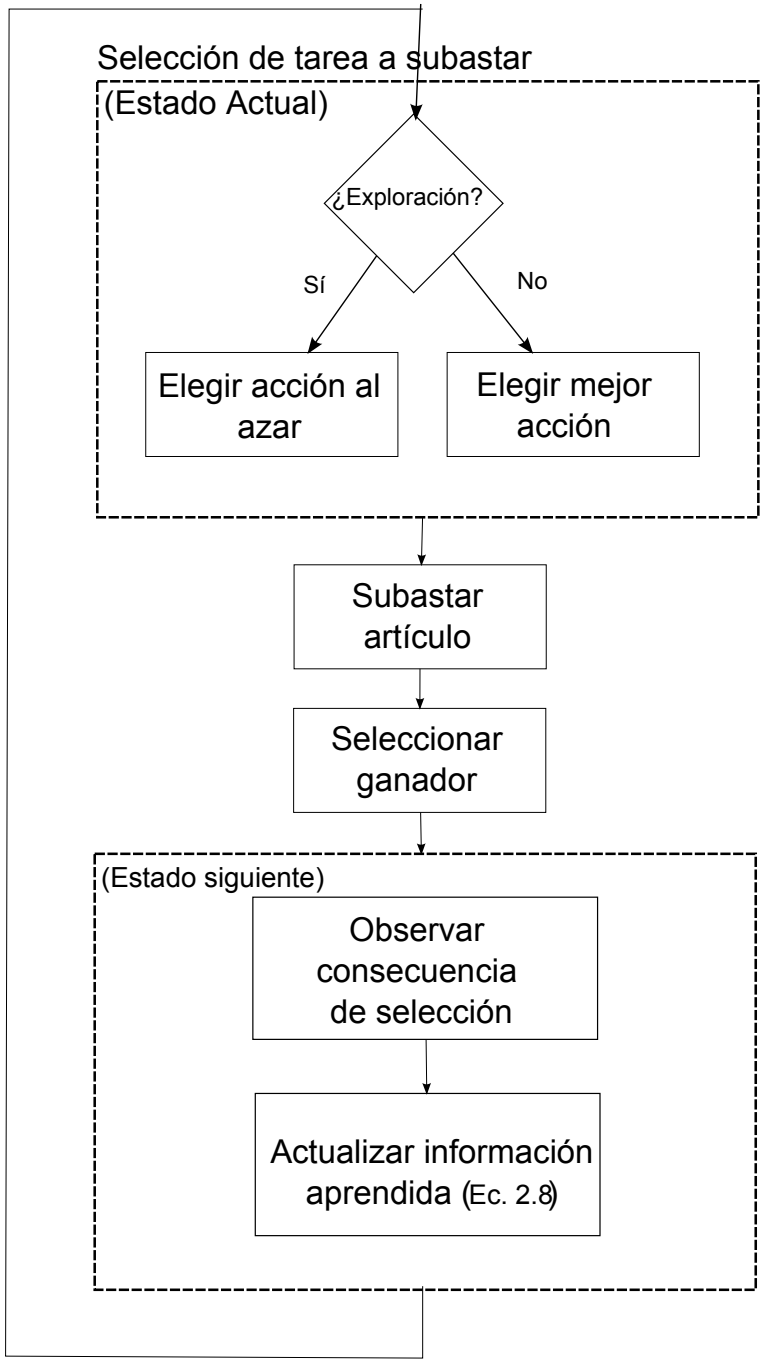


Figura 4.2: Funcionamiento de la técnica propuesta.

a pesar de que todos los robots sean iguales. En otras palabras, la homogeneidad - en el contexto económico - significa que el tipo de los compradores no influye en el interés que estos reflejan por los artículos (ya que es el mismo en todos). El tamaño del ambiente, a su vez, influye directamente en el número de subastas y en el número de participantes en estas, por lo que el tamaño del espacio de asignaciones también crece pero de forma exponencial con respecto al número de recursos y tareas. A causa de este crecimiento, se aplica una representación especial que busca frenar el ritmo de crecimiento del espacio de búsqueda. Aunque un sistema multiagente exige una representación multiagente en el formalismo adoptado (en este caso, un MDP) para representar el ambiente y las interacciones entre los agentes de forma correcta, en su lugar elegimos conservar la representación MDP convencional de un solo agente y a su vez dividir el ambiente en *regiones*. Cada una con un agente correspondiente que aprende sobre esta parte del ambiente. De esta manera, se aprovecha el enfoque recursivo de la jerarquía limitando el espacio de aprendizaje sólo a una región, haciendo el proceso de aprendizaje de dicha región más rápido. Hay que recalcar que el término ambiente y región del ambiente se usarán indiferentemente desde ahora.

Tanto la representación, el aprendizaje, la creación de las subastas y la formulación de sus respectivas ofertas se describen formalmente en las siguientes secciones.

4.3 AMBIENTE

El escenario general del problema de asignación de tareas consiste de un conjunto R con n recursos *heterogéneos* y un conjunto T de m tareas *heterogéneas*. Cada una de las tareas y recursos posee un identificador único y puede poseer requerimientos y capacidades únicas, respectivamente.

En cada instante de tiempo, pueden ocurrir los siguientes fenómenos:

- Pérdida parcial o total de un recurso.

- Terminación parcial o total de una tarea.
- Llegada de un recurso nuevo.
- Disponibilidad de un recurso previamente asignado.
- Llegada de una tarea nueva.
- Disponibilidad de una tarea previamente asignada.

Recordemos que, formalmente, el ambiente se encuentra descrito en la definición 11.

Usaremos la notación R_x para indicar el estado de un conjunto cualquiera R en el instante de tiempo x, también denotaremos al conjunto de todas las asignaciones en un instante como ϕ_x o ϕ_k. Inicialmente, no hay asignaciones entre recursos y tareas, por lo que $\phi_0 = \emptyset$.

En las siguientes secciones describiremos formalmente cómo perciben los agentes líderes al ambiente y qué acciones pueden realizar sobre él. De esta forma, podremos definir finalmente el MDP de los líderes (sección 2.1.2).

4.3.1 REPRESENTACIÓN DEL AMBIENTE PARA AGENTES LÍDER

En esta sección, enunciaremos de manera formal las características del ambiente que puede percibir un agente líder y la información que puede aprender de esta percepción. Las características descritas son el resultado de haber aplicado una transformación de representación sobre el ambiente para simplificarlo. Esto tiene como objetivo evitar que la complejidad original del ambiente se traduzca en un crecimiento exponencial tanto del espacio de estados (todas las posibles maneras en que el agente líder puede percibir al ambiente) como del espacio de acciones, de manera que el agente líder o sofisticado se enfoque solamente en tomar decisiones únicas (es decir, acciones no duplicadas, como se mencionó en la sección 2.1.3) sobre un espacio

de estados reducido. El MDP para el agente sofisticado está descrito formalmente de la misma forma que en la definición 1.

Recordando el algoritmo de aprendizaje por refuerzo visto en la sección 2.1.3 y la formulación de un problema de decisión de Markov (MDP) en la sección 2.1.2, se necesita una representación acorde del ambiente. El algoritmo *Q-Learning* modela el conocimiento del ambiente percibido por *un agente* a través de una matriz que representa la calidad de una pareja *estado, acción* en un *MDP*. Esto es, se aprende un valor estimado de cada estado y acción. Los agentes encargados de seleccionar una tarea serán los únicos modelados a través del aprendizaje por refuerzo, mientras que el resto de agentes y las tareas serán tomados en cuenta como *características del ambiente*, es decir, forman parte de la descripción del *estado* del mismo. De forma similar, la descripción de la *acción* considera las tareas visibles por el agente que no se encuentran asignadas.

ESTADO

La descripción del ambiente para el agente líder (aquel que selecciona tareas, las transforma en subastas y realiza asignaciones), incluye a los agentes del nivel inferior de la jerarquía que se encuentran disponibles y a todas las tareas que no han sido asignadas aún. La formalización del ambiente que observa el agente sofisticado es la misma que la descripción del ambiente general en la sección pasada de este capítulo. Si analizamos la cardinalidad de las asignaciones posibles en un único instante de tiempo x, podemos observar que el crecimiento es exponencial. De forma más detallada, tenemos:

- $|R| = 1$, $|T| = 1$, entonces $|\Phi_x| = (2^1 - 1) \times 1 = 1$
- $|R| = 2$, $|T| = 1$, entonces $|\Phi_x| = (2^2 - 1) \times 1 = 3$
- $|R| = 2$, $|T| = 2$, entonces $|\Phi_x| = (2^2 - 1) \times 2 = 6$

- $|R| = 3$, $|T| = 2$, entonces $|\Phi_x| = (2^3 - 1) \times 2 = 14$
- $|R| = 4$, $|T| = 4$, entonces $|\Phi_x| = (2^4 - 1) \times 4 = 60$
- $|R| = 5$, $|T| = 5$, entonces $|\Phi_x| = (2^5 - 1) \times 5 = 155$
- ...

En general,

$$|\Phi_x| = (2^{|R|} - 1) \cdot |T|$$

Si añadimos la consideración de los siguientes instantes de tiempo, el espacio de todas las asignaciones posibles aumenta de forma incluso más acelerada. Por ejemplo, en un escenario poco realista con 5 tareas y 5 recursos inicialmente, en el que cada instante de tiempo se descarta un recurso y una tarea hasta no tener nada al final, alcanzaremos un espacio de búsqueda de hasta $1,171,800$ posibles secuencias de asignaciones. De manera específica, la complejidad del problema de asignación de tareas en un *ambiente dinámico*, con asignaciones de múltiples recursos a una tarea, está acotado por: la cantidad de tiempo x en el que se extiende la asignación, el número de recursos n y el número de tareas m, es decir, $\mathcal{O}\left((2^n \cdot m)^x\right)$.

Para evitar el crecimiento acelerado del espacio de parejas (*estado, acción*) causado por una representación directa del ambiente, proponemos una categorización previa de recursos y tareas, y posteriormente un conteo de tipos *disponibles*. Definiendo un conjunto de tipos de recursos Ψ_R y un conjunto de tipos de tareas Ψ_T, procedemos a describir formalmente una representación más compacta del ambiente.

Definición 12. ***(Categorización y conteo de tipos)*** *Dado un conjunto de recursos R, un conjunto de tareas T, un conjunto de tipos de tareas Ψ_R indizado por l, un conjunto de recursos Ψ_T indizado por m,*

- *Una función de categorización para recursos Γ_R tiene la forma $\Gamma_R : R \to \Psi_R$, que mapea un recurso a un tipo $\psi \in \Psi_R$.*

- *Una función de categorización para tareas Γ_T tiene la forma $\Gamma_T : T \to \Psi_T$, que mapea una tarea a un tipo $\psi \in \Psi_T$.*

- *Un conteo de tipos para tareas es una función de la forma $\delta_T : \Psi_T \to \mathbb{Z}^+$ que mapea un tipo de tareas a un número entero positivo. Esta función cuenta cuántas tareas pertenecen a un tipo $\psi \in \Psi_T$ y regresa el conteo.*

$$\delta_T^{\psi} = \sum_{t \in T} I_T \left(\Gamma_T \left(t \right), \psi \right)$$

para alguna $\psi \in \Psi_T$. Podemos utilizar también, gracias al índice l sobre Ψ_R, la notación δ_R^l para todo $l \in \Psi_R$.

- *Un conteo de tipos para recursos es una función de la forma $\delta_R : \Psi_R \to \mathbb{Z}^+$ que mapea un tipo de recursos a un número entero positivo. Esta función cuenta cuántos recursos pertenecen a un tipo $\psi \in \Psi_R$ y regresa el conteo.*

$$\delta_R^{\psi} = \sum_{r \in R} I_R \left(\Gamma_R \left(r \right), \psi \right)$$

para alguna $\psi \in \Psi_R$. Podemos utilizar también, gracias al índice m sobre Ψ_T, la notación δ_T^m para todo $m \in \Psi_T$.

- *Las funciones identidad de la forma $I_T : \Psi_T \times \Psi_T - > \{0, 1\}$ y $I_R : \Psi_R \times \Psi_R - > \{0, 1\}$. Ambas se definen como*

$$I\left(\psi, \psi'\right) = \begin{cases} 1 & \psi = \psi' \textit{y el recurso/tarea no está asignado} \\ 0 & \textit{en otro caso} \end{cases}$$

Dada una categorización y conteo de recursos y tareas $\langle T, R, \Psi_T, \Psi_R, \Gamma_T, \Gamma_R \rangle$, un par de funciones de conteo δ_T y δ_R, el conjunto de estados S del MDP de un agente sofisticado (definición 1), se define como: $s = \left(\left(\delta_R \right)_{r \in R}, \left(\delta_T \right)_{t \in T} \right)$, donde $\left(\delta_R \right)_{r \in R} = \left(\delta_R^1, \cdots, \delta_R^l \right)$ y $\left(\delta_T \right)_{t \in T} = \left(\delta_T^1, \cdots, \delta_T^m \right)$.

La consideración de tipos aunado a la clasificación de las tareas y los recursos, restringe el crecimiento del espacio de búsqueda de manera significativa. El crecimiento del espacio, en lugar de depender directamente de la cantidad de recursos o tareas (recordemos que cada tarea o recurso es único, por lo que la llegada de nuevos recursos o tareas se traduce directamente en un aumento exponencial de estados) está ahora limitado a la cantidad de tipos en K y la cantidad de recursos y tareas disponibles en un instante. Esto significa que, en la representación original, 2 o más estados a lo largo del tiempo pueden ser mapeados a un mismo estado en la representación propuesta.

Podemos calcular el número de estados mediante el cálculo del número de formas en las que podemos representar un estado. Sea n el posible número de recursos, m el posible número de tareas, k tipos de tareas, l tipos de recursos, con un tamaño de escenario $n+m$, el número de estados es $C\left(k+l+n+m, n+m\right)$. Este cálculo es equivalente a encontrar el número de soluciones enteras de la desigualdad $x_1 + x_2 + \ldots + x_{k+l} \leq n+m$, donde x_i es cada tipo de tareas o recursos, $k+l$ es el número total de tipos en los que pueden ser clasificados los recursos y tareas, finalmente $n+m$ es el número total de tareas y recursos presentes [Grimaldi, 2003, p. 37,38]. Intuitivamente, se está contando el número de formas en que podemos representar un escenario de tamaño $n+m$, clasificando n recursos en l tipos y m tareas en k tipos, considerando el cambio (decreciente, suponiendo que no habrá tareas ni recursos nuevos) en el tiempo. Aplicando este conteo al ejemplo anterior, suponiendo 5 tipos para recursos y tareas (el peor caso), tenemos $C\left(5+5+5+5, 5+5\right) = C\left(20, 10\right) = 184,756$ estados, de los cuales sólo se considerarán una pequeña parte, debido a que en el cálculo suponemos que los recursos y tareas han sido distribuidos en tipos de todas las maneras posibles porque no sabemos de qué tipo son los recursos y tareas que habrá en el ambiente, pero en la práctica esto sólo sucede si los recursos o tareas llegan de forma aleatoria en un tiempo infinito.

