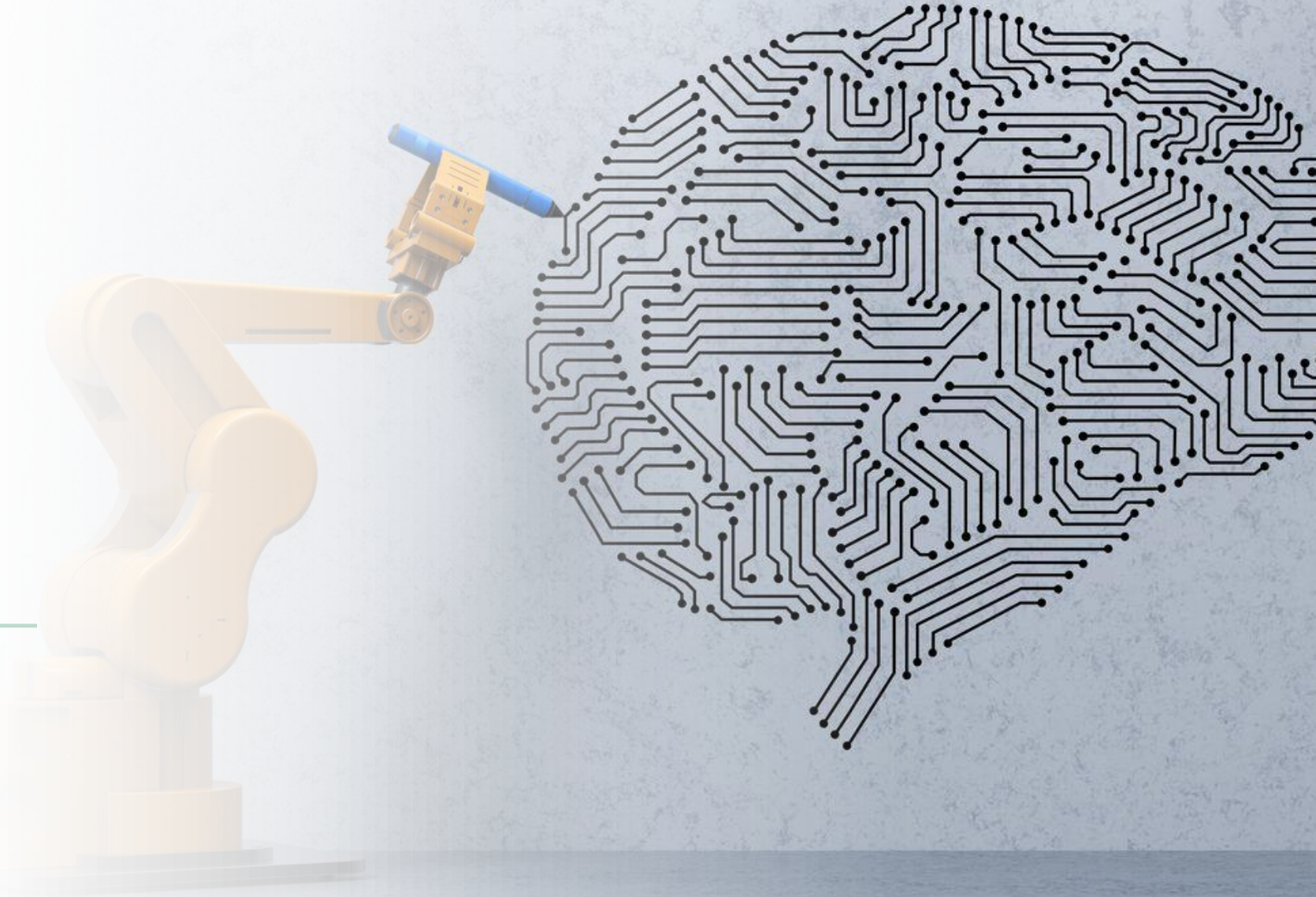


# Aprendizaje por refuerzo

Clase 22: RL multi-agente II





## Para el día de hoy...

- Juegos de Markov
- Sistemas multi-agentes
- Juegos
- Arquitecturas multi-agente





# Antes de empezar...

- Dudas del proyecto

# Juego de Markov

- Es una tupla  $G = (N, S, A, P, \{R_i\}_{i \in N}, \delta)$
- Donde
  - $N = \{1, \dots, n\}$  es un conjunto de jugadores
  - $S$  es el espacio de estados
  - $A = A_1 \times \dots \times A_n$  es el espacio de acciones donde  $A_i$  es el conjunto de acciones de  $i$
  - Para estados  $s \in S$  y  $a \in A$ ,  $P(\cdot, s, a)$  es la distribución de probabilidad  $P_{i,s \rightarrow s'}^a$
  - Para estados  $s \in S$  y  $a \in A$ ,  $R_i(s'|s, a)$  es la recompensa  $R_{i,s \rightarrow s'}^a$
  - $\delta \in (0,1)$  es un factor de descuento

