

Algoritmos para la aproximación de un conjunto a partir de otros. Caracterización matemática del problema y estudio experimental

Laura Lázaro Soraluce

Doble Grado en Ingeniería Informática y Matemáticas
Tutorizado por: Nicolás Marín Ruiz y Daniel Sánchez Fernández



**UNIVERSIDAD
DE GRANADA**

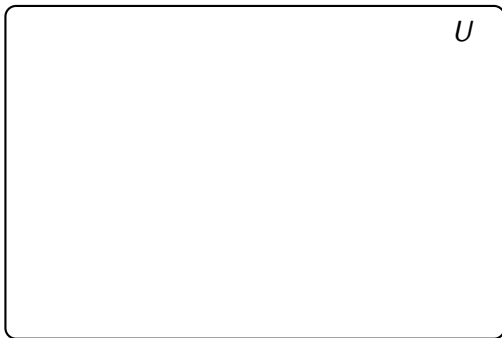
Tabla de Contenido

- 1 Introducción
- 2 Marco teórico
- 3 Aproximaciones algorítmicas
- 4 Experimentos
- 5 Conclusiones

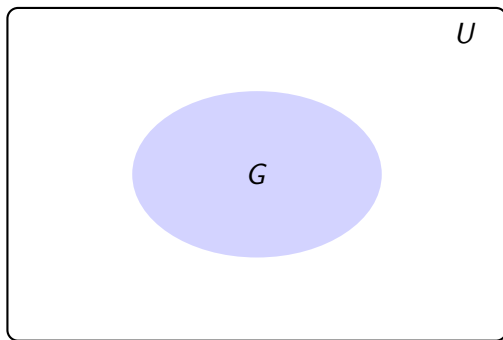
Contenido de Introducción

- 1 Introducción
 - Definición formal
 - Objetivos del trabajo
 - Aplicaciones y particularizaciones
- 2 Marco teórico
- 3 Aproximaciones algorítmicas
- 4 Experimentos
- 5 Conclusiones

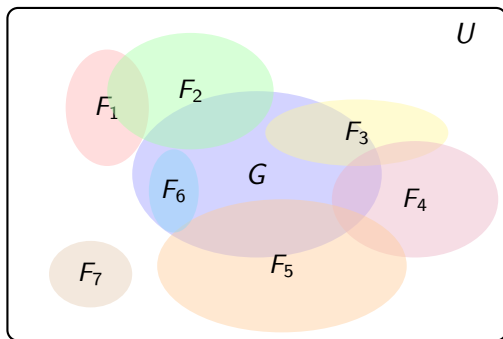
Elementos del problema



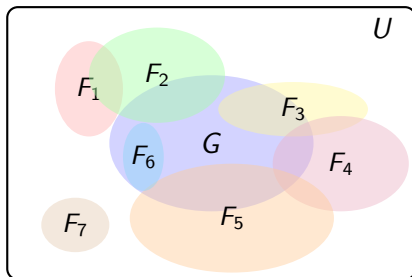
Elementos del problema



Elementos del problema



Elementos del problema



Operadores disponibles:

\cup, \cap, \setminus

Restricciones \mathcal{C}

Medidas \mathcal{M}

Definición formal

Dado un universo U , un subconjunto $G \subseteq U$ y una familia de subconjuntos $F \subseteq \mathcal{P}(U)$, consideramos **el espacio de expresiones** \mathcal{E}_F obtenido mediante subconjuntos de F y las operaciones \cup , \cap y \setminus .

Sea \mathcal{C} un conjunto de restricciones y \mathcal{M} un conjunto de medidas, de las cuales al menos una evalúa la relación entre la expresión y el conjunto objetivo G .

El problema consiste en determinar una expresión

$$e^* \in (\mathcal{E}_F^{\mathcal{C}})^{\mathcal{M}}$$

es decir, una expresión que cumpla las restricciones y que pertenezca al frente de Pareto de \mathcal{M} .

