

¿Cómo obtener soluciones heurísticas buenas?

Caso de estudio en problemas de minimización de la influencia en redes sociales

Isaac Lozano-Osorio

Universidad Rey Juan Carlos
Madrid, España
isaac.lozano@urjc.es

Jesús Sánchez-Oro

Universidad Rey Juan Carlos
Madrid, España
jesus.sanchezoro@urjc.es

Abraham Duarte

Universidad Rey Juan Carlos
Madrid, España
abraham.duarte@urjc.es

Kenneth Sörensen

Universidad de Amberes
Amberes, Bélgica
kenneth.sorensen@uantwerpen.be

Resumen—La evolución de las redes sociales ha introducido importantes retos relacionados con la sobrecarga de información. Estos retos abarcan diversos ámbitos, como el marketing viral, la gestión de enfermedades y el control de la desinformación. A medida que las redes sociales crecen en complejidad, se hace evidente la necesidad esencial de aprovechar los conocimientos basados en datos. El objetivo del Problema de Minimización de la Influencia en redes sociales (PMI), consiste en identificar y bloquear estratégicamente a los usuarios para restringir la difusión de información. Las técnicas de minería de datos, permiten extraer conocimientos estructurales que guíen el diseño de heurísticas eficientes y la identificación de usuarios influyentes a los que bloquear. Considerando soluciones de buena y mala calidad, un enfoque de aprendizaje supervisado permite clasificar las características extraídas y deducir conclusiones significativas sobre las características de estas soluciones. Para resolver el PDI se propone un método heurístico robusto, basado en las características más relevantes, que resulta eficaz y eficiente en comparación con los enfoques más avanzados.

Index Terms—Conocimiento específico del problema, Minería de datos, Influencia en redes sociales, Heurísticas.

I. INTRODUCCIÓN

Los problemas de Influencia en Redes Sociales (IRS) [1] tienen como objetivo elegir estratégicamente a los usuarios para optimizar un criterio específico, como maximizar o minimizar la influencia, cumplir con un determinado presupuesto, etc. La literatura presenta diversas variantes de IRS, cada una definida por los criterios de selección aplicados a los usuarios y las restricciones consideradas [2]. Estudios recientes destacan que estos problemas se han abordado principalmente a partir de heurísticas voraces, simples, pero eficaces, pero los enfoques metaheurísticos son escasos en este campo [2, 3].

Este trabajo aborda una variante del Problema de Minimización de la Influencia (PMI) [4] que, dada una Red Social (RS) y un conjunto de Nodos Maliciosos (NM), tiene como objetivo minimizar la propagación de la influencia mediante la selección de un conjunto de nodos bloqueantes (B), es decir, nodos especiales que son capaces de bloquear la propagación de la influencia. Hasta donde sabemos, el mejor algoritmo

propuesto para resolver este problema se denomina *Greedy Replace* (GR) [4] y consiste en un algoritmo voraz basado en árboles dominadores y relaciones entre candidatos, que supera los resultados obtenidos por investigaciones anteriores sobre este problema [5].

El interés por resolver el problema PMI se debe a que hay muchos problemas realistas que se ajustan a este modelo. En las redes sociales, los usuarios encuentran tanto información positiva (ideas innovadoras) como negativa (rumores, ciencia falsa) [6, 7]. La desinformación, especialmente los rumores, se propagan rápidamente en las redes sociales, con posibles consecuencias adversas como: el aumento de la oposición a la vacunación contra el SARS-CoV-2 o provocando importantes pérdidas económicas debido a los falsos rumores [4, 8].

El documento se organiza como sigue: En la sección II se define el problema de la influencia de las redes sociales que se aborda en este trabajo. Seguidamente, la sección III describe el algoritmo propuesto y las nuevas estrategias que hemos implementado para resolver el PMI. La sección IV incluye los resultados computacionales. Por último, las conclusiones y la investigación futura se discuten en la sección V.

