

APRENDIZAJE COOPERATIVO MULTIAGENTE

Márquez Gutiérrez, Pedro Rafael; Ferreiro Salvatierra, Brenda Sofía;
Mingura Erivez, Diego; García Mata, Carmen Leticia
Tecnológico Nacional de México/ Tecnológico de Chihuahua
División de Estudios de Posgrado e Investigación
Avenida Tecnológico 2909, Chihuahua, Chih. 31310
Tel., 614-201-2000
{pedro.mg, M20061538, M20061535, carmen.gm@chihuahua.tecnm.mx}

RESUMEN.

Los sistemas cooperativos multiagente son aquellos en los que varios agentes intentan, a través de su interacción, resolver tareas conjuntamente o maximizar una utilidad. Debido a las interacciones entre los agentes, la complejidad del problema multiagente aumenta rápidamente con el número de agentes o su sofisticación conductual. El desafío que esto presenta a la tarea de programar soluciones a problemas de sistemas multiagente ha generado un creciente interés en las técnicas de aprendizaje automático para automatizar el proceso de búsqueda y optimización.

En este artículo abordamos el trabajo de aprendizaje multiagente en un espectro de áreas, incluyendo el aprendizaje por refuerzo, la computación evolutiva, la teoría de juegos, el modelado de agentes y la robótica.

Palabras clave: sistemas multiagente, aprendizaje cooperativo, inteligencia artificial, aprendizaje de máquina.

ABSTRACT.

Cooperative multi-agent systems are ones in which several agents attempt, through their interaction, to jointly solve tasks or to maximize utility. Due to the interactions among the agents, multi-agent problem complexity rises rapidly with the number of agents or their behavioral sophistication. The challenge this presents to the task of programming solutions to multi-agent systems problems has spawned increasing interest in machine learning techniques to automate the search and optimization process.

In this paper we attempt to draw from multi-agent learning work in a spectrum of areas, including reinforcement learning, evolutionary computation, game theory, agent modeling, and robotics.

Keywords: multi-agent systems, cooperative learning, artificial intelligence, machine learning.

1. INTRODUCCIÓN

A la combinación de Sistemas Distribuidos [2,3,4,5] e Inteligencia Artificial (IA) [6] se le conoce colectivamente como Inteligencia Artificial Distribuida (DAI, por sus siglas en inglés). La DAI se puede dividir en dos grandes áreas. La primera aborda la resolución distribuida de problemas, entendida como la descomposición y distribución de un proceso de resolución de problemas entre múltiples nodos esclavos de procesamiento conectados en red, así como la reconstrucción colectiva de la solución al problema; véase la Figura 1.

La segunda parte, los sistemas multiagente (MAS), enfatiza el comportamiento conjunto de agentes inteligentes con cierto

grado de autonomía y la complejidad que surge de sus interacciones.

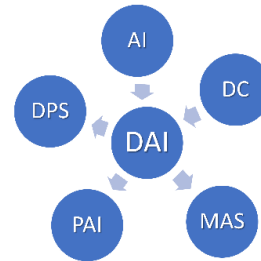


Figura 1. Inteligencia Artificial (AI) Distribuida (DAI), Resolución Distribuida de Problemas (DPS), Computación Distribuida (DC), IA en Paralelo (PAI) y Sistemas Multiagente (MAS)

En este artículo se revisa brevemente la problemática del aprendizaje automático (*machine learning*) en el contexto de problemas de MAS.

En general, *Machine Learning* [8] explora formas de automatización del proceso inductivo, i.e., como conseguir que un agente descubra por sí solo, a menudo a través de ensayos repetidos, cómo resolver una tarea determinada o minimizar el error en la solución. La complejidad de este problema aumenta grandemente si consideramos un entorno con múltiples agentes que aprenden a resolver un mismo problema de manera conjunta.

