# Algoritmo metaheurístico para el problema de la minimización de la influencia con costes

Mario Casas-Pérez[1][0009-0009-9828-3558],
Isaac Lozano-Osorio[2][0000-0002-2608-8464] y
Oscar Cordón[1][0000-0001-5112-5629]

[1] Dept. Computer Science and AI and Andalusian Research Institute in Data Science and Computational Intelligence, University of Granada, 18071, España
mariocp11@correo.ugr.es,ocordon@decsai.ugr.es
[2] GRAFO Research, Departamento de Informática y Estadística, Universidad Rey Juan Carlos, C/Tulipán S/N, Móstoles, 28933, Madrid, España
isaac.lozano@urjc.es

**Resumen** Este artículo aborda el problema de minimización de influencia (*IMIN*) bajo restricción presupuestaria en redes sociales, donde dicha influencia está iniciada por un conjunto de actores malintencionados denominados semillas. El objetivo es minimizar esta difusión perniciosa bloqueando nodos a cambio de un coste asociado a cada nodo individual. El problema se formula como una tarea de optimización combinatoria $\mathcal{NP}$*-hard*, para la que se propone la metaheurística *Greedy Randomized Adaptive Search (GRASP)* dónde la fase de mejora consiste en una búsqueda local mediante intercambio de nodos y evaluación subrogada para reducir el coste de las simulaciones Monte Carlo. Consideramos una estrategia heurística de priorización en la selección de nodos candidatos basada en el tamaño esperado de los subárboles en un árbol de dominadores calculado mediante el algoritmo de *Lengauer-Tarjan* sobre múltiples realizaciones de Monte Carlo, lo que permite identificar de forma precisa los cuellos de botella estructurales que controlan la difusión. La metodología se evalúa experimentalmente sobre ocho redes reales de distinta escala bajo el modelo de difusión *Independent Cascade*, considerando distintos presupuestos. Los resultados muestran que la variante basada en la heurística propuesta obtiene el mejor valor de la función objetivo en todas las instancias evaluadas y con una desviación porcentual promedio del 0.00 %, aunque con un coste computacional superior en la mayoría de casos. No obstante, la evidencia experimental indica que la combinación de construcción aleatorizada y refinamiento local constituye una estrategia efectiva para abordar la *IMIN* con presupuesto heterogéneo.

**Keywords:** Ciencia de Redes · Minimización de Influencia · GRASP

## 1. Introducción

El análisis de procesos de difusión en redes sociotécnicas constituye uno de los pilares de la ciencia de redes contemporánea. La capacidad de modelar cómo

se propagan ideas, comportamientos, patógenos o desinformación tiene implicaciones en ámbitos tan diversos como la salud pública, la ciberseguridad y la resiliencia de infraestructuras críticas [3]. Tradicionalmente, la literatura se ha centrado en el problema de la maximización de influencia ($IM$), orientado a identificar un conjunto de nodos (habitualmente llamados semillas) que maximice el alcance de una difusión bajo modelos estocásticos de difusión de la influencia como *Independent Cascade* ($IC$) o *Linear Threshold* ($LT$) [5]. No obstante, la creciente amenaza de las campañas de desinformación y los ataques coordinados en plataformas digitales ha cambiado el foco hacia el problema inverso: la minimización de influencia ($IMIN$).

El objetivo de la $IMIN$ es identificar un conjunto de nodos bloqueadores que, al ser intervenidos, minimicen la propagación esperada de un agente dañino previamente detectado [2]. A pesar de su relevancia, este problema presenta desafíos computacionales y teóricos. En primer lugar, $IMIN$ es intrínsecamente $\mathcal{NP}$*-hard*. En segundo lugar, a diferencia de la maximización, la función objetivo en escenarios de minimización a menudo pierde propiedades de submodularidad, lo que invalida las garantías teóricas de los algoritmos voraces tradicionales. Además, la evaluación de la influencia mediante simulaciones de Monte Carlo resulta prohibitiva en redes de gran escala.

En consecuencia, la investigación actual se enfrenta a cuatro retos fundamentales: i) la escalabilidad de los algoritmos en grafos de millones de nodos, ii) el desajuste entre los modelos teóricos y la dinámica real de la red, iii) la detección tardía de brotes infecciosos, y iv) la necesidad de validación empírica bajo restricciones de privacidad. Es por ello que en este trabajo abordamos la $IMIN$ bajo una formulación de presupuesto limitado, donde los costes de intervención son heterogéneos. Definimos por tanto el problema de optimización como:

$$B^* = \arg \min_{B \subseteq V \setminus S} \mathbb{E}[IG(S \mid B)], \quad \text{sujeto a: } \sum_{v \in B} c(v) \leq \texttt{FR}$$

donde $\mathbb{E}[\mathrm{IG}(S \mid B)]$ representa el número de nodos activados obtenidos con un modelo de difusión de influencia, cuya obtención requiere un conjunto inicial de semillas maliciosas $S$ y nodos bloqueadores del conjunto candidato $B$, que no pueden pertenecer al conjunto de semillas. Además, $V$ es el conjunto total de nodos de la red social, *c(v)* el coste asociado a bloquear el nodo $v \in V$, que en este trabajo depende del grado del nodo, y `FR` es el presupuesto total disponible.

