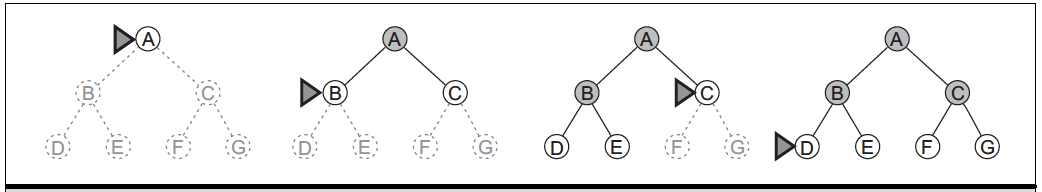
**Capítulo 3 – Resolver problemas mediante búsqueda**

**3.4 – Estrategias de búsqueda no informada**

* Una **búsqueda no informada** no tiene información adicional acerca de los estados más allá de la que proporciona la definición del problema
  + lo único que pueden hacer es generar sucesores y distinguir entre un estado objetivo de uno que no lo es
* Una **búsqueda informada** es una estrategia que sabe si un estado no objetivo es “mas prometedor” que otro

**Búsqueda primero en anchura**

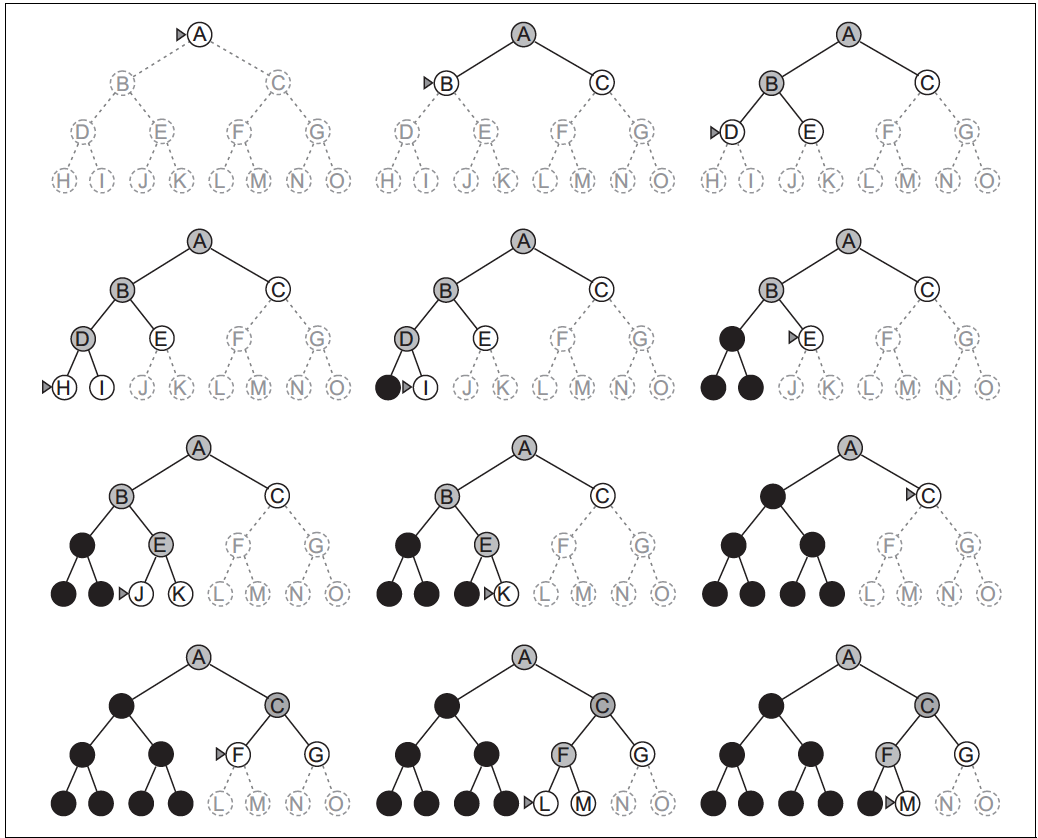
* Estrategia sencilla en la que se expande primero el nodo raíz, luego se expanden todos los sucesores de la raíz, luego sucesores, etc.
* Se expanden todos los nodos a una profundidad antes de expandir cualquier nodo del próximo nivel
* Es completa, la complejidad en espacio es la misma que la complejidad en tiempo



