

# Multi-Round Influence Maximization: A Variable Neighborhood Search Approach

## III Escuela de Invierno

*Isaac Lozano-Osorio (isaac.lozano@urjc.es)*  
*Jesús Sánchez-Oro (jesus.sanchezoro@urjc.es)*  
*Abraham Duarte (abraham.duarte@urjc.es)*



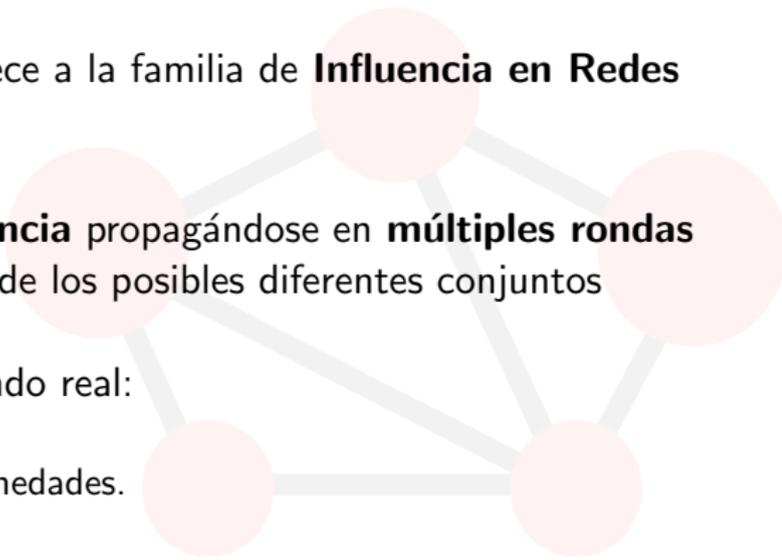
# Índice

- 1 Introducción**
  - Definición del problema
  - Adaptativo vs No Adaptativo
  - Revisión de la literatura
- 2 Variable Neighborhood Search**
  - Constructivo
  - Búsqueda Local
  - Shake
- 3 Resultados**
- 4 Conclusiones**



# Introducción

- El **problema** pertenece a la familia de **Influencia en Redes Sociales**.
- **Maximizar** la **influencia** propagándose en **múltiples rondas** independientemente de los posibles diferentes conjuntos
- Aplicaciones del mundo real:
  - Marketing Viral.
  - Análisis de enfermedades.



# Introducción

## Representación de la solución

La solución consiste en seleccionar conjuntos de semillas  $R$  de tamaño  $l$ , uno para cada ronda, i.e.,  $S = \{S_1, S_2, \dots, S_R\}$ . Obsérvese que, dado que se pueden seleccionar  $l$  nodos en cada ronda, el número total de nodos que se ajustan al conjunto final de semillas  $S$  es igual a  $l \cdot R$ .

El objetivo de MRIM es **maximizar** el número de nodos activos siguiendo un modelo de difusión de influencia (IDM) específico.

# Introducción

## Función Objetivo

El valor de la función objetivo se evalúa entonces como:

$$MRIM(S) = IDM(S_1 \cup S_2 \cup \dots \cup S_R)$$

El objetivo del MRIM es encontrar el conjunto de semillas para cada ronda que maximice el valor de la función objetivo. En términos matemáticos,

$$S^* \leftarrow \arg \max_{S \in \mathbb{S}} MRIM(S)$$

where  $\mathbb{S}$  es el conjunto de todas las combinaciones posibles de conjuntos de semillas para el problema en cuestión.

# Introducción

## Modelo de Difusión de la Influencia

La evaluación de la influencia de un determinado conjunto de semillas  $S$  requiere la definición de un modelo de difusión de la influencia (IDM). Este modelo se encarga de decidir qué nodos se ven afectados por la información recibida de sus nodos vecinos en el SN.

