Camino de costo mínimo desde un vertice fuente fijo hasta un vértice objetivo

Guillermo Palma

Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información

Plan

1. Planteamiento del problema

2. Solución con un algoritmo no informado

3. Algoritmos informados de búsqueda de caminos de costo mínimo

Planteamiento del problema

Camino de costo mínimo hasta fuentes metas

Definición del problema del camino de costo mínimo desde un vértice fuente fijo hasta un vértice objetivo

Dado un digrafo $G = (V, E)$ sin ciclos negativos y con una función de costo $w : E \to \mathbb{R}$ que asigna un costo real a cada uno de los lados del grafo, dado un vértice fuente fijo $s \in V$ y dado un conjunto de vértices objetivos O, tal que $O \subseteq V$. Se define **el problema del camino de costo mínimo desde un vértice fuente fijo hasta un vértice objetivo**, como el de encontrar un camino de costo mínimo $\delta(s, v)$ desde el vértice s hasta cualquier vértice $v \in O$.

Ejemplo de un grafo con vértice de partida y objetivo

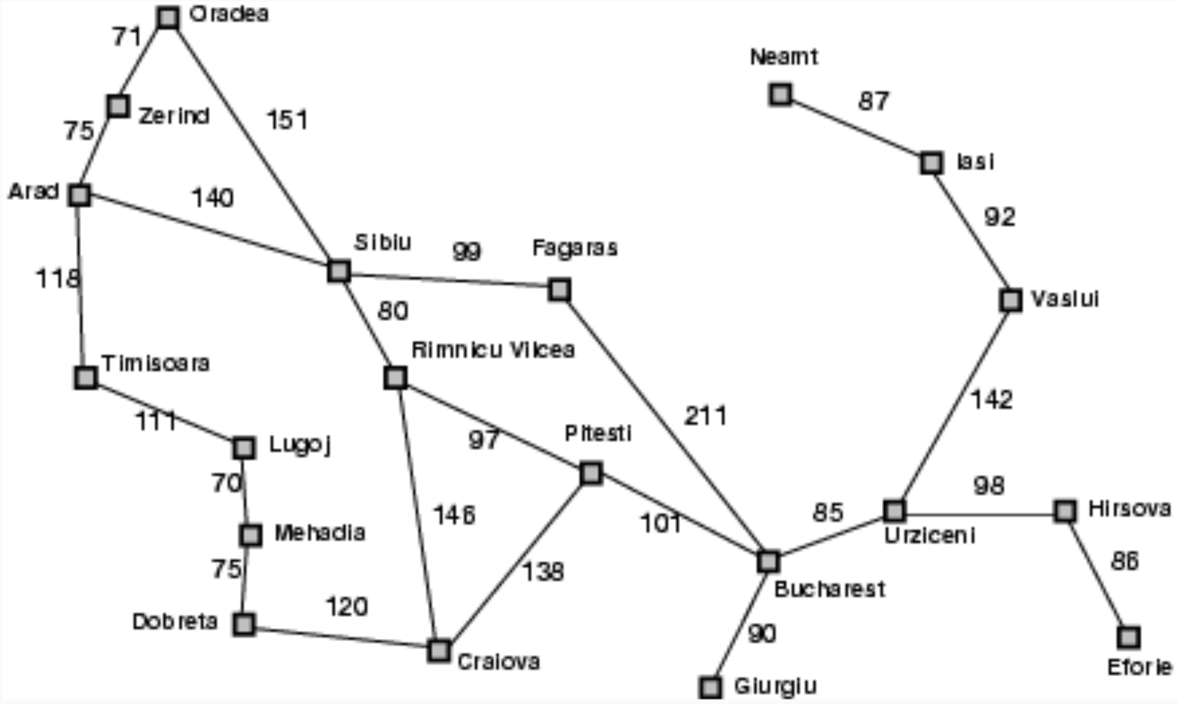

Figura 1: Grafo que representa a un mapa de Rumania. El vértice de partida es la ciudad de **Oradea** y el de llegada **Bucharest**.