# Tipos de sistemas multi-agente



## Cooperativos

Recompensa conjunta  
Problema de coordinación



## Competitivo

Juegos de suma cero  
Recompensas opuestas  
Equilibrio minimax



## Mixtos

Juegos de suma general  
Equilibrio de Nash

## El estado del arte en juegos

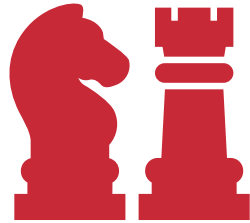
| Program            | Level of Play | Program to Achieve Level  |
|--------------------|---------------|---|
| Checkers           | Perfect       | <i>Chinook</i>  |
| Chess              | Superhuman    | <i>Deep Blue</i>  |
| Othello            | Superhuman    | <i>Logistello</i>   |
| Backgammon         | Superhuman    | <i>TD-Gammon</i>  |
| Scrabble           | Superhuman    | <i>Maven</i>  |
| Go                 | Grandmaster   | <i>MoGo<sup>1</sup>, Crazy Stone<sup>2</sup>, Zen<sup>3</sup></i> |
| Poker <sup>4</sup> | Superhuman    | <i>Polaris</i>  |

# Un par de enfoques

- La mejor respuesta es la solución al problema de RL con un solo agente
  - Los otros agentes se tratan como parte del ambiente
  - El juego se reduce a un MDP
  - La mejor respuesta es la política óptima del MDP
- Equilibrio de Nash es el punto fijo para juego en solitario
  - Se genera experiencia de los agentes
  - Cada agente aprende la mejor respuesta a los otros agentes
  - La política de un agente determina el ambiente del otro
  - Todos los jugadores se adaptan a los otros

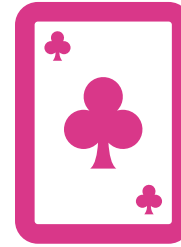
# Juegos de información perfecta e imperfecta

---



**Un juego de información perfecta es completamente observable (Juegos de Markov)**

Ajedrez  
Damas inglesas  
Otelo  
Backgammon  
Go



**Información imperfecta**

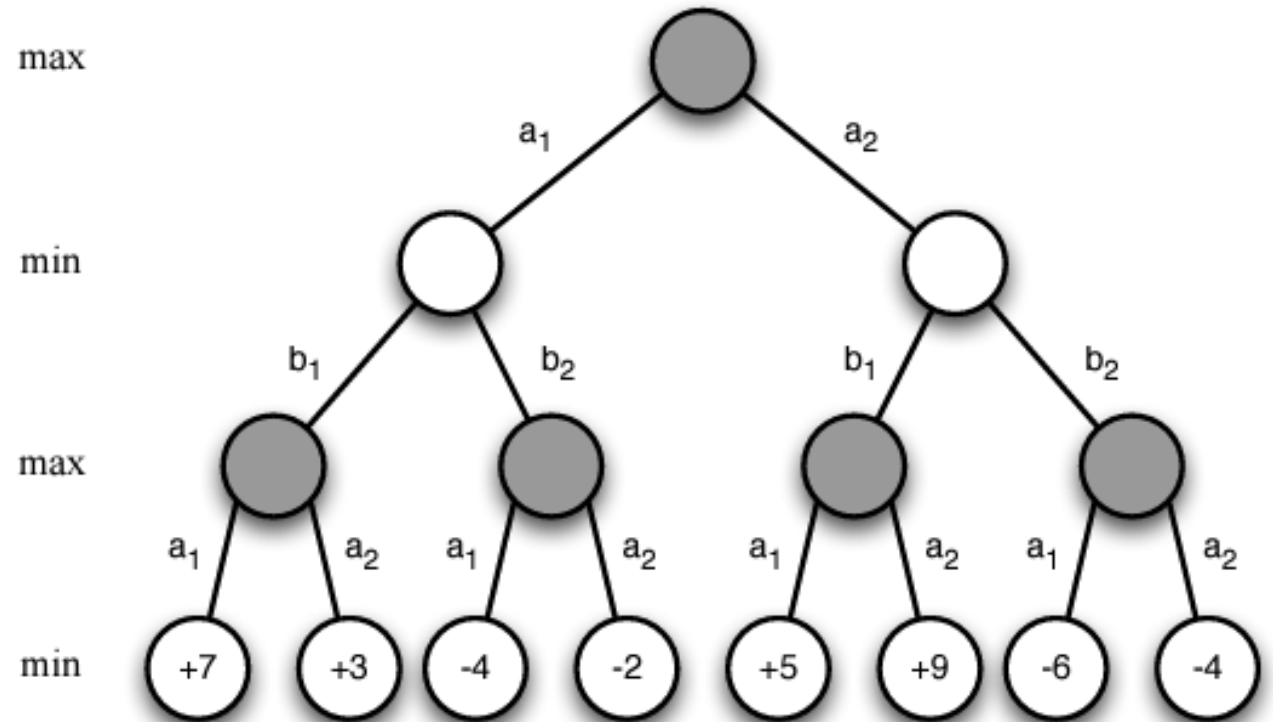
Scrabble  
Poker



# Minimax

- Una función de valor define la recompensa esperada total de las políticas  $\pi = (\pi^1, \pi^2)$
- $v^\pi(s) = \mathbb{E}_\pi[G_t | S_t = s]$
- La función de valor minimax resuelve
- $v_*(s) = \max_{\pi^1} \min_{\pi^2} v_\pi(s)$
- Una política minimax es la política  $\pi = (\pi^1, \pi^2)$  que alcanza el valor minimax
- La política minimax es un equilibrio de Nash

# Búsqueda minimax




# Función de valor en minimax

- El árbol de búsqueda crece exponencialmente
- Es impráctico llegar al final del juego
- Se suele usar alguna función de aproximación  $v(s, w) \approx v_*(s)$
- Utiliza funciones de valor para estimar el valor de las hojas
- Minimax se ejecuta a alguna profundidad fija