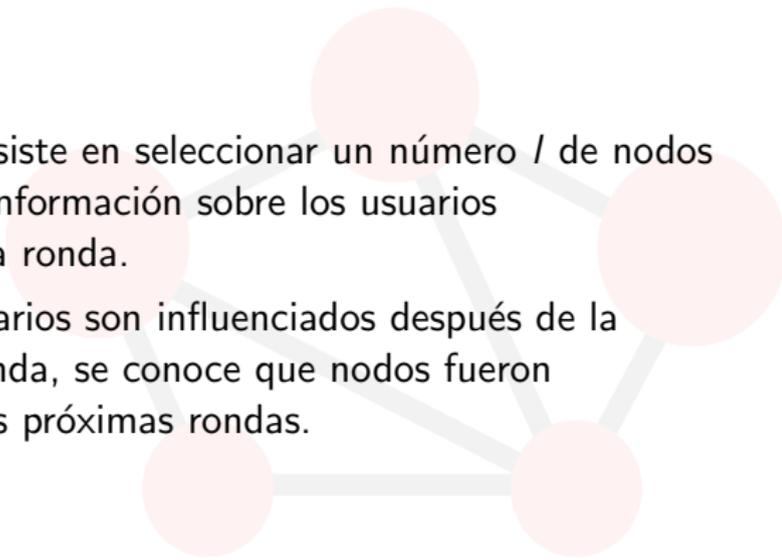
Los IDM más extendidos son:

- Independent Cascade Model (ICM)
- Weighted Cascade Model (WCM)
- Linear Threshold Model (LTM)
- Triggering Model (TM)

Todos ellos se basan en asignar una probabilidad de influencia a cada enlace relacional en la SN.

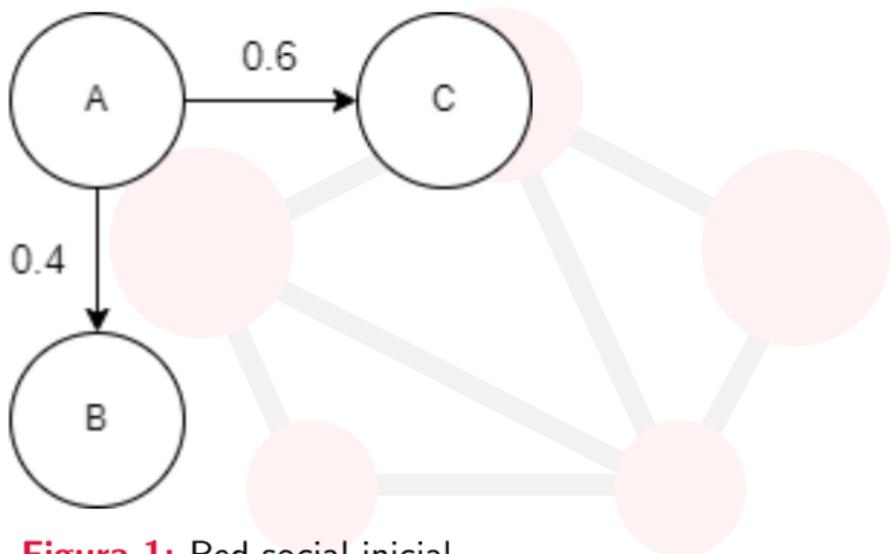
# Introducción

## Adaptativo vs No Adaptativo

- **No Adaptativo** consiste en seleccionar un número  $l$  de nodos por ronda sin tener información sobre los usuarios influenciados en cada ronda.
  - **Adaptativo**, los usuarios son influenciados después de la selección de cada ronda, se conoce que nodos fueron influenciados para las próximas rondas.
- 

