MINISTRY OF EDUCATION
AND TRAINING

VIETNAM ACADEMY OF
SIENCE AND TECHNOLOGY

GRADUATE UNIVERSITY SCIENCE AND TECHNOLOGY

VU CHI QUANG

STUDYING SOME METHODS FOR SOLVING THE MAXIMUM INFLUENCE PROBLEM ON SOCIAL NETWORKS WITH PRIORITY AND COST CONSTRAINTS

SUMMARY OF DOCTOR THESIS IN INFORMATION SYSTEMS
Code: 9 48 01 04

Ha Noi – 2024

The dissertation is completed at: Graduate University of Science and Technology, Vietnam Academy of Science and Technology.

Supervisors:

Supervisor 1: Dr. Nguyen Nhu Son, Institute of Information Technology

Supervisor 2: Assoc. Prof. Dr. Ngo Quoc Dung, Institute of Posts and Telecommunications Technology

Referee 1: ....……..

Referee 2: ....……

Referee 3: .....……..

The dissertation will be examined by Examination Board of Graduate University of Science and Technology, Vietnam Academy of Science and Technology at..... (time, date, year...)

This dissertation can be found at:

1) Graduate University of Science and Technology Library
2) National Library of Vietnam

INTRODUCTION

1. The urgency of the thesis

- *In terms of practice*: With a large number of users, social networks (SN) have been bringing many practical benefits to users. It can be said that SN has become a useful tool in people's lives, as well as a huge store of knowledge that everyone can easily access. SN has brought great political and economic benefits to the entire society. Therefore, research is needed to maximize information spread on SN more and more effectively.

- *In terms of science*: Studying the problem of Maximum Influence on SN is a research direction that many scientists are interested in, belonging to the group of information spreading problems (Spread Information - SI). Besides, SN has a huge amount of data, dispersion and random information spreading process, network structure is complex, heterogeneous and constantly fluctuating, so it is necessary to introduce effective solutions in terms of time and memory.

2. Research objectives of the thesis

- Researching maximum influence problems on information spread models. Thereby proposing new variations with practical applications.

- Propose models to solve the above problems, study their complexity based on information spread models.

- Propose effective algorithms to solve the above problems, with special attention to improving solution quality as well as applicability to large networks ranging from hundreds of thousands to millions or thousands. billion edges or vertices.

3. The main contents of the thesis

Chapter 1: Theoretical basis of the thesis and related research. In this chapter, the thesis introduces SN, its basic components, some characteristics as well as the benefits and downsides of SN; Introducing models and some

common SI problems on SN. General, foundational knowledge for the research in the following chapters of the thesis.

Chapter 2: Maximizing influence with priority constraints on social networks. In this chapter, the thesis poses and defines the IMP problem based on the information spread model; proposed an integrated greedy algorithm (IG) and an integrated greedy sampling algorithm (IGS) for the IMP problem; prove that the algorithm performance is approximately equivalent to the optimal solution; Theoretical analysis and algorithm evaluation based on experiments with SN data sets.

Chapter 3: Maximize the impact of spreading information on many topics with limited costs. The thesis proposes a new model for the multi-topic information spreading problem, defines the BkIM problem, and proposes two streaming algorithms that browse data once to provide theoretical limits of the problem. To examine the performance of the proposed algorithms in practice, the thesis conducted experiments on the Maximum Influence application with k topics under limited cost conditions.

CHAPTER 1
THEORETICAL BASIS OF THE THESIS AND RELATED RESEARCH

1.1 Introduction to social networks

The concept of social network was first mentioned and used by Barnes in 1954. Since then, hundreds of thousands of social networks have been built with billions of users around the world. Each network has its own structure and purpose, but they all have four basic components: Users, links between users, information spread on the network and user interactions with each other. In addition, SN also has four common characteristics: Small world characteristics, population characteristics, community structure characteristics and exponential distribution characteristics.

With a large number of users, SN has been bringing many practical benefits to users. Besides, it also allows the rapid spread of false information,

causing significant damage to human life. To make SN more and more useful to the community, we need to find effective solutions to promote the benefits and limit the downsides of SN.

1.2 Modeling information spread on social networks

Modeling information spreading problems on SN plays an important role in solving SI problems. Helps researchers have the most general and concise overview of SN. From there, come up with effective solutions to solve problems on the model and gradually apply it into practice. The discrete propagation model is widely used in research. Typically, the Linear Threshold (LT) and Independent Cascade (IC) models are considered discrete propagation models used in the thesis.

1.2.1 Linear Threshold Model (LT)

A social network is represented by a graph, each edge has a weight $w(u,v)$ is a positive real number that satisfies the condition $\sum_{u\in N_{in}(v)} w(u,v) \leq 1$. $N_{in}(u), N_{out}(u)$ is the set of input nodes and set of output nodes of the vertex. Each button has an activated or deactivated state and has an activation threshold $\theta_u \in [0,1]$. Let S be the source set (seed set), S_t is the set of buttons activated by at time *t*. When $t = 0$, the nodes in the set S_0 all have an active state; when $t \geq 1$, each button v will be activated if: $\sum_{u\in N_{in(v)}\cap S_{t-1}} w(u,v) \geq \theta_u$. The propagation process ends when after each step no further nodes are activated.

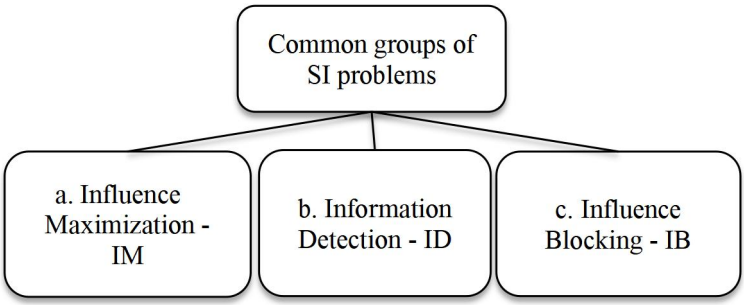
1.2.2 Independent Degree Model (IC)

Different from the LT model, in the IC model each edge is assigned an influence probability $p(u,v) \in [0,1]$. Let S_t be the set of buttons activated by S at a time t. When $t = 0$, nodes in the source set S_0all have an active state. At a time $t \geq 1$, each button $u \in S_0$ has a unique activation chance to the button $v \in N_{out}(u)$ with probability of success $p(u,v)$. The propagation process ends when no further nodes are activated between the two steps.

