Caminos mínimos.
Algoritmo de Dijkstra

Asignatura: **Investigación Operativa II**, 2° Grado en Estadística

Autor: Miguel Rodríguez Rosa

En la mayoría de las situaciones reales, las conexiones (arcos) que tienen los estados (nodos) tienen una característica propia, es decir, tienen un peso, que con frecuencia suele ser un coste, tiempo, distancia, etc. En estos **grafos ponderados** el objetivo es encontrar el camino que permita pasar de un nodo a otro, usando los arcos en un sentido dado (**grafo dirigido**) con el menor valor, peso, coste, tiempo, distancia... de tal manera que la suma de los pesos de los arcos que lo formen sea **mínima**. A esto se le denomina encontrar el camino mínimo (que no tienen porqué ser único) entre dos nodos. Por ejemplo, el camino más rápido entre dos ciudades, o el más corto, o en el que se consuma menos gasolina, o en el que se gaste menos dinero en peajes...

Existen diversos algoritmos de búsqueda de caminos mínimos. En la mayoría, la manera más eficaz de representar todo el proceso es mediante una matriz con tantas columnas como nodos, y una sucesión de filas, que indican la iteración del algoritmo.

A continuación se presenta uno de los algoritmos más usuales para realizar la búsqueda de un camino mínimo.

1. Algoritmo de Dijkstra

Suponiendo que los pesos entre los pares de nodos son **no negativos**, el algoritmo de Dijkstra encuentra los caminos mínimos entre un **nodo inicial** fijado y **todos los demás nodos**. Durante el proceso, se calculan etiquetas temporales para cada nodo, que representan los distintos pesos mínimos encontrados hasta el momento entre el nodo origen y el resto, y se van convirtiendo en etiquetas permanentes en cada iteración.

Sea n el número de nodos, 1 el nodo origen, $e_k(j)$ la etiqueta (temporal o permanente) del nodo j en la iteración k, y $c(i, j)$ el peso del arco que va del nodo i al nodo j. Si no existe un arco del nodo i al nodo j se establece $c(i, j) = \infty$.

- Iteración $k = 0$:
  - $e_0(1) = 0$.
  - $e_0(j) = c(1, j), \qquad \forall$ nodo $j \neq 1$.
  - Se marca como permanente la etiqueta del nodo origen.
  - Sea $i = \arg \min\limits_{j=1,\dots,n} \{e_0(j)\}$ para todo nodo j que todavía no tenga etiqueta permanente. Se marca como permanente la etiqueta del nodo i.

- Iteración $k = 1, 2, \ldots$:
  - $e_k(j) = e_{k-1}(j)$ para todo nodo j que ya tenga etiqueta permanente.
  - Para todo nodo j que todavía no tenga etiqueta permanente: $e_k(j) = \min \left\{ \begin{array}{c} e_{k-1}(j) \\ e_k(i) + c(i, j) \end{array} \right.$, siendo i el último nodo que se ha marcado como permanente.
  - Sea $i = \arg \min_{j=1,\ldots,n} \{e_k(j)\}$ para todo nodo j que todavía no tenga etiqueta permanente. Se marca como permanente la etiqueta del nodo i.
- Se repite el paso anterior hasta que se marquen como permanentes todos los nodos, obteniendo así los pesos de los caminos mínimos del nodo origen a todos los demás. Sea K el número de iteraciones que han sido necesarias.
- Para encontrar el camino mínimo entre el nodo origen y un nodo destino x:
  - Se comprueba si $e_K(x) - e_K(1) = c(1, x)$:
    - Si es así, el camino mínimo usa el arco que va del nodo origen al nodo x, y se termina el algoritmo de Dijkstra.
    - Si no es así, sea $i = \min_{j=2,\ldots,n} \{j \ : \ e_K(x) - e_K(j) = c(j, x)\}$ para todo nodo $j \neq x$. El camino mínimo usa el arco que va del nodo i al nodo x, y se busca el camino mínimo entre el nodo origen y el nodo i.