Efficient adaptive approximation algorithms in online social networks

Huang, Keke

2019

Huang, K. (2019). Efficient adaptive approximation algorithms in online social networks.
Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/136951>

<https://doi.org/10.32657/10356/136951>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0
International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 18:39:01 SGT

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

EFFICIENT ADAPTIVE APPROXIMATION ALGORITHMS IN ONLINE SOCIAL NETWORKS

HUANG Keke

School of Computer Science and Engineering (SCSE)

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2019

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

17 July 2019

.....

Date

huang keke

.....

Huang Keke

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

17 July 2019

.....
Date

.....
Sun Aixin

Authorship Attribution Statement

This thesis contains material from 1 paper published in the following peer-reviewed conference in which I am listed as an author.

Chapter 4 is published as Jing Tang, Keke Huang, Xiaokui Xiao, Laks V.S. Lakshmanan, Xueyan Tang, Aixin Sun, and Andrew Lim. Efficient Approximation Algorithms for Adaptive Seed Minimization. Proceedings of the ACM SIGMOD International Conference on Management of Data, 1096-1113 (2019). DOI: 10.1145/3299869.3319881

The contributions of the co-authors are as follows:

- A/Prof Xiaokui Xiao provided the initial research direction and gave feedbacks to the algorithms designed.
- As co-first author, Dr. Jing Tang and I designed the algorithms, conducted theoretical analysis, and prepared the manuscript draft together.
- I designed experiments, wrote the source code, and conducted experiments.
- A/Prof Laks V.S. Lakshmanan provided insightful comments on the draft, particularly experiment design.
- A/Prof Xueyan Tang, Aixin Sun and Andrew Lim revised and edited the draft and provided various feedbacks.

17 July 2019

.....
Date

huang keke

.....
Huang Keke

Abstract

Online social networks (OSNs) have witnessed their prosperous developments in recent years. Based on the established relations among individuals in OSNs, ideas and opinions can be spread via a word-of-mouth effect. To exploit this effect, many companies have taken OSNs as the major advertising channels to promote their products, which has motivated substantial research on viral marketing. Among them, *influence maximization*, *seed minimization*, and *profit maximization* have been extensively studied in the past decades. These three problems are mainly studied in the *non-adaptive setting* in the literature, i.e., seed user selection phase does not involve any observations on influence propagation in reality. In this thesis, we extend profit maximization into a generalized version as *target profit maximization* and investigate these three problems in the *adaptive setting*, formulated as *adaptive influence maximization*, *adaptive seed minimization*, and *adaptive target profit maximization* respectively.

Firstly, we study the adaptive influence maximization (IM) problem. Given a social network G and an integer k , the influence maximization problem asks for a seed set S of k nodes from G to maximize the expected number of nodes activated via an influence propagation. In the adaptive setting, the k seed nodes are selected in batches of equal size b , such that the i -th batch is identified after the actual influence results of the former $i - 1$ batches are observed. We propose the first practical algorithm for the adaptive IM problem that could provide the *worst-case* approximation guarantee of $1 - e^{\alpha(\varepsilon-1)}$ with high probability, where $\varepsilon \in (0, 1)$ is a user-specified parameter and $\alpha = 1 - (1 - 1/b)^b$ is set by the batch size b . Our approach is based on a novel general framework ADAPTGREEDY which could be instantiated by any existing non-adaptive IM algorithms with expected approximation guarantee. In this regard, we design a novel non-adaptive IM algorithm called EPIC which not only provides strong expected approximation guarantee, but also

presents superior performance compared with the existing IM algorithms. Meanwhile, we clarify some existing misunderstandings in recent work and shed light on further study of the adaptive IM problem.

Subsequently, we consider the adaptive seed minimization (SM) problem. As a dual problem of influence maximization, the seed minimization problem asks for the minimum number of seed nodes to influence a required number η of users in a given social network G . In the adaptive setting, the seed nodes are selected in several batches, such that the choice of current batch may exploit information about the actual influence of the previous batches. We devise a novel algorithm, ASTI, which addresses the adaptive seed minimization problem in $O\left(\frac{\eta \cdot (m+n)}{\varepsilon^2} \ln n\right)$ expected time and offers an approximation guarantee of $\frac{(\ln \eta + 1)^2}{(1 - (1 - 1/b)^b)(1 - 1/e)(1 - \varepsilon)}$ in expectation, where η is the targeted number of influenced nodes, b is size of each seed node batch, and $\varepsilon \in (0, 1)$ is a user-specified parameter. To the best of our knowledge, ASTI is the first algorithm that provides such an approximation guarantee without incurring prohibitive computation overhead.

Eventually, we explore the adaptive target profit maximization (TPM) problem. Given a social network G , the profit maximization (PM) problem asks for a set of seed nodes to maximize the profit, i.e., revenue of influence spread less the cost of seed selection. The TPM problem, which generalizes the PM problem, aims to select a subset of seed nodes from a given target user set T to maximize the profit. In the adaptive setting, the seed users are selected through multiple batches, such that the selection of current batch exploits the knowledge of actual influence in the previous batches. To acquire an overall understanding, we study adaptive TPM under the *oracle model* and the *noise model*, to which we propose ADG and ADDATP algorithms with strong theoretical guarantees respectively. In addition, we propose the idea of *hybrid error* to better handle the sampling errors under the noise model and design a novel algorithm HATP that boosts the efficiency of ADDATP significantly.

Acknowledgments

First and foremost, I would like to express my sincerest gratitude towards my supervisors, Prof. Xiaokui Xiao and Prof. Aixin Sun. It is their guidance of patience on my PhD research that makes the accomplishment of this thesis possible. In particular, Prof. Xiao's rigorous attitude, the pursuit of high standards, and unlimited enthusiasm for research have impressed and shaped me deeply. Prof. Sun, who comes from the hometown of Confucius in China, well interprets the charms of being a professor. It is his continuous help and care that support me forward. Their positive influence on me would be the priceless treasure for my future career and life.

Then, I would like to thank my collaborators, especially Dr. Jing Tang. We have collaborated for two years. I learn so much from his great intelligence and experience on research such that I could conquer the challenges in my research. They also include: Prof. Laks V.S. Lakshmanan, Prof. Xueyan Tang, Prof. Sibbo Wang, and Prof. Kai Han. It is a great honor for me to work with them and learn from them.

Besides, I would like to thank to my fellow students in the Data Management and Analytics Lab (DMAL) for their concern and companionship. They include but not limited to: Mr. Xi Lin, Mr. Renchi Yang, Dr. Jing Li, Dr. Xin Zheng, Miss Grace Lee, Mr. Zhe Chen, Mr. Ting Lin, and Mr. Phan Cong Minh. It is a pleasant experience to live and work with them. Meanwhile, I am also truly grateful for the kind help from the lab manager Thomas and all the administrative staff.

Next, I would express my gratefulness for my friends who company and encourage me all the way. They include but not limited to Mr. Zhenyu Zhao, Miss Linkai Feng, Miss Liu Wang, Miss Kaicheng Chen, Mr Zhenxing Xiao, Miss Jingfang Yuan, and Mr Chenggang He. I really value those memorable days we spent together in my PhD life.

Last but not least, I would like to thank my family for their selfless love and care such that I could focus on my PhD study. My priority will always be them.

Contents

List of Figures	vii
List of Tables	viii
1. Introduction	1
1.1 Background	1
1.2 Problems and Motivations	3
1.3 Approaches and Contributions	6
1.4 Thesis Organization	9
2. Literature Review	11
2.1 Influence Propagation in Online Social Networks	11
2.1.1 Influence Spread and Realization	11
2.1.2 The State of the Art on Spread Estimation	13
2.1.3 Advantage of Adaptivity over Non-adaptivity on Seed Selection	15
2.2 Influence Maximization	16
2.2.1 Non-adaptive Influence Maximization	17
2.2.2 Adaptive Influence Maximization	21
2.3 Seed Minimization	23
2.3.1 Non-adaptive Seed Minimization	23
2.3.2 Adaptive Seed Minimization	25
2.4 Profit Maximization	26
2.5 Summary	29
3. Adaptive Influence Maximization	31
3.1 Preliminary	32

3.2	Our Solution	32
3.2.1	Solution Framework	33
3.2.2	Expected Approximation Guarantee	34
3.2.3	Worst-case Approximation Guarantee	35
3.3	Instantiations of ADAPTGREEDY	39
3.3.1	Design of Instantiation Algorithm	39
3.3.2	Theoretical Analysis	45
3.3.3	Expected and Worst-Case Approximation Guarantee	49
3.4	Misclaims in Previous Work [1, 2]	50
3.5	Performance Evaluation	51
3.5.1	Experimental Setting	51
3.5.2	Comparison of the Influence Spread	53
3.5.3	Comparison of the Running Time	56
3.6	Conclusion	59
4.	Adaptive Seed Minimization	60
4.1	Preliminary	61
4.1.1	Adaptive Seed Minimization	61
4.1.2	Truncated Influence Spread	62
4.2	ASTI: A General and Efficient Framework	64
4.2.1	Framework Overview	64
4.2.2	Truncated Influence Maximization	65
4.2.3	Multi-Root Reverse Reachable Set	67
4.2.4	The Design of TRIM	69
4.2.5	Theoretical Analysis	71
4.3	Extensions of Batch Size	73
4.3.1	Batched Version of TRIM	73
4.3.2	Theoretical Analysis	74
4.4	Experiments	75
4.4.1	Experimental Setting	76
4.4.2	Results under the IC model	78
4.4.3	Results under the LT model	81

4.4.4	Discussions on Related Spreads	84
4.5	Summary	85
5.	Adaptive Target Profit Maximization	88
5.1	Preliminary	89
5.2	Adaptive Double Greedy Algorithm	90
5.2.1	Double Greedy	91
5.2.2	Adaptive Double Greedy under the Oracle Model	91
5.2.3	Adaptive Double Greedy under the Noise Model	95
5.3	Optimization with Hybrid Error	99
5.3.1	Rationale of Optimization	99
5.3.2	Description of HATP	101
5.3.3	Theoretical Analysis of HATP	101
5.4	Experiments	102
5.4.1	Experimental Setting	103
5.4.2	Comparison of Profit	104
5.4.3	Comparison of Running Time	107
5.4.4	Comparison of Profit with Predefined Cost	109
5.5	Summary	110
6.	Conclusion and Future Work	112
6.1	Conclusion	112
6.2	Future Directions	113
6.2.1	Sample Reuse without Bias	113
6.2.2	Online Processing for Dynamic Social Graphs	114
6.2.3	Profit Maximization Beyond Non-negativity Assumption	115
A.	Proofs	117
	Publications	135
	References	136

List of Figures

1.1	Thesis Overview.	9
2.1	A social network and three of its possible realizations.	13
2.2	Input graph G and its corresponding random graphs.	14
2.3	Adaptive vs. non-adaptive seed selection with $k = 2$	16
3.1	Spread vs. batch size.	54
3.2	Spread vs. seed size.	55
3.3	Running time vs. batch size.	57
3.4	Running time vs. seed size.	58
4.1	A social graph and all of its possible realizations.	64
4.2	Degree distribution of tested datasets.	76
4.3	Number of seed nodes vs. threshold under the IC model.	79
4.4	Running time vs. threshold under the IC model.	80
4.5	Number of seed nodes vs. threshold under the LT model.	82
4.6	Running time vs. threshold under the LT model.	83
4.7	Spread for 20 realizations on NetHEPT.	84
4.8	Spread vs. threshold under the IC model.	86
4.9	Marginal truncated spread under the IC model.	87
5.1	Profit in degree-proportional cost.	105
5.2	Profit in uniform cost.	106
5.3	Running time in degree-proportional cost.	107
5.4	Running time in uniform cost.	108
5.5	Profits of HATP and NDG on LiveJounral.	109
5.6	Profits of HATP and NSG on LiveJounral.	109

List of Tables

2.1	Frequently used notations	12
3.1	Frequently used notations	33
3.2	Dataset details. ($K = 10^3, M = 10^6$)	51
4.1	Frequently used notations.	61
4.2	Dataset details. ($K = 10^3, M = 10^6$)	76
4.3	Improvement ratio of ASTI over ATEUC	78
5.1	Frequently used notations	89
5.2	Dataset details. ($K = 10^3, M = 10^6$)	103

Chapter 1

Introduction

1.1 Background

Online social networks (OSNs) are in a booming development with a rapid increase of active users for the past decades. One online statistic site [3] reports the most popular social networks worldwide as of April 2019. According to the statistics on monthly active users, there are 2.32 billion users on Facebook, 1.60 billions users on WhatsApp, and 1.10 billions users on Wechat, to name a few. Based on the established relations among individuals in OSNs, ideas and opinions can be spread through social networks via a word-of-mouth effect. To exploit this effect, many companies have taken OSNs as the major advertising channels to promote their products. For example, advertisers could provide a few individuals (referred to as “seed nodes”) in social networks with free product samples, in exchange for them to spread good words about the product to others, so as to create a large cascade of influence on other social network users via word-of-mouth recommendations. This phenomenon is formulated as the *influence maximization (IM)* problem [4] which aims to select a certain number of seed nodes to maximize the influence propagation created. At the same time, advertisers might also want to know the minimum number of seed users, so as to draw sufficient attention. As a dual problem of IM, this demand is formulated as a *seed minimization (SM)* problem [5], which asks for the minimum number of seed nodes required to influence at least a predefined number of users. In the meantime, when the monetary factor is taken into consideration for the influence propagation, there is *investment cost* (e.g. cashback rewards, coupons, or discounts) associated with each seed user and *revenue* associated with the number

of influenced users. In this regard, advertisers aim to maximize the *profit*, i.e., the final revenue less the total investment costs. This problem is studied as the *profit maximization (PM)* problem [6, 7] which seeks for a proper set of seed users so as to maximize the final expected profit. However, this vanilla PM problem overlooks one fact that even though the advertisers have the full knowledge about the whole network, they are likely to only have access to a fraction of users [8], which is quite common in marketing applications. For example, companies advertise new products to users in their subscription mailing list, or new shop owners provide free samples to the popularities or celebrities who visit their store on site, to name a few. To circumvent this potential issue, we extend the vanilla profit maximization into a more generalized version and propose the *target profit maximization (TPM)* problem. Given a target set with nonnegative expected profit where each user in this set is associated with a cost, the TPM problem aims to select a subset from the target set to maximize the expected profit. The target user set can be either made up by current accessible users in the social network or selected by current seed selection algorithms. In particular, if the target set contains all users in the social network, TPM then degenerates to PM. The three problems are closely related but each focuses on a different aspect. Specifically, both IM and SM seek for the most influential nodes as seed set. IM provides limited budget on seed selection, such that it aims to maximize the influence spread achievable within the given budget. On the contrary, SM sets no constraints on the budget but it requires to satisfy a predefined quantity of influence. As with TPM, it is defined based on IM but considers an extra monetary factor, i.e., the cost of each seed node. By taking this cost into consideration, the objective function becomes profit maximization in the TPM problem which is different from influence maximization in the IM problem. In summary, the three problems focus on different aspects on advertising. If users aim to promote new products to as many people as possible, they are expected to adopt the methods for influence maximization. If users need to advertise new products to at least a certain number of potential customers, they are advised to adopt the methods for seed minimization. Finally, if users consider the costs on advertising and seek to maximize the final profits, they shall utilize the methods for target profit maximization.

In the literature, influence maximization, seed minimization, and target profit maximization are more studied under the *non-adaptive* setting, where seed nodes are selected

in one batch before observing the results on how they influence the other users in reality. On the contrary, under the *adaptive* setting, the seed nodes are selected in multiple batches, such that the later batch is identified based on the observation of the actual influence results of all the former batches. The procedures of non-adaptive and adaptive seed selection are depicted in Chapter 2.1.3. By taking advantage of the knowledge on the past observations, the performance could be further improved due to the *adaptivity gap* [2, 6, 9]. Motivated by this fact, in this thesis, we conduct an in-depth exploration on the three problems under the adaptive setting, which, however, turns out to be quite challenging and intricate. In what follows, we briefly introduce the problems under the adaptive setting and propose efficient adaptive approximation algorithms with non-trivial theoretical guarantees to address them respectively.

1.2 Problems and Motivations

Adaptive Influence Maximization. Given a social network G , a propagation model M on how nodes in G may influence each other, and a positive integer k , the influence maximization (IM) problem asks for a seed set S of k nodes from G to maximize the expected number of nodes influenced under the propagation model M which captures the uncertainty of influence propagation in G . The IM problem is firstly formulated by the seminal work [4]. In this work, Kempe et al. introduce two basic propagation models, namely the *independent cascade (IC)* and *linear threshold (LT)* models and prove that it is NP-hard to derive the optimal solution for the IM problem. Meanwhile, they also prove that the spread function is *monotone* and *submodular* under the two models. By exploiting the monotonicity and submodularity, greedy algorithm could achieve an $1 - 1/e$ -approximation guarantee. Since then, an abundance of techniques have been proposed for non-adaptive IM [4, 10–19].

Contrarily, in adaptive IM, the k seed nodes are selected in batches of equal size b , such that the i -th batch is identified after the actual influence results on the stochastic propagation process of the former $i - 1$ batches are observed. To interpret the uncertainty of the influence process, we consider the set Ω of all possible *realizations* ϕ , each of which represents a possible scenario of influence propagation among the nodes in G . For

example, once we have a set of seed users and then put into the marketing for advertising, we then observe how the influence spread is propagated. If node u is activated (or as a seed node) and influences its outgoing neighbor v , edge $\langle u, v \rangle$ is kept; otherwise $\langle u, v \rangle$ is removed. The residual graph is called a realization when the influence propagation terminates. In a nutshell, a realization represents one possible scenario of the influence propagation in reality. Adaptive IM seeks to *activate* (i.e., influence) a seed set S of k nodes that can maximize the expected number of influenced individuals over all the realizations in Ω . Golovin et al. [9] are the first to study IM under the adaptive setting. Specifically, they consider that (i) the social network conforms to a realization ϕ sampled from a realization space Ω , but (ii) ϕ is not known to the advertiser before the selection of the first seed node. Then, after the i -th seed node v_i is chosen, the part of ϕ relevant to $\{v_1, v_2, \dots, v_i\}$ (i.e., the nodes that they can influence in ϕ) is revealed to the advertiser, based on which she can (i) eliminate the realizations in Ω that contradict what she observes, and (ii) select the next seed node as one that has a large expected influence over the remaining realizations.

Based on the above analysis, Golovin et al. [9] propose a simple greedy algorithm for adaptive IM that returns a seed set S whose influence is at least $1 - 1/e$ of the optimum under the case that only one seed is selected in each batch (i.e., $b = 1$). Nevertheless, the algorithm requires knowing the *exact* expected influence of every node, which is impractical since the computation of expected spread is #P-hard in general [20, 21]. Motivated by this fact, Vaswani and Lakshmanan [22] extend Golovin et al.'s model by allowing selecting $b \geq 1$ seed nodes in each batch, and by accommodating errors in the estimation of expected spreads. Their method returns an $(1 - e^{-(1-1/e)^2/\gamma})$ -approximation under this setting, where γ is certain number bigger than 1. However, even this relaxed approach is still impractical in that its requirements on the accuracy of expected spread estimation cannot be met by any existing algorithms (see Section 2.2 for more details). This motivates us to develop a practical and efficient algorithm with theoretical guarantee to address the adaptive IM problem.