Objetivos del trabajo

Objetivos teórico–matemáticos

- Formalizar el problema y sus elementos (universo, familia F , expresiones, restricciones).
- Estudiar las estructuras algebraicas implicadas (álgebra de Boole, retículos, anillos/semianillos de conjuntos, etc.).
- Definir y estudiar distintas medidas.
- Analizar su complejidad y la relación con algunos problemas clásicos (Set Cover, Exact Cover, etc.).

Objetivos práctico–computacionales

- Diseñar e implementar tres aproximaciones: Exhaustiva (con profundidad limitada), Greedy-MO y NSGA-II.
- Construir un entorno experimental reproducible y extensible.
- Ilustrar el comportamiento de los algoritmos en distintos escenarios.

Aplicación: marketing

En un mercado con miles de clientes potenciales (U), una empresa quiere dirigirse a un público objetivo muy concreto (G).
Cada canal publicitario cubre a un subconjunto distinto de personas:

- Revista $X \rightarrow F_1$
- Canal de TV $Y \rightarrow F_2$
- Red social $Z \rightarrow F_3$

El problema consiste en combinar estos subconjuntos para **cubrir lo mejor posible al público objetivo.**

Aplicación: marketing

En un mercado con miles de clientes potenciales (U), una empresa quiere dirigirse a un público objetivo muy concreto (G).
Cada canal publicitario cubre a un subconjunto distinto de personas:

- Revista $X \rightarrow F_1$
- Canal de TV $Y \rightarrow F_2$
- Red social $Z \rightarrow F_3$

El problema consiste en combinar estos subconjuntos para **cubrir lo mejor posible al público objetivo**.

Particularización a problemas clásicos

Nuestro Problema

- **Objetivo:** Aproximar un objetivo $G \subseteq U$.
- **Operaciones:** $\{ \cup, \cap, \setminus \}$.
- **Criterio:** multiobjetivo, cualesquiera medidas (donde al menos una tiene que ver con G).

Particularización a problemas clásicos

Nuestro Problema

- **Objetivo:** Aproximar un objetivo $G \subseteq U$.
- **Operaciones:** $\{\cup, \cap, \setminus\}$.
- **Criterio:** multiobjetivo, cualesquiera medidas (donde al menos una tiene que ver con G).



Caso Particular A: Set Cover

Restricciones:

- 1 Solo se permite la unión (\cup).
- 2 Restricción dura: $G \subseteq \bigcup F_i$.

Medidas: número de conjuntos F_i utilizados (a minimizar).

Particularización a problemas clásicos

Nuestro Problema

- **Objetivo:** Aproximar un objetivo $G \subseteq U$.
- **Operaciones:** $\{\cup, \cap, \setminus\}$.
- **Criterio:** multiobjetivo, cualesquiera medidas (donde al menos una tiene que ver con G).



Caso Particular A: Set Cover

Restricciones:

- 1 Solo se permite la unión (\cup).
- 2 Restricción dura: $G \subseteq \bigcup F_i$.

Medidas: número de conjuntos F_i utilizados (a minimizar).

Caso Particular B: Exact Cover

Restricciones:

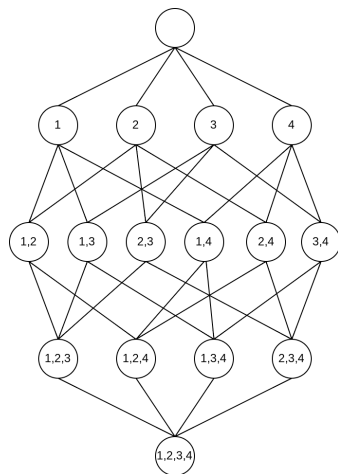
- 1 Las mismas que Set Cover.
- 2 Intersección nula ($F_i \cap F_j = \emptyset$).

Medidas: número de conjuntos F_i utilizados (a minimizar).