Nuestra contribución se puede estructurar en tres ejes. Primero, introducimos la propia formulación del problema de optimización con costes. Segundo, proponemos y evaluamos una jerarquía de estrategias que van desde heurísticas voraces hasta la metaheurística *GRASP* (*Greedy Randomized Adaptive Search Procedure*). Finalmente, para la experiencia computacional implementamos técnicas de paralelización para mitigar el cuello de botella que suponen las simulaciones estocásticas en modelos $IC$, permitiendo la exploración de espacios de soluciones más amplios en tiempos competitivos.

A través de una experimentación exhaustiva en diversas topologías de red, este estudio demuestra que la combinación de metaheurísticas adaptativas per-

mite obtener reducciones de influencia considerables, manteniendo un equilibrio eficiente entre la calidad de la solución y el coste computacional.

El resto del artículo continúa en la Sección 2 dónde se contextualiza el estado del arte en *IMIN* y modelos de presupuesto heterogéneo. La Sección 3 detalla el diseño de la metaheurística *GRASP*, profundizando en sus fases constructiva y de búsqueda local. La Sección 4 describe el entorno experimental, las redes sociales analizadas y los estadísticos de evaluación. Finalmente, la Sección 5 resume los hallazgos principales y comenta las líneas de trabajo futuro.

## 2. Literatura

El estudio de la propagación en redes sociales y el marketing viral tiene su origen en los trabajos de *Domingos* y *Richardson*. No obstante, la formalización algorítmica fundamental fue establecida por *Kempe et al.* [5], al definir los modelos de *IC* y *LT*. En su trabajo, demostraron que la selección del conjunto óptimo de semillas en maximización de influencia es un problema $\mathcal{NP}$*-hard*, lo que consolidó el uso de simulaciones Monte Carlo y algoritmos voraces basados en la propiedad de submodularidad para obtener garantías de aproximación de (1-1/e). Estas simulaciones presentan un enfoque computacional algorítmico donde se utiliza el muestreo aleatorio repetido para obtener resultados numéricos y estimar el comportamiento de sistemas complejos o matemáticamente intratables. A medida que el número de simulaciones tiende a infinito, el promedio de los resultados muestrales converge al valor esperado real del sistema.

Aunque la literatura inicialmente se centró en la *IM* mediante aceleradores como *CELF* [8] o técnicas de muestreo inverso (*RIS*) [12], el foco se ha orientado recientemente hacia la *IMIN*. Este campo se divide en dos vertientes principales, destacando las estrategias de bloqueo, donde se eliminan nodos o aristas para romper las rutas de propagación, y las estrategias de contrainformación, donde la difusión de contenido positivo se utiliza para neutralizar difusiones perniciosas. A diferencia de la *IM*, la *IMIN* suele enfrentarse a la pérdida de submodularidad y a una mayor complejidad computacional, especialmente bajo el modelo *IC*, donde la evaluación de la reducción de influencia es altamente costosa.

Por otro lado, la literatura distingue dos marcos operativos. El modelo tradicional prefija el número de semillas $k$ a seleccionar y asume costes homogéneos, facilitando la aplicación de heurísticas clásicas. Sin embargo, trabajos más recientes enfatizan la necesidad de modelos de presupuesto heterogéneo, donde el coste de intervención sobre un nodo es proporcional a su importancia estructural. Esta evaluación es más realista pero algorítmicamente más exigente, al requerir optimizar el ratio beneficio/coste en escenarios de recursos limitados [1].

Las soluciones actuales varían entre la aceleración estructural y las metaheurísticas. Autores como *Xie et al.* [13] han propuesto el uso de estructuras como el árbol de dominadores (*dominator tree*) para identificar nodos críticos de bloqueo de forma escalable en modelos *LT*, superando las limitaciones de tiempo de las simulaciones exhaustivas. Por otro lado, ante la ausencia de garantías teóricas en modelos no submodulares, enfoques *data-driven* y metaheurísticos como *GRASP*

han demostrado robustez en la exploración de espacios complejos de soluciones [10], permitiendo refinar soluciones iniciales mediante operadores de intercambio (*swap*).

