

A fast Genetic Algorithm for the Max Cut-Clique Problem

Ing. Giovanna Fortez Hitateguy

Tutores de Tesis

Dr. Ing. Franco Robledo - Dr. Ing. Pablo Romero

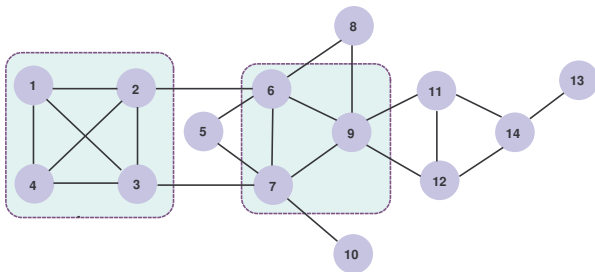
Instituto de Computación
Facultad de Ingeniería
Universidad de la República

25 de noviembre de 2019

Agenda

- 1 Motivación
- 2 Estado del arte
- 3 Problema
- 4 Solución

Motivación



Observaciones

- Densidad de la estructura de vecindad de los cliques
- Componente altamente relacionada con el resto del grafo
- Gran cantidad de cliques máximos
- Minería de datos.

Publicaciones

Publicaciones

- Martins, P.(2012). Cliques with maximum/minimum edge neighborhood and neighborhood density.
- Gouveia, L. and Martins, P.(2015). Solving the maximum edge-weight clique problem in sparse graphs with compact formulation. JOURNAL ON COMPUTATIONAL OPTIMIZATION, 3(1):1-30.
- Martins, P., Ladrón, A. and Ramalhinho, H. (2014).Maximum cut-clique problem: ILS heuristics and a data analysis application. International Transactions in Operational Research, 22(5):775-809.

Publicaciones

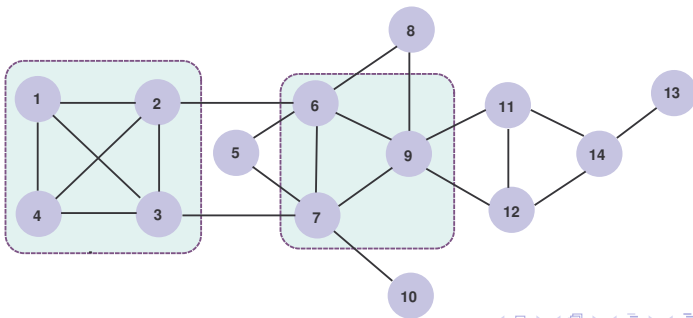
Publicaciones Locales

- Bourel, M., Canale, E., Robledo, F., Romero, P., and Stábile, L. (2018a). Complexity and heuristics for the max cut-clique problem. In International Conference on Variable Neighborhood Search, pages 28-40. Springer.
- Bourel, M., Canale, E., Robledo, F., Romero, P., and Stábile, L. (2018b). A grasp/vnd heuristic for the max cut-clique problem. In International Conference on Machine Learning, Optimization, and Data Science, pages 357-367. Springer.
- Bourel, M., Canale, E., Robledo, F., Romero, P., and Stábile, L. (2019). Complexity and Heuristics for the Weighted Max Cut-Clique Problem. International Transactions in Operational Research. Unpublished.

Problema

Máximo Clique-Corte (MCC)

Dado un grafo G , encontrar un *Clique* C donde el *Clique-Corte* generado es máximo.



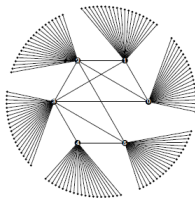
Complejidad

Proposition

The MCC belongs to the \mathcal{NP} -Complete set.

Demostración.

Reduction from *CLIQUE*.



Solución

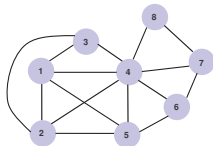
Algoritmos Genéticos

```
1 - Initialize( $P_0$ );  
2 - generation = 0;  
3 - While (notstopCriteria);  
4 -     evaluate( $P(\text{generation})$ );  
5 -     parents  $\leftarrow$  selection( $P(\text{generation})$ );  
6 -     offspring  $\leftarrow$  evolutiveOperators(parents);  
7 -     newpop  $\leftarrow$  replacement(offspring,  $P(\text{generation})$ );  
8 -     generation ++;  
9 -      $P(\text{generation}) \leftarrow$  newpop;  
Return BestSolutionEverFound;
```


Representación

La solución es una tupla binaria de largo igual a n , cantidad de nodos del grafo $\mathcal{G} = (V, E)$.

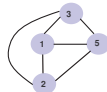
$$X_i = \begin{cases} 1 & \text{si nodo } i \in \mathcal{C}, \forall i \in V \\ 0 & \text{sino} \end{cases}$$



(0,0,1,1,0,0,1,1)



(0,0,0,1,0,1,0,1)



(1,1,1,0,1,0,0,0)

Función de Fitness

Función de Fitness

$$|E(C)| = \sum \text{grado}(i) - |C| * |C - 1|, \forall \text{ nodo } i \in C \quad (1)$$

Coincide con la función objetivo y busca maximizar la cantidad de aristas en el corte generado por el clique C

Cruzamiento

Cruzamiento de 2 puntos

Padres
Progenitores



Puntos de Corte



Individuos a cruzar

Hijos
Descendientes



Puntos de Corte



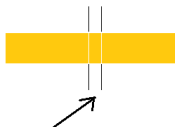
Nuevos individuos



Mutación

Mutación Simple

Individuo ANTES



Alelo a mutar

Individuo DESPUÉS

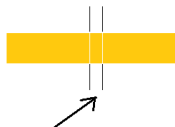


Individuo con alelo mutado

Selección

Selección por Torneo de tamaño 2

Individuo ANTES



Alelo a mutar

Individuo DESPUÉS



Individuo con alelo mutado

Inicialización y Criterio de Parada

- Inicialización de soluciones, mediante proceso aleatorio de forma de otorgar diversidad.
- Criterio de Parada, que la población evolucione durante 1000 generaciones.