CS303 Project1 Report

Introduction

Information Exposure Maximization is modeled as an algorithmic problem in [1] to reduce the echo chamber and filter-bubble effect: users get less exposure to conflicting viewpoints and are isolated in their own informational bubble. This problem studies a social network represented as a graph **G=(V,E)**, where **V** is the set of nodes in **G** (i.e., users) and **E** is the set of edges in **G** (i.e., social links between users). The goal of this problem is to select two user sets (referred to as "activities") in a social network to maximize the expected number of nodes. These nodes either connect to two activities simultaneously or are unaware of both activities. This process involves not only the dissemination of information, but also how to balance the exposure of different viewpoints or movements to promote broader information exchange and discussion.

By combining heuristic algorithms and evolutionary algorithms, this project will explore how to effectively select seed nodes to achieve a balance between effective information dissemination and exposure in dynamic social networks. This is not only of great significance for understanding the information flow in social media, but also provides new perspectives and ideas for algorithm design and social network applications. In the following, several preliminary definitions are provided first, and then a formal definition and an example of the IEM problem are presented.

Preliminary

Social Network: $\boldsymbol{G}$ = ($\boldsymbol{V}$, $\boldsymbol{E}$) , where V = {$v1$, ⋯ , vn} represents the node set, and E = V × V represents the edges between nodes.

Campaigns: $\boldsymbol{C}$ = {$\boldsymbol{c1}$, $\boldsymbol{c2}$} represents two campaigns; each campaign holds a viewpoint.

Initial Seed Set: $\boldsymbol{Ii} \subseteq \boldsymbol{V}$,$\boldsymbol{i} \in \{\boldsymbol{1,\ 2}\}$ represents the initial seed set for campaigns ci.

Balanced Seed Set: $\boldsymbol{Si} \subseteq \boldsymbol{V}$,$\boldsymbol{i} \in \{\boldsymbol{1,\ 2}\}$ represents the target seed set that you need to find for each campaign ci .

Budget: $\boldsymbol{k}$ represents the size of the target seed set; $|S1| + |S2| \leq k$.

Diffusion Probability: $\boldsymbol{Pi}$ = {$\boldsymbol{p}(\boldsymbol{i}\ \boldsymbol{u},\boldsymbol{v})\ |(\boldsymbol{u},\boldsymbol{v}) \in \boldsymbol{E}\}$,$\boldsymbol{i} \in \{\boldsymbol{1,\ 2}\}$ represents the edge weight associated with campaign ci , where $p(i\ u,v)$ represents the probability of node u activating node v under each campaign ci .

Diffusion Model: $\boldsymbol{M}$ captures the stochastic process for seed set $Ui = Ii \cup Si$ spreading information on G . ee assume that information on the two campaigns propagates in the network following the independent cascade (IC) model. The two campaigns' messages propagate independently of each other (such propagation is often called *heterogeneous propagation*). The diffusion

process of the first campaign (the process for the second campaign is analogous) unfolds in the following discrete steps:

- In step t = 0, the nodes in seed set $U1$ are activated, while the other nodes stay inactive;
- Each active user u for campaign $c1$ in step t will activate each of its outgoing neighbor v that is inactive for campaign $c1$ in step t −1 with probability $p1(u,v)$;

  The activation process can be considered as flipping a coin with head probability $p1(u,v)$: if the result is head, then v is activated; otherwise, v stays inactive;

Note that u has only one chance to activate its outgoing neighbors for campaign $c1$. After that, u stays active and stops the activation for campaign $c1$;

- The diffusion instance terminates when no more nodes can be activated.

Exposed Nodes: Given a seed set U, $ri(U)$ is the vertices that are reached from U using the aforementioned cascade process for campaign ci . Note that in one propagation, in addition to nodes that were successfully activated by U , nodes that were once attempted to be activated but were not successfully activated by U are also considered to be reached by U. Since the diffusion process is random, $ri(U)$ is a random variable.