En este contexto, la presente investigación aborda la *IMIN* bajo un presupuesto financiero `FR`. Se evalúa la eficacia de estrategias voraces frente a metaheurísticas híbridas (*GRASP*), estudiando la reducción esperada de la propagación además de la estabilidad y la eficiencia computacional. Para abordar este estudio, bloquearemos un subconjunto de nodos, seleccionado por el algoritmo de *IMIN*, para reducir los caminos de la propagación de información maliciosa. Dado que cada nodo tiene un coste de bloqueo en función de su grado, el subconjunto seleccionado debe ajustarse al presupuesto disponible para ser considerado una solución factible.

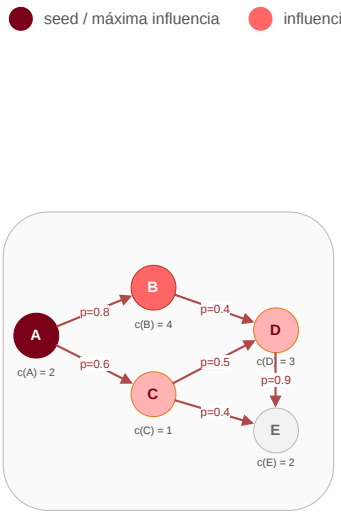



Figura 1: Red original

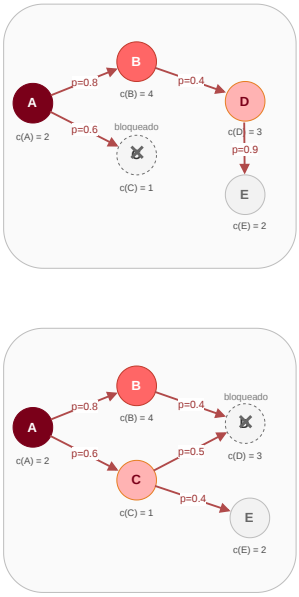



Figura 2: Tras bloquear un nodo

El ejemplo mostrado en las Figuras 1 y 2 ilustra el problema de *IMIN* bajo presupuesto, donde la primera figura muestra la red original y la propagación maliciosa esperada inicialmente considerando el nodo `A` como semilla. El valor `p` muestra la probabilidad del nodo de infectar a los nodos que direcciona (influencia), asumiendo que un valor de `p` superior a 0.50 provoca la activación del nodo adyacente y donde la intensidad de color muestra la probabilidad de activación

que presenta este. El valor `c(X)` refleja el coste de bloquear cada nodo, que en este ejemplo no se corresponde con su grado. Con un presupuesto $\texttt{FR} = 3$ podemos bloquear el nodo `D` o el nodo `C`, pero no ambos. La primera opción (Figura 2 abajo) corta la ruta de alto impacto hacia el nodo `E`, reduciendo la propagación esperada en 1. Sin embargo, eliminando `C` (Figura 2 arriba) se reduce la propagación en 2, dado que la arista de `B` a `D` presenta un `p` de 0.4 y en consecuencia no se produciría la activación de `D`. Por tanto, la solución óptima sería bloquear el nodo `C` ya que se reduciría la propagación hacia `D` y `E`.

## 3. Descripción algoritmica

La metaheurística *Greedy Randomized Adaptive Search Procedure* (*GRASP*), introducida originalmente por *Feo* y *Resende* [4,11], es un proceso iterativo de arranque múltiple ampliamente reconocido por su eficacia y flexibilidad en la resolución de problemas de optimización combinatoria. Su arquitectura se fundamenta en la ejecución sucesiva de dos fases principales: una fase de construcción donde se genera una solución inicial mediante un proceso voraz adaptativo con un componente de azar controlado y una fase de mejora que refina dicha solución hasta alcanzar un óptimo local. La principal ventaja de GRASP reside en su notable escalabilidad y su capacidad para equilibrar la exploración y explotación del espacio de búsqueda. Debido a su sólido rendimiento documentado en trabajos previos de diversa naturaleza, se ha seleccionado como la base algorítmica para esta investigación. [4,7,9,10]

En el contexto específico de los problemas de difusión en redes, *GRASP* ofrece características críticas como la diversidad de soluciones, el control de la aleatoriedad y la posibilidad de incluir procedimientos de evaluación incrementales que reducen el coste computacional.