2. SISTEMAS MULTIAGENTE INTELIGENTES

Aunque no existe una definición universalmente aceptada acerca del concepto de inteligencia, podemos entender por agente inteligente, un ente computacional que exhibe un alto grado de autonomía, que analiza su entorno y realiza acciones sobre él con base en la información (sensores, retroalimentación) percibida [6]. La figura 2 muestra un esquema general acerca de la idea de agente inteligente. Como se puede observar, el concepto de agente inteligente es suficientemente general y flexible, de tal modo que prácticamente cualquier sistema se puede modelar mediante agentes, inteligentes o no.

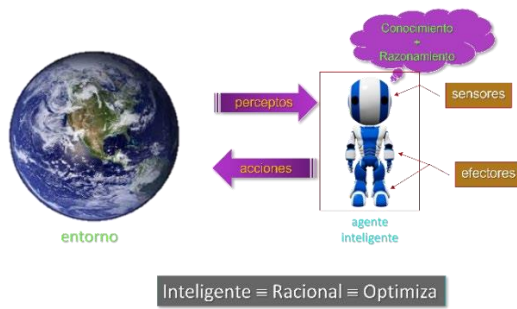


Figura 2. Agente Inteligente y su Entorno

Un diagrama más detallado de la estructura interna de un agente inteligente tomado como base, es el siguiente, Fig. 3.

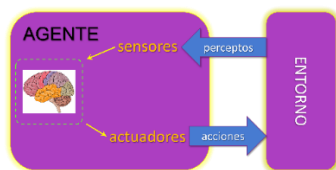


Figura 3. Arquitectura de un Agente Inteligente

Un entorno multiagente es aquel en el que coexisten más de un agente, que normalmente interactúan entre sí. El entorno posee restricciones tales que los agentes pueden no saber en todo momento lo que otros agentes perciben y conocen, incluidos los estados internos de los propios agentes. Además, el mundo real (entorno) en el cual se encuentran ubicados los agentes es de tal complejidad natural, que la construcción de agentes con inteligencia universal es una tarea formidable, Fig. 4.

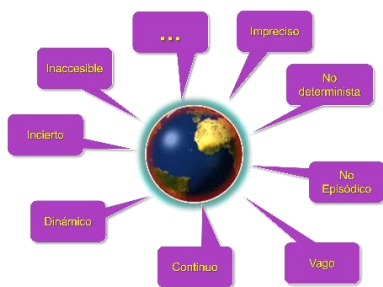


Figura 4. El Mundo es Absurdamente Complejo

Como Jennings et al. [7] sugieren, pocos dominios de problemas del mundo real permiten reducciones a la equivalencia de un solo agente: la mayoría de los agentes tienen características de información incompleta sobre el entorno, falta de control centralizado, información distribuida, y computación asincrónica [2,3,4,5]. La figura 5 muestra un esquema de configuración modular de un sistema multiagente.

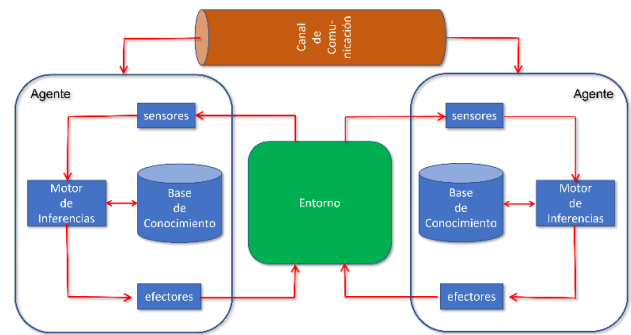


Figura 5. Sistema Multiagente

Algunas plataformas para el desarrollo de MAS son: JACK, JADE, JAFMAS y MADKit [28].

2.1. Aprendizaje de Máquina.

Comúnmente se considera que existen tres enfoques principales en el aprendizaje de máquina: aprendizaje supervisado, no supervisado y basado en recompensas [8]. Estos métodos se distinguen por el tipo de retroalimentación que el módulo de crítica proporciona al aprendiz. En el aprendizaje supervisado [9], el crítico proporciona la salida correcta, véase Fig. 6. Su objetivo es crear una función capaz de predecir el valor correspondiente a cualquier objeto de entrada válida después de haber visto una serie de ejemplos (en gran cantidad).