# Introducción

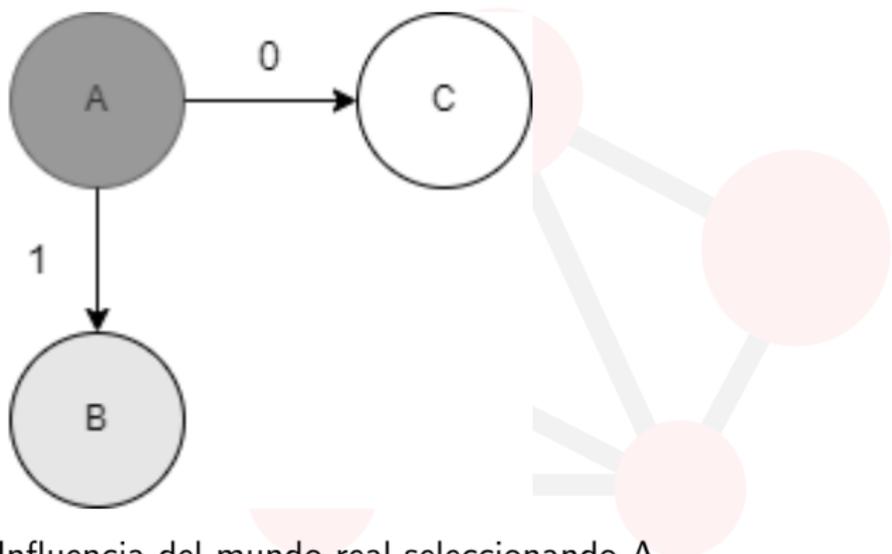
## Adaptativo vs No Adaptativo



**Figura 1:** Red social inicial.

# Introducción

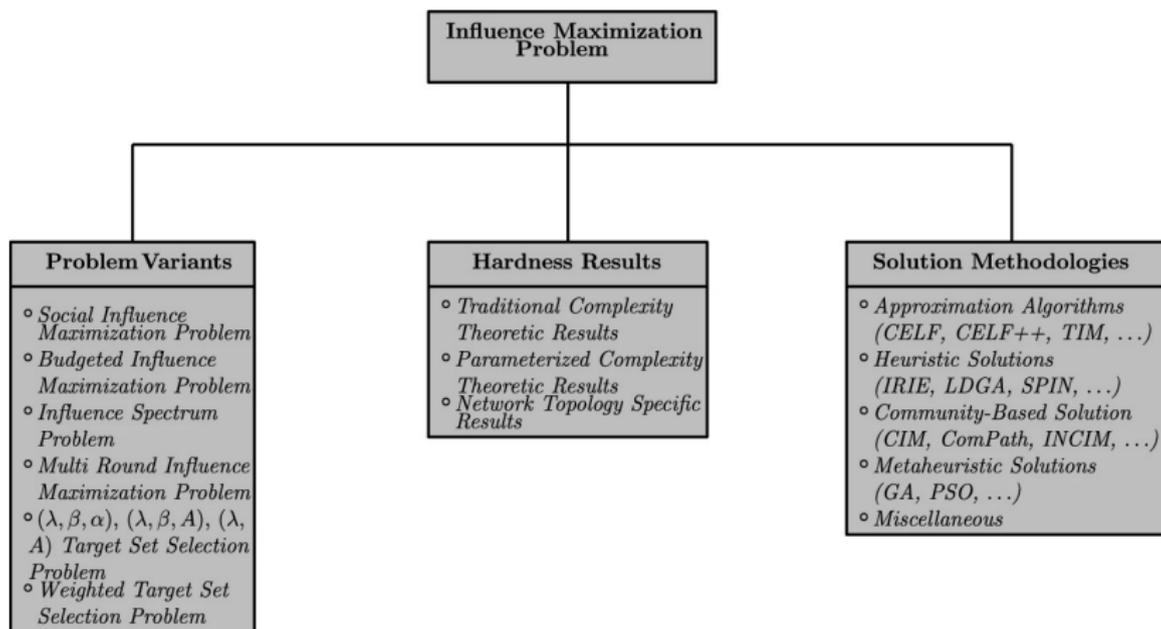
## Adaptativo vs No Adaptativo



**Figura 2:** Influencia del mundo real seleccionando A

# Introducción

## Revisión de la Literatura



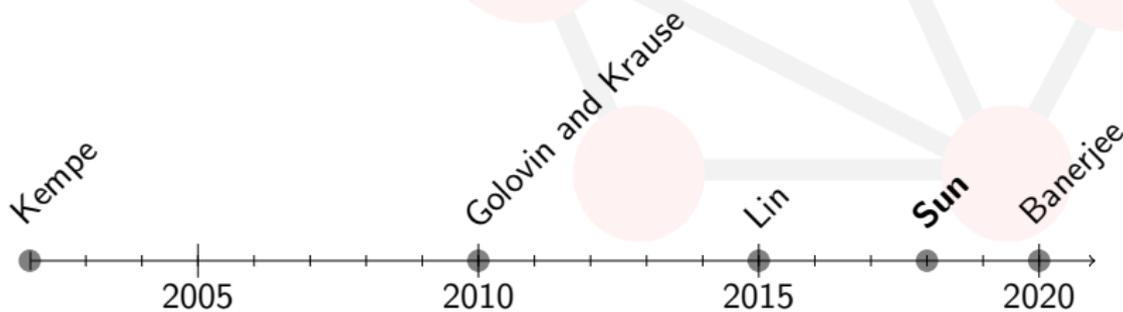
**Figura 3:** Un estudio sobre la maximización de la influencia en una red social, Banerjee et al (2020).

# Introducción

## Revisión de la literatura

Si se considera una sola ronda, es decir,  $R = 1$ , el problema es