La eficiencia práctica de este enfoque ha sido validada en estudios como el de *Lozano-Osorio et al.* [9], donde se propuso un método rápido para la selección de nodos semilla en problemas de *IM*. Siguiendo dicha estrategia, es posible diseñar una fase de construcción que evite simulaciones costosas, utilizando en su lugar medidas de vecindad para obtener puntos de partida competitivos de forma casi instantánea. Esta aproximación permite generar soluciones iniciales de manera eficiente, aunque su utilidad depende de que dicha medida refleje de forma razonable el impacto real de los nodos sobre la difusión. Cuando esta relación es débil, la solución inicial puede ser menos informativa y necesitar una mayor labor de refinamiento en la búsqueda local [6]. Por tanto, nuestro diseño busca un compromiso óptimo entre la calidad de la solución y el tiempo de cómputo, permitiendo que la búsqueda local ejecute evaluaciones de mayor precisión de manera selectiva.

Para evaluar la calidad de solución de nuestro *GRASP*, la función objetivo de nuestro problema es la minimización de la propagación esperada. Dado un conjunto de semillas $S$ y un conjunto de nodos bloqueadores $B$ seleccionados por el algoritmo, la función $\sigma(S, B)$ representa el número esperado de nodos activados al finalizar el proceso de difusión bajo el modelo *IC*. Debido a la

naturaleza estocástica del modelo $IC$, este valor se estima mediante el promedio de $T$ réplicas independientes de Monte Carlo.

**Algoritmo: GRASP de selección de nodos bloqueadores en *IMIN bajo presupuesto***

---

**Require:** Grafo $G$, $S$, $FR$, $\alpha, I$
**Ensure:** Conjunto de bloqueadores $B_{best}$

1: $B_{best} \leftarrow \emptyset$
2: **for** $i = 1$ **to** $I$ **do**
3: {Fase 1: Construcción Voraz aleatorizada}
4: $B_{curr} \leftarrow \emptyset$
5: $\texttt{FR}_{curr} \leftarrow \texttt{FR}$
6: **while** true **do**
7: $C \leftarrow \{v \in V \setminus (S \cup B_{curr}) \mid c(v) \leq \texttt{FR}_{curr}\}$
8: **if** $C$ es vacío **then**
9: **break**
10: **end if**
11: Calcular una puntuación heurística $s(v)$ para cada $v \in C$
12: $s_{min} \leftarrow \text{mín}_{v \in C}\, s(v)$
13: $s_{max} \leftarrow \text{máx}_{v \in C}\, s(v)$
14: $\tau \leftarrow s_{min} + \alpha \cdot (s_{max} - s_{min})$
15: $RCL \leftarrow \{v \in C \mid s(v) \geq \tau\}$
16: Seleccionar aleatoriamente $v_{sel}$ de $RCL$
17: $B_{curr} \leftarrow B_{curr} \cup \{v_{sel}\}$
18: $\texttt{FR}_{curr} \leftarrow \texttt{FR}_{curr} - c(v_{sel})$
19: Actualizar el estado residual del grafo
20: **end while**
21: {Fase 2: Fase de mejora}
22: $B_{curr} \leftarrow$ LOCAL_SEARCH_SURROGATE($B_{curr}$)
23: **if** $B_{best} = \emptyset$ **or** $f(B_{curr}) < f(B_{best})$ **then**
24: $B_{best} \leftarrow B_{curr}$
25: **end if**
26: **end for**
27: **return** $B_{best}$

---

El algoritmo comienza con la inicialización, donde se define el valor del conjunto óptimo de bloqueadores $B_{best}$ como un conjunto vacío (línea 1). El bucle principal (línea 2) controla el número de iteraciones totales ($I$). Cada iteración constituye un proceso de multiarranque independiente, diseñado para explorar diferentes regiones del espacio de búsqueda y mitigar el riesgo de convergencia en óptimos locales.

### 3.1. Fase constructiva

En esta etapa se genera una solución factible de forma iterativa y adaptativa. En las líneas 4 y 5 se definen el conjunto de bloqueadores actual ($B_{curr}$) y el presupuesto disponible ($\texttt{FR}_{curr}$) para la iteración actual, respectivamente. En la línea 6, el proceso de construcción continúa mientras persista un presupuesto suficiente. En la línea 7 se identifica el conjunto $C$, compuesto por nodos que no pertenecen a las semillas iniciales $B$, no han sido seleccionados previamente en $B_{curr}$ y cuyo coste sea asumible por el presupuesto actual. Si $C$ resulta ser un conjunto vacío (línea 8), el proceso de construcción finaliza anticipadamente (línea 9 y 10). En caso contrario, se calcula un valor heurístico para cada candidato (línea 11).