Contenido de Marco teórico

- 1 Introducción
- 2 Marco teórico
 - Estructuras algebraicas
 - Medidas
 - Complejidad
- 3 Aproximaciones algorítmicas
- 4 Experimentos
- 5 Conclusiones

Estructura de $\mathcal{P}(U)$



Conexión con Lógica

El isomorfismo con el álgebra de Boole ($\cup \leftrightarrow \vee, \cap \leftrightarrow \wedge, \setminus \leftrightarrow \wedge \neg$) permite trazar paralelismos con la **Satisfacibilidad Booleana**:

- Útil al estudiar la complejidad del problema.
- Permite conectar con uno de los problemas clásicos en computación.

Diagrama de Hasse de $\mathcal{P}(U)$ donde $U = \{1, 2, 3, 4\}$

Estructuras algebraicas de los elementos

Relación estructural entre F y G

- Recubrimiento, partición.

Estructuras algebraicas de los elementos

Relación estructural entre F y G

- Recubrimiento, partición.

Estructuras inducidas al cerrar F bajo ciertas operaciones

- Seminillo, anillo, álgebra de conjuntos.

Medidas

Las medidas cumplen dos papeles principales en el problema:

- **Definir restricciones** sobre las expresiones permitidas.
- **Evaluar la calidad** de una expresión e , para poder comparar soluciones y construir frentes de Pareto.

Medidas

Agrupamos las medidas estudiadas de la siguiente manera:

- **Medidas de asociación:** cuantifican la relación entre e y G sin priorizar un conjunto sobre otro.
- **Medidas direccionales:** cuantifican la relación entre e y G utilizando uno como predicción y el otro como realidad.
- **Otras medidas.**

Medidas

En la práctica utilizamos:

- Índice de Jaccard:

$$M(e) = \frac{|G \cap eval(e)|}{|G \cup eval(e)|}.$$

- Número de subconjuntos F_i distintos utilizados.
- Número de operaciones utilizadas.

Complejidad del problema

- 1 Variantes clásicas como *Set Cover* y *Exact Cover* se obtienen como particularizaciones de nuestro problema.

Si un problema A contiene como caso particular un problema NP -Completo B , entonces A es NP -duro.

- 2 Para verificar una expresión: calcular el subconjunto que forma $+$ evaluarla en \mathcal{M} .

Complejidad del problema

- 1 Variantes clásicas como *Set Cover* y *Exact Cover* se obtienen como particularizaciones de nuestro problema.

Si un problema A contiene como caso particular un problema NP -Completo B , entonces A es NP -duro.

- 2 Para verificar una expresión: calcular el subconjunto que forma $+$ evaluarla en \mathcal{M} .

\implies nuestro problema es NP -Completo con medidas polinómicas. Como además el **espacio de búsqueda es infinito**, no buscamos una solución exacta, sino buenas aproximaciones en un tiempo razonable.

Contenido de Aproximaciones algorítmicas

- 1 Introducción
- 2 Marco teórico
- 3 Aproximaciones algorítmicas
 - Búsqueda exhaustiva con profundidad limitada
 - Greedy-MO
 - Algoritmo genético
- 4 Experimentos
- 5 Conclusiones

Búsqueda exhaustiva con profundidad limitada

- Genera todas las expresiones posibles hasta un máximo de k operaciones.
- Evalúa cada expresión con las tres medidas (índice de Jaccard, número de subconjuntos distintos, número de operaciones).
- Obtiene el frente de Pareto **exacto** para el espacio explorado.

Limitación

Su coste crece de forma **exponencial** con k , por lo que solo es viable para instancias muy pequeñas.

Búsqueda exhaustiva con profundidad limitada

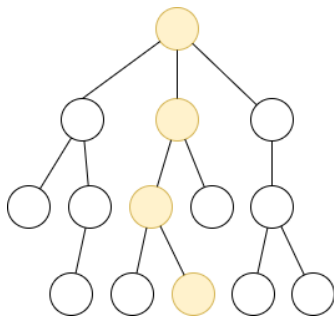
- Genera todas las expresiones posibles hasta un máximo de k operaciones.
- Evalúa cada expresión con las tres medidas (índice de Jaccard, número de subconjuntos distintos, número de operaciones).
- Obtiene el frente de Pareto **exacto** para el espacio explorado.

