SPRINT 2

Gaizka Álvarez, Beñat Zubizarreta, Aitziber Cibrian

***Abstract*—Este trabajo aborda el problema de la detección de comunidades en grafos de coautoría científica, comparando tres enfoques: un algoritmo Greedy Constructivo (GRASP), Simulated Annealing (SA) y un Algoritmo de Estimación de Distribuciones (EDA). Los resultados experimentales revelan que, contrariamente a lo esperado para métodos poblacionales, el algoritmo EDA sufre una severa degradación en la calidad de la solución para un alto número de comunidades (K), alcanzando una modularidad media de solo 0.265. Por el contrario, el Simulated Annealing obtiene los mejores resultados de calidad ($\bar{x} \approx 0.94$), seguido muy de cerca por el GRASP ($\bar{x} \approx 0.92$). El análisis concluye que el algoritmo GRASP ofrece el mejor equilibrio coste-beneficio, logrando soluciones casi óptimas con un coste computacional significativamente inferior (10x más rápido) que las otras metaheurísticas.**

I. Introducción

Este trabajo se centra en redes de coautoría, donde los nodos representan autores y las aristas colaboraciones en publicaciones. El objetivo es maximizar la modularidad para segmentar la red en K comunidades, una solución es buena cuando los autores que hayan colaborado (han hecho al menos un trabajo de investigación juntos) estén en la misma comunidad.

II. El Problema

A. Definición y Representación

Se define un grafo $G = (V, E)$. La solución se representa como un vector S de longitud $|V|$, donde cada elemento $s_i \in \{1, \ldots, K\}$ indica la comunidad asignada al nodo i.

B. Función Objetivo

La calidad de la partición se mide mediante la modularidad Q:

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j) \tag{1}$$

donde m es el número de aristas, A la matriz de adyacencia, k_i es el grado del nodo i, δ es una función que devuelve 1 si la comunidad de los autores i y j son la misma, devuelve 0 en caso contrario.

III. Propuesta de Implementación

Se detallan los tres algoritmos diseñados:

A. GRASP

En esta implementación, se inicia con una solución aleatoria y se recorren los nodos en un orden estocástico. Para cada nodo, se evalúan las diferentes comunidades disponibles y se asigna aquella que maximice el $fitness$ (modularidad), buscando un equilibrio entre explotación y aleatoriedad.

B. Simulated Annealing (SA)

Este algoritmo regula el equilibrio entre la explotación de soluciones óptimas y la diversificación del espacio de búsqueda. Si el algoritmo encuentra una solución mejor que la actual, la acepta automáticamente. En caso de que la solución sea peor, puede ser aceptada con una probabilidad P, calculada mediante la siguiente distribución: $P = e^{-\Delta Q/T}$, donde T representa la temperatura actual y ΔQ es la diferencia de calidad entre la solución mala y la mejor encontrada hasta el momento. Este mecanismo permite al algoritmo escapar de óptimos locales.

C. Estimation of Distribution Algorithm (EDA)

Los Algoritmos de EDA son una clase de algoritmos poblacionales que sustituyen los algoritmos genéticos tradicionales por el aprendizaje y muestreo de modelos probabilísticos.

El funcionamiento del algoritmo sigue un ciclo iterativo basado en los siguientes pasos:

1) **Selección:** Se elige un subconjunto de los mejores individuos (elitismo) de la población actual basándose en su *fitness* (modularidad).
2) **Estimación del Modelo:** Se construye un modelo probabilístico que captura las correlaciones entre las comunidades y los nodos, premiando a las comunidades que más aparezcan en cada nodo, es decir, por cada nodo (autor) se cuenta cuántas veces aparece cada comunidad y se normaliza ese valor asignando así una probabilidad a cada comunidad.
3) **Muestreo (Sampling):** Se genera una nueva población de individuos mediante el modelo probabilístico aprendido en el paso anterior.
4) **Reemplazo:** La nueva población sustituye a la anterior, y el proceso se repite hasta alcanzar un criterio de parada.