II. DESCRIPCIÓN DEL PROBLEMA

Según la literatura [9], en esta familia de problemas hay dos estrategias principales para abordarlos, lo que da lugar a dos tipos de problemas bien diferenciados: el primer enfoque implica una estrategia de bloqueo, en la que se obstruyen o eliminan nodos o aristas para disminuir el flujo de información errónea dentro de la red; el segundo es una estrategia de contrarrestar el mensaje, que implica difundir información verdadera para aumentar la concienciación de los usuarios, reduciendo así la aceptación y propagación de la información errónea [3]. En esta investigación se considera el primer enfoque.

Evaluar la influencia en una red social implica definir un Modelo de Difusión de Influencia (MDI) [10]. El MDI determina cómo los nodos de la RS se ven afectados por la información de sus vecinos. Algunos MDIs conocidos son: *Weighted Cascade Model* (WCM) [1] y el *Tri-Valency Model* (TV) [11]. Estos se apoyan en la misma base: asignar probabilidades de

influencia a las relaciones (aristas) de una RS. En concreto, WCM establece la probabilidad en función de la inversa del grado de entrada de un usuario v , es decir, $1/d_{in}(v)$, mientras que TV selecciona aleatoriamente las probabilidades sobre el conjunto (1% – 0,1% – 0,001%), estos valores son los más usados en la literatura. Debido a la naturaleza probabilística de los MDI, la forma más extendida de evaluar la difusión de la influencia es realizando varias iteraciones de la simulación de Monte Carlo (MC) [10], véase más información detallada en el Algoritmo 1. En esta aproximación ambos MDIs han sido empleados, la motivación es realizar una comparativa justa, ya que los dos han sido usados en el trabajo previo.

Formalmente, una RS se modela mediante un grafo $G = (V, A)$, donde el conjunto de vértices V representa a los usuarios, y el conjunto de aristas A indica las relaciones entre usuarios en la RS. Considerando un determinado modelo de difusión de la influencia ψ , un conjunto de Nodos Maliciosos (NM), con $|NM| \geq 1$, que pretenden propagar desinformación en una red social, mientras que un conjunto de bloqueadores B se encargará de reducir la propagación de desinformación ($B \subseteq V \setminus NM$, con $|B| = b$), donde b es una restricción fija. La influencia de la desinformación está determinada por el número de usuarios que aceptan la información errónea durante el proceso de difusión. El objetivo de PMI es seleccionar un conjunto de bloqueadores para minimizar el número de nodos activados. El PMI se ha demostrado en [4] ser \mathcal{NP} -difícil.

Por lo tanto, dada una solución S , la función objetivo de PMI se evalúa de la siguiente manera:

$$PMI(S) \leftarrow \arg \min_{B \subseteq V \setminus NM} \varphi_{\psi}(G, \{NM, B\}) \quad (1)$$

donde $\varphi_{\psi}(G, \{NM, B\})$ representa la capacidad de difusión de NM cuando ambos conjuntos NM y B difunden dos mensajes opuestos. Los usuarios que reciban información verdadera no aceptarán la desinformación y no la reenviarán más en la red, reduciendo así la propagación de la desinformación. Cabe mencionar que los NM no se tiene en cuenta como nodos que aceptan la desinformación. A continuación, PMI pretende encontrar una solución con el valor mínimo de la función objetivo.

El algoritmo 1 muestra el método de Monte Carlo, empleado para estimar el número de usuarios activados teniendo en cuenta los nodos bloqueadores. El algoritmo requiere cinco parámetros de entrada: la red social representada como un grafo $G = (V, A)$, el conjunto de nodos maliciosos NM, el criterio MDI representado por ψ , los nodos bloqueantes seleccionados B y, el número de iteraciones ev .