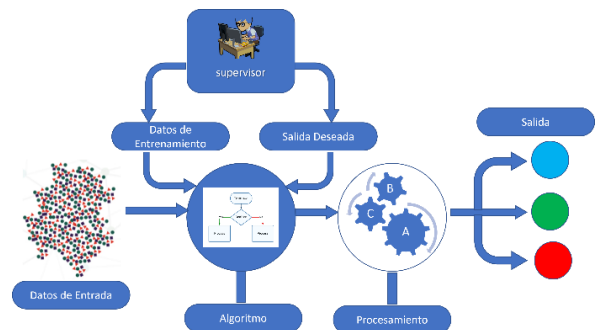


Figura 6. Aprendizaje Supervisado

En el aprendizaje no supervisado, no se proporciona retroalimentación en absoluto, Fig. 7. Se distingue del aprendizaje supervisado por el hecho de que no hay un conocimiento a priori. Típicamente trata los objetos de entrada como un conjunto de variables aleatorias, siendo construido un modelo de densidad para el conjunto de datos. El aprendizaje no supervisado se puede usar en conjunto con la inferencia Bayesiana para producir probabilidades condicionales para cualquiera de las variables aleatorias dadas.

En el aprendizaje basado en refuerzo, Fig. 7, el crítico proporciona una "recompensa" como respuesta a sus acciones [16]. Está inspirada en la psicología conductista, cuyo interés es determinar las acciones que debe escoger un agente en un

entorno predeterminado, con el fin de maximizar alguna noción de "recompensa" o premio acumulado.

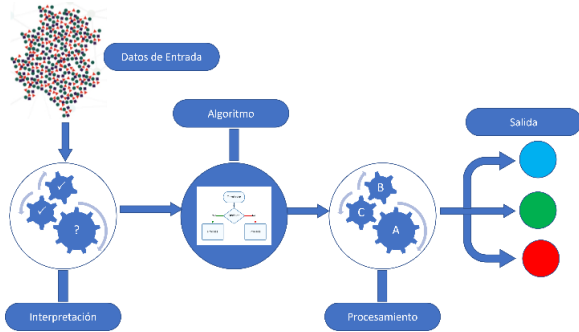


Figura 7. Aprendizaje No Supervisado

Debido a la complejidad inherente en las interacciones entre múltiples agentes, varios métodos de aprendizaje automático, en particular los métodos de aprendizaje supervisados, no se aplican fácilmente en el escenario multiagente. Esto debido a que generalmente asumen un solo crítico que pueda proporcionar a los agentes el comportamiento "correcto" para una situación dada, una excepción notable que implica la enseñanza en el contexto de los aprendices supervisados mutuamente [7]. Por lo tanto, la gran mayoría de las investigaciones en este campo han utilizado métodos basados en recompensas (por refuerzo).

2.2. Aprendizaje por Refuerzo

El Aprendizaje por Refuerzo (RL – *Reinforcement Learning*) es particularmente útil en dominios donde la información de refuerzo (expresada como penalizaciones o recompensas) se proporciona después de una secuencia de acciones realizadas sobre el entorno, como se muestra en la Fig. 8.

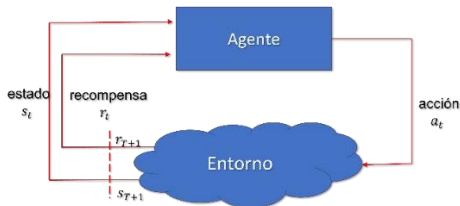


Figura 8. Aprendizaje por Refuerzo

Q-Learning y Temporal-Difference (TD(λ)) son dos métodos comunes de RL. El primero aprende la utilidad de realizar acciones en cada estado, mientras que el segundo generalmente aprende la utilidad de encontrarse en los propios estados. Los métodos de aprendizaje por refuerzo se inspiran en conceptos de programación dinámica y definen fórmulas para actualizar las utilidades esperadas y para utilizarlas en la exploración del espacio de estados. Q-Learning se basa en la ecuación de programación dinámica de Bellman, que es una condición necesaria para la optimalidad asociada con la optimización matemática [1]. La regla Q-Learning básica es [23]:

$$Q(s, a) = Q(s, a) + \alpha (R + (\lambda \max_{a'} Q(s', a')) - Q(s, a))$$

donde, $Q(s, a)$ es el "valor actual" tal que $Q: S \times A \rightarrow \mathbb{R}$, α es la tasa de aprendizaje, λ es la tasa de descuento, $\max_{a'} Q(s', a')$ es el valor óptimo esperado, R es la recompensa, y $Q(s, a)$ es el nuevo valor de Q .

Existen diferentes plataformas empleadas en la creación de aplicaciones de aprendizaje automático. Por ejemplo, Knime, Accord.net, Scikit-Learn, Tensor-Flow, Weka, Pytorch, RapidMiner, Google Cloud AutoML, Jupyter Notebook, Apache Mahout, Studio Azure Machine Learning, MLLIB, Orange3, IBM Watson y Pylearn2. El lector interesado puede encontrar sin dificultad alguna información abundante al respecto en internet.

2.3. Computación Evolutiva.

La Computación Evolutiva (EC) (o Algoritmos Evolutivos (EA)) [10,11,12] es una familia de técnicas en las que se aplican modelos Darwinianos abstractos de evolución para refinar poblaciones de soluciones candidatas (conocidas como "individuos") a un problema dado. Un algoritmo evolutivo comienza con una población inicial de individuos generados aleatoriamente. Cada miembro de esta población es evaluado y se le asigna una aptitud (una evaluación de calidad). El EA luego utiliza un procedimiento orientado a la aptitud física para seleccionar, criar y mutar individuos para producir "hijos" que luego se agregan a la población, reemplazando a los individuos mayores, Fig. 9.



Figura 9. Esquema de un Algoritmo Genético

Los algoritmos coevolutivos (CEA) representan un enfoque natural para aplicar la computación evolutiva en la refinación de comportamientos multiagente [13]. En un CEA, la aptitud de un individuo se basa en su interacción con otros individuos de la población y, por lo tanto, la evaluación de la aptitud es sensible al contexto y subjetiva. En la coevolución competitiva, los individuos se benefician a expensas de sus pares; pero en la coevolución cooperativa, los individuos tienen éxito o fracasan juntos en colaboración.

Algunas plataformas para el desarrollo de algoritmos genéticos son: JCLEC, IlliGAL, GAGS, ECJ 16, Open Beagle, lil-gp, MAFRA, GALib, OPEAL, DREAM, EASEA y Shark.

2.4. Enfoques Cooperativos de Aprendizaje Multiagente

Comúnmente se acepta que hay dos enfoques cooperativos de aprendizaje multiagente [15]. El primero, que se conoce como *aprendizaje en equipo*, emplea un solo aprendiz que busca comportamientos para todo el equipo de agentes. Este enfoque está más en la línea de los métodos tradicionales de aprendizaje automático, pero presenta problemas de escalabilidad a medida que aumenta el número de agentes. Una segunda categoría es el *aprendizaje concurrente* [21], el cual utiliza múltiples procesos concurrentes de aprendizaje. En lugar de comportamientos de aprendizaje para todo el equipo, los enfoques de aprendizaje concurrentes generalmente emplean un aprendiz por cada miembro del equipo, con la esperanza de que esto reduzca el espacio conjunto de búsqueda al proyectarlo en N espacios separados. Sin embargo, la presencia de múltiples aprendices simultáneos hace que el entorno sea no-estacionario, lo cual viola las suposiciones detrás de la mayoría de las técnicas tradicionales de aprendizaje automático. Por esta razón, el aprendizaje concurrente requiere versiones nuevas (o significativamente modificadas de) métodos de aprendizaje automático.

