Universidad de Chile
Departamento de Ciencias de la Computación
CC4102 - Diseño y Análisis de Algoritmos

Tarea 3 - Cubrimiento por Vértices Mínimo

Profesor: Pablo Barceló
Auxiliar: Manuel Cáceres
Ayudantes: Jaime Salas
Claudio Torres

1. Introducción

El problema de encontrar el **Cubrimiento por Vértices Mínimo** (CVM), consiste en, dado un grafo simple[1] y no dirigido[2] $G = (V, E)$ (donde V es el conjunto de vértices del grafo, con $|V| = n$, y E es el conjunto de aristas, con $|E| = m$), encontrar el subconjunto de vértices más pequeño $S^* \subseteq V$ que "cubre" E, es decir:

$$S^* = argmin\{|S|: \quad S \subseteq V \wedge \ \forall (u, v) \in E, \quad (u \in S \vee v \in S)\}$$

En clases de cátedras vimos una 2-aproximación para el problema, que consistía en recorrer las aristas del grafo y agregar al cubrimiento **ambos** vértices extremos, eliminando todas las aristas adyacentes a estos vértices e iterando el proceso hasta que ya no quedarán más aristas que cubrir. Además en clase auxiliar vimos que la *heurística* de "ir tomando el vértice de mayor grado", no es una ρ-aproximación para cualquier ρ constante.

El objetivo de esta tarea es implementar, evaluar y comparar en la práctica tres enfoques para encontrar una solución aproximada del CVM, los dos mencionados anteriormente, y un tercero que utiliza ambas ideas.

Se espera que se implementen los algoritmos y se entregue un informe que indique claramente los siguientes puntos:

1. Las *hipótesis* escogidas antes de realizar los experimentos.
2. El *diseño experimental*, incluyendo los detalles de la implementación de los algoritmos, la generación de las instancias y las medidas de rendimiento utilizadas.
3. La *presentación de los resultados* en forma de una descripción textual, tablas y/o gráficos.
4. El *análisis e interpretación* de los resultados.

[1] Un grafo *simple* es aquel que no tiene bucles ni multiaristas.
[2] Un grafo *no dirigido* es aquel en el que sus aristas no poseen dirección.

2. Los Algoritmos

A continuación se describirá cada uno de los tres algoritmos que se deben implementar y comparar en esta tarea. Los algoritmos están descritos en pseudocódigo, los detalles de su implementación tanto como las estructuras utilizadas deben ser escogidas apropiadamente y especificadas en su informe.

2-aproximación

La idea general detrás de este algoritmo consiste en recorrer las aristas del grafo en algún orden no especificado y tomar sus extremos como parte del cubrimiento, eliminando del proceso que continúa, todas las aristas incidentes a alguno de estos vértices.

El pseudocódigo del algoritmo es el siguiente:

Input: $G = (V, E)$
$E' \leftarrow E$;
$C \leftarrow \emptyset$;
while $E' \neq \emptyset$ **do**
  Tomar $(u, v) \in E'$;
  $C \leftarrow C \cup \{u, v\}$;
  $E' \leftarrow E' \setminus \{e \in E' : u \text{ es incidente a } e \text{ o } v \text{ lo es}\}$;
end
return C

Algoritmo 1: Entrega una 2-aproximación al problema de Cubrimiento por Vértices Mínimo, es decir, $|C| \leq 2|S^*|$, donde S^* es la solución del problema.

Heurística del grado mayor

La idea detrás de esta heurística es ir escogiendo los vértices que cubran la mayor cantidad de aristas, es decir, los que tengan mayor grado y hacer esto hasta que no queden más aristas por cubrir.

El pseudocódigo del algoritmo es el siguiente:

Input: $G = (V, E)$
$E' \leftarrow E$;
$C \leftarrow \emptyset$;
while $E' \neq \emptyset$ **do**
  Tomar $u \in (V \setminus C)$ de mayor grado en $G' = (V \setminus C, E')$;
  $C \leftarrow C \cup \{u\}$;
  $E' \leftarrow E' \setminus \{e \in E' : u \text{ es incidente a } e\}$;
end
return C

Algoritmo 2: No entrega una ρ-aproximación al problema de Cubrimiento por Vértices Mínimo, para ninguna constante ρ.

