Parte III

Reutilización de Políticas

Reutilización de Políticas en Aprendizaje por Refuerzo

Aprendizaje por Refuerzo

Máster en Ciencia y Tecnología Informática

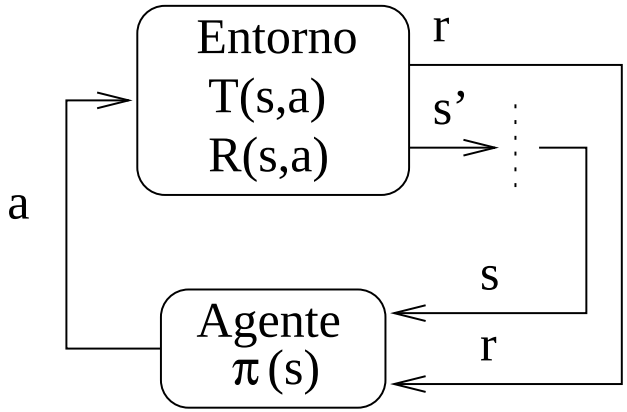
Fernando Fernández Rebollo

Grupo de Planificación y Aprendizaje (PLG)
Departamento de Informática
Escuela Politécnica Superior
Universidad Carlos III de Madrid

En Esta Sección:

Introducción

- Problema de Aprendizaje por Refuerzo (definido como un MDP):
  - Conjunto de todos los posibles estados, $\mathcal{S}$,
  - Conjunto de todas las posibles acciones, $\mathcal{A}$,
  - Función de transición de estados desconocida, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \Re$
  - Función de refuerzo desconocida, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \Re$.



Introducción

- Objetivo: aprender la política de acción $\Pi : \mathcal{S} \rightarrow \mathcal{A}$ que maximice el refuerzo medio esparado.
- Si asumimos tareas episódicas con estados meta absorbentes y posiciones iniciales aleatorias:
  - K: Número de episodios
  - H: Número de pasos por episodio

$$W = \frac{1}{K} \sum_{k=0}^{K} \sum_{h=0}^{H} \gamma^h r_{k,h}$$

Q-Learning (Watkins, 1989)

Q-Learning (γ, α).

Inicializar $Q(s, a)$, $\forall s \in \mathcal{S}, a \in \mathcal{A}$
Repetir (para cada episodio)
    Inicializa el estado inicial, s, aleatoriamente.
    Repetir (para cada paso del episodio)
        Selecciona una acción a y ejecútala
        Recibe el estado actual s', y el refuerzo, r
        $Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha[r + \gamma \text{máx}_{a'} Q(s', a')]$
        Asigna $s \leftarrow s'$
Devuelve $Q(s, a)$

Exploración vs. Explotación

- Explorar aleatoriamente
- Usar acciones derivadas de la función de valor que se ha aprendido hasta el momento (ϵ-greedy, softmax, etc.)
- Incluir sesgos adicionales en la exploración, como heurísticas, políticas aprendidas previamente, o sugerencias de un profesor o experto
- Inicializar la función Q con valores más o menos informados
- Combinar los métodos anteriores

Transferencia de Comportamiento Aprendido

- Consiste en cómo transferir conocimiento adquirido durante el aprendizaje de una tarea fuente a una nueva tarea objetivo
- Principales métodos:
  - Inicializar el aprendiz con el conocimiento: tuplas de experiencia, funciones de valor, políticas, modelos del agente/entorno, distribuciones a priori
  - Guiar la selección de acciones durante el aprendizaje: reglas o consejos, políticas parciales u opciones (*options*), características del proceso de aprendizaje, refuerzos intermedios, o definiciones (jerarquías) de sub-tareas

Aprendizaje por Demostración

- Se dispone de un entrenador/profesor que sabe resolver la tarea, no necesariamente de forma óptima
- El entrenador proporciona ejemplos de resolución de la tarea: pares estado-acción que son útiles para el proceso de aprendizaje
- Es una forma más de transferencia de conocimiento (entre distintos agentes)
- El conocimiento no tiene porqué haber sido aprendido

Transferencia de Conocimiento en Entornos Multi-agente

- Varios agentes en un mismo entorno resolviendo tareas
- Intercambio de conocimiento entre los distintos agentes
- El conocimiento puede haber sido aprendido o no

Reutilización Probabilística de Políticas (PPR)

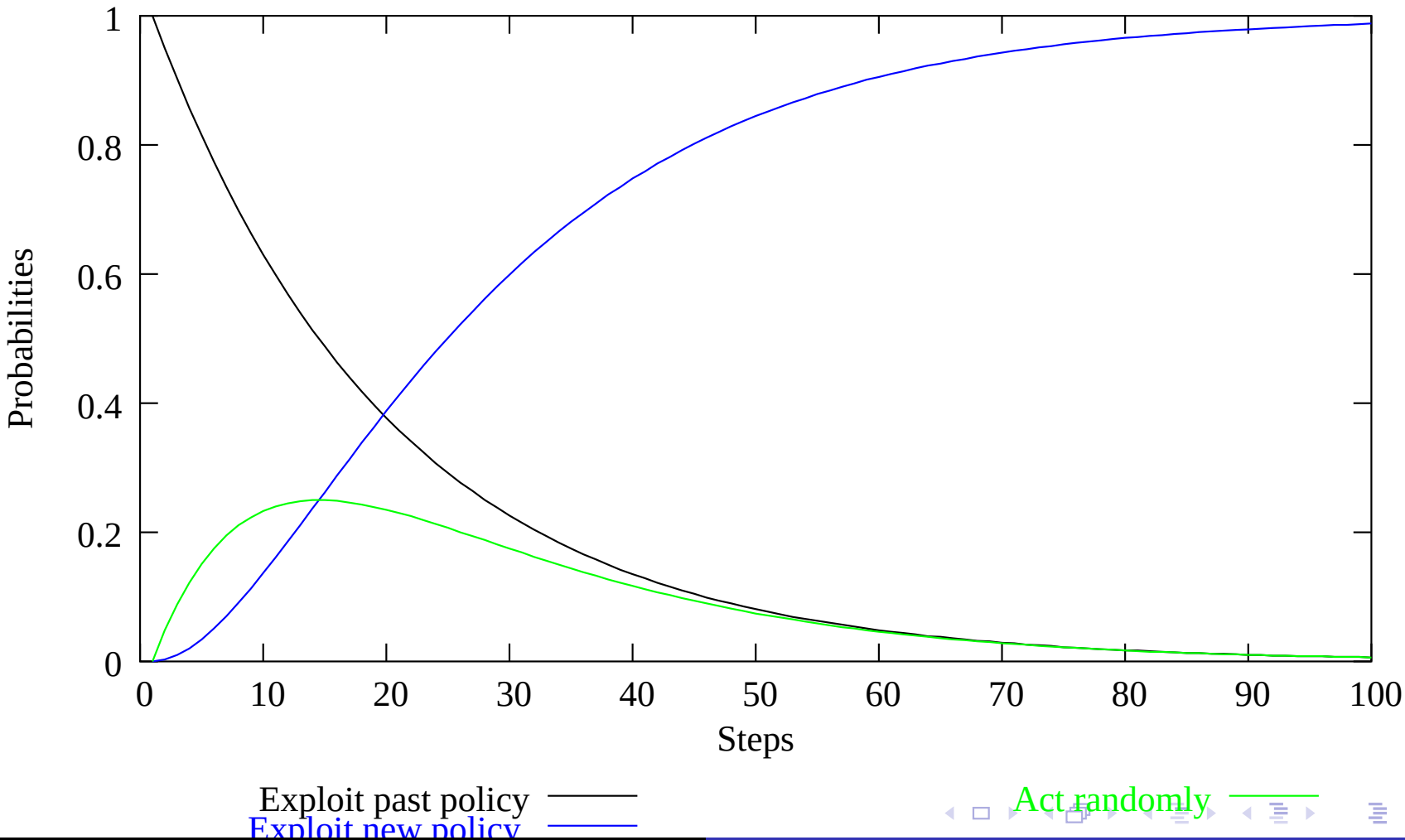
- Se asume la existencia de una política $\Pi_{past}(s)$ en el aprendizaje de una nueva, $\Pi_{new}(s)$
- Selecciona $a = \begin{cases} \Pi_{past}(s) & \text{con prob. } \psi \\ \Pi_{new}(s) & \text{con prob. } (1-\psi)\epsilon \\ rand(A) & \text{con prob. } (1-\psi)(1-\epsilon) \end{cases}$

Estrategia de Exploración π-reuse

π-reuse $(\Pi_{past}, K, H, \psi, \upsilon, \gamma, \alpha)$.

Inicializar $Q^{\Pi_{new}}(s,a) = 0$, $\forall s \in \mathcal{S}, a \in \mathcal{A}$
For $k = 1$ to K
    Seleccionar el estado inicial, s, aleatoriamente.
    Asigna $\psi_1 \leftarrow \psi$
    For $h = 1$ to H
        Con probabilidad ψ_h, $a = \Pi_{past}(s)$
        Con probabilidad $1 - \psi_h$, $a = \epsilon\text{-greedy}(\Pi_{new}(s))$
        Recibe el estado actual s', y el refuerzo, $r_{k,h}$
        Actualiza $Q^{\Pi_{new}}(s,a)$, y por tanto, Π_{new}, usando la función de actualización de Q-Le
            $Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha[r + \gamma \text{máx}_{a'} Q(s',a')]$
        Asigna $\psi_{h+1} \leftarrow \psi_h \upsilon$
        Asigna $s \leftarrow s'$
$W = \frac{1}{K}\sum_{k=0}^{K}\sum_{h=0}^{H}\gamma^h r_{k,h}$
Devuelve W, $Q^{\Pi_{new}}(s,a)$ y Π_{new}

Exploración y explotación



Human-Agent Transfer: HAT (Taylor, Benen and Chernova 2011)

- Aprendizaje por Demostración (LfD) con Aprendizaje por Refuerzo
- Tres pasos:
  - Demostración:
    - El agente realiza una tarea operado por un profesor humano (o controlador sub-óptimo)
    - El agente almacena los pares estado-acción visitados
  - Generalización de la política: usa los pares estado-acción para aprender reglas que resumen la política (usa JRip para aprender una lista de decisiones)
  - Aprendizaje independiente:
    - Aplica aprendizaje por refuerzo libre de modelo
    - Balance entre la explotación de las reglas transferidas con el aprendizaje de la política que mejora las reglas