Para la evaluación y asignación de puntuaciones a los nodos candidatos, se han implementado dos estrategias voraces diferenciadas, denominadas *OIM* y *GRADO*, combinando enfoques consolidados de la literatura con propuestas metodológicas basadas en [13]. La primera de ellas, la estrategia *OIM*, adopta una hibridación del modelo *IC* con el algoritmo de *Lengauer-Tarjan* para el cálculo de árboles de dominadores, lo que permite estimar probabilísticamente el impacto estructural global de cada nodo a través de simulaciones Monte Carlo. Por el contrario, la estrategia *GRADO* se introduce como el marco de comparación de referencia ya que, dado que no existen soluciones previas en el estado del arte que aborden este escenario específico bajo la restricción presupuestaria, esta heurística fue diseñada para calcular de manera puramente local el grado de salida esperado hacia vecinos no bloqueados.

Basándose en los valores extremos $s_{min}$ y $s_{max}$ (líneas 12 y 13), se establece un umbral de calidad $\tau$ mediante el parámetro $\alpha \in [0, 1]$ (línea 14), donde un valor alto de $\alpha$ indica mayor voracidad y un valor menor mayor aleatoriedad. Además, este umbral permite filtrar la lista restringida de candidatos (*RCL*) en la línea 15, que introduce únicamente a los nodos cuya puntuación es igual o superior a $\tau$. Luego, se selecciona aleatoriamente un nodo $v_{sel}$ de la *RCL* (línea 16) para su inclusión en $B_{curr}$ (línea 17). Tras actualizar el presupuesto (línea 18), el algoritmo recalcula el estado del grafo residual (línea 19). Esta naturaleza adaptativa es crítica debido a que las puntuaciones de los candidatos restantes se actualizan dinámicamente según las propiedades incidentes de la red tras cada selección.

### 3.2. Fase de mejora

La función de la búsqueda local es refinar la solución inicial hasta un mínimo local. Se repite hasta cumplir el criterio de parada, devolviendo la mejor solución encontrada. Por tanto, se aplica el procedimiento de búsqueda local (línea 22). Para optimizar la eficiencia computacional, este paso utiliza una función de evaluación subrogada, la cual tiene en cuenta aquellos X nodos no seleccionados cuyo grado sea mayor. Este enfoque minimiza el recurso de simulaciones Monte Carlo costosas, evaluando únicamente los intercambios de vecindad más prometedores antes de realizar la evaluación final de la solución mejorada. Si la solución

obtenida tras la búsqueda local reduce la propagación esperada respecto al mejor registro previo ($B_{best} = \emptyset$ o $f(B_{curr}) < f(B_{best})$, línea 23), se actualizan los valores de $B_{best}$ (línea 24) con los nuevos resultados.

Tras completar el ciclo de iteraciones, el algoritmo devuelve el conjunto $B_{best}$ que representa la configuración de bloqueadores más eficiente encontrada durante el proceso de búsqueda (línea 27).

## 4. Resultados

Para evaluar la robustez y escalabilidad del algoritmo propuesto, se han seleccionado ocho redes del mundo real con distinta escala (que abarca desde miles de nodos hasta millones de aristas) y diversas propiedades estructurales a partir del *Stanford Large Network Dataset Collection (SNAP)*. El conjunto de datos incluye redes sociales no dirigidas de gran escala como *com-youtube* (1.134.890 nodos y 2.987.624 aristas) y la red de colaboración científica *com-dblp* (317.080 nodos y 1.049.866 aristas). Además, se incorporan redes de comunicación por correo electrónico tanto masivas como compactas, representadas por *Email-EuAll* (265.214 nodos y 420.045 aristas) y *email-Eu-core* (1.005 nodos y 25.571 aristas), siendo ambas redes dirigidas. La experimentación también contempla círculos sociales anonimizados de *facebook_ combined* (4.039 nodos y 88.234 aristas) y *twitter_ combined* (81.306 nodos y 1.768.149 aristas), además de estructuras de hipervínculos web como el grafo *web-Stanford* (281.903 nodos y 2.312.497 aristas) y redes de interacción comunitaria como *Wiki-Vote* (7.115 nodos y 103.689 aristas). Esta diversidad en el tamaño y la naturaleza de los grafos permite validar la eficiencia de las estrategias de minimización en entornos de computación intensiva y en escenarios de topología compleja.

Para terminar de definir las instancias del *IMIN* bajo presupuesto, a cada nodo se le ha asignado un coste en base a su grado. Finalmente, cada instancia presenta el número de nodos inicialmente infectados (semillas), el cual se obtuvo mediante los métodos más avanzados del problema de *IM* [9]. Para el valor `FR`, que indica el presupuesto disponible para seleccionar los nodos bloqueadores, hemos considerado cuatro valores distintos para cada instancia base. Se han obtenido como un porcentaje del coste total de la red, calculado como la sumatoria de los costes individuales de todos los nodos. En concreto, se considera un 1, 5, 10 y 15 por ciento del coste total, lo que resulta en un total de 32 instancias (8 instancias base con 4 valores distintos de `FR` cada una).