Let $\sigma(S)$ is the influence function on the LT, IC model, this value is the expected number of nodes activated at the end of propagation. Calculating the function $\sigma(S)$ was proven by D. Kemp as #P-hard, to solve this problem they propose the LE (Live Edge) online edge model and prove it is equivalent to LT model and IC model.

1.3 Some problems of spreading information on social networks

The problem of information dissemination arises from practical needs, network developers, network users and scientists always want to find optimal solutions to exploit the strengths of SN to serve for necessary human needs and limit unwanted negative effects. In terms of research purposes, SI problems can be divided into three main groups, which are: Influence Maximizing, Information Detecting and Influence Blocking.



1.3.1 Influence Maximization - IM

This problem comes from the need to choose a set of users to start the SI so that the number of people affected by that information on the SN is maximized. IM has applications in viral marketing, Misinformation (MI), analyzing influence on SN, etc. The goal of the problem is to choose a set of seeds to start the process. disseminating information so that it affects the most users.

The challenge in solving this problem is that they belong to class NP-Hard and accurately calculate the objective function of problem class #P-Hard.

1.3.2 Information Detection - ID

Assuming that a set of users suspected of spreading information is known, the goal of the problem is to find a set A to monitor so that the ability to detect information from the set of users is greatest. This problem has applications in detecting Misinformation - MI and detecting sources of MI spread, assessing user opinion trends on social networks.

1.3.3 Influence Blocking - IB

In contrast to IM, the problem of influence blocking aims to limit the spread and spread of information of a given news source. The goal of these problems is to limit the spread of bad elements on SN including: bad news, Misinformation, or the spread of viruses, extremist ideologies, etc.

Suggested methods that can limit the impact of a given source include:

- Disable users or link sets: remove vertex or edge sets to be immune to the effect.

- Information decontamination: choose a peak set to start spreading positive influences to counteract the influence of negative information.

1.4 Combinatorial optimization problem and some methods for solving combinatorial optimization problems.

As introduced in the previous section, common SI problem groups such as IM, ID, IB are often formulated as Combination Optimization (CO) problems belonging to the class of NP-hard problems. The two problems proposed in the thesis are also given as CO problems. Therefore, to come up with a method to solve these problems, the thesis researches some basic knowledge about CO. This is the knowledge used in subsequent research of the thesis.

Define: (CO): Each CO problem corresponds to a set of three (S, f, Ω), where S is a finite set of states (potential solutions), f is the objective function determined on S and Ω is the set of constraints. The goal of these exercises is to find the maximum or minimum of the function f on set S: $\max(\min)$: $f(s)$: $s \in S$.

Problems on SN are often large in size, so common solution methods are: Approximation, Monte Carlo, Heuristic.

- *Approximate method:* The approximation method is a method of providing an algorithm that approximates a certain ratio compared to the best solution. Suppose we need to find the optimal solution to the problem of spreading information in the form of CO, belonging to the NP-hard, NP-complete class with the goal of finding the maximum function $f: S \to \mathbb{R}_+$, where S is the solution space of the problem. Let OPT(Optimal) is the optimal solution of the problem. The approximation algorithm is defined as follows:

Define: (Approximation algorithm) We say that the approximation algorithm A gives a solution of $s \subseteq S$ has an approximate ratio of $\rho > 0$ if it performs in polynomial time according to the input size of the problem and satisfies: $f(s)/OPT \geq \rho$. In the case of needing to find the minimum function f (find the smallest value), the optimal ratio is defined as: $f(s)/OPT \leq \rho$.

- *Monte Carlo (MC) method* : This method is also called simulation method or statistical testing method. The main idea of the Monte Carlo (MC) method is to approximate an expectation μ of X by averaging the results of many independent experiments, where the random variables have the same distribution. In many cases, the problem has a complex objective function and unlimited search space, so approximation methods cannot be applied. MC is an effective method at this time.

- *Heuristic method :* This is a method designed based on experience to solve a problem faster when previous methods are too slow or to find an approximate solution when previous methods have failed to find any exact solution.

- *Streaming algorithm:* In computer science, streaming algorithms are a class of algorithms designed to process data in a sequentially received data environment. In this environment, data is processed as a continuous stream, it is not possible to store all data in memory, and it is often impossible to access

processed data again. Streaming algorithms are often applied in big data processing applications, where data is generated continuously and needs to be processed immediately to produce results in real time.

Important properties of streaming algorithms include: *continuous data processing, limited memory, accuracy, data updates.*

1.5 Related studies

- *Related research in the country:*

Author Pham Van Canh has researched the problems: Preventing false information with budget and time constraints (MMR), Preventing false information with a given goal (TMB), Maximizing impact Competing with Time and Budget Constraints (BCIM) and Generalized Misinformation Detection (GMD).

Author Pham Van Dung has researched the problems: Detecting sources of false information on social networks with minimum budget (MBD) and Preventing false information on many topics on social networks with budget constraints. book (MBMT).

- *Research related to the problem of maximum influence:*

Kempe and colleagues [3] were the first to state the IM problem on two models (IC) and (LT). Prove that the IM problem is NP-Hard and the objective function of the IM problem is #P-Hard.

Chen and colleagues [97] did a general study on IM and BIM problems.

Borgs et al [46] proposed an approximation algorithm 1-1/e-ϵ with probability 1-δ, by introducing the Inverse Influence Sampling model RR (Reverse Reachable).

The authors in references [9]-[16] have studied IM problem variations by time, cost, distance and by topics.

- *Research related to the problem of maximizing the influence of information dissemination on many topics.*

The authors in reference [29] first studied the function *k*-Submodular.

The authors in references [25] -[30], [106] - [110] researched on optimizing the *k*-Submodular function with different variations such as: unconstrained, size constrained, constrained. cost binding, matroid binding, backpack binding, …

However, the authors' algorithms can only be applied to the case of monotone function f, In the case of non-monotone function f, the solution is not as expected.

1.6 Chapter conclusion

This chapter of the thesis introduces general knowledge about SN, modeling SI problems on SN, discrete SI model and 03 models LT, IC and LE; These are the models used in the thesis publications. Next, the thesis introduces an overview of the combinatorial optimization problem and methods for solving the CO problem. These studies are an important foundation for the thesis to propose IMP and BkIM problems in the following chapters of the thesis.