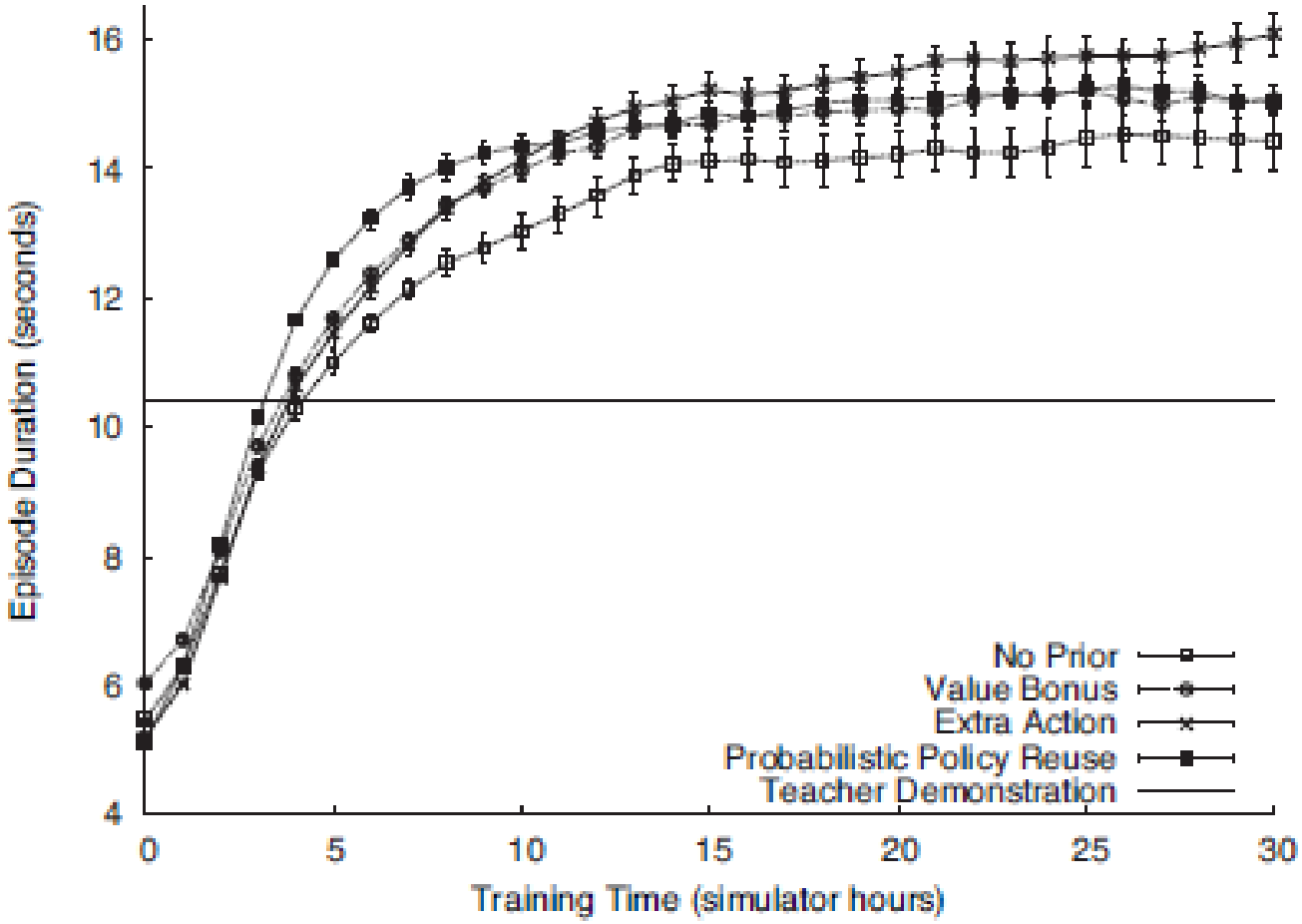
Reglas aprendidas en Keepaway

- if $(state_{11} \geq 74{,}84)$ and $(state_3 \leq 5{,}99)$ and $(state_{11} \leq 76{,}26)$ then $action = 1$
- elseif $(state_{11} \geq 53{,}97)$ and $(state_4 \leq 5{,}91)$ then $action = 2$
- elseif ...

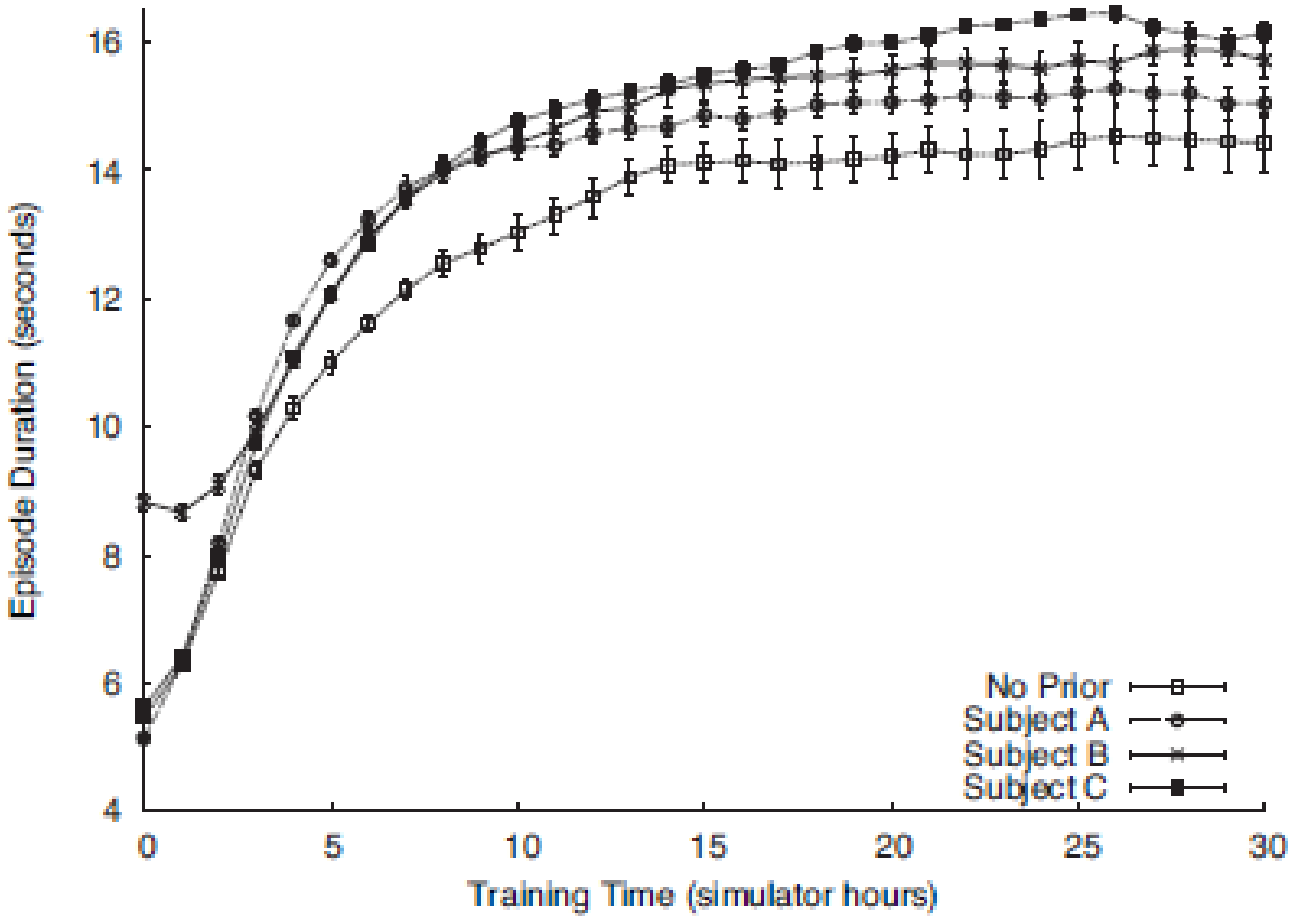
Uso de π-reuse en HAT

- Asume que $\Pi_{past}(s)$ es la política representada por el conjunto de reglas adquiridas a partir de las experiencias del profesor
- Asigna $a = \begin{cases} \Pi_{past}(s) & \text{w/prob. } \psi \\ \Pi_{new}(s)) & \text{w/prob. } (1-\psi)\epsilon \\ \mathit{Random} & \text{w/prob. } (1-\psi)(1-\epsilon) \end{cases}$

Resultados de HAT con π-reuse en Keepaway



Resultados de HAT con π-reuse en Keepaway



Teacher Confidence (Torrey and Taylor, 2012)

- El profesor puede tener más habilidad en algunas áreas del espacio de estados que en otras, porque quizá también aprendió mejor en unas áreas específicas.
  - Visits(s)
  - Update Counting, $c_t(s)$: número de veces que el profesor hizo una actualización no trivial de la función Q durante su aprendizaje
  - Update Counting, $c_s(s)$: número de veces que el estudiante ha hecho una actualización no trivial de la función Q durante el aprendizaje

Aplicación de *Teacher Confidence* a PPR

- Conditional PPR

$$p(s) = \begin{cases} 0 & \text{if } c_t(s) < 1 \\ \psi & \text{if } c_t(s) \geq 1 \end{cases} \quad (2)$$

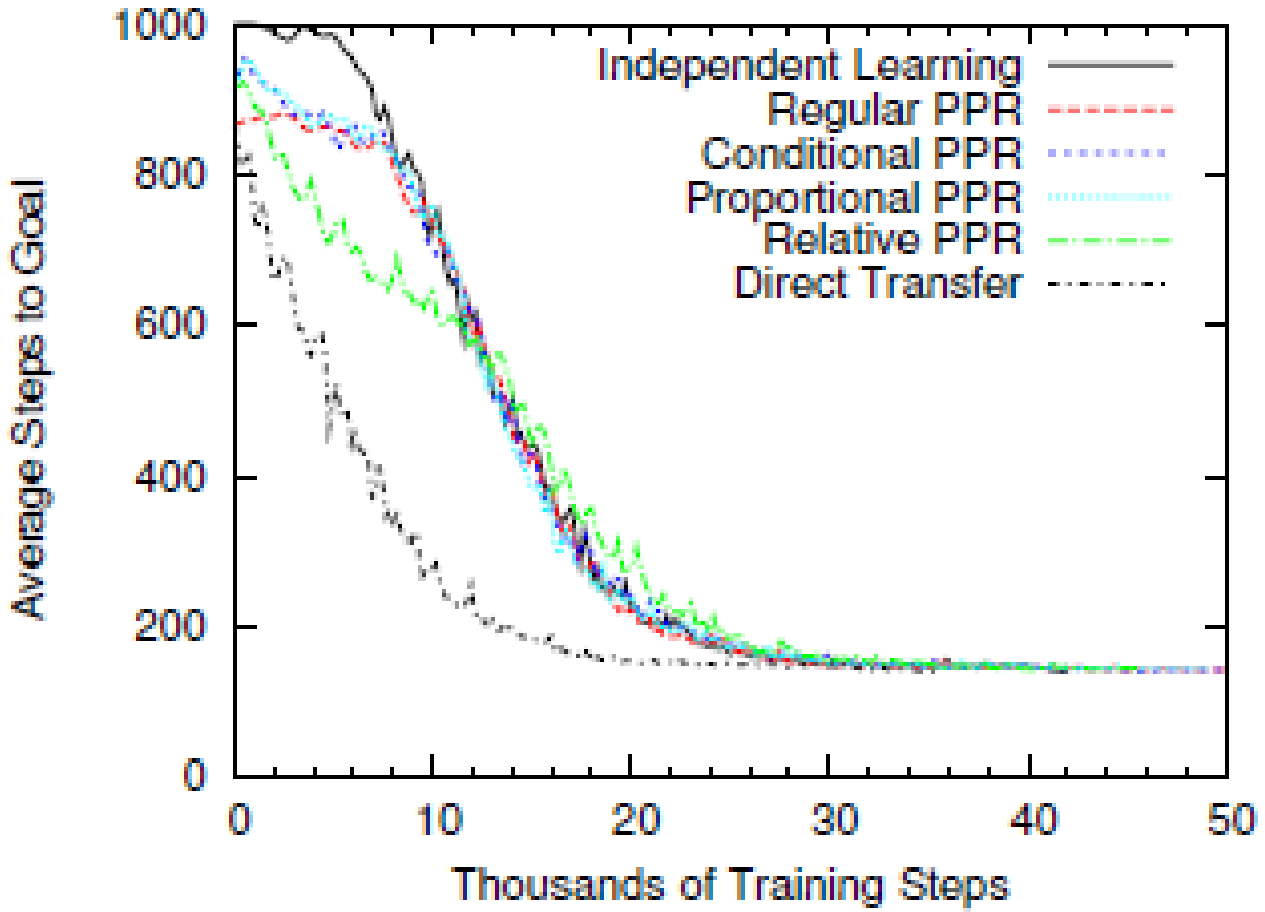
- Proportional PPR ($f \geq 0$):

$$p(s) = \begin{cases} 0 & \text{if } c_t(s) < 1 \\ \psi \frac{c_t(s)+d}{max(c_t)+f} & \text{if } c_t(s) \geq 1 \end{cases} \quad (3)$$

