



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Cuaderno de trabajo: Búsqueda Primero-El-Mejor: Búsqueda voraz (en grafo)¹

Albert Sanchis

DSIC

Departamento de Sistemas
Informáticos y Computación

¹Para una correcta visualización, se requiere Acrobat Reader v. 7.0 o superior

Objetivos formativos

- ▶ Caracterizar la búsqueda convencional en un grafo de estados.
- ▶ Describir búsqueda voraz Primero-El-Mejor (en grafo).
- ▶ Construir el árbol de búsqueda voraz Primero-El-Mejor (en grafo).
- ▶ Aplicar búsqueda voraz Primero-El-Mejor (en grafo) a un problema clásico.
- ▶ Analizar la calidad de búsqueda voraz Primero-El-Mejor (en grafo).

Problema: La ruta más corta entre dos puntos

Búsqueda de una ruta más corta desde Arad a Bucarest [1]:



$\text{Acciones}(\text{Arad}) = \{\text{Ir}(\text{Sibiu}), \text{Ir}(\text{Timisoara}), \text{Ir}(\text{Zerind})\}.$

Problema: La ruta más corta entre dos puntos

Distancias en línea recta a Bucharest

	Bucharest		Bucharest
Arad	366	Mehadia	241
Bucharest	0	Neamt	234
Craiova	160	Oradea	380
Drobeta	242	Pitesti	100
Eforie	161	Rimnicu	193
Fagaras	176	Sibiu	253
Giurgiu	77	Timisoara	329
Hirsova	151	Urziceni	80
Iasi	226	Vaslui	199
Lugoj	244	Zerind	374

El algoritmo Primero-El-Mejor (en grafo) [2]

```
BF( $G, s', f$ )           // Best-First;  $G, s', f$  función de evaluación
 $O = \text{IniCola}(s', f(s'))$            // Open: cola de prioridad  $f$ 
 $C = \emptyset$                        // Closed: nodos explorados
mientras no  $\text{ColaVacía}(O)$ :           // 1ro el mejor:  $s = \arg \min_{n \in O} f_n$ 
     $s = \text{Desencola}(O)$                // desempates a favor de objetivos
    si  $\text{Objetivo}(s)$  retorna  $s$            // solución encontrada!
     $C = C \cup \{s\}$                        //  $s$  explorado
    para toda  $(s, n) \in \text{Adyacentes}(G, s)$ : // generación:  $n$  hijo de  $s$ 
         $x = f(n)$                        // posible  $f_n$  nuevo
        si  $n \notin C \cup O$ :  $\text{Encola}(O, n, f_n \triangleq x)$ 
        si no si  $n \in O$  y  $x < f_n$ :  $\text{Modcola}(O, n, f_n \triangleq x)$ 
        si no si  $n \in C$  y  $x < f_n$ :  $C = C \setminus \{n\}$ ;  $\text{Encola}(O, n, f_n \triangleq x)$ 
retorna NULL                       // ninguna solución encontrada
```

- **Cuestión 1:** Haz una traza del algoritmo *Primero-El-Mejor* (en grafo) aplicado al problema de búsqueda de una ruta más corta desde Arad a Bucarest.

O	C	s
{Arad (c=366)}	{}	—
{Sibiu (c=253), Timisoara (c=329), Zerind (c=374)}	{Arad (c=366)}	Arad
{Fagaras (c=176), Rimnicu (c=193), Timisoara (c=329), Zerind (c=374), Oradea (c=380)}	{Arad (c=366), Sibiu (c=253)}	Sibiu
{Bucharest (c=0), Rimnicu (c=193), Timisoara (c=329), Zerind (c=374), Oradea (c=380)}	{Arad (c=366), Sibiu (c=253), Fagaras (c=176)}	Fagaras
{Rimnicu (c=193), Timisoara (c=329), Zerind (c=374), Oradea (c=380)}	{Arad (c=366), Sibiu (c=253), Fagaras (c=176)}	Bucharest

- **Cuestión 2:** Construye el árbol de búsqueda resultante de aplicar el algoritmo **Primero-El-Mejor** (en grafo) al problema de búsqueda de una ruta más corta desde Arad a Bucarest.

- ▶ **Cuestión 3:** ¿El algoritmo encuentra solución? **Sí**
- ▶ **Cuestión 4:** Si la respuesta es “Sí”:
 - ▷ ¿Cuál ha sido la solución encontrada? **El camino solución encontrado ha sido: Arad, Sibiu, Fagaras, Bucharest**
 - ▷ ¿Cuál es el coste de esta solución? **450**
 - ▷ ¿Se trata de la solución óptima? **No, porque existe otra solución con un menor coste de 418: Arad, Sibiu, Rimnicu, Pitesti, Bucharest**

Referencias

- [1] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, third edition, 2010.
- [2] J. Pearl. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley, 1984.