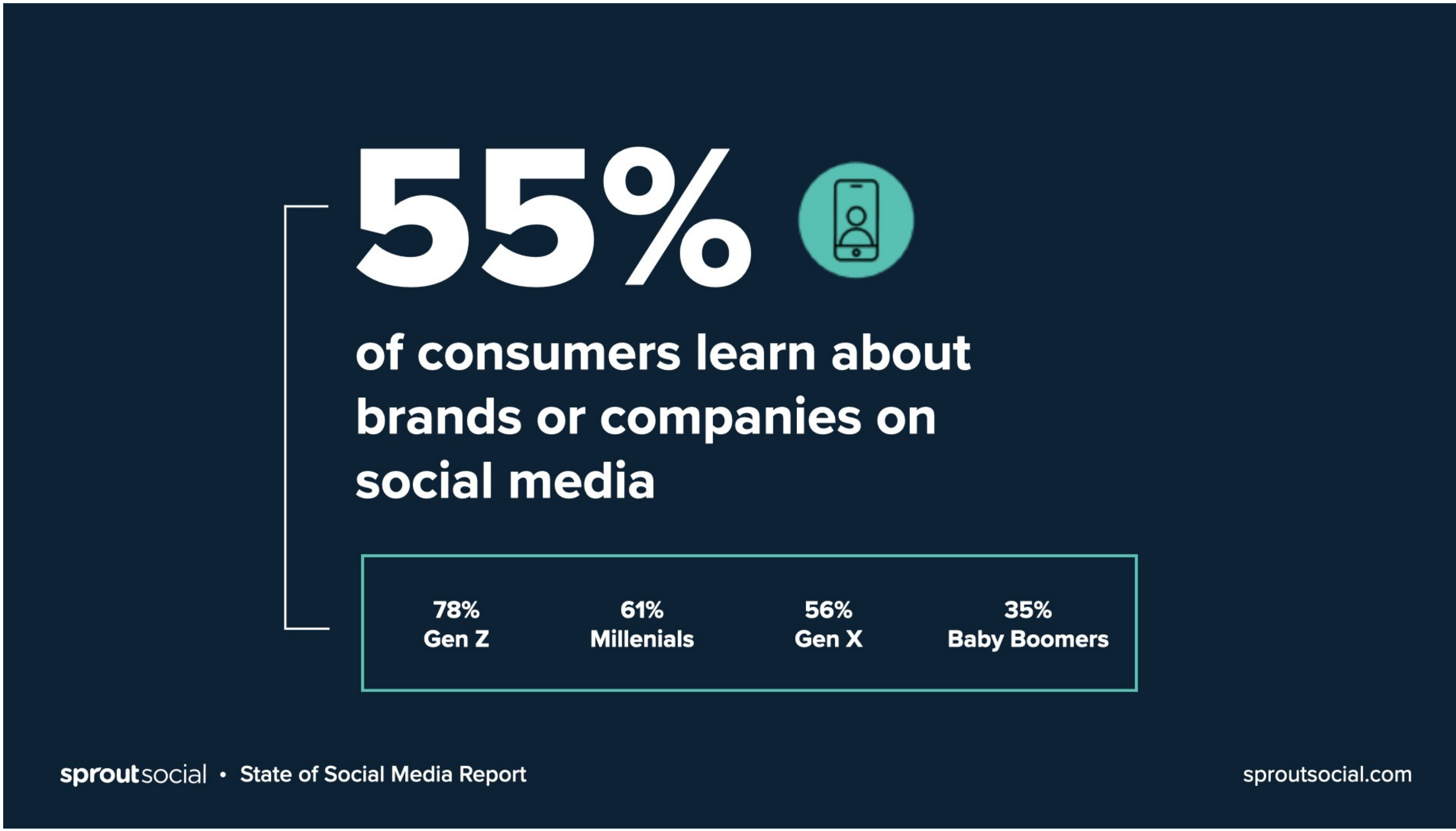
Metaheurísticas

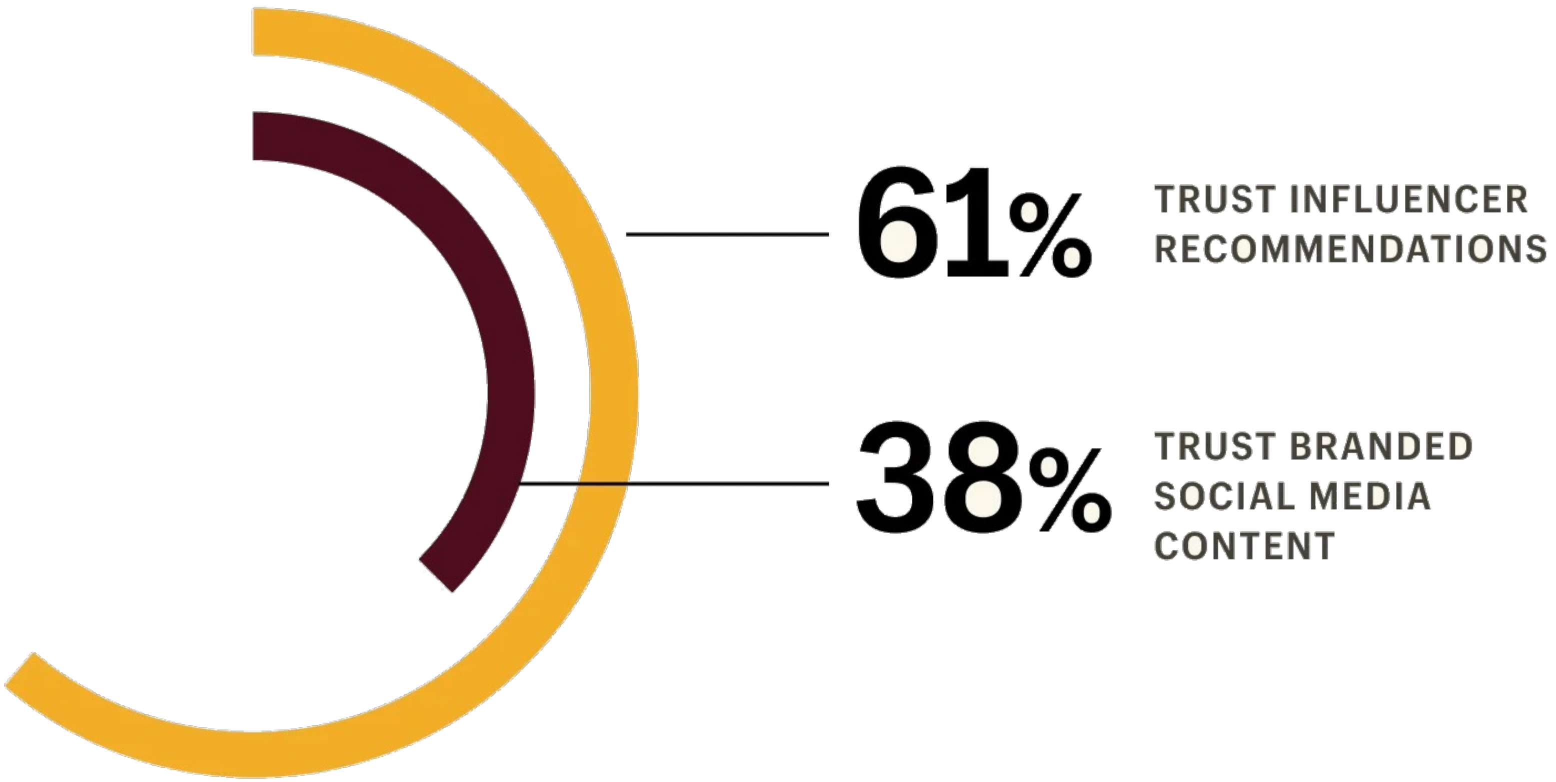
Seminario 2. Problemas de optimización con técnicas basadas en búsqueda local