ACCIÓN

Una acción se refiere a una de las posibilidades que el agente, representado por el *MDP*, puede elegir para modificar el ambiente y generar una transición de estados. Para limitar aún más el tamaño del espacio de estados y acciones, la formulación de las acciones depende completamente del conjunto K, por lo que se mantiene fijo a través del tiempo. Formalmente, el conjunto de acciones se simplifica a $A = K$. Recordando la sección 2.1.3, se tiene que si los agentes tienen acciones no duplicadas, la complejidad del algoritmo de aprendizaje por refuerzo se acota por el número de estados n de forma $\mathcal{O}(n^3)$, que es lo que logra esta clasificación.

En la etapa de *explotación* del aprendizaje por refuerzo, el tipo elegido corresponderá al que posea el mayor valor Q en la matriz del algoritmo *Q-Learning*. En la figura 4.3 se muestra gráficamente un ejemplo de la transformación del ambiente en un instante de tiempo, a su representación en estado y acción. El ejemplo muestra a un ambiente de 5 recursos y 5 tareas en un instante de tiempo (figura 4.3(a)), cada uno de estos con un color que representa su tipo. La representación del estado se construye haciendo un conteo de los recursos y tareas de cada tipo (figura 4.3(b)), mientras que el conjunto de acciones disponibles para el agente sofisticado corresponde a la selección de los tipos de las tareas que se encuentren disponibles (o sin asignar) en el ambiente. Sin embargo, en la figura, se simplifica el conjunto de acciones para que corresponda al conjunto de todas las tareas. Este detalle no tiene repercusiones, ya que el agente sofisticado no puede seleccionar un tipo que no exista en el conjunto de tareas.

RECOMPENSA

Recordando la definición de un MDP (definición 1), se requiere definir la función r que proporciona la recompensa inmediata por haber llegado a algún estado s del ambiente. Dicha función depende del ambiente y del *objetivo* del aprendizaje, es

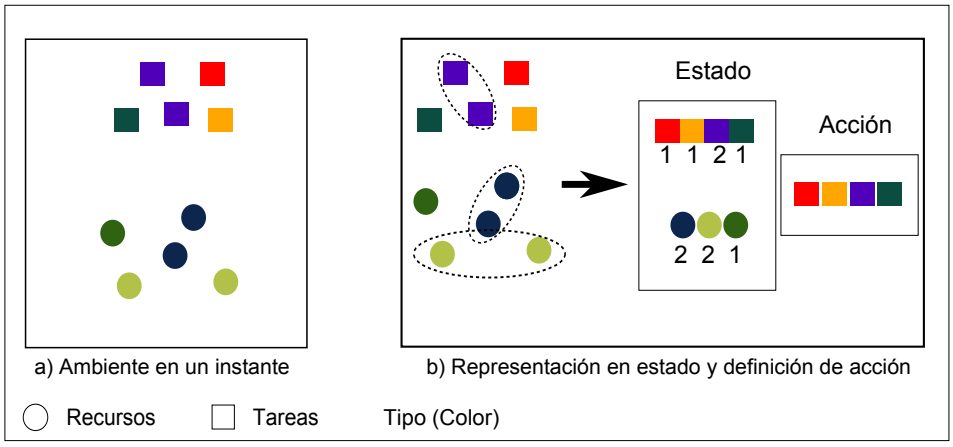


Figura 4.3: Representación del ambiente en un MDP.

decir, la recompensa inmediata debe *indicar* el impacto positivo o negativo específico del fenómeno que se desea aprender y la influencia que ha tenido sobre éste la acción elegida por el agente. En muchos ambientes dinámicos dada su naturaleza cambiante, la transición de un estado a otro no es inmediata, por lo que su recompensa tampoco lo es y debe haber mecanismos para identificar cuál es la consecuencia de la acción y en qué momento debe evaluarse esta recompensa. La definición de la función de recompensa en el MDP la definimos de forma *retrasada*, es decir, se calcula en un instante de tiempo posterior *indeterminado* a la asignación entre un recurso y una tarea, debido a que no podemos medir la consecuencia de haber elegido una tarea u otra de forma inmediata.

Definición 13. ***(Recompensa)*** *La función de recompensa r del MDP $\langle S, A, p, r\rangle$, es de la forma $r : S \times A \to \mathbb{R}$ que mapea una pareja estado-acción a un número real. De forma específica, la función de recompensa extrae de la información del estado s y de la acción a, el recurso n que fue asignado a la tarea t, además se recaba el instante $x \in X$ que indica el momento en que se hizo la asignación entre el recurso $n \in N_x$ y la tarea $t \in T_x$, y el instante $y \in X$, que indica el momento en que la asignación deja de ser válida (es decir, el recurso $n \notin N_y$ o $t \notin T_y$).*

4.4 SUBASTAS

Una vez que el proceso de selección ha terminado, el agente *sofisticado* busca la primera tarea *disponible* cuyo tipo concuerde con el tipo seleccionado. Esto inicia la fase de *apertura de mercados*, en la que la tarea elegida se vuelve el *artículo* a *subastar*. Todos los agentes del nivel inferior de la jerarquía, *agentes subordinados*, *postores* o *recursos*, *pujan* o envían su oferta al agente sofisticado. El agente ganador es asignado a la tarea por la que ofertó. Esta tesis propone dos tipos de subastas: subastas paralelo-secuenciales y secuenciales inversas, debido a un fenómeno muy particular. Las subastas descritas en la sección 2.3 asumen que el conjunto de postores y el conjunto de artículos en venta es *fijo* (a excepción de las subastas secuenciales), por lo que la aplicación de alguno de estos tipos de subastas, cuando hay menos recursos que tareas, provoca que todos los recursos sean asignados en un instante de tiempo, ignorando las posibles tareas futuras que puedan ser encontradas en un ambiente dinámico, provocando la degradación de la calidad de las soluciones. Para lidiar con éste fenómeno, proponemos las subastas que detallamos a continuación.

4.4.1 SUBASTAS PARALELO-SECUENCIALES

Las subastas paralelo-secuenciales están inspiradas en las *subastas de posición* y en las *subastas paralelas*, recordando la sección 2.3. En esta subasta se busca vender un conjunto de artículos. El procedimiento consiste en el anuncio, por parte del subastador, de un solo artículo inicialmente. Posteriormente, todos los compradores en el mercado envían una oferta, que representa la *preferencia* o el valor que le da cada comprador al artículo subastado. Posteriormente, el subastador recibe todas las ofertas y elige como *único* ganador al que haya enviado la *oferta más alta*. Finalmente, el subastador le otorga el artículo. En el contexto de la asignación de tareas, el subastador realiza la asignación del recurso ganador a la tarea subastada y espera un tiempo breve para que el estado del mercado se estabilice, es decir, para que

el ganador haga uso del artículo ganado y el resto de compradores actualicen sus criterios en las futuras subastas después de haber visto rápidamente la consecuencia de la asignación. El proceso se repite con el siguiente artículo seleccionado para subastar hasta que todas las tareas sean subastadas o hasta que una nueva tarea sea descubierta. De las nuevas subastas se excluyen a todos los recursos que ya tienen tareas asignadas. El precio que se paga es *dependiente del dominio del problema*. En la figura 4.4(b) se muestra un ejemplo de las posibles asignaciones resultantes de esta subasta, en un ambiente con cinco tareas y cinco recursos. Se incluye también un ejemplo de las subastas paralelas (figura 4.4(a)) para poder visualizar la diferencia entre la subasta propuesta y la subasta que se encuentra en la literatura que ha servido de inspiración.

4.4.2 SUBASTAS SECUENCIALES INVERSAS

Las subastas *secuenciales inversas* se inspiran en las *subastas secuenciales* y en las *subastas inversas*. Recordando a las subastas inversas de la sección 2.3, estas invierten roles entre compradores y subastadores. Por otra parte, las subastas secuenciales permiten a los compradores ofertar y ganar más de un artículo. Proponemos una subasta que combina ambos mecanismos para que los recursos tomen el papel de los artículos en venta, las tareas se vuelvan los compradores y las asignaciones finales puedan incluir el caso de que varios recursos puedan ocuparse de una sola tarea. A diferencia del resto de las subastas, en este caso, el agente sofisticado elige a la menor oferta como la ganadora. El resto del mecanismo de las subastas secuenciales inversas es idéntico a las subastas paralelo-secuenciales, es decir, se anuncia el artículo a vender (recurso), los compradores envían sus ofertas al agente sofisticado, este anuncia al ganador y realiza la asignación. El recurso vendido se encarga de ejecutar la tarea que ganó la subasta. Después de una breve pausa, se anuncia el siguiente artículo y el proceso se repite. Sin embargo, las tareas que ya están asignadas no se excluyen de la subasta. De esta manera, cada tarea puede ganar más de

un recurso y ser asignada a un grupo de recursos simultáneamente. Para evitar que se formen grupos muy grandes de recursos, puede establecerse un número máximo de recursos en un grupo o utilizar el cálculo de la oferta para provocar que grandes grupos formulen ofertas poco atractivas para el agente líder. En la figura 4.4(c) y 4.4(d) se muestran ejemplos de las posibles asignaciones resultantes de las subastas secuenciales y secuenciales inversas.

4.4.3 OFERTAS

La formulación de las ofertas es, de igual manera, dependiente del problema. Sin embargo, se pueden enunciar algunas sugerencias básicas para indicar la preferencia de un agente.

- El ganador de la subasta es aquel cuya oferta sea la que representa la mayor preferencia de un agente con el artículo en venta.
- Las ofertas representan el valor objetivo que se desea minimizar o maximizar. Por ejemplo, si se desea minimizar el tiempo, ofertas altas deben reflejar a un agente que evalúe que puede completar la tarea en menos tiempo.

4.5 RESUMEN

La técnica propuesta consiste de las siguiente etapas que se repiten hasta que todas las tareas han sido asignadas y terminadas, ya no haya recursos disponibles o la llegada de nuevas tareas y/o recursos finalice.

1. *Particionamiento del ambiente.* La técnica comienza con la división de los recursos en grupos o la asignación de nuevos recursos en algún grupo ya existente. Los criterios para esta división o asignación dependen del ambiente, pero no

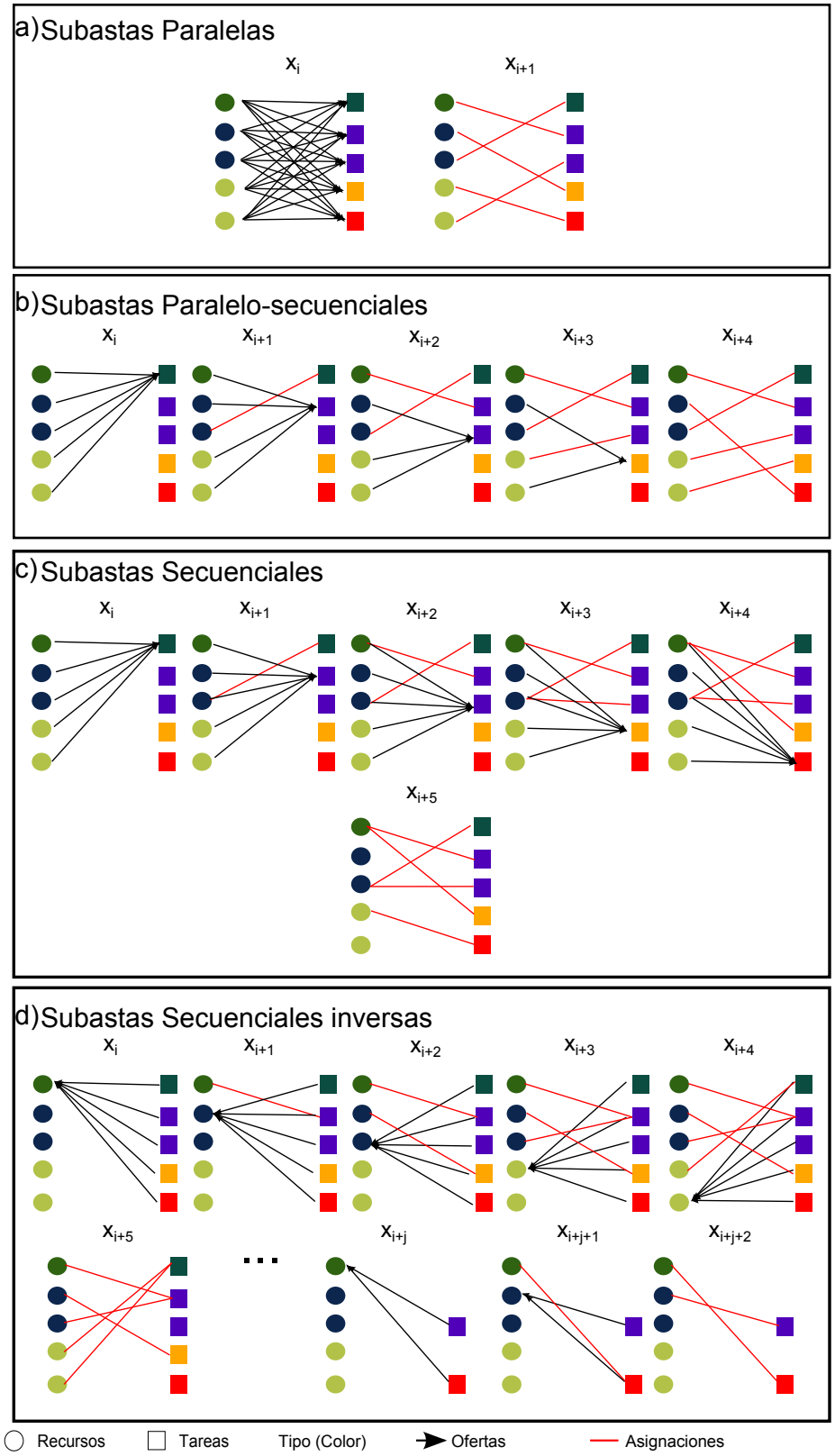


Figura 4.4: Asignaciones resultantes por tipo de subastas.

requieren de demasiada especificidad dado que sólo es necesario saber si existe alguna condición que mejore la interacción entre agentes. Si esta no existe, una división al azar es suficiente.

   a) *Detección y particionado de tareas.* Una vez que los recursos han sido divididos, se comienza con la detección de tareas. Este proceso también depende del ambiente. En alguna casos, se requiere de un proceso de exploración. En otros casos, este proceso no es necesario debido a la existencia de algún mecanismo que suministre las tareas.

   b) *Emparejamiento de conjuntos de tareas y recursos.* Al tener identificadas las tareas, es necesario emparejarlas con un conjunto de recursos usando un criterio similar al usado cuando se agrupan recursos.

