Pràctica Bloc II

Climent Alzamora Alcover
41657734L
caa232@id.uib.eu

Joan Tenerife Perelló
43481522E
jtp409@id.uib.eu

Assignatura: Intel·ligència Artificial
Grau: Enginyeria Informàtica
Data d'entrega: 24 de novembre 2025

Universitat de les Illes Balears

Índex

1 Introducció i Descripció del Problema

1.1 L'Entorn Frozen Lake

El problema central resolt en aquesta pràctica és el joc del **Frozen Lake** (Llac Congelat), un **l·laberintçlàssic** en l'àmbit de l'Aprenentatge per Reforç (Reinforcement Learning - RL). L'objectiu de l'agent és aconseguir arribar a la **meta** des del punt de partida sense caure en cap dels **forats** dispersos pel mapa. Si l'agent arriba a la meta, guanya; si cau en un forat, perd.

Per a la simulació d'aquest entorn, s'ha utilitzat la biblioteca **Gymnasium** (desenvolupada originalment per OpenAI) dins del llenguatge de programació **Python**.

1.2 Objectius de la Pràctica

L'objectiu principal de la pràctica ha estat l'aplicació i l'experimentació amb **diferents algorismes d'Aprenentatge per Reforç** vistos a classe per resoldre el problema del Frozen Lake. Els algorismes utilitzats inclouen: **Q-Learning**, **SARSA** i **Montecarlo**. Addicionalment, s'ha dissenyat i implementat un **Algorisme Genètic** com a mètode alternatiu de solució.

2 Metodologia: Aprenentatge per Reforç (RL)

L'Aprenentatge per Reforç (RL) és un paradigma d'Aprenentatge Automàtic en el qual un agent aprèn a prendre decisions a través de la interacció amb un entorn. L'objectiu és aprendre una política que aconsegueixi **maximitzar la recompensa acumulada** al llarg del temps.

2.1 Decisions de Disseny Comunes

En la implementació dels algorismes d'Aprenentatge per Reforç (Q-Learning i SARSA), s'han establert els mateixos **paràmetres** inicials per garantir una base de comparació consistent.

- El **Factor de Descompte** (γ) sempre s'ha definit a **0.9**. Aquest valor determina la importància de les recompenses futures.
- La **Taxa d'Aprenentatge** (α) s'ha definit inicialment a **0.9**. Controla la rapidesa amb què l'agent actualitza els valors Q.
- **Epsilon** (ϵ) **i el** *decay_rate*: Hem definit Epsilon inicialment a **1** i la seva taxa de decrement ha estat de **0.0001** per a cada episodi, gestionant l'equilibri entre exploració i explotació.

2.2 Implementació de Monte Carlo

L'algorisme de **Montecarlo** és un mètode d'Aprenentatge per Reforç que, a diferència de Q-Learning i SARSA, **no es basa en estimacions futures pas a pas**, sinó en el resultat complet de l'episodi. Pren les decisions basant-se en la **recompensa total observada** des d'un estat donat fins al final de l'episodi.

A l'hora de construir el codi ens hem basat amb el pseudocodi que trobem a l'aula digital. El que destaquem és que l'equació d'actualització **no es realitza pas a pas**, sinó un cop **finalitzat l'episodi** i s'ha obtingut la recompensa total (G_t):

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[G_t - Q(S_t, A_t)]$$

La funció clau és el terme $\mathbf{G_t}$, que representa la **suma total de recompenses futures descomptades** observades des de l'estat S_t fins al final de l'episodi (el retorn).

2.3 Implementació de SARSA

L'algorisme **SARSA** és un mètode d'Aprenentatge per Reforç que es basa en el mètode **On-Policy** (Dins-política). La presa de decisions d'aquest algorisme es basa en les accions dutes a terme anteriorment. L'agent utilitza la seva pròpia acció per a l'actualització, fent que aprengui a evitar el risc que l'ha fet perdre. Això comporta que l'agent aprengui una política que minimitza els riscos.