1. Problema de Maximizar la Influencia en Redes Sociales (SNIMP)
   - Definición del Problema
   - Ejemplo de Aplicación
   - Análisis del Problema
   - Solución Greedy
   - Búsquedas por Trayectorias Simples
   - Casos del problema.
   - Agradecimientos

Definición del Problema



Definición del Problema



Definición del Problema

Influencer Marketing Stats

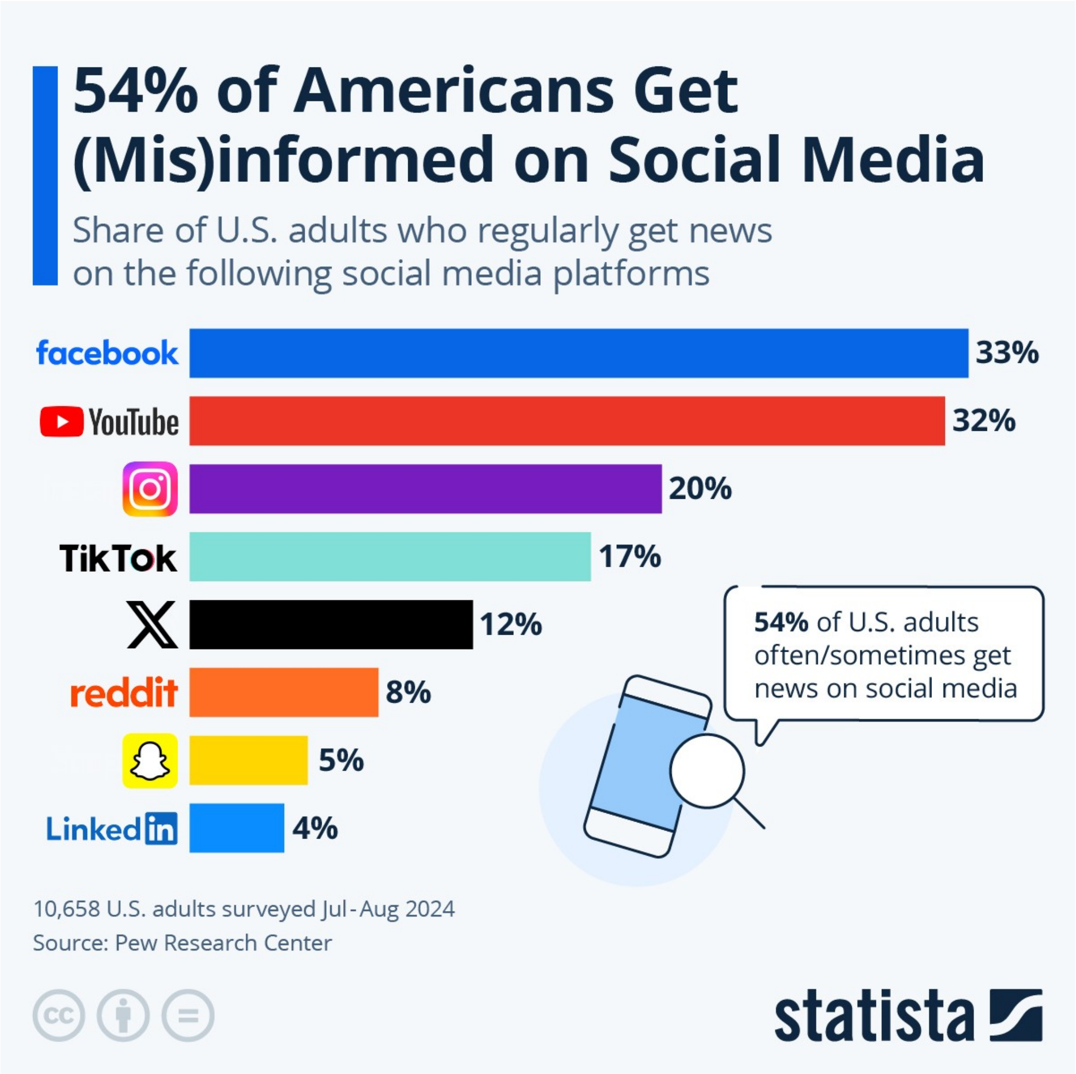
90%
of consumers trust peer recommendations