2. *Clasificación de recursos y tareas.* Una vez que los grupos se han hecho, se procede a simplificar la forma en que los agentes líderes perciben el ambiente. Esto se hace mediante la clasificación de tareas y recursos en tipos, aplicando la definición 12.

3. *Creación de mercados.* Posterior a la clasificación de las tareas y los recursos, los líderes de cada grupo (seleccionados al azar o por cualquier otro criterio) comienzan a crear mercados, eligiendo un tipo de tarea al azar o el que mejor resultados le ha dado (recordando el dilema de exploración/explotación del aprendizaje por refuerzo). Después de haber hecho la elección, el agente líder selecciona una tarea del tipo correspondiente y la pone en subasta.

4. *Subastas.* Los agentes subordinados participan en la subasta, formulando sus ofertas por la tarea y enviando dicha oferta al líder de su grupo.

5. *Selección de ganador.* El líder del grupo recibe las ofertas y elige al ganador de la subasta. El ganador es asignado para realizar la tarea que subastó.

6. *Observación.* El líder observa la consecuencia de haber realizado la asignación

hasta el momento en que la tarea se ha completado o el recurso se vuelve incapaz de completarla. La información adquirida es aprendida y utilizada en el futuro. En el caso específico de la tesis donde se utiliza $Q - Learning$, el agente líder guarda dicha información en su matriz Q.

De forma visual, la figura 4.5 muestra el procedimiento de selección de tareas y creación de subastas que se lleva a cabo en la etapa de aprendizaje en un ambiente dinámico. Esto corresponde a los puntos 3, 4 y 5 del resumen anterior. Se puede observar que la figura representa la dinámica del ambiente en 5 instantes de tiempo. En la primera y última fila se señala la llegada de nuevos recursos y tareas que aparecerán disponibles en el siguiente instante de tiempo. En la segunda fila se muestra el proceso de subastas. Debajo de cada columna, se representa la selección de una tarea por parte del agente sofisticado. La asignación se realiza después de comparar todas las ofertas que recibe por la tarea que eligió (las asignaciones son señaladas por líneas rojas y las ofertas por líneas negras).

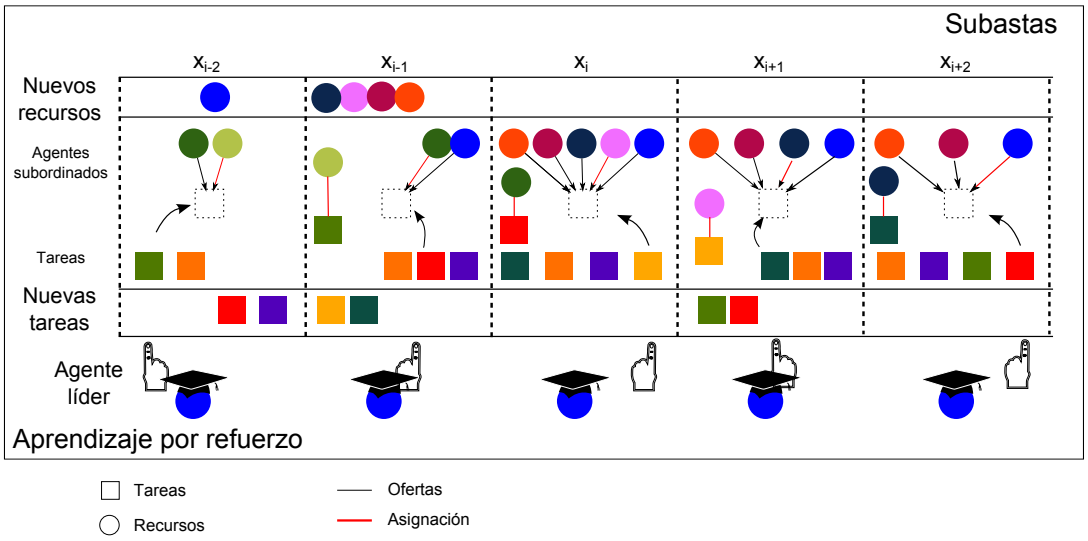


Figura 4.5: Proceso de aprendizaje sobre la subastas.

Capítulo 5

Experimentación

En este capítulo se presenta una descripción del ambiente de trabajo utilizado, así como su configuración específica para el desarrollo de la experimentación. También se especifican detalladamente los parámetros utilizados y las definiciones que, hasta el capítulo 4, permanecían dependientes del problema, como la función de clasificación, la especificación de tipos y la formulación de ofertas.

5.1 Ambiente de trabajo

Utilizamos un juego de estrategia en tiempo real como la plataforma de simulación, específicamente Starcraft [Blizzard®, 1998], la cual es ideal para simular ambientes dinámicos altamente complejos. Starcraft® es un juego de estrategia en tiempo real que permite controlar, por parte de los jugadores, un conjunto de unidades y estructuras que deben ser maniobradas y administradas para conseguir un conjunto de objetivos, ya sea el dominio de alguna área o la destrucción de activos enemigos.

Para tomar control del simulador, se utiliza una interfaz de programación de aplicaciones (API) desarrollada para interactuar con el simulador llamada *BWAPI*[Heinerm, 2011]. Al utilizar esta API, es posible tanto recuperar información de las unidades presentes como enviar una gran cantidad de instrucciones a estas. Pa-

ra tener un mayor control sobre el simulador, los ambientes dinámicos que ofrece Starcraft® se han limitado a escenarios de combate en donde se presenta un factor competitivo y son, a su vez, relativamente más rápidos que los escenarios de administración de recursos y construcción de estructuras.

Específicamente, el escenario de combate que instanciamos consta de un conjunto rival de unidades que tienen como objetivo destruir a cada una de las unidades enemigas. Cada unidad posee un conjunto de características que describen sus habilidades generales, tales como una cantidad específica de resistencia al daño, una cantidad específica de daño que puede causar a otras unidades, velocidad, etc. Adicionalmente, algunas unidades poseen características especiales que les dotan de ventajas sobre otras unidades o son capaces de realizar tareas que ninguna otra unidad es capaz de hacer. El término *tiempo real* se refiere a que no existe retraso entre la orden que un jugador emite y su correspondiente acción en el simulador, en contraste con ambientes donde ocurre una sola acción por instante o *turno*.

El modelado de un problema de asignación de tareas en este simulador se realiza considerando como *recursos* a aquellas unidades a las que puedan ser enviadas instrucciones y, por lo tanto, ser controladas por la técnica. Como *tareas* consideramos a las unidades sobre las que no poseemos control alguno y cuyo objetivo sea destruir a las unidades que la técnica controla.

Para medir la calidad de las asignaciones que resultan de nuestra técnica, se realiza la comparación con las *subastas secuenciales* [Schoenig and Pagnucco, 2011], una técnica distribuida utilizada para resolver el problema de asignación de tareas en sistemas multiagentes.

5.2 Descripción y clasificación de recursos y tareas

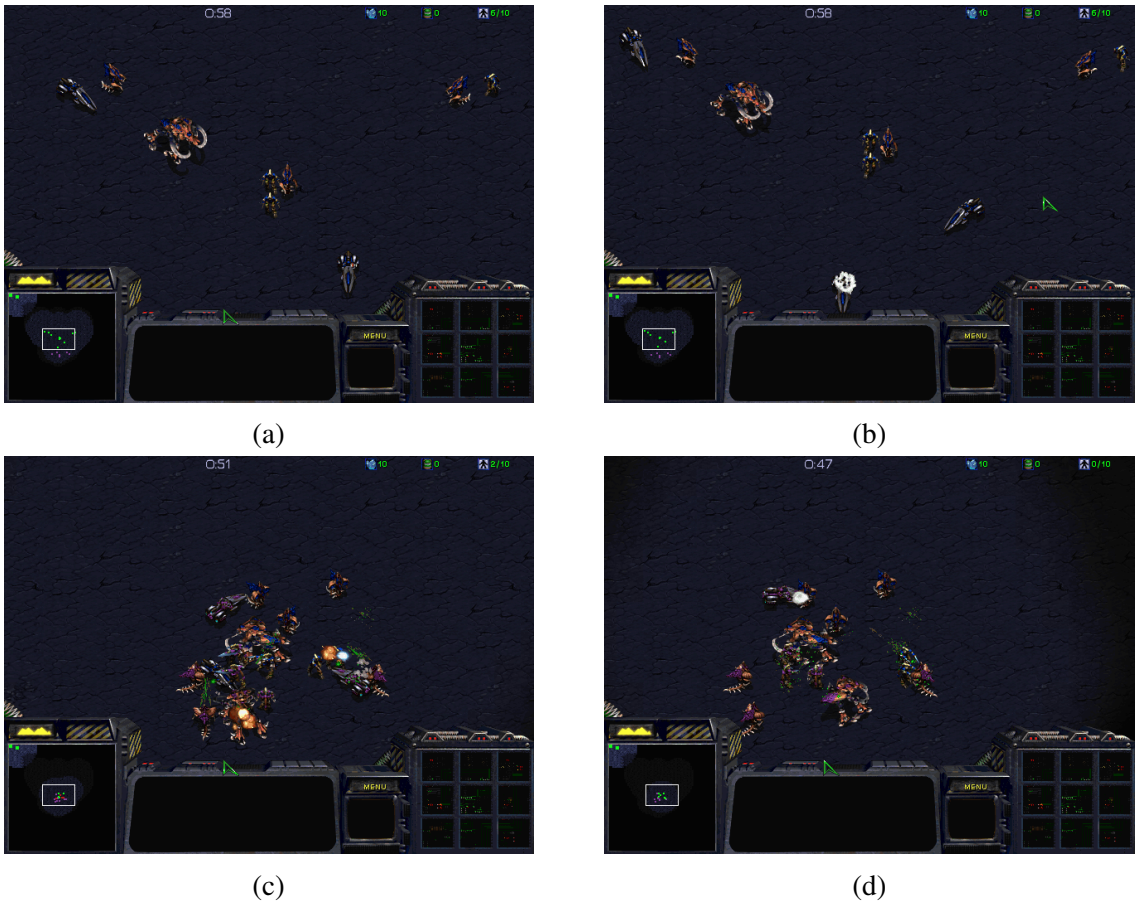
En el simulador, los recursos y las tareas tienen en común las siguientes capacidades:

- *Energia o salud.* Capacidad de resistir el daño que otras unidades pueden provocar.
- *Movimiento.* Capacidad para trasladarse de un lugar a otro.
- *Ataque.* Capacidad de provocar daño a otras unidades.

Cada recurso o tarea tiene las siguientes características, todas representadas como números enteros positivos y accesibles durante la ejecución siempre que la unidad sea visible:

- iHP_a es la salud que posee la unidad a en el momento de su creación.
- cHP_a es la salud que posee la unidad a en el momento de su observación.
- $Dam_{a,b}$ es el daño que causa la unidad a cuando ataca a una unidad b.
- $TipDam_a$ es el tipo de arma que posee la unidad a.
- Arm_a es una protección parcial al daño que recibe la unidad a.
- $TipArm_a$ es el tipo de protección parcial que posee la unidad a.
- Vel_a es la velocidad con la que la unidad a se mueve a través del escenario.
- $Rang_a$ tipo de ataque que la unidad a realiza: corto o largo alcance.

Debido a la variedad de estas características, un recurso puede ser más o menos efectivo que otro al ser asignado a una tarea. En la figura 5.1 se muestran algunas imágenes del simulador en funcionamiento. Se puede apreciar el movimiento de las unidades y su comportamiento (el ataque) cuando el grupo de unidades azules se acerca al grupo de unidades violetas.



(a) (b)

(c) (d)

Figura 5.1: Capturas de Starcraft®

Para realizar la clasificación inicial de los recursos y tareas, recordando la sección 4.3.1, se tomaron en cuenta los tipos que ofrece por defecto el simulador. Estos son 5 tipos y agrupan unidades que poseen el mismo conjunto de características previamente mencionadas. Estos tipos son *zealots, marines, vultures, hydralisks y ultralisks*. La función de clasificación (definición 12) verifica estas características y proporciona el tipo del recurso o tarea que se le suministró.

5.3 OFERTAS

El proceso por el cual los agentes subordinados calculan su oferta, es decir la representación numérica de su interés o preferencia por un artículo ante el agente sofisticado se define por partes, y se detalla a continuación. Cada una de estas partes presenta alguna característica o mezcla de características (descritas en la sección anterior) del agente ponderada con alguna característica de la tarea por la que oferta, con el fin de representar numéricamente su preferencia. La preferencia de cada agente trata de reflejar su percepción sobre cuán útil sería al ser asignado a esa tarea. Hay que remarcar que estas definiciones se refieren a las ofertas de las subastas *paralelo-secuenciales*. Las ofertas de las subastas *secuenciales inversas* son idénticas, a excepción de que se intercambian los conjuntos de tareas por los de recursos. Los parámetros de las distribuciones utilizadas en algunas definiciones se derivan de las características propias de las unidades del simulador.

Definición 14. ***(Proporción de salud)*** *La proporción de salud, denotada por Hq, se define como el cociente entre la salud de los recursos y la salud de la tarea. Se define como:*

$$Hq = \frac{\sum_{r \in R_x^w} cHP_r}{cHP_t} \tag{5.1}$$

donde:

- *t es la tarea subastada.*
- *R_x^w es el conjunto de recursos que pujan en el grupo w, en un instante x.*

Definición 15. ***(Tiempo de terminación)*** *El tiempo de terminación, denotado por Ft, se define como la cantidad de tiempo necesario para que el recurso (o recursos) pueda terminar la tarea, o la tarea destruya al recurso. Se define como:*

$$Ft = \frac{cHP_t}{\sum_{r \in R_x^w} Dam_{r,t}} \tag{5.2}$$

donde:

- r *es la tarea subastada.*
- R_x^w *es el conjunto de recursos que pujan en el grupo* w *en un instante* x*.*

Definición 16. ***(Daño promedio)*** *El daño promedio, denotado por* Ad*, se define como el cociente entre el daño que puede realizar un recurso (o recursos) a una tarea y la cantidad de recursos que participan en la subasta. En el caso de las subastas paralelo-secuenciales, la cantidad de recursos que formulan la oferta siempre es 1. Se define como:*

$$Ad = \frac{\sum_{r \in R_x^w} Dam_{r,t}}{|R_x^w|} \tag{5.3}$$

donde:

- t *es la tarea subastada.*
- R_x^w *es el conjunto de recursos que pujan en el grupo* w *en un instante* x*.*

Definición 17. ***(Componente de salud)*** *El componente de salud, denotado por* Hc*, se refiere a la influencia de la proporción de la salud entre el recurso que oferta y la tarea subastada en la oferta final. Se define como:*

$$Hc = Hq \sim \mathcal{N}(1.4, 0.25) \tag{5.4}$$

donde $\mathcal{N}(\mu, \sigma^2)$ *es la distribución normal con media* μ *y varianza* σ^2*. Los parámetros propuestos surgen de estimar cuál es el cociente y la velocidad del crecimiento o decrecimiento más adecuados, tomando en cuenta la energía o salud (*HP*) de las unidades del simulador.*

Definición 18. ***(Factor de tiempo)*** *El factor de tiempo, denotado por* Tf*, se refiere a la influencia que tiene el tiempo en el que el recurso podría completar la tarea (tiempo de*

terminación Ft). Se define como:

$$Tf = 1 + \sin(\frac{\pi}{200}Ft + \frac{pi}{25}) \tag{5.5}$$

Esta función se propone debido a que el decrecimiento del valor de una función sinusoidal es más suave que el de una recta.