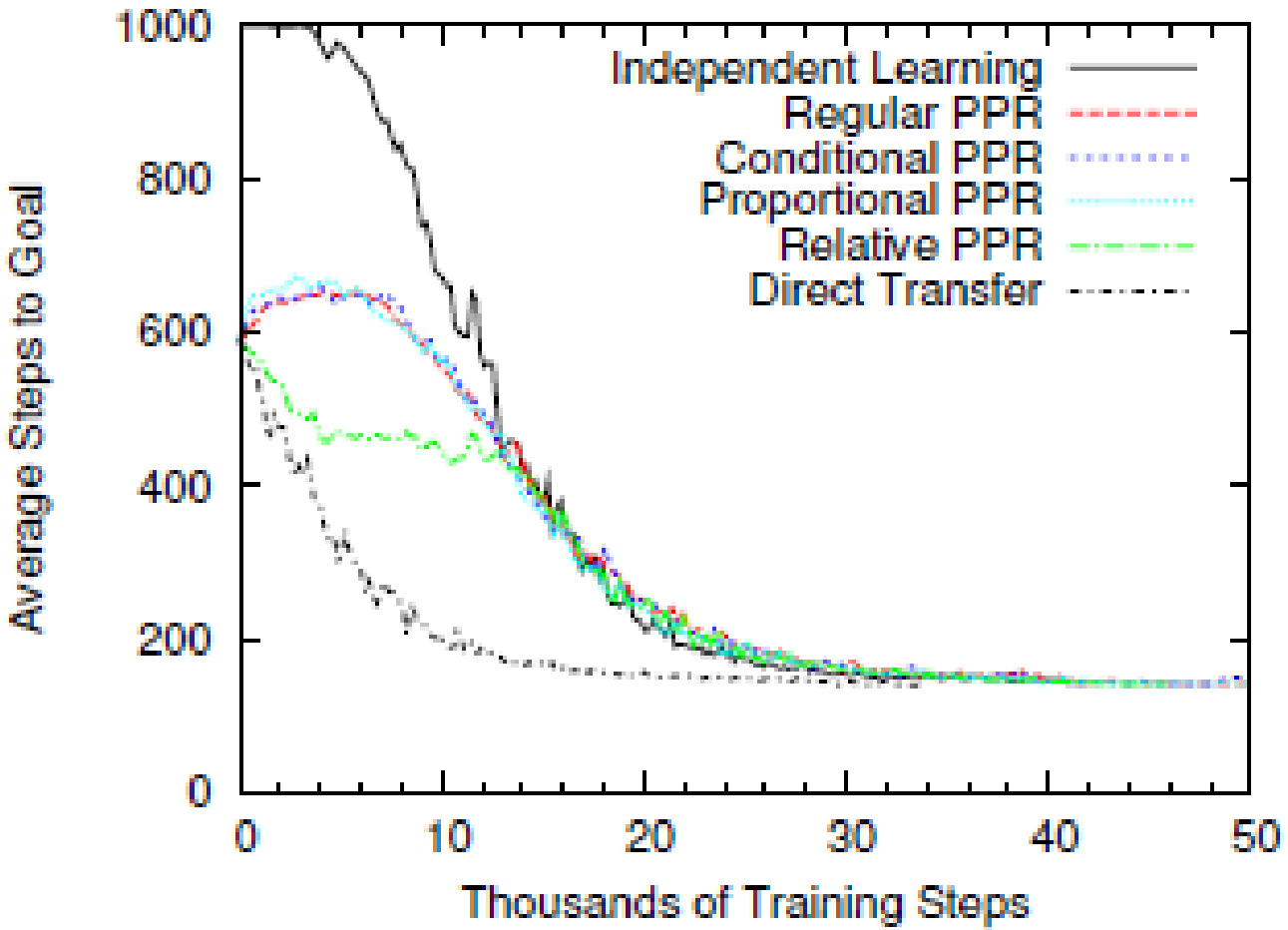
- Relative PPR:

$$p(s) = \begin{cases} 0 & \text{if } c_t(s) < 1 \\ min(1 - \frac{c_s(s)}{c_t(s)+d}, \psi) & \text{if } c_t(s) \geq 1 \end{cases} \quad (4)$$

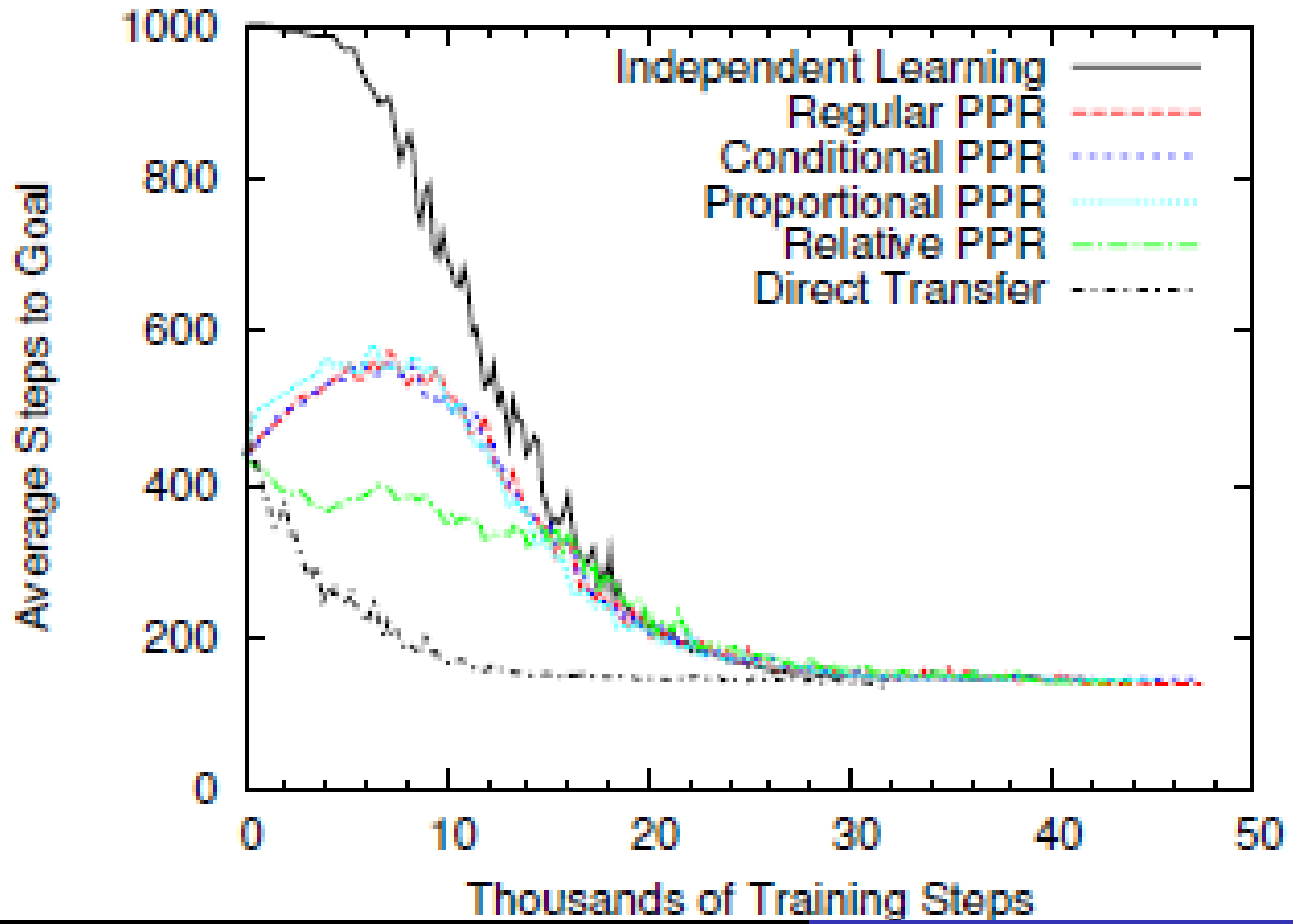
PPR basado en confianza en *Mountain Car* (el profesor aprendió durante 10 episodios)



PPR basado en confianza en *Mountain Car* (el profesor aprendió durante 20 episodios)



PPR basado en confianza en *Mountain Car* (el profesor aprendió durante 30 episodios)



Transferencia de Convenciones Sociales en Simulación de peatones

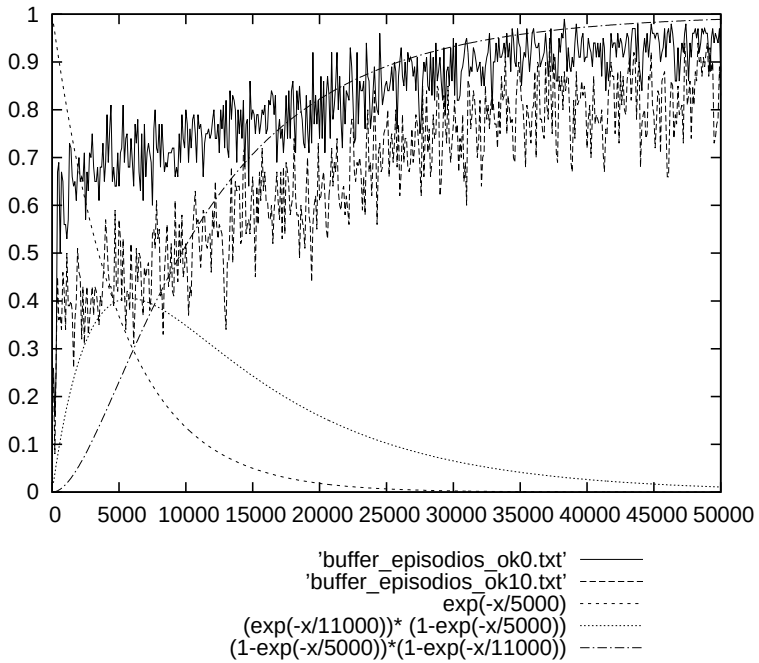
- Simulación de peatones como un problema de Aprendizaje por Refuerzo Multi-agente
- Convenciones sociales: en un pasillo, la gente camina por su derecha (izquierda)
- Comportamiento emergente: filas de peatones