IV. Experimentación

A. Configuración e Hiperparámetros

Para garantizar una comparación de calidad, los hiperparámetros de los algoritmos fueron optimizados utilizando la librería Optuna. Se han utilizado los siguientes valores:

- **Simulated Annealing (SA):**
  - Temperatura inicial (T_{ini}): 3.529
  - Temperatura final (T_{fin}): 1.061×10^{-5}
- **Estimation of Distribution Algorithm (EDA):**
  - Tamaño de la población: 81 individuos.
  - Porcentaje de elitismo: $\approx 25\%$ (0.2517).

B. Análisis de Resultados Globales

En esta sección analizamos el comportamiento general de los algoritmos variando el número de comunidades objetivo (K).

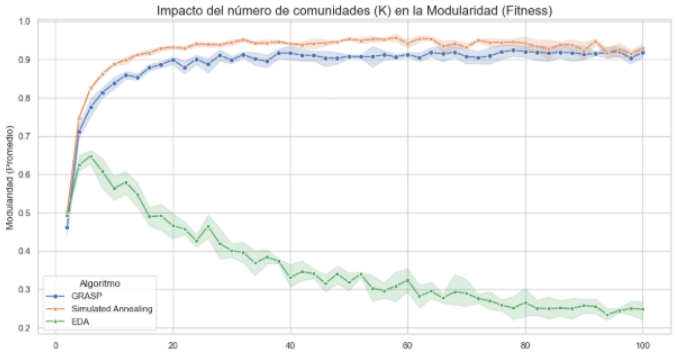

Fig. 1. Relación entre la Modularidad media y el número de comunidades (K).

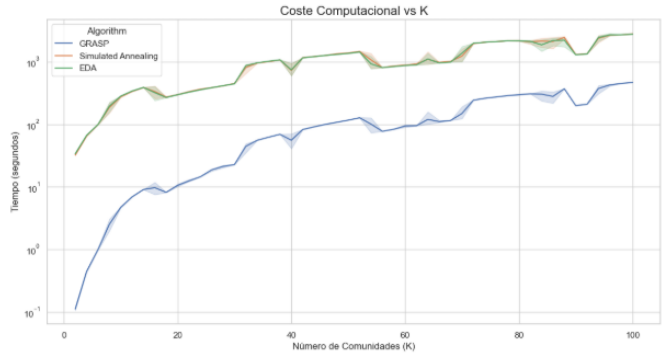

Fig. 2. Relación entre el Coste Computacional (escala logarítmica) y K.

Al analizar conjuntamente la **Figura 1** y la **Figura 2**, se observa un claro compromiso entre calidad y eficiencia:

1) **Calidad (Fig. 1):** Por un lado, los algoritmos $GRASP$ y SA demuestran ser superiores en términos de modularidad, especialmente para valores altos de K, aunque a partir de $K = 20$ la modularidad se estabiliza en ambos algoritmos (deja de crecer). Por otra parte, el algoritmo EDA funciona igual de bien que los otros dos hasta $K = 6$, a partir de aquí conforme tenemos más comunidades empeoran los resultados.
2) **Coste (Fig. 2):** En esta imagen se evidencia que las tres metaheuristicas tienen el mismo comportamiento. Ademas de esto, se puede observar que SA y EDA tienen el mismo coste llegando a casi 1 hora por ejecución en K-s grandes y que $GRASP$ es mucho menos costoso que los otros dos algoritmos, llegando a ser casi 10 veces más barato durante todas las ejecuciones.
3) **Conclusión comparativa:** El claro ganador en relación calidad-coste es $GRASP$ ofrece resultados realmente buenos y es 10 veces más barato que los otros dos algoritmos. En segunda posición, tenemos al SA que aunque sea muy costoso ofrece los mejores resultados (ligeramente mejores que $GRASP$). Por ultimo, está EDA que es costoso y los resultados que ofrece son muy pobres.

C. Análisis Detallado para el Mejor Caso ($K = 80$)

Tras observar la evolución global, detectamos que en $K = 80$ se maximiza la modularidad global. Los resultados de EDA son pobres pero SA y $GRASP$ obtienen resultados óptimos.

