1 Ejercicio 1

1.1 Relación entre k-pmp y el k-coloreo

Podemos empezar preguntándonos si podemos resolver el problema de $k-coloreo$ con $k-pmp$. La respuesta es si, obteniendo ciertos resultados con $k-pmp$.
Primero hacemos una modificación en el grafo de entrada, poniendo peso 1 a todas las aristas. El problema de $k-pmp$ busca agrupar los vértices en k conjuntos tales que la suma de los costos(todas las aristas que se generan dentro) en cada conjunto,costo de la partición, sea mínima. Con esta modificación vamos a obtener la mínima cantidad de aristas (porque cada una suma 1) que se generan con k conjuntos.
Sabemos que cada vértice de un conjunto es no adyacente a otro, de otro conjunto, entonces podemos interpretar a cada conjunto como un color. Si el costo de la partición es 0, para tener un coloreo válido necesitamos k' colores, donde k' es la cantidad de conjuntos no vacíos y $k' \leq k$.
Si todos los conjuntos son no vacíos y el costo de la partición, es mayor que 1 quiere decir que hay por lo menos una arista dentro de un conjunto. Si sucede esto no tenemos un coloreo válido con k colores, porque existen 2 nodos que son adyacentes que tienen el mismo color. Este es un problema para coloreo pero no para k-pmp, entonces podemos decir que necesitamos 1 color más por cada arista que se genera, la cantidad de aristas es el costo de la partición. Si k-pmp da un costo t, podemos hacer un (k+t)-coloreo y este será válido.
Esto parece no servir mucho para encontrar $\chi\ (G)$, si tenemos un grafo G y $\chi\ (G)$ es c, corremos $k-pmp$ con k muy alejado de c, vamos a obtener un coloreo válido pero muy alejado del óptimo.
Si bien no encontramos el óptimo corriendo $k-pmp$ una vez podemos, tal vez podemos encontrarlo de alguna forma. Vimos que cuando aumentamos el k se aleja del óptimo, pero ¿Qué pasa si vamos bajando el k?
Corremos $k-pmp$ para $k = n-1$ (con $k = n$ es trivialmente $n-coloreable$) y nos guardamos el costo total T. Luego corremos i-pmp para $i \leq n-2$, actualizando T cuando es más grande que el costo obtenido en $i-pmp$, por el costo de $i-pmp$. En algún momento el costo del $i-pmp$ va a ser mayor a T (intuitivamente cuando disminuye k, menos puedo separar los nodos sin generar aristas en los conjuntos).
Entonces cuando encontremos un $i-pmp$ cuyo costo sea mayor al T, vamos a saber que todos los $T' > T$ van a ser coloreos válidos y los $T'' < T$ van a ser no validos con lo cual $\chi(G) = T$ que es el óptimo.
Si bien encontramos el óptimo, la complejidad de encontrarlo de esta forma no es buena. Tenemos que aplicar un algoritmo $k-pmp$ que no es polinomial sino exponencial, en peor caso n veces(es una cota muy grosera). Para un n suficientemente grande el tiempo de ejecución puede ser muy grande, con lo cual para la práctica no sirve este algoritmo.
Ya vimos que se puede resolver k-coloreo con k-pmp y que se puede encontrar el coloreo óptimo.

Ahora vamos a ver que sucede si tenemos $k-coloreo$ y queremos resolver $k-pmp$.
Si un grafo es $k-coloreable$ entonces el resultado de aplicar $k-pmp$ al grafo es costo 0 y k conjuntos sin aristas, sin conjuntos vacíos o $k' < k$ conjuntos sin arista, y el resto conjuntos vacíos. Veamos más detalladamente esto:
Si es $k-coloreable$ tenemos k grupos de vértices con un color cada grupo y en cada grupo los vértices no son adyacentes. Relacionándolo con $k-pmp$ vemos que es casi lo mismo, el costo de la partición va a ser 0, porque siempre busca el mínimo y no hay costo negativos porque todos los costos de las aristas son positivos, vamos a obtener k conjuntos sin aristas o $k' < k$ conjuntos no vacíos.
Si obtengo k' conjuntos no vacíos y costo 0, quiere decir que es $k'-coloreable$. Esto es porque tengo una partición de k' colores cuyo costo es 0, lo cual quiere decir que no tengo aristas en un conjunto, lo cual implica que todo vértice en un conjunto es no adyacente.