User generated content is
50%
more trusted by internet users than traditional media

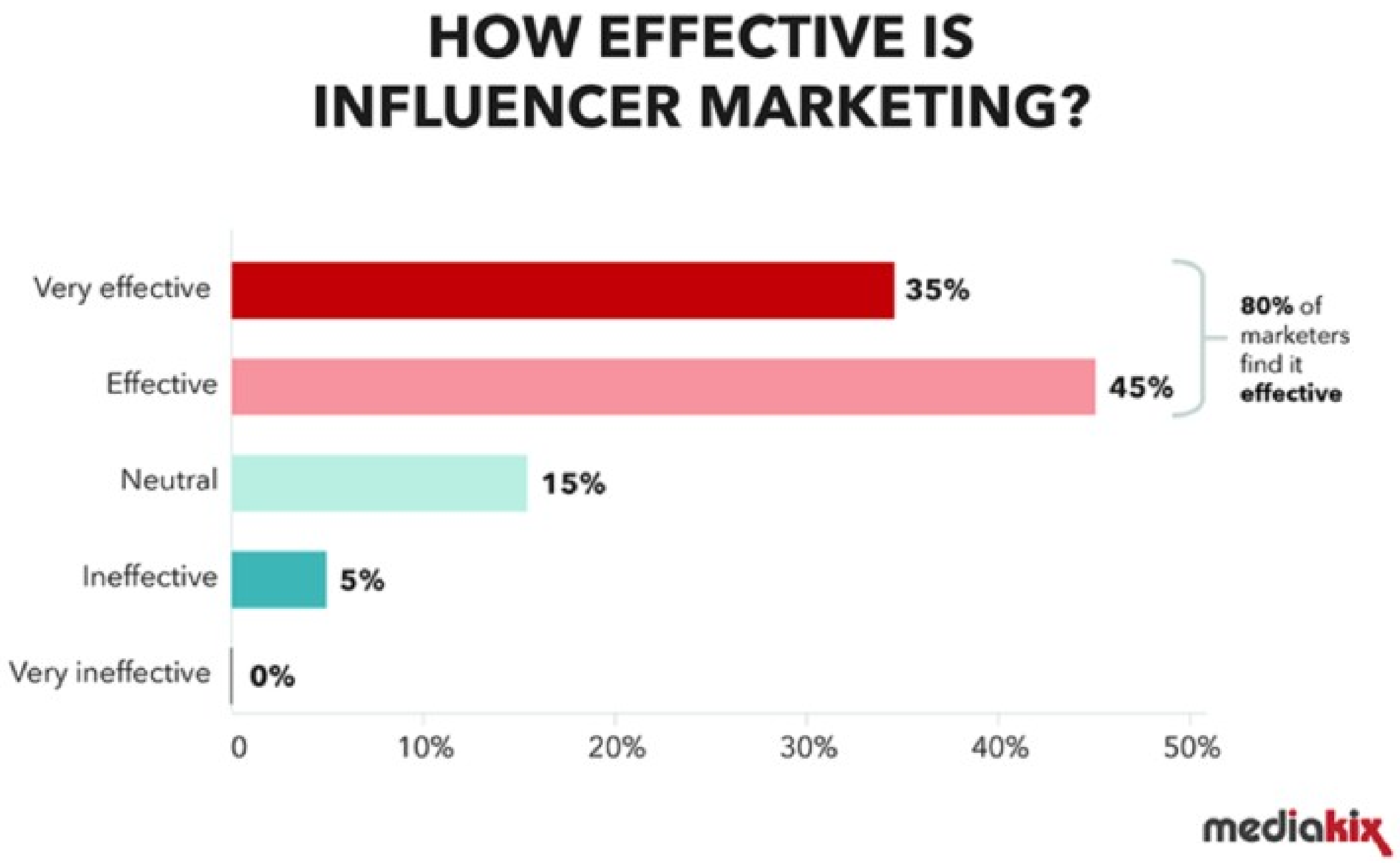
Consumers are
71%
more likely to make a purchase based on social media referrals