A l'hora de construir el codi ens hem basat amb el pseudocodi que trobem a l'aula digital. El que destaquem és l'equació d'actualització, que utilitza el valor de la pròxima acció ($\mathbf{A_{t+1}}$) escollida per la política d'exploració:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma Q(S_{t+1}, \mathbf{A_{t+1}}) - Q(S_t, A_t)]$$

La funció clau és el terme $\gamma Q(S_{t+1}, \mathbf{A_{t+1}})$, que utilitza el valor de la pròxima acció que escollirà la política d'exploració.

2.4 Implementació de Q-Learning

El segon algorisme implementat és el **Q-Learning**, el qual es basa en el mètode **Off-Policy** (Fora-política). Aquest és capaç de prendre decisions sense haver de dependre de la pròxima acció que la política actual faria. Pretén maximitzar la recompensa amb el menor nombre d'accions per arribar a la solució.

A l'hora de construir el codi també ens hem basat amb el pseudocodi que trobem a l'aula digital. El que destaquem és l'equació d'actualització, que utilitza el valor de la **millor acció possible** del pròxim estat:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_{\mathbf{a}} \mathbf{Q}(\mathbf{S_{t+1}}, \mathbf{a}) - Q(S_t, A_t)]$$

La funció clau és el terme $\gamma \max_{\mathbf{a}} \mathbf{Q}(\mathbf{S_{t+1}}, \mathbf{a})$, que fa que utilitzi el valor de la millor acció possible des del nou estat S_{t+1}, independentment de quina acció s'hagi escollit a l'episodi següent.

3 Metodologia: Algorisme Genètic (AG)

L'Algorisme Genètic (AG) s'ha aplicat com a alternativa per trobar una **seqüència d'accions** que porti a l'objectiu, en lloc d'aprendre una política dinàmica com en RL.

- **Representació (Cromosoma):** Una seqüència fixa de $L = 50$ accions, on cada acció (gen) és un enter $\{0, 1, 2, 3\}$ (Amunt, Avall, Esquerra, Dreta). `CHROMOSOME_LENGTH = 50`.

- **Població:** $N = 200$ individus. `POPULATION_SIZE = 200`.
- **Funció de Fitness:** El *fitness* es basa en el **Retorn Descomptat** (G) obtingut en executar la seqüència. A més, s'aplica **Shaping** addicional:
  1. Penalització si es cau en un forat.
  2. Recompensa addicional (negativa) basada en la distància de Manhattan a la casella objectiu.

  Això permet discriminar millor entre seqüències que no arriben a la meta (recompensa esparsa).
- **Selecció: Torneig** de mida $k = 3$. `TOURNAMENT_SIZE = 3`.
- **Creuament: Un Punt** amb probabilitat `CROSSOVER_RATE = 0.8`.
- **Mutació:** Mutació independent per gen, substituint l'acció per una d'aleatòria, amb probabilitat `MUTATION_RATE = 0.05`.
- **Elitisme:** El millor individu de cada generació passa directament a la següent.

4 Mètodes no utilitzats: Programació dinàmica

El problema del **Frozen Lake amb lliscament** (*slippery*) no és un problema adequat per poder utilitzar la **Programació Dinàmica (PD)**. La Programació Dinàmica requereix conèixer completament el **Model de l'Entorn** (un **MDP determinista**) per poder calcular la funció de valor òptima.

En canvi, l'entorn en aquest problema és **estocàstic** i **no determinista**. No podem saber ni la **probabilitat de transició entre estats** ni les **recompenses associades a cada transició** a priori.

5 Anàlisi i Comparació de Rendiments

L'anàlisi del rendiment dels diferents algoritmes es basa en dues fonts principals d'informació:

- La **corba d'aprenentatge**, que mostra la mitjana de recompenses dels darrers 100 episodis.
- La **política final** π^*, obtinguda a partir de la Q-Taula final (en RL) o del millor cromosoma (en l'Algorisme Genètic).

Aquesta anàlisi permet comparar la velocitat de convergència, l'estabilitat i l'eficiència de cada mètode en un entorn estocàstic com *Frozen Lake*, on el factor de lliscament dificulta l'aprenentatge òptim.

5.1 Aprenentatge per Reforç

A la Taula 1 es resumeixen les diferències qualitatives entre els tres mètodes de RL utilitzats: Monte Carlo, SARSA i Q-Learning.