# Función de valor lineal



$$v(s, \mathbf{w}) = \mathbf{x}(s) \cdot \mathbf{w} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ \vdots \end{bmatrix} \cdot \begin{bmatrix} +5 \\ +3 \\ +1 \\ -5 \\ -3 \\ -1 \\ \vdots \end{bmatrix}$$



$$v(s, \mathbf{w}) = 5 + 3 - 5 = 3$$

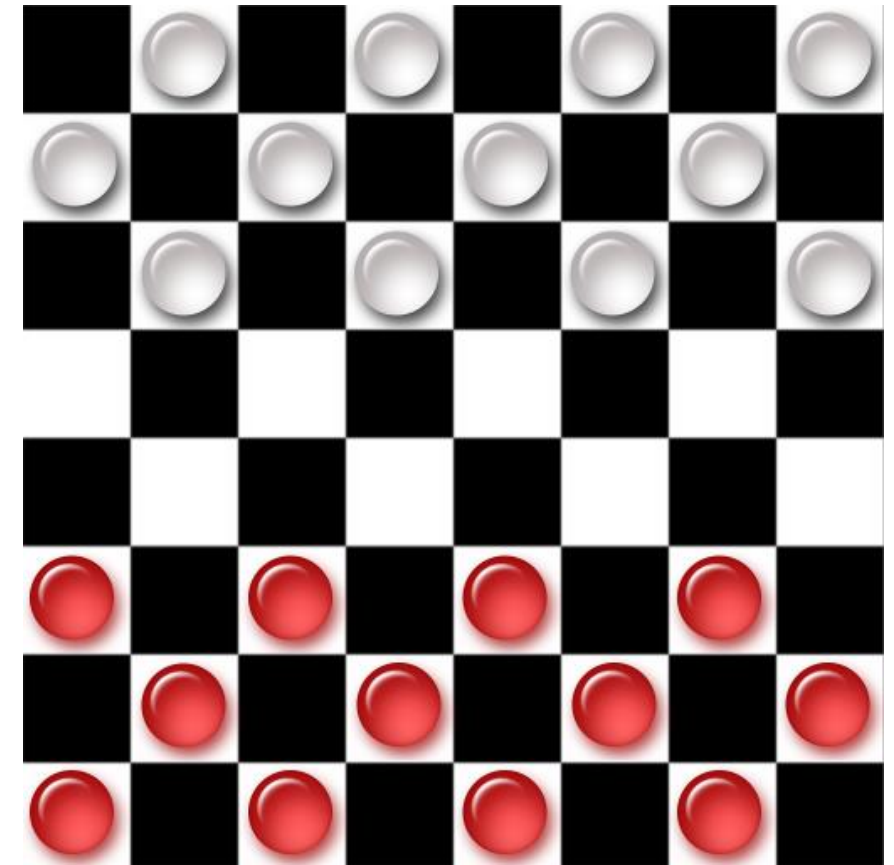
# Deep Blue

- Conocimiento
  - 8000 características desarrolladas a mano
  - Funciones de valor con aproximación lineal
  - Los pesos fueron ajustados por humanos expertos
- Búsqueda
  - Búsqueda en paralelo con búsqueda alfa-beta
  - 200 millones de posiciones por segundo
- Resultados
  - Derrotó a Garry Kasparov 4-2 (1997)



# Chinook

- Conocimiento
  - Funciones lineales
  - 12 características
  - 4 etapas del juego
- Búsqueda
  - Alfa beta
  - Análisis
    - Búsqueda hacia atrás de posiciones de victoria
    - Almacena todas las posiciones de victoria
    - Juega perfecto con las últimas n piezas
- Resultado
  - Derrota a Marion Tinsley en 1994
  - Se resuelve el juego en 2007



# Juego en solitario con diferencia temporal

- Usar algoritmos de RL basados en valor para generar juegos
  - MC
  - $TD(0)$
  - $TD(\lambda)$

# Mejora de política

- Para juegos deterministas, estimar  $v_*(s)$
- Es posible evaluar
  - $q_*(s, a) = v_*(succ(s, a))$
- Las reglas del juego determinan el estado sucesor  $succ(s, a)$
- Las acciones son seleccionadas maximizando/minimizando
  - $A_t = \arg \min_a v_*(succ(S_t, a))$  para el jugador 1
  - $A_t = \arg \max_a v_*(succ(S_t, a))$  para el jugador 2
- Esto mejora la política de ambos jugadores

# Logistello

## Representación

- Genera sus propias características
- Inicia con características crudas
- Construye características con conjunciones y disyunciones
- Genera 1.5 millones de características
- Funciones de aproximación lineales