81%
of U.S. consumers trust advice and information from blogs

Definición del Problema



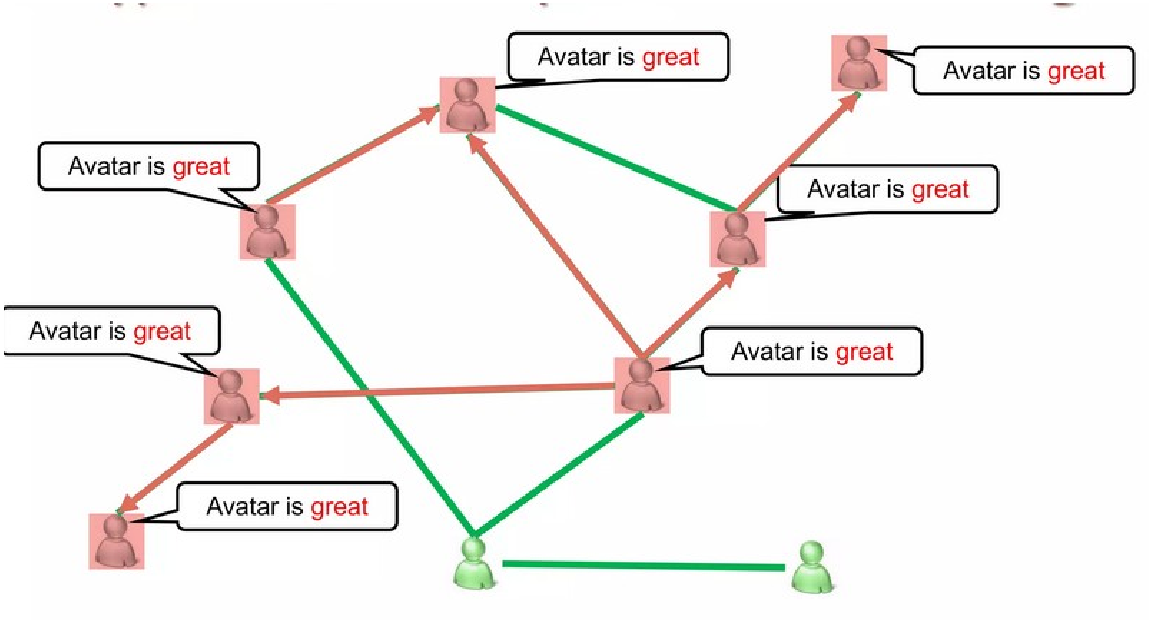
Definición del Problema



Definición del Problema

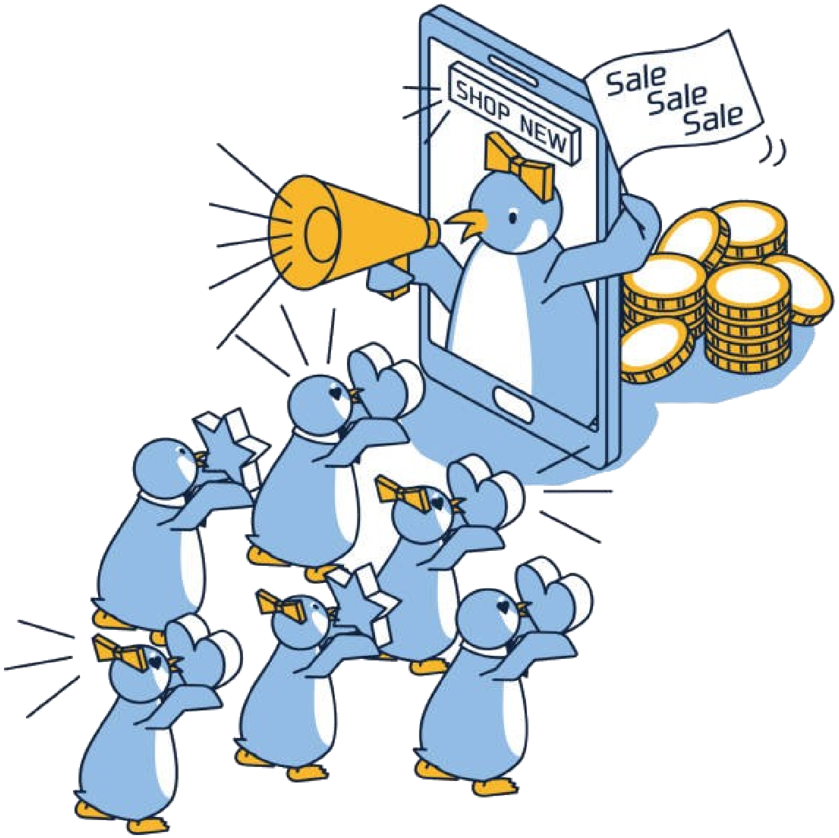
- Social Network Influence Maximization Problem, SNIMP