Adaptive Seed Minimization. As a dual problem of influence maximization problem, the seed minimization (SM) problem asks for the minimum number of *seed nodes* to influence at least a required number η of users, taking into account the randomness in

the influence propagation process. Existing work on seed minimization mostly focuses on the *non-adaptive* setting [5, 23, 24], which requires that all seed nodes should be selected in one batch without observing the actual influence of any node, i.e., no randomness in the influence propagation process can be removed until all seed nodes are fixed. As a consequence of the non-adaptiveness, these solutions may return a seed set that fails to influence at least η nodes in the actual propagation process, or may select an excessive number of seed nodes that generate an actual influence spread much larger than required.

To address the above issue, Vaswani and Lakshmanan [22] are the first to consider the seed minimization under the *adaptive* setting, where (i) the seed nodes are selected one by one, and (ii) before selecting the i -th seed node, the actual influence of the first $i - 1$ seed nodes can be observed, i.e., we may optimize the choice of the i -th seed node to influence those users that have not been influenced by the previous $i - 1$ seed nodes. Such an adaptive strategy ensures that (i) the seed set returned always achieves the required number of influenced users (since the actual influence of each seed node is known after selection), and (ii) the number of seed nodes would not be excessive (because we can stop selecting seed nodes as soon as the targeted influence η is achieved).

However, the solution in [22] requires that the expected influence of any seed set should be estimated with extremely high accuracy, which results in prohibitive computation overhead, as aforementioned. Furthermore, the solution does not provide any non-trivial approximation guarantee, due to an ineffective approach used to select each seed node in the adaptive setting (See Section 2.3 for more details). Therefore, it remains an open problem to devise efficient approximation algorithms for adaptive seed minimization.

Adaptive Target Profit Maximization. Given a social network $G = (V, E)$ and a target user set $T \subseteq V$ with nonnegative expected profit where each node $u \in T$ is associated with a cost $c(u)$ for seed selection, the target profit maximization (TPM) problem asks for a seed set $S \subseteq T$ that maximizes the expected profit, i.e., the expected spread of S less the total investment cost $c(S)$. The TPM problem degenerates to the PM problem if the target set T contains all users in the social network, i.e., $T = V$. Existing work on profit maximization mostly focuses on the *non-adaptive* setting [6, 7, 25, 26], where all seed nodes are selected in one batch without any knowledge on how they may influence other users. As a consequence of the non-adaptiveness, the maximal possible

profit might not be achieved. On the contrary, if we could observe the real-time feedback from the market and response with a smarter seed selection procedure, we could make further improvement on the final profit due to the adaptivity gap. Motivated by this fact, we aim to propose an adaptive strategy on seed selection for the adaptive TPM problem.

1.3 Approaches and Contributions

Adaptive Influence Maximization. Motivated by the deficiency of existing techniques, we conduct an extensive study on the adaptive IM problem, and propose the first practical solution with rigorous theoretical analysis. Meanwhile, we clarify existing common misunderstanding towards adaptive IM and lay solid foundations for further study (See more details in Chapter 3). Specifically, our contributions include the following.

First, we propose ADAPTGREEDY, a framework that enables us to construct strong approximation solutions for adaptive IM using existing non-adaptive IM methods as building blocks. In particular, we prove that ADAPTGREEDY achieves a *worst-case* approximation guarantee of $1 - e^{\alpha(\varepsilon-1)}$ with high probability, where $\varepsilon \in (0, 1)$ is a user-specified parameter and $\alpha = 1 - (1 - 1/b)^b$ is set by the batch size b . The derivation of this approximation result requires a non-trivial extension of the existing theoretical results on adaptive algorithms (e.g., [9] and [22]) with Azuma-Hoeffding inequality [27], since ADAPTGREEDY imposes far fewer constraints on the building blocks used for adaptive IM. Moreover, we show that ADAPTGREEDY can also provide an *expected* approximation guarantee of $1 - e^{\alpha(\varepsilon-1)}$. Meanwhile, our analyses uncover some crucial gaps in two recent works [1, 2] and shed light on the future work of the adaptive IM problem.

Second, we conduct an in-depth analysis on how the framework ADAPTGREEDY could be instantiated with the state-of-the-art non-adaptive IM algorithms. The overall approximation guarantee of ADAPTGREEDY relies on the *expected* approximation guarantee of the non-adaptive IM algorithm used by ADAPTGREEDY. However, existing non-adaptive IM algorithms do not benefit ADAPTGREEDY in this regard, as there is no known result on their expected approximation guarantees. Motivated by this, we develop

a new non-adaptive IM method, EPIC, that provides an attractive expected approximation ratio by utilizing martingale stopping theorem [27]. We establish ADAPTGREEDY's performance guarantee when it is instantiated with EPIC.

Adaptive Seed Minimization. We address this open problem with ASTI, a novel framework tailored for *adaptive seed minimization*. The key idea of ASTI is to adaptively choose the seed node with the maximum expected *truncated influence spread* in each round of seed selection. For each possible realization $\phi \in \Omega$, the influence spread of a seed set S , denoted as $I_\phi(S)$ is the number of nodes influenced by S , while the truncated influence spread of S is defined as $\Gamma_\phi(S) = \min\{\eta, I_\phi(S)\}$. We consider $\Gamma_\phi(S)$ instead of $I_\phi(S)$ because, intuitively, the extra influence spread beyond η is useless for fulfilling the requirement on influence. (In fact, as we show in Chapter 4, the extra influence spread may even lead to incorrect choice of seed nodes, and hence, it has to be ignored.)

When developing algorithms under the ASTI framework, the key challenge that we face is the design of methods to accurately estimate a seed set S 's expected truncated influence spread over a given set of possible realizations. We show that existing methods [10–13, 17, 28] for estimating un-truncated influence spread cannot be applied in our truncated setting, since they are unable to take into account the effect of truncation by η . Motivated by this, we propose a novel sampling method based on the concept of *multi-root reverse reachable (mRR)* sets, and prove that our method provides non-trivial guarantees in terms of the efficiency and accuracy of truncated influence estimation. Building upon this sampling method, we develop TRIM, an algorithm for maximizing truncated influence spread with a provable approximation guarantee of $(1 - 1/e)(1 - \varepsilon)$. We show that instantiating ASTI using TRIM leads to strong theoretical guarantees of $\frac{(\ln \eta + 1)^2}{(1 - 1/e)(1 - \varepsilon)}$ for adaptive seed minimization with expected time complexity of $O\left(\frac{\eta(m+n)}{\varepsilon^2} \ln n\right)$. Additionally, TRIM can be extended into a batched version TRIM-B that selects a batch of b nodes in each round, so as to accelerate seed selection. ASTI instantiated by TRIM-B provides an approximation guarantee of $\frac{(\ln \eta + 1)^2}{(1 - (1 - 1/b)^b)(1 - 1/e)(1 - \varepsilon)}$ with the same time complexity as TRIM.

Adaptive Target Profit Maximization. To acquire an overall understanding, we first study this problem in *oracle model* where the profit of any node set can be obtained in $O(1)$ time. We then consider it in a more practical *noise model* where the expected

profits (or expected spreads) can only be estimated through sampling. Eventually, we design an efficient algorithm with nontrivial approximation guarantees to address the adaptive TPM problem. In summary, our major contributions are as follows.

First, we propose the adaptive target profit maximization. To the best of our knowledge, we are the first to consider profit maximization under the adaptive setting and propose the adaptive target profit maximization problem, aiming to provide inspirational insights for future research. By utilizing the knowledge from the market feedback, we study adaptive strategies to select more effective seed nodes to maximize the expected profit.

Second, we conduct rigorous theoretical analyses on adaptive TPM under both oracle and noise model. To gain a comprehensive understanding, we study adaptive TPM in both oracle model and noise model. Specifically, in oracle model where the profit of any node set is accessible in $O(1)$ time, we propose ADG algorithm and prove that it could achieve an $\frac{1}{3}$ -approximation. In the noise model where any expected profit (or expected spread) only can be estimated, we extend ADG into ADDATP by considering *additive sampling error* and derive the corresponding approximation guarantee.

Third, we design a practical algorithm with optimized efficiency. To further optimize the efficiency of ADDATP in the noise model, we propose HATP algorithm by adopting the idea of *hybrid error*. Specifically, additive error could incur prohibitive computation overhead (see Section 5.3.1). To tackle this issue, we propose hybrid error which combines both relative error and additive error, fitting nicely to the scenario of nodes with diverse expected spreads. Based on this novel technique, we design HATP that boosts the efficiency of ADDATP significantly. Finally, its approximation guarantee and time complexity are derived.

As aforementioned, these three problems are closely related to some extent, while the algorithms designs for one of them cannot be adopted to solve another problem without violating the theoretical guarantees. Specifically, even though IM and SM both ask for the most influential nodes as seed set, the underlying challenge in IM is to address the *influence maximization* problem while that in SM is to solve the *truncated influence maximization* problem, which demands totally different sampling techniques to maintain the theoretical guarantees. TPM is defined based on IM. However, their objective functions

Figure 1.1: Thesis Overview.

are fundamentally distinct. Specifically, the spread function in IM is submodular and monotone, due to which greedy method is adopted. Contrarily, the profit function in TPM is submodular but non-monotone, therefore only double greedy is feasible. Due to this fact, methods adopted for IM and TPM are entirely different. More details are presented in the following corresponding chapters.

1.4 Thesis Organization

Figure 1.1 demonstrates the thesis overview. Chapter 2 demonstrates a detailed survey on the three problems we study. Before into the details, we first introduce the procedure of influence propagation in OSNs as the background of the three problems. As follows, we then summarize the existing work on these problems under the non-adaptive setting and adaptive setting respectively. Specifically, we first inspect the work on influence maximization and then analyze the existing deficiencies which motivates our study on this problem. Subsequently, we survey the work on seed minimization. We briefly introduce the work and then uncover some potential drawbacks that make the proposed algorithms either intractable or impractical. Finally, we investigate the work on the profit maximization problem. Notice that profit maximization is only considered under the non-adaptive setting in the literature, thus we only discuss related work on non-adaptive profit maximization in this part.

Chapter 3 presents our study on *adaptive influence maximization (IM)*. To address this problem, we first propose a general framework ADAPTGREEDY and prove that ADAPTGREEDY could provide both *worst-case* and *expected* approximation guarantees. To better suit the requirement of ADAPTGREEDY, we devise a novel non-adaptive IM algorithm EPIC which not only provides strong expected approximation guarantee, but also presents superior performance compared with the existing IM algorithms. Finally, we conduct experiments on real social networks to evaluate our proposed algorithms comprehensively. The experimental results strongly corroborate the superiorities and effectiveness of our approach.

Chapter 4 demonstrates our work on *adaptive seed minimization (SM)*. We first define the problem and introduce one core concept that serves as the foundation of our approach. Then we propose a general framework ASTI tailored for the problem. To instantiate the framework ASTI, we design TRIM and conduct the corresponding theoretical analysis. In what follows, we extend TRIM into a batched version TRIM-B that selects seed nodes in a predefined batch size in each round. We then conduct a comprehensive evaluation on our algorithms and the experimental results well prove the efficiency and efficacy of our approach.

Chapter 5 introduces our research on *adaptive target profit maximization (TPM)*. To explore adaptive TPM comprehensively, we first study this problem in both oracle model and noise model and propose ADG and ADDATP for these two models respectively. Meanwhile, we also analyze their corresponding approximation guarantees. To boost the efficiency, we optimize ADDATP into HATP by utilizing the idea of *hybrid error*. Afterwards, rigorous theoretical analyses are conducted on the approximation guarantee and time complexity of HATP. Last, we evaluate our proposed algorithms and the experimental results verify the significant efficiency improvement.

Chapter 6 concludes the thesis and also comes up with some possible inspirational future work on the topic of adaptive approximation algorithms for viral marketing in online social networks.

Chapter 2

Literature Review

In this chapter, we first introduce some fundamental concepts in influence propagation that are commonly used for the three studied problems. Next we will describe how seed node is selected in *non-adaptive* and *adaptive* settings respectively. Based on this knowledge, we then review the existing work on *influence maximization*, *seed minimization*, and *profit maximization* in the two settings. Considering that target profit maximization is newly proposed as the extension of profit maximization in our study, we thus focus on the related work of the profit maximization problem. Finally, we briefly summarize this chapter. Table 2.1 summarizes the notations that are frequently used.

2.1 Influence Propagation in Online Social Networks

2.1.1 Influence Spread and Realization

Let G be a social network with a node set V and a directed edge set E , where $|V| = n$ and $|E| = m$. For any edge $\langle u, v \rangle \in E$, we refer to u as an *incoming neighbor* of v , and v as an *outgoing neighbor* of u . Each edge $e = \langle u, v \rangle$ is associated with a *propagation probability* $p(e) \in (0, 1]$. We refer to such a social network as a *probabilistic social network*.

In the literature, *independent cascade (IC)* model [4] and *linear threshold (LT)* model [4] are the two of the most well-adopted models in influence propagation. Under the IC model, the propagation process initialized by a node set $S \subseteq V$ is illustrated as a discrete-time stochastic process as follows.

1. At time slot t_0 (the subscript indicates the index of the time slot), all nodes in S are *activated*, while all other nodes remain *inactive*.

Table 2.1: Frequently used notations

Notation	Description
$G = (V, E)$	a social network with node set V and edge set E
n, m	the numbers of nodes and edges in G , respectively
S	a subset of nodes in V , i.e., $S \subseteq V$
$G_i = (V_i, E_i)$	the i -th residual graph, where $G_1 = G$
n_i, m_i	the numbers of nodes and edges in G_i
$I_{G_i}(u_i)$	the number of nodes activated by node u_i on G_i
$\text{Cov}_{\mathcal{R}}(S)$	the number of RR-sets in $\mathcal{R}$ that overlap S
$\mathbb{E}[I(S)]$	the expected spread of seed set S
ϕ, Φ, Ω	a specific realization, a random realization, and the realization space

2. If a node u is first activated at time slot t_i , then it has a chance to activate its outgoing neighbors at time slot t_{i+1} , after which it can no longer activate any other node. In particular, if u has an inactive outgoing neighbor v , then u 's activation attempt on v succeeds with probability $p(e)$ at slot t_{i+1} , where $e = \langle u, v \rangle$ is the edge connecting u to v .
3. The influence propagation process terminates when none of the remaining inactive nodes can be activated.

As with the IC model, the LT model also assumes that each edge e in the social network G is associated with a propagation probability $p(e)$, but it requires that for each node v in G , the propagation probabilities of all edges ending to v sum up to no more than 1.

In addition, given a seed set S , it considers a different type of influence propagation process as follows:

1. At time slot t_0 , for each node v in G , we assign a threshold $\theta(v)$ that is sampled uniformly at random from $[0, 1]$. Then, we activate the nodes in S .
2. At time slot t_i , for each inactive node w , we examine all of its incoming edges from activated nodes and check if the sum of those edges' propagation probabilities is not smaller than $\theta(v)$. If it is, then we activate w ; otherwise, w remains inactive.

Figure 2.1: A social network and three of its possible realizations.

3. The influence propagation process terminates when none of the remaining inactive nodes can be activated.

Let $I(S)$ be the total number of active nodes in G when the influence propagation terminates. We refer to S as the *seed set*, and $I(S)$ as the *spread* of S .

Alternatively, the influence propagation process can also be described by the *live edge* procedure [4] which is specified by the underlying propagation model. For ease of exposition, we take the IC model as an example. For each edge $e \in E$, we independently flip a coin of head probability $p(e)$ to decide whether the edge e is *live* or *blocked* to generate a sample of influence propagation. All the blocked edges are removed and the remaining graph is referred to as a *realization*¹ of the probabilistic social network G , denoted as ϕ . Note that there are 2^m distinct possible realizations. Let Ω be the set of all possible realizations (i.e., the sample space) such that $|\Omega| = 2^m$, and $\Phi \sim \Omega$ denote that Φ is a realization randomly sampled from Ω . For example, Figure 2.1 shows a social network and three of its possible realizations. Given a realization $\phi \in \Omega$, the spread of any seed set $S \subseteq V$ under ϕ is the total number of nodes that are reachable from S , denoted as $I_\phi(S)$. Thus, for any seed set S , its *expected spread* $\mathbb{E}[I(S)]$ is defined as

$$\mathbb{E}[I(S)] := \mathbb{E}_{\Phi \sim \Omega}[I_\Phi(S)] = \sum_{\phi \in \Omega} I_\phi(S) \cdot p(\phi), \quad (2.1)$$

where $p(\phi)$ is the probability for realization ϕ to occur. In other words, the expected spread of S is the (weighted) average spread over all the realizations in Ω .

2.1.2 The State of the Art on Spread Estimation

In viral marketing, node sets are evaluated by their expected spreads under one propagation model, which, however, is #P-hard to calculate [29]. Normally, the expected spread

¹It is also called *possible world* in the literature, thus we use these two terms interchangeably.

 Figure 2.2: Input graph G and its corresponding random graphs.

can only be estimated by sampling. The current state of the art on spread estimation is the *reverse influence sampling (RIS)* technique introduced by Borges et al. [13]. RIS technique is widely adopted in the literature, and it is based on the concept of *random reverse reachable (RR) set* [10], which we explain in the following: Let g be a directed graph obtained by removing each edge e in G with probability $1 - p(e)$ independently, and $\mathcal{G}$ be the distribution of g induced by the randomness in edge removal. Given an instance of g and a node v , the RR set R for v in g is the set of nodes in g that can reach v . R is a *random RR set*, if v is selected uniformly at random from G . We illustrate the above concepts with an example.

Example 2.1. Figure 2.2(a) shows a social network with three nodes and three edges. (The number besides each edge e indicates its propagation probability $p(e)$.) If we independently remove each edge e in G with $1 - p(e)$ probability, then the resulting graph has four possibilities, which we illustrate as g_1, g_2, g_3, g_4 in Figures 2.2(b)-2.2(e). The RR set for node v_1 in g_1 is $\{v_1, v_2\}$, since v_1 and v_2 are the only nodes that can reach v_1 in g_1 . Meanwhile, the RR set for v_1 in any of g_2, g_3 , and g_4 is $\{v_1, v_2, v_3\}$. $\square$

Observe that if a node u appears in an RR set for another node v , then G must contain a directed path from u to v , which indicates that u has a probability to influence v in a propagation process. Based on this observation, Borges et al. prove that random RR sets can be used to estimate the expected spread of any seed set, as shown in the following lemma.

