
OPTIMIZACIÓN

Primer Cuatrimestre 2025

Entrega N°4

El objetivo de este trabajo práctico es comparar dos estrategias de resolución para el problema de **Set Cover**: una heurística greedy y una formulación de programación entera (utilizando JuMP en Julia). Se busca analizar el desempeño y la calidad de las soluciones obtenidas por ambas metodologías en diferentes instancias del problema.

Problema

Dado un conjunto universal $U = \{1, 2, \dots, n\}$ y una colección $\mathcal{S} = \{S_1, S_2, \dots, S_m\}$ de subconjuntos de U , cada uno con un costo asociado c_i , el objetivo es seleccionar un subconjunto de \mathcal{S} tal que su unión cubra todo U , minimizando el costo total.

Ejercicio 1 Implementar una función `generar_conjuntos` que genere un conjunto U con n elementos y m subconjuntos, con coberturas aleatorias y costos asociados.

Ejercicio 2 Implementar una función `greedy_set_cover` que resuelva el problema a través de una heurística Greedy que seleccione los conjuntos en función del costo por elemento cubierto.

Ejercicio 3 Formular e implementar el problema de Set Cover como un problema de programación entera, y utilizar el paquete JuMP con el optimizador GLPK para resolverlo.

Ejercicio 4 Resolver el problema utilizando ambas aproximaciones (Greedy y JuMP) para distintos n y m .

Ejercicio 5 Comparar:

- (a) Costo total de la solución.
- (b) Número de conjuntos seleccionados.
- (c) Tiempo de ejecución de cada algoritmo utilizando el macro `@time`. ¿Qué se observa para tamaños grandes?
- (c) Verificar la cota de aproximación en cada caso.