El algoritmo comienza inicializando el conjunto que almacena el número de usuarios influenciados (paso 1). A continuación, realiza un número de iteraciones predefinidas ev (pasos 2 – 18), en cada iteración se detectan los nodos influenciados por el conjunto de NM. Inicialmente, el conjunto de nodos activados R^* incluye al conjunto de NM (paso 3). A continuación, el método itera hasta que no hay nuevos nodos influenciados (pasos 5-16). En cada iteración del bucle interno, se recorren los vecinos de cada nodo (pasos 8-12). Para cada

Algoritmo 1 $MC(G = (V, A), NM, \psi, B, ev)$

```

1:  $I \leftarrow \emptyset$ 
2: for  $i \in 1 \dots ev$  do
3:    $R^* \leftarrow NM$ 
4:    $R \leftarrow NM$ 
5:   while  $R \neq \emptyset$  do
6:      $B \leftarrow \emptyset$ 
7:     for  $v \in R$  do
8:       for  $(u, v) \in A$  do
9:         if  $\psi$  and  $u \notin B$  then
10:            $C \leftarrow C \cup \{u\}$ 
11:         end if
12:       end for
13:     end for
14:      $R^* \leftarrow R^* \cup C$ 
15:      $R \leftarrow C$ 
16:   end while
17:    $I \leftarrow I + |R^*|$ 
18: end for
19: return  $I/ev$ 

```

vecino, se realiza el criterio específico ψ , si es válido y no pertenece a un nodo bloqueante, el nodo vecino es influenciado (pasos 9-11). Al final, se actualiza el conjunto de nodos influenciados (paso 14) así como los nodos influenciados en la iteración anterior (paso 15). Finalmente, el algoritmo devuelve el número medio de nodos influenciados entre todas las simulaciones realizadas (paso 19). Nótese que este valor se considera como la función objetivo a optimizar. Cabe mencionar que, al ser la influencia un proceso estocástico, el MC debe ejecutarse varias veces (ev en nuestro caso) para conseguir una estimación adecuada.

La Figura 1 ilustra el PMI con un ejemplo que considera ambos MDI. La Figura se representa mediante un gráfico con 8 nodos y 9 aristas, cada arista denota dos valores según los respectivos MDI. El valor de TV se indica mediante un número seguido de un símbolo de porcentaje, y el WCM se representa como una división. El número de nodos bloqueadores se fija en 2. Los NM se resaltan en gris, mientras que un bloqueador seleccionado específicamente se resalta en negro. Para favorecer la simplicidad, se supone que se activarán todos los nodos con un valor de TV igual o superior al 0.1% en ambas figuras.

En la Figura 1a se seleccionan los bloqueadores $S_1 = \{B, F\}$. Obsérvese que B es un nodo que no tiene aristas de salida, por lo que solo podrá bloquearse a sí mismo. F es un nodo interno que bloquea H, pero H también puede ser activado por G. También es notable que F solo puede ser activado por E, por lo que si E está bloqueado, entonces F nunca se activará. En este caso, el valor de la función objetivo es $PMI(S_1) = 3$ ya que los nodos $\{D, E, H\}$ tienen un valor de TV igual o superior al 0,1% y, por lo tanto, se activan.

La Figura 1b muestra otra solución factible, $S_2 = \{E, D\}$. En esta solución, como se dijo antes, E se bloquea a sí mismo y a F, y D, que tiene dos aristas entre NM, se bloquea entonces,

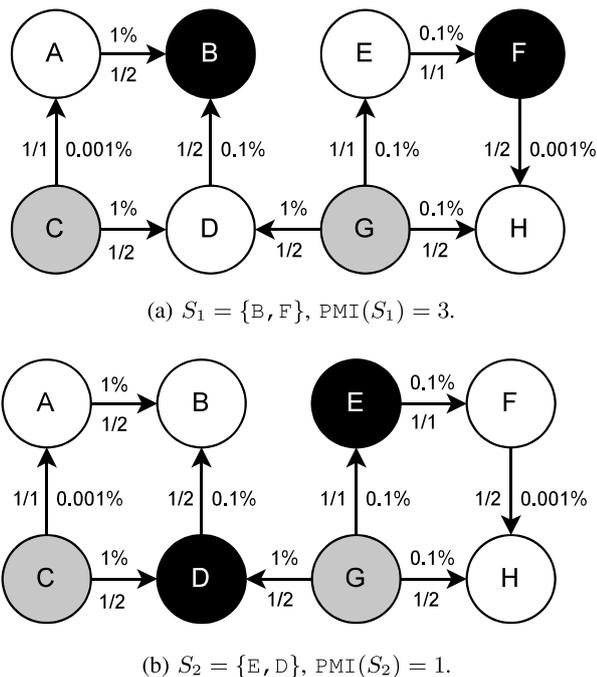


Figura 1: Ejemplo de 2 posibles soluciones con 2 nodos bloqueadores (resaltados en negro) y 2 NM (resaltados en gris).

impidiendo la activación de B. En este caso, el valor de la función objetivo es $\text{PMI}(S_2) = 1$ ya que solo se activará H, siendo S_2 la solución óptima para este caso.