Definición 19. ***(Componente de daño)*** *El componente de daño, denotado por Dc, se refiere a la influencia del daño que causa el recurso a la tarea sobre la oferta. Se define como:*

$$Dc = \sin(\frac{\pi}{50}Ad) \tag{5.6}$$

Definición 20. ***(Oferta)*** *La oferta que un recurso le enviará al agente sofisticado y que representa su interés sobre la tarea se define como:*

$$Oferta = Hc + Tf \cdot Dc \tag{5.7}$$

donde:

- *Hc es el componente de salud del recurso y la tarea (ec. 5.4).*
- *Tf es el factor de tiempo de la tarea (ec. 5.5).*
- *Dc es el componente de daño del recurso (ec. 5.6).*

La definición de la oferta reúne los criterios mencionados de manera que cada uno contribuye de manera diferente.

La figura 5.2 muestra la influencia del componente de salud Hc, mientras cambia la proporción Hq. El componente influye más cuando la salud del recurso es ligeramente mayor que la salud de la tarea pero decae rápidamente mientras ésta aumenta. En otras palabras, un recurso cuya salud supera significativamente a la

de la tarea por la que subasta no tendrá un interés significativo (que se refleje en su oferta), debido a que esa ventaja podría ser más útil en algún otro objetivo más importante en el futuro.

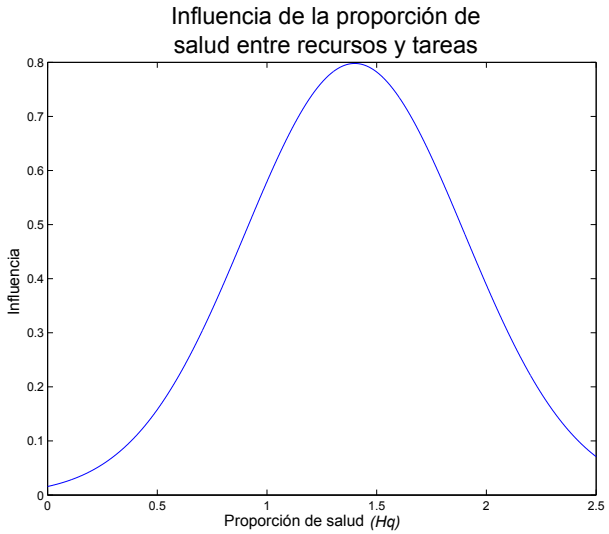


Figura 5.2: Representación gráfica de la ecuación 5.4.

La figura 5.3 muestra la curva que representa la influencia del tiempo de terminación Hq. Mientras más tiempo tarde el recurso en completar la tarea, menor interés se reflejará en su oferta.

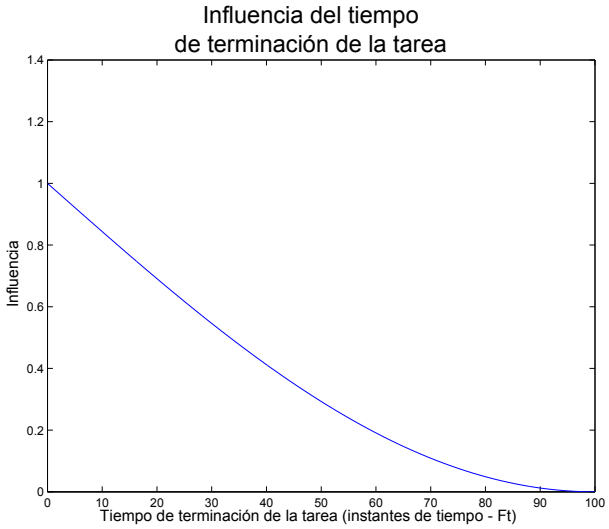


Figura 5.3: Representación gráfica de la ecuación 5.5.

La figura 5.4 muestra la curva que representa la influencia del daño promedio *Ad* sobre la oferta. Mientras más daño cause el recurso sobre la tarea, más interés reflejará en su oferta.

Figura 5.4: Representación gráfica de la ecuación 5.6.

5.4 RECOMPENSA

Recordando la notación en la definición de la recompensa (def. 13) y en la definición del MDP (def. 1), definimos la función de recompensa de la forma $r : S \times A \to \mathbb{R}$, como:

$$r(s,a) = (\sum_{n \in N_x} cHP_n - \sum_{n \in N_y} cHP_n) - (\sum_{t \in T_x} cHP_t - \sum_{t \in T_y} cHP_t) \tag{5.8}$$

De forma intuitiva, la recompensa representa la diferencia entre la pérdida de salud grupal de los recursos y la pérdida de salud grupal de las tareas. Con esta diferencia, intentamos medir la efectividad *global* de haber asignado un recurso o conjunto de recursos a una tarea. Es decir, la consecuencia que tuvo una sola asignación en todo el ambiente. La recompensa es positiva si los recursos perdieron menos salud que las tareas, esto es, la asignación ha sido benéfica si ha ayudado

a que la cantidad de desgaste que sufrieron los recursos sea menor a la cantidad de trabajo que realizaron para completar tareas. Ambos criterios (el desgaste y el trabajo realizado) son medidos por la característica *salud* de los recursos y las tareas.

5.5 Preparación de la experimentación

Para suministrar evidencia de que nuestra técnica cumple el objetivo principal planteado en el capítulo 1 es necesario realizar 2 *etapas de experimentación*. Con etapas de experimentación nos referimos al conjunto de experimentos necesarios para dar el primer paso en el proceso de suministrar evidencia para verificar o refutar la hipótesis planteada (sección 1.1). Es decir, para ofrecer evidencia de que la técnica propuesta es competitiva frente al estado del arte, realizamos una comparación, en la primera etapa experimental, de la calidad de las asignaciones resultantes entre nuestra técnica y una técnica distribuida para la asignación de tareas en ambiente dinámicos con tareas y recursos heterogéneos que no requiere conocimiento a priori del ambiente: subastas secuenciales [Schoenig and Pagnucco, 2011]. Para ofrecer evidencia con el fin de ayudar a refutar o demostrar la segunda parte de la hipótesis (que la técnica es escalable), en la segunda etapa experimental, comparamos los resultados de nuestra propia técnica usando grupos de diferentes tamaños (2 niveles de la jerarquía en la figura 4.1) con los resultados de nuestra propia técnica sin usar grupos y con los resultados de las subastas secuenciales previamente mencionadas.

En cada etapa de experimentación, configuramos un *escenario experimental*. Es decir, cada escenario representa la combinación de una *técnica*, un *tamaño de ambiente* y, en el caso de la segunda etapa experimental, un tamaño de grupo (que dicta en cuántas particiones se dividen los conjuntos de recursos y tareas). Los tamaños del ambiente y su descripción específica se muestran en la tabla 5.1. En la primera columna se muestra el nombre propio de las unidades dado por el simulador para identificarlas. Cada una de estas unidades posee diferentes valores en sus

características (definidas en la parte inicial de la sección 5.2). Las unidades, para los distintos escenarios, fueron elegidas al azar. Los escenarios (combinación entre técnica, tamaño del ambiente y número de grupos) correspondientes a la segunda etapa experimental se encuentran detallados en la tabla 5.2. A cada escenario le corresponden 5 *experimentos*. Cada experimento consta de 5000 *rondas* y cada *ronda* consiste de las siguientes etapas:

- El proceso de creación del conjunto de recursos y el conjunto de tareas con el tamaño especificado por el *escenario*. El ambiente donde se lleva a cabo cada ronda se muestra en la figura 5.5. Esta figura muestra el espacio que nos otorga el simulador para colocar y controlar las unidades. Los círculos amarillos indican las ubicaciones en donde aparecen los recursos al ser creados. De la misma forma, las cruces rojas representan la ubicación en donde se colocan las tareas al ser creadas. Desde estas ubicaciones, cada recurso y tarea se desplaza a diferente velocidad (que depende de la característica Vel_a de cada unidad) hacia las tareas (abajo) y hacia los recursos (arriba) respectivamente.
    - En el caso de la segunda etapa experimental, después de la creación del conjunto de tareas y recursos, se efectúa su particionamiento determinada por el tamaño del grupo que se especifica en la tabla 5.2.
- El proceso de exploración de los recursos, es decir, la etapa en la que los recursos se mueven por el ambiente para descubrir tareas.
- La creación de mercados, la realización de asignaciones y el aprendizaje de la experiencia adquirida durante esta ronda, para culminar con el cumplimiento de todas las tareas o la destrucción de todos los recursos.

Primero, para evidenciar que nuestra técnica ofrece mejores asignaciones que la técnica propuesta en [Schoenig and Pagnucco, 2011] (subastas secuenciales), compararemos el resultado final promedio de una secuencia de asignaciones obtenidas

Tamaño	Zealot	Marine	Vulture	Hydralisk	Ultralisk
5	2	0	1	1	1
10	3	0	3	3	1
15	3	4	3	4	1
20	4	5	4	5	2
25	5	7	5	5	3
30	6	8	6	6	4

Tabla 5.1: Detallado de unidades por tipo en los escenarios de diferentes tamaños.

por las subastas secuenciales y por la técnica propuesta sin particiones. La consideración de más particiones mejora la calidad de las asignaciones en ambientes más grandes, por lo que en esta primera etapa, no es necesario incluirlas. Segundo, para dar evidencia de que la técnica es escalable, es necesario mostrar que la calidad de los resultados se mantiene (o sufre una degradación mínima) en ambientes más grandes a los suministrados por nuestra propia técnica, sin particiones. También se hace la comparación con los resultados obtenidos en las subastas secuenciales para proveer más evidencia de lo competitivo de nuestra técnica.

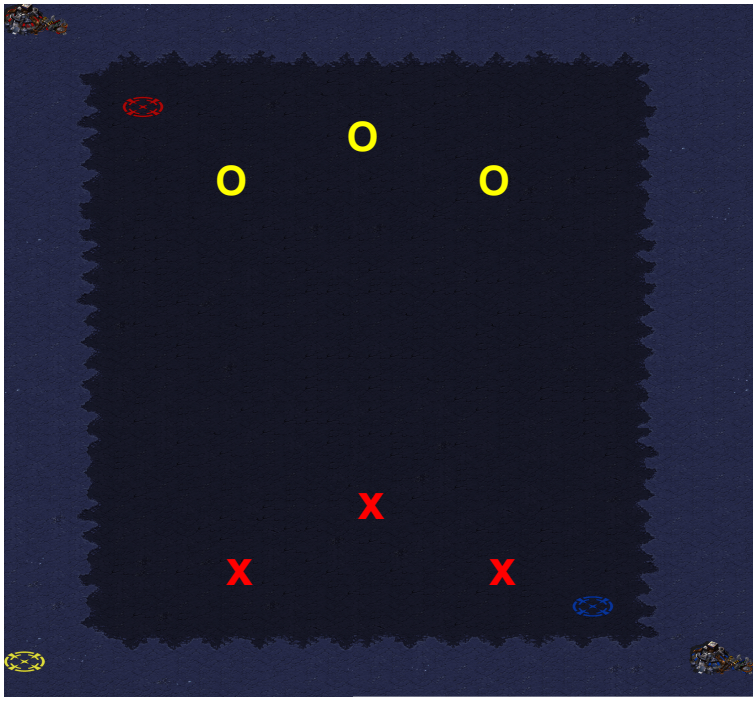


Figura 5.5: Ambiente de simulación.

Grupo / Tamaño	15	20	25	30
5	X	X	X	X
10		X		X
15				X

Tabla 5.2: Especificación de los escenarios para la segunda etapa experimental (para subastas paralelo-secuenciales y secuenciales inversas).

Al terminar una ronda, se comienza una nueva sin olvidar lo aprendido en la ronda anterior.

Los parámetros utilizados en la técnica, específicamente para el agente sofisticado, son los siguientes:

- $\alpha = 0.15$ Factor de aprendizaje. Se utiliza para el método de aprendizaje por refuerzo. Representa la importancia del conocimiento nuevo sobre el viejo.
- $\gamma = 0.95$ Factor de descuento. Se utiliza determinar la importancia del conocimiento adquirido hace s pasos (veces que se usa la ecuación 2.8 con el mismo estado siguiente) con una importancia γ^s. Por ejemplo, el conocimiento adquirido hace 2 pasos ($0.95^2 = 0.90$) es más importante que el adquirido hace 6 pasos ($0.95^2 = 0.74$).
- $\epsilon = 0.5$ Probabilidad inicial de selección de exploración (búsqueda de conocimiento) sobre explotación (uso del conocimiento adquirido). Utilizada para que la la selección sea aleatoria.
- $upd_{thr} = 4000$ Umbral en el que la probabilidad de exploración en la selección de la acción se vuelve 0. Utilizada para que en la etapa final de cada experimento se seleccionen las mejores acciones de acuerdo al conocimiento adquirido.

Los parámetros para la técnica *Q-Learning* fueron elegidos debido a que

5.5.1 MÉTRICAS

Los resultados de ambos experimentos (con dos y tres niveles en la jerarquía) se miden como la *diferencia entre la salud* conjunta de los dos grupos de unidades al terminar cada *ronda*.

$$Dif = \sum_{r \in R} cHP_r - \sum_{t \in T} cHP_t \tag{5.9}$$

donde:

- R es el conjunto de recursos.
- T es el conjunto de tareas.

