

Diseño e implementación de metaheurísticas con Python

Sergio Pérez Peló

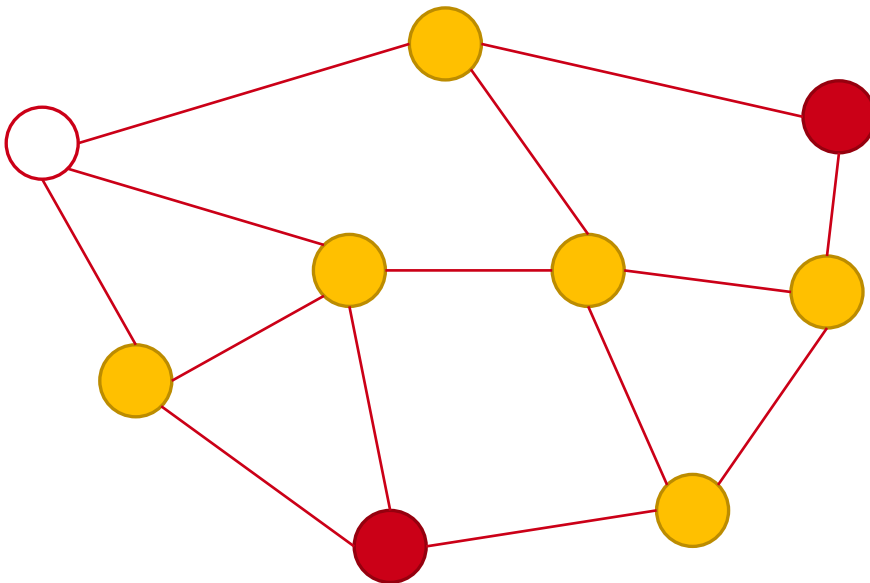
Jesús Sánchez-Oro



Universidad
Rey Juan Carlos

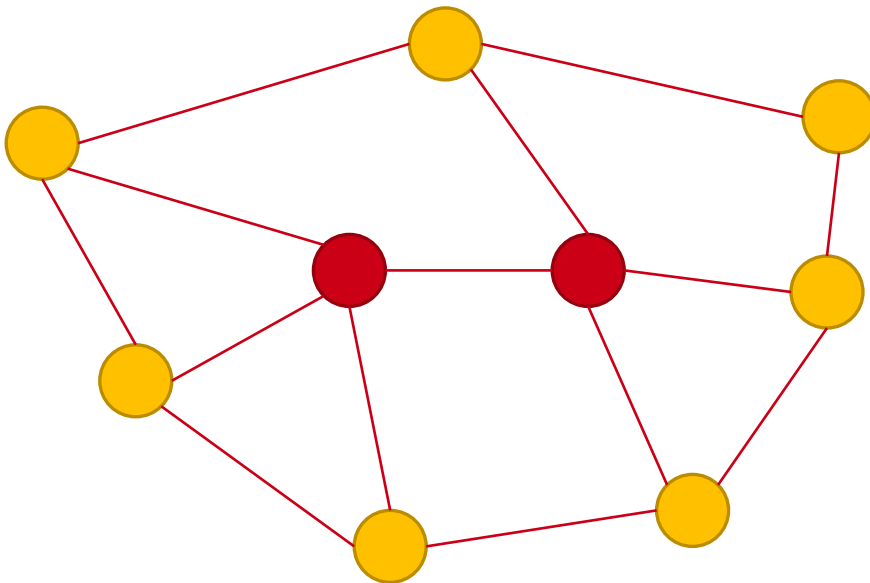
Dominación en grafos

- Dado un grafo $G = (V, E)$, encontrar un subconjunto $S \subseteq V$ tal que cualquier vértice pertenece a S o está conectado a un vértice de S



Dominación en grafos

- Dado un grafo $G = (V, E)$, encontrar un subconjunto $S \subseteq V$ tal que cualquier vértice pertenece a S o está conectado a un vértice de S



Dominación en grafos

- El coste de establecer un nodo dominante suele ser elevado en sus aplicaciones prácticas:
 - Vigilancia, redes sociales, etc.
- Buscamos encontrar la solución con un menor número de nodos dominantes.



Estructura del código

- Dividiremos el código en módulos.
- De momento, nos centramos en representar una instancia y una solución.



Instancia

- La instancia contiene toda la información relativa al grafo de entrada.
- ¿Cuál es la mejor representación para el grafo?
 - Matriz de adyacencias
 - Lista de adyacencias

Instancia

- **Lista de adyacencia:** por cada nodo, almacenamos una lista con sus adyacentes
 - Conocer los adyacentes $\rightarrow O(1)$
 - Saber si dos nodos son adyacentes $\rightarrow O(n)$
- **Matriz de adyacencia:** un 1 en la posición i,j indica que los nodos i y j son adyacentes.
 - Conocer los adyacentes $\rightarrow O(n)$
 - Saber si dos nodos son adyacentes $\rightarrow O(1)$

Instancia

- Debemos elegir una de las dos representaciones ... o no
- La memoria es un recurso que no suele estar limitado, el tiempo sí.
- En general, podemos considerar que la memoria es gratis.



Solución

- Debemos almacenar todo lo necesario para representar una solución al problema.
- Al menos, deberíamos guardar:
 - Nodos seleccionados.
 - Evaluación de la función objetivo.

Mejoras en la solución

- Nodos hoja: nodos que están conectados a un único nodo en el grafo.
- Nodos soporte: nodos que están conectados a un nodo hoja.
- Los nodos soporte **siempre** van a estar en una solución óptima, así que los incluiremos por defecto.



Generación de soluciones aleatorias

- En este punto ya podemos generar soluciones aleatorias.
- No serán buenas, pero sí que son un buen punto de inicio



Generación de soluciones aleatorias

- Partimos de una solución con los nodos soporte seleccionados.
- Iremos añadiendo nodos de manera aleatoria hasta que la solución sea factible.



Diseño e implementación de metaheurísticas con Python

Sergio Pérez Peló

Jesús Sánchez-Oro



Universidad
Rey Juan Carlos