Lemma 2.1. ([13]) *Let S be a seed set, g be a sample from G , and R be the RR set for a node v in g . Let ρ_1 be the probability that S can activate v in an influence propagation process, and ρ_2 be the probability that R overlaps with S . Then, $\rho_1 = \rho_2$.*

Let $\mathcal{R}$ be a set of random RR sets, and $Cov_{\mathcal{R}}(S)$ be the number of RR sets in $\mathcal{R}$ that overlap with S . By Lemma 2.1, $\frac{n}{|\mathcal{R}|}Cov_{\mathcal{R}}(S)$ can be used as an unbiased estimator of the expected spread of S :

Corollary 2.1. ([13]) $\mathbb{E} \left[\frac{n}{|\mathcal{R}|}Cov_{\mathcal{R}}(S) \right] = \mathbb{E}[I(S)]$.

2.1.3 Advantage of Adaptivity over Non-adaptivity on Seed Selection

The fundamental challenge to solve problems in viral marketing is to how to select seed nodes efficiently and effectively among millions or billions of users in social networks. There are an abundance of algorithms [4, 11, 12, 17, 20, 21, 30] proposed to identify seed nodes. In the literature, seed selection is mostly studied in *non-adaptive* setting but barely studied in *adaptive* setting. However, we can gain more benefits in adaptive setting than that in the non-adaptive setting. To explain, in the non-adaptive setting, seed nodes are selected in one batch before observing how they influence the other users in real world. Contrarily, in the adaptive setting, seed nodes are selected in multiple batches, such that the later batch is identified after the actual influence results of all the former batches are observed. By exploiting the knowledge on the past observation, seed nodes are selected more smartly such that the influence overlap among seed users is avoided effectively. We design the following example as an illustration.

Example 2.2. Consider the social network in Figure 2.1(a), and suppose that the realization sampled from Ω is $\phi = \phi_1$, as shown in Figure 2.1(b). Assume that we choose v_1 as the first seed node. In that case, we can observe v_1 's influence on v_2 and v_4 , since v_1 has two outgoing edges (v_1, v_2) and (v_1, v_4) in ϕ_1 . Similarly, we can observe v_4 's influence on v_5 . In addition, we can also observe that v_1 (resp. v_4) cannot influence v_3 (resp. v_6), as ϕ_1 does not contain an edge from v_1 to v_3 (resp. v_4 to v_6). Figure 2.3(a) shows the results of the influence propagation from v_1 , with each double-line (dashed-line) arrow denoting a successful (resp. failed) step of influence.

In general, after choosing a partial set S' of seed nodes, we can learn all nodes that S' can reach in ϕ_1 , as well as the out-edges of those nodes in ϕ_1 . This enables us to optimize the choices of the remaining seed nodes since we can focus on the nodes that

 Figure 2.3: Adaptive vs. non-adaptive seed selection with $k = 2$.

have not been influenced by S' . For instance, consider that selecting another seed node based on the result in Figure 2.3(a). In that case, we can omit the nodes that have been influenced (i.e., v_1 , v_2 , v_4 , and v_5), and focus on the subgraph induced by the remaining nodes, as shown in Figure 2.3(b). Based on this, we can choose v_3 as the second seed node, which yields the results in Figure 2.3(c), where we have 6 nodes influenced in total. In contrast, if we are to non-adaptively choose two seed nodes from the social network in Figure 2.1(a), we may end up choosing v_1 and v_4 , in which case we would obtain the result in Figure 2.3(d) when the underlying realization is ϕ_1 in Figure 2.1(b). In other words, we can only influence 4 nodes instead of 6. $\square$

2.2 Influence Maximization

Let G be a social network with a node set V and a directed edge set E , with $|V| = n$ and $|E| = m$. Given G and an integer $k > 0$, the *influence maximization (IM)* problem asks for a seed set S_k with the largest expected spread, i.e.,

$$S_k = \arg \max_{S': |S'|=k} \mathbb{E}[I(S')].$$

As mentioned in Section 1.2, the influence maximization (IM) problem is initially formulated in the seminal work by Kempe et al. [4]. This work has proven that i) it is NP-hard to derive the optimal solution for IM, ii) the spread function is monotone and submodular under both the independent cascade (IC) model and linear threshold (LT) model, iii) greedy algorithm (Algorithm 1) could achieve an $(1 - 1/e)$ -approximation guarantee by utilizing the monotone and submodular properties of the spread function. These three findings lay the foundation for the subsequent study on the IM problem. The

Algorithm 1: Greedy

```
1 Initialize  $S_k = \emptyset$ 
2 for  $i = 1$  to  $k$  do
3    $v = \arg \max_{u \in V \setminus S} \{\mathbb{E}[I(S \cup \{u\})] - \mathbb{E}[I(S)]\}$ 
4   Insert  $v$  into  $S_k$ 
5 return  $S_k$ 
```

core challenge to address the IM problem lies in how to identify seed nodes efficiently and effectively among millions or billions of users in OSNs because it is #P-hard [20, 21] to calculate the exact spread of any node sets. Since then, IM has attracted plenty of attention from academia for the past decades. Based on the seed selection procedure in the literature, we classify the related work into two parts, i.e., one part on *non-adaptive influence maximization* and the other part on *adaptive influence maximization* as follows.

2.2.1 Non-adaptive Influence Maximization

In non-adaptive setting, to identify the influential individuals efficiently, numerous sampling techniques have been developed to estimate the expected spreads of node sets. Based on the approaches to spread estimations, existing algorithms could be classified into three categories, i.e., *simulation based* approach, *proxy based* approach and *sketch based* approach, following the categories in the survey [31].

Simulation-based approach. Kempe et al. proves that greedy algorithm could achieve $(1 - 1/e - \varepsilon)$ -approximation guarantee with at least probability of $1 - 1/n^l$ by conducting sufficient number of MC simulations on spread estimation, where $\varepsilon \in (0, 1)$ is the *relative error* on estimation and l is constant. For the sake of clarity, this greedy algorithm based on MC simulation is referred to as NAIVEGREEDY. As pointed out in [10, 32], it requires the total number of $O(k^2 l n \varepsilon^{-2} \log n / OPT)$ MC simulations for each spread estimation, where OPT is the expected spread of the optimal seed set. Moreover, notice that i) there are k iterations in total, ii) $O(n)$ MC simulations are required for each iteration, which results in the total number of $O(k^3 l n^2 \varepsilon^{-2} \log n / OPT)$ MC simulations. For each simulation, it takes $O(m)$ time in expectation, resulting in the time complexity of $O(k^3 l m n^2 \varepsilon^{-2} \log n / OPT)$ [10]. Such prohibitive time complexity motivates the work

done by Leskovec et al. [30] who propose a Cost-Effective Lazy Forward (CELF) algorithm. CELF is shown to improve the efficiency for over 700 times compared with NAIVEGREEDY. The rationale behind is that CELF avoids a large number of unnecessary MC simulations by exploiting the submodularity property of spread function. Specifically, the social networks are largely power-law graphs and the influence of most nodes are rather small. In this regard, if the estimated spread of one node is small in the previous iterations, it is quite unlikely that this node will be selected as the seed node in current iteration. In such case, MC simulations for spread estimation on such kinds of nodes can be avoided without affecting the correctness. Later, Goyal et al. further optimize CELF into CELF++ and claim to achieve around 35% ~ 55% performance improvement. However, as reported by [33], this improvement is due to one unexpected flaw of experiment setup and the improvement is not statistically significant. Even with such considerable performance improvement, as shown in [10], CELF is still of poor efficiency on large graphs.

In short, even though those optimizations are applied, simulation-based approaches are prone to incur expensive computation overhead and not scalable well on large graph.

Proxy-based approach. To bypass the high computation cost of the simulation, there are a plenty of algorithms proposed utilizing proxy approaches instead. Specifically, proxy approaches aim to accelerate the diffusion process on corresponding diffusion model by taking advantage its characteristics. In dosing so, most of the proxy approaches estimate influence spread heuristically and could not provide any theoretical guarantees.

Maximum influence arborescence (MIA) [20] model is one of the simplified versions of IC model with better scalability. MIA model estimates the spread of each node using a local arborescence structure which is a local subgraph claimed to capture the spread of each node efficiently. The arborescence structure is constructed as follows. First, given a threshold $\theta \in (0, 1)$, it computes the *maximum influence paths* (MIP) for every pair of nodes in the graph, but only stores the MIPs with probability larger than θ . Second, the MIPs ending and starting at each node u are combined into a *maximum influence in-arborescence* and *maximum influence out-arborescence* respectively, referred to as $\text{MIIA}(u, \theta)$ and $\text{MIOA}(u, \theta)$. Each such arborescence structure is a local influence

region from which the spread can be estimated more efficiently. Similarly, LDAG algorithm in [21] adopts a *directed acyclic graphs* (DAG) structure for spread estimation under the LT model. Specifically, $\text{LDAG}(u, \theta)$ also constructs such an influence region that consists of nodes that could influence node v with probability larger than θ . However, there are two major issues existing in MIA and LDAG as i) it needs to construct the arborescence structure (resp. LDAG) for each node, which is computation expensive, and ii) all those structures are stored, which is memory intensive. Motivated by these facts, SIMPATH [34] computes the spread by enumerating simple paths under the LT model. To explain, given a node $u \in V$ and a threshold $\theta \in (0, 1)$, SIMPATH enumerates all possible simple paths starting from u , pruning the paths with probability less than θ . Then the spread of u is estimated by summing up the probabilities of the simple paths.

In a nutshell, proxy-based approaches could avoid the high computation overheads efficiently in some degree. However, one of the critical issues is that it is hard to devise an optimal proxy. As a result, the estimated spreads are likely to be heavily biased, which might seriously damage the quality of the selected seed set.

Sketch-based approaches. Recall that the enormous computation cost of simulation-based approaches lies in i) the large number of MC simulations, ii) the high cost ($O(m)$) for each simulation. To relieve these two issues, the sketch-based approaches randomly generate certain number of *sketches* based on the specific diffusion model. The advantages of sketch-based approaches mainly lie in two aspects as follows. First, sketches can be reused among $O(n)$ influence estimations. Second, spread estimation through sketch is unbiased and thus sketch-based approaches could provide theoretical guarantee. The core challenge is to design efficient sketches with much lower cost than $O(m)$.

Among them, NEWGREEDYIC [29] can be seen to utilize the sketch of *realization* which is at the cost of $O(m)$, as introduced in Chapter 2.1. Notice that constructing one realization sample is at the same cost of running one MC simulation, but NEWGREEDYIC still boosts the performance of NAIVEGREEDY significantly due to the sketch reused for remaining spread estimations instead of rerunning from scratch as NAIVEGREEDY does. A recent significant breakthrough on sketch-based approaches on the IM problem is proposed by Borgs et al. in [13]. Borgs et al. introduce a novel method, referred as *Reverse Influence Sampling (RIS)* in [10].

As introduced in Section 2.1.2, RIS technique is built on the sample of *random reverse reachable (RR) set* [10]. Based on the Corollary 2.1, Borgs et al. [13] propose an elegant two-step method for influence maximization:

1. Generate a sufficiently large set $\mathcal{R}$ of random RR sets.
2. Use the greedy algorithm for maximum coverage [35] to identify a size- k seed set S_k that overlaps with a large number of RR sets in $\mathcal{R}$.

The intuition of the above method is that, if a seed set S intersects a large number of RR set in $\mathcal{R}$, then, by Corollary 2.1, the expected spread of S is likely to be large, in which case S could be a good solution for the influence maximization problem. Nevertheless, if we are to ensure the worst-case quality of S , then we have to carefully decide the size of $\mathcal{R}$. Towards this end, Borgs et al. propose to keep generating random RR sets in $\mathcal{R}$, until the total number of edges inspected during the generation process exceeds a predefined threshold τ . They show that when τ is properly set, there is at least $\frac{3}{5}$ probability that the method returns a seed set S with a $(1 - 1/e - \epsilon)$ -approximation ratio, i.e.,

$$\mathbb{E}[I(S)] \geq (1 - 1/e - \epsilon)OPT,$$

where OPT denotes the maximum expected spread of any size- k seed set in G . In addition, Borgs et al. show that the method's success probability can be increased to $1 - 1/n$ at the cost of degrading its expected time complexity to $O(k(m+n)\epsilon^{-3} \log^2 n)$.

Following Borgs et al.'s work, Tang et al. [10] propose an improved solution, TIM^+ , which runs in $O(k(m+n)\epsilon^{-2} \log n)$ expected time and returns a $(1 - 1/e - \epsilon)$ -approximation with at least $1 - 1/n$ probability. Their idea is to avoid imposing a threshold on the number of edges inspected during the construction of $\mathcal{R}$, but to apply the Chernoff bounds to more accurately decide whether the number of random RR sets in $\mathcal{R}$ is sufficiently large. The experiments in [10] show that TIM^+ is several order of magnitude faster than Borgs et al.'s algorithm, and it is able to process a billion-edge social network in reasonable time. Subsequently, Tang et al. [11] present a further enhanced algorithm, IMM, which offers the same theoretical guarantee as TIM^+ does but provides much higher empirical efficiency. The improvement of IMM is mainly due to a martingale-based technique for deciding the size of $\mathcal{R}$ [11], which is shown to be less pessimistic than the method used in TIM^+ .

Very recently, Nguyen et al. [12] observe that the computation time of IMM still leaves room for improvement, as it tends to generate an excessive number of RR sets when k is large. To address this issue, they propose two new algorithms, the *Stop-and-Stare algorithm* (SSA) and the *Dynamic Stop-and-Stare algorithm* (D-SSA), both of which are similar in spirit to IMM but gauge the number of RR sets required by adopting the *trial-and-error*, i.e., stop and stare strategy. Specifically, SSA & D-SSA set a serial of stopping time T . At each of stopping time $t \in T$, they stop sampling to check if the current solution S satisfying the $(1 - 1/e - \varepsilon)$ -approximation. If so, SSA & D-SSA terminate and return S immediately; otherwise they continue to generate random RR sets until next stopping time. As pointed out by Huang et al. [28], there exist serious gaps in their theoretical analyses in [12] which invalidate the approximation guarantees of SSA & D-SSA. Later on, Nguyen et al. rectify the issues in [36]. Inspired by Nguyen et al.'s work, Tang et al. [17] propose OPIM-C based on the trial-and-error strategy. The main idea is as follows. At each stopping time with current solution S , it directly derives the lower bound of $\mathbb{E}[I(S)]$ as $\mathbb{E}^l[I(S)]$ and the upper bound of OPT as OPT^u online based on Chernoff bounds. It then compares the ratio of $\mathbb{E}^l[I(S)]/OPT^u$ with $(1 - 1/e - \varepsilon)$ to see if current solution S is qualified enough. Additionally, OPT^u is optimized further by exploiting the submodularity of spread function, which makes it approach to the real OPT . In doing so, OPIM-C stops at one stopping time earlier than SSA & D-SSA do, thus achieving much higher efficiency.

2.2.2 Adaptive Influence Maximization

Compared with non-adaptive influence maximization, the amount of work to address adaptive influence maximization is much less in the literature. The first solution to adaptive IM is by Golovin et al. [9]. In [9], it assumes that $b = 1$ (i.e., each batch consists of only one seed node), and adopts a greedy approach as follows. Given G , it first identifies the node v_1 whose expected spread $\mathbb{E}[I_G(\{v_1\})]$ on G is the largest, and selects it as the first seed. Then, it observes the nodes that are influenced by v_1 (which are in accordance to the realization ϕ_0), and removes them from G . Let G_2 denote the subgraph of G induced by the remaining nodes. After that, for the i -th ($i > 1$) batch, it (i) selects the node v_i with the maximum expected spread $\mathbb{E}[I_{G_i}(\{v_i\})]$ on G_i , (ii) observes

the influence of v_i on G_i , and then (iii) generates a new graph G_{i+1} by removing from G_i those nodes that are influenced by v_i . For convenience, we refer to G_i as the i -th *residual graph*, and let $G_1 = G$.

Let $\text{OPT}_{k,b}$ denote the expected spread of the optimal solution to the adaptive IM problem parameterized with k and b . Golovin et al. [9] show that the above greedy approach returns a solution whose expected spread is at least $(1 - 1/e) \cdot \text{OPT}_{k,1}$. This approximation guarantee, however, cannot be achieved in polynomial time because (i) in the i -th batch, it requires identifying a node v_i with the maximum largest expected spread $\mathbb{E}[I_{G_i}(\{v_i\})]$ on G_i , but (ii) computing the exact expected spread of a node is $\#P$ -hard in general [20].

To remedy the above deficiency, Vaswani and Lakshmanan [22] propose a relaxed approach that allows errors in the estimation of expected spreads. In particular, they assume that for any node set S and any residual graph G_i , we can derive an estimation $\tilde{\mathbb{E}}[I_{G_i}(S)]$ of $\mathbb{E}[I_{G_i}(S)]$, such that

$$\alpha^\perp \cdot \mathbb{E}[I_{G_i}(S)] \leq \tilde{\mathbb{E}}[I_{G_i}(S)] \leq \alpha^\top \cdot \mathbb{E}[I_{G_i}(S)], \quad (2.2)$$

with $\alpha^\top/\alpha^\perp$ bounded from above by a parameter η . They show that, by feeding such estimated expected spreads to the greedy approach in [9], it can achieve an approximation guarantee of $1 - e^{-1/\eta}$. In addition, they show that the greedy approach can be extended to the case when $b > 1$, with one simple change: in the i -th batch, instead of selecting only one node, we select a size- b seed set S_i whose estimated expected spread on G_i is at least $1 - 1/e$ fraction of the largest estimated expected spread on G_i . In that case, they show that the resulting approximation guarantee is $1 - e^{-(1-1/e)^2/\eta}$.

Unfortunately, the accuracy requirement in Equation (2.2) is still impractical as no existing algorithm for evaluating expected spread can meet the requirement. Indeed, as computing $\mathbb{E}[I_{G_i}(S)]$ is $\#P$ hard, the existing algorithms can only derive $\tilde{\mathbb{E}}[I_{G_i}(S)]$ in a probabilistic manner, which implies that both $\alpha^\perp$ and $\alpha^\top$ are random numbers depending on G_i . As G_i is also random, it is hard to derive a meaningful fixed upper bound η for $\alpha^\top/\alpha^\perp$. Therefore, we think that the approximation ratio proposed in [22] only has theoretical value and cannot be implemented in practice.

Recently one work [37] studying *online influence maximization* by S. Lei et al. is similar to our study on adaptive influence maximization in some extent. In particular,

both this work and our work study the influence maximization in multiple batches of seed selection, and collect the influence feedbacks to improve the following seed selection for each batch. However, their work differs from us in the following three aspects. First, S. Lei et al. focus on the estimation of influence probability of each edge, while we assume that those influence probabilities are given as input and focus on seed selection. Second, during the graph update phase, S. Lei et al. update the influence probability without altering the structure of the graph, whilst we update the graph by removing the active nodes and associated edges. Third, algorithm proposed by S. Lei et al. is heuristic and could not provide theoretical guarantees. Our algorithm provides theoretical approximations with rigorous mathematical proofs.