CHAPTER 2
MAXIMUM INFLUENCE WITH PRIORITY CONSTRAINTS ON SOCIAL NETWORKS

The influence maximization (IM) problem requires finding a set of k nodes in a social network to start spreading influence so that the number of influential nodes after the information diffusion process is maximized. However, previous studies have ignored the limitation of the priority constraint, leading to inefficient collection of the seed set.

To solve this problem, the thesis proposes a new problem called influence maximization with priority constraint (IMP), with the goal of finding a group of k nodes in the SN to be able to influence the number of nodes. largest nodes while affecting a set of priority users U not less than a threshold T. The PhD student points out that this problem is NP-Hard and existing algorithms for IM cannot be applied to the problem This. To find a solution, the PhD student proposed two effective algorithms, called Integrated Greedy - IG and Integrated Greedy-based

Sampling - IGS with the following guarantees: ensure the approximate ratio of the solution.

2.1 State the IMP problem

Definition: (IMP problem). Given the graph G = (V, E) according to the IC model, a positive integer k (cost), priority set $U \subset V$ and threshold T with $T \leq k$, $T \leq |U|$. The IMP problem requires finding a seed set $S \subset V$, wit $|S| \leq k$ and $\sigma_U(S) \geq T$ such that the level of influence spread $\sigma(S)$ is maximized, that is, finding S is the solution to the following optimization problem:

maximize: $\sigma(S)$; subject to: $|S| \leq k$; $\sigma_U(S) \geq T$.

IMP becomes an IM problem when U is empty. Therefore, IM is a special case of IMP, and IMP is also NP-Hard. Additionally, computing the influence function from the seed set is proven to be #P-Hard.

2.2 Algorithm proposal

The thesis proposes two algorithms: Integrated Greedy Algorithm-IG and Integrated Greedy Sampling Algorithm-IGS.

2.2.1 Integrated Greedy Algorithm IG

The integrated greedy algorithm (IG), based on a modification of the traditional greedy algorithm to solve monotonic and *submodular* problems, ensures an approximate rate of solution.

Algorithm 2.1: The integrated greedy algorithm IG

Input: Graph G = (V, E), U ⊂ V, k, T
Output: Seed set S, t

1. $S_1 \leftarrow \emptyset$, $S_2 \leftarrow \emptyset$
   /* Phase 1: Greedy strategy for prior set */
2. **while** $\sigma_U(S_1) < T$ **do**
3. $u \leftarrow \mathbf{argmax}_{v \in V \setminus S1}(\sigma_U(S_1 \cup \{v\}) - \sigma_U(S_1))$
4. $S_1 \leftarrow S_1 \cup \{u\}$
5. **end**
6. $t \leftarrow k - |S_1|$, $i \leftarrow 0$
   /* Phase 2: Greedy strategy for IM within remain budget*/
7. **while** $i < t$ **do**

8. $u \leftarrow \mathbf{argmax}_{v \in V \setminus S2 \setminus S1}(\sigma(S_2 \cup \{v\}) - \sigma(S_2))$
9. **if** $u \in S_1$ **then**
10. $t \leftarrow t + 1$
11. **end**
12. $S_2 \leftarrow S_2 \cup \{u\}, i \leftarrow i + 1$
13. **end**
14. $S \leftarrow S_1 \cup S_2$
15. **return** S, t;

Algorithm evaluation 2.1. *The IG algorithm returns (S, t), where S is a feasible solution and t ≥ 1, satisfying:*

$$\sigma(S) \geq \left(1 - \left(1 - \frac{1}{k}\right)^t\right)\sigma(S^*)$$

The worst-case approximation ratio is obtained when t=1 and it is equal to 1/k.

2.2.2 Integrated Greedy Sampling algorithm IGS

Although Algorithm 2.1 can provide an approximate guarantee, it cannot work with real social networks because computing the influence function $\sigma(S)$ is #P-Hard. To overcome this challenge, the thesis proposes a random algorithm with approximate guarantee based on combining IG with sampling technique.

The idea of IGS is to create a collection of sets N_u TRR $\mathcal{R}_1$ and set two candidate solutions S_1, S_2 empty. The body of the IGS is divided into two phases.

In phase 1, the algorithm finds a candidate solution S1 with the smallest size such that $\hat{\sigma}(S) \geq (1 + \alpha)T$ using a greedy strategy with a potential function $\hat{\sigma}$ over $\mathcal{R}_1$. The candidate solution S_1 obtained in this stage satisfies the priority constraint $\sigma_U(S_1) \geq T$ with probability at least 1 - δ.

Phase 2, select a candidate solution S_2 with the remaining budget (t= k - $|S_1|$) so that the level of influence spread $\sigma(\cdot)$ is maximized. In this stage, the algorithm sets the parameters ϵ_1, t_{max}, N_{max} and generate N_1 sample sets RR $\mathcal{R}_2$. In each iteration IGS finds a candidate solution S_2 using a greedy strategy. The algorithm selects a node u with approximately maximum incremental influence $\hat{\sigma}(\cdot)$ over $\mathcal{R}_2$ until t nodes are selected. The

algorithm then checks the quality of the candidate solution S_2. Next the algorithm calculates the functions $F_l(S_2, \mathcal{R}_2, \delta)$- lower border of $\sigma(S_2)$, and $F_u(S_2, \mathcal{R}_2, \delta)$- upper bound of an optimal solution to the IMP problem.

Algorithm 2.2: Integrated Greedy -based Sampling (IGS) algorithm

Input: Graph $G = (V, E)$, $U \subset V$, k, T, ϵ , α, $\delta \in (0, 1)$

Output: Seed set S

1. Generate a set of $N_U = (2 + \frac{2}{3}\alpha)|U|\frac{\ln\left(\binom{|U|}{\lfloor |U|/2 \rfloor}/\delta\right)}{(T+T\alpha)\alpha^2}$TRIS sets $\mathcal{R}_1$
2. $S_1 \leftarrow \emptyset$, $S_2 \leftarrow \emptyset$

/* *Phase 1*/

3. **While** $\hat{\sigma}_u(S_1) < T + \alpha T$ **do**
4. $\quad u \leftarrow \arg\max_{v \in V \setminus S_1}\left(\hat{\sigma}_u(S_1 \cup \{v\}) - \hat{\sigma}_u(S_1)\right)$
5. $\quad S_1 \leftarrow S_1 \cup \{u\}$
6. **End**