Methodology

The following will use heuristic algorithms and evolutionary algorithms to analyze the IEM problem separately.

Heuristic Algorithms

1.Monte Carlo simulation

A computational algorithm that uses repeated random sampling to obtain the likelihood of a range of results of occurring

$$max\Phi(S_1,S_2)=max\mathrm{E}[|V\backslash(r_1(I_1\cup S_1)\Delta r_2(I_2\cup S_2))|]$$
$$\Downarrow$$
$$\hat{\Phi}(S_1,S_2)=\frac{\sum_{i=0}^{N}\Phi_{g_i}(S_1,S_2)}{N}$$

Relevant terms and symbols are explained as follows:

S_i	$Balanced\ seed\ seti$
I_i	$Initial\ seed\ set$
V	$Complete\ seed\ set$
N	$Total\ seed\ quantity$
r_i	$Random\ variables$
$\Phi(S_x,S_y)$	$Estimation\ function$

2.Heuristic algorithm for IEM

Main idea: expand the node with the largest $h(v)$ value

```
Algorithm: Greedy best-first search
S1 ← S2 ← ∅;
while S1 + S2 ≤ k do
    v1* ← arg max v Φ(S1 ∪ v, S2) - Φ(S1, S2);
    v2* ← arg max v Φ(S1, S2 ∪ v) - Φ(S1, S2);
    add the better option between <v1*, ∅> and <∅, v2*> to <S1, S2> while
respecting the budget.
```

3.Combining Monte Carlo and greed search

```
S1 ← S2 ← ∅;
```

while $S1$ + $S2$ ≤ k do
    for j = 1 to N:
        do the following Monte Carlo sampling, each sampling to calculate the h(v) value for all vertices:
        1. simulate an IC model using seed set $I1 \cup S1$, record the activate set $a1$ and exposure set $r1$
        2. simulate an IC model using seed set $I2 \cup S2$, record the activate set $a2$ and exposure set $r2$
        3. for each vi in G:
            3.1 simulate an IC model base on the $a1$ and $r1$, record the $a1\_vi\_increment$ and $r1\_vi\_increment$
            3.2 simulate an IC model base on the $a2$ and $r2$, record the $a2\_vi\_increment$ and $r2\_vi\_increment$
            3.3 calculate and record the $h1j(vi)$ = Φ $S1$ ∪ vi, $S2$ − Φ $S1$, $S2$
            3.4 calculate and record the $h2j(vi)$ = Φ $S1$, $S2$ ∪ vi − Φ $S1$, $S2$
calculate the average $h1avg(v)$ value and $h2avg(v)$ for all vertices
$v1*$ ← arg max$vh1avg(v)$;
$v2*$ ← arg max$vh2avg(v)$;
add the better option between <$v1*$, ∅> and <∅, $v2*$> to <$S1$, $S2$> while respecting the budget.

Evolutionary Algorithms

1.genetic makeup

Binary representation:

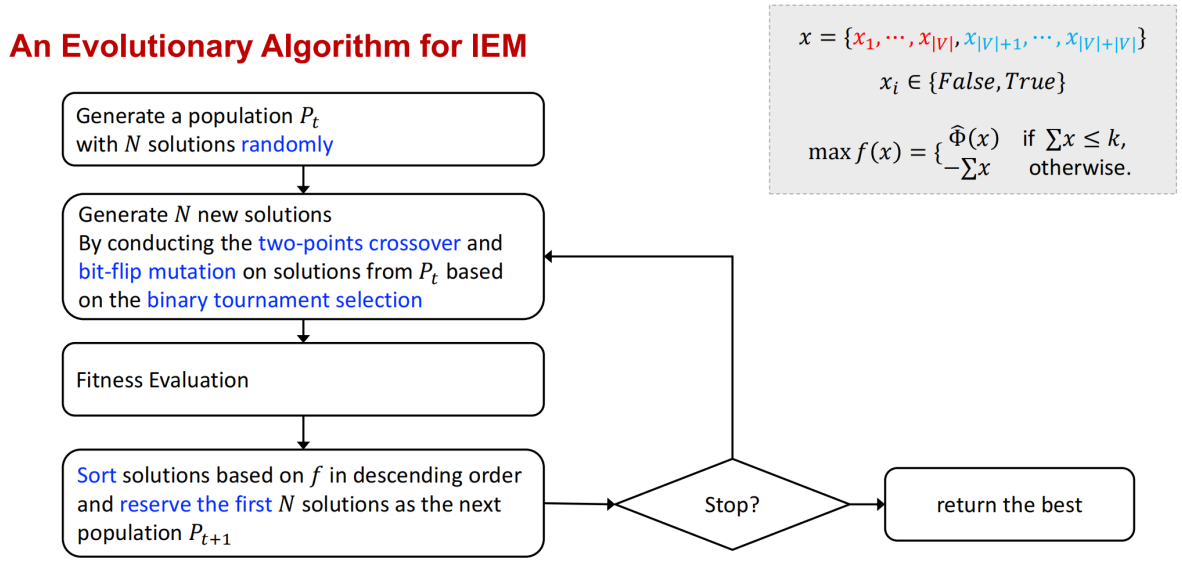
$$x = \{x_1, x_2, \dots x_{|V|}, x_{|V|+1}, x_{|V|+2}, \dots, x_{|V|+|V|}\}$$
$$x_i \in \{False, True\}$$
$$\downarrow$$
$$ith\ node\ is\ added\ into\ S1, i \in [1, |V|]$$
$$ith\ node\ is\ added\ into\ S2, i \in [V+1, V+|V|]$$

2.Fitness Function

Distinguish between feasible and infeasible solutions:

$$fitness(S_1, S_2) = \begin{cases} \hat{\Phi}(S_1, S_2), & if\ |S_1| + |S_2| \le k, \\ -(|S_1| + |S_2|), & \text{otherwise} \end{cases}$$

3.flow charts



Experiments

Setup

1.Environment

Programming Language

Python Version: 3.10

Lib Version

```
pymoo == 0.6.0.1
pandas == 2.0.3
numpy == 1.24.4
scipy == 1.14.1
networkx == 2.8.8
```

2.Dataset

Following provides an overall description of the test datasets:

1. **Type Column:** graph type, all datasets are directed graphs.
2. **Nodes Column:** number of nodes in the graph.
3. **Edges Column:** number of edges in the graph.

All data provided from the Department of Computer Science and Technology at SUSTech.

Heuristic Algorithms data set:

Case No.	Nodes	Edges	Baseline TL	Higher TL
case 0	475	13289	90s	30s
case 1	36742	49248	840s	540s
case 2	36742	49248	840s	540s

Case No.	Nodes	Edges	Baseline TL	Higher TL
case 3	7115	103689	660s	450s
case 4	3454	32140	540s	420s

Good Result:

Problem: Project 1 Phase 2

Submitter: 12211615

Submitted at Fri Oct 11 2024 16:20:30 GMT+0800 (中国标准时间)

Score: 6.5

```
Task Name is Heuristic
In graph map1, Usability Test result is 455.067. Result is right, get 0.1 score.
In graph map1, Accuracy Test result is 455.067. Result is right, get 0.9 score.
In graph map1, Efficiency Test result is 455.067. Result is right, get 0.3 score in 90s.
In graph map2, Usability Test result is 35921.596. Result is right, get 0.1 score.
In graph map2, Accuracy Test result is 35921.596. Result is right, get 0.9 score.
In graph map2, Efficiency Test result is 35917.299. Result is right, get 0.3 score in 540s.
in In graph map3, Usability Test result is 36158.168. Result is right, get 0.1 score.
In graph map3, Accuracy Test result is 36158.168. Result is right, get 0.9 score.
In graph map3, Efficiency Test result is 36158.668. Result is right, get 0.3 score in 540s.
in In graph map4, Usability Test result is 6962.174. Result is right, get 0.1 score.
In graph map4, Accuracy Test result is 6962.174. Result is right, get 0.9 score.
In graph map4, Efficiency Test result is 6962.174. Result is right, get 0.3 score in 660s.
In graph map5, Usability Test result is 3410.978. Result is right, get 0.1 score.
In graph map5, Accuracy Test result is 3410.978. Result is right, get 0.9 score.
In graph map5, Efficiency Test result is 3410.978. Result is right, get 0.3 score in 540s.
```