Solución con un algoritmo no informado

Algoritmo de Dijkstra para un vértice objetivo

Función DijkstraHastaObjetivo($G = (V, E)$, s, w, O)

inicio
- `Inicializar-fuente-fija`(G, s) ;
- $CLOSED \leftarrow \emptyset$;
- $OPEN \leftarrow V$; `// Cola de prioridad, implícito Insert`
- **mientras** $OPEN \neq \emptyset$ **hacer**
  - $u \leftarrow$`Extract-MIN`($OPEN$) ;
  - **si** $u \in O$ **entonces**
    - **devolver** El camino de costo mínimo desde s hasta u;
  - **en otro caso**
    - $CLOSED \leftarrow CLOSED \cup \{u\}$;
    - **para cada** $v \in G.adyacentes[u]$ **hacer**
      - **si** $(v.d > u.d + w(u, v))$ **entonces**
        - $v.d \leftarrow u.d + w(u, v)$; `// Implícito Decrease-Key`
        - $v.pred \leftarrow u$;
- **devolver** NIL

Algoritmos informados de búsqueda de caminos de costo mínimo

Función de costo mínimo h

Definición de la función de costo mínimo h [1]

Sean A y B dos subconjuntos del conjunto de vértices de un digrafo $G = (V, E)$. Se define el **costo mínimo para ir desde A hasta B** como:

$$h(A, B) = \textit{ínfimo}\ \{w(P) : P \text{ es un camino que comienza con un vértice de } A \text{ y finaliza con un vértice de } B \text{, y cualquier vértice distinto de los extremos, no se encuentra ni en } A \text{ ni en } B\} \quad (1)$$

Por convención:

- $\textit{ínfimo}(\emptyset) = \infty$.
- Si A o B son singletones, se obvia la notación de conjunto. Por ejemplo: si $A = \{s\}$, entonces $h(\{s\}, B) = h(s, B)$.

Función de costo mínimo h hasta el objetivo

Definición de la función de costo mínimo h hasta el objetivo [1]

Sean A y O dos subconjuntos del conjunto de vértices de un digrafo $G = (V, E)$, y sea O un conjunto de vértices objetivos. Se define:

$$h(A, O) = h(A) \quad (2)$$

Por convención:

- $\textit{ínfimo}(\emptyset) = \infty$.
- Si A es un singleton, se obvia la notación de conjunto. Por ejemplo: si $A = \{n\}$, entonces $h(\{n\}) = h(n)$.

Función de costo g

Definición de la función de costo g [1]

Sean A un subconjunto del conjunto de vértices de un digrafo $G = (V, E)$, y sea $s \in V$ un vértice fuente (también llamado de partida). Se define:

$$g(A) = h(s, A) \tag{3}$$

Por convención:

- Si A es un singleton, se obvia la notación de conjunto. Por ejemplo: si $A = \{n\}$, entonces $g(\{n\}) = g(n)$.

Función estimativa $\hat{h}$

Definición de la función estimativa $\hat{h}$

Dado un digrafo $G = (V, E)$ sin ciclos negativos y con una función de costo $w : E \to \mathbb{R}$ que asigna un costo real a cada uno de los lados del grafo, dado un vértice fuente fijo $s \in V$ y dado un conjunto de $O \subseteq V$ de vértices objetivos. Se define la función $\hat{h} : V \to \mathbb{R}$, como una función que para todo vértice $n \in V$, se tiene que $\hat{h}(n)$ **es un valor estimado de** $h(n)$.