/* *Phase 2* */

7. $\epsilon_1 \leftarrow \frac{\epsilon}{2(1-1/e)-\epsilon}$
8. $t_0 \leftarrow k - |S_1|$, $\delta_1 \leftarrow \frac{\delta}{6}$, $t_{max} \leftarrow \arg\max_{j \in \{t, t+1, t+2, \ldots, k\}} \ln(\binom{n}{j}/\delta_1)/j$
9. $N_1 \leftarrow \frac{\ln(1/\delta_1)}{\epsilon_1^2}$, $N_{max} \leftarrow \frac{(2+\frac{2}{3}\epsilon_1)\, n \ln\left(\binom{n}{t_{max}}/\delta_1\right)}{t_0 \epsilon_1^2}$
10. $i_{max} = \left\lceil \frac{N_{max}}{N_1} \right\rceil$, $\delta_2 \leftarrow \frac{\delta}{3 i_{max}}$
11. Generate set of N_1 RR samples $\mathcal{R}_2$
12. **Repeat**
13. $\quad t \leftarrow t_0$, $i \leftarrow 0$
14. $\quad$ **While** $i < t$ **do**
15. $\quad\quad u \leftarrow \arg\max_{v \in V \setminus S_2 \setminus S_1}\left(\hat{\sigma}(S_2 \cup \{v\}) - \hat{\sigma}(S_2)\right)$
16. $\quad\quad$ **If** $u \in S_1$ **then**
17. $\quad\quad\quad t \leftarrow t + 1$
18. $\quad\quad$ **End**
19. $\quad\quad S_2 \leftarrow S_2 \cup \{u\}$, $i \leftarrow i + 1$
20. $\quad$ **End**
21. $\quad$ Calculate $F_l(S_2, \mathcal{R}_2, \delta_2)$ and $F_u(S_2, \mathcal{R}_2, \delta_2)$
22. $\quad$ **If** $\frac{F_l(S_2, \mathcal{R}_2, \delta_2)}{F_u(S_2, \mathcal{R}_2, \delta_2)} \geq 1 - \left(1 - \frac{1}{k}\right)^t - \epsilon$ **then**
23. $\quad\quad$ **Return** S_2
24. $\quad$ **Else**

25. Generate $|\mathcal{R}_2|$ RR samples and add them into $\mathcal{R}_2$
26. **End**
27. **Until** $|\mathcal{R}_2| \geq N_{max}$
28. $S \leftarrow S_1 \cup S_2$
29. **Return** S;

Algorithm evaluation 2.2. The IGS algorithm provides a solution S and an integer t, satisfying: - $Pr[\sigma_U(S) \geq T] \geq 1 - \delta$.

$$- Pr[\ \sigma(S) \geq \left(1 - \left(1 - \frac{1}{k}\right)\right)^t OPT\ ] \geq 1 - \delta.$$

2.3 Experiment and evaluate the results

2.3.1 Experiments

Experimental data: The PhD student performed on 05 data sets of SN. The proposed algorithm is compared with the algorithms DSSA[110], BCT[97], OPIM-C[116] and Degree (baseline greedy algorithm). Evaluate results based on the following criteria: Influence function value, running time and memory usage.

Table 2.1. Dataset's statistics.

Database	#Nodes	#Edges	Types	Avg.degree
netHEPT	15K	59K	Directed	4.1
ENRON	37K	184K	Directed	5
netPHY	37K	181K	Directed	13.4
DBLP	655K	2M	Directed	6.1
Twitter Retweet	1M	2M	Directed	4

2.4.2 Evaluate experimental results

- Influence function value: Figure 2.1 and Table 2.2 show that the IGS algorithm performs better than other algorithms when influencing priority nodes according to a certain threshold T.

Figure 2.1. Looking at the red bars, we can see that IGS affects the set U approximately twice the value of the threshold T on most databases except Re-Tweet but is still higher than T.

Table 2.2 Looking at the values in bold, we can see that although U and S are both large and T increases gradually, the impact on U of IGS is always significantly higher than that of T, even up to more than ten times.

From Figure 2.1 and Table 2.2, we can see $\sigma_U(S)$ of IGS is significantly higher than T and produces better results than state-of-the-art algorithms.

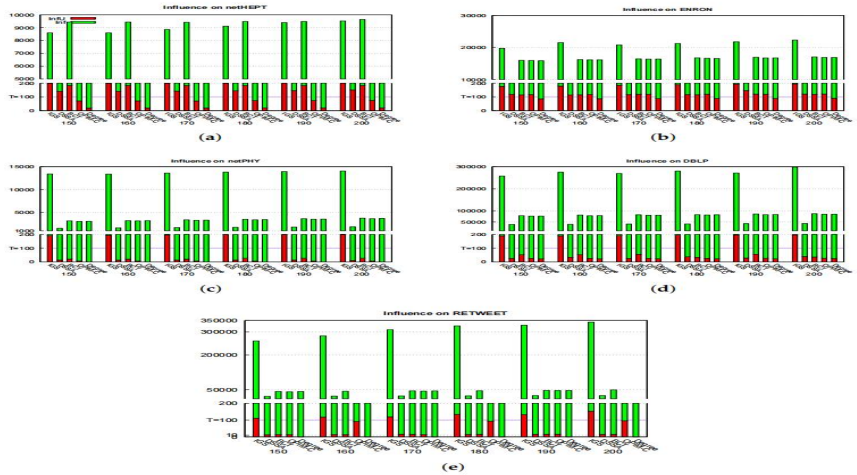


Figure 2.1. Compare the spread of influence with k=100→500, T=100 and U=200

- Running time: Figure 2.2 compares the running time of the algorithms. The IGS algorithm runtime gives the lowest value on the netHEPT, ENRON and netPHY databases. However, IGS remains in the top 3 on DBLP while spending the highest running time on the rest of the dataset to find seed sets 150 and 160 but returns to the top 3 at other values of budget k. IGS only takes about 0.1 seconds to find the seed set in most cases except RETWEET. In general, IGS's run time gives the most stable results and usually runs around the 0.1 second mark.

The time of IGS is fast and stable because of parallel programming and this algorithm costs most of time to find out S_1 while the loop to calculate S_2 usually stops at 1–2 rounds. The TRR sampling technique also helps to quickly identify which seeds will affect to the priority U.

Table 2.2. Comparisons about σ(*S*) and σ$_U$(*S*) between IGS and the others with $k = 500$, $U = 1000$ và $T = 100 \rightarrow 500$.

