



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Investigación operativa(hay que esforzarse más en esto)

Investigación operativa
Segundo Cuatrimestre de 2015

Integrante	LU	Correo electrónico
Confalonieri, Gisela Belén	511/11	gise_5291@yahoo.com.ar
Mignanelli, Alejandro Rubén	609/11	minga_titere@hotmail.com



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

Índice

1. Introducción	3
2. Problema del Coloreo Particionado de Grafos	4
3. Modelado del problema	5
4. Código	6
5. Experimentación	7
6. Conclusiones y trabajo futuro	8

1. Introducción

En el presente trabajo se realizará un estudio comparativo entre las estrategias Branch and Bound y Cut and Branch sobre el problema del coloreo particionado de grafos (que se explicará en otra sección de este informe) encarado desde Programación Lineal Entera. Los objetivos del trabajo son los siguientes:

- Interacción con CPLEX:

CPLEX es un paquete de software comercial y académico para resolución de problemas de Programación Lineal y Programación Lineal Entera. Uno de los fines de nuestro trabajo consiste entonces en aprender a usar este paquete tanto resolviendo problemas, como reemplazando parte del trabajo que este realiza con código propio.

- Modelado de un PLE:

El modelado de un PLE cumple un rol fundamental, ya que como en general se modelan problemas cuyo mejor algoritmo es exponencial, es importante que el modelo no solo represente correctamente el problema, sino que permita observar la menor cantidad de soluciones posibles (no se como explicar bien esto (ALE)). Por ello, parte del objetivo de este trabajo es no solo realizar un PLE correcto, sino elaborar estrategias que permitan ver la menor cantidad de sol posibles.

- Desigualdades válidas, cortes e implementación de estos:

Otro de los propósitos de este trabajo es lograr un mayor entendimiento de que son y como funcionan los cortes. Si bien en la materia los estudiamos a nivel teórico, es importante toparnos con ellos a nivel práctico e intentar implementarlos. Para esto, utilizaremos las desigualdades válidas clique y odd-hole.

- Comparación entre distintos métodos de resolución

La última meta será realizar una comparación entre algunos métodos de resolución general:

- Branch-and-Bound:

(Breve explicación de que es branch and bound (ALE)). Es la resolución automática de CPLEX, por lo que le quitaremos todo el preprocesamiento y cortes que este paquete le añade, con el fin de obtener un resultado basado solamente en Branch-and-Bound

- Cut-and-Branch:

La idea es añadir cortes en el nodo raíz, para luego realizar un Branch-and-Bound clásico, que en teoría debería tomar menos tiempo dado que la formulación del PLE es más restrictiva. Sin embargo, dado que agregar un sinfín de restricciones también provoca una demora, un subobjetivo en esta etapa será encontrar una cantidad apropiada de cortes de manera que se acorte el branch and bound, pero que el preprocesamiento no tome demasiado (lo escribí como el orto (ALE)).

2. Problema del Coloreo Particionado de Grafos

El problema de coloreo de grafos ha sido ampliamente estudiado y aparece en numerosas aplicaciones de la vida real, como por ejemplo en problemas de scheduling, asignación de frecuencias, secuenciamiento, etc. Formalmente, el problema puede ser definido de la siguiente forma:

Dado un grafo $G = (V, E)$ con $n = |V|$ vértices y $m = |E|$ aristas, un coloreo de G consiste en una asignación de colores o etiquetas a cada vértice $p \in V$ de forma tal que todo par de vértices $(p, q) \in E$ poseen colores distintos. El problema de coloreo de grafos consiste en encontrar un coloreo que utilice la menor cantidad posible de colores distintos.

A partir de diferentes aplicaciones, surgieron variaciones o generalizaciones de este problema, como el problema de coloreo particionado de grafos. En este problema el conjunto V se encuentran dividido en una partición V_1, \dots, V_k , y el objetivo es asignar un color a sólo un vértice de cada partición, de manera tal que dos vértices adyacentes no reciban el mismo color, minimizando la cantidad de colores utilizados.

3. Modelado del problema

4. Código

5. Experimentación

6. Conclusiones y trabajo futuro