

Practica de Métodos Algorítmicos de Resolución de Problemas

Pablo Vazquez Gomis
Universidad Complutense de Madrid

May 17, 2019

Abstract

Practica opcional del segundo cuatrimestre para la asignatura de Métodos Algorítmicos de Resolución de Problemas.

1 Introducción

1.1 Descripción de la práctica

Programar el problema de encontrar el clique de tamaño máximo en un grafo no dirigido Dado un grafo $G = (V, E)$, un clique es un conjunto $C \subseteq V$ tal que $\forall u, v \in C, \exists (u, v) \in E$.

El problema consiste en encontrar el clique de tamaño máximo en G , para ello diferentes estrategias pueden ser utilizadas. La más común es utilizar la técnica de ramificación y poda. Este problema ha sido estudiado por varios investigadores a lo largo de los años.

En 1990 Carraghan y Pardalos [1] asentaron las bases sobre las que otros investigadores han ido construyendo. Especialmente introdujeron el concepto del set $P(C) \subseteq V$, cuya definición informal se reduce a el conjunto de vertices que pueden formar clique con C aunque no garantiza $C + P(C)$ sea un clique. Es decir $\forall v \in P(C) : C + v$ es clique.

Las estrategias posteriores, en general, tratan de acercar lo máximo posible a la solución el set $P(V)$ para mejorar la poda. Una aproximación común es utilizar el coloreado de grafos para mejorar el upper bound como hacen Tomita y Seki [2] o J. Konc, D. Janežič [3].

1.2 Implementación

Grafo El grafo se ha implementado mediante listas de adyacencia ordenadas, para evitar duplicar la información (dado que $(u, v) = (v, u) \forall u, v \in V$) solo contiene la arista el vertice menor.

Debido a esta implementación el grafo necesita utilizar una lista ordenada, dado que los primeros vertices contienen mas informacion que los últimos.

2 El algoritmo

El algoritmo usado es el de ramificacion y poda visto en clase, con una cola de prioridad para explorar los nodos del arbol más prometedores, un test de factibilidad y dos cotas una optimista y otra pesimista para poder podar el arbol.

2.1 El arbol

El arbol calcula todas las posibilidades, es decir toda la combinación de nodos $\mathbb{P}(V)$ de menor a mayor tamaño, es decir, en el primer nivel calculamos los sets $C = \{v\} \forall v \in V$. En cada nivel generamos los hijos de cada nodo mediante el set $P(V)$ el cual nos garantiza que todos los hijos son solución al problema.

Debido a que solo el nodo menor contiene la arista en su lista de adyacencia, se garantiza que ningún set esta repetido en otras ramas.

2.2 Test de factibilidad

El test de factibilidad es sencillo, con comprobar que C es solución basta, dado que si C es solucion cualquier $C' \subseteq C$ es solución.

Para comprobar que un set es solución solo tenemos que hacer una sencilla operación matematica

$$|E'| = \frac{|C| * (|C| - 1)}{2}$$

2.3 Cotat

2.3.1 Simple

Optimista. La cota optimista simple es sencilla de ver utilizando la notación de Carraghan y Pardalos, dado que para cualquier posible solución C optimisticamente todos los elementos en $P(V)$ formaran la solución final. Es decir:

$$\text{cota-optimista}(C) = |C| + |P(C)|$$

Pesimista. En este problema es imposible comprobar si explorar los nodos de $P(V)$ si pueden o no formar parte del cliqué maximo por lo que la cota pesimista ha de ser

$$\text{cota-pesimista}(C) = |C|$$

2.3.2 Compleja

Optimista. La cota optimista compleja se aprovecha de la tecnica del coloreado de vertices presentada por Tomita y Seki [2] de forma simplificada, damos a un color diferente a todos aquellos vertices $u, v \in P(C)$ tal que $(u, v) \in E'$. Podemos utilizar esto para delimitar mejor la cota optimista de la siguiente forma:

$$\text{cota-optimista}(C) = |C| + S$$

Donde S es el numero de colores diferentes.

Pesimista. La cota pesimista sigue siendo igual que en la Simple por la misma razón.

3 Resultados Experimentales

3.1 Grafos Aleatorios

Todas las mediciones se han obtenido aleatoriamente con una semilla ”-s 1” para poder replicar los resultados

Tamaño	Densidad	Sin Poda	Nodos	Sec/Nodo	Poda Simple	Nodos	Sec/Nodo	Poda Compleja	Nodos	Sec/Nodo
10	0.1	0.0027	11	0.0001	0.0028	2	0.0006	0.0019	1	0.0004
10	0.3	0.0927	102	0.0007	0.0462	13	0.0031	0.0334	7	0.0040
10	0.5	0.3108	795	0.0003	0.0968	39	0.0022	0.0282	17	0.0014
10	0.7	65.574	139373	0.0003	3.4457	1958	0.0016	0.0808	35	0.0021
10	0.9	123.5	235274	0.0004	18.874	13701	0.0012	0.1055	36	0.0026
20	0.1	0.0529	61	0.0006	0.0409	7	0.0036	0.0303	4	0.0037
20	0.3	2.0066	2269	0.0008	0.5331	127	0.0040	0.1804	21	0.0073
20	0.5	-	-	-	-	-	-	18.679	1779	0.0103
50	0.1	0.7174	536	0.0011	0.4065	57	0.0055	0.1860	12	0.0079
50	0.3	-	-	-	152.62	7612	0.0199	5.6021	187	0.0290
100	0.1	17.0103	4691	0.0035	7.3379	349	0.0202	1.0722	16	0.0519
100	0.2	-	-	-	-	-	-	14.185	175	0.0787

4 Conclusión

Es evidente la progresión exponencial del algoritmo, aun asi la aplicacion de cotas aun más costosas por nodo limita el numero de nodos y permite solucionar

grafos más grandes y densos.

El problema del cliqué es un problema estudiado en profundidad como cualquier otro problema NP-Completo. Aun habiendo echo grandes avances en el campo que permiten en un tiempo mas o menos razonable gestionar grafos con muchas aristas, aun queda espacio por recorrer dado que la curva exponencial de su complejidad sigue limitando la resolución del problema incluso para ordenadores modernos.

References

- [1] Carraghan, R. & Pardalos, P.M. (1990). *An exact algorithm for the maximum clique problem*. Operations Research Letters, 9(6), 375–382.
- [2] E. Tomita, T. Seki, *An efficient branch-and-bound algorithm for finding a maximum clique*, Lecture Notes in Computer Science 2631 (2003) 278- 289.
- [3] J. Konc & D. Janežič, D. (2007). *An improved branch and bound algorithm for the maximum clique problem*. MATCH - Communications in Mathematical and in Computer Chemistry, 58, 569-590.