Time used: 470.7565360069275

Evolutionary Algorithms data set:

Case No.	Nodes	Edges	Baseline TL	Higher TL
case 0	475	13289	420s	380s
case 1	13984	17319	860s	780s
case 2	13984	17319	860s	780s
case 3	3454	32140	1350s	1250s
case 4	3454	32140	1350s	1250s

Good Result:

Problem: Project 1 Phase 3

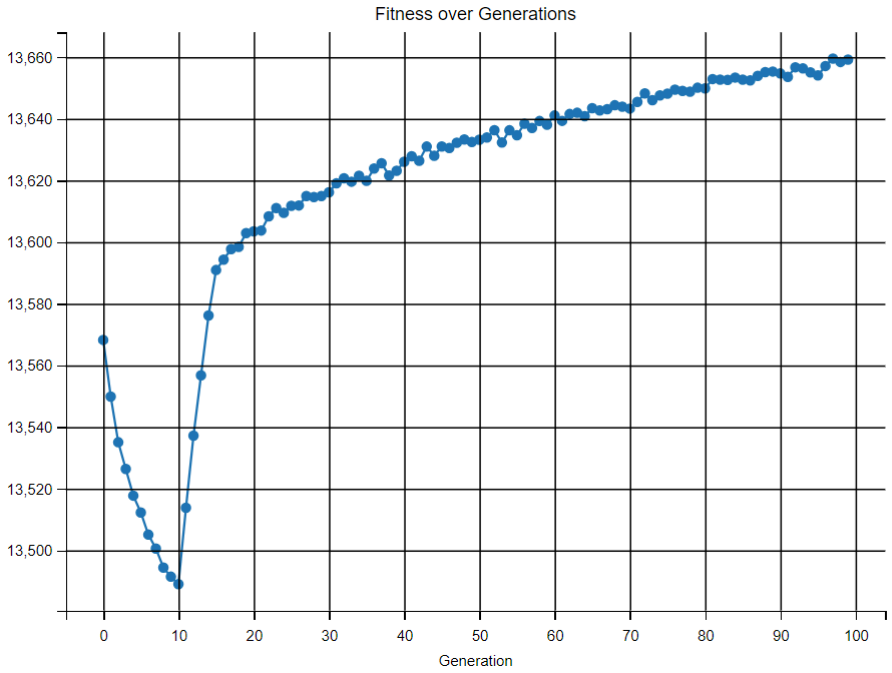
Submitter: 12211615

Submitted at Sun Oct 20 2024 00:00:14 GMT+0800 (中国标准时间)

Score: 6.5

```
Task Name is Evolutionary
In graph map1, Usability Test result is 436.547. Result is right, get 0.1 score.
In graph map1, Accuracy Test result is 436.547. Result is right, get 0.9 score.
In graph map1, Efficiency Test result is 436.007. Result is right, get 0.3 score in 380s.
In graph map2, Usability Test result is 13680.382. Result is right, get 0.1 score.
In graph map2, Accuracy Test result is 13680.382. Result is right, get 0.9 score.
In graph map2, Efficiency Test result is 13680.382. Result is right, get 0.3 score in 860s.
In graph map3, Usability Test result is 13459.363. Result is right, get 0.1 score.
In graph map3, Accuracy Test result is 13459.363. Result is right, get 0.9 score.
In graph map3, Efficiency Test result is 13455.183. Result is right, get 0.3 score in 780s.
In graph map4, Usability Test result is 3389.081. Result is right, get 0.1 score.
In graph map4, Accuracy Test result is 3389.081. Result is right, get 0.9 score.
In graph map4, Efficiency Test result is 3389.081. Result is right, get 0.3 score in 1350s.
In graph map5, Usability Test result is 3318.089. Result is right, get 0.1 score.
In graph map5, Accuracy Test result is 3318.089. Result is right, get 0.9 score.
In graph map5, Efficiency Test result is 3318.089. Result is right, get 0.3 score in 1350s.
```