2.5. Aprendizaje en Equipo

En el aprendizaje en equipo hay un solo aprendiz involucrado que descubre un conjunto de comportamientos para todo un equipo de agentes [14, 20]. Aun y cuando carece del aspecto de la teoría del juego con múltiples aprendices, plantea grandes desafíos porque a medida que los agentes interactúan entre sí, el comportamiento conjunto puede ser inesperado. A esta noción se le denomina la *complejidad emergente* del sistema multiagente. El aprendizaje en equipo se puede dividir en dos categorías: aprendizaje en equipos homogéneos y heterogéneos.

En el aprendizaje en equipos homogéneos, a todos los agentes se les asignan comportamientos idénticos, aunque los agentes no sean idénticos. Con ello, el espacio de búsqueda para el proceso de aprendizaje se reduce drásticamente, Fig. 10.



Figura 10. Equipos homogéneos

En el aprendizaje con equipos heterogéneos, el equipo está compuesto por agentes con diferentes comportamientos, con un solo aprendiz intentando mejorar al equipo en su conjunto. Este enfoque permite una mayor diversidad en el equipo a costa de aumentar el espacio de búsqueda.

En el aprendizaje con equipos híbridos, el conjunto de agentes se divide en varios escuadrones, y cada agente pertenece a un solo

escuadrón. Todos los agentes de un escuadrón tienen el mismo comportamiento. Un extremo (un solo escuadrón), es equivalente a un aprendizaje homogéneo en equipo, mientras que el otro extremo (un agente por escuadrón) equivale a un aprendizaje heterogéneo en equipo. El aprendizaje en equipo híbrido permite al experimentador lograr algunas de las ventajas de cada método. La Fig. 11 muestra un escenario típico de estudio de equipos híbridos de aprendizaje.

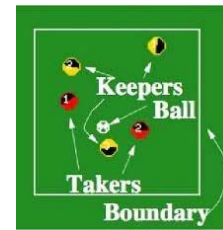


Figura 11. Keepaway Soccer Domain.
Equipo Híbrido de Aprendizaje.

3. APRENDIZAJE CONCURRENTE

La alternativa más común del aprendizaje en equipo en sistemas cooperativos multiagente es el aprendizaje concurrente [21,22]. En él, múltiples procesos de aprendizaje intentan mejorar partes del equipo. Por lo general, cada agente tiene un proceso de aprendizaje para modificar su propio comportamiento.

El desafío central para el aprendizaje concurrente es que cada aprendiz debe adaptar su comportamiento en el contexto de otros aprendices co-adaptados sobre los cuales no tiene control. En escenarios de un solo agente, un aprendiz explora su entorno y, al hacerlo, mejora su comportamiento. Las cosas cambian drásticamente con múltiples aprendices: a medida que los agentes aprenden, modifican sus comportamientos, lo que a su vez puede arruinar el aprendizaje de otros agentes. Un enfoque simplista para lidiar con esta co-adaptación es tratar a los otros aprendices como parte de un entorno dinámico al que debe adaptarse.

Hay tres ejes principales de investigación en el área del aprendizaje concurrente. En primer lugar, está el problema de la *asignación de créditos*, que se ocupa de cómo distribuir la recompensa obtenida a nivel de equipo entre los aprendices individuales. En segundo lugar, hay desafíos en la dinámica del aprendizaje que tiene como objetivo comprender el impacto de la co-adaptación en los procesos de aprendizaje. En tercer lugar, está el modelado de otros agentes con el fin de mejorar las interacciones y la colaboración con ellos. La figura 12 muestra un escenario de aprendizaje concurrente que se puede dar frecuentemente en la realidad. En este caso, los robots intentan aprender como manipular objetos entre ambos, de tal manera que se cumpla un objetivo preestablecido, como construir un objeto.

