Algoritmos y Estructuras de Datos III
Segundo Cuatrimestre 2020
Trabajo Práctico 2

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Introducción

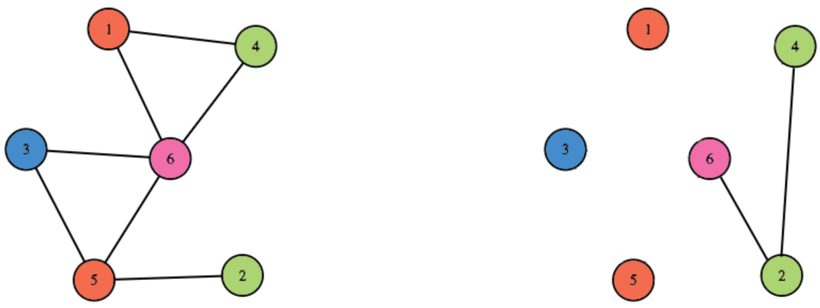
En optimización combinatoria se busca la mejor manera de resolver un problema entre un conjunto de posibles soluciones al mismo. Para hablar de la mejor manera, se debe asociar cada una de las posibles soluciones a una función objetivo que nos permita evaluarlas y compararlas entre sí.

Estos problemas son de mucho interés en la práctica porque se pueden modelar situaciones reales en donde tenemos que tomar una decisión sobre una tarea y la valuación puede representar una métrica que nos resulte de interés como una ganancia asociada o un costo a pagar.

Existen una gran variedad de problemas reales que se pueden resolver mediante optimización combinatoria. En nuestro caso apuntaremos al Problema de Coloreo de Máximo Impacto (PCMI). Formalmente, dados dos grafos $G = (V, E_G)$ y $H = (V, E_H)$ definidos sobre el mismo conjunto de vértices V y dado un conjunto de colores C, se define el impacto $I(c)$ sobre H de un coloreo $c : V \to C$ de G como el número de aristas $(i, j) \in E_H$ tales que $c(i) = c(j)$.

El PCMI surge naturalmente en problemas reales de asignación, como por ejemplo, en asignación de clases a aulas. Supongamos que tenemos un conjunto de clases cada uno identificado con el par (horario, materia). Podemos querer asignar las aulas para cada clase, de manera tal de maximizar la cantidad de veces que una materia está en la misma aula. En este caso, se define V como el conjunto de clases, G tiene una arista entre los vértices $i, j \in V$ si comparten horario, y H incluye una arista entre i y j si son de la misma materia.

La Figura 1 muestra un ejemplo del PCMI con $n = 6$ vértices. En este caso una solución óptima utiliza 4 colores, y tiene impacto 1.



(a) Grafo $G = (V, E_G)$ (b) Grafo $H = (V, E_H)$

Figura 1: Ejempo del PCMI con $n = 6$ vértices.

Problema

El Problema de Coloreo de Máximo Impacto (PCMI) toma dos grafos $G = (V, E_G)$ y $H = (V, E_H)$ definidos sobre el mismo conjunto de vértices V y un conjunto de colores C. Además, define el impacto $I(c)$ sobre H de un coloreo $c : V \to C$ de G como el número de aristas $(i, j) \in E_H$ tales que $c(i) = c(j)$. El objetivo es encontrar un coloreo válido de G con máximo impacto sobre H.

Dado que el PCMI pertenece a la categoría de problemas $\mathcal{NP}$-hard, no buscaremos dar la solución óptima para cada instancia sino una de la mejor calidad posible.

Parámetros y formato de entrada/salida

La entrada consistirá de una primera línea con tres enteros n, m_G, y m_H indicando la cantidad de vértices y aristas de los grafos G y H respectivamente. Luego le sucederán m_G líneas con 2 enteros (i, j) indicando los extremos de las aristas del grafo G, y m_H líneas con 2 enteros indicando las aristas de H.

Para la salida se debe imprimir una primera línea que contenga un entero I que indique el máximo impacto de la solución óptima, y luego una linea con los colores de los n vértices, en orden, separados por un espacio.

Entrada de ejemplo	**Posible salida esperada de ejemplo**
`6 7 2`	`1`
`4 6`	`4 1 6 1 4 3`
`5 6`	
`1 6`	
`2 5`	
`3 6`	
`1 4`	
`3 5`	
`2 4`	
`2 6`	

Enunciado

En el presente trabajo práctico se pide:

1. Describir el problema de PCMI dando ejemplos y soluciones
2. Describir situaciones de la vida real que puedan modelarse utilizando PCMI.
3. Diseñar e implementar para el PCMI las siguientes soluciones:
    - Al menos dos heurísticas constructivas golosas.
    - Metaheurística Tabú Search
        - Memoria basada en últimas soluciones exploradas.
        - Memoria basada en estructura (vértices).
4. Para los métodos implementados, desarrollar los siguientes puntos:
    a) Explicar detalladamente el algoritmo implementado.
    b) Calcular el orden de complejidad temporal de peor caso del algoritmo.
    c) Describir (si es posible) instancias de PCMI para las cuales el método no proporciona una solución óptima. Indicar (si es posible) qué tan mala puede ser la solución obtenida respecto de la solución óptima.
    d) Realizar una experimentación que permita observar la performance del algoritmo comparando la calidad de las soluciones obtenidas y los tiempos de ejecución en función de la entrada (y de otros parámetros de ser apropiado). Dentro de los casos de prueba se deben incluir también, como casos patológicos, aquellos descriptos en el ítem 4c. Para evaluar la calidad de las heurísticas con respecto a las soluciones óptimas conocidas, utilizar las instancias disponibles en el campus de la materia. En caso de que el algoritmo tenga algún parámetro configurable que determine su comportamiento (la metaheurística por ejemplo, aunque queda abierto a los demás también), se debe experimentar variando los valores de los parámetros y elegir, si es posible, la configuración que mejores resultados provea para el grupo de instancias utilizado. Presentar los resultados obtenidos mediante gráficos adecuados.
5. Una vez elegidos los mejores valores de configuración para cada heurística implementada, realizar una experimentación **sobre un conjunto nuevo de instancias** para observar la performance de los métodos comparando nuevamente la calidad de las soluciones obtenidas y los tiempos de ejecución en función del tamaño de entrada. Presentar todos los resultados obtenidos mediante gráficos adecuados y discutir al respecto de los mismos.