			Dataset				
	T		**NetHept**	**Enron**	**netPHY**	**DBLP**	**RETWEET**
IGS	100	σ (*S*)	5,666.16	14,267.40	1,865.92	54,033.50	17,307.70
		σ $_U$(*S*)	**1,482.04**	**1,075.77**	**1,192.84**	**1,271.62**	**511.08**
	200	σ (*S*)	5,581.34	14,162.20	1805.26	53,553.90	18,581.50
		σ $_U$(*S*)	**1,478.93**	**1,079.74**	**1,175.32**	**1,267.52**	**491.35**
	300	σ (*S*)	5,645.40	14,284.80	1,773.33	53,240.50	19,459.10
		σ $_U$(*S*)	**1,476.08**	**1,074.30**	**1,153.32**	**1,264.79**	**492.39**
	400	σ (*S*)	5,640.21	14,196.50	1,688.53	52,918.80	18,832.20
		σ $_U$(*S*)	**1,468.48**	**1,075.68**	**1,125.69**	**1,260.31**	**490.46**
	500	σ (*S*)	5,039.45	14,245.50	1,593.66	52,130.90	228,801.00
		σ $_U$(*S*)	**1,238.54**	**1,079.28**	**1,104.20**	**1,252.70**	**994.40**
DSSA		σ (*S*)	4,098.63	9960.35	3230.27	58,197.70	38,253.70
		σ $_U$(*S*)	*1,093.70*	*857.61*	*174.48*	*474.64*	*168.09*
BCT		σ (*S*)	11,088.10	19,901.70	6,675.95	117,197.00	77,316.90
		σ $_U$(*S*)	*1,280.54*	*1,701.60*	*386.49*	*474.64*	*159.77*
OPIM-C		σ (*S*)	3,779.09	19,326.30	6,262.50	112,334	72,026.10
		σ $_U$(*S*)	*600.93*	*894.18*	*194.04*	*459.80*	*173.41*
Degree		σ (*S*)	3824.44	19,349.10	6,345.86	114,249.00	73,936.00
		σ $_U$(*S*)	*292.82*	*779.84*	*164.00*	*260.94*	*22.77*

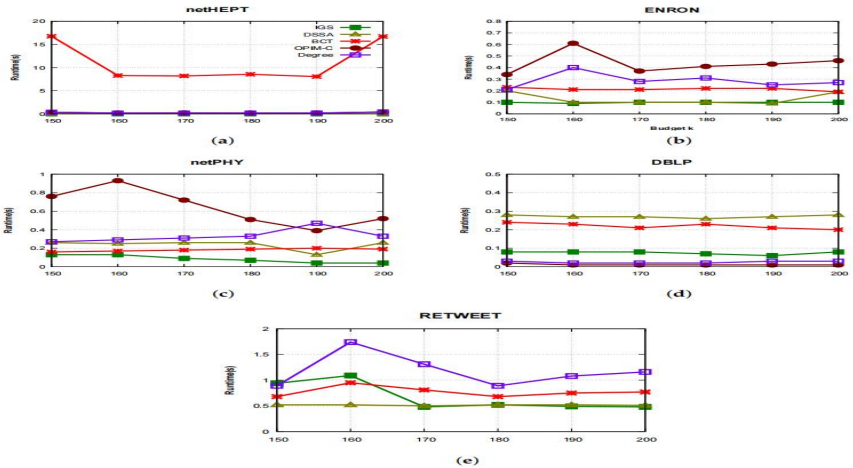


Figure 2.2. Comparisons about Runtime (s) with *k* varies from 150 to 200 between IGS and the other algorithms

- ***Memory usage:*** Table 2.3 illustrates the memory consumption of the IGS algorithm and state-of-the-art methods. The smallest numbers are highlighted in bold while the largest numbers are highlighted in red. The results show that IGS outperforms other algorithms, consuming about four times less memory.

Table 2.3. Memory usage (MB) comparisons between IGS and the others

Dataset	Algorithm	Budget k					
		150	160	170	180	190	200
NetHEPT	**IGS**	**9.90**	**9.90**	**9.90**	**9.89**	**9.89**	**9.95**
	DSSA	22.84	22.84	22.84	22.84	22.84	22.84
	BCT	1023.79	1017.52	1021.60	1012.21	1020.18	1020.74
	OPIM-C	47.76	47.91	48.03	48.11	48.30	48.46
	Degree	49.14	49.18	49.48	49.68	49.86	50.13
ENRON	**IGS**	**16.82**	**16.79**	**16.81**	**16.81**	**16.82**	**16.82**
	DSSA	30.48	28.07	28.07	28.07	28.07	30.48
	BCT	30.35	30.35	30.39	30.39	30.39	30.39
	OPIM-C	27.16	27.20	42.00	27.22	27.25	27.30
	Degree	27.98	28.08	43.77	28.19	28.27	28.41
NetPHY	**IGS**	**15.18**	**15.18**	**15.18**	**15.18**	**15.18**	**15.04**
	DSSA	52.12	52.12	52.12	52.12	38.50	52.14
	BCT	34.82	34.82	34.82	34.82	34.82	34.80
	OPIM-C	87.88	88.39	88.92	89.31	90.26	90.51
	Degree	92.26	92.71	93.33	93.88	94.68	94.98
DBLP	**IGS**	**138.66**	**138.66**	**138.66**	**138.66**	**138.66**	**138.66**
	DSSA	152.90	152.87	152.87	152.91	152.91	152.83
	BCT	162.88	162.87	162.87	162.88	162.88	162.89
	OPIM-C	475.05	373.72	373.78	373.95	477.18	477.51
	Degree	500.87	395.00	394.26	395.35	504.52	505.26
RETWEET	**IGS**	**214.67**	**214.67**	**214.67**	**214.67**	**214.67**	**214.67**
	DSSA	253.14	253.14	253.14	253.14	253.14	253.14
	BCT	282.50	282.50	282.50	282.47	282.50	282.48
	OPIM-C	877.31	874.20	722.91	876.99	886.78	877.80
	Degree	918.53	916.23	756.93	920.00	930.33	921.95

The TRR sampling technique focuses on finding seeds that influence priority U first, then Algorithm 2.2 explores other seeds to push up the seed set. Therefore, algorithm 2.2 saves more memory to run the loop than other algorithms because it does not have to check whether a seed node affects U or not.

Moreover, the condition of $\frac{F_l(S_2, \mathcal{R}_2, \delta_2)}{F_u(S_2, \mathcal{R}_2, \delta_2)} \geq \left(1 - \left(1 - \frac{1}{\mathrm{k}}\right)\right)^{\mathrm{t}} - \epsilon$ helps S_2 generated soon without waiting for the stop condition of the repeat.