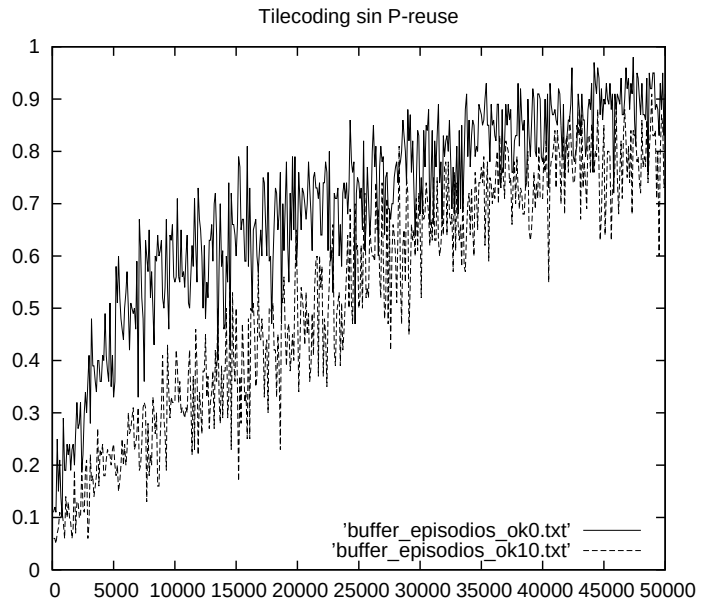
Inclusión de conocimiento social en PPR

- Estrategia de selección de acciones basada en π-reuse

$$a = \begin{cases} \mathit{adelante} - \mathit{derecha} & \text{con probabilidad } \psi \\ \epsilon - \mathit{greedy}(\Pi_{\mathit{new}}(s)) & \text{con probabilidad } (1-\psi) \end{cases} \quad (5)$$

Filas en un pasillo

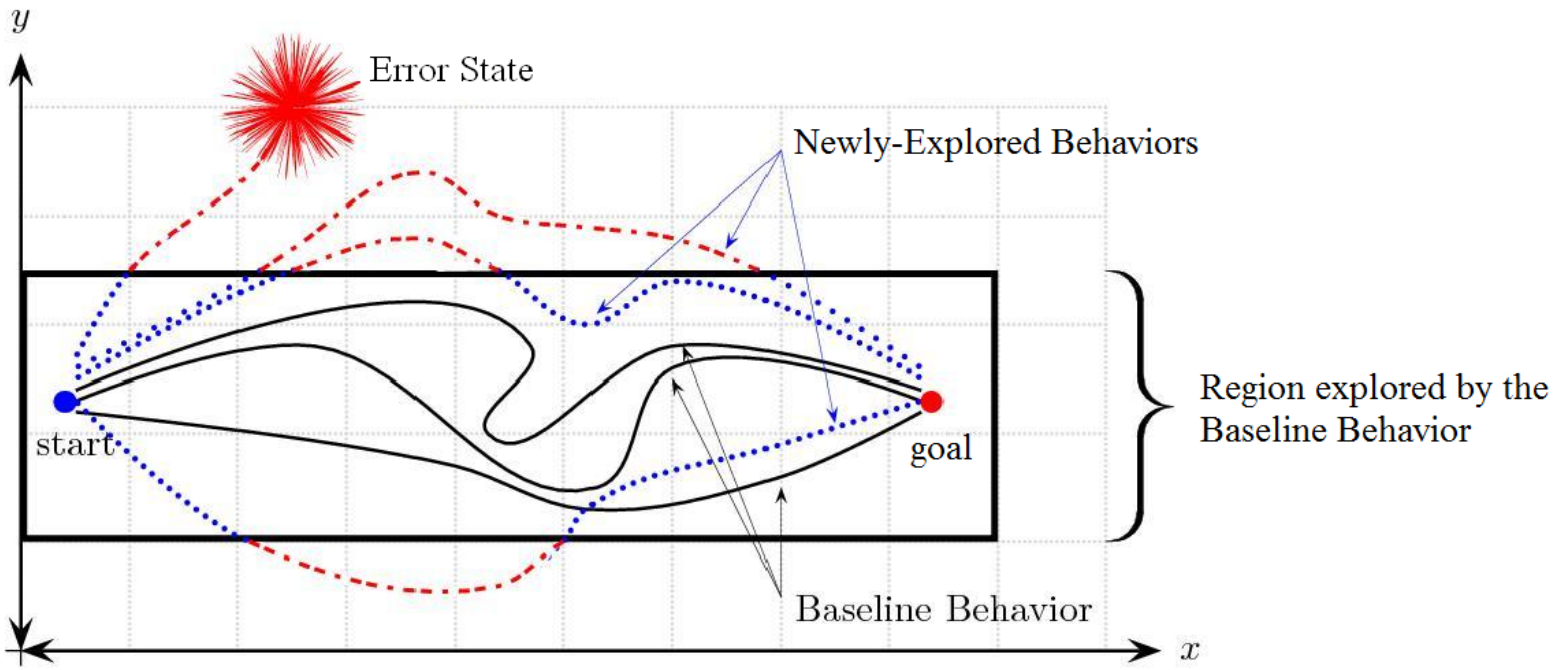




π-reuse

ϵ-greedy

Aprendizaje por Refuerzo Seguro



Estados Conocidos

- Dada una base de casos $B = \{c_1 \ldots, c_\eta\}$ compuesta de casos $c_i = (s_i, a_i, V(s_i))$, un estado s_q es considerado **conocido**, cuando $\min_{1 \leq i \leq \eta} d(s_q, s_i) \leq \theta$;
- Los casos en B describen una **Política Basada en Casos** de un agente, π_B^θ, y su función de valor asociada $V^{\pi_B^\theta}$.
- **(Comportamiento Base)**. La política π_T es considerado un comportamiento básico si:
  1. proporciona demostraciones seguras de la tarea que se desea aprender
  2. apoya el proceso de exploración, proporcionando acciones sub-óptimas e estados desconocidos para reducir la probabilidad de entrar en estados de error y devolver al agente a estados conocidos
  3. su comportamiento es seguro, pero lejos de óptimo

Funciones de Riesgo

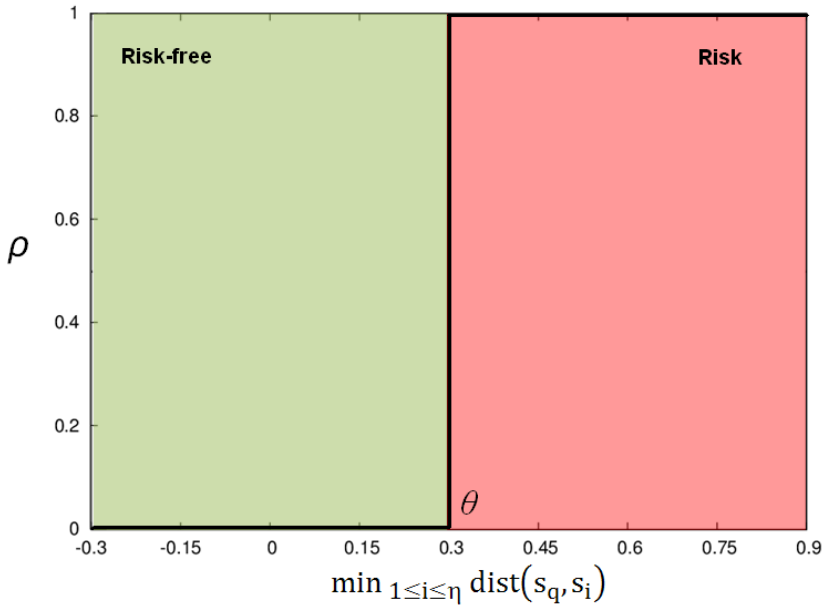
- **(Función de Riesgo Discreta Basada en Casos)** Dada una base de casos $B = \{c_1 \ldots, c_\eta\}$ compuesta de casos $c_i = (s_i, a_i, V(s_i))$, el riesgo de cada estado s se define como:

$$\varrho^{\pi_B^\theta}(s) = \begin{cases} 0 & \text{if } \text{mín}_{1 \leq j \leq \eta}\, d(s, s_j) < \theta \\ 1 & \text{en cualquier otro caso} \end{cases} \tag{6}$$

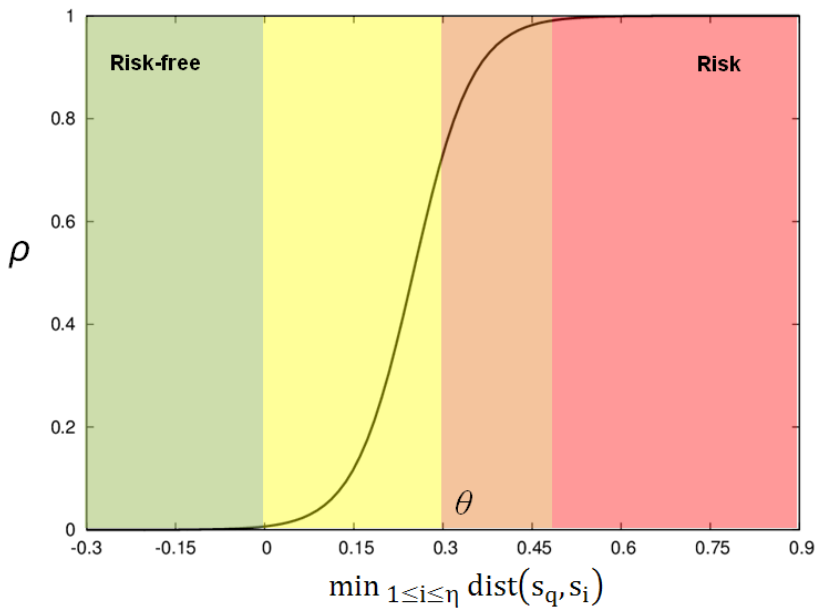
- **(Función de Riesgo Continua Basada en Casos)** Dada una base de casos $B = \{c_1 \ldots, c_\eta\}$ compuesta de casos $c_i = (s_i, a_i, V(s_i))$, el riesgo de cada estado s se define como:

$$\varrho^{B}(s) = 1 - \frac{1}{1 + e^{\frac{k}{\theta}((\text{mín}_{1 \leq j \leq \eta}\, d(s,s_j) - \frac{\theta}{k}) - \theta)}} \tag{7}$$

Funciones de riesgo continuas y discretas



(a)



(b)

Ratio de transferencia: explorar o seguir la sugerencia del tutor

- La función de riesgo sirve de ratio de transferencia: sustituye el parámetro ψ de π-reuse:
  - Con probabilidad $\varrho^B(s_h)$: $a_h = \pi_T(s_h)$
  - Con probabilidad $1 - \varrho^B(s_h)$: $a_h = rnd\_gaussian(\pi_B(s_h), \sigma)$

Safe π – *Reuse*

Safe π-reuse (π_T, H, B, σ)