2.3 Seed Minimization

Seed minimization is the dual problem of influence maximization, defined as follows. Let $G = (V, E)$ be a probabilistic social network with $n = |V|$ and $m = |E|$, given a threshold $\eta \in [1, n]$, the *seed minimization* (SM) problem seeks to identify a seed set S^* with minimum number of seed nodes to influence at least η nodes in expectation, i.e.,

$$S^* = \arg \min_{S \subseteq V: \mathbb{E}[I(S)] \geq \eta} |S|.$$

Seed minimization is motivated in viral marketing of the requirement to raise the awareness for all or a large fraction of individuals in social networks with the minimum cost [38]. Since then, the SM problem has been studied in both non-adaptive and adaptive settings, summarized as follows.

2.3.1 Non-adaptive Seed Minimization

To the best of our knowledge, seed minimization (SM) is initially proposed by Chen [38]. In [38], the SM problem is assumed under the fixed threshold model, where each node in the networks is assigned a fixed threshold such that one node is activated only when the number of its active *neighbors* exceeds this fixed threshold. In this setting, the influence propagation process is deterministic and the spread functions are non-submodular. Then Chen proves that this problem cannot be approximated within a ratio of $O(2^{\log^{1-\epsilon} n})$ for

any fixed constant $\varepsilon > 0$, unless $\text{NP} \subseteq \text{DTIME}(n^{\text{polylog}(n)})$. Afterwards, Goldberg et al. [39] study this problem under a variant of fixed threshold model. In this model, it counts not only the active neighbors of u , but also the active nodes that can reach u by a path consisting of active nodes. Then node u is activated when the total number of reachable active nodes is no smaller than the threshold of u . Under this model, they propose an approximation algorithms by utilizing the linear programming technique. It is worth noting that spread function is non-submodular under these fixed-threshold models which are different from the linear threshold model [4] commonly studied in viral marketing.

Concurrently, Long and Wong [40] extend seed minimization into the widely adopted independent cascade (IC) and linear threshold (LT) models, as introduced in Section 2.1. Under these two models, it is $\#$ -P hard [20] to calculate the expected spread of any node set. Thus, Long et al. estimate expected spreads by generating sufficient number of random realizations. Based on this way, they propose an approximation algorithm with expected time complexity of $O(\frac{mn^2 \log(1/\delta)}{\varepsilon^2})$ where $\delta, \varepsilon \in (0, 1)$ are user-specified parameters. This complexity is still too high to tolerate, even for small social graphs with only thousands of nodes. Under the same models, Goyal et al. [23] provide a bi-criteria approximation algorithms for the SM problem. Specifically, instead of reaching the required threshold of η , they assume a shortfall gap $c \in (0, \eta)$ to strike a balance between efficiency and efficacy. Then they prove that a seed set selected by greedy algorithm could approach the optimal solution within approximation of $O(1 + \ln(\eta/c))$. To fill such shortfall gap of c and reach the goal of η , Zhang et al. [5] propose a γ -multiplicative error which requires that for any node set S , the distance between the spread estimation $I_{est}(S)$ and its expected spread $\mathbb{E}[I(S)]$ is smaller than $\gamma \mathbb{E}[I(S)]$, i.e., $|I_{est}(S) - \mathbb{E}[I(S)]| \leq \gamma \mathbb{E}[I(S)]$. However, such requirement is too stringent that makes the proposed algorithms vastly inefficient in reality. To explain, for node set S with relatively small expected spread, the spread estimation is supposed to be extremely precise to meet the multiplicative error requirement, which could incur prohibitive computation overhead. To relieve this efficiency pain, Han et al. [24] propose ATEUC algorithm by utilizing RR sets (Section 2.2) for spread estimation. Specifically, ATEUC uses a *trial-and-error* strategy on seed selection as follows. It first selects two seed set candidates S_u and S_l

whose set sizes are taken as the upper bound and lower bound on size the optimal set respectively. As a consequence, only when the condition $|S_u| \leq 2|S_l|$ is satisfied, the candidate set S_u is returned as the solution; otherwise ATEUC will continue to refine the qualities of S_u and S_l to shrink their sizes. One potential issue lies in ATEUC is that the expected time complexity is unknown. Meanwhile, its worst-case time complexity is prohibitively large.

2.3.2 Adaptive Seed Minimization

Golovin and Krause [41] study the *adaptive stochastic minimum cost coverage* problem, which can be regarded as a variant of adaptive SM by assuming an available *oracle* that could accurately report the expected marginal truncated spread for any given seed set. Truncated spread is the influence spread with the part larger than the input η truncated (More details are discussed in Section 4.1.2). Specifically, they propose to adopt a greed policy as follows. First, the policy selects the node s_1 with the largest expected truncated spread, i.e., $\Delta(s_1 | S_0) \geq \Delta(v | S_0)$ for all $v \in V$. Then, it observes the actual nodes that are activated by s_1 during the stochastic process, and removes them from G to induce the residual graph G_2 . After that, identify the node s_2 with the maximum expected marginal truncated spread $\Delta(s_2 | S_1)$ in the residual graph G_2 . This process continues, such that each round selects the node with the largest expected marginal truncated spread, until we observe that no less than η nodes have been influenced.

Golovin and Krause [41] show that the above greedy policy returns a $(\ln \eta + 1)^2$ -approximate solution to the optimum.² This approximation guarantee, however, does not lead to a practical algorithm for the adaptive SM problem, because (i) it requires the help from an oracle to exactly identify the node with the maximum expected marginal truncated spread in each round, but (ii) computing the exact expected spread of any node set is #P-hard [20].

Motivated by this observation, Vaswani and Lakshmanan [22] attempt to extend Golovin and Krause's method by replacing the oracle with an spread estimator with

²Golovin and Krause claim that the approximation guarantee is $(\ln \eta + 1)$ in an earlier version of their work [9], but point out that the proof has gaps in a revised version [41]. Whether the logarithmic bound holds is still an interesting open problem.

bounded errors. In particular, they assume that for any node set S , the estimation $\mathbb{E}[\tilde{I}(v \mid S)]$ of the marginal gain $\mathbb{E}[I(v \mid S)] := \mathbb{E}[I(S \cup \{v\})] - \mathbb{E}[I(S)]$ should satisfy

$$\alpha^\perp \mathbb{E}[I(v \mid S)] \leq \mathbb{E}[\tilde{I}(v \mid S)] \leq \alpha^\top \mathbb{E}[I(v \mid S)], \quad (2.3)$$

where $\alpha^\top / \alpha^\perp$ denotes the multiplicative error in calculating the marginal gains. Unfortunately, this requirement on the spread estimation is so stringent that no existing methods for influence estimation could fulfill the requirement without incurring prohibitive estimation overhead. To explain, suppose that the expected marginal spread $\mathbb{E}[I(v \mid S)]$ of a node v on top of S is small. In that case, Equation (2.3) would only allow a trivial amount of estimation error, which is rather difficult to achieve by existing methods for spread estimation, as explained aforementioned.

2.4 Profit Maximization

Profit maximization can be taken as one variant of influence maximization problem by incorporating the monetary factor, defined as follows. Given a probabilistic social graph $G = (V, E)$, the *profit maximization (PM)* problem aims to identify a set of seed users T^* with each user $u \in T^*$ associated with a cost $c(u)$ on seed selection to maximize the expected profit, i.e.,

$$T^* = \arg \max_{T \subseteq V} \mathbb{E}[I(T)] - \sum_{u \in T} c(u).$$

The cost of each user on seed selection can be taken as the investment cost (e.g., cashback rewards, coupons, or discounts) as an initial product adopter on advertising. Let $\rho(T)$ be the expected profit of T , i.e., $\rho(T) = \mathbb{E}[I(T)] - c(T)$ where $c(T) = \sum_{u \in T} c(u)$. As a variant of IM, one may consider the simple greedy method for IM can also be applied for the PM problem to achieve some approximation guarantees, which, actually, does not hold. As pointed out in Section 2.2, spread function in the IM problem is submodular and monotone [4]. However, under the IC and LT model, profit function $\rho(\cdot)$ is submodular but not necessarily monotone, which makes the profit maximization problem fall into the category of *unconstrained submodular maximization (USM)* [42, 43]. For the USM problem, as pointed out by Feige et al. [42], there is no efficient algorithm that could provide approximation guarantee without additional assumption. Considering

Algorithm 2: Deterministic Double Greedy [43]

```

1 Initialize  $S \leftarrow \emptyset, T \leftarrow V$ ;
2 foreach node  $u \in V$  do
3    $z^+ \leftarrow f(u \mid S)$ ;
4    $z^- \leftarrow -f(u \mid T \setminus \{u\})$ ;
5   if  $z^+ \geq z^-$  then  $S \leftarrow S \cup \{u\}$ ;
6   else  $T \leftarrow T \setminus \{u\}$ ;
7 return  $S (= T)$ ;
```

that i) PM is an application of unconstrained submodular maximization, and ii) there is no work to consider PM in adaptive setting, in what follows, we survey the related work on the USM problem and the non-adaptive profit maximization problem.

Unconstrained Submodular Maximization. For a submodular function $f : 2^V \rightarrow R$, the submodular maximization problem is to find a set $T \subseteq V$ that maximizes $f(T)$, i.e.,

$$T = \arg \max_{S \subseteq V} f(S).$$

Submodular maximization has attracted considerable interest [8, 42–45] in the past decades. A plethora of problems in real world are the applications of submodular maximization, such as maximum facility location [46], Max-Cut [47] and influence maximization (IM) [4]. For submodular maximization, approximation algorithms with theoretical guarantee are all based on additional assumptions [42, 48]. If the submodular function is nonnegative, Feige et al. [42] prove that a uniformly random set could achieve a $\frac{1}{4}$ -approximation. Furthermore, this approximation ratio could be improved to $\frac{1}{2}$ if the submodular function is symmetric. Meanwhile, they also prove that it is NP-hard to derive a $(\frac{1}{2} + \varepsilon)$ -approximation for symmetric submodular maximization for any fixed $\varepsilon > 0$. Based on Feige et al.’s work, Buchbinder et al. [43] devise a deterministic and randomized *double greedy* methods to address the submodular maximization problem. Algorithm 2 and Algorithm 3 present the pseudo-code of deterministic and randomized double greedy respectively.

Deterministic double greedy works as follows. It first maintains two sets S and T which are initialized with $\emptyset$ and the whole node candidate set V respectively. In what follows, it then checks each node $u \in V$ by calculating $f(u \mid S)$ and $f(u \mid T \setminus \{u\})$

Algorithm 3: Randomized Double Greedy [43]

```
1 Initialize  $S = \emptyset, T = V$ ;  
2 foreach node  $u \in V$  do  
3    $z^+ \leftarrow f(u \mid S)$ ;  
4    $z^- \leftarrow f(u \mid T \setminus \{u\})$ ;  
5    $z^+ \leftarrow \max\{z^+, 0\}, z^- \leftarrow \max\{z^-, 0\}$ ;  
6   With probability  $z^+/(z^+ + z^-)$  do:  $S = S \cup \{u\}$ ;  
7   Otherwise do:  $T = T \setminus \{u\}$ ;  
8 return  $S (= T)$ ;
```

one by one, denoting as z^+ and z^- respectively. If $z^+ \geq z^-$, node u is inserted into S . Otherwise, u is removed from T . When all nodes in V have been checked, S is equal to T and either of them is returned. Randomize double greedy works in a similar manner but keeps each node u with a probability determined by z^+ and z^- (Algorithm 3 Line 6). In what follows, Buchbinder et al. prove that deterministic (resp. randomized) double greedy could achieve $\frac{1}{3}$ (resp. $\frac{1}{2}$)-approximation under the assumption that submodular function on the ground set is nonnegative, i.e., $f(V) \geq 0$. Notice that it is NP-hard to achieve $(\frac{1}{2} + \varepsilon)$ -approximation ratio for any $\varepsilon > 0$. To the best of our knowledge, double greedy method is the state of the art on the submodular maximization problem.

Non-adaptive Profit Maximization. As aforementioned, the profit maximization (PM) problem is one instance of unconstrained submodular maximization and has been a hot topic in academia. Arthur et al. [6] initially incorporate the price factor into spread propagation by extending the independent cascade model and linear threshold model, and then propose the profit maximization in the viral marketing. To start the advertising campaign, a target set of people are encouraged by certain investment cost (e.g. cashback rewards, coupons, and discounts) as the initial adopters. The profit maximization problem is to seek for such a set of target users T and the corresponding investment costs for each user $u \in T$, so as to maximize the final expected profit. To identify such a set of target users, Arthur et al. propose to construct a *maximum-leaf spanning tree* from the social graph and take all the interior nodes (non-leaf nodes) as the set of target users T , and the total investment cost $c(T)$ is set to a fixed fraction of the expected spread of $\mathbb{E}[I(T)]$ so as to maximize the expected profit $\rho(T) = \mathbb{E}[I(T)] - c(T)$. Later, Lu et al. [7] propose PAGE algorithm based on simple greedy. First, PAGE

evaluates the marginal influence of each node and sets its cost accordingly conditioned on the selected seed set and determined prices. Then it will select the node that provides the largest marginal expected profit. This process is repeated until the current identified node provides negative profit. However, both of the two methods above are based on empirical analysis and provide no theoretical guarantees.

On the other hand, Tong et al. [49] consider coupon discounts for seed users and propose RA-T and RA-S algorithms by utilizing the randomized double greedy in [43]. They claim that the algorithms could achieve $\frac{1}{2}$ -approximation with high probability. Similarly, the theoretical guarantee is based on the assumption that giving coupons to each user in the social network could still bring benefit, which is impractical in reality. Liu et al. [50] also consider the coupon in profit maximization and define a new diffusion model named *independent cascade model with coupons and valuations*. To address this problem, they propose the PMCA algorithm based on the local search algorithm in [42]. PMCA is claimed to achieve a $\frac{1}{3}$ -approximation based on the assumption that the submodular function is nonnegative for any subset S with high probability. However, their weaknesses are as follows. First, the assumption that taking any node set in the social graph could bring profits is too strong in practice. Second, the time complexity of PMCA is $O(\log(n)mn^4/\varepsilon^3)$ which is too prohibitive to apply. Similarly, Tang et al. [26] utilize both the deterministic and randomized double greedy algorithm to address the profit maximization problem. With the assumption that submodular function on the ground set is nonnegative, they prove the $\frac{1}{3}$ - and $\frac{1}{2}$ -approximation guarantees respectively. Furthermore, to relax this strong assumption, they design a novel method and extend this assumption to a much weaker one. However, they do not consider the sampling errors in spread estimation, which makes the proposed algorithms heuristic without any theoretical guarantees.

2.5 Summary

In this chapter, we first introduce the influence propagation process and then describe the seed selection procedure in non-adaptive and adaptive settings respectively. Afterwards, we provide an overview of some related work on influence maximization, seed minimization, and profit maximization. Specifically, we briefly summarize the recent algorithms

to address the IM and SM problems under both non-adaptive and adaptive settings and the PM problem in non-adaptive setting, and also discuss the existing issues.

Chapter 3

Adaptive Influence Maximization

In this chapter, we study the *adaptive influence maximization (IM)* problem where the k seed nodes are selected in batches of equal size b , such that the i -th batch is identified after the actual influence results of the former $i - 1$ batches are observed. To this end, we propose the first practical algorithm for adaptive IM that could provide both the *expected* and *worst-case* approximation guarantee of $1 - e^{\alpha(\varepsilon-1)}$, where $\alpha = 1 - (1 - 1/b)^b$ and $\varepsilon \in (0, 1)$ is a user-specified parameter. Our approach is based on a novel ADAPTGREEDY framework. Specifically, ADAPTGREEDY is a general framework that could be instantiated by any existing non-adaptive IM algorithms with expected approximation guarantee. In addition, we propose a novel non-adaptive IM algorithm called EPIC which not only provides strong expected approximation guarantee, but also presents superior performance compared with the existing IM algorithms. Meanwhile, we clarify some existing misunderstandings in recent work and shed light on further study of the adaptive IM problem. We conduct experiments on real social networks to evaluate our proposed algorithms comprehensively, and the experimental results strongly corroborate the superiorities and effectiveness of our approach.

The remainder of the chapter is organized as follows. First we describe the problem in Section 3.1. Afterwards, we propose a general framework ADAPTGREEDY based on which we could derive algorithms with strong approximation for the adaptive IM problem in Section 3.2. To the benefit of ADAPTGREEDY, we develop a new non-adaptive IM method, EPIC, that provides a desired expected approximation ratio. By instantiating ADAPTGREEDY with EPIC, we then derive the corresponding *expected* and *worst-case*

approximation guarantees achieved in Section 3.3. Last, we evaluate our proposed algorithm in Section 3.5 and summarize the work in Section 3.6.

3.1 Preliminary

This section briefly introduces the background of the adaptive IM problem. Table 3.1 lists the notations that are frequently used in the remainder of this chapter.

Suppose that the influence propagation on G conforms to a realization ϕ that is sampled from Ω , i.e., for any seed set S , the nodes that it can influence are exactly the nodes that it can reach in ϕ . The *adaptive influence maximization (IM)* problem [9] considers that ϕ is unknown in advanced, but can be partially revealed after we choose some nodes as seeds.

Assume that we are to choose k seed nodes in r batches of equal size $b = k/r$, and that we are allowed to observe the influence propagation in ϕ for r times in total, once after the selection of each batch. The adaptive IM problem asks for r seed sets $S_1, S_2, \dots, S_r$, such that selecting S_i ($i \in [1, r]$) as the i -th batch maximizes the expected influence spread over the choices of $\phi \sim \Omega$. Observe that when $b = k$ (i.e., $r = 1$), the problem degenerates to the vanilla IM problem.

We aim to develop algorithms for adaptive IM that provide non-trivial guarantees in terms of both accuracy (i.e., the expected influence of $\bigcup_i S_i$) and efficiency (i.e., the time required to identify S_i). We do not consider the “waiting time” required to observe the influence of a seed node batch S_i before the selection of the next batch S_{i+1} , since it is independent of the algorithms used. That is, we target at helping the advertiser to identify S_{i+1} as quickly as possible after the effects of S_i have been observed.

3.2 Our Solution

In this section, we first propose a general framework ADAPTGREEDY upon which we can build specific algorithms with seed selection algorithms to address the adaptive IM problem. The seed selection algorithms are randomized algorithms as they estimate node influence through randomized sampling approaches. To cope with the randomness, we derive the corresponding *expected* and *worst-case* approximation guarantees that ADAPTGREEDY achieves respectively.