III. DE LOS DATOS AL CONOCIMIENTO ESPECÍFICO DEL PROBLEMA

La selección de características clave que incluye una buena solución es una parte crítica de esta investigación [12]. El objetivo de esta tarea es comprender los aspectos que definen una buena solución. Para ello, es necesario responder a la siguiente pregunta: ¿Qué distingue una buena de una mala solución? La identificación de esas propiedades distintivas podría ayudar a encontrar esas buenas soluciones con mayor eficacia, que es el objetivo de toda heurística. Así pues, es necesario calcular y comparar soluciones de distinta calidad para aprender de ellas [13].

Un estudio exhaustivo basado en la minería de datos requiere un número suficiente de datos. Entonces, estas características se derivan de una solución para una instancia y, por lo tanto, para la generación de un conjunto de datos, necesitamos calcular un conjunto de un número considerablemente grande de soluciones en un tiempo de computación razonable.

Las características de una solución se obtienen evaluando diferentes métricas sobre ella, por lo que es necesario transformar la estructura de una solución en características cuantitativas. Las características generadas servirán entonces como entrada del modelo predictivo. Este paso es altamente exploratorio, ya que no existen directrices sobre qué métricas deben incluirse, y suele depender del problema.

En esta investigación, las características seleccionadas para cada solución se definen a partir de métricas bien conocidas en análisis de redes sociales, ya que son buenas candidatas para caracterizar una solución. Dado que los valores de las métricas seleccionadas dependen de la instancia en la que se evalúan, sería difícil comparar soluciones generadas para instancias diferentes sin normalización (por ejemplo, si comparamos una red social con 100000 nodos, es probable que el grado de salida sea mucho mayor que en una red social con 10 nodos debido a su escala). Para la normalización se han utilizado las siguientes métricas: número de nodos (I1), número de aristas (I2), número de componentes débiles totalmente conectadas (I3), grado de entrada (I4) y grado de salida (I5) medio.

Las características consideradas para caracterizar una solución en este trabajo son los siguientes (si la característica se debe normalizar, se indicaría con el nombre de la métrica de normalización entre paréntesis): (S1) nodos que pueden ser influenciados (I1); (S2) aristas que pueden propagar influencia (I2), (S3) total de componentes débiles conectadas influenciadas (I3); (S4) promedio de grado de entrada de los bloqueadores (I4); (S5) promedio de grado de salida de los bloqueadores (I5); (S6) suma de ranking de bloqueadores en la instancia según grado de salida (I1); (S7) promedio de la probabilidad influencia de los bloqueadores a sus vecinos; (S8) probabilidad media de activación de NM a bloqueadores; (S9) suma de distancia mínima de bloqueadores hacia NM.

Una vez definidas las características, es necesario generar un conjunto de datos representativo de soluciones de buena y mala calidad para entrenar un clasificador. Las soluciones de mala calidad (MC) se crean mediante una selección aleatoria de bloqueadores. Sin embargo, las de buena calidad (BC) son las más difíciles de generar. En este trabajo, hemos considerado un método del estado del arte basado en el seleccionar los nodos con mayor grado de salida para generar estas soluciones. El método ha sido modificado, incluyendo cierta aleatorización para generar un conjunto diverso de soluciones, este método no siempre selecciona el nodo con mayor grado de salida, puede descartarlo según un parámetro aleatorio. De cara a tener un conjunto de datos amplio, se han generado 100 soluciones por instancia de cada uno de los tipos, véase la Tabla I.

