Tarea 2: Minimum Set Cover Problem
INFO145 - Diseño y Análisis de Algoritmos.

Académico: Héctor Ferrada.
Auxiliar: José Vásquez
Instituto de Informática, Universidad Austral de Chile.

6 de junio de 2022

Entrega. Debe subir todo a siveduc en un archivo, t2Apellidos.zip (3 o 4 integrantes por grupo), con su implementación (incluya todo lo necesario para su ejecución, como Makefile en el caso de C++) y su informe. Fecha de entrega: **Lunes 27 de Junio, 2022**.

Informe. El informe debe ser claro, objetivo y detallado, este debe contemplar: resumen, introducción, metodología —la cual debe incluir un análisis teórico y las hipótesis de investigación—, experimentación y conclusiones. Se debe introducir adecuadamente el contexto del problema, explicar la lógica que usó en sus implementaciones y la justificación de ello (p. ej. explicar por qué no uso otra alternativa más eficiente si la hubiese), detallar la experimentación realizada, incluir gráficas de sus resultados y dar buenas conclusiones de su trabajo; con todo esto debe validar o rechazar sus hipótesis iniciales justificando detallada y objetivamente.

Minimum Set Cover Problem (MSCP)

El objetivo de este trabajo es adquirir un mayor dominio en el análisis de algoritmos y de estructuras de datos para resolver un mismo problema. Se evalúa que usted pueda llevar a la práctica la teoría vista en clases, aplicando técnicas de diseño de algoritmos a fin de implementar una solución correcta y eficiente que permita encontrar el set-covering para la instancia del problema, tanto de forma óptima como de manera aproximada.

Definición del Problema

Dado un universo $\mathcal{X}$ y una familia de conjuntos $\mathcal{F} = \{S_1, S_2, ..., S_m\}$, tal que: $\mathcal{X} = \bigcup_{i=1}^{m} S_i$, $S_i \in \mathcal{F}$; se desea determinar la cantidad mínima de conjuntos de $\mathcal{F}$ cuya unión forme el universo $\mathcal{X}$.
En base a esto, tenemos:

- El conjunto $\mathcal{C} = \{S'_1, \ldots, S'_k\}$ con $S'_j \in \mathcal{F}, 1 \leq j \leq k$ es un set-cover (SC) para la instancia $(\mathcal{X}, \mathcal{F})$, si y solo si $\mathcal{X} = \bigcup_{S' \in \mathcal{C}} S'$.
- La solución óptima al MSCP corresponde al SC de $(\mathcal{X}, \mathcal{F})$ de tamaño mínimo.

La siguiente imagen ilustra el problema, donde vemos que la solución óptima al MSC es $\mathcal{C}^* = \{S_3, S_4, S_5\}$ de tamaño 3.

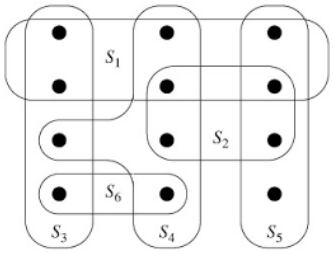

Figura 1: Ilustración del MSCP, donde el MSC óptimo es $\mathcal{C}^* = \{S_3, S_4, S_5\}$ de tamaño 3

Diseño de Soluciones

En esta tarea, se debe trabajar en 4 soluciones al problema del MSCP, donde dos de ellas son para determinar la solución óptima al problema y las otras dos son aproximaciones al mismo. Para cada solución, proponga un algoritmo que determine cual es la solución al problema, no solo el tamaño de la solución sino también la composición del conjunto solución. Para cada propuesta —omita en este paso la propuesta 3, ya que el algoritmo greedy esta dado y analizado en la bibliografía— entregue el pseudocódigo de cada rutina de búsqueda y realice un analizis asintótico mediante notación O.

Solución 1. MSC Óptimo por medio de Búsqueda Exhaustiva

Diseñe un algoritmo que mediante la búsqueda exhaustiva, la cual debe testear todas las combinaciones posibles y sin podar la búsqueda, determine una solución óptima al problema.

sin terminar cuando encuentre el optimo

Solución 2. MSC Óptimo por medio de una Búsqueda Optimizada

Diseñe un algoritmo que mediante una búsqueda exhaustiva, pero optimizada, determine una solución óptima al problema. Las optimizaciones que se piden son las siguientes:

1. Note que en el ejemplo de la Firuga **??**, el conjunto S_5 debe estar siempre en tolos los SC de la instancia, ya que este conjunto es el único que incluye a un elemento que no forma parte de ningún otro subconjunto. Por tanto, en primera instancia, es posible incluir a todos los conjuntos que contienen elementos que ningún otro conjunto incluye —como S_3 en el ejemplo; lo cual, le debe ayudar a aceletar la búsqueda exhaustiva de todas las combinaciones de conjuntos que incluyen a estos conjuntos con elementos úicos.

2. Durante la búsqueda exhaustiva que este implementando, encontrará quizá a muchos SC candidatos a ser MSC para la instancia $(\mathcal{X}, \mathcal{F})$. Observe que cada vez que aparece un nuevo SC que es mejor que todos los anteriormente encontrados, a partir de ese momento es posible detener la búsqueda por esa rama en particular, si es que ya no es posible encontrar una mejor solución por dicha rama. Esto ocurre, si por ejemplo, el mejor SC encontrado es de tamaño δ, y en esta rama del árbol de búsqueda ya se han incluido a δ subconjuntos de $\mathcal{F}$ y aún no se cubre todo $\mathcal{X}$, entonces ya no tiene sentido contunuar buscando un SC por este camino; por tanto puede terminar la búsqueda por esta rama y saltar a la siguiente. Este valor δ mínimo durante el algoritmo, lo debe ir actualizando cada vez que encuentr un SC de tamaño menor al δ actual.