TABLE I
ESTADÍSTICAS DESCRIPTIVAS PARA $K = 80$

Algoritmo	Media	Std (Desv)	Mín	Máx
EDA	0.2652	0.0477	0.2046	0.3266
GRASP	0.9211	0.0182	0.8979	0.9389
SA	0.9429	0.0222	0.9132	0.96815

1) Dispersión y Estabilidad (Boxplot): La **Figura 3** combina un diagrama de Cajas y Bigotes con un *Swarmplot* para evaluar la fiabilidad.

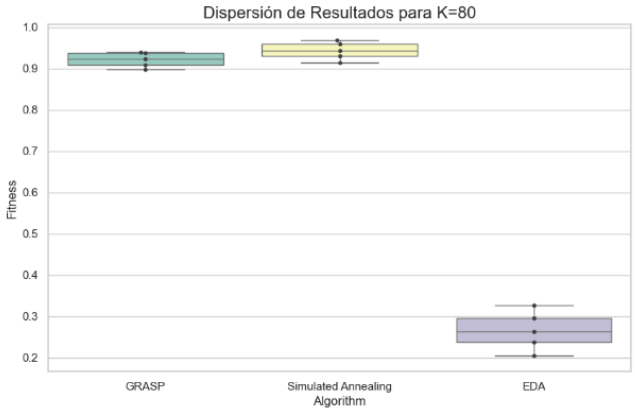

Fig. 3. Dispersión de resultados para $K = 80$. Comparativa de estabilidad.

El análisis de la gráfica revela comportamientos opuestos:

- **La Caja (Estabilidad) y los puntos (Swarmplot):** La línea central indica la mediana del algoritmo y la altura de la caja representa el 50% central de las ejecuciones (De Q3 a Q1). Observamos que la caja del $GRASP$ y SA es plana, lo que indica una gran estabilidad en el 50% central de los datos, aparte de esto, si nos fijamos en los intervalos de valores no atípicos (los bigotes de las cajas) vemos que se repite el patrón de poca variabilidad. Hay un poco de distancia entre las cajas y los intervalos minimos porque al en el primer cuarto el $fitness$ de ambos algoritmos empieza "bajo" y "crece" mucho.
  En contraste, la caja del EDA es muy alta, señalando que los resultados varían drásticamente entre ejecuciones en la mitad central, tambien, se repite el patron cuando miramos los intervalos de valores no atípicos, tanto en el primer como en el último cuarto la diferencia entre el primer y ultimo punto es notable.

2) Probabilidad de Éxito (KDE): La **Figura 4** muestra la Estimación de Densidad de Kernel (KDE), que representa la probabilidad de obtener una cierta nota de *fitness* mediante la forma de ”montañas”.

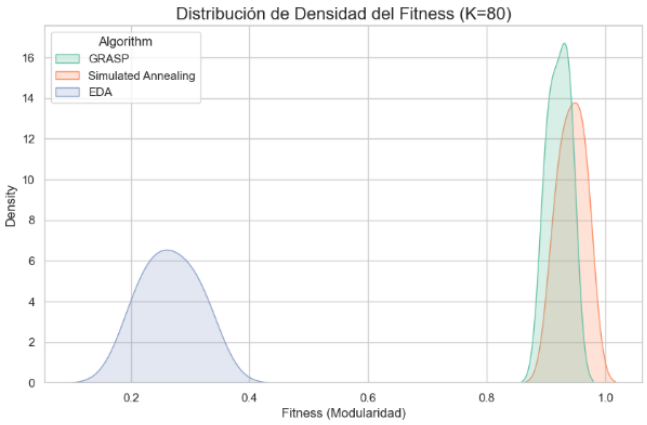

Fig. 4. Distribución de Densidad (KDE) para $K = 80$.