	MDI	$ B $	Promedio BC	Promedio MC
WCM (1)	20		10077.73	10681.29
	40		9748.10	10676.86
	60		9506.52	10669.80
	80		9301.19	10665.14
	100		9120.10	10658.45
TV (1)	20		15707.57	18496.34
	40		14711.78	18488.77
	60		13637.40	18479.56
	80		13059.29	18472.95
	100		12674.92	18463.24

Tabla I: Promedio de función objetivo con 100 soluciones por instancia de buena y mala calidad.

La anterior tabla muestra como en ambos casos, con mayor cantidad de nodos bloqueadores, los nodos influenciados se

reducen. Es importante remarcar, que comparando el promedio entre ambos tipos de soluciones, se observa que las soluciones de buena calidad obtienen mejor función objetivo en promedio frente a todas las de mala calidad.

El siguiente paso consiste en seleccionar un clasificador para aprender del conjunto de datos. En este trabajo preliminar, el clasificador seleccionado es un árbol de decisión [14], capaz de realizar una clasificación rápida del conjunto de datos, proporcionando algunas pistas sobre la relevancia de cada característica. La precisión de predicción obtenida con este clasificador es del 99,87 %.

La Figura 2 muestra los resultados del algoritmo supervisado (árbol de decisión) sobre el conjunto de datos generado.

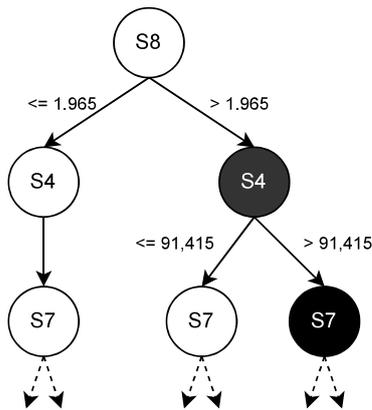


Figura 2: Clasificación mediante el árbol de decisión.

En la figura se muestra un nodo representando la característica (según el identificador) y dos aristas con la clasificación según el valor. Cada arista lleva a una solución buena (nodos negros) y solución mala (nodos blancos) según el valor de las aristas.

Analizando la figura, las características seleccionadas son: la suma media de la probabilidad (S8) y la suma media del grado de entrada (S4) de los nodos bloqueadores a sus vecinos, siendo estas características relevantes para el PMI.

La heurística propuesta g_{in} selecciona $|B|$ nodos que no sean NM, en orden decreciente por grado de entrada (S4) y que la suma media de propagación de la influencia de los vecinos sea superior a 1,965 (S8).

De cara a comprobar computacionalmente los resultados, la siguiente sección compara el mejor método en el estado del arte (GR) para este problema frente a la heurística propuesta (g_{in}). Ambos métodos utilizan solo una llamada al método de simulación de MC.

IV. RESULTADOS COMPUTACIONALES

Esta sección está dedicada a proporcionar un análisis detallado del rendimiento del algoritmo propuesto. Todos los experimentos se han realizado en una CPU virtual AMD EPYC 7282 de 16 núcleos y 32 GB de RAM, utilizando Java 17 y el *Metaheuristic Optimization framework* (MORK) 17 [15].

El conjunto de instancias utilizado en este trabajo es el mismo conjunto considerado en el trabajo anterior, que se deriva del conocido Stanford Network Analysis Project (SNAP). Como se indica en [4], el número de iteraciones para las simulaciones Monte Carlo es de 10000. La Tabla II muestra las instancias utilizadas.

Instancia	Nodos	Aristas	Componentes
email-Eu-core	1005	25571	41
facebook_combined	4039	88234	1
Wiki-Vote	7115	103689	4742
twitter_combined	81306	1768149	2
Email-EuAll	265214	420045	190821
web-Stanford	281903	2312497	22527
com-dblp.ungraph	317080	1049866	1
com-youtube.ungraph	1134890	2987624	1

Tabla II: Métricas de las instancias utilizadas.

Cada una de las instancias es ejecutada para cada uno de los dos MDI (WCM y TV) y con diferentes cantidades de nodos bloqueadores $B = \{20, 40, 60, 80, 100\}$. Resultando en un total de $2 \cdot |B| \cdot 8 = 80$ instancias.