equivalente al bien estudiado Problema de Maximización de la Influencia de la Red Social (SNIMP).

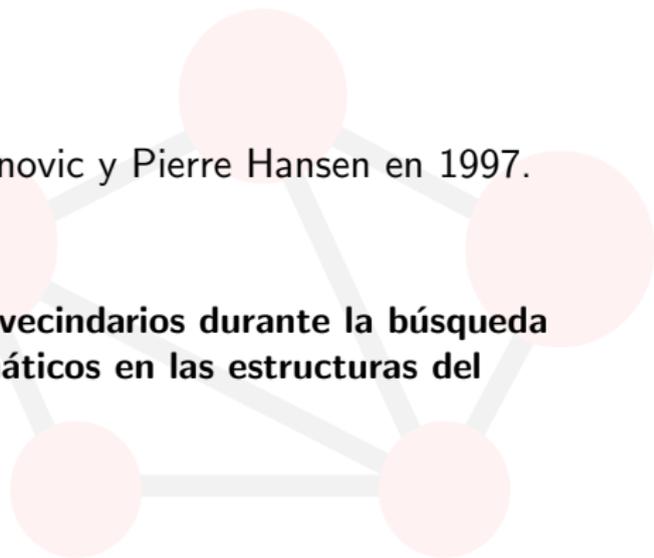
El SNIMP clásico es  $\mathcal{NP}$ -duro (Kempe, 2005), por lo que tanto la versión no adaptativa como la adaptativa de MRIM son también  $\mathcal{NP}$ -duro.



# Propuesta

## Variable Neighborhood Search (VNS)

- Propuesto por Nenad Mladenovic y Pierre Hansen en 1997.
- Contribuciones principales:
  - 1 **Tener en cuenta varios vecindarios durante la búsqueda**
  - 2 **Realizar cambios sistemáticos en las estructuras del vecindario**