## Algoritmo

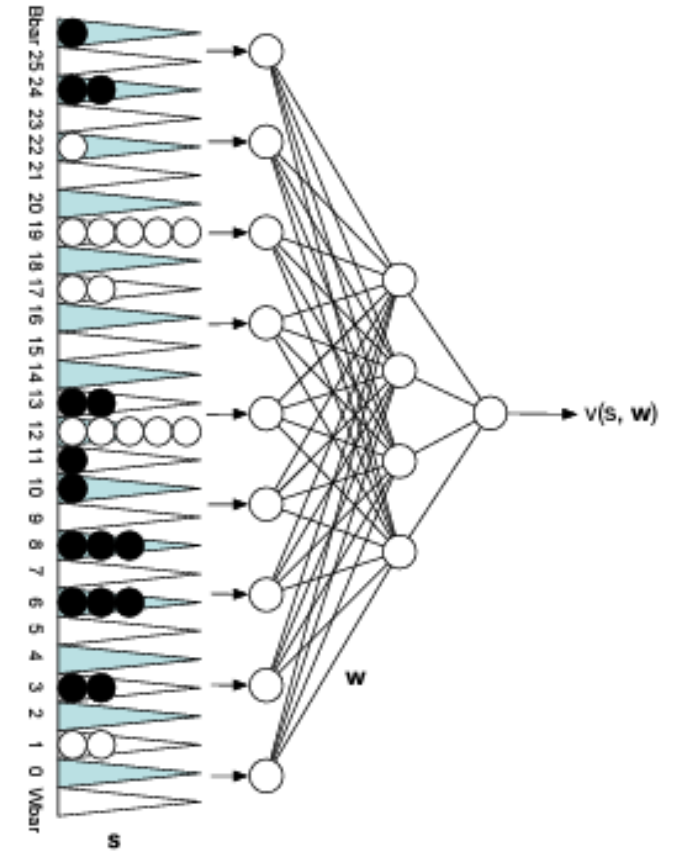
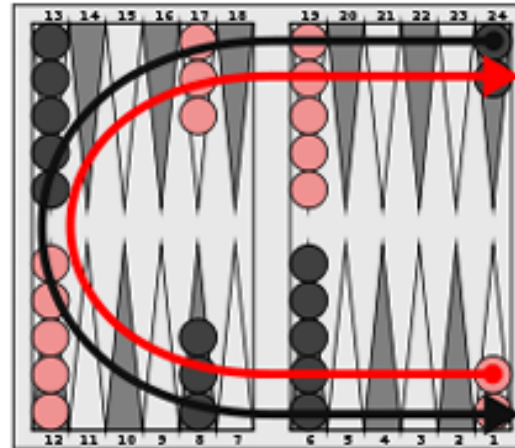
- Iteración de política
- Juego en solitario
- Evaluación de Monte-Carlo
- Mejora de política voraz

## Resultado

- Derrota a Takeshi Murukami 6-0

# TD Gammon

- Red neuronal inicializada aleatoriamente
- Juego en solitario
- Aprendizaje de diferencia temporal
- Política voraz
- Siempre converge en la práctica
- Derrota a Luigi Villa 7-1 (1992)





# Búsqueda basada en simulación

- El juego en solitario puede reemplazar la búsqueda
- Simula juegos desde la raíz
- Utiliza RL para simular experiencia
  - Monte-Carlo Tree Search
  - UCB

# Desempeño de MCTS en juegos

- Consigue muy buenos resultados en juegos como
  - Go
  - Hex
  - Lines of action
- Búsqueda de Monte-Carlo puede funcionar en
  - Scrabble
  - Backgammon

# Maven

- Aprendizaje
  - Aproximación de funciones lineales
  - Iteración de política con Monte-Carlo
- Búsqueda
  - Imaginar n pasos de juego en solitario
  - Evaluar la posición resultante por el puntaje
  - Seleccionar el movimiento con el puntaje más alto
- Resultado
  - Derrota a Adam Logan 9-5
  - Maven tenía un error promedio de 3 puntos por juego

|                |                |                |                |                |                 |                |                |                |                 |                |                |                |                |    |
|----------------|----------------|----------------|----------------|----------------|-----------------|----------------|----------------|----------------|-----------------|----------------|----------------|----------------|----------------|----|
| H <sub>3</sub> | O <sub>1</sub> | U <sub>1</sub> | T <sub>1</sub> | H <sub>4</sub> |                 | A <sub>1</sub> | R <sub>1</sub> | T <sub>1</sub> |                 |                | 2L             |                |                | 3W |
| A <sub>1</sub> | E <sub>1</sub> |                |                |                | 3L              |                |                |                | Q <sub>10</sub> |                |                |                |                | 2W |
| T <sub>1</sub> |                | 2W             |                |                |                 | 2L             |                | 2L             | U <sub>1</sub>  |                |                | G <sub>2</sub> |                |    |
| H <sub>4</sub> | U <sub>1</sub> | R <sub>1</sub> | T <sub>1</sub> |                |                 |                | 2L             |                | A <sub>1</sub>  |                | 2W             | R <sub>1</sub> |                | 2L |
|                | N <sub>1</sub> | E <sub>1</sub> | O <sub>1</sub> | N <sub>1</sub> |                 |                |                |                | I <sub>1</sub>  | S <sub>1</sub> |                | E <sub>1</sub> |                | L  |
|                | 3L             |                | D <sub>2</sub> | O <sub>1</sub> | Z <sub>10</sub> | Y <sub>4</sub> |                |                | 3L              | P <sub>3</sub> |                | A <sub>1</sub> | X <sub>8</sub> | E  |
|                |                | E <sub>1</sub> |                |                |                 | E <sub>1</sub> |                | 2L             | J <sub>8</sub>  | A <sub>1</sub> | W <sub>4</sub> | S <sub>1</sub> |                | I  |
| I <sub>1</sub> | A <sub>1</sub> | H <sub>3</sub> | B <sub>3</sub> |                | C <sub>3</sub>  | A <sub>1</sub> | V <sub>4</sub> | Y <sub>4</sub> |                 | N <sub>1</sub> | 2L             | E <sub>1</sub> |                | 3W |
|                | W <sub>4</sub> | E <sub>1</sub> |                |                |                 | R <sub>1</sub> |                | 2L             |                 | K <sub>5</sub> |                | 2L             |                |    |
|                | 3L             | N <sub>1</sub> |                | F <sub>4</sub> | 3L              | L <sub>1</sub> |                |                | B <sub>3</sub>  |                |                |                |                | 3L |
|                |                | D <sub>2</sub> |                | E <sub>1</sub> |                 | O <sub>1</sub> |                |                | O <sub>1</sub>  | R <sub>1</sub> |                |                |                |    |
| 2L             | D <sub>2</sub> | E <sub>1</sub> | V <sub>4</sub> | I <sub>1</sub> | A <sub>1</sub>  | N <sub>1</sub> | C <sub>3</sub> | E <sub>1</sub> | S <sub>1</sub>  |                | 2W             |                |                | 2L |
|                |                | D <sub>2</sub> |                | G <sub>2</sub> |                 | G <sub>2</sub> | O <sub>1</sub> | 2L             |                 |                |                | 2W             |                |    |
|                | 2W             |                |                | N <sub>1</sub> | 3L              |                | F <sub>4</sub> |                | 3L              |                |                |                | 2W             |    |
| P <sub>3</sub> | I <sub>1</sub> | L <sub>1</sub> | I <sub>1</sub> | S <sub>1</sub> |                 |                | T <sub>1</sub> | U <sub>1</sub> | T <sub>1</sub>  | O <sub>1</sub> | R <sub>1</sub> | I <sub>1</sub> | A <sub>1</sub> | L  |