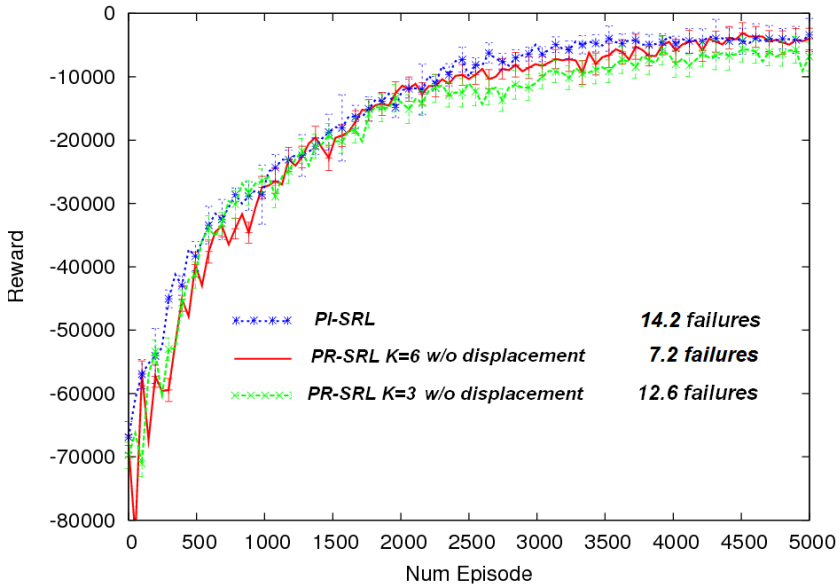
00 listCasesEpisode $\leftarrow \emptyset$.
01 $totalRwEpisode = 0$.
02 Set $h = 1$.
03 Set the initial state, s_h.
04 for h to H
05 Compute the case $< s, a, V(s) > \in B$ closest to the current state s_h
06 Set $\varrho^B(s_h) = 1 - \frac{1}{1+e^{\frac{k}{\theta}((\min_{1\leq j\leq \eta} d(s_h, s_j) - \frac{\theta}{k}) - \theta)}}$
07 With a probability of $\varrho^B(s_h)$: $a_h = \pi_T(s_h)$, $c^{new} := (s_h, a_h, 0)$
08 With a probability of $1 - \varrho^B(s_h)$:
$a_h = rnd\_gaussian(\pi_B(s_h), \sigma)$, $c^{new} := (s, a_h, V(s))$
09 Execute a_h and receive the next state s'_h, and reward, r_{s_h, a_h}
10 $totalRwEpisode := totalRwEpisode + r(s_h, a_h)$
11 $listCasesEpisode := listCasesEpisode \cup c^{new}$
12 Set $s_h \leftarrow s'_h$
13 Return $listCasesEpisode$, $totalRwEpisode$

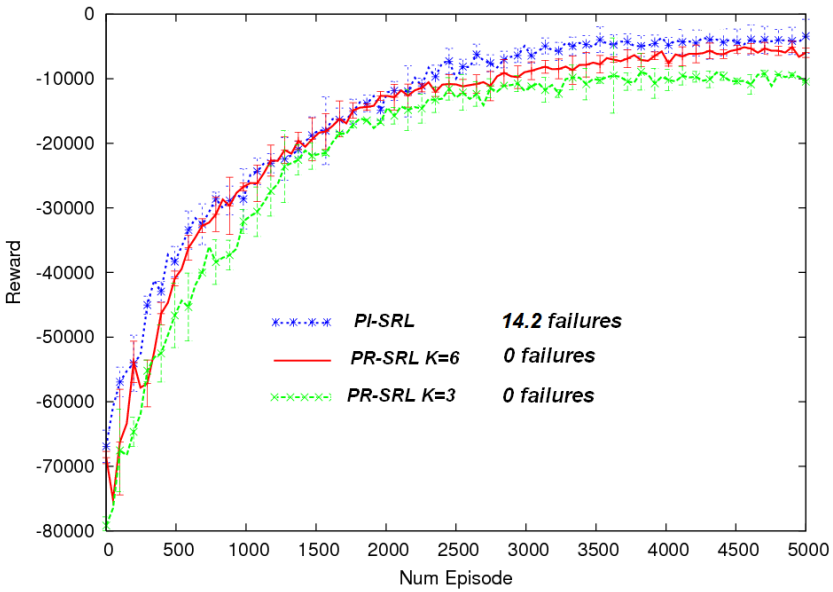
Policy Reuse for Safe Reinforcement Learning (PR-SRL)

PR-SRL

00 Given the case-base B, and the maximum number of cases η
01 Given the baseline behavior π_T
02 Given the update threshold Θ and the risk parameter σ
03 1. Set $maxTotalRwEpisode = 0$, the maximum cumulative reward reached in an episode
04 2. **Repeat**
05 (a) **Case generation**:
06 $listCasesEpisode, totalRwEpisode :=$Safe-$\pi$-reuse$(\pi_T, H, B)$
07 (b) **Computing the state-value function for the unknown states**
12 (c) **Updating the cases in B using the experience gathered**
24 **if** $\|B\| > \eta$ **then**
25 Remove the $\eta - \|B\|$ least-frequently-used cases in B
26 **until** stop criterion becomes true
27 3. **Return** B

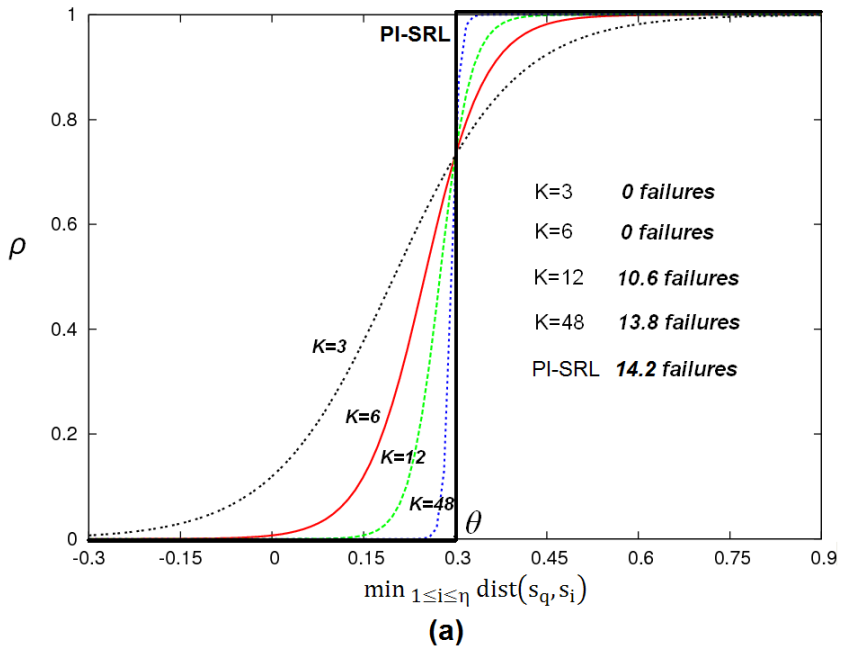
Resultados en el dominio del Helicóptero



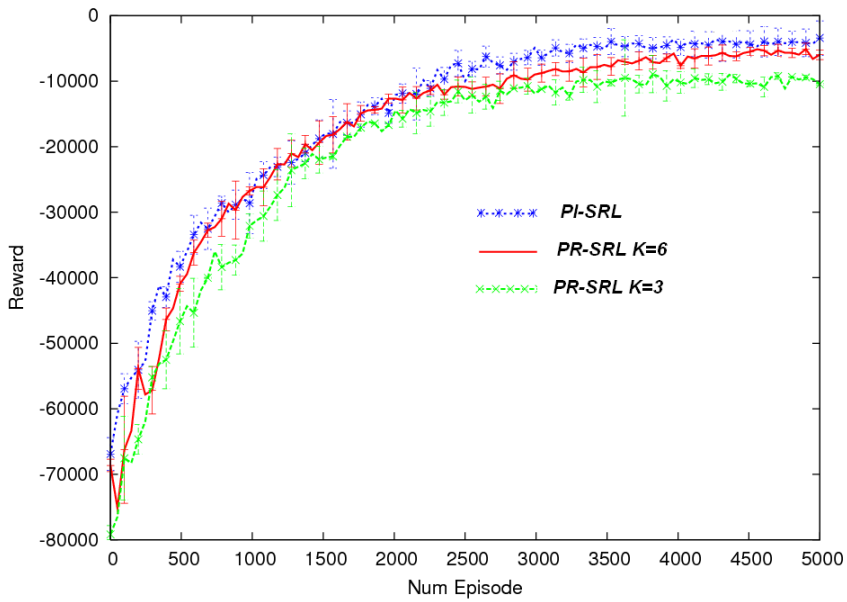


(b)

Resultados en el dominio del Helicóptero



(a)



(b)

Transferencia de Conocimiento Aprendido

- Dada una tarea objetivo, seleccionar una o varias tareas fuente apropiadas desde la que transferir conocimiento aprendido.
- Aprender la relación entre las tareas fuente y la objetivo
- Transferir el conocimiento desde las tareas fuente a la objetivo

Dominios y Tareas en PPR

- Un **dominio** $\mathcal{D}$ se define como una tupla $< \mathcal{S}, \mathcal{A}, \mathcal{T} >$, donde $\mathcal{S}$ es el conjunto de todos los posibles estados; $\mathcal{A}$ es el conjunto de todas las posibles acciones; y $\mathcal{T}$ es una función de transición de estados, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \Re$
- Una **tarea** Ω se define como una tupla $< \mathcal{D}, \mathcal{R}_\Omega >$, donde $\mathcal{D}$ es un dominio; y $\mathcal{R}_\Omega$ es la función de refuerzo, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \Re$
- Una **política de acción** Π_Ω que resuelve una tarea Ω es una función $\Pi_\Omega : \mathcal{S} \rightarrow \mathcal{A}$.