Limitación

Su coste crece de forma **exponencial** con k , por lo que solo es viable para instancias muy pequeñas.

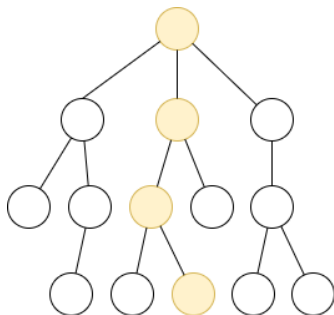
Greedy-MO

- Construcción incremental por niveles.
- En cada nivel $1 \leq i \leq k$:
 - Se parte del frente del nivel anterior ($i - 1$).
 - Se añade un par (op, F_j) .
 - Mantenemos las **no dominadas**.



Greedy-MO

- Construcción incremental por niveles.
- En cada nivel $1 \leq i \leq k$:
 - Se parte del frente del nivel anterior ($i - 1$).
 - Se añade un par (op, F_j) .
 - Mantenemos las **no dominadas**.



Limitación

Tiende a quedar estancado en **óptimos locales**, donde el camino parecía inicialmente prometedor, pero no acaba siendo el mejor.

Algoritmo genético (NSGA-II)

- Metaheurística **multiobjetivo** basada en una **población** de expresiones.
- Cada individuo codifica una expresión, su evaluación en las tres medidas, y otros dos valores necesarios para el algoritmo: distancia de aglomeración y rango de Pareto.
- En cada generación se aplican de forma estocástica los siguientes operadores:
 - **selección por torneo**,
 - operadores de **cruce** y **mutación** para generar nuevas expresiones,
 - **elitismo**: se combinan padres e hijos y se conservan los mejores.
- NSGA-II aproxima el **frente de Pareto** buscando buen equilibrio entre convergencia y diversidad de soluciones.

Contenido de Experimentos

- 1 Introducción
- 2 Marco teórico
- 3 Aproximaciones algorítmicas
- 4 Experimentos**
 - Diseño experimental
 - Resultados
- 5 Conclusiones

Hipótesis de trabajo

- **Exhaustiva:** alcanza el frente óptimo real
- **NSGA-II:** aproxima bien ese frente bajo tiempos razonables
- **Greedy-MO:** muy rápido, pero se queda en óptimos locales

Hipótesis de trabajo

- **Exhaustiva:** alcanza el frente óptimo real
- **NSGA-II:** aproxima bien ese frente bajo tiempos razonables
- **Greedy-MO:** muy rápido, pero se queda en óptimos locales

Hipótesis de trabajo

- **Exhaustiva**: alcanza el frente óptimo real
- **NSGA-II**: aproxima bien ese frente bajo tiempos razonables
- **Greedy-MO**: muy rápido, pero se queda en óptimos locales

Variables consideradas

Controlamos:

- Algoritmo (Exhaustiva / Greedy-MO / Genético)
- Tamaño de todos los conjuntos
- Profundidad máxima k
- Semilla aleatoria

Medimos:

- Índice de Jaccard
- Número de subconjuntos F_i distintos usados
- Número de operaciones en la expresión
- Número de soluciones en el frente de Pareto
- Tiempo de ejecución en ms

Variables consideradas

Controlamos:

- Algoritmo (Exhaustiva / Greedy-MO / Genético)
- Tamaño de todos los conjuntos
- Profundidad máxima k
- Semilla aleatoria

Medimos:

- Índice de Jaccard
- Número de subconjuntos F_i distintos usados
- Número de operaciones en la expresión
- Número de soluciones en el frente de Pareto
- Tiempo de ejecución en ms

Diseño experimental

● Experimento 1 – Instancias pequeñas

- Compara: Exhaustiva vs Greedy-MO vs NSGA-II con $k = 3$.
- Objetivo: ilustrar el comportamiento de las heurísticas comparándolo con el de la búsqueda exhaustiva.