Finally, our algorithm, IGS, was designed very well to get a balance between the target to influence on the given priority set and the influence that has to propagate to the largest number of nodes. Hence, running time, memory used and the influence of IGS give significantly high results and even more steadily rather than the others in general.

2.6 Chapter conclusion

In this chapter, the thesis studies the IMP problem with priority constraints arising in a real scenario. The goal of the IMP problem is to select a source set with k nodes whose influence of a given priority set U is greater than a threshold T and the total influence on the nodes is maximized.

To solve this challenge, the thesis models the IMP problem and proposes two algorithms IG and IGS with provable theoretical guarantees. The thesis shows that IG provides an approximate solution $1 - \left(1 - \frac{1}{k}\right)^t$; IGS is an efficient stochastic approximation algorithm based on a sampling method that returns an approximate solution $\left(1 - \left(1 - \frac{1}{\mathrm{k}}\right)^{\mathrm{t}} - \epsilon\right)$ with probability at least 1 - δ with $\epsilon > 0$, $\delta \in (0, 1)$ with probability at least. Real-world social network experiments show that recommendation algorithms outperform algorithms DSSA [110], BCT [97] , OPIM [116] and the underlying greedy algorithm in terms of influence function value, running time, and memory used.

CHAPTER 3
MAXIMUM INFLUENCE ON INFORMATION SPREAD MANY TOPICS WITH LIMITED COST

This chapter continues to research the problem of maximizing the influence of spreading information on many topics with limited costs.

To solve this problem, the author proposes two pass-through streaming algorithms with approximate guarantees: One for the case where element e has only one cost value when added to all i-th tuples, and one for the general case with different values of $c_i(e)$. Experimental results indicate that the proposed algorithms can return competitive results but require significantly less number of queries, complexity, and running time than existing methods.

3.1 Question

Maximizing the *k-Submodular* function has attracted a lot of interest because of its potential in solving various combinatorial optimization problems, such as influence maximization, sensor placement, Choose features and information that maximize coverage. In addition to the unconstrained case, the researchers also addressed the problem under size constraints, matroid constraints, and knapsack constraints. However, these problems do not include some practical applications that customize each element according to its cost requirements as well as cost constraints. The thesis will discuss the following application:

Influence Maximization with k topics subject to a budget constraint. In SN follows an information dissemination model and k topics. Each user has a cost to initiate influence of a topic that indicates how difficult it is to initially influence the corresponding person for that topic. With a cost B, the PhD student considers the problem of finding a set of users,

each initially accepting a topic with a total cost of at most B to maximize the expected number of users activated by at least a topic.

In the above application, the objective functions are *k-submodular*. Although we tried to find a solution that maximizes the function *k-submodular*, the scientists did not mention that each element would have a different cost when added to different sets of solutions with limited costs as shown in the example above. Motivated by that fact in this chapter, the researcher studies a new problem called Budgeted *k*-Influence Maximization problem (B*k*IM), the B*k*IM problem is defined as follows.

***Definition*:** (B*k*IM problem) Given a finite set V, a cost B and a *k-submodular* function f: $(k + 1)^V \mapsto \mathbb{R}_+$ The problem requires finding a solution $s = (S_1, S_2, \dots, S_k) \in (k + 1)^V$, in which an element $e \in V$ has a costs $c_i(e) > 0$ when added into S_i, with total cost $c(s) = \sum_{i \in [k]} \sum_{e \in S_i} c_i(e) \leq B$ so that $f(s)$ is maximized.

3.2 Algorithm proposal

With the continuous increase of input data makes it impossible for input data to be stored in computer memory. Therefore it is important to devise streaming algorithms for the BkIM problem.

The thesis proposes two streaming algorithms including: Deterministic streaming algorithm for the case of costs for all subsets $c_i(e) = c_j(e)$, $\forall e \in V$, $i \neq j$. and Random streaming Algorithm for the General Case.

3.2.1 A deterministic streaming algorithm

Basically, the deterministic streaming algorithm is based on the idea of approximating the optimal value opt of the problem. Then apply the opt value to choose the optimal solution.

Algorithm 3.1: Deterministic streaming algorithm

Input : a k-submodular function f , $B > 0$, $\epsilon \in (0,1/5)$

Output : a solution **s**

if *f is monotone* **then**

  $\alpha \leftarrow \frac{1}{2}$, $\epsilon' \leftarrow 4\,\epsilon$

else

  $\alpha \leftarrow \frac{2}{5}$, $\epsilon' \leftarrow 5\,\epsilon$

end

$(e_{max}, i_{max}) \leftarrow (\emptyset,1)$, $t_j \leftarrow 0\ \forall j \in \mathbb{Z}^+$

foreach $e \in V$ **do**

  $i_e \leftarrow \arg max_{i\in[k]} f((e,i))$

  $(e_{max}, i_{max}) \leftarrow \arg max_{(e_1,i_1) \in \{(e_{max}, i_{max}),(e, i_e\}} f((e_1, i_1))$

  $O \leftarrow \{j \mid f((e_{max}, i_{max})) \le \left(1 + \epsilon'\right)^j \le Bf((e_{max}, i_{max})), j \in \mathbb{Z}^+\}$

  for $j \in O$ **do**

    if $c(\mathbf{s}_j^{t_j}) + c(e) \le B$ **then**

      $i' \leftarrow \arg max_{i\in[k]} f(\mathbf{s}_j^{t_j} \sqcup (e,i))$

      if $\frac{f(\mathbf{s}_j^{t_j} \sqcup (e,i'))}{c(\mathbf{s}_j^{t_j}) + c(e)} \ge \frac{\alpha(1+\epsilon')^j}{B}$ **then**

        $\mathbf{s}_j^{t_j+1} \leftarrow s^{t_j} \sqcup (e, i')$

        $t_j \leftarrow t_j + 1$

      end

    end

  end

end

return $\arg max_{s\in\{(\mathbf{s}_j^{t_j}: j\in O),(e_{max}, i_{max})\}} f(\mathbf{s})$ if f is monotone,

  $\arg max_{\mathbf{s}\in\{(\mathbf{s}_j^{t_j}: i\le t, j\in \mathbf{O}),(e_{max}, i_{max})\}} f(\mathbf{s})$ if f is non-monotone.