Table 3.1: Frequently used notations

Notation	Description
k	the total number of selected seed nodes
b	the number of nodes selected in each batch
S_i	the seed set selected from G_i
S_i^o	the optimal seed set in G_i
α	approximation guarantee for MAXCOVER with $\alpha = 1 - (1 - 1/b)^b$.
$\text{OPT}_{k,b}$	the optimal expected influence spread of k seed nodes under the setting of selecting b nodes in each batch
$\text{OPT}_b(G_i)$	the optimal expected influence spread of b seed nodes in G_i
$I_G(S)$	the number of nodes activated by S in G

3.2.1 Solution Framework

Algorithm 4 illustrates the framework for adaptive IM, referred to as ADAPTGREEDY. At the first glance, ADAPTGREEDY may seem similar to Vaswani and Lakshmanan’s method [22], since both techniques (i) adaptively select seed nodes in r batches and (ii) do not require exact computation of expected spreads. However, there is a crucial difference between the two: Vaswani and Lakshmanan’s method requires that the expected spread of *every* node set should be estimated with a small *fixed* relative error with respect to its own expectation, whereas ADAPTGREEDY just requires an expected approximation ratio of $\alpha(1 - \varepsilon_i)$ with respect to $\text{OPT}_b(G_i)$ (Line 5 in Algorithm 4), where $\text{OPT}_b(G_i)$ denotes the maximum expected spread of any size- b seed set on G_i and $\alpha = 1 - (1 - 1/b)^b$. The error requirement of ADAPTGREEDY is much more lenient than that of Vaswani and Lakshmanan’s method, and it can be achieved by several state-of-the-art solutions [10–12, 17] for vanilla influence maximization, i.e., it admits practical implementations.

In addition, ADAPTGREEDY is *flexible* in that it allows each batch of seed nodes S_i to be selected with different approximation guarantee $\alpha(1 - \varepsilon_i)$, whereas the existing solutions (e.g., [9]) for adaptive IM require that all seed sets $S_1, \dots, S_r$ should be processed with identical accuracy assurance. Therefore, ADAPTGREEDY is a general framework for the adaptive IM problem.

Algorithm 4: ADAPTGREEDY

Input: social network G , seed set size k , batch number r , approximation error

$\varepsilon_1, \varepsilon_2, \dots, \varepsilon_r$

Output: adaptively selected seed sets $S_1, \dots, S_r$

```

1  $b \leftarrow k/r$ ;
2  $G_1 \leftarrow G$ ;
3  $\alpha \leftarrow 1 - (1 - 1/b)^b$ ;
4 for  $i \leftarrow 1$  to  $r$  do
5   Identify a size- $b$  seed set  $S_i$  from  $G_i$ , such that  $S_i$  achieves an expected
     approximation ratio of at least  $\alpha(1 - \varepsilon_i)$  on  $G_i$ ;
6   Observe the influence of  $S_i$  in  $G_i$ ;
7   Remove all nodes in  $G_i$  that are influenced by  $S_i$ , and denote the resulting
     graph as  $G_{i+1}$ ;
8 return  $S_1, \dots, S_r$ 
    
```

3.2.2 Expected Approximation Guarantee

For sake of clarity, we first define some notations frequently used in the following discussion.

Let $\mathcal{G}_i$ be the set of possible choices for G_i , where the distribution on $\mathcal{G}_i$ is over both the randomness of realizations and the internal randomness of IM algorithm (Recall that the existing vanilla IM algorithms [10–12, 17] all adopt randomized sampling techniques to estimate the expected spread of node sets, resulting in randomized algorithms.). Let $\xi(G_i)$ be the *relative error* such that the returned solution S_i achieves an approximation ratio of $\alpha(1 - \xi(G_i))$ on G_i . Taking expectation over the internal randomness of the algorithm, let $\mathbb{E}[\xi(G_i)]$ be its expectation that the algorithm achieves an expected approximation ratio of $\alpha(1 - \mathbb{E}[\xi(G_i)])$ on G_i . Then, the following theorem gives the final approximation guarantee achieved by ADAPTGREEDY.

Theorem 3.1. *If ADAPTGREEDY returns an expected $\alpha(1 - \varepsilon_i)$ -approximate solution in the i -th batch, i.e., $\mathbb{E}[\xi(G_i)] \leq \varepsilon_i$, then it achieves an expected approximation guarantee of $1 - e^{\alpha(\varepsilon - 1)}$, where $\varepsilon = \frac{1}{r} \sum_{i=1}^r \varepsilon_i$, $\alpha = 1 - (1 - 1/b)^b$, r is the batch number and b is the batch size.*

Theorem 3.1 presents the corresponding expected approximation guarantee that ADAPT-GREEDY could achieve if the expected approximation of certain solution is obtained for

each batch. It is noteworthy that different combinations of batch number r and batch size b define different policies. Specifically, given any r and b , the approximation guarantee obtained by ADAPTGREEDY is against the corresponding optimal policy for the same batch number r and batch size b .

The proof of Theorem 3.1 is rather intricate as it requires non-trivial extensions of the theoretical results developed for adaptive submodular optimization [9]. We refer interested readers to Appendix for the details.

3.2.3 Worst-case Approximation Guarantee

Note that for non-adaptive IM, the existing algorithms [10–12, 17] have provided worst-case approximation guarantees with high probability, i.e., $(1 - 1/e - \varepsilon)$ -approximation guarantee with probability at least $1 - \delta$, where $\varepsilon, \delta \in (0, 1)$. In what follows, we also derive such a worst-case approximation guarantee for ADAPTGREEDY.

To begin with, we first provide a random approximation guarantee for ADAPTGREEDY as follows. The proof is analogous to Theorem 3.1. We refer interested readers to Appendix for the details.

Lemma 3.1. *Let X_i be the overall relative error on the i -th batch of seed selection and X be the average of X_i for r batches, i.e.,*

$$X_i = \frac{\mathbb{E}_{G_i \sim \mathcal{G}_i}[\xi(G_i) \cdot \text{OPT}_b(G_i)]}{\mathbb{E}_{G_i \sim \mathcal{G}_i}[\text{OPT}_b(G_i)]}, \text{ and } X = \frac{1}{r} \sum_{i=1}^r X_i.$$

Then, ADAPTGREEDY achieves an approximation guarantee of $1 - e^{\alpha(X-1)}$.

Note that, in Lemma 3.1, X_i and X are random variables as they rely on the random variable $\xi(G_i)$. Thus, the approximation guarantee in Lemma 3.1 is a random variable which is crucial in providing worst-case theoretical guarantees. To this end, a simple and intuitive idea is to show that $X_i \leq \varepsilon_i$ with high probability for every batch i . Then, by a union bound for r batches, we can obtain a worst-case approximation guarantee. However, it is hard to derive such a non-trivial worst-case approximation guarantee, as the probability that $X_i > \varepsilon_i$ could be large though the probability of $\xi(G_i) > \varepsilon_i$ on G_i is small. To explain, the number of possible G_i in the set $\mathcal{G}_i$ can be as large as $O(2^m)$

where m is the number of edges in G . Once there exists one instance of G_i from $\mathcal{G}_i$ such that $\xi(G_i) > \varepsilon_i$, it is possible that $X_i > \varepsilon_i$. In other words, to ensure that $X_i \leq \varepsilon_i$, one sufficient way is to guarantee that $\xi(G_i) \leq \varepsilon_i$ for every G_i . However, such a requirement is too stringent to satisfy. In particular, the probability of $X_i > \varepsilon_i$ can be as large as $O(2^m \cdot \delta_i)$ even though $\Pr[\xi(G_i) > \varepsilon_i] \leq \delta_i$ for each G_i . Unfortunately, two recent papers [2] and [1] claim that if $\Pr[\xi(G_i) > \varepsilon_i] \leq \delta_i$ for every G_i , then $\Pr[X_i > \varepsilon_i] \leq \delta_i$. This misclaim makes their approximation guarantees invalid. More details are presented in Section 3.4.

On the other hand, the strategy that demands every $X_i \leq \varepsilon_i$ with high probability is also overly conservative. For example, suppose that there exists one X_j satisfying $X_j > \varepsilon_j$, i.e., it fails to achieve the overall $\alpha(1 - \varepsilon_j)$ -approximation in the j -th batch. Even in that case, the overall approximation ratio of ADAPTGREEDY could still be better than $1 - e^{\alpha(\varepsilon-1)}$, as long as there exists another X_i satisfying $X_i \leq \varepsilon_i + \varepsilon_j - X_j$. In other words, the deficiency of one batch can be compensated, as long as there exists other batches whose quality is above the bar by a sufficient margin.

Formally, as the approximation ratio $\alpha(1 - X_i)$ in each batch of ADAPTGREEDY is a random variable, the overall approximation guarantee of ADAPTGREEDY, namely, $1 - e^{\alpha(X-1)}$, depends on the mean of all r variables. Intuitively, when r is sizable, $X = \frac{1}{r} \sum_{i=1}^r X_i$ should be concentrated to its expectation, i.e., $\mathbb{E}[X]$. Note that $\mathbb{E}[X_i] \leq \varepsilon_i$ holds if $\mathbb{E}[\xi(G_i)] \leq \varepsilon_i$ holds for every G_i , which is exactly the requirement of ADAPTGREEDY for each seed selection batch. That is, instead of formulating the approximation ratio of ADAPTGREEDY based on the worst-case guarantee of each S_i (i.e., ε_i), we might derive it based on each S_i 's expected approximation ratio.

To make the above idea work, the distance between X and its expectation $\mathbb{E}[X]$ is to be bounded with high probability. However, there is a challenge that we need to address. As the selection of the i -th seed set S_i is dependent on the results of the $(i - 1)$ -th seed set S_{i-1} , the random variables $X_1, X_2, \dots, X_r$ are *correlated*, which makes it rather non-trivial to derive concentration results for $\frac{1}{r} \sum_{i=1}^r X_i$. We circumvent this issue with a theoretical analysis by leveraging Azuma-Hoeffding inequality for martingales [27].

Definition 3.1 (Martingale [27]). *A sequence of random variables $Z_0, Z_1, \dots$ is a martingale with respect to the sequence $Y_0, Y_1, \dots$ if, for all $i \geq 0$, the following conditions hold:*

- Z_i is a function of $Y_0, Y_1, \dots, Y_i$;
- $\mathbb{E}[|Z_i|] < \infty$;
- $\mathbb{E}[Z_{i+1} \mid Y_1, \dots, Y_i] = Z_i$.

Lemma 3.2 (Azuma-Hoeffding Inequality [27]). *Let $Z_0, Z_1, \dots$ be a martingale with respect to the sequence of random variables $Y_0, Y_1, \dots$ such that*

$$B_i \leq Z_i - Z_{i-1} \leq B_i + c_i$$

for some constants c_i and for some random variables B_i that may be functions of $Y_0, \dots, Y_{i-1}$. Then, for any $r \geq 0$ and any $\lambda > 0$,

$$\Pr [Z_r \geq Z_0 + \lambda] \leq e^{-2\lambda^2 / \sum_{i=1}^r c_i^2}.$$

Based on the above Azuma-Hoeffding inequality, we provide a concentration bound for possibly correlated random variables as follows.

Corollary 3.1. *Let $Y_1, \dots, Y_r$ be any sequence of random variables satisfying $|Y_i| \leq \beta$ and $\mathbb{E}[Y_i \mid Y_1, \dots, Y_{i-1}] \leq \gamma_i$ for every $i = 1, \dots, r$. Then, we have*

$$\Pr \left[\sum_{i=1}^r Y_i \geq \sum_{i=1}^r \gamma_i + \sqrt{r}\beta\lambda \right] \leq e^{-\lambda^2/2}. \quad (3.1)$$

Corollary 3.1. Let $Z_0 = Y_0 = 0$ and for every $i = 1, \dots, r$,

$$Z_i := \sum_{j=1}^i (Y_j - \mathbb{E}[Y_j \mid Y_1, \dots, Y_{j-1}]).$$

Then, it is easy to verify that $\mathbb{E}[|Z_i|] < \infty$ and $\mathbb{E}[Z_i \mid Y_1, \dots, Y_{i-1}] = Z_{i-1}$, which indicates that Z_i is a martingale. In addition, let $B_i := -\beta - \mathbb{E}[Y_i \mid Y_1, \dots, Y_{i-1}]$. As $|Y_i| \leq \beta$, we can get that $Z_i - Z_{i-1} = Y_i - \mathbb{E}[Y_i \mid Y_1, \dots, Y_{i-1}]$ is in the range of $[B_i, B_i + 2\beta]$. Thus, according to Lemma 3.2,

$$\Pr [Z_r \geq \sqrt{r}\beta\lambda] \leq e^{-2r\beta^2\lambda^2 / \sum_{i=1}^r (2\beta)^2} = e^{-\lambda^2/2}. \quad (3.2)$$

On the other hand, as $\mathbb{E}[Y_i \mid Y_1, \dots, Y_{i-1}] \leq \gamma_i$ for every $i = 1, \dots, r$, we have $Z_r \geq \sum_{i=1}^r (Y_i - \gamma_i)$. As a consequence

$$\Pr \left[\sum_{i=1}^r (Y_i - \gamma_i) \geq \sqrt{r}\beta\lambda \right] \leq \Pr [Z_r \geq \sqrt{r}\beta\lambda]. \quad (3.3)$$

Combining (3.2) and (3.3) completes the proof. $\square$

Recall that we have shown in Section 3.2.2 that the approximation guarantee of ADAPTGREEDY is determined by the summation of the error factors, i.e., $X = \frac{1}{r} \sum_{i=1}^r X_i$, and these error factors could be correlated. Therefore, if we consider them as the random variables in Corollary 3.1, we can get a bound on their summation, based on which we can derive the overall approximation guarantee by Theorem 3.2.

Theorem 3.2. *For any given $\delta \in (0, 1)$, let $\varepsilon = \frac{1}{r} \sum_{i=1}^r \varepsilon_i + \sqrt{1/(2r) \cdot \ln(1/\delta)}$. Then, ADAPTGREEDY achieves the worst-case approximation ratio $1 - e^{\alpha(\varepsilon-1)}$ with a probability of at least $1 - \delta$.*

Theorem 3.2. When the first $i - 1$ batches of nodes are selected, the distribution $\mathcal{G}_i$ over all realizations is determined. For simplicity, we use the sequence $\{X_1, \dots, X_{i-1}\}$ to represent the overall relative error for node selection of the first $i - 1$ batches for all realizations. Thus, for a given sequence of $\{X_1, \dots, X_{i-1}\}$, we have

$$\begin{aligned} \mathbb{E}[X_i \mid X_1, \dots, X_{i-1}] &= \frac{\mathbb{E}_{G_i \sim \mathcal{G}_i}[\mathbb{E}[\xi(G_i)] \cdot \text{OPT}_b(G_i)]}{\mathbb{E}_{G_i \sim \mathcal{G}_i}[\text{OPT}_b(G_i)]} \\ &\leq \frac{\mathbb{E}_{G_i \sim \mathcal{G}_i}[\varepsilon_i \cdot \text{OPT}_b(G_i)]}{\mathbb{E}_{G_i \sim \mathcal{G}_i}[\text{OPT}_b(G_i)]} \\ &= \varepsilon_i, \end{aligned}$$

where the expectation $\mathbb{E}[\cdot]$ is over the internal randomness of seed selection algorithm, and $\mathbb{E}[\xi(G_i)] \leq \varepsilon_i$ by the requirement. Thus, by letting $Y_i = X_i - 1/2$, we have $|Y_i| \leq 1/2$ and $\mathbb{E}[Y_i \mid Y_1, \dots, Y_{i-1}] \leq \varepsilon_i - 1/2$. Hence, by Corollary 3.1, it holds that

$$\begin{aligned} \Pr[X \geq \varepsilon] &= \Pr \left[X \geq \frac{1}{r} \sum_{i=1}^r \varepsilon_i + \sqrt{\frac{1}{2r} \ln \frac{1}{\delta}} \right] \\ &= \Pr \left[\sum_{i=1}^r Y_i \geq \sum_{i=1}^r (\varepsilon_i - \frac{1}{2}) + \sqrt{r} \frac{1}{2} \sqrt{2 \ln \frac{1}{\delta}} \right] \\ &\leq e^{-2 \ln \frac{1}{\delta} / 2} \\ &= \delta, \end{aligned}$$

which completes the proof. □ □

Note that ADAPTGREEDY's worst-case approximation ratio (Theorem 3.2) is worse than its expected approximation guarantee (Theorem 3.1), where the overall error factor ε of the former is larger by an additive factor of $\sqrt{1/(2r) \cdot \ln(1/\delta)}$ than the later.

Besides, the worst-case approximation ratio is meaningful only when $\varepsilon < 1$, i.e., $\delta > e^{-2r(1-\frac{1}{r}\sum_{i=1}^r \varepsilon_i)^2}$.

3.3 Instantiations of AdaptGreedy

In this section, we first design a non-adaptive IM algorithm EPIC to instantiate ADAPT-GREEDY. We then analyze its expected approximation guarantee and time complexity for each batch. Finally, we discuss how ADAPTGREEDY instantiated with EPIC achieves the expected and worst-case approximation guarantee and the corresponding expected time complexity respectively.

3.3.1 Design of Instantiation Algorithm

3.3.1.1 Instantiation using Existing Algorithms

As shown in Algorithm 4, ADAPTGREEDY requires identifying a size- b seed set S_i from the i -th residual graph G_i , such that

$$\mathbb{E}_{\mathcal{A}}[\mathbb{E}[I_{G_i}(S_i)]] \geq \alpha(1 - \varepsilon_i)\text{OPT}_b(G_i), \quad (3.4)$$

where $\mathbb{E}[I_{G_i}(S_i)]$ is the expected spread of S_i on G_i and its expectation $\mathbb{E}_{\mathcal{A}}[\mathbb{E}[I_{G_i}(S_i)]]$ is over the internal randomness of the algorithm, $\text{OPT}_b(G_i)$ is the maximum expected spread of any size- b seed set on G_i . We observe that such a seed set S_i could be obtained by applying the state-of-the-art algorithms (e.g., [10–12, 17]) for vanilla influence maximization (IM) on G_i . In particular, these algorithms are randomized, and they provide a *worst-case* approximation guarantee as follows: given a seed set size b , a relative error threshold ε_i and a failure probability δ_i , they output a size- b seed set S_i in G_i whose expected spread is $\alpha - \xi(G_i)$ times the maximum expected spread of any size- b seed set on G_i , such that $\xi(G_i) \leq \varepsilon_i$ with at least $1 - \delta_i$ probability. To fulfill the requirement of Equation (3.4), we can set $\xi(G_i) = \alpha\varepsilon_i$.

By applying such algorithms with any given parameters ε_i and δ_i , we obtain that

$$\mathbb{E}[\xi(G_i)] \leq \varepsilon_i \cdot (1 - \delta_i) + \delta_i = \hat{\varepsilon}_i, \quad (3.5)$$

where $\hat{\varepsilon}_i := \varepsilon_i + (1 - \varepsilon_i) \cdot \delta_i \approx \varepsilon_i$. According to Theorem 3.1, we can guarantee that such an instantiation of ADAPTGREEDY yields an *expected approximation ratio* of $1 - e^{\alpha(\varepsilon-1)}$ by setting certain $(\varepsilon_1, \delta_1), \dots, (\varepsilon_r, \delta_r)$ satisfying $\frac{1}{r} \sum_{i=1}^r \hat{\varepsilon}_i \leq \varepsilon$.