Time used: 421.9963638782501



This chart records the best fitness of each generation, and there is still significant room for improvement in the end. It can improve algebra and further enhance fitness.

3.Analysis

Regarding the setting of genetic algorithm control parameters. The current setting is 100 generations, which meets the requirements of the question. Improving algebra significantly increases time expenditure and can optimize the final result, but it does not correspond to an increase in comprehensive income. At present, the initial race size is 500, and increasing it does not significantly change the results. Reducing it will have a certain negative impact on the results, and there is no significant increase in time benefits. At present, each generation produces 50 new offspring, and increasing the number does not significantly change the results. Reducing the number will have a certain negative impact on the results, and there is no significant increase in time benefits. The probability of genome mutation for each individual is 0.1. Each time a gene is

mutated, both reduction and increase are likely to have a significant negative impact on the results, without any increase in time benefits. Using the tournament method to randomly select parents, currently 3 sets of genomes are selected for competition each time. If the number of individuals in the competition is increased, it will lead to too fast convergence, which will have a significant negative impact on the results. If the number of individuals in the competition is reduced, it will lead to too fast or too slow, which will have a significant negative impact on the results.

Bad return:

Problem: Project 1 Phase 3

Submitter: 12211615

Submitted at Sat Oct 19 2024 20:33:07 GMT+0800 (中国标准时间)

Score: 1.3

```
Task Name is Evolutionary
In graph map1, Usability Test result is 432.524. Result is right, get 0.1 score.
In graph map1, Accuracy Test result is 432.524. Result is right, get 0.9 score.
In graph map1, Efficiency Test result is 431.805. Result is right, get 0.3 score in 380s.
in In graph map2, Usability Test RunError, Error info is Cannot generate seed in given
time limitation
In graph map2, Accuracy Test RunError, Error info is Cannot generate seed in given time
limitation
In graph map2, Efficiency Test RunError, Error info is Cannot generate seed in given time
limitation
In graph map3, Usability Test RunError, Error info is Cannot generate seed in given time
limitation
In graph map3, Accuracy Test RunError, Error info is Cannot generate seed in given time
limitation
In graph map3, Efficiency Test RunError, Error info is Cannot generate seed in given time
limitation
In graph map4, Usability Test RunError, Error info is Cannot generate seed in given time
limitation
In graph map4, Accuracy Test RunError, Error info is Cannot generate seed in given time
limitation
In graph map4, Efficiency Test RunError, Error info is Cannot generate seed in given time
limitation
In graph map5, Usability Test RunError, Error info is Cannot generate seed in given time
limitation
In graph map5, Accuracy Test RunError, Error info is Cannot generate seed in given time
limitation
In graph map5, Efficiency Test RunError, Error info is Cannot generate seed in given time
limitation
```

Time used: 461.9250776767731

In the use of evolutionary algorithms, due to excessively high algebraic settings and high memory usage, the link memory cannot be fully utilized, resulting in low performance.