Cada cálculo del número esperado de nodos activados al finalizar el proceso de difusión $\sigma(S, B)$ para cada solución candidata $B$ implica un total de 100 iteraciones Monte Carlo. Los experimentos se llevaron a cabo en la infraestructura proporcionada por la Universidad Rey Juan Carlos, un servidor con procesador *AMD EPYC 7643* (48 núcleos físicos, aunque la ejecución se limitó a 16 núcleos) y 32 GB de *RAM*. El sistema operativo fue *Linux*, el entorno de compilación y ejecución incluyó *GCC 13*, configurado para emplear el estándar *C++17* con la optimización *-O3*, y el lenguaje de programación empleado para scripts fue *C++* y para herramientas *Python 3.12.3*.

Para ilustrar los resultados se muestran las diferentes métricas utilizadas. La función objetivo $\sigma(S, B)$ (FO) representa la magnitud de la difusión esperada tras la reducción de influencia. Se calcula como el promedio de nodos infectados una vez que el algoritmo ha bloqueado los nodos seleccionados en $B$ en las 100 iteraciones Monte Carlo. Además, tendremos en cuenta la desviación porcentual (Desv. (%)) para medir qué tan lejos se queda de la solución de un algoritmo respecto a la mejor solución de referencia. Adicionalmente, el tiempo (Tiempo (s)) de cómputo empleado y el valor Mejor indica si el algoritmo alcanza el mejor valor conocido de la función objetivo para cada instancia.

### 4.1. Experimentación preliminar

Se ha realizado una fase preliminar, orientada a ajustar los hiperparámetros del algoritmo *GRASP*, seleccionando el 25 % de las instancias disponibles para los experimentos. De las 32 instancias consideradas, se eligieron 8 instancias representativas. Para garantizar la diversidad, se seleccionó una instancia base por cada red con su respectivo valor de `FR` y sobre esa muestra se evaluó la calidad de nuestro *GRASP*. Cabe destacar que el número de iteraciones $I$ de *GRASP* fue fijado a 10.

| **Algoritmo** | $\alpha$ | **FO** | **Desv. (%)** | **Tiempo (s)** | **Mejor** |
|---|---|---|---|---|---|
| **GRASP 100** | 0.25 | 4163.98 | 5.17 | 5246.06 | 2/8 |
| | 0.50 | 4145.23 | 2.05 | 4258.11 | 1/8 |
| | 0.75 | 4183.81 | 1.58 | 4350.61 | 1/8 |
| | 1.00 | 4150.23 | 1.66 | 4296.00 | 2/8 |
| | RND | 4138.89 | 0.95 | 4365.50 | 2/8 |
| **GRASP 50** | 0.25 | 4179.50 | 6.14 | 3387.23 | 1/8 |
| | 0.50 | 4152.67 | 1.97 | 3400.71 | 1/8 |
| | 0.75 | 4181.00 | 1.73 | 3465.05 | 1/8 |
| | 1.00 | 4160.40 | 1.37 | 3505.80 | 2/8 |
| | RND | **4118.35** | 0.83 | 3422.41 | 2/8 |
| **GRASP 25** | 0.25 | 4188.63 | 6.99 | 3134.09 | 1/8 |
| | 0.50 | 4155.44 | 2.74 | 3166.14 | 1/8 |
| | 0.75 | 4177.53 | 2.00 | 3137.51 | 1/8 |
| | 1.00 | 4150.05 | 0.97 | 3158.80 | 2/8 |
| | RND | 4135.87 | **0.52** | **3133.79** | **3/8** |

Cuadro 1: Comparación preliminar con varios $\alpha$ y surrogate.

Hemos realizado un estudio del parámetro $\alpha$ del *GRASP* (véase la Sección 3.1) considerando los valores $\{0.25, 0.50, 0.75, 1.00\}$ y una variante aleatorizada denotada como *RND*. En esta última, el valor de $\alpha$ se asigna de forma

estocástica en cada iteración para fomentar la diversificación en la búsqueda. Cabe destacar que un incremento en $\alpha$ proporciona al algoritmo un comportamiento más voraz, mientras que valores menores favorecen la aleatoriedad. Asimismo, para la búsqueda local subrogada (véase la Sección 3.2) se han evaluado límites de exploración de 25, 50 y 100 nodos (denotado como GRASP 25, 50 y 100 en la tabla). Estos umbrales definen el tamaño de la vecindad de búsqueda basándose en el orden de importancia estructural de los nodos (grado), optimizando así el compromiso entre la calidad de la mejora y el coste computacional.