3. Si se le ocurre otra heurística para podar el árbol de búsqueda y que continúe asegurando que su algoritmo determina el MSC óptimo, puede incluirla aquí.

Solución 3. MSC Aproximado por medio del algoritmo greedy clásico

Esta corresponde al algoritmo greedy estudiado en clases y que se encuentra en la bibliografía. Por tanto no es necesario incluir el detalle del análisis de este algoritmo.

Solución 4. MSC Aproximado por medio de un algoritmo greedy optimizado

Diseñe un algoritmo que mediante la estrategia greedy clasica, pero optimizada, determine una solución aproximada al problema. Las optimizaciones que se piden son las siguientes:

1. Aplique la misma optimización de incluir a todos los conjuntos que contienen elementos que ningún otro conjunto incluye, que se describió en la solución 1.

2. La estrategia greedy clásica selecciona a un conjunto por iteración. Se pide que gereralice esta técnica incluyendo a k subconjuntos por cada iteración. Es decir, dado un valor k —en la experimentación probará con $k \in \{1, 2, \ldots, \lg(min\{n, m\})\}$— se debe seleccionar a los k conjuntos de $\mathcal{F}$ cuya union aporte con la mayor cantidad de nuevos elementos a la solución. Note que la estrategia greedy clásica corresponde al caso $k = 1$.

3. Si se le ocurre otra heurística que crea ser buena, dando la justificación de ello, puede incluirla aquí.

Se pide que acompañe a los pseudocódigos los análisis del tiempo asintótico de cada algoritmo; y para la solución 4, debe incluir un análisis de la calidad de la solución que entrega su algoritmo.

Experimentación

Las cuatro soluciones deberán ser testeadas, graficando sus resultados adecuadamente, con datos de entrada correspondientesx a los siguientes escenarios:

1. **Normal uniforme aleatoria**. Proponga los tamaños para la instancia $(\mathcal{X}, \mathcal{F})$, además de los valores para la media y la desviación estandar que considere relevantes para la experimentación y así inferir buenas conclusiones. Proponga los límites mínimos y máximos tanto para la cantidad de conjunto en $\mathcal{F}$, asi como para el tamaño de daca conjunto S_i. El universo $\mathcal{X}$ lo formará por defecto a unir todos los conjuntos en $\mathcal{F}$.

2. **Datos reales**. Investigue en el estado del arte sobre alguna colección, disponible para su descarga, de datasets reales para el problema del MSCP.

Para los casos aleatorios, asegúrese de tener datos representativos, es decir no pueden ser valores pequeños ni para n ni para m. Puede considerar que un $p\,\%$ de los conjuntos S en $\mathcal{F}$, contengan al menos un elemento que no este en ningún otro subconjunto y así la primera optimización de las soluciones 1 y 4 tenga sentido. Trabaje con un p adecuado y realista para el problema, por ejemplo $p = 5\,\%$.
Además, las soluciones óptimas (1 y 2) de seguro tardarán mucho tiempo en determinar el MSC para instancias grandes de $(\mathcal{X}, \mathcal{F})$; por tanto, detenga la ejecución de los experimentos luego de un tiempo razonable —alrededor de un par de horas.

Para los tres escenarios anteriores realice experimentos para estimar:

1. **Tiempo de búsqueda del MSC**. Obtenga el tiempo promedio de CPU que tarda cada solución en encontrar el MSC para la instancia $(\mathcal{X}, \mathcal{F})$.

2. **Tamaño de la solución**. Corresponde al tamaño del SC encontrado en cada solución, obviamente asegúrese que las dos primeras soluciones arrojan resultados idénticos en el tamaño óptimo del MSC. Desde luego los resultados para las soluciones 3 y 4 deben ser igual o de mayor tamaño al óptimo.

Grafique adecuadamente los resultados de sus experimentos para luego **analizar y concluir sobre los eventos.** Comente también sobre la forma y las condiciones en las cuales llevó a cabo la experimentación, incluyendo detalles que permitan reproducir sus resultados.

Evaluación

Su trabajo será evaluado con dos notas:

- **Nota de Informe**. Una nota por el informe que pondera un 60 % del trabajo. En esta se evaluará el trabajo teórico y el análisis que ha realizado. Además de la estructura, presentación y calidad de su escrito; la cual además incluye la presentación de los resultados obtenidos y sus conclusiones.
- **Nota de Implementación**. Una nota por las 4 implementaciones pedidas que pondera un 40 % del trabajo. No solo se espera que entregue códigos correctos, sino que además esten ordenados, sean modulares y que esten debidamente documentados. Se pide que su código sea implementado en C++ con el mayor grado de optimización (basta con el flag -O3 eb su Makefile). Por favor incluya un archivo README.txt con las instrucciones de como se ejecuta su código.

Ante cualquier duda con el trabajo, puede escribir un correo o contactar por Slack al ayudante de la asignatura, José Luis Vásquez {jose.vasquez01@alumnos.uach.cl}.