Algorithm evaluation 3.1: Algorithm 3.1 is a single-pass streaming algorithm that has $O\left(\frac{kn}{\epsilon} \log n\right)$ query complexity, $O\left(\frac{n}{\epsilon} \log n\right)$ space complexity and provides an approximation ratio:

- 1/4− ϵ when f is monotone;
- 1/5− ϵ when f is non-monotone.

3.2.2 Random streaming algorithm

In this case, each element has a different cost, making the problem more difficult and making it impossible to apply previous algorithms. The PhD student introduces a one-pass data streaming algorithm that provides the expected approximate rate for the BkIM problem.

Algorithm 3.2: Random streaming algorithm

Input : a *k-submodular* function f , $B > 0$, $\epsilon \in (0, 1)$, $\alpha \in (0, 1]$

Output : a solution **s**

$\mathbf{s}^o \leftarrow \mathbf{0}$

$(e_{max}, i_{max}) \leftarrow (\emptyset,1)$, $t_j \leftarrow 0 \; \forall j \in \mathbb{Z}_+$

foreach $e \in \mathrm{V}$ **do**

  $i_e \leftarrow \arg max_{i\in[k]} f((e,i))$

  $(e_{max}, i_{max}) \leftarrow \arg max_{(e_1,i_1) \in \{(e_{max}, i_{max}),(e, i_e\}} f((e_1, i_1))$

  $O \leftarrow \{j \mid f((e_{max}, i_{max})) \leq (1 + \epsilon)^j \leq Bf((e_{max}, i_{max})), j \in \mathbb{Z}^+\}$

  foreach $j \in O$ **do**

    $j \leftarrow \emptyset$

    foreach $j \in O$ **do**

      if $c(\mathbf{s}_j^{t_j}) + c_i(e) \leq B$ *and* $\frac{f(\mathbf{s}_j^{t_j} \sqcup (e,i)) - f(\mathbf{s}_j^{t_j})}{c_i(e)} \geq \frac{\alpha(1+\epsilon)^j}{B}$ ***then***

        $p_j \leftarrow \frac{f(\mathbf{s}_j^{t_j} \sqcup (e,i)) - f(\mathbf{s}_j^{t_j})}{c_i(e)}$

        $j \leftarrow j \cup \{i\}$

      end

    end

    $T \leftarrow \sum_{i \in J} p_i^{|J|-1}$

    Select a possition $i \in J$ with probability $\frac{p_l^{|J|-1}}{T}$

    $\mathbf{s}_j^{t_j+1} \leftarrow \mathbf{s}_j^{t_j} \sqcup (e, i)$

    $t_j \leftarrow t_j + 1$

  end

end

return $\arg max_{s\in\{(\mathbf{s}_j^{t_j}; j\in O),(e_{max}, i_{max})\}} f(\mathbf{s})$ if f is monotone,

$\arg max_{\mathbf{s}\in\{(\mathbf{s}_j^i: j\in\mathbf{0},\, i\leq t),(e_{max}, i_{max})\}} f(\mathbf{s})$ if f is non-monotone

Algorithm evaluation 3.2: Random streaming algorithm is one pass streaming algorithm that has $O(\frac{kn}{\epsilon}\log n)$ query complexity, $O(\frac{\mathrm{n}}{\epsilon}\log n)$ space complexity and

- *If f is monotone* $\mathbb{E}[f(s)] \geq (min\left\{\frac{\alpha}{2}, \frac{(1-\alpha)k}{(1+\beta)k-\beta}\right\} - \epsilon)\boldsymbol{opt}$. *The right hand side is maximized to* $(\frac{v}{3+\beta-\frac{\beta}{k}} - \epsilon)\boldsymbol{opt}$ *when* $\alpha = \frac{2}{3+\beta-\frac{\beta}{k}}$.

- *If f is none-monotone* $\mathbb{E}[f(s)] \geq (min\left\{\frac{\alpha}{2}, \frac{(1-\alpha)k}{(1+\beta)k-\beta}\right\} - \epsilon)\boldsymbol{opt}$. *The right hand side is maximized to* $(\frac{v}{3+2\beta-\frac{2\beta}{k}} - \epsilon)\boldsymbol{opt}$ *when* $\alpha = \frac{2}{3+2\beta-\frac{2\beta}{k}}$.

3.3 Experiment and evaluate the results

In this section, the PhD student compares the performance of the proposed algorithms with the greedy algorithm [26] in the application of BkIM: Influence Maximization with k topics subject to the budget constraint (IMkB) on three main metrics: price value of objective function, number of queries and running time.

Definition 3.2: (IMkB *problem*) Assume that each user *u* has a cost $c_i(u)$ for i-th topic which manifests how hard it is to initially influence the respective person for that topic. Given the budget *B*, the problem asks to find a seed set $\boldsymbol{s}$ with $c(\boldsymbol{s}) \leq B$ so that $\sigma(\boldsymbol{s})$ is maximal.

- *Experimental data:* We use the Facebook social network dataset from SNAP (2020). The network contains 4,039 nodes and 88,234 edges.

- *Experiment results:* For the purpose of providing a comprehensive experiment, we divide the experiment into two cases: the special case when β=1 and the general case.

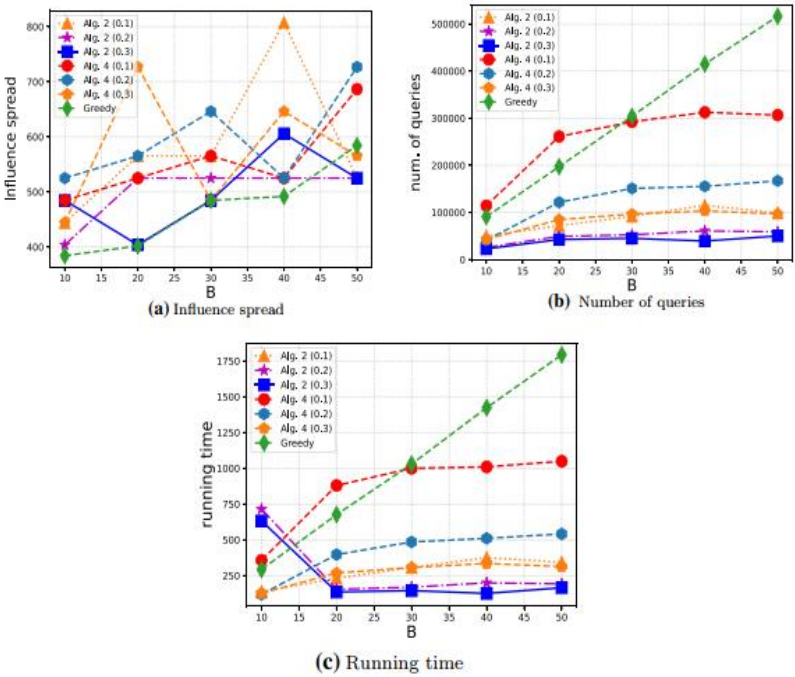


Figure 3.1 Performance of algorithms for IMkB when $\beta=1$

Figure 3.1 shows the performance of algorithms for $\boldsymbol{\beta}$ =1. With solution quality (influence spread), our algorithms outperform Greedy algorithm [26] in most cases. For query complexity, the proposed algorithms completely outperform the Greedy algorithms by a large margin. They require up to 10 times fewer queries than the Greedy algorithm.

