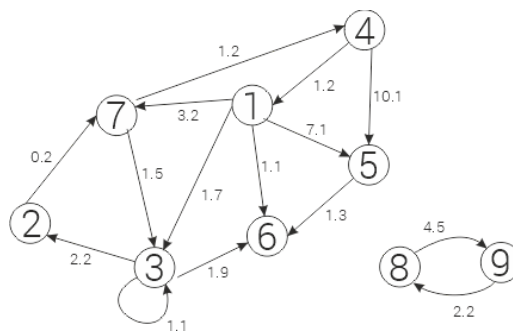


Optimiza!ción

Resolución tipo para grafo dirigido Actividades del tema 5

En este documento, explicaremos de forma sencilla las claves de cómo codificar un grafo dirigido. Resolveremos: conjunto de sucesores, conjunto de predecesores, grados de entrada y salida, matriz de incidencia vértice-arco, matriz de adyacencia y la matriz de costes.

Usaremos un ejemplo para ilustrar cada tipo, partiendo de su expresión gráfica:



Se trata de un grafo dirigido de de orden 9, con 15 arcos, es un 1-grafo, esto es, no hay más de un arco con el mismo origen y destino, y tiene un bucle en el nodo 3.

Sucesores del nodo i , son aquellos nodos j para los que existe el arco, (i, j) , es decir, los nodos j a los que me puedo desplazar desde i . Así,

$$\Gamma_1^+ = \{7, 6, 3, 5\}$$

$$\Gamma_2^+ = \{7\}$$

$$\Gamma_3^+ = \{3, 2, 6\}$$

$$\Gamma_4^+ = \{1, 5\}$$

$$\Gamma_5^+ = \{6\}$$

$$\Gamma_6^+ = \emptyset$$

$$\Gamma_7^+ = \{3, 4\}$$

$$\Gamma_8^+ = \{9\}$$

$$\Gamma_9^+ = \{8\}$$

Una vez contruidos los conjuntos de sucesores, el cómputo del grado de salida, es sencillo:

$$\delta^+(1) = 4, \delta^+(2) = 1, \delta^+(3) = 3, \delta^+(4) = 2, \delta^+(5) = 1, \delta^+(6) = 0, \delta^+(7) = 2, \delta^+(8) = 1 \text{ y}$$

$\delta^+(9) = 1$, que suman 15, el número de arcos, puesto que cada arco se codifica como sucesor de un nodo: el arco (i, j) aparece como $j \in \Gamma_i^+$.

Predecesores del nodo j , son aquellos nodos i para los que existe el arco, (i, j) , es decir, los nodos i de donde pudo haber venido desde i . Así,

$$\Gamma_1^- = \{4\}$$

$$\Gamma_2^- = \{3\}$$

$$\Gamma_3^- = \{1, 7, 3\}$$

$$\Gamma_4^- = \{7\}$$

$$\Gamma_5^- = \{1, 4\}$$

$$\Gamma_6^- = \{3, 7, 1\}$$

$$\Gamma_7^- = \{2, 1\}$$

$$\Gamma_8^- = \{9\}$$

$$\Gamma_9^- = \{8\}$$

Una vez contruidos los conjuntos de predecesores, el cómputo del **grado de llegada**, es sencillo: $\delta^-(1) = 1$, $\delta^-(2) = 1$, $\delta^-(3) = 3$, $\delta^-(4) = 1$, $\delta^-(5) = 2$, $\delta^-(6) = 3$, $\delta^-(7) = 2$, $\delta^-(8) = 1$ y $\delta^-(9) = 1$, que suman 15, el número de arcos, puesto que cada arco se codifica como predecesor de un nodo: el arco (i, j) aparece como $i \in \Gamma_j^-$.

Puedes comprobar que es una codificación completa, es decir, a partir de los sucesores puedes dibujar el grafo, y también, a través de los predecesores. Sucesores y predecesores son la estructura común para trabajar con grafos en algoritmos de optimización combinatoria.