Es decir, se realiza la sumatoria de la cantidad de salud de cada unidad que tuvo el conjunto de recursos y se le resta la sumatoria de la cantidad de salud de cada unidad que tuvo el conjunto de tareas, al finalizar el enfrentamiento. Esta diferencia representa el resultado final de toda una serie de asignaciones que terminan en éxito o fracaso para la técnica propuesta. El resultado mostrado de una ronda es el promedio de las diferencias de las 5 rondas (una ronda por experimento) para cada escenario de experimentación. Naturalmente, si los recursos logran completar sus tareas (en este contexto, significa destruir al conjunto de unidades enemigas) la diferencia es un número positivo. Por el contrario, si hay tareas que permanecen sin terminar (lo que significa que las unidades controladas por la técnica fueron destruidas) la diferencia es un número negativo.

5.5.2 CONSIDERACIONES DEL SIMULADOR

Tanto el simulador(Starcraft®), como las librerías externas (BWAPI® [Heinerm, 2011]) utilizadas para crear el sistema de software que lo manipulan, nos han ofrecido gran profundidad de personalización en la preparación del ambiente de trabajo. Algunas de sus ventajas fueron:

- Rutinas internas que controlan el movimiento y la ubicación de las unidades sin necesidad de librerías externas. Fueron de utilidad para establecer una conducta inicial y equilibrada de las unidades que sirviera como marco para la aplicación de las técnicas mencionadas en este trabajo. Este equilibrio se consiguió después de varios intentos de reubicación y cambio en la trayectorias de las unidades, de manera que el resultado de sus enfrentamientos, aplicando la métrica 5.9, se acercara a 0 siempre. Así, es fácil medir una ligera variación de la conducta de las unidades.
- Herramienta incluída para crear mapas propios. Esto nos permitió controlar el tamaño y, de forma un poco limitada, la selección de colores contrastantes para poder identificar con mayor facilidad los recursos y las tareas.
- Acceso, a través de librerías externas, al motor gráfico del simulador. Esto nos permitió imprimir datos en pantalla y trazar indicadores para facilitar la visualización de las asignaciones que se realizan en tiempo real. Fue vital para las labores de desarrollo y depuración del sistema.

Sin embargo, algunas de las desventajas son:

- Limitaciones de rutinas internas. Debido a la simpleza del lenguaje interno con el que cuenta el usuario, no es posible controlar todos los posibles parámetros de la creación de unidades. Por ejemplo, no se tiene control sobre la distribución de probabilidad cuando se crean las unidades de forma aleatoria, ni hay

manera de garantizar que la selección obedezca a una distribución uniforme. Por tal motivo, la selección de las unidades al azar, para la experimentación de diferentes escenarios, se realizó antes.

- Limitaciones en el movimiento de las unidades. La calidad de algunas asignaciones se ve afectada por la manera en la que las unidades se desplazan por el mapa, ya que surgen conductas impredecibles por cuestiones de colisiones en las trayectorias. Sin embargo, el impacto es menor.

A pesar de los problemas encontrados en el simulador, creemos que el ambiente tiene las características suficientes para poder realizar los experimentos de forma satisfactoria y así dar evidencia que ayude a demostrar o regutar la hipótesis que establecimos.

5.5.3 Resultados de la primera etapa de experimentación: Dos niveles de la jerarquía

En esta etapa experimental, el objetivo es verificar los beneficios de dotar a un agente con la capacidad de aprender sobre la dinámica de los mercados y las interacciones de los recursos y tareas al realizarse asignaciones entre estos. Se puede observar en las figuras de los resultados por técnica (figuras 5.6, 5.7, 5.8 y 5.9) que la técnica propuesta mejora en todos los casos a las subastas secuenciales, usando la métrica de comparación definida en la ecuación 5.9. Hay que notar que, mientras el tamaño del escenario crece, la calidad de las asignaciones decrece. Este fenómeno se ve con claridad en la figura 5.10 y significa que un modelo que sólo posea dos niveles de la jerarquía propuesta pierde gradualmente su propiedad de escalabilidad.

La mejora en los resultados que obtiene esta técnica, en esta etapa experimental, se debe a varios factores:

- La adaptación de *subastas paralelas* a ambientes dinámicos (*subastas paralelo-*

secuenciales). Debido a la generalidad de las subastas paralelas, su adaptación a ambientes dinámicos funciona bien en cualquier escenario. Las *subastas secuenciales* ([Schoenig and Pagnucco, 2011]), por otro lado, requieren de mucha especificidad en la formulación de su oferta para que puedan funcionar bien en ambientes dinámicos complejos. Con las *subastas secuenciales inversas* que proponemos, el requerimiento de especificidad disminuye, porque no es necesario saber de forma explícita la relación entre una tarea y otra a través del tiempo. Las *subastas secuenciales inversas* sólo aprovechan los mecanismos de las *subasta secuenciales* para poder hacer ofertas *grupales*.

- La consideración del aprendizaje por refuerzo en los aspectos del problema donde no se tiene conocimiento para hacer decisiones. Uno de los fenómenos en cualquier clase de subasta, es que las asignaciones resultantes son diferentes si los artículos a subastar, aunque sean los mismos, llegan en diferente orden. Este fenómeno en el problema de asignación de tareas es común en los ambientes dinámicos. También se da el caso cuando las consecuencias de una decisión no siempre se presentan de forma inmediata. En ambos acontecimientos, una técnica de aprendizaje es sumamente útil.

- La clasificación de los recursos y tareas en tipos. Esto permite acortar el tiempo de respuesta entre un cambio en el ambiente y el efecto que tiene el aprendizaje sobre las decisiones que toma un agente líder, ya que el ambiente que percibe, producto de la clasificación, es más simple.

Hemos incluido también una medición aproximada de la convergencia de la técnica de aprendizaje por refuerzo en la figura 5.11, esto es, el tiempo (en rondas) que tarda un agente líder en aprender una *política óptima*. Sin embargo, con el tiempo que se le dio a cada experimento (5000 rondas), es todavía posible que la medición presentada corresponda a una *política óptima local* y no global. Una *política óptima global* se garantiza cuando el tiempo que se le brinda a la técnica de aprendizaje es

Subasta / Tamaño	5	10	15	20
Paralelo-secuenciales	±839	±1648	±3267	∼
Secuenciales inversas	±1274	±2724	∼	∼

Tabla 5.3: Desviación estándar de la convergencia en el aprendizaje (complemento de los datos de la figura 5.11).

infinito [Watkins and Dayan, 1992]. La tabla 5.3 muestra los datos de la gráfica de forma numérica. Se observa que, en el caso de los ambientes con 20 recursos (además del ambiente con 15 recursos de la subastas secuenciales-inversas), el aprendizaje no llegó a converger en alguna política. Esto se debe al incremento de complejidad en el ambiente.

5.5.4 Resultados de la segunda etapa de experimentación: Tres niveles de la jerarquía

En esta etapa experimental, el objetivo es verificar las consecuencias de incluir varios agentes sofisticados o líderes, cada uno con su grupo de agentes subordinados, es decir, cada agente sofisticado abre, coordina y cierra mercados en los que sólo los agentes subordinados asociados pueden participar. Esto se representa con un nivel adicional superior en la jerarquía, mostrada en la figura 4.1. Resultados mejores a los que se obtuvieron en la etapa experimental anterior ofrecen evidencia de que esta técnica retiene la propiedad de escalabilidad en ambientes más grandes.

Se suministran las figuras 5.12 y 5.13 para dar evidencia de que, comparando directamente los resultados de esta etapa experimental con los resultados de nuestra propia técnica de la sección anterior, la inclusión de agentes líderes en escenarios con el mismo tamaño mejora, no sólo la calidad de las asignaciones, también acelera el aprendizaje de los agentes líderes, debido a que cada uno de ellos tiene que explorar menos cantidad de estados. Estas gráficas hacen referencia a un número de líderes específico, que corresponden al número de grupos máximo posible en un escenario

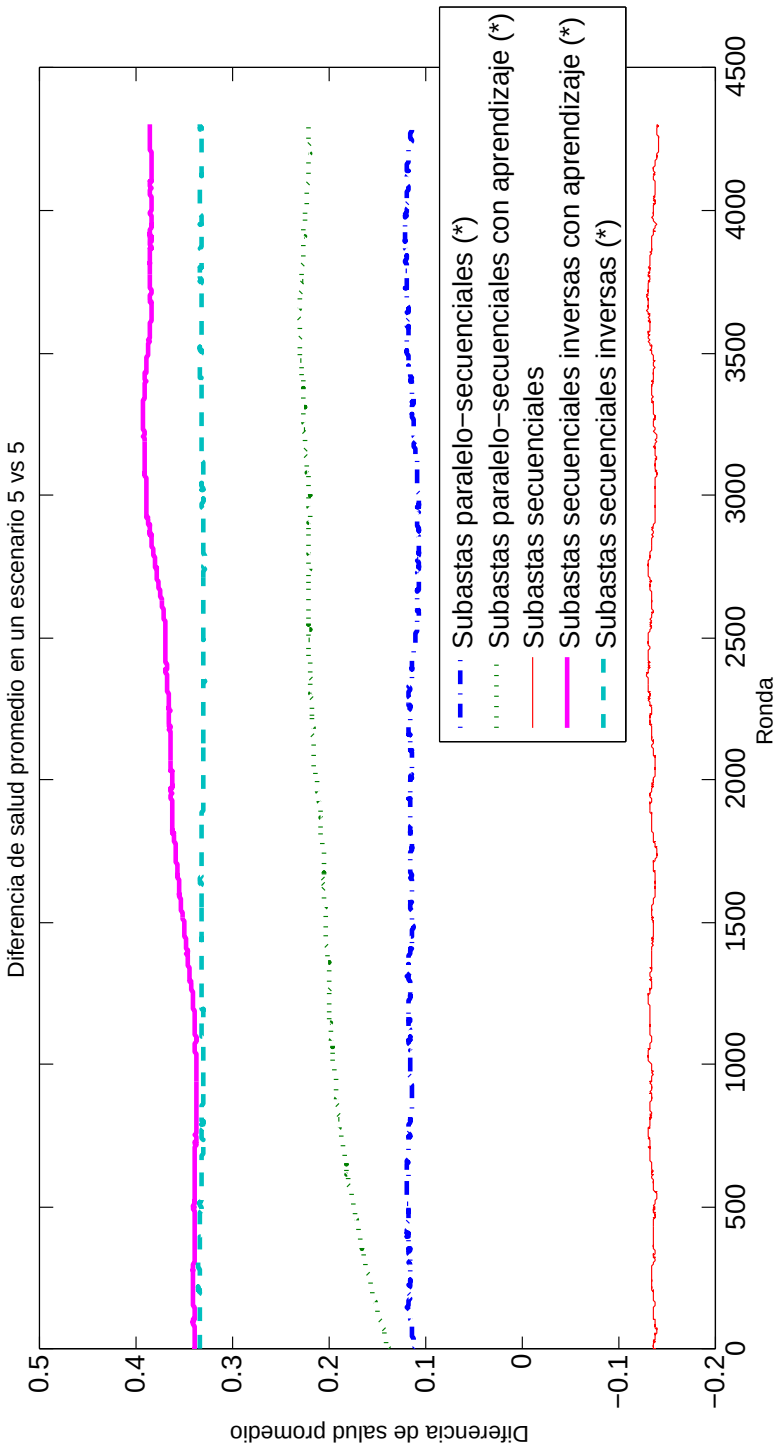


Figura 5.6: Comparación de técnicas en el ambiente con 5 recursos *vs* 5 tareas.

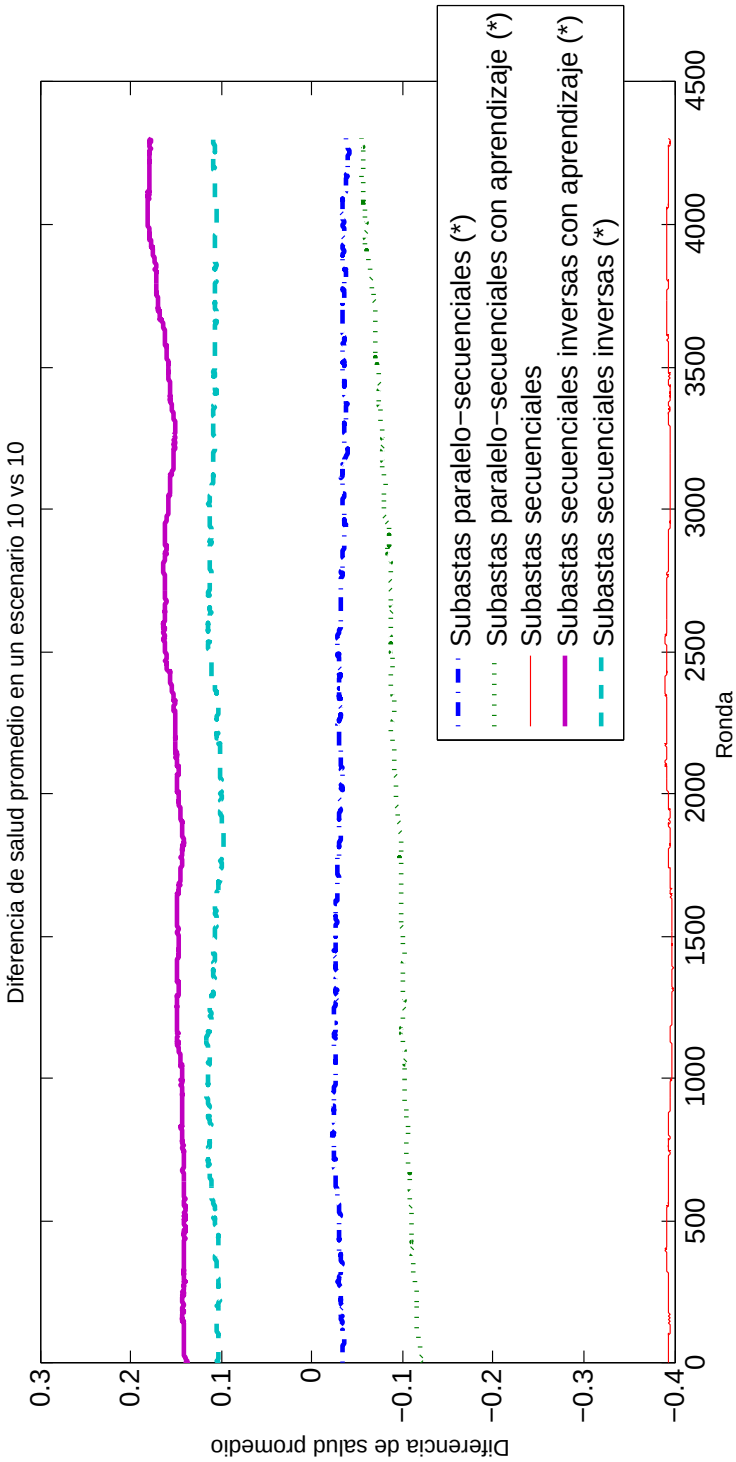


Figura 5.7: Comparación de técnicas en el ambiente con 10 recursos *vs* 10 tareas.