Función estimativa del camino de costo mínimo

Definición de la función estimativa $\hat{f}$

Dado un digrafo $G = (V, E)$ sin ciclos negativos y con una función de costo $w : E \to \mathbb{R}$ que asigna un costo real a cada uno de los lados del grafo, dado un vértice fuente fijo $s \in V$ y dado un conjunto de $O \subseteq V$ de vértices objetivos. Sea $P = \langle s, \ldots, n \rangle$ un camino desde s hasta cualquier vértice $n \in V$. Se define **la función estimativa del camino de costo mínimo**, desde s hasta un vértice objetivo en O a través de P como: $\hat{f}(P) = w(P) + \hat{h}(n)$.

Sobre los algoritmos informados

- El objetivo es encontrar a un vértice meta desde un vértice objetivo.
- Se está resolviendo un problema de búsqueda.
- Hay casos en los que el grafo a explorar es muy grande.
- Se quieren algoritmos que lleguen a la meta de forma más eficiente en tiempo y espacio.
- La idea es explotar las características del grafo para hacer más eficiente la búsqueda.
- Los algoritmos informados explotan la información sobre el dominio del problema de búsqueda.
- Los algoritmos informados pueden llegar al vértice meta explorando menos vértices del grafo que los algoritmo no informados.
- Una información a explotar podría ser un estimado de $\hat{h}(n)$, para todo vértice $n \in V$.
- Los algoritmos sobre grafos vistos hasta ahora son no informados.

Ejemplo de un grafo con vértice de partida y objetivo

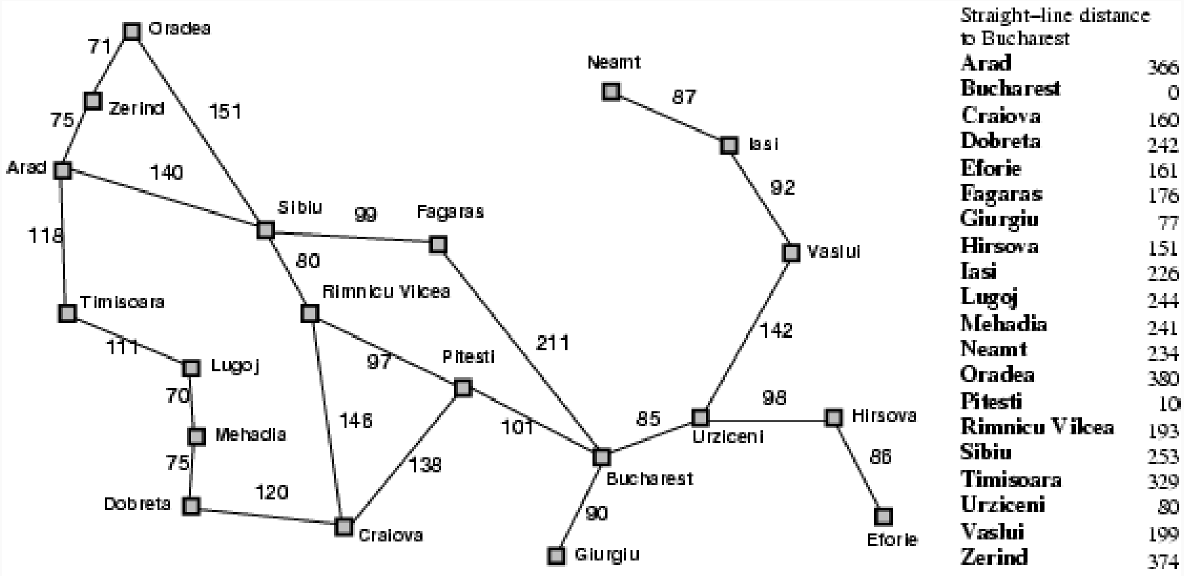

Figura 2: Grafo que representa a un mapa de Rumania. En el lado derecho del mapa se observa la distancia en línea recta desde cada ciudad a **Bucharest**.

Ejemplo de un vértice de partida y objetivo del Eight-Puzzle

Figura 3: Ejemplo de un vértice partida y un vértice objetivo. Los vértices son estados del Eight-Puzzle. Fuente [2].