# Juegos de información imperfecta

- Los jugadores tienen árboles diferentes
- Existe un nodo que contine la información que el agente conoce
- Métodos
  - Métodos de búsqueda hacia adelante
  - Juego en solitario
- Resultado
  - 3 medallas de plata en Poker (limit Hold'em) para 2 y 3 jugadores

# Un algoritmo UCT Search

- Usar MCTS al árbol de estado de información del juego
- Los agentes aprenden y responden al comportamiento promedio
- Extrae la estrategia promedio de los nodos de acción
- En cada nodo, elige una acción de acuerdo a alguna probabilidad



# Algunos otros enfoques

## Independent Q-learning [Tan 1993]

- Cada agente aprende de forma independiente su función Q
- Cada agente trata al resto como parte del ambiente

## Independent actor-critic [Foerster et al. 2018]

- Cada agente aprende independientemente con su propio actor crítico
- Cada agente trata al resto como parte del ambiente

## Aprendizaje rápido con parámetros compartidos

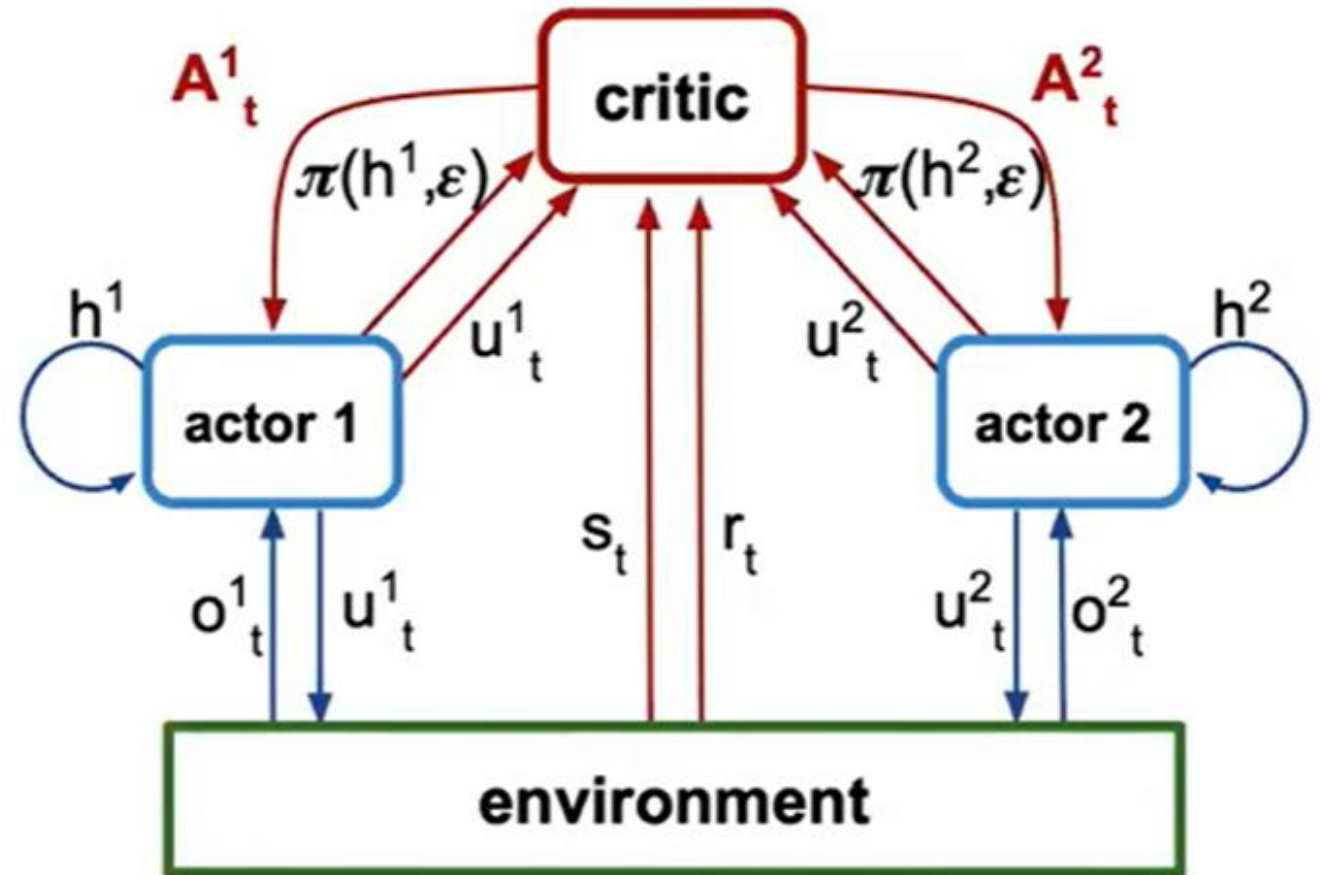
- Diferentes entradas inducen diferentes comportamientos