  Encontrar los K usuarios más influyentes en una red social, simulando un modelo de difusión de influencia.

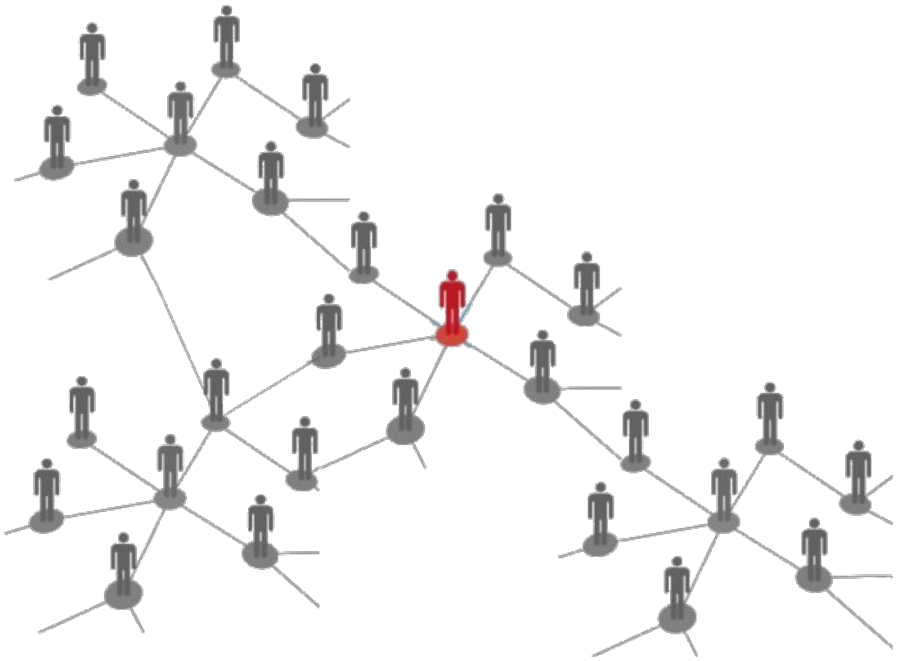


Utilidades del Problema

- Márketing digital.



- Modelos epidémicos



Definición del Problema

- El Problema de Maximizar la Influencia en Redes Sociales, *Social Networks Influence Maximization Problem*, SNIMP, es un problema de optimización combinatoria con una formulación sencilla pero una resolución compleja (**es NP-completo**).
- El problema general consiste en seleccionar un subconjunto *Sel* de m elementos ($|M|=m$) de un conjunto inicial S de n elementos (obviamente, $n > m$) de forma que se **maximice** la influencia entre los elementos escogidos.
- Además de los n elementos (e_i, i=1,...,n) y el número de elementos a seleccionar m, se dispone de una matriz $C=(C_{ij})$ de dimensión $n \times n$ no simétrica que contiene las conexiones entre los nodos de la red.
- C puede ser bastante dispersa, con un número muy variable de conexiones para cada nodo.