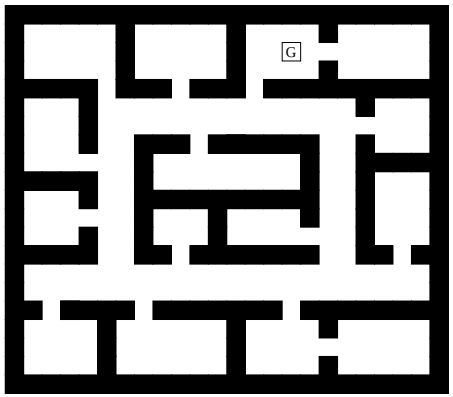
Reutilización Probabilística de un Conjunto de Políticas

- Tenemos que resolver la tarea Ω, es decir, aprender Π_Ω
- Hemos resuelto con anterioridad el conjunto de tareas $\{\Omega_1, \ldots, \Omega_n\}$ por lo que tenemos una librería de políticas compuesta de n políticas que las resuelven, es decir, $L = \{\Pi_1, \ldots, \Pi_n\}$
- Reto: ¿cómo podemos usar la librería de políticas, L, para aprender la nueva política, Π_Ω?
- Solución: reutilizar la política "más parecida"

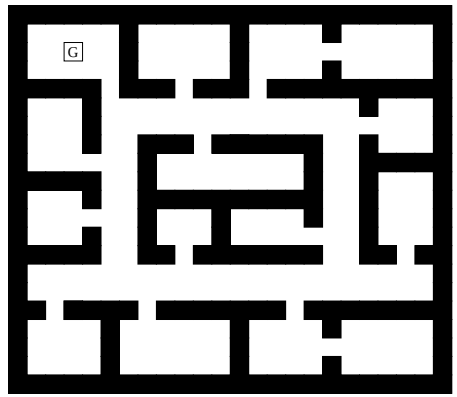
Ganancia de Reutilización

- Dados:
  - una política Π_i que resuelve la tarea $\Omega_i =< \mathcal{D}, R_i >$
  - una nueva tarea $\Omega =< \mathcal{D}, R_\Omega >$;
    la **ganancia de reutilización** de una política Π_i en la tarea Ω, digamos W_i, es la ganancia obtenida cuando se aplica la estrategia de exploración π-reuse con la política Π_i para aprender la política Π.
- La ganancia de reutilización es una métrica de similitud entre políticas que puede tener diversos usos

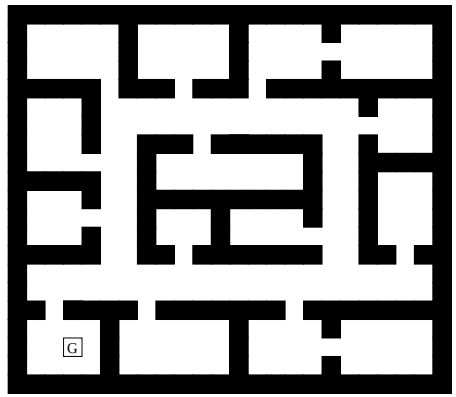
Dominio de oficinas



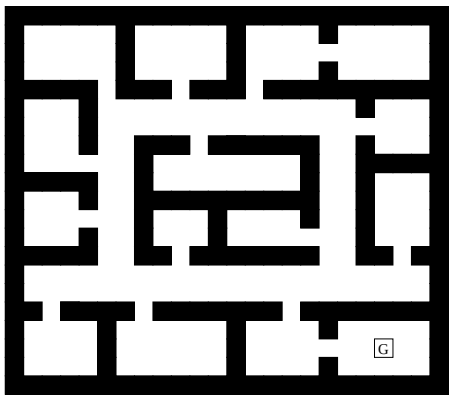
(a) Task Ω_1



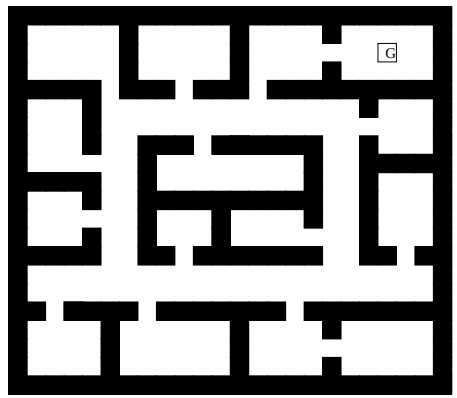
(b) Task Ω_2



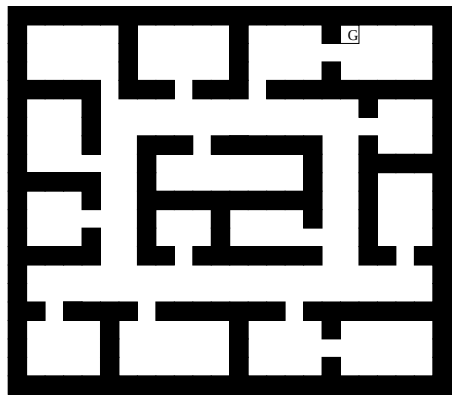
c) Task Ω_3



(d) Task Ω_4

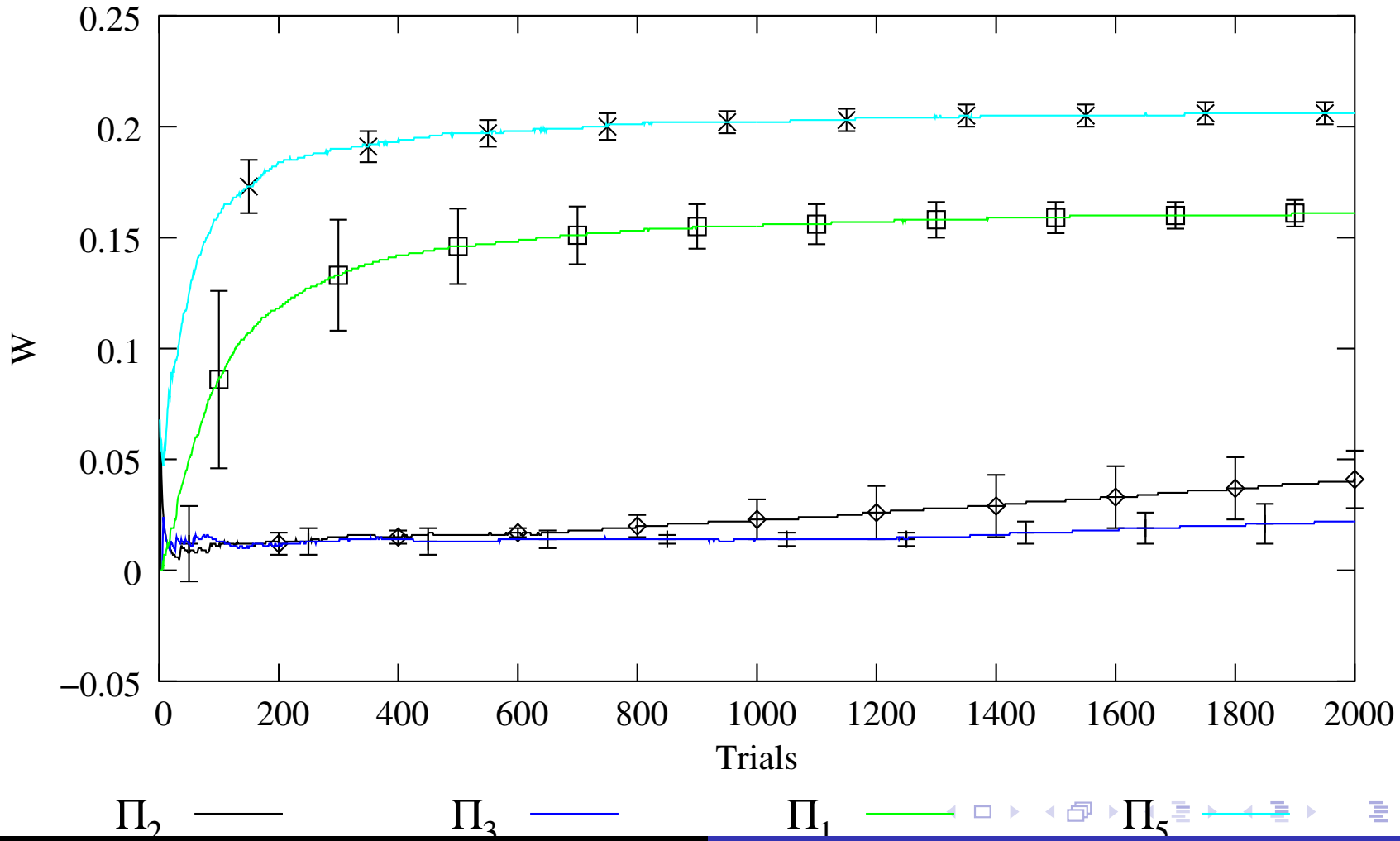


(e) Task Ω_5



(f) Task Ω

Cálculo de la Ganancia de Reutilización



La Ganancia de Reutilización como Ratio de Transferencia

1. Dado el conjunto de políticas $L \cup \{\Pi_\Omega\} = \{\Pi_\Omega, \Pi_1, \ldots, \Pi_n\}$, qué política debe ser seguida en cada episodio?
   - $P(\Pi_j) = \frac{e^{\tau W_j}}{\sum_{p=0}^{n} e^{\tau W_p}}$
2. Una vez que una política es seleccionada, Π_k qué estrategia de exploración se debe seguir?
   - Depende de la política elegida:
     - Si $\Pi_k \neq \Pi_\Omega$, entonces $\pi - reuse$
     - Si $\Pi_k = \Pi_\Omega$, entonces avariciosa.
3. Cómo se calcula W_j?
   - En línea, durante el aprendizaje de la nueva política