Se han detectado los NM utilizando el trabajo [16], donde los autores encuentran los k nodos más influyentes de una red social según un MDI.

Las métricas de rendimiento representadas en la Tabla III son las siguientes: el valor medio de la función objetivo, Promedio; el tiempo medio de ejecución del algoritmo medido en segundos, Tiempo(s); la desviación media respecto a la mejor solución, Des. (%); y, por último, el número de veces que el algoritmo es capaz de alcanzar la mejor solución en el experimento (#B).

La Tabla III muestra los resultados al comparar ambos enfoques. En términos de desviación, g_{in} reporta 0,02 % y 0,00 % frente a 0,34 % y 0,00 % del GR en un tiempo de computación similar (80,64 y 592,81 segundos en el estado del arte frente a 81,34 y 559,18 segundos en él g_{in}). Ambos tienen un tiempo de computación similar, puesto que el método que consume más tiempo es la simulación de MC y en ambos enfoques son voraces, por lo que solo se necesitan una vez obtener la función objetivo. Analizando el número de mejores soluciones encontradas, g_{in} es capaz de alcanzar 76 de las 80 instancias disponibles, mientras que GR obtiene 70 de las 80 mejores soluciones.

Los resultados muestran tanto en tiempo como en calidad unos valores similares, la única diferencia se muestra en el MDI de WCM dónde la propuesta de este artículo obtiene 6 mejores valores con una desviación menor.

Es importante destacar, que el método basado en el conocimiento extraído del problema mediante minería de datos y aprendizaje supervisado. Obtienen un nuevo método heurístico que es competitivo con el actual estado del arte.

		GR				<i>Gin</i>			
		Promedio	Tiempo(s)	Des. (%)	#B	Promedio	Tiempo(s)	Des. (%)	#B
WCM (1)	20	10077,13	28,40	0,02	7	10077,02	28,34	0,00	8
	40	9750,55	55,49	0,01	7	9750,50	55,80	0,03	7
	60	9513,50	81,54	0,34	6	9512,33	81,99	0,05	7
	80	9301,72	106,63	0,44	6	9299,53	107,71	0,03	7
	100	9137,11	131,14	0,91	5	9132,98	132,87	0,00	8
		9556,00	80,64	0,34	31	9554,47	81,34	0,02	37
TV (2)	20	15716,42	211,56	0,00	8	15716,42	199,17	0,00	8
	40	14979,66	413,64	0,00	8	14979,66	387,90	0,00	8
	60	14465,62	604,81	0,00	8	14465,62	569,66	0,00	8
	80	13411,32	781,76	0,01	7	13411,30	739,35	0,00	8
	100	13004,12	952,28	0,00	8	13004,21	899,81	0,01	7
		14315,43	592,81	0,00	39	14315,44	559,18	0,00	39

Tabla III: Resultados de la comparación entre el método más avanzado y el enfoque heurístico basado en las mejores características según el algoritmo supervisado.

V. CONCLUSIONES

Este estudio profundiza en la forma de determinar la calidad de las soluciones al problema del PMI sin evaluar la función objetivo, que es muy exigente desde el punto de vista computacional y requiere múltiples iteraciones del algoritmo de Monte Carlo. Para ello, se han extraído las características de la RS y se han analizado mediante un algoritmo de aprendizaje supervisado que ha permitido extraer conclusiones significativas sobre las características de estas soluciones. De acuerdo con las características principales, se ha desarrollado un método basado en la difusión de la probabilidad de los vecinos y en el promedio de grado de entrada que supera el estado del arte. Como trabajo futuro, se pretende explorar algoritmos metaheurísticos como *Greedy Randomized Adaptive Search Procedure* (GRASP) [17] para mejorar la calidad de las soluciones. Por último, se ampliará el análisis incorporando algoritmos adicionales de aprendizaje supervisado para profundizar en la comprensión de las soluciones y reforzar las conclusiones.