2-aproximación mejorada

Finalmente, la idea detrás de este último algoritmo consiste en ir añadiendo pares de vértices en una arista, para así mantener la propiedad de ser una 2-aproximación, pero al mismo tiempo elegir estos vértices según el criterio del grado máximo.

El pseudocódigo del algoritmo es el siguiente:

Input: $G = (V, E)$
$E' \leftarrow E$;
$C \leftarrow \emptyset$;
while $E' \neq \emptyset$ **do**
  Tomar $u \in (V \setminus C)$ de mayor grado en $G' = (V \setminus C, E')$;
  Tomar $v \in N(u) = \{x \in V \setminus C : x \text{ es adyacente a } u \text{ en } G'\}$ de mayor grado en G';
  $C \leftarrow C \cup \{u, v\}$;
  $E' \leftarrow E' \setminus \{e \in E' : u \text{ es incidente a } e \text{ o } v \text{ lo es}\}$;
end
return C

Algoritmo 3: Este algoritmo es una 2-aproximación del problema CVM que impone un orden en el que se escogen las aristas respecto al primer algoritmo presentado, según el criterio del grado máximo.

Nota

En los algoritmos en donde se escoge un vértice de mayor grado, si hay más de uno se elige cualquiera de ellos.

3. Pruebas y Datos

En esta tarea se pide comparar el tamaño de los resultados dados por las tres aproximaciones antes explicadas. Para esto tendrá que correr todas las implementaciones en grafos de distintos tamaños y orígenes explicados más adelante.
Recuerde repetir los experimentos para los distintos tamaños y orígenes de manera de obtener promedios confiables, tampoco olvide documentar los intervalos de confianza de sus resultados.

Grafos Aleatorios

Construya grafos de $n = 2^i$ con $i \in \{10, 11, \ldots, 20\}$ vértices y genere las aristas según el siguiente proceso probabilístico:

Input: (V, p)
$E \leftarrow \emptyset$;
for $(u, v) \in \binom{V}{2}$ **do**
  Con probabilidad p hacer: $E \leftarrow E \cup \{(u, v)\}$;
end
return $G = (V, E)$

Escoja al menos 5 valores de p distribuidos en $[1/n, 1]$, escoja los valores de p de modo de poner a prueba sus hipótesis.

Grafos Externos

Utilice grafos externos, es decir, grafos obtenidos de una fuente externa de datos (no construidos por usted). Utilice al menos 10 grafos distintos. En la sección 5 puede encontrar algunos orígenes de datos, preprocéselos si es necesario para obtener resultados significativos, indique esto procedimientos en su informe. Indique los tamaños de los grafos con los que se trabajó y su origen. Especifique también si estos cumplen con alguna propiedad en particular. Base la discusión de sus resultados en estos parámetros, de ser posible escoja los grafos de manera de poner a prueba sus hipótesis.

4. Entrega de la Tarea

- La tarea puede realizarse en grupos de a lo más 2 personas.
- No se permiten atrasos.
- Para la implementación puede utilizar C, C++ o Java. Para el informe se recomienda utilizar LaTeX.
- Siga buenas prácticas (good coding practices) en sus implementaciones.
- Escriba un informe claro y conciso. Las ponderaciones del informe y la implementación en su nota final son las mismas.
- Tenga en cuenta las sugerencias realizadas en la primera clase auxiliar y en los archivos subidos a U-Cursos sobre la forma de realizar y presentar experimentos.
- La entrega será a través de U-Cursos y deberá incluir el informe junto con el código fuente de la implementación (y todas las indicaciones necesarias para su ejecución).

5. Links

Los siguientes links son fuentes de datos externas que puede utilizar en su tarea:

- https://users.dcc.uchile.cl/~jfuentess/datasets/graphs.php
- http://www.info.univ-angers.fr/pub/porumbel/graphs/
- http://mat.gsia.cmu.edu/COLOR/instances.html
- https://turing.cs.hbg.psu.edu/benchmarks/clique.html