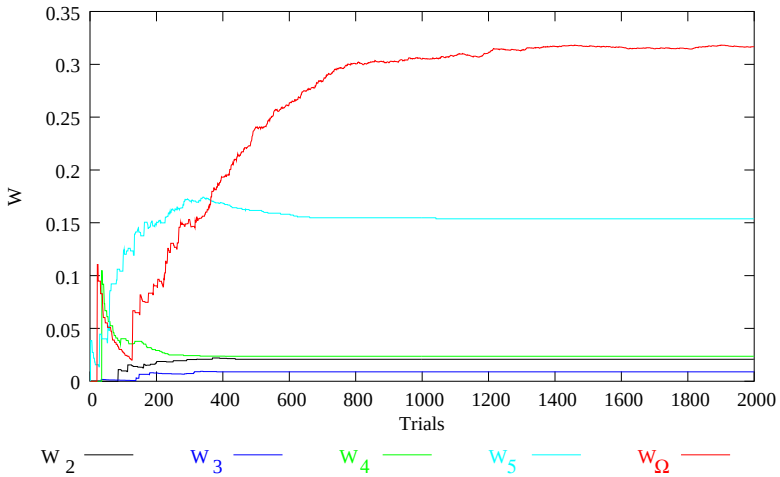
PRQ-Learning Algorithm

PRQL$(\Omega, L, \tau, \Delta\tau, K, H, \psi, \upsilon, \gamma, \alpha)$

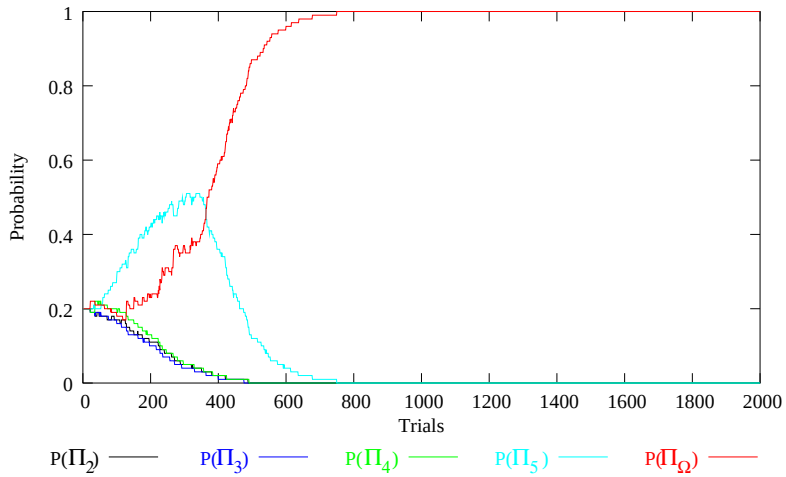
- Given:
  1. A new task Ω we want to solve
  2. A Policy Library $L = \{\Pi_1, \ldots, \Pi_n\}$
  3. An initial value of the temperature parameter, τ, and an incremental size, $\Delta\tau$, for the Boltzmann policy selection strategy
  4. A maximum number of episodes to execute, K
  5. A maximum number of steps per episode, H
  6. The parameters ψ and υ for the π-exploration strategy
  7. The parameters γ and α for the Q-learning update equation
  8. A ϵ parameter, for ϵ-greedy action selection strategy in a policy
- Initialize:
  1. $Q_\Omega(s, a) = 0, \forall s \in \mathcal{S}, a \in \mathcal{A}$
  2. Initialize W_Ω to 0
  3. Initialize W_i to 0
  4. Initialize the number of episodes where policy Π_Ω has been chosen, $U_\Omega = 0$
  5. Initialize the number of episodes where policy Π_i has been chosen, $U_i = 0, \forall i = 1, \ldots, n$
- For $k = 1$ to K do
  1. Choose an action policy, Π_k: $P(\Pi_j) = \frac{e^{\tau W_j}}{\sum_{p=0}^{n} e^{\tau W_p}}$ where W_0 is set to W_Ω
  2. Execute the learning episode k
     - If $\Pi_k = \Pi_\Omega$, execute a Q-Learning episode following a fully greedy strategy
     - Otherwise, use the π-reuse exploration strategy to reuse Π_k, i.e. call π-reuse$(\Pi_k, 1, H, \psi, \upsilon)$
     - In any case, receive the reward obtained in that episode, say R, and the updated Q function, $Q_\Omega(s, a)$
  3. Set $W_k = \frac{W_k U_k + R}{U_k + 1}$
  4. Set $U_k = U_k + 1$
  5. Set $\tau = \tau + \Delta\tau$
- Return the policy derived from $Q_\Omega(s, a)$

Evolución del ratio de reutilización en el dominio de oficinas

- PRQ-Learning reutilizando $L_3 = \{\Pi_2, \Pi_3, \Pi_4, \Pi_5\}$

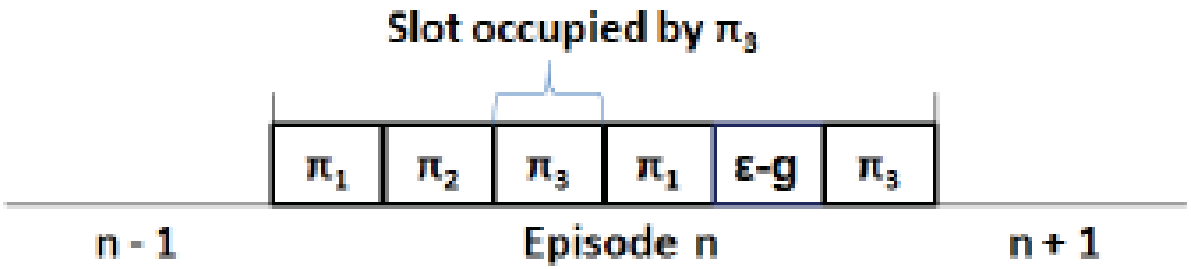


(a)Evolución de W_i



(b) Evolución de $P(\Pi_i)$

Spatial Hints: Episodes are Divided in Slots

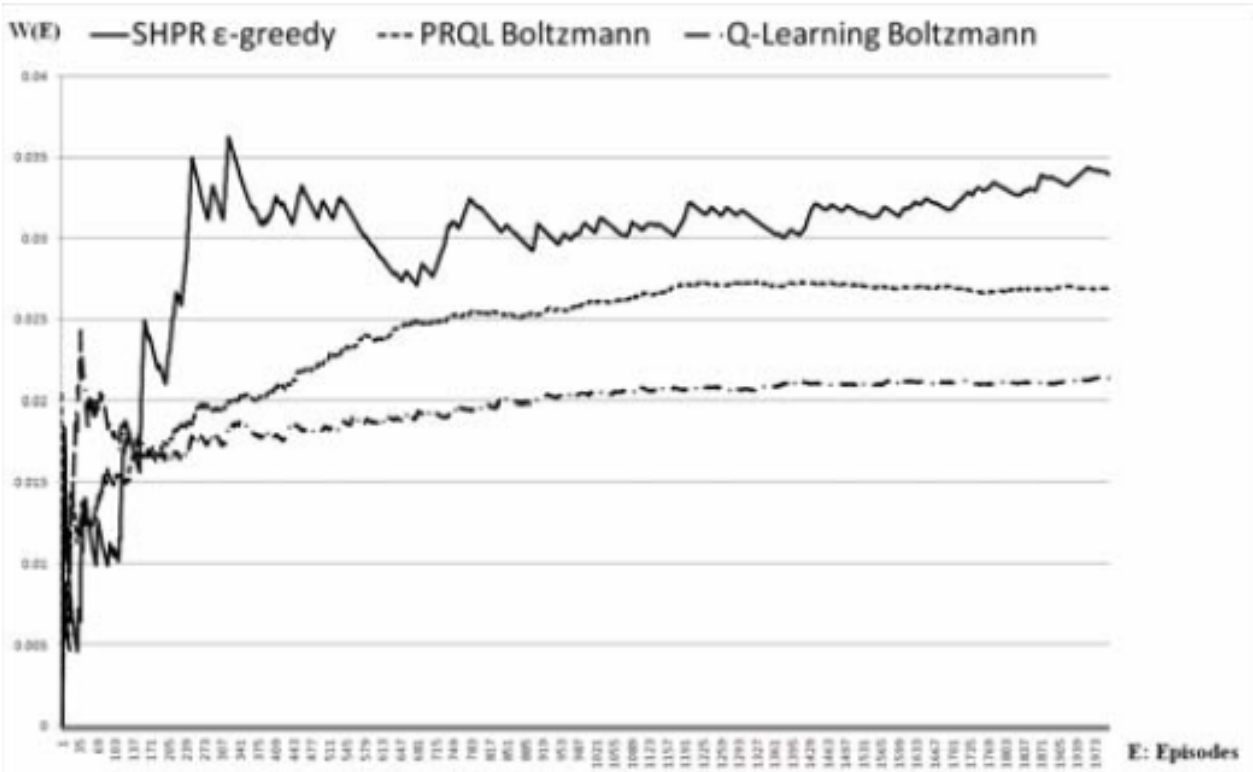


Spatial Hints

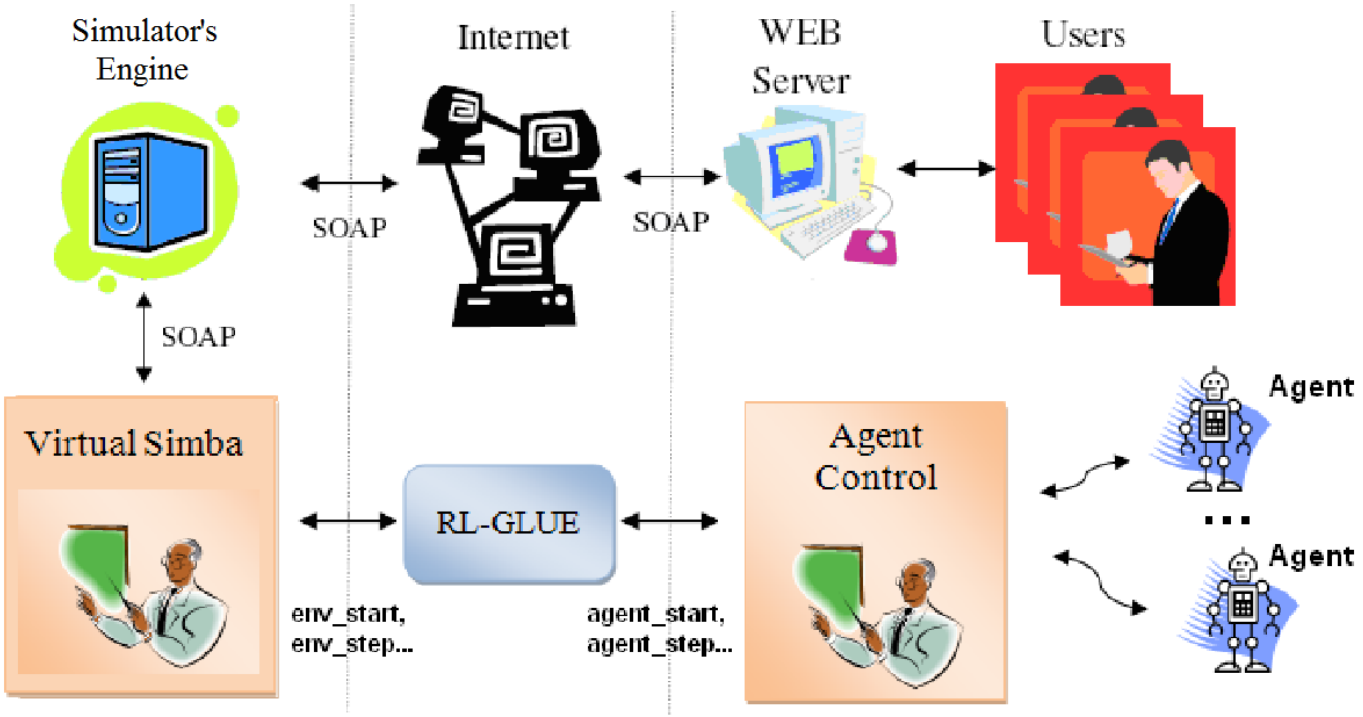
- **(Hint)**. A Hint h is a pair $< \pi_h, s_h >$, where π_h is a policy defined in the domain D and s_h is an state in the state space S of D.
- **(Hint Library)**. A Hint library is a set of hints, $\{h_1 \ldots, h_n\}$, where all the hints are defined in the same domain.
- $reach_i$ estimates how good the policy π_i is around its reference state s_i.
- The reuse probability of the policy π_i of a hint, h_i, in a current state, s is proportional to the reach value of such policy, but inversaly proportional to the distance to the reference state of such policy (s_i):