Los resultados preliminares recogidos en la Tabla 1 muestran una tendencia clara donde la reducción del tamaño de *Surrogate* disminuye drásticamente los tiempos de ejecución, pasando de un promedio general superior a los 4300 segundos en *Surrogate 100* a una media cercana a los 3140 segundos en *Surrogate 25*. En cuanto a la calidad de las soluciones y la estabilidad del algoritmo, las configuraciones con $\alpha$ en modo *RND* superan consistentemente a los valores fijos en los tres bloques evaluados. Se observa también que los valores bajos de $\alpha$ como 0.25 incrementan el tiempo de cómputo y elevan la desviación porcentual, mostrando que una exploración excesivamente aleatorizada perjudica tanto a la eficiencia como la consistencia del algoritmo para nuestro problema.

A partir de este comportamiento, se determina que la combinación de *Surrogate 25* junto con el parámetro $\alpha$ en modo *RND* es la mejor opción para la configuración definitiva del algoritmo. Aunque *Surrogate 50* (*RND*) consigue un valor promedio ligeramente inferior, la variante *Surrogate 25* (*RND*) ofrece el equilibrio óptimo rendimiento/tiempo de ejecución y el comportamiento más robusto del estudio preliminar. Esta combinación destaca por registrar el tiempo de ejecución más rápido de toda la comparativa (3133.79), la menor desviación porcentual (0.52 %), asegurando una alta repetibilidad, y la mayor tasa de éxito al conseguir la mejor solución en 3 de las 8 instancias estudiadas.

### 4.2. Experimentación final

Una vez encontrada la mejor combinación para nuestra metaheurística, se usa todo el conjunto de redes sociales y se comparan las dos heurísticas de selección: *OIM* y *GRADO* (véase la Sección 3.1). Los resultados para todas las instancias se muestran en la Tabla 2

Los resultados revelan el buen comportamiento de la propuesta presentada, en comparación con el enfoque clásico de la literatura que usa la función heurística de selección de nodos bloqueadores basada en el grado del nodo. En términos de la calidad de las soluciones, *GRASP OIM* minimiza eficazmente la función objetivo en la totalidad de las instancias evaluadas. Otro aspecto a considerar es la robustez estocástica de nuestra propuesta, la cual registra una desviación porcentual del 0.00 % en todos los escenarios. Este comportamiento contrasta drásticamente con la alta inestabilidad de *GRASP GRADO*, cuya desviación promedio global se eleva al 206.60 %, registrando picos críticos de dispersión en redes topológicamente complejas como *web-Stanford* y *com-dblp*.

En lo que respecta al desempeño computacional, los datos evidencian un compromiso estructural entre el tiempo de ejecución y la precisión heurística. Si bien

| *IC* | **GRASP OIM** | | | | **GRASP GRADO** | | | |
|---|---|---|---|---|---|---|---|---|
| | FO | Desv. (%) | Tiempo (s) | Mejor | FO | Desv. (%) | Tiempo (s) | Mejor |
| com-dblp | **24.48** | **0.00** | 41669.65 | **4/4** | 112.77 | 663.01 | **120.23** | 0/4 |
| com-youtube | **12755.96** | **0.00** | **8558.25** | **4/4** | 14275.07 | 17.24 | 10456.15 | 0/4 |
| Email-EuAll | **1202.03** | **0.00** | 190.60 | **4/4** | 1392.03 | 30.11 | **17.02** | 0/4 |
| email-Eu-core | **279.77** | **0.00** | 5.85 | **4/4** | 281.30 | 0.68 | **5.62** | 0/4 |
| facebook_combined | **366.94** | **0.00** | 12.61 | **4/4** | 392.96 | 8.43 | **11.79** | 0/4 |
| twitter_combined | **7528.69** | **0.00** | 1227.62 | **4/4** | 10047.48 | 58.79 | **453.73** | 0/4 |
| web-Stanford | **37.17** | **0.00** | 30974.13 | **4/4** | 146.38 | 866.14 | **264.13** | 0/4 |
| Wiki-Vote | **577.77** | **0.00** | **15.31** | **4/4** | 613.86 | 8.42 | 19.43 | 0/4 |
| **Resumen** | **2846.60** | **0.00** | 10331.75 | **32/32** | 3407.73 | 206.60 | **1418.51** | 0/32 |

Cuadro 2: Comparación final con $\alpha = RND$ y Surrogate 25.