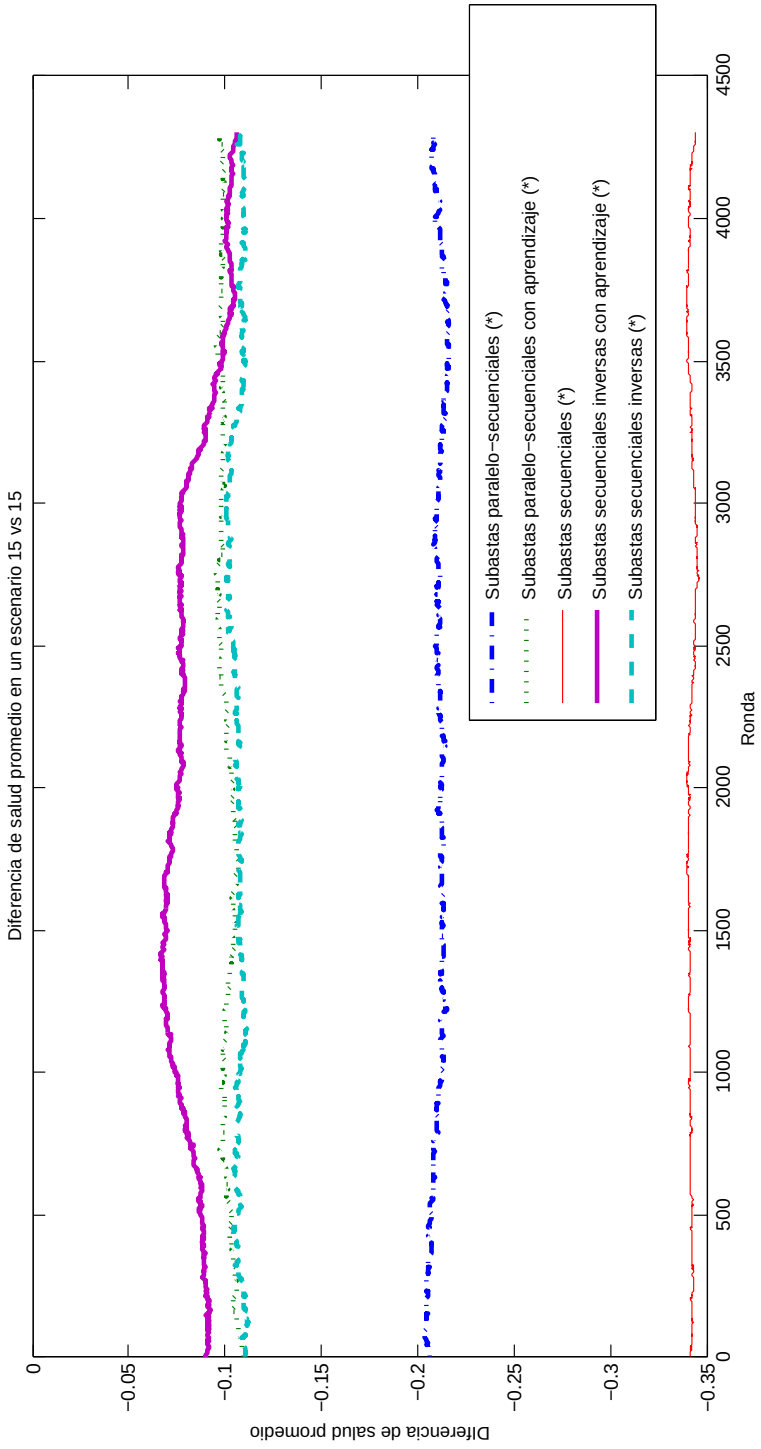


Figura 5.8: Comparación de técnicas en el ambiente con 15 recursos *vs* 15 tareas.

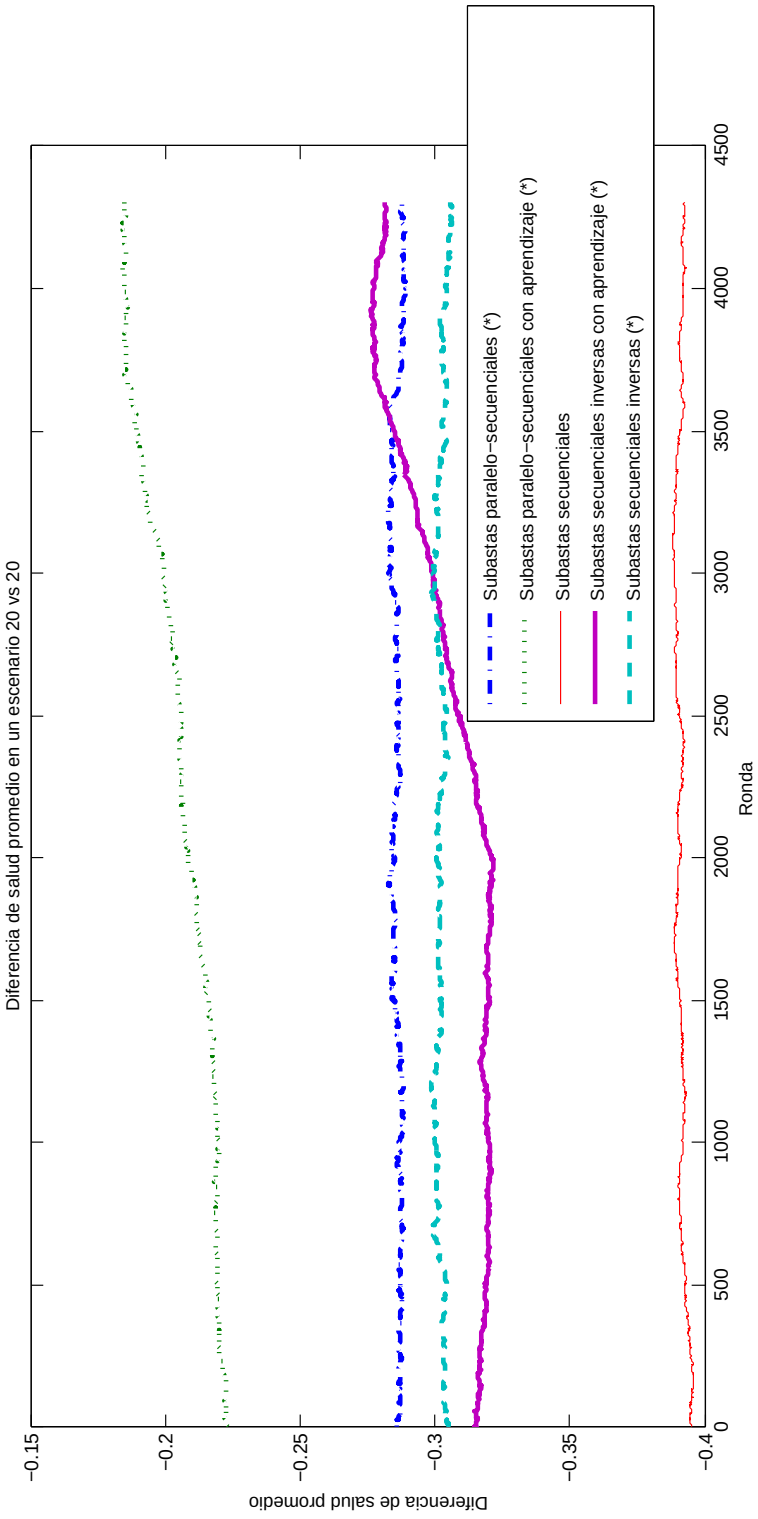


Figura 5.9: Comparación de técnicas en el ambiente con 20 recursos *vs* 20 tareas.

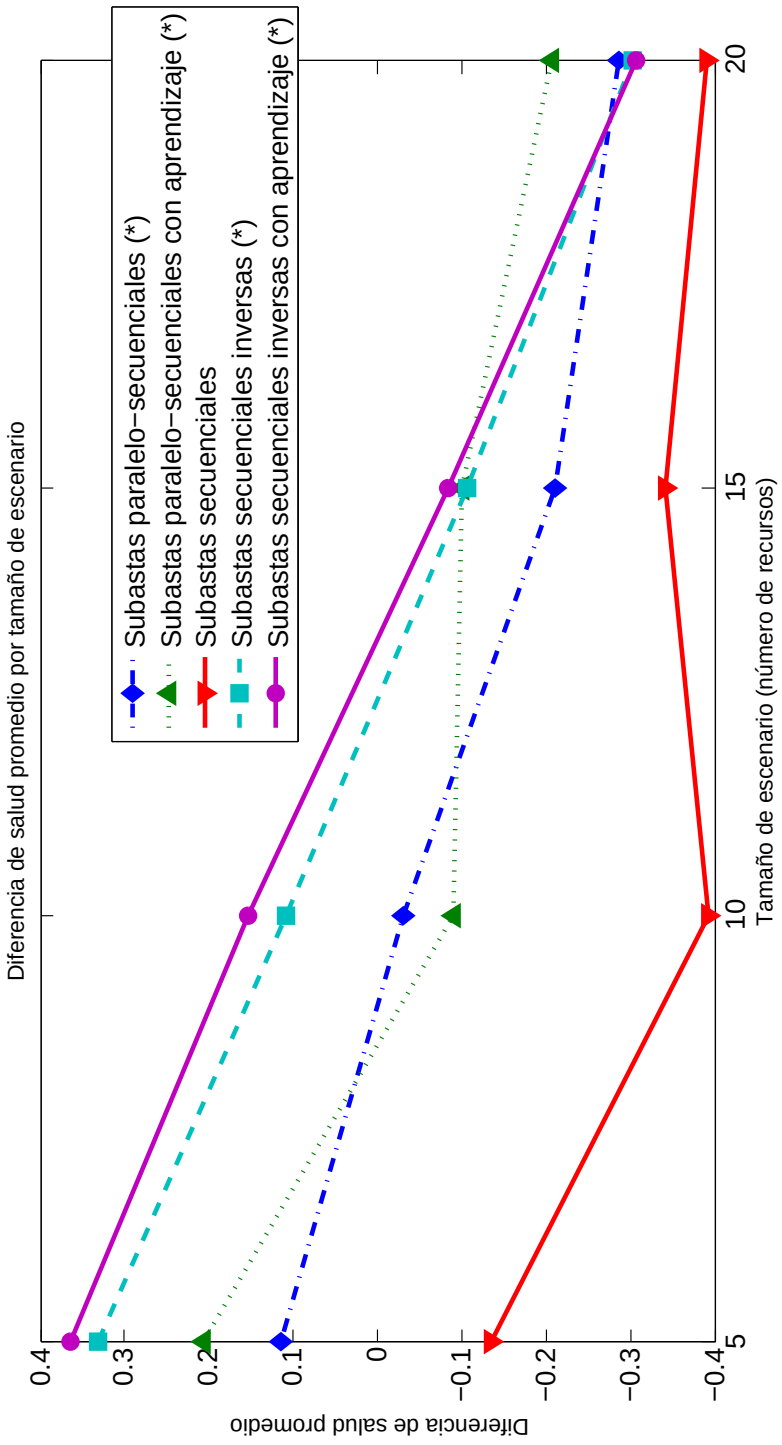


Figura 5.10: Comparación de técnicas en todos los ambientes.

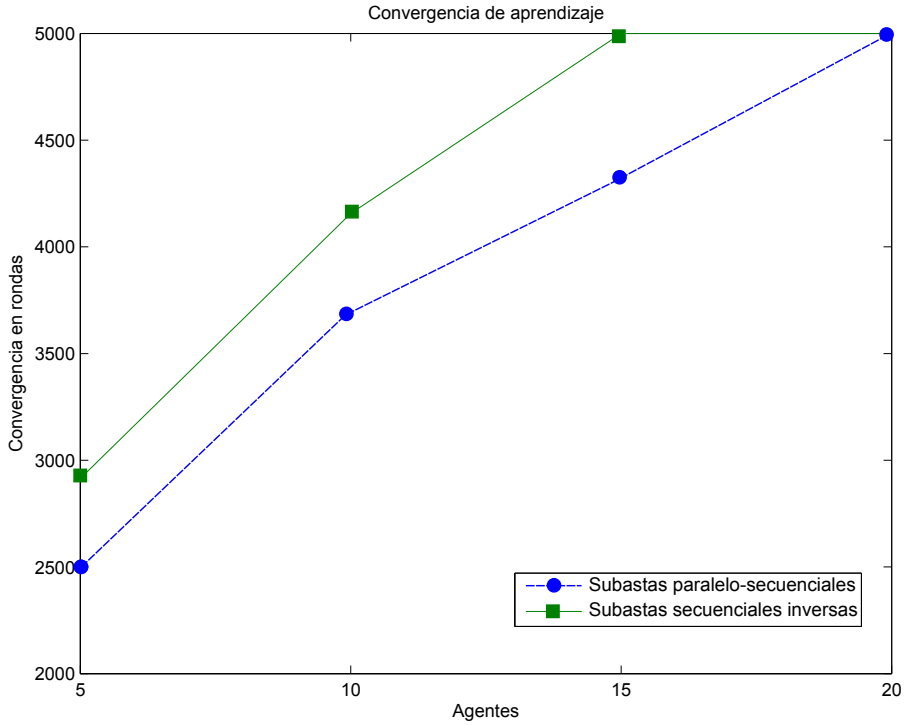


Figura 5.11: Convergencia del método de aprendizaje

de tamaño determinado. Por ejemplo, observando la tabla 5.2, en un ambiente de 30 de recursos con grupos de 5 agentes, se puede deducir que la cantidad de líderes en este escenario es 6.

Un fenómeno que se puede observar en ambas figuras es que la calidad de las asignaciones crece con el paso del tiempo, por lo que se deduce que la exploración realizada por los agentes líderes para aprender la dinámica del ambiente no ha terminado y es de esperar que se encuentren mejores asignaciones. Esto sucede, incluso, en la primera etapa experimental. Específicamente en los ambientes grandes (por ejemplo, en el ambiente con 15 recursos *vs* 15 tareas), donde se puede observar que la exploración no ha terminado e incluso, a causa de esta, la calidad de las asignaciones disminuye temporalmente. La figura 5.14 muestra los resultados de todos los escenarios de experimentación especificados en la tabla 5.2. En esta gráfica se puede observar la eficacia de tener varios grupos liderados por un agente sofisticado en un

ambiente cada vez más grande y que, al aumentar el tamaño de los grupos, esta eficacia se ve afectada. En general, aunque los resultados también decrecen mientras el tamaño del ambiente aumenta, como en el caso de la etapa experimental anterior, el impacto de la degradación es menos rotundo que en la primera etapa experimental. Estos resultados nos dan evidencia de que la técnica propuesta retiene su propiedad de escalabilidad para ambientes más grandes.