## Limitaciones

- Aprendizaje no estacionario
- Difícil de aprender a coordinarse

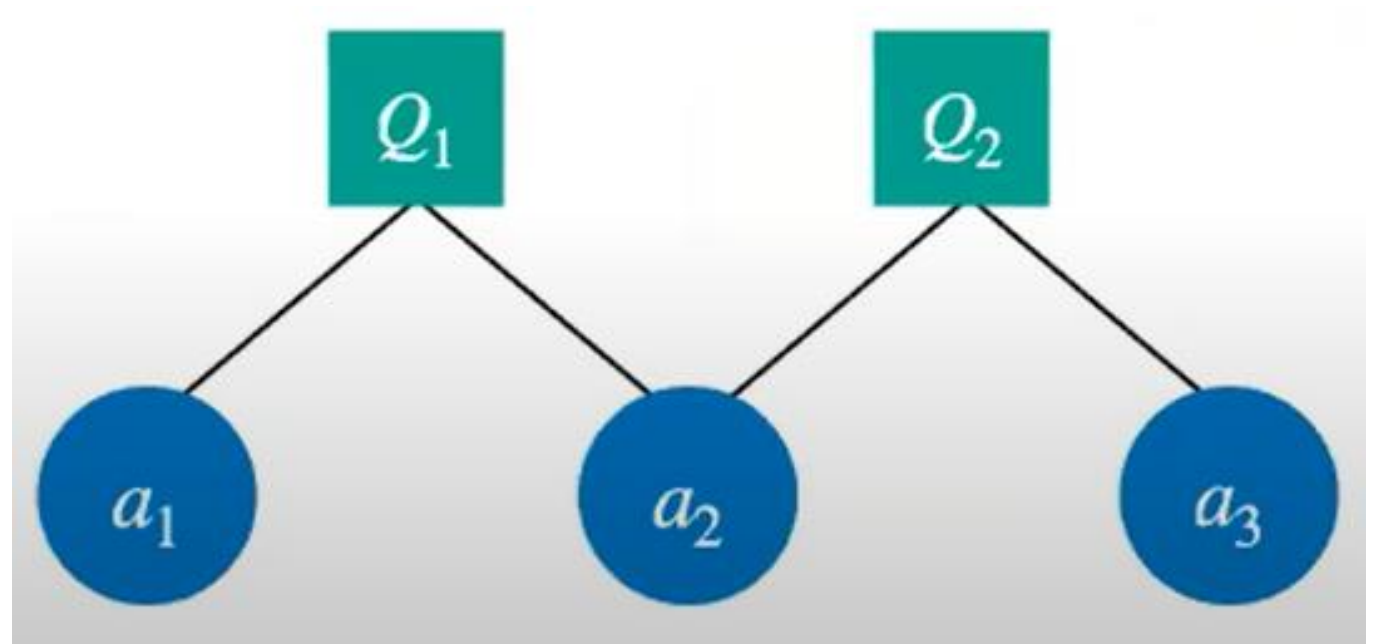
# Centralised critics

- Lowe et al. 2017; Foerster et al. 2018
- Centralizar  $V(s, \tau)$  o  $Q(s, \tau, u)$



# Factored value functions

- Guestrin et al. 2003
- Permite mejorar escalabilidad
- $Q_{tot} = (\tau, u, \theta) = \sum_{(e=1)}^E Q_e(\tau^e, u^e, \theta^e)$
- Donde cada  $e$  indica un subconjunto de agentes



A detailed photograph of a butterfly with black and white wings, featuring a complex pattern of black veins and spots on a white background. The butterfly is perched on a cluster of small, bright red flowers with yellow centers. The background is a soft, out-of-focus green and yellow, suggesting a natural habitat. The word "Coevolución" is overlaid in white text on the left side of the image.