# Propuesta

## BVNS

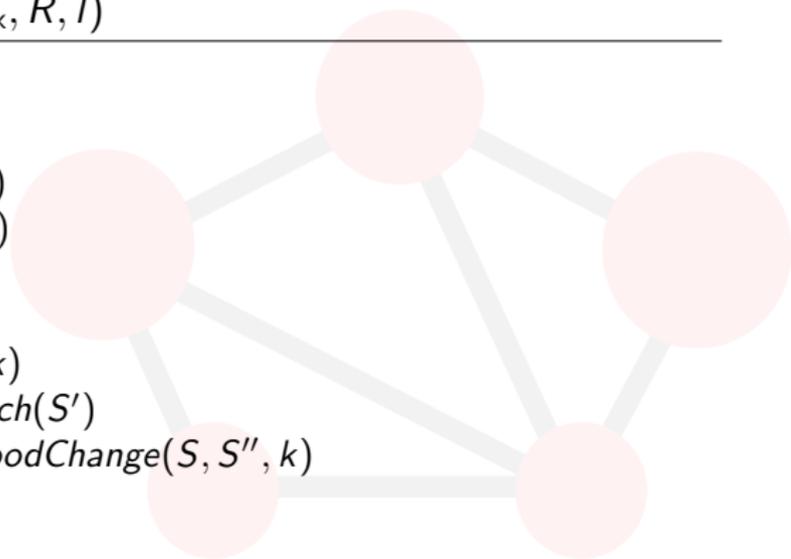
---

**Algorithm 1**  $BVNS(k_{\text{máx}}, R, I)$ 

---

```
1:  $S \leftarrow \emptyset$ 
2: while  $R > 0$  do
3:    $S \leftarrow \text{Construct}(S, I)$ 
4:    $S \leftarrow \text{LocalSearch}(S)$ 
5:    $k \leftarrow 1$ 
6:   while  $k \leq k_{\text{máx}}$  do
7:      $S' \leftarrow \text{Shake}(S, k)$ 
8:      $S'' \leftarrow \text{LocalSearch}(S')$ 
9:      $k \leftarrow \text{NeighborhoodChange}(S, S'', k)$ 
10:  end while
11:   $R \leftarrow R - 1$ 
12: end while
13: return  $S$ 
```

---



# Propuesta

## Constructivo

Algoritmo voraz basado la función objetivo.

- Calcular el número de nodos activados al seleccionar el siguiente nodo semilla  $u$ ,  $MRIM(u)$ .
- Con el objetivo de reducir el esfuerzo computacional, se reduce el número de iteraciones requeridas por Monte Carlo.

La selección de nodos para las siguientes rondas depende de la aproximación considerada.

- Modelo no adaptativo. En el primero, el método selecciona los siguientes  $l$  nodos que son los más influyentes.
- Modelo adaptativo selecciona aquellos que son capaces de influir en un mayor número de nodos no influenciados previamente.

# Propuesta

## Búsqueda Local

El movimiento considerado es un movimiento de intercambio  $Swap(S, u, v)$  en el que el nodo  $u$  es eliminado del conjunto de semillas, siendo reemplazado por  $v$ , con  $u \in S$  y  $v \notin S$ . Este movimiento de intercambio se define formalmente como:

$$Swap(S, u, v) = (S \setminus \{u\}) \cup \{v\}$$

Así, la vecindad  $N_S(S)$  de una solución dada  $S$  consiste en el conjunto de soluciones que se pueden alcanzar desde  $S$  realizando un único movimiento de intercambio. Más formalmente,

$$N_S(S) = \{Swap(S, u, v) \mid \forall u \in S, \forall v \in V \setminus S\}$$

# Propuesta

## Búsqueda Local

El tamaño de la vecindad resultante,  $l \cdot (n - l)$ , hace que la exploración completa de la vecindad no sea adecuada para MRIM, incluso considerando una implementación eficiente de la evaluación de la función objetivo.

A continuación, se sigue la estrategia de exploración de vecindades inteligentes propuesta en Lozano-Osorio 2021, con el objetivo de reducir el número de soluciones exploradas dentro de cada vecindad (y así limitar el número de simulaciones de IDM). Esta reducción del tamaño del espacio de búsqueda se lleva a cabo explorando sólo una pequeña fracción  $\delta$  de los nodos disponibles para el movimiento de intercambio.

# Propuesta

## Shake

El mecanismo de perturbación en la VNS suele denominarse procedimiento Shake.

Proponemos un método que modifica la estructura de la solución en función de un parámetro  $k$ . Su valor va de 1 a  $k_{max}$ , que es un parámetro de entrada del procedimiento completo.