Para construir la **matriz de incidencia vértice-arco** hay que listar los arcos, puesto que cada columna de la matriz va a representar uno. Una forma de hacerlo sería por orden lexicográfico, y así, las columnas representarían los arcos en este orden único:

$(1, 3); (1, 5); (1, 6); (1, 7); (2, 7); (3, 2); (3, 3); (3, 6); (4, 1); (4, 5); (5, 6); (7, 3); (7, 4); (8, 9); (9, 8)$

Las filas serían los nodos, con lo que tenemos la **matriz de incidencia vértice-arco A** siguiente:

	(1, 3)	(1, 5)	(1, 6)	(1, 7)	(2, 7)	(3, 2)	(3, 3)	(3, 6)	(4, 1)	(4, 5)	(5, 6)	(7, 3)	(7, 4)	(8, 9)	(9, 8)
1	1	1	1	1	0	0	0	0	-1	0	0	0	0	0	0
2	0	0	0	0	1	-1	0	0	0	0	0	0	0	0	0

3	-1	0	0	0	0	1	1	1	0	0	0	-1	0	0	0
4	0	0	0	0	0	0	0	0	1	1	0	0	-1	0	0
5	0	-1	0	0	0	0	0	0	0	-1	1	0	0	0	0
6	0	0	-1	0	0	0	0	-1	0	0	-1	0	0	0	0
7	0	0	0	-1	1	0	0	0	0	0	0	1	1	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1
9	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1

Para construir la matriz, hemos tenido en cuenta:

- en cada columna, de cabecera (i, j) , situamos un +1 en la fila i , y un -1 en la fila j . Si es un bucle (i, i) , sólo un 1 en la fila i .
- el resto son 0.
- la suma por columnas es 0, excepto en los bucles, que suma 1.
- por filas, en la fila i , el número de “unos” (+1) es el grado exterior o de salida del nodo i ; el número de “menos unos” (-1) es el grado interior o de llegada del nodo i .

La **matriz de adyacencia** es, sin embargo, una matriz cuadrada de orden n , el orden del grafo. En este caso, referenciamos cada posición de la matriz según su fila y columna, y por tanto, el coeficiente de la posición ij -ésima es la casilla de la fila i y columna j . La codificación es sencilla: recorreremos los arcos y situamos un 1 para el arco (i, j) en la posición ij -ésima; cero en el resto. Así, construimos **la matriz de adyacencia A** siguiente:

	1	2	3	4	5	6	7	8	9
1	0	0	1	0	1	1	1	0	0
2	0	0	0	0	0	0	1	0	0
3	0	1	1	0	0	1	0	0	0
4	1	0	0	0	1	0	0	0	0
5	0	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0	0
7	0	0	1	1	0	0	0	0	0
8	0	0	0	0	0	0	0	0	1
9	0	0	0	0	0	0	0	1	0

Revisando la matriz de adyacencia, tenemos cosas curiosas:

- los sucesores están por filas (en la fila 1, están marcadas las columnas 3, 5, 6, y 7), con lo que el grado de salida es la suma por filas.
- los predecesores están por columnas (en la columna 3, están marcadas las filas 1, 3 y 7), con lo que el grado de entrada es la suma por columna.
- los bucles se sitúan en la diagonal principal (la posición 3, 3, es la del bucle en 3).

Finalmente, la **matriz de costes** sería una matriz cuadrada de orden n , y de forma similar a la matriz de adyacencia, identificamos la posición ij -ésima con el arco (i, j) , pero en esta ocasión, situamos el peso o coste del arco.

Así, construimos la **matriz de costes** W siguiente:

	1	2	3	4	5	6	7	8	9
1	∞	∞	1,7	∞	7,1	1,1	3,2	∞	∞
2	∞	∞	∞	∞	∞	∞	0,2	∞	∞
3	∞	2,2	1,1	∞	∞	1,9	∞	∞	∞
4	1,2	∞	∞	∞	10,1	∞	∞	∞	∞
5	∞	∞	∞	∞	∞	1	∞	∞	∞
6	∞	∞	∞	∞	∞	∞	∞	∞	∞
7	∞	∞	1,5	1,2	∞	∞	∞	∞	∞
8	∞	∞	∞	∞	∞	∞	∞	∞	4,5
9	∞	∞	∞	∞	∞	∞	∞	2,2	∞