**Búsqueda de costo uniforme**

* Se expande el nodo n con el camino de costo más pequeño
* Si todos los costos son iguales, entonces es idéntico a búsqueda primero en anchura
* No se preocupa por el número de pasos que tiene un camino, pero si el costo total
* Se puede garantizar completitud si el costo de cada paso es mayor o igual a una constante positiva pequeña, también puede garantizar optimización
* Como el costo de un camino siempre aumenta cuando vamos por él, el algoritmo expande nodos que incrementan el costo del camino, el primer nodo objetivo seleccionando para la expansión es la solución óptima

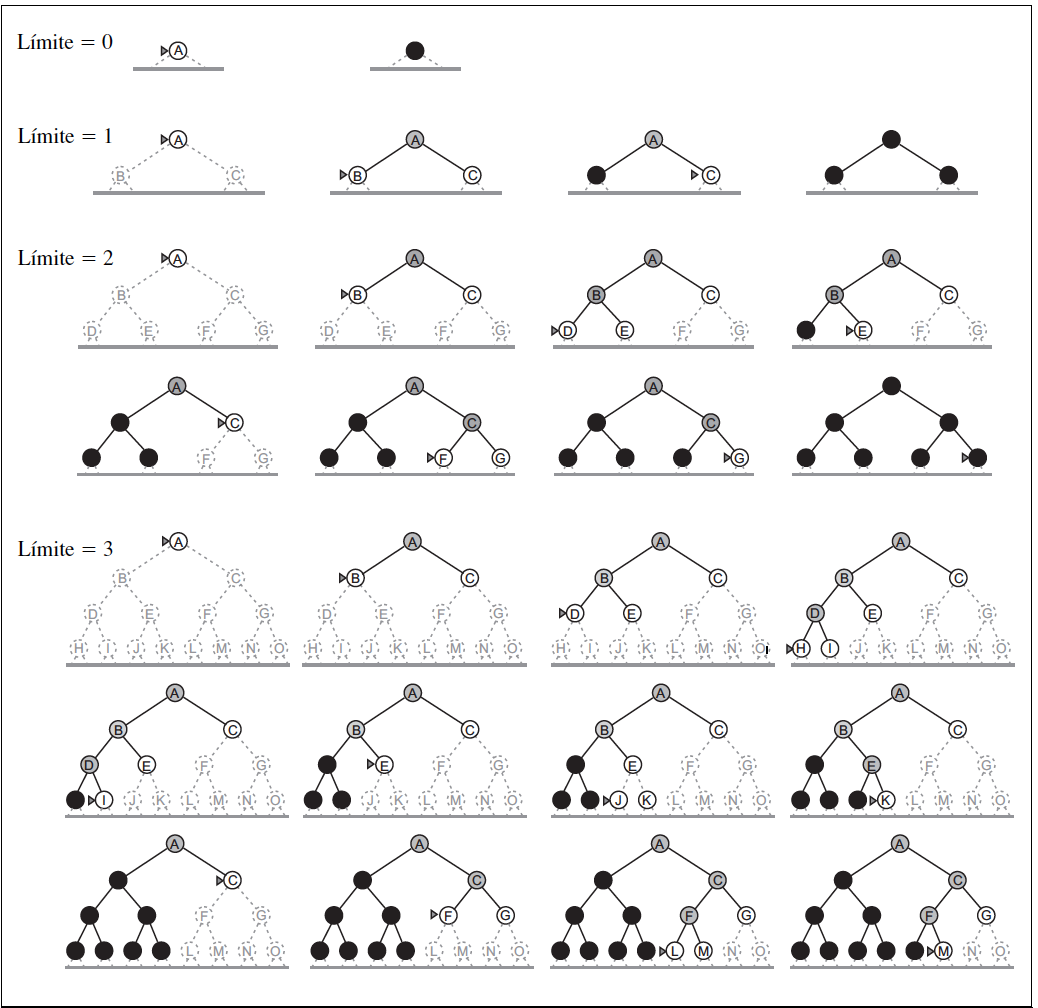
**Búsqueda primero en profundidad**

* Siempre expande el nodo más profundo en la frontera actual del árbol de búsqueda
* La búsqueda procede inmediatamente al nivel más profundo del árbol de búsqueda
* Cuando esos nodos se expanden, son quitados de la frontera, la búsqueda “retrocede” al siguiente nodo más superficial que todavía tenga sucesores inexplorados
  + Se puede implementar con cola LIFO
* Tiene requisitos modestos en memoria, necesita almacenar sólo un camino desde la raíz a un nodo hoja, junto con los nodos hermanos restantes no expandidos. Una vez el nodo se ha expandido, se puede quitar de la memoria tan pronto como todos su descendientes han sido explorados
* 
* Una variante de la búsqueda primero en profundidad es **búsqueda hacia atrás**
  + Solo se genera un nodo sucesor a la vez, cada nodo parcialmente expandido recuerda qué sucesor se expande a continuación
* Un inconveniente es que puede hacer una elección equivocada y obtener un camino muy largo aun cuando una elección diferente llevaría a una solución cerca de la raíz del árbol. No es óptima