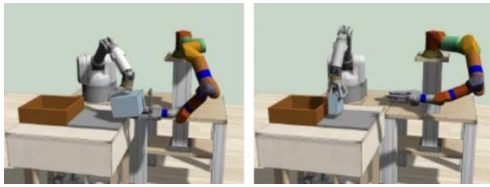


Figura 12. Un Escenario de Aprendizaje Concurrente

3.1. Aprendizaje en Ambientes Dinámicos.

Al aplicar el aprendizaje de un solo agente a entornos estacionarios, el agente experimenta con diferentes comportamientos hasta que, con suerte, descubre un comportamiento globalmente óptimo. En entornos dinámicos, el agente puede, en el mejor de los casos, tratar de mantenerse al tanto con los cambios habidos en el entorno y realizar un seguimiento constante del comportamiento óptimo cambiante. Las cosas son aún más complicadas en sistemas multiagente, donde los agentes pueden cambiar adaptativamente los entornos de aprendizaje de los demás, Fig. 13.



Figura 13. Entorno Dinámico de un Piso de Fabricación

Desafortunadamente, la gama de herramientas para modelar y analizar la dinámica de los aprendices concurrentes es muy limitada. Muchas de estas herramientas son particularmente adecuadas solo para ciertos métodos de aprendizaje, y solo unas pocas ofrecen un marco común para múltiples técnicas de aprendizaje. La teoría evolutiva de juegos es la principal herramienta en esta segunda categoría.

4. MODELADO DE COMPAÑEROS DE EQUIPO

Una área final de investigación en el aprendizaje concurrente es el modelado de compañeros de equipo: aprender sobre otros agentes en el entorno para hacer buenas conjeturas sobre su comportamiento esperado y actuar en consecuencia (para cooperar con ellos de manera más efectiva). En este enfoque normalmente se emplea un método de aprendizaje Bayesiano para actualizar los modelos de otros agentes [19]. Sobre la base de estos modelos, los agentes pueden estimar el comportamiento actual de otros agentes y tratar de cooperar mejor con ellos. Un enfoque similar consiste en aprender las creencias, capacidades y preferencias de los compañeros de equipo. El sistema almacena un conjunto de estos modelos junto con su probabilidad de ser correctos, con base en los comportamientos

observados de los otros agentes. Como es probable que otros agentes a su vez lo estén modelando, modelarlos trae a colación el espectro de la recursión infinita: "El agente A está haciendo X porque piensa que el agente B piensa que el agente A piensa que el agente B piensa que ..." Esto debe ser ordenado racionalmente en tiempo finito. Este comportamiento es estudiado en el área de la psicología del sentido común, Fig. 13.

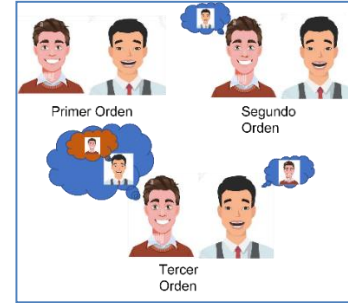


Figura 13. Psicología del Sentido Común

5. COMUNICACIÓN ENTRE AGENTES

La comunicación se puede definir como la alteración del estado del entorno de tal manera que otros agentes pueden percibir esta modificación y decodificar la información contenida. Entre otras razones, los agentes se comunican para coordinarse de manera más efectiva, distribuirse modelos más precisos del entorno y aprender entre sí soluciones de subtarefas [24, 25, 26, 27]. La comunicación explícita puede aumentar significativamente el espacio de búsqueda del método de aprendizaje, tanto al aumentar el tamaño del estado externo (ahora conoce la información de estado comunicada por otros agentes) como al aumentar las opciones disponibles del agente (agregando una acción de "comunicarse con el agente i"). Este aumento en el espacio de búsqueda puede dificultar el aprendizaje de un comportamiento óptimo más de lo que la comunicación misma puede ayudar. Por lo tanto, incluso cuando se requiere comunicación para un rendimiento óptimo, muchas aplicaciones ignoran la comunicación, o la codifican, para simplificar el proceso de aprendizaje, Fig. 14.