En esta etapa experimental, las mejoras que se han tenido sobre los resultados de la etapa anterior se debe a lo siguiente:

- Mayor número de líderes. Dado los tamaños especificados del ambiente (tabla 5.2), más líderes significa particiones más pequeñas y esto, a su vez, significa un ambiente menos complejo del cual aprender. Sin embargo, particiones muy pequeñas se traducirían a ambientes tan simples que cualquier conocimiento sobre estos sería irrelevante. Por lo que particiones menores a 5 no son recomendables.

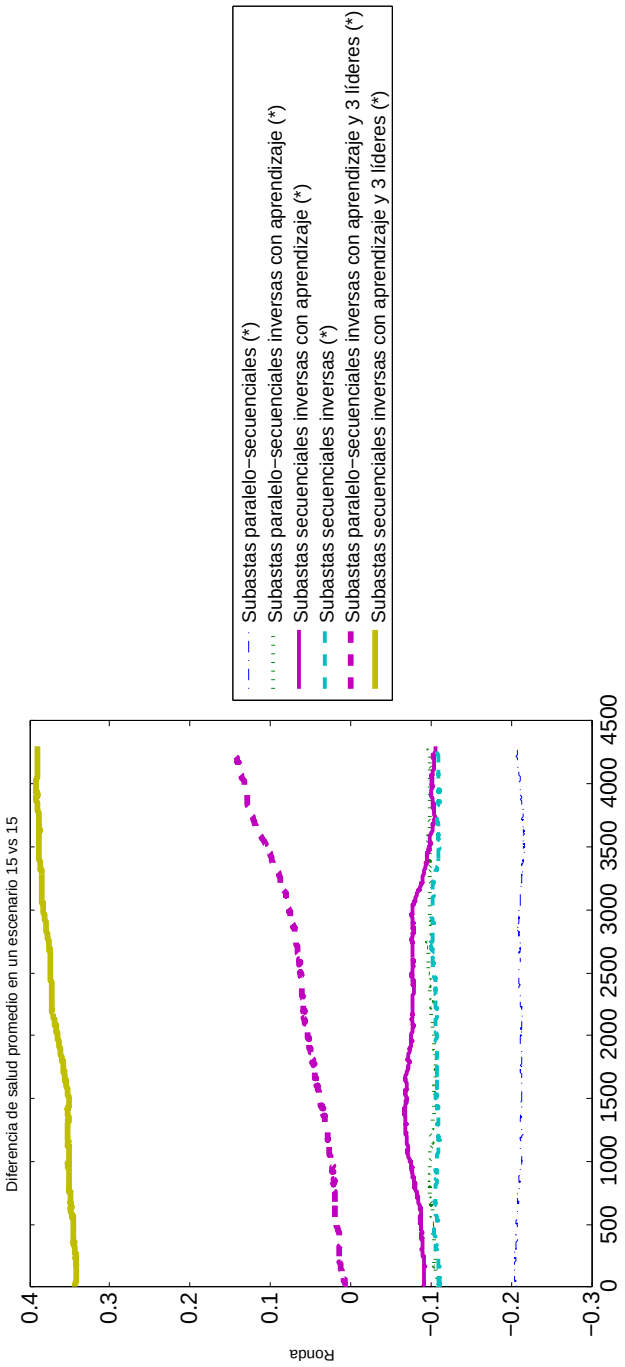


Figura 5.12: Comparación de técnicas en el ambiente con 15 recursos *vs* 15 tareas.

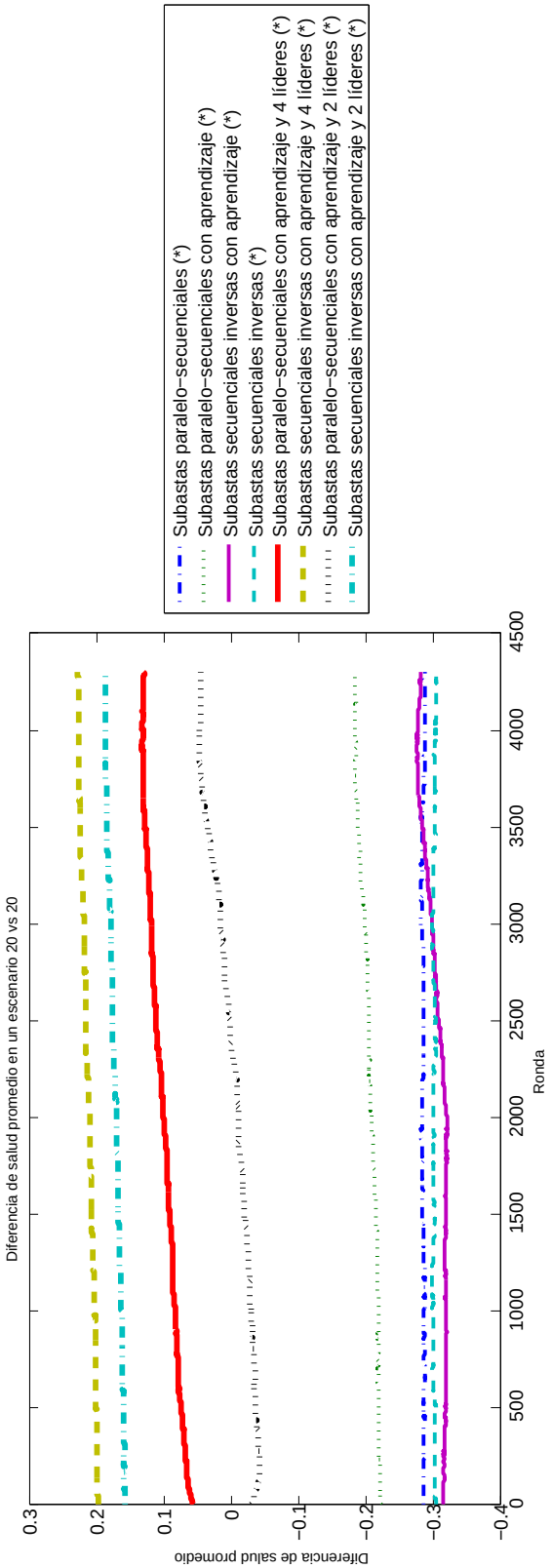


Figura 5.13: Comparación de técnicas en el ambiente con 20 recursos *vs* 20 tareas.

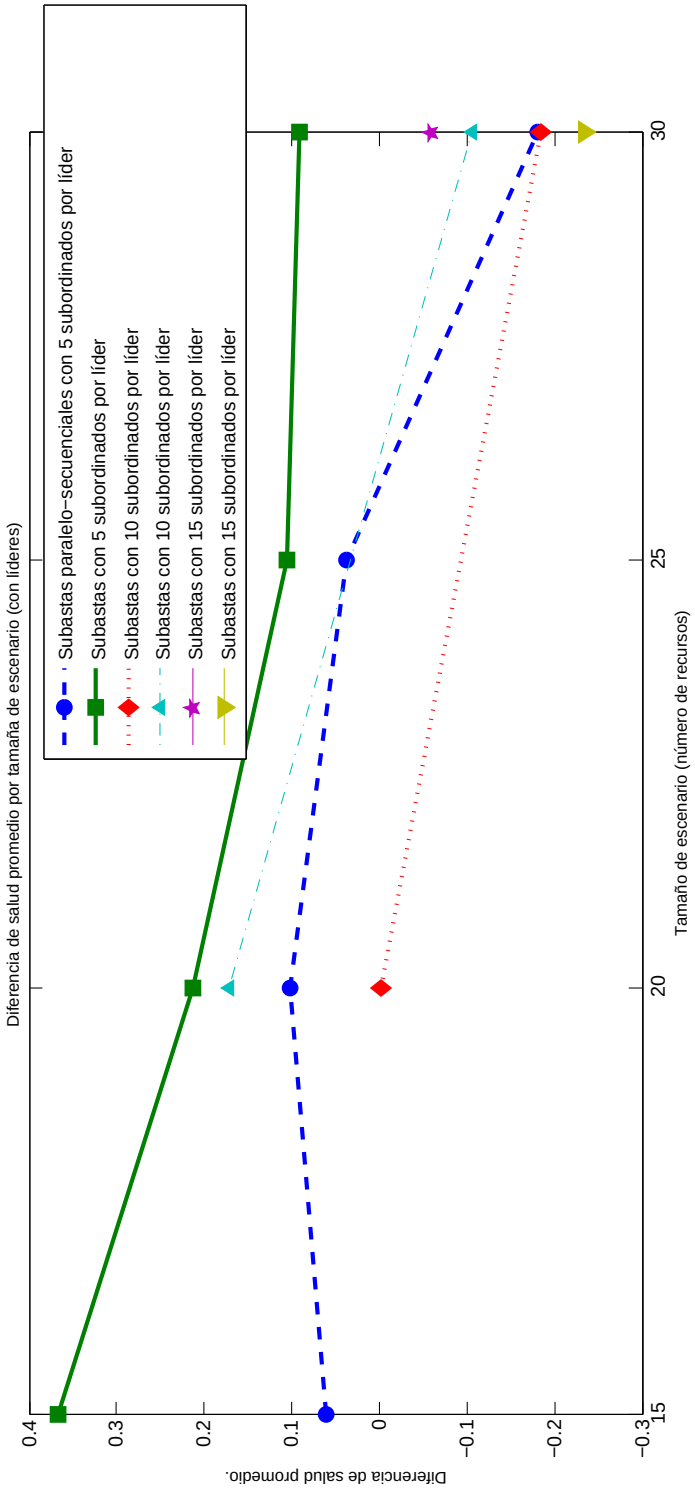


Figura 5.14: Resultados en todos los ambientes.

Capítulo 6

Conclusiones y trabajo futuro

En este capítulo haremos una breve recapitulación de lo realizado en este trabajo, además reuniremos la información que hemos obtenido después de realizar el proceso de experimentación del capítulo anterior y resumiremos el conocimiento que hemos obtenido de dicho ejercicio. También enunciaremos las ventajas y desventajas de este trabajo y para finalizar, sugerimos algunos posibles caminos para continuar esta investigación.

6.1 Resumen

La investigación que se llevó a cabo en este trabajo, como todo proceso metodológico, pasó por varias etapas.

- La observación de fenómenos en la asignación de tareas con subastas como la degradación de la calidad de la asignación en relación con el tamaño del ambiente, el mecanismo de la subasta y la formulación de la oferta.
- La formulación de las preguntas que guiaron la investigación: *"¿qué pasa si se modifica el mecanismo de la subasta?"*, *"¿qué sucede si el ambiente cambia?"*, *"En un ambiente cambiante, ¿puedo tomar mejores decisiones si conozco su dinámica?"*.

- La formulación de las hipótesis iniciales: La formulación de un mecanismo nuevo de subastas (*subastas paralelo-secuenciales*) mejora la calidad de asignación en todos los casos.
- Las primeras etapas de experimentación y su correspondiente observación y análisis.
- La reformulación de las hipótesis para considerar ambientes grandes.
- El planteamiento de la jerarquía propuesta: agentes líderes que exploran el ambiente para aprender a abrir subastas, donde participan agentes subordinados.
- La formulación de un cambio de representación del ambiente, es decir, la clasificación en tipos, para agilizar el proceso de aprendizaje.
- La experimentación del nuevo enfoque, su correspondiente observación y análisis.
- La formulación de otro mecanismo de subastas (*subastas secuenciales inversas*) para considerar más que asignaciones puramente lineales.
- La experimentación final, con su correspondiente observación y análisis.

6.2 CONCLUSIONES

En este trabajo, gracias a la experimentación realizada y a la evidencia que nos suministró esta actividad, se llegaron a las siguientes conclusiones:

- La adaptación de las subastas propuestas a un ambiente dinámico mejoran la calidad de las asignaciones comparado con las subastas secuenciales, desde ambiente pequeños hasta ambientes medianos (de hasta 20 agentes). Este enfoque retiene escalabilidad pero la calidad de las asignaciones se degrada mientras el tamaño del ambiente crece.

- La inclusión de un agente sofisticado que supervise la apertura de mercados, seleccione el tipo de tarea que sea más conveniente en un instante de tiempo, es beneficioso para obtener mejores resultados a mediano/largo plazo sobre la asignación de tareas resuelta sólo con subastas, en ambiente pequeños hasta ambientes medianos (de hasta 20 agentes). Este enfoque combinado también retiene escalabilidad pero la calidad de las asignaciones se degrada mientras el tamaño del ambiente crece y el beneficio de aplicar aprendizaje desaparece, debido a que los agentes requieren cada vez más tiempo de exploración en el aprendizaje del dominio.

- La división de un conjunto grande de recursos en subconjuntos más pequeños supervisados por su propio agente sofisticado mejora los resultados a comparación de un enfoque sin sub-divisiones de recursos. Es decir, la aplicación del modelo jerárquico propuesto ofrece beneficios en ambientes medianos a ambientes grandes. La calidad de las asignaciones no se degrada mientras se elija un número adecuado de agentes líderes.

- La aplicación del modelo jerarquizado disminuye la tasa con la que la calidad de las soluciones obtenidas disminuye en relación con el aumento del tamaño del ambiente, es decir, al aplicar un modelo jerarquizado, el aumento en el tamaño del ambiente degrada la calidad de las soluciones de manera más lenta que sin la jerarquía.

- Las subastas secuenciales inversas, al aprovechar las asignaciones de muchos recursos a una tarea, exploran de manera más efectiva el espacio de asignaciones. Esto resulta en asignaciones con mayor calidad que las subastas paralelo-secuenciales. Sin embargo, estas - en su variación con aprendizaje - requieren que sus agentes líderes exploren más el ambiente para descubrir buenas asignaciones.

6.3 LIMITACIONES

- Aunque no se requiere conocimiento *a priori* del ambiente, una buena noción de él mejora el rendimiento y la eficacia de la técnica. Esto se hace muy presente en la categorización por tipos de los recursos y las tareas, y en menor medida en la formulación de las ofertas y el cálculo de la recompensa.
- El aumento del tamaño del ambiente hace que el tiempo de aprendizaje aumente, debido a que el método de aprendizaje debe explorar más parejas estado-acción.
- La efectividad del aprendizaje individual de cada líder es inversamente proporcional al grado de dependencia entre las regiones del ambiente que cada agente líder observa. Es decir, si los acontecimientos en una región afectan débilmente a otra, entonces el conocimiento que haya aprendido cada agente líder será bueno y conducirá a buenas decisiones y viceversa.