Definición del Problema SNIMP

- Para el problema **con el que trabajaremos en prácticas**, se busca lo siguiente:

$$s^{*} \leftarrow_{s \in S} max\, ICM\left(G, S, p, ev\right)$$

  - G es el grafo del problema.
  - S es el conjunto de posibles combinaciones.
  - p es la probabilidad de influencia/contagio (p=0.01)
  - ev es el número de simulación del ICM (ev=10)

Definición del Problema SNIMP

Algorithm 1 $ICM(G = (V, E), S, p, ev)$

$I \leftarrow \emptyset$
for $i \in 1 \ldots ev$ **do**
    $A^\star \leftarrow S$
    $A \leftarrow S$
    while $A \neq \emptyset$ **do**
        $B \leftarrow \emptyset$
        for $v \in A$ **do**
            for $(u, v) \in E$ **do**
                if $rnd(0, 1) \leq p$ **then**
                    $B \leftarrow B \cup \{u\}$
                end if
            end for
        end for
        $A^\star \leftarrow A^\star \cup B$
        $A \leftarrow B$
    end while
    $I \leftarrow I + |A^\star|$
end for
return I/ev

A* es el conjunto de nodos activados, I su tamaño

B es el conjunto de nuevos nodos

Por cada vecino de cada nodo en A se comprueba si se ha infectado

Se añaden los nuevos.
En la siguiente iteración se van a explorar los vecinos de los nuevos nodos *influidos*

Se devuelve el promedio de nodos infectados en las 10 iteraciones del bucle

Solución Greedy

I. Lozano-Osorio, J. Sánchez-Oro, A. Duarte, y O. Cordón, “A quick GRASP-based method for influence maximization in social networks”, J Ambient Intell Human Comput, vol. 14, n. 4, pp. 3767-3779, abr. 2023, doi: 10.1007/s12652-021-03510-4.

- La mayoría de propuestas abordadas son Greedy.
- Hay varias definiciones.
- Vamos a usar una de las más sencillas que ofrece un buen resultado según experimentos:

Añadir secuencialmente el elemento no seleccionado que presente un mayor número de vecinos y vecinos de éstos

Solución Greedy

ALGORITMO GREEDY:

$S \leftarrow \emptyset$
$\mathrm{CL} \leftarrow V$
$\boxed{v_0 \leftarrow \texttt{SelectRandom}(\mathrm{CL})}$ **Solución Inicial**
$S \leftarrow S \cup \{v_0\}$
$\mathrm{CL} \leftarrow \mathrm{CL} \setminus \{v_0\}$
while $|S| < m$ **do**
  $\mathrm{RCL} \leftarrow \mathrm{CL}$
  $\boxed{u \leftarrow \arg\min_{v \in \mathrm{RCL}} g(v)}$ **Aplicar heurística**
  $S \leftarrow S \cup \{u\}$
  $\mathrm{CL} \leftarrow \mathrm{CL} \setminus \{u\}$
end while
return S

Solución Greedy

1) Se calcula para cada nodo *u* el número de nodos vecinos:

$$d+(u)=|N_u^+|, N_u^+=w\in V:(u,v)\in E$$

2) Se define como valor heurístico de un nodo el total de vecinos, y vecinos de éstos:

$$g(u)=d_u^+ + s\sum_{v\in N_u^+} d_u^+$$

3) Se listan los valores posibles ordenados por su valor heurístico.

4) Se escogen los *k* primeros.

Búsquedas por Trayectorias Simples: Búsqueda Local del Mejor

- **Representación**: Problema de selección: un conjunto $Sel=\{s_1, \ldots, s_m\}$ que almacena los *m* elementos seleccionados de entre los *n* elementos del conjunto *S*

  Para ser una solución candidata válida, tiene que satisfacer las restricciones (ser un conjunto de tamaño *m*):

  - No puede tener elementos repetidos
  - Ha de contener exactamente *m* elementos
  - El orden de los elementos no es relevante

Búsquedas por Trayectorias Simples: Búsqueda Local del Mejor

- **Operador de vecino de intercambio y su entorno**: El entorno de una solución *Sel* está formado por las soluciones accesibles desde ella a través de un movimiento de intercambio