● Experimento 2 – Óptimo conocido (Jaccard = 1)

- Generamos instancias donde sabemos que existe una solución óptima en Jaccard.
- Objetivo: ver si Greedy-MO y NSGA-II son capaces de alcanzar ese óptimo, y analizar los valores en las demás medidas.

● Experimento 3 – Instancias más grandes

- 50 instancias y expresiones más profundas ($k = 10$).
- Objetivo: comparar Greedy-MO y NSGA-II en un escenario más realista, y evaluar calidad de soluciones, estabilidad y tiempos de ejecución.

Configuración común

Universo fijo $|U| = 128$; operadores $\{\cup, \cap, \setminus\}$.

Diseño experimental

- **Experimento 1 – Instancias pequeñas**

- Compara: Exhaustiva vs Greedy-MO vs NSGA-II con $k = 3$.
- Objetivo: ilustrar el comportamiento de las heurísticas comparándolo con el de la búsqueda exhaustiva.

- **Experimento 2 – Óptimo conocido (Jaccard = 1)**

- Generamos instancias donde sabemos que existe una solución óptima en Jaccard.
- Objetivo: ver si Greedy-MO y NSGA-II son capaces de alcanzar ese óptimo, y analizar los valores en las demás medidas.

- **Experimento 3 – Instancias más grandes**

- 50 instancias y expresiones más profundas ($k = 10$).
- Objetivo: comparar Greedy-MO y NSGA-II en un escenario más realista, y evaluar calidad de soluciones, estabilidad y tiempos de ejecución.

Configuración común

Universo fijo $|U| = 128$; operadores $\{\cup, \cap, \setminus\}$.

Diseño experimental

- **Experimento 1 – Instancias pequeñas**

- Compara: Exhaustiva vs Greedy-MO vs NSGA-II con $k = 3$.
- Objetivo: ilustrar el comportamiento de las heurísticas comparándolo con el de la búsqueda exhaustiva.

- **Experimento 2 – Óptimo conocido (Jaccard = 1)**

- Generamos instancias donde sabemos que existe una solución óptima en Jaccard.
- Objetivo: ver si Greedy-MO y NSGA-II son capaces de alcanzar ese óptimo, y analizar los valores en las demás medidas.

- **Experimento 3 – Instancias más grandes**

- 50 instancias y expresiones más profundas ($k = 10$).
- Objetivo: comparar Greedy-MO y NSGA-II en un escenario más realista, y evaluar calidad de soluciones, estabilidad y tiempos de ejecución.

Configuración común

Universo fijo $|U| = 128$; operadores $\{\cup, \cap, \setminus\}$.

Diseño experimental

- **Experimento 1 – Instancias pequeñas**

- Compara: Exhaustiva vs Greedy-MO vs NSGA-II con $k = 3$.
- Objetivo: ilustrar el comportamiento de las heurísticas comparándolo con el de la búsqueda exhaustiva.

- **Experimento 2 – Óptimo conocido (Jaccard = 1)**

- Generamos instancias donde sabemos que existe una solución óptima en Jaccard.
- Objetivo: ver si Greedy-MO y NSGA-II son capaces de alcanzar ese óptimo, y analizar los valores en las demás medidas.

- **Experimento 3 – Instancias más grandes**

- 50 instancias y expresiones más profundas ($k = 10$).
- Objetivo: comparar Greedy-MO y NSGA-II en un escenario más realista, y evaluar calidad de soluciones, estabilidad y tiempos de ejecución.

Configuración común

Universo fijo $|U| = 128$; operadores $\{\cup, \cap, \setminus\}$.