$$w_i = \frac{reach_i}{dist(s, s_i)} \tag{8}$$

Results in the Maze



SIMBA: *Simulator for Business Administration*



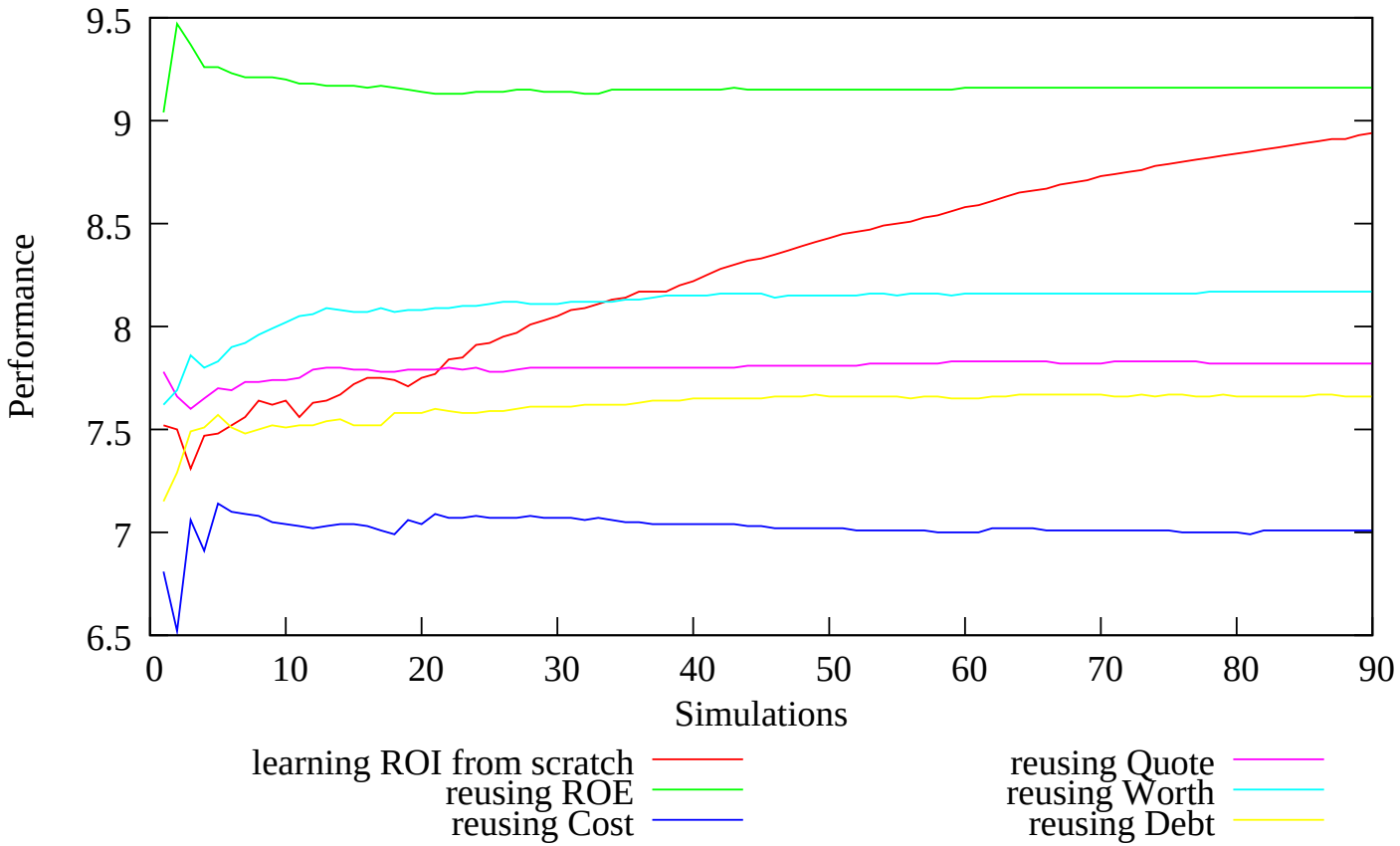
Espacio de estados y acciones en SIMBA

FEATURES of the State Space	**FEATURES of the Action Space**
Account value	Selling price
Human resources	Advertising expenses
Material cost	Network sales budget
Operating margin	Commercial information
Financial expenses	Training budget
Pre-tax income	Production scheduled
Tax	Material order
Training expenses	Research and Development budget
Bank overdraft	Loan
Economic productivity	Term loan
Advertising prediction	
Effort sales network	

Objetivos Distintos

- ROI (Retorno de la Inversión)
- ROE (Rentabilidad Financiera)
- Cuota de Mercado
- Valor
- Deuda
- Resultado del Ejercicio
- Etc.

Ganancia de Reutilización en SIMBA



Posibles Aplicaciones de PPR en SIMBA

- Similitud entre metas: la ganancia de reutilización es una medida de similitud entre dos metas:
  - ROI es muy similar al ROE
  - El coste del producto es muy distinto al ROI
- Optimización multi-criterio: podría ser calculada una frontera de Pareto mediante PPR?

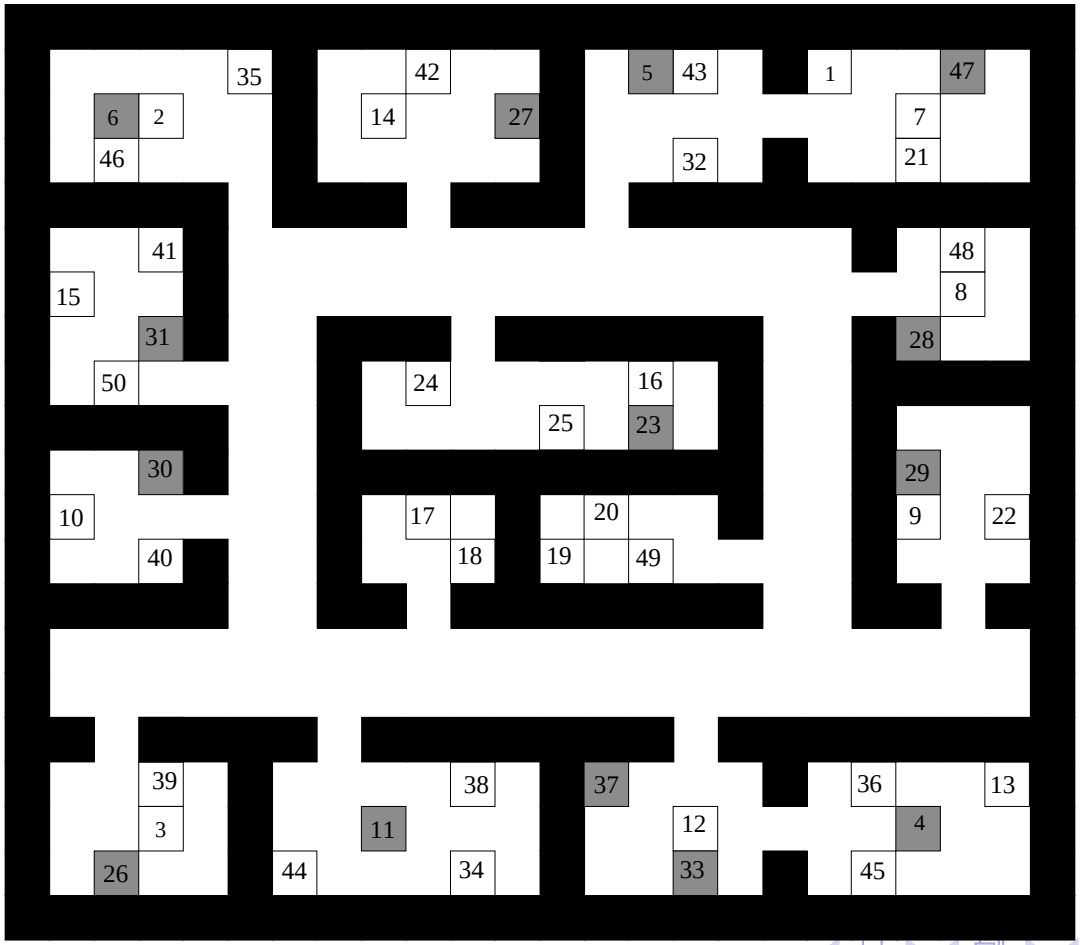
Algoritmo PLPR

PLPR Algorithm

- Given:
  1. A Policy Library, L, composed of n policies, $\{\Pi_1, \ldots, \Pi_n\}$
  2. A new task Ω we want to solve
  3. A δ parameter
- Execute the PRQ-Learning algorithm, using L as the set of past policies. Receive from this execution Π_Ω, W_Ω and W_{max}, where:
  - Π_Ω is the learned policy
  - W_Ω is the average gain obtained when the policy Π_Ω was followed
  - $W_{max} = \max W_i$, for $i = 1, \ldots, n$
- Update PL using the following equation:

$$L = \begin{cases} L \cup \{\Pi_\Omega\} & \text{if } W_{max} < \delta W_\Omega \\ L & \text{otherwise} \end{cases} \tag{9}$$

Core-Policies Obtained ($\delta = 0{,}25$)



Agrupación de Procesos de Decisión de Markov para Transferencia Continua (Ramamoorthy et al. 2013)

- Un algoritmo de reutilización de políticas que reutiliza de forma óptima un conjunto dado de políticas fuente cuando se resuelve un MDP concreto
- Una plataforma de agrupación que ha resuelto previmante un conjunto de MDPs fuente, de forma que se optimiza el rendimiento de reutilización de los algoritmos de transferencia.
    - una clase de funciones de distancia entre MDPs que permite definir los grupos de MDPs
    - una función de coste que mide cómo de bueno es un grupo particular para generar tareas fuente para el algoritmo de transferencia
    - algoritmo de optimización para calcular el grupo óptimo a transferir

Reconfiguración Multi-agente (Chen et al. 2012)

- Ajustar dinámicamente la formación y el tamaño de un equipo de robots que deben cubrir de manera distribuida una determinada área
- Formulación del problema de determinar la formación del equipo de robots como un juego de coalición determinado "juego de voto ponderado" (weighted voting game (WVG))
- El tamaño del equipo de robots se adapta dinámicamente mediante el ajuste de un parámetro de cuota del WVG
- Aplica el algoritmo Q-learning para aprender el valor de los parámetros de cuota
- Mecanismo de reutilización de políticas para adaptar el proceso de aprendizaje a cambios en el dominio

Aprendizaje en MDPs no estacionarios como Transferencia del Aprendizaje (Mahmud and Ramamoorthy 2013)

- La política de comportamiento de los agentes viene determinado por una variable latente que cambia muy esporádicamente, pero que puede modificar el comportamiento de los agentes de forma drástica cuando lo hace.
- Este cambio impredecible en la variable latente produce la no-estacionariedad
- Cada tarea/MDP requiere de un sitema de aprendizaje que interactúe con distintos oponentes con comportamientos fijos
- Entre distintas tareas, los espacios de estados y de acciones se mantiene y es conocido, pero las políticas de los agentes cambian.
- Se transfieren políticas desde tareas aprendidas en el pasado para inferir de forma rápida la política de comportamiento del

Resumen

- Diferencias entre Programación Dinámica, métodos libres de modelo y basados en el modelo
- Planificación en Procesos de Decisión de Markov
- Aprendizaje por Refuerzo:
  - Algoritmo Q-Learning
  - Importancia de la representación de los estados, las acciones y las funciones de valor
  - Discretizaciones incorrectas pueden romper la propiedad de Markov
  - Divergencia de los métodos de aproximación de las funciones de valor

Bibliografía

- Machine Learning, Tom Mitchell. Capítulo 13.
- Reinforcement Learning: An Introduction. Richard Sutton y Andrew Barto. MIT Press. 1998
- Reinforcement Learning Repository: http://www-anw.cs.umass.edu/rlr/
- Aprendizaje Automático: conceptos básicos y avanzados. Basilio Sierra Araujo. Pearson Prentice Hall. 2006
- Reinforcement Learning: a Survey. Lelie Pack Kaelbling and Michael L. Littman and Andrew W. Moore. International Journal of Artificial Intelligence Research 4, 1996, pp 237-285
- Martijn van Otterlo. The Logic of Adaptive Behavior - Knowledge Representation and Algorithms for Adaptive Sequential Decision Making under Uncertainty in First-Order and Relational Domains. Ios Press 2009