But how efficient is the above instantiation? To answer this question, we need to investigate the time complexity of the vanilla IM algorithms in [11]. The theoretical analysis in [11] shows that if we are to achieve $(\alpha - \varepsilon_i)$ -approximation on G_i with at least $1 - \delta_i$ probability, then the expected computation cost is $O((b \log n_i + \log \frac{1}{\delta_i})(m_i + n_i)/\varepsilon_i^2)$, where n_i and m_i denote the number of nodes and edges in G_i respectively. Since $n_i \leq n$ and $m_i \leq m$, the expected time required to process G_i is $O((b \log n + \log \frac{1}{\delta_i})(m + n)/\varepsilon_i^2)$. As such, all r batches of seed nodes can be identified in $O(\sum_{i=1}^r (b \log n + \log \frac{1}{\delta_i})(m + n)/\varepsilon_i^2)$ expected time. By setting a pair of parameters $(\varepsilon'_i, \delta_i)$ in the vanilla IM algorithms as $\varepsilon'_i = (\varepsilon_i - \delta_i)/(1 - \delta_i)$, we can achieve an expected approximation ratio of $\alpha(1 - \varepsilon_i)$ in the i -th batch. This shows that the total time complexity for achieving the final expected approximation ratio of $1 - e^{\alpha(\frac{1}{r} \sum_{i=1}^r \varepsilon_i - 1)}$ is $O(\sum_{i=1}^r (b \log n + \log \frac{1}{\delta_i})(m + n)/\varepsilon_i^2)$.

Rationale for an Improved Approach. The aforementioned instantiation of ADAPTGREEDY is straightforward and intuitive, but is far from optimized in terms of its approximation guarantee. To explain, recall that it requires each seed set S_i to achieve $\alpha(1 - \varepsilon_i)$ -approximation on G_i with probability at least $1 - \delta_i$, based on which it provides an overall expected approximation ratio of $1 - e^{\alpha(\varepsilon-1)}$ with $\varepsilon \geq \frac{1}{r} \sum_{i=1}^r \hat{\varepsilon}_i$ where $\hat{\varepsilon}_i := \varepsilon_i + (1 - \varepsilon_i) \cdot \delta_i \approx \varepsilon_i$. In other words, it imposes a stringent worst-case approximation guarantee on each seed set S_i . This, however, might be overly conservative. Intuitively, the expected approximation error factor $\mathbb{E}[\xi(G_i)]$ should be much smaller than the naive upper bound $\hat{\varepsilon}_i$ deduced from the worst-case approximation. To the best of our knowledge, there is no known result for vanilla IM with tight expected approximation guarantees. This motivates us to develop a vanilla IM algorithm tailored for ADAPTGREEDY, as we show in the following section.

3.3.1.2 IM Algorithm with Expected Approximation

As discussed in Section 3.3.1.1, the existing IM algorithms provide only a worst-case approximation guarantee $\alpha - \xi(G_i)$, i.e., they ensure that their absolute error factor $\xi(G_i)$ is no more than the input threshold ε_i with high probability. To optimize the performance

of ADAPTGREEDY, however, we are in need of one non-adaptive IM algorithm with expected approximation guarantee $\alpha(1 - \mathbb{E}[\xi(G_i)])$, where the expectation is over the randomness of IM algorithm, to lead to one tighter approximation guarantee for adaptive influence maximization than that using the existing IM vanilla algorithms.

In what follows, we present a new non-adaptive IM algorithm, referred to as EPIC¹, that returns a solution with expected approximation guarantee. To this end, we first introduce the concept of *reverse reachable sets* (RR-sets) [13], which is the basis of our algorithm.

RR-Sets. In a nutshell, RR-sets are subgraph samples of G that can be used to efficiently estimate the expected spreads of any given seed sets. Specifically, a random RR-set of G is generated by first selecting a node $v \in V$ uniformly at random, and then taking the nodes that can reach v in a random graph generated by independently removing each edge $e \in E$ with probability $1 - p(e)$. If a seed node set S has large expected influence spread, then the probability that S intersects with a random RR-set is high, as shown in the following equation [13]:

$$\mathbb{E}[I_G(S)] = n \cdot \Pr[R \cap S \neq \emptyset], \quad (3.6)$$

where R is a random RR-set. This result suggests a simple method for estimating the expected influence spread of any node set S : we can use a set $\mathcal{R}$ of random RR-sets to estimate the value of $\Pr[R \cap S \neq \emptyset]$ and hence $\mathbb{E}[I_G(S)]$. In particular, let $\text{Cov}_{\mathcal{R}}(S)$ denote the number of RR-sets in $\mathcal{R}$ that overlap S . Then the value of $\mathbb{E}[I_G(S)]$ can be unbiasedly estimated by $n \cdot F_{\mathcal{R}}(S)$, where

$$F_{\mathcal{R}}(S) = \text{Cov}_{\mathcal{R}}(S)/|\mathcal{R}|. \quad (3.7)$$

By the law of large numbers, $n \cdot F_{\mathcal{R}}(S)$ should converge to $\mathbb{E}[I_G(S)]$ when $|\mathcal{R}|$ is sufficiently large, which provides a way to estimate $\mathbb{E}[I_G(S)]$ to any desired accuracy level. However, due to the cost of generating RR-sets, there is a tradeoff between accuracy and efficiency in any algorithms using RR-set sampling.

The EPIC Algorithm. Algorithm 5 shows the pseudo-code of our EPIC algorithm which borrows the idea from the OPIM-C algorithm in [17] via (i) starting from a small

¹Expected approximation for influence maximization.

Algorithm 5: EPIC(G_i, b, ε_i)

Input: Graph G_i , seed set size b , and error threshold ε_i .

Output: A b -size seed set S_i that provides an expected approximation guarantee of at least $\alpha(1 - \varepsilon_i)$.

```

1 let  $\delta_i \leftarrow 0.01\varepsilon_i b / n_i$ ;
2 let  $\varepsilon'_i \leftarrow (b\varepsilon_i - \delta_i n_i) / (b - \delta_i n_i)$ ;
3 let  $\varepsilon_a \leftarrow \varepsilon'_i / (1 - \varepsilon'_i)$ ;
4 let  $i_{\max} \leftarrow \lceil \log_2 \frac{(2+2\varepsilon_a/3)n_i}{\varepsilon_a^2} \rceil + 1$  and  $a_i \leftarrow \ln \frac{2i_{\max}}{\delta_i}$ ;
5 initialize  $\theta_0 \leftarrow \frac{1}{b} (\ln \frac{2}{\delta_i} + \ln \binom{n_i}{b})$ ;
6 generate two sets  $\mathcal{R}_1$  and  $\mathcal{R}_2$  of random RR-sets, with  $|\mathcal{R}_1| = |\mathcal{R}_2| = \theta_0$ ;
7 for  $i \leftarrow 1$  to  $i_{\max}$  do
8    $\langle S_i, F^u(S_i^o) \rangle \leftarrow \text{MAXCOVER}(\mathcal{R}_1, b)$ ;
9    $F^l(S_i) \leftarrow \left( \sqrt{F_{\mathcal{R}_2}(S_i) + \frac{2a_i}{9|\mathcal{R}_2|}} - \sqrt{\frac{a_i}{2|\mathcal{R}_2|}} \right)^2 - \frac{a_i}{18|\mathcal{R}_2|}$ ;
10  if  $\frac{F^l(S_i)}{F^u(S_i^o)} \geq \alpha(1 - \varepsilon'_i)$  or  $i = i_{\max}$  then return  $S_i$ ;
11  double the sizes of  $\mathcal{R}_1$  and  $\mathcal{R}_2$  with new RR-sets;
    
```

number of RR-sets and (ii) iteratively increasing the RR-set number until a satisfactory solution is identified. The key difference between the two algorithms lies in the way that they compute the upper bound $F^u(S_i^o)$, i.e., the fraction of RR-sets in $\mathcal{R}_1$ covered by S_i^o in each iteration where S_i^o is an optimal seed set in G_i . In particular, in OPIM-C, the upper bound $F^u(S_i^o)$ is ensured to be no smaller than the *expected* fraction of RR-sets in $\mathcal{R}_1$ covered by S_i^o with high probability. This needs OPIM-C to provide the worst-case approximation guarantee with high probability. In contrast, in each iteration of EPIC, the upper bound $F^u(S_i^o)$ is only required to be no smaller than the *true* fraction of RR-sets in $\mathcal{R}_1$ covered by S_i^o , based on which rigorous bounds on its expected approximation guarantee can be derived². In what follows, we discuss the details of EPIC and its subroutine MAXCOVER (in Algorithm 6).

Based on the RR-set sampling method described previously, a simple approach for selecting S_i with a large expected influence spread is to first generate a set $\mathcal{R}$ of RR-sets, and then invoke the MAXCOVER algorithm on $\mathcal{R}$. In particular, MAXCOVER uses a simple greedy approach to identify $S_i \subseteq V_i$ such that S_i overlaps as many RR-sets in $\mathcal{R}$

²We use the *Mean Absolute Deviation (MAD)* lemma to derive this bound in [2]. However, we realize later that the sampling stop condition in EPIC makes MAD lemma ineligible to apply.

Algorithm 6: MAXCOVER($\mathcal{R}, b$)

Input: A set $\mathcal{R}$ of random RR-sets, and b .

Output: A node set S_i , and an upper bound $F^u(S_i^o)$ on the fraction of RR-sets in $\mathcal{R}$ covered by S_i^o .

```

1  $S_i \leftarrow \emptyset$ ;
2  $F^u(S_i^o) \leftarrow F^u(S_i^o \mid S_i)$  which is computed by (3.8);
3 for  $i \leftarrow 1$  to  $b$  do
4    $u \leftarrow \arg \max_{v \in V_i} (\text{Cov}_{\mathcal{R}}(S_i \cup \{v\}) - \text{Cov}_{\mathcal{R}}(S_i))$ ;
5   insert  $u$  into  $S_i$ ;
6   compute  $F^u(S_i^o \mid S_i)$  by (3.8) based on the new  $S_i$ ;
7   update  $F^u(S_i^o) \leftarrow \min\{F^u(S_i^o), F^u(S_i^o \mid S_i)\}$ ;
8 return  $\langle S_i, F^u(S_i^o) \rangle$ ;
```

as possible. Since $F_{\mathcal{R}}(\cdot)$ is a submodular function for any set $\mathcal{R}$ of RR-sets [13], given any node set $S \subseteq V$ with $|S| \leq b$, we know that

$$F^u(S_i^o \mid S) = F_{\mathcal{R}}(S) + \sum_{v \in \text{maxMC}(S, b)} (F_{\mathcal{R}}(S \cup \{v\}) - F_{\mathcal{R}}(S)) \quad (3.8)$$

is an upper bound on $F_{\mathcal{R}}(S_i^o)$, where S_i^o is an optimal seed set in G_i and $\text{maxMC}(S, b)$ is the set of b nodes with the top- b largest marginal coverage in $\mathcal{R}$ with respect to S . As a consequence, the smallest one $F^u(S_i^o) = \min_{S_i} \{F^u(S_i^o \mid S_i)\}$ during the greedy procedure ensures that

$$F^u(S_i^o) \geq F_{\mathcal{R}}(S_i^o). \quad (3.9)$$

In addition, according to [17], we also have

$$F_{\mathcal{R}}(S_i) \geq \alpha F^u(S_i^o) \quad (3.10)$$

where α is as defined in Algorithm 4. Putting (3.9) and (3.10) together yields

$$F_{\mathcal{R}}(S_i) \geq \alpha F^u(S_i^o) \geq \alpha F_{\mathcal{R}}(S_i^o). \quad (3.11)$$

Note that $n \cdot F_{\mathcal{R}}(S_i)$ and $n \cdot F_{\mathcal{R}}(S_i^o)$ are unbiased estimations of $\mathbb{E}[I(S_i)]$ and $\mathbb{E}[I(S_i^o)]$, respectively. Therefore, when $|\mathcal{R}|$ is large, the approximation guarantee of S_i converges to α according to Equation (3.11).

To strike a balance between the quality of S_i and the number of RR-sets used to derive S_i , EPIC iterates in a careful manner as follows. In each iteration, it maintains two sets of

random RR-sets $\mathcal{R}_1$ and $\mathcal{R}_2$ with $|\mathcal{R}_1| = |\mathcal{R}_2|$. It invokes MAXCOVER on $\mathcal{R}_1$ to identify a seed set S_i , and then utilizes $\mathcal{R}_2$ to test whether S_i provides a good approximation guarantee. Initially, the cardinalities of $\mathcal{R}_1$ and $\mathcal{R}_2$ are small constants determined by the parameter θ_0 in Line 5 in the first iteration of EPIC. Then, whenever EPIC finds that the quality of the seed set S_i generated in an iteration is not satisfactory, it doubles the sizes of $\mathcal{R}_1$ and $\mathcal{R}_2$. This process repeats until that a qualified solution is identified or the sizes of $\mathcal{R}_1$ and $\mathcal{R}_2$ reach $2^{i_{\max}-1}\theta_0$ which exceeds the threshold $\frac{(2+2\varepsilon_a/3)n_i}{\varepsilon_a^2 b} \left(\ln \frac{2}{\delta_i} + \ln \binom{n_i}{b} \right)$ (Line 10).

As explained before, one of the main designing goals for EPIC is to achieve an expected approximation ratio of $\alpha(1 - \varepsilon_i)$. EPIC achieves this goal by a series of operations in each iteration, whose implications are briefly explained as follows.

In each iteration, EPIC first applies MAXCOVER on $\mathcal{R}_1$ (Line 8), which returns a seed set S_i and an upper bound $F^u(S_i^o)$ on $F_{\mathcal{R}_1}(S_i^o)$, i.e.,

$$F^u(S_i^o) \geq F_{\mathcal{R}_1}(S_i^o). \quad (3.12)$$

After that, EPIC uses $\mathcal{R}_2$ to estimate the expected spread of S_i (i.e., $\mathbb{E}[I_{G_i}(S_i)]$). Observe that $|\mathcal{R}_2|F_{\mathcal{R}_2}(S_i)$ is a binomial random variable due to Equation (3.7). Accordingly, EPIC uses the Chernoff-like martingale concentration bound to set a threshold $F^l(S_i)$ (Line 9) such that

$$\mathbb{E}[n_i F_{\mathcal{R}_2}(S_i)] \geq n_i F^l(S_i) \quad (3.13)$$

should hold with high probability. Intuitively, Equation (3.13) implies that $n_i F^l(S_i)$ gives a sufficiently accurate lower bound on $\mathbb{E}[I_{G_i}(S_i)]$. After that, EPIC checks whether

$$F^l(S_i)/F^u(S_i^o) \geq \alpha(1 - \varepsilon'_i) \quad (3.14)$$

holds in Line 10. Intuitively, if Equation (3.14) is true, then we know that $\mathbb{E}[n_i F_{\mathcal{R}_2}(S_i)]$ is no smaller than $\alpha(1 - \varepsilon'_i)n_i F_{\mathcal{R}_1}(S_i^o)$ and it suffices to conclude our result by taking the expectation. Specifically, combining Equations (3.12)–(3.14) and taking the expectation with respect to the randomness of the algorithm, we can derive a quantitative relationship

between $\mathbb{E}_{\mathcal{A}}[\mathbb{E}[I_{G_i}(S_i)]]$ and $\text{OPT}_b(G_i)$ when a seed set S_i is returned:

$$\begin{aligned}
 & \mathbb{E}_{\mathcal{A}}[\mathbb{E}[I_{G_i}(S_i)]] \\
 & \geq \mathbb{E}_{\mathcal{A}}[n_i F^l(S_i)] - \delta_i \alpha (1 - \varepsilon'_i) n_i \\
 & \geq \mathbb{E}_{\mathcal{A}}[\alpha (1 - \varepsilon'_i) n_i F^u(S_i^o)] - \delta_i \alpha (1 - \varepsilon'_i) n_i \\
 & \geq \mathbb{E}_{\mathcal{A}}[\alpha (1 - \varepsilon'_i) n_i F_{\mathcal{R}_1}(S_i^o)] - \delta_i \alpha (1 - \varepsilon'_i) n_i \\
 & = \alpha (1 - \varepsilon'_i) \text{OPT}_b(G_i) - \delta_i \alpha (1 - \varepsilon'_i) n_i \\
 & \geq \alpha (1 - \varepsilon'_i) (1 - \delta_i n_i / b) \text{OPT}_b(G_i),
 \end{aligned}$$

where the first inequality is due to the fact that (3.13) holds with high probability and $\delta_i \alpha (1 - \varepsilon'_i) n_i$ is used to offset the failed scenario, and the equality is due to the martingale stopping theorem [27] (see details in Section 3.3.2). This proves the $\alpha(1 - \varepsilon_i)$ expected approximation ratio of $\mathbb{E}_{\mathcal{A}}[\mathbb{E}[I_{G_i}(S_i)]]$ as $\varepsilon_i = \varepsilon'_i + (1 - \varepsilon'_i) \delta_i n_i / b$.

It is easy to see that the expected approximation guarantee of EPIC is better than those of vanilla IM algorithms, and thus instantiating ADAPTGREEDY using EPIC can lead to performance improvement for adaptive IM. Note that EPIC does not provide the worst-case approximation guarantee with high probability, as against the state-of-the-art IM algorithms. The reason behind is that $n_i F^u(S_i^o)$ is likely to be smaller than $\text{OPT}_b(G_i)$ though $n_i F^u(S_i^o)$ is an upper bound on $n_i F_{\mathcal{R}_1}(S_i^o)$ as shown in (3.12).

3.3.2 Theoretical Analysis

Based on the discussions in Section 3.3.1.2, we show the details of theoretical analysis of EPIC. We prove our main results for the expected approximation guarantee and the time complexity of EPIC as follows.

Expected Approximation Guarantee. We establish the expected approximation guarantee of EPIC in the following theorem.

Theorem 3.3. *For any G_i , EPIC returns a seed set S_i satisfying*

$$\mathbb{E}_{\mathcal{A}}[\mathbb{E}[I_{G_i}(S_i)]] \geq \alpha(1 - \varepsilon_i) \text{OPT}_b(G_i). \quad (3.15)$$

To prove Theorem 3.3, we first prove the following lemma.