Ejemplo del grafo del Eight-Puzzle

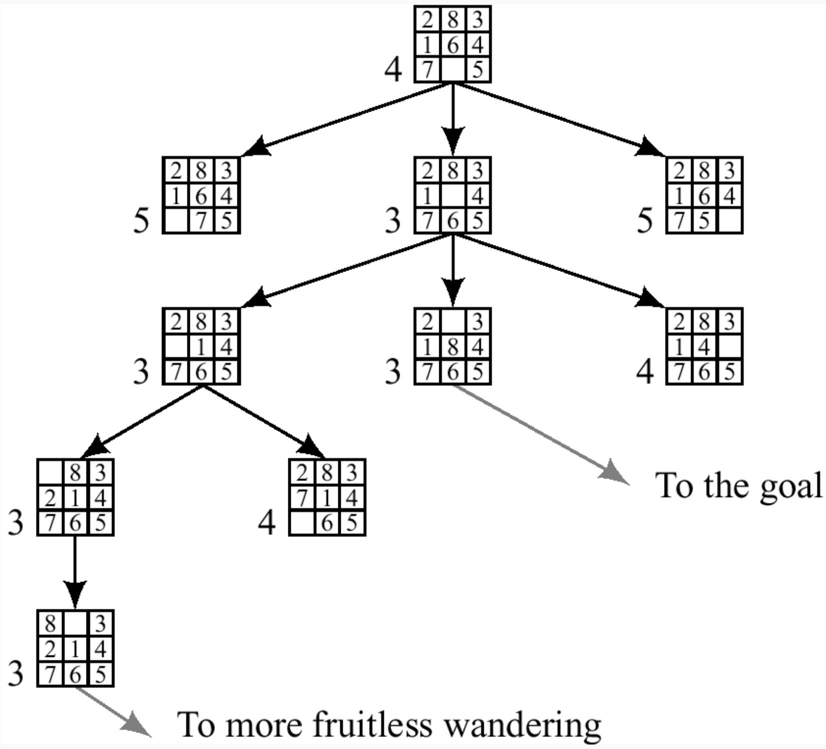

Figura 4: Grafo parcial del 8-Puzzle. El número de los vértices, es el número de cuadros en posición diferente del vértice objetivo de la Figura 3. Fuente [2].

Ejemplo del comportamiento del algoritmos de Dijkstra

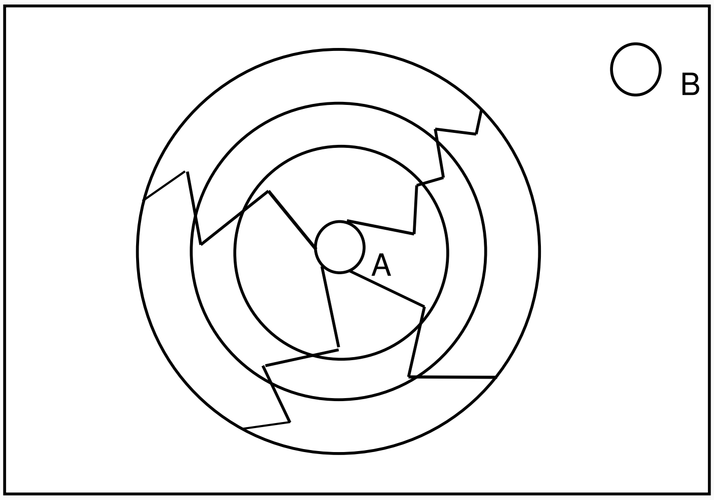

Figura 5: Ejemplo del avance del algoritmo de Dijkstra desde un vértice A hasta un vértice objetivo B por etapas. Fuente [1].