# Coevolución

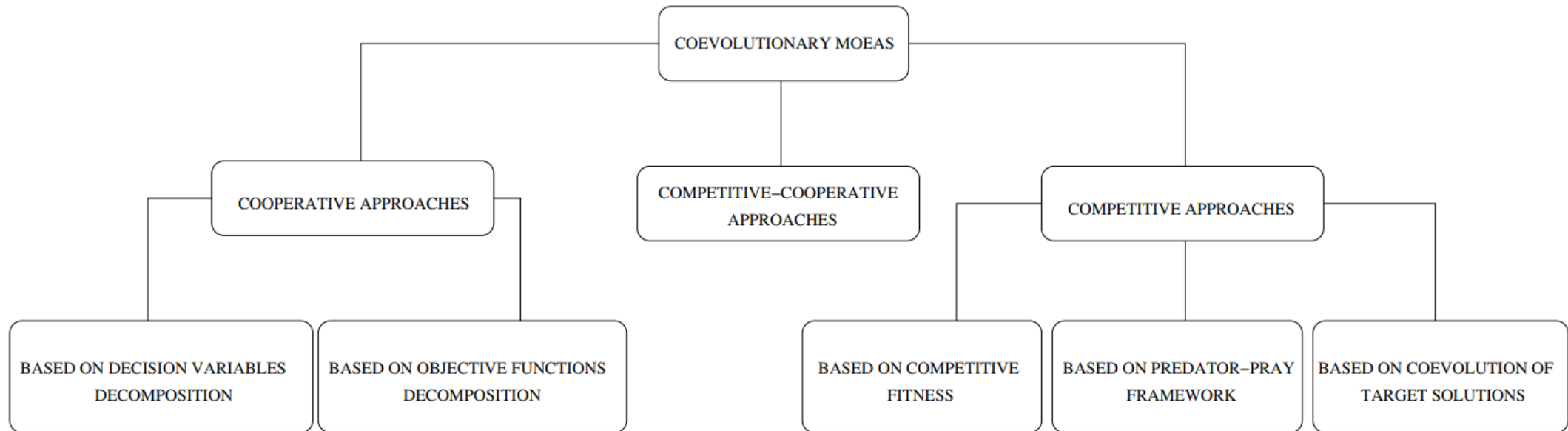
# Coevolución

- Llamaremos coevolución a un cambio en la composición genética de una especie como respuesta a un cambio de otra
- Se ha utilizado para juegos, incertidumbre, problemas de gran escala entre otros
- En general existen dos clases de coevolutivos
  - Competitivos
  - Cooperativos

|              | A | B |                                    |
|--------------|---|---|------------------------------------|
| Neutralismo  | 0 | 0 | Las poblaciones son independientes |
| Mutualismo   | + | + | Ambas se benefician                |
| Comensalismo | + | 0 | A se beneficia                     |
| Competición  | - | - | Las dos se perjudican              |
| Depredador   | + | - | A se beneficia y B se perjudica    |
| Parasito     | + | - | A se beneficia y B se perjudica    |



# Taxonomia de coevolutivos



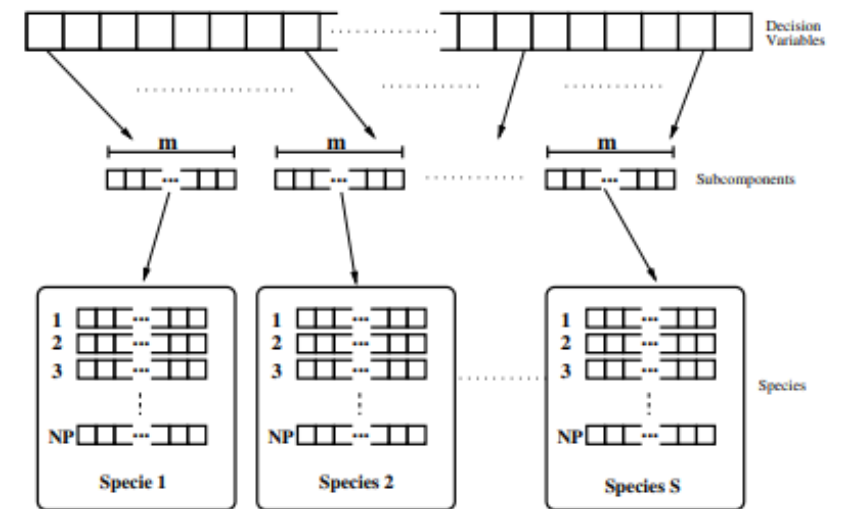
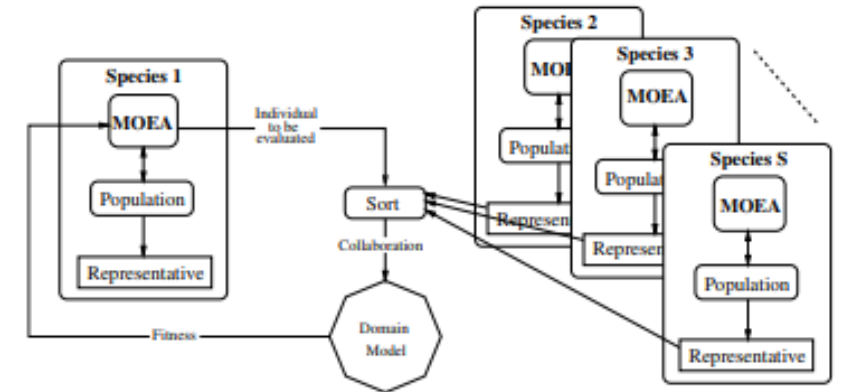
# MOEAs coevolutivos cooperativos

## Basados en descomposición de variables de decisión

- Descomponen el espacio de búsqueda. Una variable es asignada a una especie y cada especie optimiza una o varias variables de decisión. Para la evaluación es necesario combinar individuos de cada especie
- Basados en indicadores
- Para Hyperspectral sparse unimixing