**Búsqueda de profundidad limitada**

* Se hace búsqueda primero en profundidad pero se agrega un límite L de profundidad determinado. Los nodos a profundidad L se tratan como si no tuviera sucesor
* Resuelve el problema de camino infinito pero introduce problema de incompletitud si el objetivo está fuera del límite de profundidad

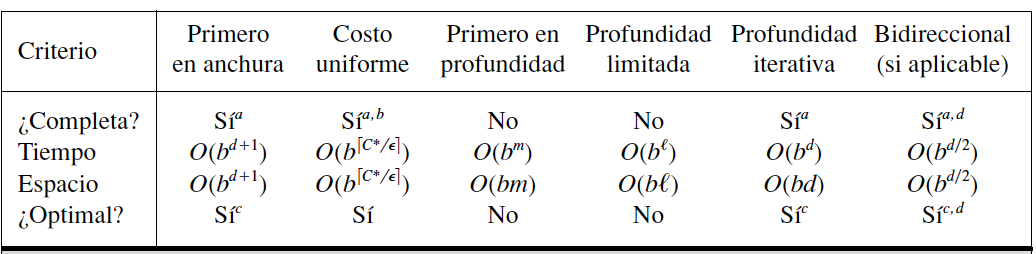
**Búsqueda primero en profundidad con profundidad iterativa**

* Estrategia en la cual encuentra el mejor límite profundidad
* Se usa combinada con búsqueda primero en profundidad
* Se va aumentando gradualmente el límite (0, 1, 2..) hasta encontrar un objetivo
* La profundidad iterativa es el método de búsqueda no informada preferido cuando hay un espacio grande de búsqueda y no se conoce la profundidad de la solución
* 

**Búsqueda bidireccional**

* Ejecutar dos búsquedas simultáneas: una hacia adelante desde el estado inicial, y la otra hacia atrás desde el objetivo
* Se implementa teniendo una o dos búsquedas que comprueban antes de ser expandido si cada nodo está en la frontera del otro árbol de búsqueda
  + Si esto ocurre, se ha encontrado una solución
* La debilidad más significativa es el espacio ya que por lo menos uno de los dos árboles de búsqueda se debe mantener en memoria
* Es completo y óptimo si las búsquedas son primero en anchura
* Como se busca hacia atrás?
  + Usando los **predecesores** todos los nodos que tienen como sucesor a “n

**Comparación de las estrategias de búsqueda no informada**

****

**Palabras clave**

|  |  |
| --- | --- |
| **Búsqueda no informada** | Estrategia que no se tiene información adicional acerca de los estados mas allá de la que proporciona la definición del problema |
| **Búsqueda informada** | Estrategia que sabe si un estado no objetivo es “mas prometedor” que otro |
| **Búsqueda primero en anchura** | Estrategia sencilla en la que se expande primero el nodo raíz, luego se expanden todos los sucesores de la raíz, luego sucesores, etc. |
| **Búsqueda de costo uniforme** | Se expande el nodo n con el camino de costo más pequeño |
| **Búsqueda primero en profundidad** | Siempre expande el nodo más profundo hasta que el nodo no tenga sucesores |
| **Búsqueda hacia atrás** | Solo se genera un nodo sucesor a la vez, cada nodo parcialmente expandido recuerda qué sucesor se expande a continuación |
| **Búsqueda de profundidad limitada** | Se hace búsqueda primero en profundidad pero se agrega un límite L de profundidad determinado. Los nodos a profundidad L se tratan como si no tuviera sucesor |
| **Diámetro** | Cantidad de pasos máximos que se toma de un nodo a otro |
| **Búsqueda primero en profundidad con profundidad iterativa** | Hace primero en profundidad pero cuando todos los nodos se han expandido y no se ha encontrado la solución, se incrementa el límite de profundidad |
| **Búsqueda bidireccional** | Ejecutar dos búsquedas simultáneas: una hacia adelante desde el estado inicial, y la otra hacia atrás desde el objetivo |
| **Predecesor** | Todos los nodos que tienen ese nodo como sucesor |