Ejemplo del comportamiento de un algoritmos informado

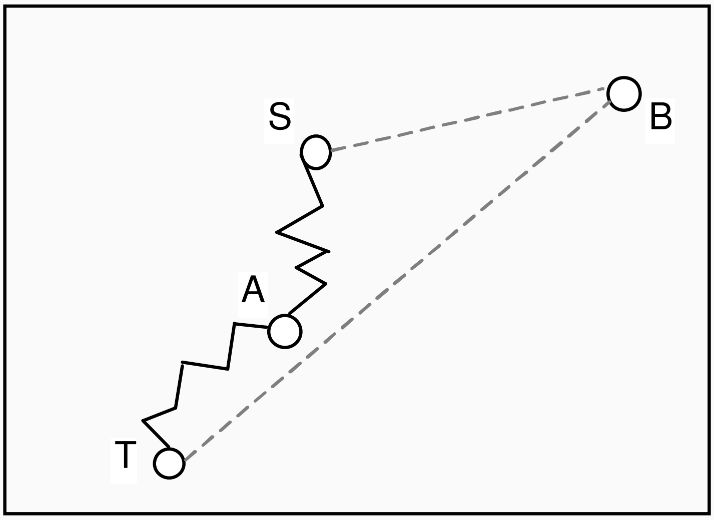

Figura 6: Comportamiento de un algoritmo informado. El vértice de partida es A y el objetivo es B. Las líneas punteadas son las distancias euclidianas. Observe que $w(A, S) > w(A, T)$ y $\hat{h}(S) < \hat{h}(T)$. Se tiene que $(\hat{f}(S) = w(A, S) + \hat{h}(S)) < (\hat{f}(T) = w(A, T) + \hat{h}(T))$. Por lo tanto, es más prometedor para un algoritmo informado expandir el vértice S. Fuente [1].

Sobre el algoritmo A^*

- Es un algoritmo de búsqueda informado.
- Suponemos que se tiene una función estimativa $\hat{h}(v)$, $\forall v \in V$.
- Se expande en la búsqueda el camino P_v con el vértice más prometedor v, con el menor valor dado por $\hat{f}(v) = w(P_v) + \hat{f}(v)$.
- Mantiene una cola de prioridad $OPEN$ de vértices abiertos que no han sido explorados ordenados por su valor de $\hat{f}$.
- Se tiene un conjunto de nodos explorados llamada $CLOSED$.
- En cada iteración se escoge a explorar el vértice con menor valor $\hat{f}$ de $OPEN$
- Se verifica si el vértice a explorar es una meta. Si no lo es, se visitan (agregando a $OPEN$) los vértices adyacentes (o succesores) que no hayan sido explorados.
- El algoritmo termina si se cumplen las condiciones del problema *camino de costo mínimo hasta fuentes metas* y si $O \neq \emptyset$.

Algoritmo de A^*

Función A-Estrella($G =(V, E), s, w, \hat{h}, O$)

inicio
  $CLOSED \leftarrow \emptyset$; $OPEN \leftarrow$ `Inicializar-Cola-Prioridad()` ;
  $s.\hat{f} \leftarrow \hat{h}(s)$;
  $s.pred \leftarrow NIL$;
  `Insert(`$OPEN, s$`)` ;
  $u \leftarrow$`Extract-MIN(`$OPEN$`)` ;
  mientras $u \notin O$ **hacer**
    $CLOSED \leftarrow CLOSED \cup \{u\}$;
    para cada $v \in (G.adyacentes(u) \setminus CLOSED)$ **hacer**
      $\hat{f_{new}} \leftarrow u.\hat{f} - \hat{h}(u) + w(u,v) + \hat{h}(v)$;
      si $v \notin OPEN$ **entonces**
        $v.\hat{f} \leftarrow \hat{f_{new}}$;
        $v.pred \leftarrow u$;
        `Insert(`$OPEN, v$`)` ;
      en otro caso
        si $\hat{f_{new}} < v.\hat{f}$ **entonces**
          $v.\hat{f} \leftarrow \hat{f_{new}}$; // `Implícito Decrease-Key`
          $v.pred \leftarrow u$;
    $u \leftarrow$`Extract-MIN(`$OPEN$`)` ;
  devolver El camino encontrado desde s hasta u

Función $\hat{h}$ admisible

Definición función $\hat{h}$ admisible [1]

Una función estimativa $\hat{h}$ se llama **admisible** si para todo $n \in V$, se cumple que $\hat{h}(n) \leq h(n)$.