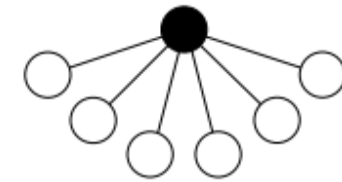
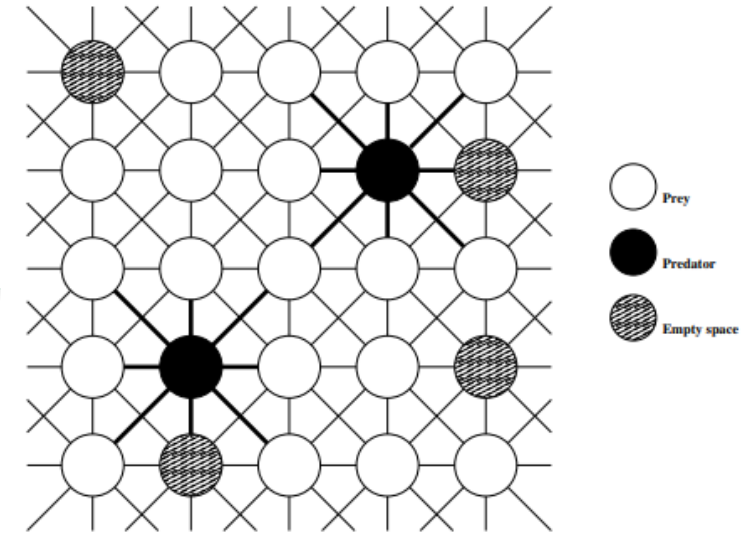
## Basados en descomposición de funciones objetivo

- Cada función es asignada a una especie
- Evolución diferencial con múltiples poblaciones para múltiples objetivos
- Basada en equilibrio de Nash
- Basado en preferencias utilizando vectores de pesos

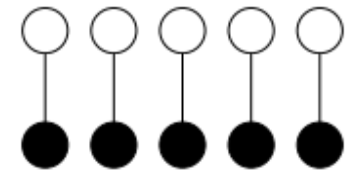


# MOEAs coevolutivos competitivos

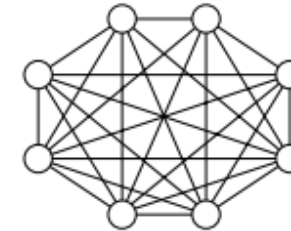
- CMOEAs basados en el modelo depredador-presa
  - Una presa representa un individuo de la población y el depredador “caza” a la presa más débil de acuerdo al valor en algún objetivo en particular
- CMOEAs basados en aptitud competitiva
  - Se utiliza una función objetivo que toma en cuenta las dependencias entre las especies
  - La aptitud de los individuos se compara con respecto a los otros y aquel con mejor aptitud gana la competencia
- CMOEAs basados en coevolución de soluciones objetivo
  - Realizan competencia entre dos poblaciones una de posibles soluciones y otra de valores deseados



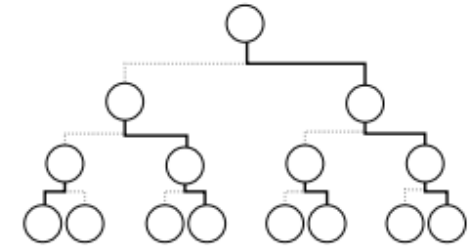
(a) All vs. all



(b) Bipartite



(c) All vs. best



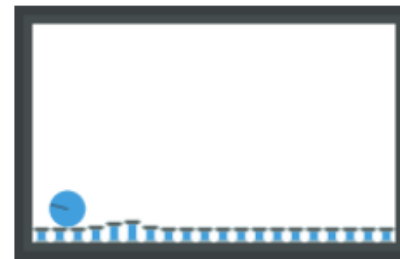
(d) Tournament

# PettingZoo



Atari

Multi-player Atari 2600 games (both cooperative and competitive)



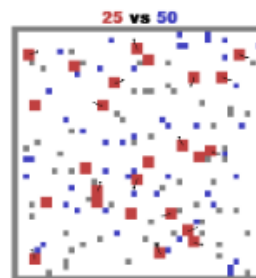
Butterfly

Cooperative graphical games developed by us, requiring a high degree of coordination



Classic

Classical games including card games, board games, etc.



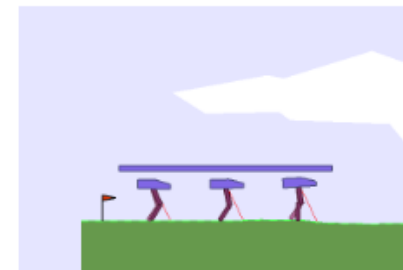
MAgent

Configurable environments with massive numbers of particle agents



MPE

A set of simple nongraphical communication tasks originally from <https://github.com/openai/multiagent-particle-envs>



SISL

3 cooperative environments, originally from <https://github.com/sisl/MADRL>



Para saber  
más

---

*QMIX: Monotonic Value Function Factorisation  
for Deep Multi-Agent Reinforcement Learning, **ICML-18***

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt,  
Gregory Farquhar, Jakob Foerster, & Shimon Whiteson

*Monotonic Value Function Factorisation for Deep Multi-Agent  
Reinforcement Learning, **Conditionally accepted to JMLR***

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt,  
Gregory Farquhar, Jakob Foerster, & Shimon Whiteson

*MAVEN: Multi-Agent Variational Exploration, **NeurIPS-19***

Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, & Shimon Whiteson





# Para la otra vez...

- RL multi-tarea



The End.



iimas