6.4 TRABAJO FUTURO

El trabajo propuesto puede ser mejorado en varias partes. A grandes rasgos, en:

- El algoritmo de aprendizaje por refuerzo.
- El esquema de subastas aplicado.
- El cálculo de las ofertas.
- La representación del estado y las acciones.
- El cálculo de la recompensa.

De manera específica, pueden utilizarse algoritmos de aprendizaje por refuerzo que consideren el conocimiento adquirido por otros agentes en los niveles superiores de la jerarquía o bien, algoritmos que aprendan más rápido. El esquema de subastas también puede ser modificado, es decir, la forma en la que se abren los mercados, la forma en la que los agentes se presentan como postores, la forma en la que estos presentan sus ofertas, la forma en la que se elige a un ganador, etc. Hay que distinguir la forma en la que se calculan las ofertas de la forma en la que estas se presentan y se consideran para la selección de un ganador. El cálculo de la oferta también puede ser modificado, es decir, puede considerar más o menos parámetros, cada uno ponderado de distintas formas y todas las posibles variaciones de ofertas pueden ser usadas en el mismo esquema de subastas.

En la técnica propuesta, se utiliza una función de clasificación específica para la representación del estado y las acciones, sin embargo, pueden utilizarse otras funciones para conseguir otras representaciones que sean más fieles al estado real del ambiente. El cambio en la representación del estado impacta la velocidad de convergencia en el proceso de aprendizaje, así como la efectividad de la técnica en su totalidad. Recordando que la recompensa, en el aprendizaje por refuerzo, es la forma de decirle al agente qué ha hecho bien o qué ha hecho mal, la posibilidad está abierta para darle a cada agente una mejor indicación de la efectividad de sus decisiones que la que se ha propuesto.

No sólo es posible cambiar el esquema de subasta aplicada, también está abierta la posibilidad a cambiar el enfoque completo para cada agente subordinado. Por ejemplo, es posible utilizar técnicas distribuidas de optimización, en donde cada agente aporte al cálculo correspondiente para encontrar la mejor asignación a la subasta que seleccionó el agente sofisticado. También es posible aplicar otros paradigmas basados en mercados, como las *negociaciones*.

Bibliografía

[Amador et al., 2014] Amador, S., Okamoto, S., and Zivan, R. (2014). Dynamic multi-agent task allocation with spatial and temporal constraints. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*, AAMAS '14, pages 1495–1496, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

[Binetti et al., 2013] Binetti, G., Naso, D., and Turchiano, B. (2013). Decentralized task allocation for surveillance systems with critical tasks. *Robotics and Autonomous Systems*, 61(12):1653 – 1664.

[Blizzard®, 1998] Blizzard® (1998). Starcraft. `http://us.blizzard.com/es-mx/games/sc/`.

[Brutschy et al., 2014] Brutschy, A., Pini, G., Pinciroli, C., Birattari, M., and Dorigo, M. (2014). Self-organized task allocation to sequentially interdependent tasks in swarm robotics. *Autonomous Agents and Multi-Agent Systems*, 28(1):101–125.

[Celaya and Arronategui, 2013] Celaya, J. and Arronategui, U. (2013). A task routing approach to large-scale scheduling. *Future Generation Computer Systems*, 29(5):1097 – 1111.

[Chapman et al., 2010] Chapman, A. C., Micillo, R. A., Kota, R., and Jennings, N. R. (2010). Decentralized dynamic task allocation using overlapping potential games. *Comput. J.*, 53(9):1462–1477.

[Chekuri and Khanna, 2000] Chekuri, C. and Khanna, S. (2000). A ptas for the multiple knapsack problem. In *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '00, pages 213–222, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.

[Cheng et al., 2013] Cheng, K., Zhang, H., and Zhang, R. (2013). A task-resource allocation method based on effectiveness. *Knowledge-Based Systems*, 37(0):196 – 202.

[Choi et al., 2009] Choi, H.-L., Brunet, L., and How, J. (2009). Consensus-based decentralized auctions for robust task allocation. *Robotics, IEEE Transactions on*, 25(4):912–926.

[Claus and Boutilier, 1998] Claus, C. and Boutilier, C. (1998). The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, AAAI '98/IAAI '98, pages 746–752, Menlo Park, CA, USA. American Association for Artificial Intelligence.

[Cohen et al., 2006] Cohen, R., Katzir, L., and Raz, D. (2006). An efficient approximation for the generalized assignment problem. *Information Processing Letters*, 100(4):162 – 166.

[Deb and Kalyanmoy, 2001] Deb, K. and Kalyanmoy, D. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc., New York, NY, USA.

[Dias et al., 2006] Dias, M. B., Zlot, R., Kalra, N., and Stentz, A. (2006). Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE*, 94(7):1257–1270.

[Fogue et al., 2013] Fogue, M., Garrido, P., Martinez, F. J., Cano, J.-C., Calafate, C. T., and Manzoni, P. (2013). A novel approach for traffic accidents sanitary

resource allocation based on multi-objective genetic algorithms. *Expert Systems with Applications*, 40(1):323 – 336.

[Franklin and Graesser, 1997] Franklin, S. and Graesser, A. (1997). Is it an agent, or just a program?: a taxonomy for autonomous agents. *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*, 3(2):21–35.

[Garey and Johnson, 1990] Garey, M. R. and Johnson, D. S. (1990). *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA.

[Gerkey, 2003] Gerkey, B. P. (2003). *On Multi-robot Task Allocation*. PhD thesis, University of Southern California, Los Angeles, CA, USA.

[Grimaldi, 2003] Grimaldi, R. (2003). *Discrete and Combinatorial Mathematics: An Applied Introduction*. Pearson.

[Grossberg, 1988] Grossberg, S. (1988). Nonlinear neural networks: Principles, mechanisms, and architectures. *Neural Networks*, 1(1):17 – 61.

[Heap, 2013] Heap, B. (2013). Sequential single-cluster auctions. In Klusch, M., Thimm, M., and Paprzycki, M., editors, *Multiagent System Technologies*, volume 8076 of *Lecture Notes in Computer Science*, pages 408–411. Springer Berlin Heidelberg.

[Heap and Pagnucco, 2012a] Heap, B. and Pagnucco, M. (2012a). Analysis of cluster formation techniques for multi-robot task allocation using sequential single-cluster auctions. In Thielscher, M. and Zhang, D., editors, *AI 2012: Advances in Artificial Intelligence*, volume 7691 of *Lecture Notes in Computer Science*, pages 839–850. Springer Berlin Heidelberg.

[Heap and Pagnucco, 2012b] Heap, B. and Pagnucco, M. (2012b). Repeated sequential auctions with dynamic task clusters. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2012*, pages 1997–2002. AAAI Press.

[Heinerm, 2011] Heinerm, A. (2011). Brood war application programming interface. `https://code.google.com/p/bwapi/`.

[Huang et al., 2013] Huang, C.-J., Guan, C.-T., Chen, H.-M., Wang, Y.-W., Chang, S.-C., Li, C.-Y., and Weng, C.-H. (2013). An adaptive resource management scheme in cloud computing. *Engineering Applications of Artificial Intelligence*, 26(1):382 – 389.

[Kaelbling et al., 1996] Kaelbling, L. P., Littman, M. L., and Moore, A. W. (1996). Reinforcement learning: A survey. *J. Artif. Int. Res.*, 4(1):237–285.

[Kang et al., 2013] Kang, Q., He, H., and Wei, J. (2013). An effective iterated greedy algorithm for reliability-oriented task allocation in distributed computing systems. *Journal of Parallel and Distributed Computing*, 73(8):1106 – 1115.

[Koenig and Simmons, 1993] Koenig, S. and Simmons, R. G. (1993). Complexity analysis of real-time reinforcement learning. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, AAAI'93, pages 99–105. AAAI Press.

[Korsah et al., 2013] Korsah, G. A., Stentz, A., and Dias, M. B. (2013). A comprehensive taxonomy for multi-robot task allocation. *Int. J. Rob. Res.*, 32(12):1495–1512.

[Kosko, 1992] Kosko, B. (1992). *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[Leme et al., 2012] Leme, R. P., Syrgkanis, V., and Éva Tardos (2012). Sequential auctions and externalities. In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 869–886. SIAM.

[Macarthur, 2011] Macarthur, K. (2011). *Multi-agent coordination for dynamic decentralised task allocation*. PhD thesis, University of Southampton, Hampshire, ENG, UK.

[Marler and Arora, 2004] Marler, R. and Arora, J. (2004). Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26(6):369–395.

[Muñoz de Cote, 2008] Muñoz de Cote, E. (2008). *Socially Driven Multiagent Learning*. PhD thesis, Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria, Plaza Leonardo da Vinci, 32, Milán, Italia.

[Murugesan, 1998] Murugesan, S. (1998). Intelligent agents on the internet and web. *TENCON 98. 1998 Region 10 International Conference on Global Connectivity in Energy, Computer, Communication and Control*, 1:97–102.

[Nanjanath and Gini, 2010] Nanjanath, M. and Gini, M. (2010). Repeated auctions for robust task execution by a robot team. *Robotics and Autonomous Systems*, 58(7):900 – 909.

[Papadimitriou and Yannakakis, 1991] Papadimitriou, C. H. and Yannakakis, M. (1991). Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43(3):425 – 440.

[Pelikan et al., 2002] Pelikan, M., Goldberg, D., and Lobo, F. (2002). A survey of optimization by building and using probabilistic models. *Computational Optimization and Applications*, 21(1):5–20.

[Pippin and Christensen, 2011] Pippin, C. E. and Christensen, H. (2011). A Bayesian formulation for auction-based task allocation in heterogeneous multi-agent teams. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 8047 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*.

[Pippin and Christensen, 2013] Pippin, C. E. and Christensen, H. (2013). Learning task performance in market-based task allocation. In Lee, S., Cho, H., Yoon, K.-J., and Lee, J., editors, *Intelligent Autonomous Systems 12*, volume 194 of *Advances in Intelligent Systems and Computing*, pages 613–621. Springer Berlin Heidelberg.

[Russell and Norvig, 2009] Russell, S. and Norvig, P. (2009). *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd edition.

[Said, 2011] Said, M. (2011). Sequential auctions with randomly arriving buyers. *Games and Economic Behavior*, 73(1):236 – 243.

[Schoenig and Pagnucco, 2011] Schoenig, A. and Pagnucco, M. (2011). Evaluating sequential single-item auctions for dynamic task allocation. In Li, J., editor, *AI 2010: Advances in Artificial Intelligence*, volume 6464 of *Lecture Notes in Computer Science*, pages 506–515. Springer Berlin Heidelberg.

[Shmoys and Tardos, 1993] Shmoys, D. and Tardos, E. (1993). An approximation algorithm for the generalized assignment problem. *Mathematical Programming*, 62(1-3):461–474.

[Shoham and Leyton-Brown, 2008] Shoham, Y. and Leyton-Brown, K. (2008). *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, New York, NY, USA. `http://www.masfoundation.org`.

[Strehl, 2007] Strehl, A. L. (2007). *PAC Exploration in reinforcement learning*. PhD thesis, Rutgers University.

[Thomas and Williams, 2009] Thomas, G. and Williams, A. (2009). Sequential auctions for heterogeneous task allocation in multiagent routing domains. In *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, pages 1995–2000.

[Thrun, 1992] Thrun, S. B. (1992). Efficient exploration in reinforcement learning. Technical report, Carnegie Mellon University, Pittsburgh, PA, USA.

[Tolmidis and Petrou, 2013] Tolmidis, A. and Petrou, L. (2013). Multi-objective optimization for dynamic task allocation in a multi-robot system. *Engineering Applications of Artificial Intelligence*, 26(5 - 6):1458 – 1468.

[Tsai et al., 2013] Tsai, J.-T., Fang, J.-C., and Chou, J.-H. (2013). Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm. *Computers & Operations Research*, 40(12):3045 – 3055.

[van der Hoek and Wooldridge, 2008] van der Hoek, W. and Wooldridge, M. (2008). Chapter 24 multi-agent systems. In van Harmelen, F., Lifschitz, V., and Porter, B., editors, *Handbook of Knowledge Representation*, volume 3 of *Foundations of Artificial Intelligence*, pages 887 – 928. Elsevier.

[van der Horst and Noble, 2010] van der Horst, J. and Noble, J. (2010). Distributed and centralized task allocation: When and where to use them. In *Self-Adaptive and Self-Organizing Systems Workshop (SASOW), 2010 Fourth IEEE International Conference on*, pages 1–8.

[Vidal, 2010] Vidal, J. M. (2010). Fundamentals of multiagent systems. `http://www.multiagent.com` Fecha de última consulta: Diciembre, 2014.

[Vlassis, 2007] Vlassis, N. (2007). *A Concise Introduction to Multiagent Systems and Distributed Artificial Intelligence*. Morgan and Claypool Publishers, 1st edition.

[Watkins and Dayan, 1992] Watkins, C. and Dayan, P. (1992). Q-learning. *Machine Learning*, 8(3-4):279–292.

[Weiss, 2000] Weiss, G. (2000). *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press, Cambridge, MA, USA, 1st edition.

[Wooldridge, 2009] Wooldridge, M. (2009). *An Introduction to MultiAgent Systems*. Wiley Publishing, 2nd edition.

[Wooldridge and Jennings, 1995] Wooldridge, M. and Jennings, N. R. (1995). Intelligent agents: theory and practice. *The Knowledge Engineering Review*, 10:115–152.

[Yedidsion et al., 2011] Yedidsion, L., Shabtay, D., and Kaspi, M. (2011). Complexity analysis of an assignment problem with controllable assignment costs and its applications in scheduling. *Discrete Applied Mathematics*, 159(12):1264 – 1278.

[Zhang et al., 2010] Zhang, K., Collins, E., and Barbu, A. (2010). A novel stochastic clustering auction for task allocation in multi-robot teams. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3300–3307.

[Zhang et al., 2009] Zhang, K., Collins, E. G., Shi, D., Liu, X., and Chuy, O. (2009). A stochastic clustering auction (sca) for centralized and distributed task allocation in multi-agent teams. In Asama, H., Kurokawa, H., Ota, J., and Sekiyama, K., editors, *Distributed Autonomous Robotic Systems 8*, volume 8, pages 345–354. Springer Berlin Heidelberg.

[Zheng and Koenig, 2010] Zheng, X. and Koenig, S. (2010). Sequential incremental-value auctions. In Fox, M. and Poole, D., editors, *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*, pages 941 – 946. AAAI Press.

[Zheng et al., 2006] Zheng, X., Koenig, S., and Tovey, C. (2006). Improving sequential single-item auctions. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 2238–2244.

[Zimmermann, 1978] Zimmermann, H.-J. (1978). Fuzzy programming and linear programming with several objective functions. *Fuzzy Sets and Systems*, 1(1):45 – 55.