Propiedades de A^*

Proposición V.4.1 [1]

Supongamos que $\hat{h}$ es admisible y que el objetivo es alcanzable desde s. Si al comienzo de una iteración cualquiera no se ha cerrado ningún camino con vértice terminal en O, entonces el algoritmo A^* cierra un camino P_v desde s hasta v con $\hat{f}(P_v) \leq h(s)$.

Propiedades de A^*, cont.

Proposición V.4.2 [1]

Sea $\hat{h}$ admisible en el algoritmo de A^*. Si P_j es un camino desde s hasta j, cuando j es escogido en la cola de prioridad en la línea 17, y sea $j \in O$ y sea j el primer vértice que se cierra de O, entonces P_j es un camino de costo mínimo desde s hasta O, y $w(P_J) = h(s)$.

Propiedades de A^*, cont.

Prueba proposición V.4.2

- Por Proposición V.4.1, se tiene que $\hat{f}(P_j) \leq h(s)$.
- Esto es, $\hat{f}(P_j) = w(P_j) + \hat{h}(j) \leq h(s)$.
- $\hat{h}(j) \leq h(j)$ y $h(j) = 0$ porque $j \in O$, entonces $\hat{h}(j) = 0$.
- Entonces, $w(P_j) \leq h(s)$, en consecuencia $w(P_j) = \delta(s, j)$.
- Por lo tanto, P_j es un camino de costo mínimo desde s hasta O $\square$.

Función $\hat{h}$ consistente

Definición función $\hat{h}$ consistente [1]

Decimos que una función estimativa $\hat{h}$ es **consistente** si es admisible y para todo $u, v \in V$, se cumple que $\hat{h}(u) \leq h(u, v) + \hat{h}(v)$.

Propiedades de A^*, cont.

Proposición V.4.3 [1]

Sea $\hat{h}$ una estimativa consistente. Entonces en cualquier iteración se tiene que $\hat{f}(P_u) \leq \hat{f}(P_v)$, para $v = 1, \ldots, q$, donde P_v son los caminos obtenidos de la expansión de los vértices $v \in (G.adyacentes(u) \setminus CLOSED)$ en la línea 7.

Propiedades de A^*, cont.

Prueba proposición V.4.3

Sea P_v un camino obtenido hasta v expandido desde P_u en la línea 7.

$$\begin{aligned}
\hat{f}(P_v) &= w(P_v) + \hat{f}(v) \\
&= w(P_u) + w(u,v) + \hat{f}(v) \text{ (se expandió } v \text{ desde } u) \\
&\geq w(P_u) + h(u,v) + \hat{f}(v) \ (h(u,v) \leq w(u,v)) \\
&\geq w(P_u) + \hat{h}(u) \ (\hat{h}(u) \text{ es consistente: } \hat{h}(u) \leq h(u,v) + \hat{h}(v)) \\
&= \hat{f}(P_u)
\end{aligned}$$

$\square$

Propiedades de A^*, cont.

Proposición V.4.4 [1]

Sea $\hat{h}$ consistente. Todo vértice $v \in V$ agregado al conjunto $CLOSED$, satisface que $w(P_v) = g(v) = \delta(s, v)$. Es decir, P_v es un camino de costo mínimo.

Referencias

[1] O. Meza and M. Ortega.
Grafos y Algoritmos.
Editorial Equinoccio, 2005.

[2] N. Nils.
Artificial Intelligence: A New Synthesis.
Morgan Kaufmann, 1998.