Taula 1: Comparació qualitativa dels mètodes RL

Mètode	Política	Actualització	Estimació
Monte Carlo (MC)	On-Policy (ϵ-greedy)	Al final de l'episodi	Retorn (Baixa biaix, Alta variança
SARSA	On-Policy (ϵ-greedy)	A cada pas (TD(0))	$Q(s', a')$ (Baix biaix)
Q-Learning	Off-Policy (greedy)	A cada pas (TD(0))	$\max_a Q(s', a)$ (Més agressiu)

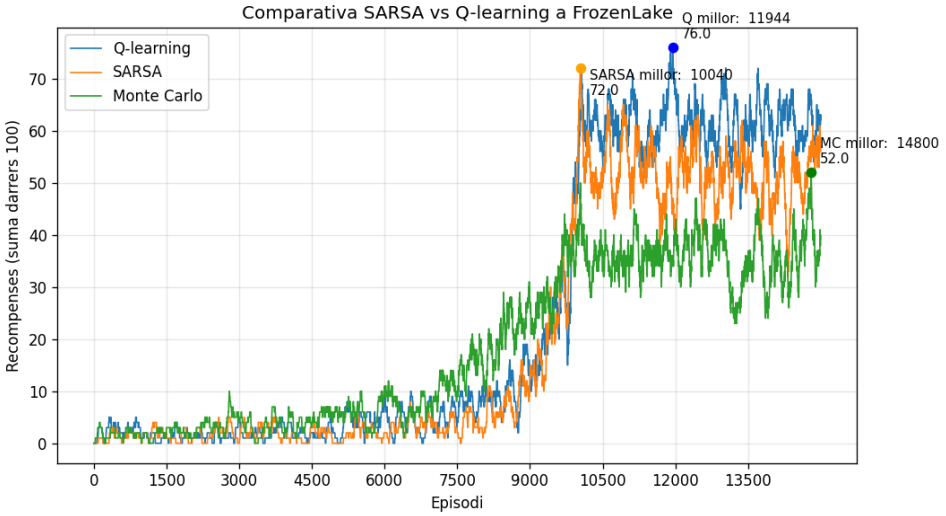

Figura 1: Corba d'aprenentatge de SARSA, Q-Learning i Monte Carlo (mitjana de recompenses sobre 100 episodis).

Convergència i Estabilitat

- **SARSA:** Com que és *on-policy*, té en compte els moviments no desitjats produïts pel lliscament. En conseqüència, tendeix a aprendre polítiques **més segures però menys òptimes**. La seva corba d'aprenentatge és més suau i estable.
- **Q-Learning:** És *off-policy* i per tant aprèn la política greedy encara que explori. En entorns estocàstics això provoca que intenti seguir camins "òptims" que realment no ho són si hi ha lliscament, cosa que pot retardar la convergència. Tot i així, acaba trobant polítiques amb **millors recompenses finals** que SARSA.
- **Monte Carlo:** Actualitza només al final de cada episodi i té alta variança. Per això mostra una convergència **més lenta i sorollosa**. Tot i ser simple d'implementar, no és el mètode més eficient per a Frozen Lake.

Política Final π^* Les polítiques obtingudes mostren diferències clares:

- **SARSA** tendeix a rodejar forats, prioritzant seguretat sobre velocitat.
- **Q-Learning** tendeix a triar camins més directes, però més arriscats.
- **Monte Carlo** depèn molt de la qualitat de les primeres exploracions, i pot quedar atrapat en polítiques subòptimes.