  Dada una solución (conjunto de elementos seleccionados) se escoge un elemento y se intercambia por otro que no estuviera seleccionado (*Int*(*Sel*,*i*,*j*)):

$$Sel = \{s_1, \ldots, i, \ldots, s_m\} \quad \Rightarrow \quad Sel' = \{s_1, \ldots, j, \ldots, s_m\}$$

- *Int*(*Sel*,*i*,*j*) verifica las restricciones: si la solución original *Sel* es factible y el elemento *j* se escoge de los no seleccionados en *Sel*, es decir, del conjunto *S*-*Sel*, siempre genera una solución vecina *Sel'* factible

Búsquedas por Trayectorias Simples: Búsqueda Local del Mejor

- Su aplicación provoca que el tamaño del entorno sea demasiado grande (m!), m=10 => más de 3 millones combinaciones.
- La BL del Mejor del MDP explora todo el vecindario, las soluciones resultantes de los $m \cdot (n-m)$ intercambios posibles, escoge el mejor vecino y se mueve a él siempre que se produzca mejora
- Si no la hay, detiene la ejecución y devuelve la solución actual
- El método funciona bien pero es muy lento incluso para casos no demasiado grandes.
- Vamos a probar también parar cuando tras Nvecinos no mejore (Nvecinos=50).

Casos del Problema

- Se utilizarán 4 casos reales seleccionados de varios de los conjuntos de instancias del Stanford *Large Network Dataset Collection:*

  https://snap.stanford.edu/data/index.html.

- Se podría usar esos ficheros, pero hemos modificado y puesto en PRADO una versión sin los nodos sin conexiones, para agilizar y reducir el uso de memoria.
- No planteamos el óptimo, porque en la mayoría de los casos no se conocen, compararemos directamente los valores obtenidos de la función objetivo.
- Para el MDDP, disponemos de cuatro conjunto de datos:
  - **Ca-GrCQc**: Relación de autores en revista científica, tiene 5 242 nodos y 14 496 enlaces.
  - **P2p-Gnutella05**: Intercambio de mensajes en una red social el 5 Agosto 2002, tiene 8846 nodos, y 31 839 enlaces.
  - **P2p-Gnutella08**: Intercambio de mensajes en una red social el 8 Agosto 2002, tiene 6301 nodos, y 20 777 enlaces.
  - **P2p-Gnutella25**: Intercambio de mensajes en una red social, 25 Agosto 2002, tiene 22 687 nodos, y 54 705 enlaces.

La Biblioteca MDPLIB

- El formato de los ficheros de datos es un fichero de texto con la siguiente cabecera:

  # Directed graph (each unordered pair of nodes is saved once): NombreFichero.txt

  # Descripción

  # Nodes: N Edges: E

  # FromNodeId ToNodeId

  u1 v1

  u2 v2

- *Donde N* es el número de elementos, *E* es el número de conexiones.
- A continuación aparecen los pares de valores (u, v) pertenecientes a E. La matriz de conexiones es una matriz dispersa, por lo que se solo se guardan las conexiones con valor de 1.

La Biblioteca MDPLIB

EJEMPLO: FICHERO DEL CASO *ca-GrQc.txt*:

```
#....
# FromNodeId    ToNodeId
0       1
0       2
0       3
0       4
0       5
0       6
0       7
0       8
4       9
4       0
4       10
...
4410    370
4974    4973
4974    4976
1189    1061
1189    1176
1189    1177
```

Agradecimientos

- Para la preparación de las transparencias de presentación del problema MDDP se han usado material de los profesores.
  - Isaac Lozano-Osorio. Universidad Rey Juan Carlos
  - Jesús Sánchez-Oro. Universidad Rey Juan Carlos
  - Abraham Duarte. Universidad Rey Juan Carlos
  - Óscar Cordón. Universidad de Granada.

- Otra referencia que me ha servido muy útil:

Y. Ye, Y. Chen, y W. Han, «Influence maximization in social networks: Theories, methods and challenges», Array, vol. 16, p. 100264, dic. 2022, doi: 10.1016/j.array.2022.100264.

- Además, agradecimientos a Óscar por introducirme al problema.