*GRASP GRADO* demanda un menor tiempo de cómputo global, esta ventaja operativa se consigue a expensas de una degradación considerable en la calidad de los resultados, siendo este algoritmo incapaz de obtener la mejor solución en ninguna de las pruebas. Por tanto, el incremento en el coste computacional requerido por *GRASP OIM* queda plenamente justificado por su capacidad para mitigar de forma óptima el impacto de la influencia en la red con una consistencia matemática absoluta. Este hecho posiciona a nuestra configuración como la alternativa ideal para escenarios donde la fiabilidad y la optimización estricta de la solución resulten críticas.

## 5. Conclusiones y trabajos futuros

A la vista de la experimentación realizada y bajo la mejor combinación encontrada, nuestra implementación *GRASP* para el problema de *IMIN con presupuesto* `FR` obtiene soluciones de alta calidad con respecto a la puntuación de nodos basada en el grado, consiguiendo un 206.60 % de desviación porcentual. Además, se consigue un buen equilibrio entre calidad de solución y tiempo de cómputo, uno de los retos principales para este tipo de problemas tan costosos.

En futuras líneas de investigación, se contempla profundizar en la precisión de las estimaciones mediante un incremento significativo en el número de réplicas de Monte Carlo, permitiendo una convergencia más estrecha de la propagación esperada y una reducción del error típico en redes de gran escala. Además, se pretende realizar un estudio comparativo frente a otras metaheurísticas, tanto de naturaleza heurística como basadas en técnicas de muestreo inverso. De esta manera podremos contrastar la eficiencia del enfoque *GRASP* bajo restricciones presupuestarias heterogéneas.

**Agradecimientos** Este trabajo ha sido posible gracias al apoyo del MICIU/AEI/ 10.13039/501100011033 y FEDER/UE (subvenciones con referencia PID2024-156434NB-

I00 (CONFIA2) y PID2024-156045NB-I00), al programa de Empleo Juvenil y la Comunidad Autónoma de Madrid (subvención con referencia TEC-2024/COM-404).

**Declaración de intereses** Los autores no tienen ningún conflicto de intereses que declarar que sea relevante para el contenido de este artículo.

## Referencias

1. Bian, S., Guo, Q., Wang, S., Yu, J.X.: Efficient algorithms for budgeted influence maximization on massive social networks. Proceedings of the VLDB Endowment **13**(9), 1498–1510 (2020).
2. Budak, C., Agrawal, D., El Abbadi, A.: Limiting the spread of misinformation in social networks. In: Proceedings of the 20th international conference on World wide web. p. 665–674. WWW '11, ACM (Mar 2011).
3. Domingos, P., Richardson, M.: Mining the network value of customers. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 57–66 (2001).
4. Feo, T.A., Resende, M.G.: Greedy randomized adaptive search procedures. Journal of global optimization **6**(2), 109–133 (1995).
5. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 137–146 (2003).
6. Laguna, M., Martí, R., Martínez-Gavara, A., Pérez-Peló, S., Resende, M.G.C.: Greedy randomized adaptive search procedures with path relinking. an analytical review of designs and implementations. European Journal of Operational Research **327**(3), 717–734 (2025).
7. Laguna, M., Martí, R., Martinez-Gavara, A., Perez-Peló, S., Resende, M.G.: 20 years of greedy randomized adaptive search procedures with path relinking. arXiv preprint arXiv:2312.12663 (2023).
8. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective outbreak detection in networks. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 420–429 (2007).
9. Lozano-Osorio, I., Sánchez-Oro, J., Duarte, A., Cordón, O.: A quick grasp-based method for influence maximization in social networks. Journal of Ambient Intelligence and Humanized Computing **14**(4), 3767–3779 (Sep 2021).
10. Lozano-Osorio, I., Sánchez-Oro, J., Sörensen, K.: Determining good solutions and validating them with a metaheuristic approach in social network influence minimization problems. European Journal of Operational Research **330**(1), 257–267 (Apr 2026).
11. Resende, M.G., Ribeiro, C.C.: Grasp: Greedy randomized adaptive search procedures. In: Search methodologies: introductory tutorials in optimization and decision support techniques, pp. 287–312. Springer (2013).
12. Tang, Y., Xiao, X., Shi, Y.: Influence maximization: Near-optimal time complexity meets practical efficiency. In: Proceedings of the 2014 ACM SIGMOD international conference on Management of data. pp. 75–86 (2014).
13. Xie, J., Zhang, F., Wang, K., Liu, J., Lin, X., Zhang, W.: Influence minimization via blocking strategies. INFORMS Journal on Computing **37**(6), 1587–1604 (2025).