Problem: Project 1 Phase 2

Submitter: 12211615

Submitted at Fri Oct 11 2024 16:07:37 GMT+0800 (中国标准时间)

Score: 5.3

```
Task Name is Heuristic
In graph map1, Usability Test result is 439.18. Result is right, get 0.1 score.
In graph map1, Accuracy Test result is 439.18. Result is right, get 0.9 score.
In graph map1, Efficiency Test result is 437.763. Result is right, get 0.3 score in 30s.
in In graph map2, Usability Test result is 35933.847. Result is right, get 0.1 score.
In graph map2, Accuracy Test result is 35933.847. Result is right, get 0.9 score.
In graph map2, Efficiency Test result is 35930.906. Result is right, get 0.3 score in
540s.
in In graph map3, Usability Test result is 36082.325. Result is right, get 0.1 score.
In graph map3, Accuracy Test result is 36082.325. Result is right, get 0.9 score.
In graph map3, Efficiency Test result is 36083.767. Result is right, get 0.3 score in
540s.
in In graph map4, Usability Test result is 6881.33. Result is right, get 0.1 score.
In graph map4, Accuracy Test result is 6881.33. Result is Not Good enough, get 0 score.
In graph map4, Efficiency Test result is 6881.918 in 450s. Test result is 6881.33 in 660s.
Result is Not Good enough, get 0 score.
In graph map5, Usability Test result is 3406.623. Result is right, get 0.1 score.
In graph map5, Accuracy Test result is 3406.623. Result is right, get 0.9 score.
In graph map5, Efficiency Test result is 3406.623. Result is right, get 0.3 score in 540s.
```

Time used: 459.5342104434967

Due to low Monte Carlo parameter settings. However, being too high can lead to timeouts, and we are seeking a balance between the two.

Conclusion

Through this project, we propose an algorithm that combines Monte Carlo simulation with greedy best priority search, as well as an algorithm that combines Monte Carlo simulation with natural evolution simulation to solve the problem of maximizing information exposure. These algorithms aim to select seed nodes in dynamic social networks, balancing the relationship between effective information dissemination and exposure. The experimental results show that our algorithm has significant effects in improving information exposure and reducing echo chamber effects.

In the methodology section, we provided a detailed description of each step of the algorithm and demonstrated the specific implementation process through pseudocode and flowcharts. We analyzed the complexity and performance of the algorithm, and explored the impact of different components and hyperparameters on the experimental results. In particular, we discussed the strategy for selecting seed nodes and the contribution of different heuristic methods to the final results, which provides in-depth insights into understanding the effectiveness of the algorithm.

In the experimental section, we provided a detailed introduction to the experimental setup, including features of the dataset, software and hardware configurations, etc. Through experiments on social network graphs of different scales, we found that the algorithm performs well in processing these graphs and the running time is also within an acceptable range. In addition, we analyzed the relationship between experimental results and theoretical expectations, providing explanations for possible differences. For example, in some cases, the complexity of the information propagation path may result in actual effects not meeting expectations, which suggests the need for further optimization of algorithm design.

In summary, our research not only provides a new perspective for understanding information flow in social media, but also offers innovative ideas for algorithm design and social network applications. However, we also recognize that algorithms have some limitations when dealing with large-scale networks, such as efficiency issues, the conflicting balance between information dissemination and exposure, and how to set relevant parameters to ensure higher exposure rates. Future work will continue to explore these issues and attempt to introduce more efficient algorithms and strategies to improve the effectiveness of solving the problem of maximizing information exposure.

In addition, considering the dynamic nature of social networks, we also plan to study the adaptability of algorithms in real-time environments, including how to handle dynamic changes in nodes and edges, and how to maintain the effectiveness of information dissemination in changing network structures. This will provide stronger support for the practical application of algorithms, enabling them to play a role in rapidly changing social networks.

Reference

[1] K Garimella, A Gionis, N Parotsidis, N Tatti. Balancing information exposure in social networks. NeurIPS 2017: 4663-4671
[2] S Cheng, H Shen, J Huang, W Chen, X Cheng. IMRank: influence maximization via finding self-consistent ranking. SIGIR 2014: 475-484
[3] M Gong, J Yan, B Shen, L Ma, Q Cai. Influence maximization in social networks based on discrete particle swarm optimization. Inf. Sci. 367-368: 600-614 (2016)
[4] Q Jiang, G Song, G Cong, Y Wang, W Si, K Xie. Simulated Annealing Based Influence Maximization in Social Networks. AAAI 2011