Interpretamos la forma y posición de las curvas:

- **Forma de la Montaña (Predictibilidad):** La curva del $GRASP$ y del SA son una ”montaña” estrecha, alta y estan posicionadas a la derecha, lo que significa que el algoritmo obtiene resultados parecidos (Poca anchura), con una alta probabilidad (Mucha altura) y ademas esta nota es muy buena (La montaña esta a la derecha). Por el contrario, la curva del EDA es ancha, baja y esta posicionada en la izquierda del gráfico, indicando que es más impredecible (explora un rango más amplio de soluciones (ancha), con poca probabilidad de elegir una respuesta (baja)) y estas soluciones son malas (Montaña posicionada en la izquierda).
- **Posición (Comparación):** Por un lado, vemos que EDA tiene una mala distribuición al ser una montaña baja y ancha. Por otro lado, $GRASP$ y SA tienen resultados mucho más positivos siendo montañas más estrechas y mejor posicionadas. La imagen exhibe que SA abarca resultados ligeramente mejores que $GRASP$.

V. Conclusiones y Trabajo Futuro

A partir del análisis experimental realizado sobre grafos de coautoría científica para diferentes valores de K (número de comunidades), se extraen las siguientes conclusiones principales:

A. Supremacía de Simulated Annealing (SA) en Calidad

El algoritmo de Simulated Annealing se posiciona como la metaheurística más eficaz en términos de maximización de la modularidad. En el escenario más complejo ($K = 80$), alcanzó el mejor promedio ($\bar{x} \approx 0.9429$) y el máximo absoluto (0.9681). Esto demuestra su capacidad superior para escapar de óptimos locales y explorar el espacio de búsqueda de manera efectiva en comparación con las otras técnicas evaluadas.

B. GRASP como la opción más Eficiente

Aunque el SA obtiene resultados ligeramente superiores en calidad, el algoritmo $GRASP$ se presenta como la alternativa más equilibrada en la relación calidad-coste. Ofrece resultados de muy alta calidad ($\bar{x} \approx 0.9211$), muy cercanos a los de SA, pero con un coste computacional drásticamente menor (aproximadamente 10 veces más rápido). Esto lo convierte en el algoritmo idóneo cuando se dispone de recursos computacionales limitados o se requiere una respuesta rápida.

C. Desempeño deficiente del EDA en espacios complejos

Contrario a las expectativas iniciales para algoritmos poblacionales, el Estimation of Distribution Algorithm (EDA) mostró un rendimiento pobre para valores de $K > 6$. Mientras que $GRASP$ y SA estabilizaron su modularidad en valores altos a partir de $K = 20$, el EDA sufrió una degradación severa de la calidad a medida que aumentaba el número de comunidades, alcanzando apenas una media de 0.2652 en $K = 80$. Esto sugiere que el modelo probabilístico utilizado no logra capturar adecuadamente las correlaciones complejas de la red de vértices.

D. Estabilidad y Predictibilidad

El análisis estadístico (basado en los diagramas de caja y KDE) confirma que tanto $GRASP$ como SA son algoritmos altamente estables y predecibles. Sus distribuciones de resultados son estrechas y están concentradas en valores de alta calidad (representadas como “montañas” altas a la derecha en las gráficas de densidad), lo que garantiza confianza en la ejecución. Por el contrario, el EDA exhibió una alta inestabilidad y dispersión (cajas altas y montañas anchas a la izquierda), haciendo que la calidad de la solución final dependa excesivamente del azar en la ejecución.

E. Trabajo Futuro y Mejoras

Para extender el alcance de esta investigación, se proponen las siguientes líneas de actuación:

- **Escalabilidad del Dataset:** Los experimentos actuales se han limitado al 50% de la base de datos disponible. Es necesario validar si los patrones y resultados obtenidos se mantienen tambien con la base de datos completa.
- **Hibridación (Mezclar algoritmos):** Una mejora prometedora sería implementar un algoritmo híbrido. Se podría utilizar la solución del $GRASP$ para inicializar el modelo probabilístico de EDA.
- **Nuevas Metaheurísticas:** Se propone explorar otros algoritmos para abordar este problema. Por ejemplo, basados en inteligencia de enjambre, como *Ant Colony Optimization* (ACO) o *Particle Swarm Optimization* (PSO)

References

[1] Mark EJ Newman, Aaron Clauset and Cristopher Moore “Finding community structure in very large networks,” 2004.