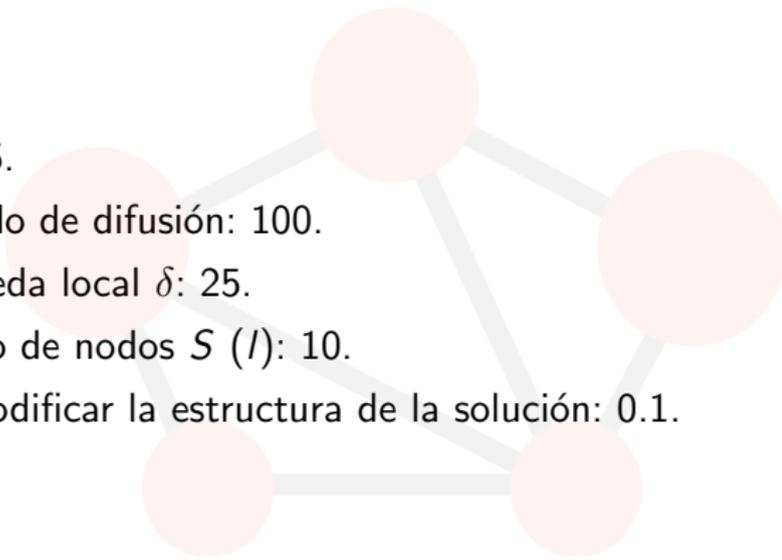
El método de perturbación propuesto realiza  $k$  movimientos de intercambio a la solución inicial. Estos elementos se seleccionan al azar.

# Resultados

- Lenguaje de programación: **Java 17**.
- *Metaheuristic Optimization framewoRK* (MORK) 13.
- Características de la máquina experimental: Intel Core i7-9750H (2.6 GHz) con 16GB RAM.
- Instancias: 2 (Flixter con 95969 nodos y 484865 aristas dirigidas, NetHEPT 15233 nodos y 32235 aristas dirigidas).
- Métricas de rendimiento:
  - **Avg.**: el valor medio de la función objetivo (es decir, el número de nodos influenciados, por término medio, tras 100 simulaciones).
  - **Dev.**: desviación media con respecto a la mejor solución conocida.
  - **Time (s)**: tiempo de ejecución medido en segundos.
  - **#B**: el número de veces que el algoritmo es capaz de alcanzar la mejor solución en el experimento.

# Resultados

## Parámetros utilizados

- Número de rondas: 5.
  - Iteraciones del modelo de difusión: 100.
  - Parámetro de búsqueda local  $\delta$ : 25.
  - Tamaño del conjunto de nodos  $S(I)$ : 10.
  - Parámetro  $k$  para modificar la estructura de la solución: 0.1.
- 

# Resultados

## Versión no adaptativa

		BVNS				AdalMM			
	R	Avg.	Dev.	Time (s)	#B	Avg.	Dev.	Time (s)	#B
NetHEPT	1	<b>355.89</b>	0.00	0.34	1	302.40	15.03	0.21	0
	2	<b>584.66</b>	0.00	0.49	1	544.40	6.89	0.28	0
	3	719.36	5.57	0.61	0	<b>761.80</b>	0.00	0.33	1
	4	900.89	6.72	0.75	0	<b>965.80</b>	0.00	0.51	1
	5	1015.53	11.41	0.82	0	<b>1146.30</b>	0.00	0.60	1
Flixster	1	<b>16637.78</b>	0.00	78.21	1	13560.51	18.50	6.59	0
	2	<b>17964.82</b>	0.00	89.16	1	13349.29	25.69	7.88	0
	3	<b>18288.07</b>	0.00	98.57	1	13655.16	25.33	9.50	0
	4	<b>18603.10</b>	0.00	106.82	1	14033.71	24.56	13.33	0
	5	<b>19120.03</b>	0.00	114.90	1	14820.86	22.49	17.22	0
		<b>9419.01</b>	<b>2.37</b>	49.07	<b>7</b>	7312.96	14.15	<b>5.64</b>	3

**Tabla 1:** Resultados del algoritmo BVNS frente al mejor algoritmo de la literatura en la versión no adaptativa. Los mejores resultados están resaltados en negrita.