We can find that Algorithm 3.2 provides the better solution than Algorithm 3.1 with the same value of ϵ for most cases. This is consistent with the theoretical analysis. However, Algorithm 3.2 requires more queries and running time than Algorithm 3.1. It might be because of Algorithm 3.1 that reaches to the budget B faster than Algorithm 3.2.

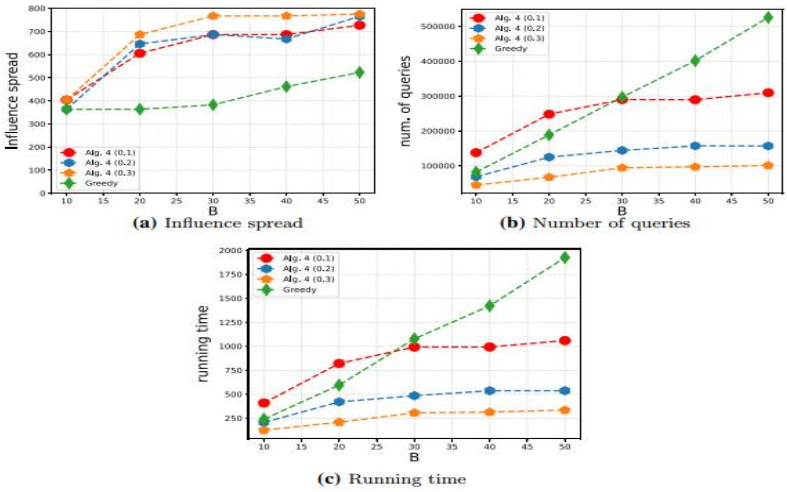


Figure 3.2: Performance of algorithms for IMkB in general case

3.4 Chapter conclusion

This chapter studies the B*k*IM problem, generalizing the problem BkIM. The PhD student proposes two streaming algorithms that traverse the data once with provable guarantees. The core of the algorithms is to exploit the relationship between proposed solutions and the optimal solution by analyzing intermediate quantities and using new probability distributions then comparing the contribution values (objective marginal expenditure per cost) to a certain appropriate threshold.

To examine the performance of the proposed algorithms in practice, the researcher conducted some experiments on the application "Influence Maximization with k topics subject to the budget constraint". Experimental results have shown that the proposed algorithms not only return good solutions in terms of quality requirements but also have a significantly smaller number of queries than the state-of-the-art greedy algorithms.

CONCLUDE

The thesis has completed the research goal which is: Researching influence maximization problems with priority constraints (IMP) and Budgeted *k*-Influence maximization (B*k*IM). The thesis has proposed four effective algorithms to solve the given problems. Research is published in conference proceedings and prestigious journals in the fields of Optimization and Information Technology domestically and internationally, some publications are in the SCIE and SCOPUS categories.

Problems that maximize the influence of information spread on SN are difficult problems with large data and complex constraints on space, time and cost.

The main contributions of the thesis include:

1. Proposing the problem "Influence maximization with priority constraint" - IMP. Although the objective function is still a monotonic function and a Submodular function, when considering the priority constraint, the latest IM algorithms do not To solve this challenge, the thesis proposes two algorithms IG and IGS with provable theoretical guarantees.

2. Proposing the problem "Budgeted *k*-Influence maximization" - BkIM. The thesis proposes a deterministic streaming algorithm and a random streaming algorithm that browses data once to provide theoretical limits of the BkIM problem.

Research direction of the thesis: The PhD student continues to research and propose more effective solutions for the proposed problems. Develop applications of research results that can be applied to SN. Research variations with practical applications for IM, ID and IB problems. Propose algorithms that are powerful enough to handle social networks with billions of nodes and edges in acceptable time.

LIST OF THE PUBLICATIONS RELATED TO THE DISSERTATION

1. Canh V. Pham, Dung K. T. Ha, **Quang C. Vu**, Anh N. Su and Huan X. Hoang (2020). *Influence Maximization with Priority in Online Social Networks* Algorithms 2020, tập 13, số 183; doi:10.3390/a13080183 (Tác giả liên hệ).
2. **Vũ Chí Quang**, Phạm Văn Cảnh, Hà Thị Kim Dung, Phạm Văn Dũng, Nguyễn Như Sơn. *"Tối đa ảnh hưởng với sự ưu tiên trên mạng xã hội trực tuyến"*, Hội thảo quốc gia lần thứ XXIII **(@)**: Một số vấn đề chọn lọc của Công nghệ thông tin và truyền thông, Quảng Ninh, ngày 5-6/11/2020.
3. Pham, C.V., **Vu, Q. C.**, Ha, D.K.T., Nguyen, T.T. (2021). *Streaming Algorithms for Budgeted k-Submodular Maximization Problem*. In: Mohaisen, D., Jin, R. (eds) Computational Data and Social Networks. CSoNet 2021. Lecture Notes in Computer Science(), vol 13116. Springer, Cham.
4. Pham, C.V., **Vu, Q. C.**, Ha, D.K.T., Nguyen, T.T. et al (2022). *Maximizing k-submodular functions under budget constraint: applications and streaming algorithms*. Journal of Combinatorial Optimization 44, 723–751.
5. **Vũ Chí Quang**, Phạm Văn Dũng, Nguyễn Thị Tuyết Trinh, Nguyễn Như Sơn.*"Tối ưu hàm k-submodular ứng dụng trong bài toán Cực đại ảnh hưởng của thông tin nhiều chủ đề lan truyền trên mạng xã hội"* Hội thảo quốc gia lần thứ XXVI: Một số vấn đề chọn lọc của Công nghệ thông tin và truyền thông – Bắc Ninh, 5-6/10/2023.