**Capítulo 3 – Resolver problemas mediante búsqueda**

**3.5 – Evitar estados repetidos**

* Se puede perder tiempo expandiendo estados que ya han sido visitados y expandidos
* Para algunos problemas la repetición de estados es inevitable
  + Incluye problemas donde las acciones son reversibles
* Se puede cortar el árbol de búsqueda en un tamaño finito generando solo parte del árbol
* Considerando solamente el árbol de búsqueda hasta cierta profundidad fija, es fácil encontrar casos donde la eliminación de estados repetidos produce una reducción en costo de búsqueda
* La detección significa comparación del nodo a expandir con aquellos que han sido ya expandidos
* Si un algoritmo recuerda cada estado que ha visto, puede verse como la exploración directamente del grafo de espacio de datos
  + Modificamos el algoritmo general de búsqueda-arboles para incluir una estructura de datos, **lista cerrada,** almacena cada nodo ya expandido.
  + Si el nodo actual se empareja con un nodo de la lista cerrada, se elimina en vez de expandirlo
  + Puede ocurrir que eliminemos el camino recién descubierto y este era el óptimo, pero con la búsqueda de coste uniforme o búsqueda primero en anchura con costos constantes no ocurre.

**Palabras clave**

|  |  |
| --- | --- |
| **Rejilla rectangular** | Sobre una rejilla, cada estado tiene cuatro sucesores |
| **Lista cerrada** | Almacena cada nodo expandido |