REFERENCIAS

- [1] David Kempe, Jon Kleinberg y Éva Tardos. «Maximizing the spread of influence through a social network». En: *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, ago. de 2003, págs. 137-146. DOI: 10.1145/956750.956769.
- [2] Suman Banerjee, Mamata Jenamani y Dilip Kumar Pratihari. «A survey on influence maximization in a social network». En: *Knowledge and Information Systems* 62.9 (mar. de 2020), págs. 3417-3455. DOI: 10.1007/s10115-020-01461-4.
- [3] Zahra Aghaee et al. «A survey on meta-heuristic algorithms for the influence maximization problem in the social networks». En: *Computing* 103.11 (sep. de 2021), págs. 2437-2477. DOI: 10.1007/s00607-021-00945-7.
- [4] Jiadong Xie et al. *Minimizing the Influence of Misinformation via Vertex Blocking*. 2023. DOI: 10.1109/icde55515.2023.00066.
- [5] Ceren Budak, Divyakant Agrawal y Amr El Abbadi. «Limiting the spread of misinformation in social networks». En: *Proceedings of the 20th international conference on World wide web*. ACM, mar. de 2011, págs. 665-674. DOI: 10.1145/1963405.1963499.
- [6] Hunt Allcott y Matthew Gentzkow. «Social Media and Fake News in the 2016 Election». En: *Journal of Economic Perspectives* 31.2 (mayo de 2017), págs. 211-236. DOI: 10/gc5t4c.
- [7] Wei Chen, Chi Wang y Yajun Wang. «Scalable Influence Maximization for Prevalent Viral Marketing in Large-Scale Social Networks». En: *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '10. Washington, DC, USA: Association for Computing Machinery, 2010, págs. 1029-1038. ISBN: 9781450300551. DOI: 10.1145/1835804.1835934.
- [8] Fan Xu, Victor S. Sheng y Mingwen Wang. «A Unified Perspective for Disinformation Detection and Truth Discovery in Social Sensing: A Survey». En: *ACM Comput. Surv.* 55.1 (nov. de 2021). ISSN: 0360-0300. DOI: 10.1145/3477138.
- [9] Ahmad Zareie y Rizos Sakellariou. «Minimizing the spread of misinformation in online social networks: A survey». En: *Journal of Network and Computer Applications* 186 (2021), pág. 103094. DOI: 10.1016/j.jnca.2021.103094.
- [10] Isaac Lozano-Osorio, Jesús Sánchez-Oro y Abraham Duarte. «An efficient and effective GRASP algorithm for the Budget Influence Maximization Problem». En: *Journal of Ambient Intelligence and Humanized Computing* (sep. de 2023). DOI: 10.1007/s12652-023-04680-z.

- [11] Mark Granovetter. «Threshold models of collective behavior». En: *American journal of sociology* 83.6 (1978), págs. 1420-1443. DOI: 10.1086/226707.
- [12] Florian Arnold y Kenneth Sörensen. «What makes a VRP solution good? The generation of problem-specific knowledge for heuristics». En: *Computers & Operations Research* 106 (jun. de 2019), págs. 280-288. DOI: 10.1016/j.cor.2018.02.007.
- [13] Galit Shmueli. «To Explain or to Predict?» En: *Statistical Science* 25.3 (2010), págs. 289-310. DOI: 10.1214/10-STS330.
- [14] Sotiris B Kotsiantis. «Decision trees: a recent overview». En: *Artificial Intelligence Review* 39 (2013), págs. 261-283.
- [15] Raúl Martín-Santamaría et al. «A practical methodology for reproducible experimentation: an application to the Double-row Facility Layout Problem». En: *Evolutionary Computation* (nov. de 2022), págs. 1-35. ISSN: 1063-6560. DOI: 10.1162/evco_a_00317.
- [16] Isaac Lozano-Osorio et al. «A quick GRASP-based method for influence maximization in social networks». En: *Journal of Ambient Intelligence and Humanized Computing* (sep. de 2021). DOI: 10.1007/s12652-021-03510-4.
- [17] Thomas A Feo y Mauricio GC Resende. «Greedy randomized adaptive search procedures». En: *Journal of global optimization* 6 (1995), págs. 109-133. DOI: 10.1007/bf01096763.