Lemma 3.3. *Let $\varepsilon_a \in (0, 1)$, $\delta_a \in (0, 1)$, and*

$$\theta_{\max} = \frac{(2 + 2\varepsilon_a/3)n_i}{\varepsilon_a^2 b} \left(\ln \frac{1}{\delta_a} + \ln \binom{n_i}{b} \right). \quad (3.16)$$

If a set of random RR-sets $\mathcal{R}$ are generated such that $|\mathcal{R}| \geq \theta_{\max}$, then with probability at least $1 - \delta_a$, the greedy algorithm returns a solution S_i satisfying

$$\mathbb{E}[I_{G_i}(S_i)] \geq \frac{\alpha}{1 + \varepsilon_a} n_i F_{\mathcal{R}}(S_i^o). \quad (3.17)$$

Lemma 3.3. According to Chernoff-like martingale concentration bound [11], for any S that is independent of $\mathcal{R}$, we have

$$\begin{aligned} & \Pr [n_i F_{\mathcal{R}}(S) > (1 + \varepsilon_a) \mathbb{E}[n_i F_{\mathcal{R}}(S)]] \\ &= \exp \left(-\frac{\varepsilon_a^2}{2 + 2\varepsilon_a/3} \cdot \frac{|\mathcal{R}| \mathbb{E}[n_i F_{\mathcal{R}}(S)]}{n_i} \right) \\ &\leq \delta_a / \binom{n_i}{b}. \end{aligned}$$

By the union bound, S_i returned by the greedy algorithm satisfies

$$\Pr [n_i F_{\mathcal{R}}(S_i) > (1 + \varepsilon_a) \mathbb{E}[n_i F_{\mathcal{R}}(S_i)]] \leq \delta_a. \quad (3.18)$$

On the other hand, due to the submodularity of $F_{\mathcal{R}}(\cdot)$, we have

$$F_{\mathcal{R}}(S_i) \geq \alpha F_{\mathcal{R}}(S_i^o). \quad (3.19)$$

Combining (3.18) and (3.19) completes the proof. □ □

Next, we use the following martingale stopping theorem [27] to prove Theorem 3.3.

Lemma 3.4 (Martingale Stopping Theorem [27]). *If $Z_0, Z_1, \dots$ is a martingale with respect to $Y_1, Y_2, \dots$ and if T is a stopping time for $Y_1, Y_2, \dots$, then*

$$\mathbb{E}[Z_T] = \mathbb{E}[Z_0] \quad (3.20)$$

whenever one of the following holds:

- *the Z_i are bounded, so there is a constant c such that, for all i , $|Z_i| \leq c$;*
- *T is bounded;*

- $\mathbb{E}[T] < \infty$, and there is a constant c such that $\mathbb{E}[|Z_{i+1} - Z_i| \mid Y_1, \dots, Y_i] < c$.

Theorem 3.3. Let $\mathcal{E}_1$ and $\mathcal{E}_2$ denote the following events:

$$\begin{aligned}\mathcal{E}_1(S_i): \mathbb{E}[I_{G_i}(S_i)] &\geq \alpha(1 - \varepsilon'_i)n_i F_{\mathcal{R}_1}(S_i^o), \\ \mathcal{E}_2(S_i): \mathbb{E}[I_{G_i}(S_i)] &\geq n_i F^l(S_i).\end{aligned}$$

Let T be the stopping time (i.e., the iteration in which EPIC returns S_i), which is bounded by $i_{\max}$. Let $\varepsilon_a = \varepsilon'_i/(1 - \varepsilon'_i)$ and $\delta_a = \delta_i/2$ for $\theta_{\max}$ defined in Equation (3.16). When $T = i_{\max}$, it is easy to verify that $|\mathcal{R}_1| = 2^{i_{\max}-1}\theta_0 \geq \theta_{\max}$. Hence, by Lemma 3.3, we have

$$\Pr[(T = i_{\max}) \wedge \neg \mathcal{E}_1(S_i)] \leq \delta_i/2. \quad (3.21)$$

On the other hand, when $T = t < i_{\max}$, let $\mathcal{S}_t$ be the set of possible node sets selected by EPIC (but not necessarily returned), where each $S \in \mathcal{S}_t$ has a probability $\Pr[S]$ such that $\sum_{S \in \mathcal{S}_t} \Pr[S] = 1$. Then, we have

$$\begin{aligned}\Pr[(T = t) \wedge \neg \mathcal{E}_1(S_i)] &\leq \Pr[(T = t) \wedge \neg \mathcal{E}_2(S_i)] \\ &\leq \sum_{S \in \mathcal{S}_t} \Pr[(T = t) \wedge \neg \mathcal{E}_2(S)] \cdot \Pr[S_i = S] \\ &\leq \sum_{S \in \mathcal{S}_t} \delta_i/(2i_{\max}) \cdot \Pr[S_i = S] \\ &\leq \delta_i/(2i_{\max}),\end{aligned}$$

where the first inequality is because if $\mathcal{E}_2$ happens then $\mathcal{E}_1$ must also happen, the second inequality is by the fact that only a subset of the node sets in $\mathcal{S}_t$ are returned, and the third inequality is obtained from [17] for any node set S that is independent of $\mathcal{R}_2$. As a consequence, by a union bound,

$$\Pr \left[\bigvee_{t=1}^{i_{\max}-1} \left((T = t) \wedge \neg \mathcal{E}_1(S_i) \right) \right] \leq \delta_i/2. \quad (3.22)$$

Combining Equations (3.21) and (3.22) shows that the event $\mathcal{E}_1(S_i)$ does not happen with probability at most δ_i no matter when the algorithm stops. Therefore, EPIC returns a

random solution S_i satisfying $\mathbb{E}[I_{G_i}(S_i)] \geq \alpha(1 - \varepsilon'_i)n_i F_{\mathcal{R}_1}(S_i^o)$ with $1 - \delta_i$ probability. Thus, adding an additive factor of $\delta_i \alpha(1 - \varepsilon'_i)n_i$ ensures that

$$\begin{aligned} \mathbb{E}_{\mathcal{A}}[\mathbb{E}[I_{G_i}(S_i)]] + \delta_i \alpha(1 - \varepsilon'_i)n_i \\ \geq \alpha(1 - \varepsilon'_i)\mathbb{E}_{\mathcal{A}}[n_i F_{\mathcal{R}_1}(S_i^o)]. \end{aligned} \quad (3.23)$$

Subsequently, the main challenge lies in how we connect $n_i F_{\mathcal{R}_1}(S_i^o)$ with $\text{OPT}_b(G_i)$. Note that $F_{\mathcal{R}_1}(S_i^o)$ is also a random variable with respect to $\mathcal{R}_1$. At the first glance, it seems that this analysis is difficult as the stopping time is a random variable. However, fortunately, by utilizing the martingale stopping theorem [27], we can bridge the gap between $n_i F_{\mathcal{R}_1}(S_i^o)$ and $\text{OPT}_b(G_i)$ as follows.

Note that T is bounded within $i_{\max}$ so that it satisfies the condition of martingale stopping theorem as shown in Lemma 3.4. Thus, we have

$$\mathbb{E}_{\mathcal{A}}[n_i F_{\mathcal{R}_1}(S_i^o)] - \text{OPT}_b(G_i) = 0. \quad (3.24)$$

Combining Equations (3.23) and (3.24) yields:

$$\mathbb{E}_{\mathcal{A}}[\mathbb{E}[I_{G_i}(S_i)]] \geq \alpha(1 - \varepsilon'_i)(1 - \delta_i n_i/b) \text{OPT}_b(G_i). \quad (3.25)$$

By replacing $\varepsilon'_i = (b\varepsilon_i - \delta_i n_i)/(b - \delta_i n_i)$, we can immediately acquire Equation (3.15), by which we complete the proof of Theorem 3.3. $\square$ $\square$

Expected Time complexity. The expected time complexity of EPIC is given in the following theorem.

Theorem 3.4. *For any $\mathcal{G}_i$, the expected time complexity of EPIC is $O((b \log n_i + \log \frac{n_i}{b\varepsilon_i})(m_i + n_i)/\varepsilon_i^2)$, where n_i and m_i are the number of nodes and edges of G_i , respectively.*

Theorem 3.4. Suppose that we set the parameters (ε, δ) as $(\varepsilon'_i, \delta_i)$ for OPIM-C. According to [17], the expected time complexity of OPIM-C is

$$\begin{aligned} O\left(\frac{m_i + n_i}{\varepsilon_i'^2} \left(b \log n_i + \log \frac{1}{\delta_i}\right)\right) \\ = O\left(\frac{m_i + n_i}{\varepsilon_i^2} \left(b \log n_i + \log \frac{1}{\varepsilon_i}\right)\right). \end{aligned}$$

On the other hand, for any given $\mathcal{R}_1$ and $\mathcal{R}_2$, if OPIM-C stops, then EPIC must also stop. This implies that EPIC always finishes earlier than OPIM-C, which completes the proof on the expected time complexity of EPIC. $\square$ $\square$

3.3.3 Expected and Worst-Case Approximation Guarantee

In this section, we derive the expected and worst-case approximation guarantees achieved by ADAPTGREEDY instantiated using EPIC. Recall that the requirement of the former is much more lenient than that of the latter, which indicates that the former is relatively easier to derive.

Expected Approximation Guarantee. Theorem 3.3 indicates ADAPTGREEDY instantiated using EPIC achieves an expected approximation guarantee of at least $\alpha(1 - \varepsilon_i)$ in the i -th iteration. Based on this fact, we have the following theorem.

Theorem 3.5. *Suppose that we instantiate ADAPTGREEDY using EPIC with the parameter ε_i for the i -th batch, then ADAPTGREEDY achieves the expected approximation ratio of $1 - e^{\alpha(\varepsilon-1)}$ where $\varepsilon = \frac{1}{r} \sum_{i=1}^r \varepsilon_i$ with expected time complexity of $O(\sum_{i=1}^r (b \log n + \log \frac{1}{\varepsilon_i})(m + n)/\varepsilon_i^2)$.*

Theorem 3.3 proves that EPIC could achieve the expected approximation of $\alpha(1 - \varepsilon_i)$ as required by ADAPTGREEDY, then Theorem 3.5 is an immediate fact based on Theorem 3.1 and Theorem 3.4.

To achieve the expected approximation ratio of $1 - e^{\alpha(\varepsilon-1)}$, instantiating ADAPTGREEDY using EPIC takes shorter running time compared with that of using the naive expected approximation guarantee of the existing IM algorithms. As discussed in Section 3.3.1.2, the intuition behind is that EPIC avoids the additional estimation error on $\text{OPT}_b(G_i)$ which is considered by all the existing IM algorithms.

Worst-Case Approximation Guarantee. Theorem 3.2 indicates that by setting $\varepsilon = \frac{1}{r} \sum_{i=1}^r \varepsilon_i + \sqrt{1/(2r) \cdot \ln(1/\delta)}$, ADAPTGREEDY achieves the worst-case approximation ratio of $1 - e^{\alpha(\varepsilon-1)}$ with a probability of at least $1 - \delta$. In this case, we could decrease the ε_i in EPIC by a constant of $\sqrt{1/(2r) \cdot \ln(1/\delta)}$ for each batch so as to maintain the same approximation ratio as that in Theorem 3.5. With this regard, we could have the following theorem.

Theorem 3.6. *Suppose that we instantiate ADAPTGREEDY using EPIC with the parameters $\varepsilon'_i = \varepsilon_i - \sqrt{1/(2r) \cdot \ln(1/\delta)}$ in each batch where $\delta \in (0, 1)$. Define $\varepsilon = \frac{1}{r} \sum_{i=1}^r \varepsilon_i$. Then, ADAPTGREEDY achieves the approximation ratio $1 - e^{\alpha(\varepsilon-1)}$ with a probability of at least $1 - \delta$ and expected time complexity of $O(\sum_{i=1}^r (b \log n + \log \frac{1}{\varepsilon'_i})(m + n)/\varepsilon_i^2)$.*

Note that Theorem 3.6 requires that $\varepsilon'_i = \varepsilon_i - \sqrt{1/(2r) \cdot \ln(1/\delta)} > 0$. This implies that only when the batch number is sufficiently large, i.e., $r > \frac{\ln(1/\delta)}{2\varepsilon_i^2}$ holds for all r batches, there is a valid instantiation of ADAPTGREEDY to achieve a predefined worst-case approximation guarantee of $1 - e^{\alpha(\varepsilon-1)}$ with probability at least $1 - \delta$.

3.4 Misclams in Previous Work [1, 2]

In this section, we revisit two of the latest work proposed to address the adaptive IM problem, i.e., our preliminary work [2] and Sun et al.'s work [1]. We aim to discuss potential issues and clarify some common misunderstandings towards this problem. Specifically, their algorithms are claimed to return a worst-case approximation guarantee with high probability. However, there exist potential theoretical issues in the analysis of the failure probability, which is elaborated as follows.

In [2] (Section 4.1), it is claimed that the overall failure probability of the i -th batch satisfies

$$\Pr \left[\varepsilon_i < \sum_{G_1, \dots, G_i} (\xi_i \cdot \Pr[\xi_i, G_1, \dots, G_{i-1}]) \right] \leq \delta_i.$$

Then, the failure probability of all r batches is bounded by a union bound of $\sum_{i=1}^r \delta_i$.

Similarly, in [1] (Theorem 5.5) it is claimed that the proposed algorithm ADAIMM achieves the worst-case approximation with $1 - \frac{1}{n^l}$ probability where l is a constant. They first prove that the seed set S selected by ADAIMM returns an approximation with at least $1 - 1/(n^l \cdot r)$ probability for each batch. Sun et al. [1] thus claim that ADAIMM achieves the approximation ratio with at least $1 - 1/n^l$ probability by union bound.

The theoretical guarantees of these two papers are based on Theorem A.10 in [9]. Through a careful examination of the proof of Theorem A.10 in [9], we find that the essence is to bound the overall approximation guarantee for each batch, i.e., $X_i \leq \varepsilon_i$, where X_i represents the average relative error over all the realizations, i.e.,

$$X_i = \frac{\mathbb{E}_{G_i \sim \mathcal{G}_i} [\xi(G_i) \cdot \text{OPT}_b(G_i)]}{\mathbb{E}_{G_i \sim \mathcal{G}_i} [\text{OPT}_b(G_i)]}.$$

For adaptive IM, X_i is a random variable which is likely to be larger than ε_i . Thus, these two papers [1, 2] attempt to bound the probability of $X_i > \varepsilon_i$ as

$$\Pr[X_i > \varepsilon_i] \leq \delta_i.$$

Table 3.2: Dataset details. ($K = 10^3, M = 10^6$)

Dataset	n	m	Type	Avg. deg
NetHEPT	15.2K	31.4K	undirected	4.18
Epinions	132K	841K	directed	13.4
DBLP	655K	1.99M	undirected	6.08
LiveJournal	4.85M	69.0M	directed	28.5
Orkut	3.07M	117M	undirected	76.2

However, as long as there exists one realization such that the seed set S_i returned in the i -th batch does not meet the approximation of $\alpha(1 - \varepsilon_i)$, i.e., $\xi(G_i) > \varepsilon_i$, it is possible that $X_i > \varepsilon_i$. On the other hand, there are exponential number $O(2^m)$ of realizations, where m is the number of edges in G . Thus, although it holds that $\Pr[\xi(G_i) > \varepsilon_i] \leq \delta_i$ under a given G_i , the probability of $X_i > \varepsilon_i$ can be as large as $O(2^m \cdot \delta_i)$ by the union bound. Therefore, it is intricate to bound X_i , which indicates that their claims on failure probability do not hold. In other words, Theorem 4 in [2] and Theorem 5.5 in [1] are invalid.

In our study, we rectify the theoretical analysis of worst-case approximation guarantee utilizing Azuma-Hoeffding inequality [27]. In particular, instead of bounding the probability of each individual $X_i > \varepsilon_i$, we directly bound the probability of $X > \varepsilon$, where $X = \frac{1}{r} \sum_{i=1}^r X_i$, as X should be concentrated to its expectation $\mathbb{E}[X]$ when r is sizable and $\mathbb{E}[X_i] \leq \varepsilon_i$ can be achieved by various non-adaptive IM algorithms.

3.5 Performance Evaluation

In this section, we evaluate the performance of our proposed approach with extensive experiments. The goal of our experiments is to measure the efficiency and effectiveness of ADAPTGREEDY using real social networks. All of our experiments are conducted on a Linux machine with an Intel Xeon 2.6GHz CPU and 64GB RAM.

3.5.1 Experimental Setting

Datasets. We use five real datasets in our experiments, i.e., NetHEPT, Epinions, DBLP, LiveJournal, and Orkut, as summarized by Table 3.2. NetHEPT is obtained from [29],

representing the academic collaboration networks in “High Energy Physics-Theory” area. The rest three datasets are available from [51]. Among them, Orkut contains millions of nodes and edges. We randomly generate 20 realizations for each dataset, each of which is generated strictly following the definitions of the influence propagation models in the literature. We then report the average performance for each algorithm on those 20 realizations.

Algorithms. We evaluate three adaptive algorithms, i.e., EPTAIM, WSTAIM and EPTAIM-N, and two state-of-the-art non-adaptive algorithms, i.e., IMM [11] and D-SSA [36]. EPTAIM is the algorithm we instantiate ADAPTGREEDY with EPIC to achieve an expected approximation ratio of $1 - e^{\alpha(\varepsilon-1)}$ where $\alpha = 1 - (1 - 1/b)^b$ and b is the batch size. WSTAIM is the same implementation as EPTAIM but with well-calibrated parameters to acquire a worst-case approximation ratio of $1 - e^{\alpha(\varepsilon-1)}$ with high probability. Recall that obtaining the worst-case approximation needs a more demanding requirement than the expected approximation, pointed out in Theorem 3.6. EPTAIM-N is ADAPTGREEDY instantiated by the existing non-adaptive algorithm OPIM-C [17] directly and is the *naive* version of EPTAIM, as introduced in Section 3.3.1.1. In addition, EPTAIM-N fixes the issues of ADAPTIM-1 [2] so that it provides the correct expected approximation ratio of $1 - e^{\alpha(\varepsilon-1)}$. By including EPTAIM-N, we could evaluate the performance improvement of EPTAIM against EPTAIM-N.

We also test two state-of-the-art non-adaptive IM algorithms (i.e., IMM [11] and D-SSA [36]) in our experiments. The purpose of using D-SSA and IMM in our experiments is to measure the influence spread increase achieved by adaptive IM algorithms compared with non-adaptive IM algorithms.

Parameter settings. We use the popular independent cascade (IC) model [4] in our experiments. Following a large body of existing work on influence maximization [4, 10–12, 17], we set the propagation probability of each edge $e = (u, v)$ to $\frac{1}{d_{\text{in}}(v)}$, where $d_{\text{in}}(v)$ is the in-degree of node v .

We set $\varepsilon = 0.5$ for all three adaptive algorithms and two non-adaptive algorithms for fair comparison and approximation errors $\varepsilon_1 = \dots = \varepsilon_r = \varepsilon$ for all adaptive algorithms. Specifically, we set the failure probability of $\delta = 1/n$ for WSTAIM, IMM, and D-SSA. Based on the performance, relative error parameter $\varepsilon = 0.5$ and failure probability $\delta =$

$1/n$ could strike a good balance between efficiency and efficacy and users are suggested to follow our proposed setting.

Recall that we need to select k nodes in r batches in adaptive IM, where $b = k/r$ nodes are selected in each batch. To see how the performance of our algorithms is affected by input parameters k, b and r , we set these parameters according to the *b-setting* and *k-setting* explained as follows. Under the *b-setting*, we fix $k = 500$ and vary b such that $b \in \{1, 2, 4, 5, 10, 500\}$. Under the *k-setting*, we fix $r = 50$ and vary k such that $k \in \{50, 100, 200, \dots, 500\}$.