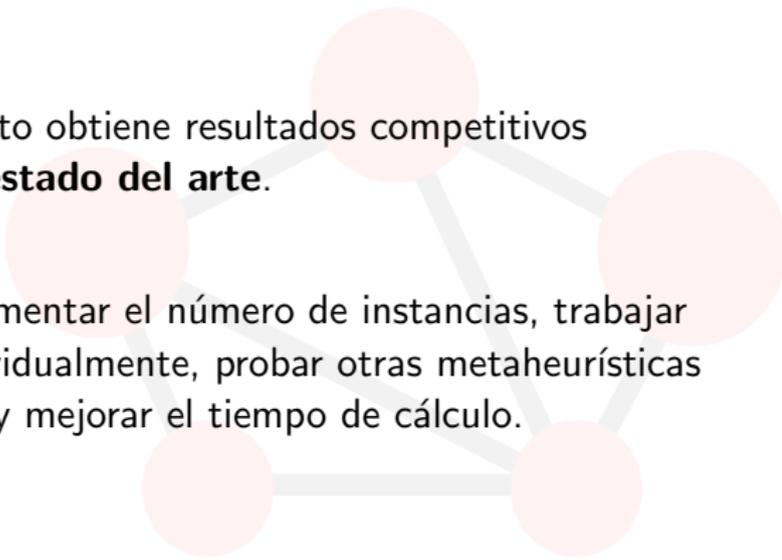
# Resultados

## Versión Adaptativa

		BVNS				AdaIMM			
		Avg.	Dev.	Time (s)	#B	Avg.	Dev.	Time (s)	#B
NetHEPT	R								
	1	<b>355.89</b>	0.00	0.34	1	302.40	15.03 %	0.21	0
	2	<b>584.66</b>	0.00	0.52	1	557.20	4.70 %	0.35	0
	3	<b>806.20</b>	0.00	0.68	1	776.90	3.63 %	0.38	0
	4	913.99	6.93	0.81	0	<b>982.00</b>	0.00 %	0.41	1
5	1021.09	12.24	0.99	0	<b>1163.50</b>	0.00 %	0.56	1	
Flixster	1	<b>16637.78</b>	0.00	78.21	1	13611.37	18.19 %	7.11	0
	2	<b>17831.39</b>	0.00	95.31	1	13749.83	22.89	7.99	0
	3	<b>18364.20</b>	0.00	106.87	1	13855.42	24.55	9.21	0
	4	<b>18929.40</b>	0.00	109.22	1	14213.69	24.91	13.24	0
	5	<b>19310.31</b>	0.00	121.23	1	14863.01	23.03	16.15	0
		<b>9475.49</b>	<b>1.92</b>	51.42	<b>8</b>	7407.53	13.69	<b>5.56</b>	2

**Tabla 2:** Resultados del algoritmo BVNS frente al mejor algoritmo de la literatura en la versión adaptativa. Los mejores resultados están resaltados en negrita.

# Conclusiones

- 
- ✓ El algoritmo propuesto obtiene resultados competitivos comparando con el **estado del arte**.
  - 🔄 **Trabajo Futuro:** aumentar el número de instancias, trabajar en cada versión individualmente, probar otras metaheurísticas como VND, GVNS, y mejorar el tiempo de cálculo.

# Multi-Round Influence Maximization: A Variable Neighborhood Search Approach

## III Escuela de Invierno

*Isaac Lozano-Osorio (isaac.lozano@urjc.es)*  
*Jesús Sánchez-Oro (jesus.sanchezoro@urjc.es)*  
*Abraham Duarte (abraham.duarte@urjc.es)*