Figura 14. Componentes de la Comunicación entre Agentes

6. CONCLUSIÓN

El aprendizaje cooperativo multiagente es una área relativamente nueva de dinámicas complejas que promete una aplicabilidad generalizada para problemas complejos. Este problema se ha abordado tradicionalmente desde varias perspectivas: aprendizaje por refuerzo y entornos sin estados, computación

evolutiva, optimización estocástica, autómatas celulares y otros modelos de "complejidad de enjambre", robótica y teoría de juegos. Se revisaron algunos enfoques para el aprendizaje multiagente que se clasifican en aprendizaje en equipo y aprendizaje concurrente. También se abordó el tema crucial de la comunicación en el aprendizaje multiagente. El trabajo inicial en el aprendizaje multiagente generalmente se ha centrado en entornos relativamente simples. Finalizamos destacando áreas en las que se puede "escalar" a entornos más realistas e interesantes, como:

- Múltiples agentes. El "multi" exige un mayor número de agentes, en el rango de diez a miles o más. Los escenarios de dos y tres agentes son simplificaciones razonables para hacer factible el análisis teórico, pero falta esforzarse por más.
- Heterogeneidad del equipo. Si hay más de dos agentes, normalmente los agentes son idénticos en comportamiento y en capacidad. Lo común es que tengan diferentes capacidades, que las misiones requieran comportamientos diferentes, etc. Cuando hay un gran número de agentes, no es plausible contar con heterogeneidad total, pero los agentes pueden caer dentro de clases heterogéneas. Estas direcciones de investigación no se han explorado suficientemente.
- Agentes con estado interno u otra complejidad. Gran parte de las investigaciones simplifican los comportamientos de los agentes para reducir el espacio de búsqueda total y permitir que los agentes se entiendan entre sí más fácilmente. Pero muchos entornos multiagente, especialmente en robótica, se benefician considerablemente de entornos reactivos con más complejidad y, sobre todo, con estados internos ocultos [18].
- Cambios dinámicos de escenarios. ¿Qué sucede cuando varios agentes fallan o se les exige que realizar alguna tarea diferente? ¿Cuándo se agregan nuevos agentes al escenario? ¿Cuándo entran en línea nuevas habilidades? Es necesario considerar escenarios más dinámicos que se aproximen a situaciones más realistas. Como conclusión final podemos destacar que el problema es extraordinariamente complejo, y que además de requerir teorías más sofisticadas, la disponibilidad de plataformas y herramientas de desarrollo de sistemas prácticos aún está muy por detrás del estado actual de investigación.

Referencias.