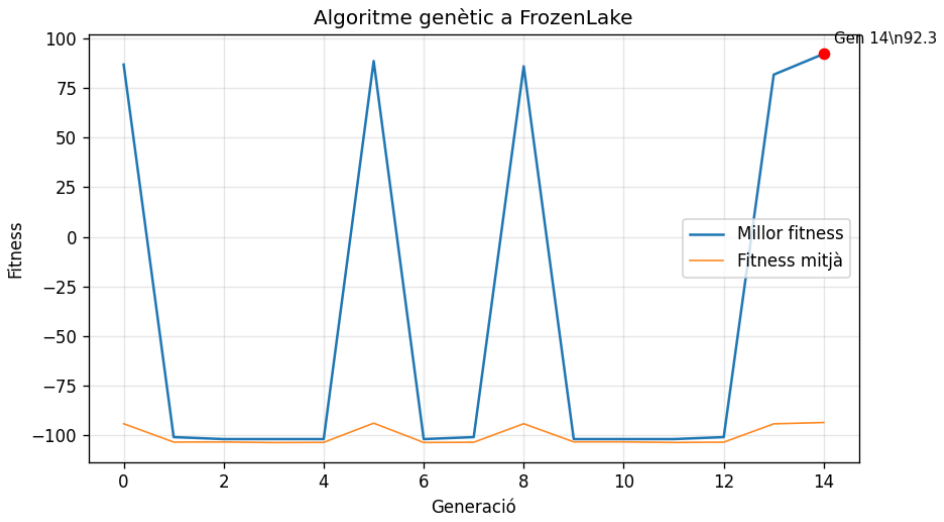

Figura 2: Evolució del fitness (millor i mitjà) de l'Algorisme Genètic.

5.2 Algorisme Genètic

L'Algorisme Genètic (AG) afronta el problema de manera totalment diferent als mètodes RL. En lloc d'aprendre una política $\pi(s)$, evoluciona una **seqüència fixa d'accions** que intenta portar l'agent del punt inicial al punt objectiu.

Limitacions en un entorn estocàstic Frozen Lake és un entorn on les accions no sempre es duen a terme tal com s'indiquen, a causa del paràmetre `is_slippery=True`. Això implica que:

- Un cromosoma que funciona bé en una avaluació pot fracassar en la següent.
- El **fitness** esdevé sorollós i de difícil interpretació.
- La convergència és més lenta i menys estable que amb RL.

Resultat global Tot i aquestes dificultats, l'AG pot trobar seqüències raonables si:

- s'aplica elitisme (mantenir el millor individu),
- es fa servir selecció per torneig,
- la funció de fitness incorpora informació sobre la distància al goal.

En general, però, els mètodes RL superen clarament l'AG en termes d'estabilitat i rendiment, ja que són capaços d'aprendre una política adaptativa, mentre que el GA es limita a una seqüència fixa que no pot reaccionar als lliscaments.

5.3 Conclusió Comparativa

En resum:

- **Q-Learning** presenta el millor rendiment final, coherent amb la seva naturalesa off-policy.

- **SARSA** mostra un aprenentatge més conservador però més estable.
- **Monte Carlo** és el més lent i amb més variança.
- **L'Algorisme Genètic** pot trobar solucions viables però no pot competir amb RL en consistència.

Aquest comportament és coherent amb la teoria del curs i amb les propietats de cada mètode.

6 Ús d'Eines d'Intel·ligència Artificial Generativa

Durant l'elaboració d'aquesta pràctica, s'han utilitzat eines d'Intel·ligència Artificial Generativa amb els següents propòsits:

- Ajuda a l'hora d'elaborar la memòria de **LaTeX** (format, no contingut), ajudant a crear una plantilla, a destacar **paraules importants** i a corregir **faltes d'ortografia i redacció**.
- Ajuda a l'hora de realitzar els **gràfics** a **Python**.
- Ajuda a l'hora de fer els **comentaris** per ajudar a entendre el codi als lectors.

7 Conclusions

En aquest treball hem aconseguit comparar **diferents algorismes** per resoldre el problema del *Frozen Lake* i hem arribat a les conclusions següents:

- **SARSA**, a causa de la seva política **On-Policy**, tendeix a aprendre polítiques **més segures** (evitant riscos) però **menys òptimes**.
- En canvi, el **Q-Learning** utilitza una política **Off-Policy**, la qual cosa fa que prengui **decisions més arriscades** però aconsegueixi **millors recompenses finals** (la política òptima).
- L'algorisme de **Montecarlo** depèn molt de la qualitat de les primeres exploracions i pot quedar atrapat en polítiques subòptimes, mostrant una convergència més **lenta** i **variable**.
- També hem observat que l'Algorisme Genètic (**AG**) ha afrontat dificultats a l'hora d'adaptar-se a l'entorn **estocàstic** . Per tant, podem concloure que els algorismes d'**Aprenentatge per Reforç (RL)** superen clarament l'AG en termes d'**estabilitat** i **rendiment** en aquest tipus de problemes.