A modo de apéndice: caminos más cortos en digrafos

En diferentes capítulos hemos estudiado diversos algoritmos con un cometidos similares: calcular el camino más corto entre un vértice s y otro t en un digrafo ponderado, o el conjunto de caminos más cortos entre un vértice s y cualquier otro, o el camino más corto entre todo par de vértices. Cada uno de los algoritmos estudiados presenta una serie de restricciones que limitan su aplicación a determinados tipos de digrafo. Hagamos una breve pausa para considerarlos.

Lo cierto es que los algoritmos que calculan el camino más corto entre un par de vértices s y t son variantes sencillas de algoritmos que calculan el conjunto de caminos más cortos entre un vértice s que detienen el cálculo al conocer la solución con destino en t. Los algoritmos que hemos estudiado son:

- Si todas las aristas del digrafo presentan idéntico peso, un recorrido por primero en anchura calcula el conjunto de caminos más en cortos en tiempo $O(|V| + |E|)$.
- Si el digrafo es acíclico, sea cual sea la función de ponderación, el algoritmo presentado en la sección 8.2 efectúa el cálculo en tiempo $O(|V| + |E|)$.
- Si la función de ponderación es positiva y el grafo no es acíclico, diferentes versiones del algoritmo de Dijkstra efectúan el cálculo en tiempo:
    - $O(|V|^2)$ sin usar diccionario de prioridad (eficiente en digrafos completos).
    - $O(|V| + |E| \lg |V|)$ usando un diccionario de prioridad basado en un heap.
    - $O(|E| + |V| \lg |V|)$ usando un diccionario de prioridad basado en un heap de Fibonacci.
- Si la función de ponderación no es positiva pero el grafo no presenta ciclos negativos, el algoritmo de Bellman-Ford resuelve el problema en tiempo $O(|V||E|)$.

El camino más corto entre todo par de vértices puede calcularse mediante el algoritmo de Floyd-Warshall en tiempo $O(|V|^3)$, pero también ejecutando $|V|$ veces el cálculo del camino más corto entre un vértice y todos los demás. Tiene interés observar que en digrafos con aristas de peso idéntico o en digrafos acíclicos, el coste temporal de esta propuesta alternativa es $O(|V||E|)$. En el caso de grafos densos, este coste temporal es comparable al de Floyd-Warshall, pero en grafos dispersos es preferible el algoritmo alternativo. Cuando el grafo presenta una función de ponderación positiva, el coste temporal de $|V|$ ejecuciones del algoritmo de Dijkstra en su versión con heaps con Fibonacci es $O(|V||E| + |V|^2 \lg |V|)$, que puede ser preferible al algoritmo de Floyd-Warshall en grafos dispersos. Finalmente, en digrafos sin ciclos negativos es siempre preferible el algoritmo de Floyd-Warshall, pues $|V|$ ejecucones de Bellman-Ford requieren tiempo $O(|V|^2|E|)$.