- [1] S. Dreyfus, 'Richard Bellman on the birth of dynamic programming' Operations Research 50 (1), pp. 48–51, 2002.
- [2] B. Burns, "Designing Distributed Systems: Patterns and Paradigms for Scalable, Reliable Services", O'Reilly, 2018.
- [3] G. Weiss, et al., "Multiagent Systems", The MIT Press, ISBN: 9780262533874, 2016
- [4] M. Wooldridge, "An Introduction to MultiAgent Systems", 2nd Ed., Wiley, ISBN-10: 0470519460, 2009.
- [5] Y. Shoham, "Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations", Cambridge University Press, ISBN: 0521899435, 2008.
- [6] S. Russell & P. Norvig, "Artificial Intelligence: A Modern Approach" 4th Ed., Pearson, ISBN-10 : 0134610997, 2020.
- [7] N. Jennings, K. Sycara, and M. Wooldridge, "A roadmap of agents research and development", Autonomous Agents and Multi-Agent Systems, 1:7–38, 1998.
- [8] E. Alpaydin, "Introduction to Machine Learning", The MIT Press, ISBN-10: 0262043793, 2020.
- [9] C. Goldman and J. Rosenschein, "Mutually supervised learning in multiagent systems", in G. Weiß and S. Sen, editors, "Adaptation and Learning in Multi-Agent Systems", pages 85–96. Springer-Verlag: Heidelberg, Germany, Berlin, 1996.
- [10] K.A.De Jong, "Evolutionary Computation: A Unified Approach", ISBN-10 : 0262529602, A Bradford Book, 2016.
- [11] T. Bäck. Evolutionary Algorithms in Theory and Practice: Evolutionary Strategies, Evolutionary Programming, and Genetic Algorithms. Oxford Press, 1996.
- [12] J. K. Bassett. A study of generalization techniques in evolutionary rule learning. Master's thesis, George Mason University, Fairfax VA, USA, 2002.
- [13] D. Andre and A. Teller. Evolving team Darwin United. In M. Asada and H. Kitano, editors, RoboCup-98: Robot Soccer World Cup II. Springer Verlag, 1999.
- [14] E. Bonabeau, M. Dorigo, and G. Theraulaz. Swarm Intelligence: From Natural to Artificial Systems. SFI Studies in the Sciences of Complexity. Oxford University Press, 1999
- [15] M. Bowling. Multiagent learning in the presence of agents with limitations. PhD thesis, Computer Science, Department, Carnegie Mellon University, 2003.
- [16] O. Buffet, A. Dutech, and F. Charpillet. Incremental reinforcement learning for designing multi-agent systems.
- [17] In J. P. M'uller, E. Andre, S. Sen, and C. Frasson, editors, Proceedings of the Fifth International Conference on Autonomous Agents, pages 31–32, Montreal, Canada, 2001. ACM Press.
- [18] H. Bui, S. Venkatesh, and D. Kieronska. Learning other agents' preferences in multi-agent negotiation using the Bayesian classifier. International Journal of Cooperative Information Systems, 8(4):275–294, 1999.
- [19] G. Chalkiadakis and C. Boutilier. Coordination in multiagent reinforcement learning: A Bayesian approach, In Proceedings of The Second International Joint Conference on Autonomous Agents & Multiagent Systems (AAMAS 2003). ACM, 2003. ISBN 1-58113-683-8.
- [20] R. H. Crites. Large-Scale Dynamic Optimization Using Teams of Reinforcement Learning Agents. PhD thesis, University of Massachusetts Amherst, 1996.
- [21] M. R. Cutkosky, R. S. Englemore, R. E. Fikes, M. R. Genesereth, T. R. Gruber, W. S. Mark, J. M. Tenenbaum, and J. C. Weber. PACT: An experiment in integrating concurrent engineering systems. In M. N. Huhns and M. P. Singh, editors, Readings in Agents, pages 46–55. Morgan Kaufmann, San Francisco, CA, USA, 1997.
- [22] A. Garland and R. Alterman. Autonomous agents that learn to better coordinate. Autonomous Agents and Multi-Agent Systems, 8:267–301, 2004.
- [23] A. Greenwald and K. Hall. Correlated Q-learning. In Proceedings of the Twentieth International Conference on Machine Learning, 2003.
- [24] Panait, L., Luke, S. Cooperative Multi-Agent Learning: The State of the Art. Auton Agent Multi-Agent Syst 11, 387–434 (2005).
<https://doi.org/10.1007/s10458-005-2631-2>
- [25] A Review of Cooperative Multi-Agent Deep Reinforcement Learning Afshin OroojlooyJadid, Davood Hajinezhad, 2021.
<https://doi.org/10.48550/arXiv.1908.03963>
- [26] Distributed Cooperative Multi-Agent Reinforcement Learning with Directed Coordination Graph Gangshan Jing, He Bai, Jemin George, Aranya Chakraborty, Piyush. K. Sharma, 2022.
<https://doi.org/10.48550/arXiv.2201.04962>
- [27] Pierre Haou, Multi-Agent Reinforcement Learning (MARL) and Cooperative AI, 2021, <https://towardsdatascience.com/ive-been-thinking-about-multi-agent-reinforcement-learning-marl-and-you-probably-should-be-too-8f1e241606ac>
- [28] Marchetti, T.J. ; García, A.J. Plataformas para Desarrollo de Sistemas Multiagente - Un Analisis Comparativo.
<https://core.ac.uk/download/pdf/15778237.pdf>