**4 - Búsqueda informada y exploración**

**4.5 – Agentes de búsqueda online y ambientes desconocidos**

* Los **agentes de búsqueda offline** calculan la solución completa antes de poner un pie en el mundo real y luego ejecutan la solución sin recurrir a sus percepciones
* Un agente de **búsqueda online** funciona **intercambiando** cálculo y la acción: primero toma una acción, luego observa el entorno y calcula la siguiente acción
  + Es buena idea para dominios dinámicos o semidinámicos. Es una idea mejor para dominios estocásticos.
* La búsqueda online es una buena idea en dominios dinámicos y semidinámicos (donde hay una penalización por holgazanear y utilizar demasiado tiempo para calcular)
* La búsqueda online es aun mejor para dominios estocásticos
* Una búsqueda offline debería presentar un plan de contingencia exponencialmente grande que considere todos los acontecimientos posibles
* Una búsqueda online necesita solo considerar lo que realmente pasa
* La búsqueda online es una idea necesaria para un **problema de exploración**
  + En un problema de este tipo los estados y acciones son desconocidos por el agente; un agente en este estado debe usar sus acciones como experimentos para determinar qué hacer después y a partir de ahí debe intercalar cálculo y la acción

**Problemas de búsqueda online**

* Un problema de búsqueda online puede resolverse solamente por un agente que ejecute acciones, más que por un proceso puramente computacional
* Asumimos que el agente sabe lo siguiente:
  + ACCIONES (s), devuelve una lista de acciones permitidas en el estado s
  + Funciones de costo individual (c, a, s’) //no puede usarse hasta que el agente sepa que s’ es el resultado
  + TEST-OBJETIVO(s)
* El agente no puede tener acceso a los sucesores de un estado excepto si intenta realmente todas las acciones en ese estado
* Asumiremos que el agente puede reconocer siempre un estado que ha visitado anteriormente y que las acciones son deterministas
* El agente podría tener acceso a una función heurística admisible h(s) que estime la distancia del estado actual a un estado objetivo
* El costo es el costo total del camino por el que el agente viaja realmente, se compara con el costo del camino que el agente seguiría si supiera el espacio de búsqueda de antemano. Esta comparación se hace con **proporción competitiva** y queremos que sea tan pequeña como es posible
* Ningún algoritmo puede evitar callejones sin salida en todos los espacios de estados
* El **argumento del adversario** es que podemos imaginar un adversario que construye el espacio de estados, mientras el agente lo explora
  + Puede poner el objetivo y callejos sin salida donde le guste
* Los callejones sin salida son una dificultad para la exploración de un robot, para avanzar asumiremos que el espacio de estados es **seguramente explorable**
  + Algún estado objetivo es alcanzable desde cualquier estado alcanzable
* Incluso en entornos seguramente explorables no se puede garantizar ninguna proporción competitiva acotada si hay caminos de costo ilimitado

**Agentes de búsqueda online**

* Después de cada acción, un agente online recibe una percepción al decirle que estado ha alcanzado, de esta información puede aumentar su mapa del entorno
* El mapa actual se usa para decidir dónde ir después
* Un algoritmo online puede expandir solo el nodo que ocupa físicamente
* Para evitar viajar a través de todo el árbol para expandir el siguiente nodo, parece mejor expandir los nodos en un orden local, esta propiedad la tiene la búsqueda primero por profundidad

**Búsqueda local online**

* La **búsqueda de ascensión de colinas** tiene la propiedad de localidad en sus expansiones de nodos
  + No es muy útil en forma simple porque deja que el agente se sitúe en máximos locales con ningún movimiento que hacer
  + Los reinicios aleatorios no puede utilizarse, porque el agente no puede moverse a un nuevo estado
  + En vez de reinicios, podemos considerar el uso de un **camino aleatorio**
    - El camino aleatorio selecciona simplemente al azar una de las acciones disponibles del estado actual, se puede dar preferencia a las acciones que todavía no se han intentado
  + Aumentar la ascensión de colinas con memoria resulta una aproximación más eficaz
    - La idea básica es almacenar la mejor estimación actual H(s) del costo para alcanzar el objetivo desde cada estado que ha visitado
    - El costo estimado para alcanzar el objetivo a través de un vecino s’ es el consto para s’ más cl costo estimado para conseguir un objetivo desde ahí
  + El **esquema AA\*TR** aprendiendo A\* en tiempo real
    - Construye un mapa del entorno usando la tabla resultado
    - Actualiza el costo estimado para el estado que acaba de dejar y entonces escoge el movimiento “aparentemente mejor” según sus costos estimados actuales
    - Las acciones que todavía no se han intentado en un estado s siempre se supone que dirigen inmediatamente al objetivo con el costo menor posible, esto es **optimismo bajo incertidumbre**
      * Esto anima al agente a explorar nuevos y posibles caminos prometedores
    - Un agente AA\*TR garantiza encontrar un objetivo en un entorno seguramente explorable y finito, no es completo para espacios de estados infinitos

**Aprendizaje en la búsqueda online**

* La ignorancia de los agentes de búsqueda online proporcionan varias oportunidades de aprender
* Primero, los agente aprenden un “mapa” del entorno (resultado de cada acción en cada estado) registrando cada una de sus experiencias
* Segundo, los agentes de búsqueda locales adquieren estimaciones más exactas del valor de cada estado usando las reglas de actualización local, como AA\*TR
  + Las actualizaciones convergen finalmente a valores exactos para cada estado
  + Una vez que se conoces los valores exactos, se pueden tomar decisiones óptimas simplemente moviéndose al sucesor con el valor más alto

**Palabras clave**

|  |  |
| --- | --- |
| **Búsqueda offline** | calcular la solución completa antes de poner un pie en el mundo real y luego ejecutan la solución sin recurrir a sus percepciones |
| **Búsqueda online** | Intercambiar cálculo y acción |
| **Problemas de exploración** | un problema de este tipo los estados y acciones son desconocidos por el agente |
| **Argumento de adversario** | es que podemos imaginar un adversario que construye el espacio de estados, mientras el agente lo explora  Puede poner callejones sin salida donde guste |
| **seguramente explorable** | Algún estado objetivo es alcanzable desde cualquier estado alcanzable |
| **Camino aleatorio** | Selecciona simplemente al azar una de las acciones disponibles del estado actual, se puede dar preferencia a las acciones que todavía no se han intentado |
| **AA\*TR** | Construye un mapa del entorno usando la tabla resultado |
| **Optimismo bajo la incertidumbre** | Las acciones que todavía no se han intentado en un estado s siempre se supone que dirigen inmediatamente al objetivo con el costo menor posible |