Experimento 1: instancias pequeñas ($k = 3$)

Resultados (Semilla 1002)

| Algoritmo | # Sols. | Max M_J | Tiempo | 1ª Expresión |
|------------|---------|-----------|--------|--|
| Exhaustiva | 28 | 0.731 | 430 ms | $(F_0 \cup (F_1 \cup (F_2 \cap F_4)))$ |
| NSGA-II | 28 | 0.731 | 150 s | $((F_2 \cap F_4) \cup F_1) \cup F_0$ |
| Greedy-MO | 3 | 0.720 | < 1 ms | $(F_1 \cup F_0)$ |

Experimento 2: validación con óptimo conocido

($M_{\text{Jaccard}} = 1$)

Resultados globales

- **NSGA-II**: alcanza $M_{\text{Jaccard}} = 1,0$ en 11 de 20 instancias (55 %).
- **Greedy-MO**: alcanza $M_{\text{Jaccard}} = 1,0$ en 4 de 20 instancias (20 %).
- En ninguna instancia el Greedy-MO supera al NSGA-II en Jaccard \implies cuando no llega al óptimo, el NSGA-II se aproxima más (ej. semilla 1030: 0.975 vs 0.843).

Experimento 3: instancias grandes

Resumen estadístico (50 instancias)

| | Greedy-MO | NSGA-II |
|--------------------------------------|-----------|---------|
| Mejor M_{Jaccard} (mediana) | 0.722 | 0.802 |
| Tamaño frente (mediana) | 3 | 65 |
| $ \mathcal{O}p^*(e) $ (mediana) | 1 | 8 |
| $ \mathcal{F}(e) $ (mediana) | 2 | 7 |
| Tiempo (mediana) | < 1 ms | 900 s |

Experimento 3: instancias grandes

Test de Wilcoxon (pares NSGA-II vs Greedy)

- Datos pareados: mejor M_{Jaccard} de cada algoritmo en las mismas 50 instancias.
- Se aplica Wilcoxon de rangos signados (H_1 : NSGA-II > Greedy).
- Estadístico $Z \approx 5,9$, $p \approx 1,8 \times 10^{-9} < 0,05$.

⇒ La mejora en M_{Jaccard} de NSGA-II es **estadísticamente significativa**.

Contenido de Conclusiones

- 1 Introducción
- 2 Marco teórico
- 3 Aproximaciones algorítmicas
- 4 Experimentos
- 5 Conclusiones**

Valoración personal y competencias

- Hemos cumplido los **objetivos propuestos** tanto en el bloque teórico-matemático como en el práctico-computacional.
- Desarrollo de competencias en:
 - Búsqueda y **evaluación crítica de fuentes**.
 - **Abstracción** y separación clara entre teoría, implementación y experimentos.
 - Integración de **contenidos de varias asignaturas** (álgebra, algorítmica, modelos de computación, metaheurísticas, estadística).
- Mayor reto: diseño y calibración del algoritmo genético NSGA-II.

Vías futuras y posibles avances

- **Análisis de rendimiento fino:**

- Medir el tiempo que necesita NSGA-II para obtener un frente que domine completamente Greedy-MO en Jaccard, en lugar de usar un tiempo límite fijo.

- **Extender el espacio de objetivos:**

- Añadir nuevas métricas como objetivos adicionales.

- **Nuevos operadores en las expresiones:**

- Estudiar el impacto de operadores como la diferencia simétrica ($F_i \oplus F_j$) o la implicación ($F_i \rightarrow F_j$).

Bibliografía fundamental

- Lipschutz, S. (1998). *Set Theory and Related Topics*. McGraw-Hill. Utilizado para el estudio de las estructuras algebraicas.
- Davey, B. A. & Priestley, H. A. (2002). *Introduction to Lattices and Order*. Cambridge University Press. Utilizado para el estudio de las estructuras algebraicas.
- Manning, C. D., Raghavan, P. & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press. Utilizado para el estudio de las medidas.
- Sipser, M. (1996). *Introduction to the Theory of Computation*. International Thomson Publishing. Referencia esencial para el análisis de complejidad, definiciones de NP, NP-dureza y NP-completitud.
- Arora, S. & Barak, B. (2009). *Computational Complexity: A Modern Approach*. Cambridge University Press. Utilizado para el estudio de la complejidad computacional y la intratabilidad del problema.

Muchas gracias por su atención

Laura Lázaro Soraluce



**UNIVERSIDAD
DE GRANADA**