3.5.2 Comparison of the Influence Spread

In this section, we study the influence spread for all tested algorithms, as shown in Figure 3.1 and Figure 3.2. In order to gain a comprehensive understanding about the efficacy of the tested algorithms, we measure their influence spreads achieved by varying the number of seed nodes k and the batch size b .

Figure 3.1 reports the influence spread obtained with 500 seed nodes selected through different numbers of batches on the five datasets. In general, the spreads acquired by the three adaptive algorithms are comparable to each other but notably larger than the spreads of the two non-adaptive algorithms, i.e., IMM and D-SSA. In particular, WSTAIM, IMM and D-SSA achieve the worst-case approximation guarantee, while WSTAIM obtains around 12% and 60% more spread than IMM and D-SSA do in average, respectively. On the one hand, this can be explained by the advantage of adaptivity over non-adaptivity that adaptive algorithms could make smarter decisions based on the feedback from previous batches. On the other hand, the considerable discrepancy on the spread of D-SSA exposes that D-SSA sacrifices its effectiveness badly for the sake of high efficiency (referring to its running time, as shown in Figure 3.3 and Figure 3.4).

Figure 3.2 shows the results of influence spreads with various number of seed nodes. We observe that (i) the spreads grow with the number of seed nodes k as expected, (ii) the three adaptive algorithms achieve similar amount of spreads under the same k and b setting, which is consistent with the results in Figure 3.1, and (iii) the percentage increase of spreads obtained by the adaptive algorithms over the spreads of IMM is around 10%

Figure 3.1: Spread vs. batch size.

Figure 3.2: Spread vs. seed size.

in average. This spread improvement is quite promising considering the large number of users in social networks. Meanwhile, it further confirms the superiority of adaptive algorithms on influence maximization.

3.5.3 Comparison of the Running Time

In this sections, we investigate the efficiency of all tested algorithms under various seed node numbers k and batch sizes b .

The settings of Figure 3.3 and Figure 3.4 follow the settings of Figure 3.1 and Figure 3.2, respectively. In particular, Figure 3.3 reports the running time with $k = 500$ and various b values under the five datasets. As can be seen, EPTAIM dominates the other two adaptive algorithms on all datasets with a non-negligible advantage. Specifically, the performance gap between EPTAIM and WSTAIM tends to enlarge along the increase of the batch size b . This expanding gap is due to that i) to maintain the same approximation ratio, WSTAIM needs to compensate for an extra factor $\sqrt{1/(2r) \cdot \ln(1/\delta)}$ on approximation error for each batch, as explained in Theorem 3.6, ii) when the seed number k is fixed, this compensation factor gets larger since the batch number $r = k/b$ gets smaller. Note that EPTAIM-N runs slower than EPTAIM for all cases on the five datasets³. When batch size $b = 1$, the efficiency gap can be up to 3 times, which demonstrates the performance improvement of our optimization in EPTAIM. One interesting observation is that WSTAIM has a slightly edge over EPTAIM-N when the batch size $b \leq 5$.

Another noticeable observation is that the running time increases along the decrease of batch size b . There are two main reasons. First, when the batch size b becomes smaller, the marginal spread drops significantly. To maintain the same approximation, more samples are generated, which incurs considerable overhead. Second, when the number of seeds k is fixed, larger b value means smaller value of r . As mentioned, RR-sets are regenerated for each batch, and thus, a fewer number of batches leads to less sampling overhead.

³Readers may note that the running time of EPTAIM-N is larger than that of ADAPTIM-1 in [2] under the same setting. To explain, note that ADAPTIM-1 algorithm reuses the RR-sets from previous batches, which, however, incurs bias on spread estimation. To address this issue, we abandon the RR-sets in previous batches and regenerate RR sets for each batch in our implementations, which incurs extra overhead.

Figure 3.3: Running time vs. batch size.

Figure 3.4: Running time vs. seed size.

Figure 3.4 plots the running time when both seed number k and batch size b vary while the batch number is fixed to $r = 50$. First of all, we can see that EPTAIM outperforms the other two adaptive algorithms with around 1.5–3 times speedup. Second, under this setting, the running time of WSTAIM is comparable with that of EPTAIM-N. Observe that the running times of the adaptive algorithms do not fluctuate as much as that in Figure 3.3 when the seed size k changes. This observation demonstrates that adaptive algorithms are more sensitive to the value of batch r than the seed size k .

Note that the two non-adaptive algorithms dominate the three adaptive algorithms in efficiency, as expected. This is because non-adaptive algorithms can be seen as special adaptive algorithms with just running in one batch, which avoids enormous sampling time.

3.6 Conclusion

We have studied the adaptive Influence Maximization (IM) problem, where the seed nodes can be selected in multiple batches to maximize their influence spread. We have proposed the first practical algorithm to address the adaptive IM problem that achieves both time efficiency and provable approximation guarantee. Specifically, our approach is based on a novel ADAPTGREEDY framework instantiated by a new non-adaptive IM algorithm EPIC, which has a provable expected approximation guarantee for non-adaptive IM. Meanwhile, we have clarified some existing misunderstandings in two recent work towards the adaptive IM problem and laid solid foundations for further study. We have also conducted extensive experiments using real social networks to evaluate the performance of our algorithms, and the experimental results strongly corroborate the superiorities and effectiveness of our approach.

Chapter 4

Adaptive Seed Minimization

As a dual problem of influence maximization, seed minimization (SM) asks for the minimum number of seed nodes to influence a required number η of users in a given social network G . In this chapter, we study seed minimization in the *adaptive* setting, where the seed nodes are selected in several batches, such that the choice of a batch may exploit information about the actual influence of the previous batches. We address this problem by developing ASTI, a novel framework tailored for the adaptive SM problem. The key idea of ASTI is to adaptively choose the seed node with the maximum expected truncated influence spread in each round of seed selection. To meet this objective, we propose a novel sampling method based on the concept of multi-root reverse reachable (mRR) sets. Building upon mRR sets, we develop TRIM, an algorithm for maximizing truncated influence spread with a provable approximation guarantee of $(1 - 1/e)(1 - \varepsilon)$. We show that instantiating ASTI using TRIM leads to strong theoretical guarantees of $\frac{(\ln \eta + 1)^2}{(1 - 1/e)(1 - \varepsilon)}$ for adaptive seed minimization with expected time complexity of $O\left(\frac{\eta \cdot (m+n)}{\varepsilon^2} \ln n\right)$. Additionally, TRIM can be extended into a batched version TRIM-B that selects a batch of b nodes in each round, so as to accelerate seed selection. ASTI instantiated by TRIM-B provides an approximation guarantee of $\frac{(\ln \eta + 1)^2}{(1 - (1 - 1/b)^b)(1 - 1/e)(1 - \varepsilon)}$ with the same time complexity as TRIM. With extensive experiments on a variety of datasets, we demonstrate the effectiveness and efficiency of ASTI over competing methods. The main contribution of this chapter has been published in [52].¹

¹It has been published as “Efficient Approximation Algorithms for Adaptive Seed Minimization” in SIGMOD 2019.

Table 4.1: Frequently used notations.

Notation	Description
η	the threshold for the targeted number of nodes to be activated
$\Gamma(S), \mathbb{E}[\Gamma(S)]$	the truncated spread of S and its expectation
η_i	the shortfall in activating η nodes in the i -th round, i.e., $\eta_i = \eta - (n - n_i)$
$I(S \mid S_{i-1})$	the marginal spread of S on top of S_{i-1} , i.e., the spread of S in G_i
$\Gamma(S \mid S_{i-1})$	the marginal truncated spread of S on top of S_{i-1} , i.e., $\Gamma(S \mid S_{i-1}) = \min\{I(S \mid S_{i-1}), \eta_i\}$
$\tilde{\Gamma}(S \mid S_{i-1})$	a binary estimator with value η_i if $S \cap R \neq \emptyset$ and 0 otherwise
$v^*, v^\diamond, v^\circ$	the optimal node maximizing $\Lambda_{\mathcal{R}}(v)$, $\mathbb{E}[\tilde{\Gamma}(v \mid S_{i-1})]$, and $\mathbb{E}[\Gamma(v \mid S_{i-1})]$, respectively
OPT_i	the optimum of $\mathbb{E}[\tilde{\Gamma}(v \mid S_{i-1})]$, i.e., $\text{OPT}_i = \max_v \mathbb{E}[\tilde{\Gamma}(v \mid S_{i-1})]$
π, π^*	a random policy, and an optimal policy

The remainder of the chapter is organized as follows. To handle the goal of seed minimization, in Section 4.1, we come up with the critical concept of *truncated influence spread*. To estimate the truncated influence spread, we propose a novel sampling technique *multi-root RR (mRR)* set. Based on mRR, we propose a general framework ASTI tailored for the adaptive SM problem in Section 4.2. Then we design TRIM to instantiate ASTI and provide the corresponding theoretical analysis. In what follows, we extend TRIM into a batched version TRIM-B that selects seed nodes in a predefined batch size in each round in Section 4.3. We then conduct a comprehensive evaluation on our algorithms in Section 4.4 and briefly summarize the work in Section 4.5.

4.1 Preliminary

This section firstly formally defines the problem of adaptive seed minimization. Then we introduce one of the fundamental concepts *truncated influence spread* built on which our algorithm is proposed. Table 4.1 summarizes the notations that are frequently used in this chapter.

4.1.1 Adaptive Seed Minimization

Given a probabilistic social network $G = (V, E)$ and a threshold $\eta \in [1, n]$, the seed minimization problem aims to select a minimum number of seed nodes to influence at

least η nodes. In the conventional “non-adaptive” setting, seed minimization requires selecting a node set S such that $\mathbb{E}[I(S)] \geq \eta$, without any knowledge of realization that would occur in the actual influence propagation process. As a consequence, the selected S may influence fewer than η nodes for some realizations or much more than η nodes for some other realizations, both of which are undesirable scenarios.

Meanwhile, the *adaptive* strategy (i.e., a recursive *select-observe-select* procedure) has been shown to be more effective than the *non-adaptive* (i.e., just select based on model) strategy in many real-world applications [44, 53, 54]. Specifically, an adaptive strategy first *selects* a node u from graph G , and then *observes* the set of nodes activated by choosing node u as a seed node. Based on this observation, the strategy would choose the next node as one that could influence as many *currently inactive* nodes as possible. This procedure is carried out in a recursive manner, until at least η active nodes are observed.

In this research, we aim to study seed selection strategies (referred to as *policies*) for *adaptive seed minimization (ASM)*, which is formally defined as follows:

Definition 4.1 (Adaptive Seed Minimization). *Given a probabilistic social graph $G = (V, E)$ and a threshold $\eta \in [1, n]$, the adaptive seed minimization problem aims to identify a policy π that minimizes the expected number of seed nodes required to achieve an influence spread of at least η on possible realizations $\phi \in \Omega$, i.e.,*

$$\min_{\pi} \mathbb{E}[|S(\pi, \phi)|] \text{ subject to } I_{\phi}(S(\pi, \phi)) \geq \eta \text{ for all } \phi,$$

where $S(\pi, \phi)$ is the seed set selected by π under realization ϕ and $\mathbb{E}[|S(\pi, \phi)|] = \sum_{\phi \in \Omega} |S(\pi, \phi)| \cdot p(\phi)$.

Note that when the propagation probability of every edge in G is 1, ASM reduces to the deterministic version of seed minimization, which is shown to be NP-hard [23]. Therefore, finding an optimal policy for ASM is also NP-hard.

4.1.2 Truncated Influence Spread

Note that, in ASM, the influence spread in excess of the threshold η has no value. Accordingly, we introduce the notion of *truncated influence spread* as follows.

Definition 4.2 (Truncated Influence Spread). *Given a seed set S and a threshold η , the truncated influence spread $\Gamma_\phi(S)$ of S under a realization ϕ is the smaller one between $I_\phi(S)$ and η , i.e.,*

$$\Gamma_\phi(S) := \min\{I_\phi(S), \eta\}. \quad (4.1)$$

Recall that ASM requires considering the influence spreads of nodes when the actual influence of some other nodes has been observed. Therefore, we also introduce the notion of *marginal* truncated influence spread as follows. Let $V_1 = V$ and $G_1 = G$. Let V_i be the subset of nodes that remain inactive after round $(i - 1)$, G_i be the subgraph of G induced by V_i . We refer to G_i as the *i*-th *residual graph*.

Let S_i be the set of nodes selected as seeds by a policy in the first i rounds. Similar to the definition of Ω , we denote Ω_i as the set of all possible realizations in the i -th round. Then, for a node set $S \subseteq V_i$, we define the marginal spread $I_\phi(S \mid S_{i-1})$ as the additional spread that S provides on top of S_{i-1} under realization $\phi \in \Omega_i$, and define truncated marginal spread $\Gamma_\phi(S \mid S_{i-1})$ accordingly, i.e.,

$$I_\phi(S \mid S_{i-1}) := I_\phi(S \cup S_{i-1}) - I_\phi(S_{i-1}), \quad (4.2)$$

$$\text{and } \Gamma_\phi(S \mid S_{i-1}) := \Gamma_\phi(S \cup S_{i-1}) - \Gamma_\phi(S_{i-1}). \quad (4.3)$$

Note that $I_\phi(S \mid S_{i-1})$ is exactly the influence spread of S in the residual graph G_i under realization ϕ .

Let $n_i = |V_i|$ be the number of nodes in G_i , i.e., $I_\phi(S_{i-1}) = n - n_i$ nodes have been activated by the end of round $i - 1$, based on the partial realization revealed so far. Define $\eta_i = \eta - (n - n_i)$. This is the amount by which the policy falls short of the target η in the beginning of round i . Before reaching the threshold η , i.e., $\Gamma_\phi(S_{i-1}) = I_\phi(S_{i-1}) < \eta$, we can rewrite $\Gamma_\phi(S \mid S_{i-1})$ as

$$\begin{aligned} \Gamma_\phi(S \mid S_{i-1}) &= \min\{I_\phi(S \cup S_{i-1}), \eta\} - I_\phi(S_{i-1}) \\ &= \min\{I_\phi(S \mid S_{i-1}), \eta_i\}. \end{aligned} \quad (4.4)$$

Then, $\Gamma_\phi(S \mid S_{i-1})$ can be easily computed in the residual graph G_i . For brevity, we define $\Gamma_\phi(v \mid S_{i-1}) := \Gamma_\phi(\{v\} \mid S_{i-1})$ for a singleton node set $\{v\}$.

Figure 4.1: A social graph and all of its possible realizations.

Algorithm 7: ASTI(G, η)

Input: an input graph G and a threshold η .

Output: a seed set S such that $\Gamma(S) = \eta$.

```

1 initialize  $S \leftarrow \emptyset$ ,  $\Gamma(S) \leftarrow 0$ ,  $i \leftarrow 1$ ;
2 repeat
3   select  $s_i$  from  $V_i$  such that  $\Delta(s_i | S_{i-1}) \geq \alpha \Delta(v | S_{i-1})$  for all  $v \in V_i$ ;
4   observe the influence of  $s_i$  in  $G_i$ ;
5   insert  $s_i$  into  $S$  and increase  $\Gamma(S)$  accordingly;
6   remove all nodes in  $G_i$  that are influenced by  $s_i$ , and denote the resulting graph as
       $G_{i+1}$ ;
7    $i \leftarrow i + 1$ ;
8 until  $\Gamma(S) \geq \eta$ ;
9 return  $S$ ;
```

Finally, we define the *expected marginal truncated spread* $\Delta(v | S_{i-1})$ as

$$\Delta(v | S_{i-1}) := \mathbb{E}_{\Phi \sim \Omega_i}[\Gamma_{\Phi}(v | S_{i-1})]. \quad (4.5)$$

In other words, the expected marginal truncated spread of a node v is defined based on the “lift” in the expected number of active nodes that v brings on top of previously selected seeds, over all realizations consistent with what has been observed in previous rounds.

4.2 ASTI: A General and Efficient Framework

4.2.1 Framework Overview

We propose a general framework, referred to as ASTI, to address the ASM problem. Algorithm 7 shows the details. Given a probabilistic social graph G and a threshold η , ASTI aims to return a seed set S such that $\Gamma(S) \geq \eta$, where $\Gamma(S)$ is the truncated influence spread of S (i.e., the smaller one of the threshold η and the number of active

nodes influenced by S). In a nutshell, ASTI iteratively (i) selects the node to maximize the *expected marginal truncated spread* (Line 3), (ii) observes the newly influenced nodes (Line 4), and then (iii) updates the corresponding information (Lines 5–7). The process stops when at least η nodes are activated (Line 8).

The key step of ASTI is *truncated influence maximization* that targets at identifying a node to maximize the expected marginal truncated spread (Line 3). If an α -approximate solution for truncated influence maximization is obtained in each round (Line 3), ASTI provides a non-trivial approximation guarantee, as shown in the following theorem.

Theorem 4.1. *Suppose π is an α -approximate greedy policy, for some $\alpha \in (0, 1]$, i.e., for any G_i and $v \in V_i$, it selects a node s_i satisfying*

$$\Delta(s_i \mid S_{i-1}) \geq \alpha \Delta(v \mid S_{i-1}). \quad (4.6)$$

Then π achieves an approximation ratio of $\frac{(\ln \eta + 1)^2}{\alpha}$ to the optimal adaptive seed minimization policy.

The proof² of Theorem 4.1 is based on adaptive submodular optimization [41]. Theorem 4.1 requires that the policy should be an α -approximate greedy one with respect to the expected marginal truncated spread $\Delta(v \mid S_{i-1})$. The challenge for designing such an α -approximate greedy policy lies in how to develop a proper sampling method for estimating the truncated influence spread.

4.2.2 Truncated Influence Maximization

According to Theorem 4.1, in order to provide the theoretical guarantee, the algorithm is supposed to identify a node whose truncated marginal spread is an α -approximation to the maximum truncated marginal spread in each round. At a first glance, it seems that we can utilize Borgs et al.’s reverse influence sampling method [13]. Unfortunately, in what follows, we show that Borgs et al.’s sampling method [13] fails to estimate the truncated influence spread accurately.

Specifically, Borgs et al. [13] propose to generate random *reverse reachable (RR)* sets for influence maximization. Compared with the Monte-Carlo simulation [4], RR-sets can

²The formal proofs of all theoretical results are given in Appendix.

dramatically accelerate the seed selection process while retaining the same approximation guarantees for influence maximization [13]. In particular, a random RR-set of G is generated by first selecting a node $v \in V$ uniformly at random, and then taking the nodes that can reach v in a random realization. Evidently, a random RR-set is a subgraph of the corresponding random realization Φ , which is generated by performing a reverse breadth first search (BFS) on Φ starting from the random node v . A random RR-set R is an unbiased spread estimator, i.e., for any seed set S ,

$$\mathbb{E}[I(S)] = n \cdot \Pr[R \cap S \neq \emptyset].$$

Unfortunately, RR-sets fail to estimate truncated influence spread accurately. Intuitively, the expectation of this estimator for truncated influence spread of S is

$$\eta \cdot \Pr[R \cap S \neq \emptyset] = \frac{\eta}{n} \cdot \mathbb{E}[I(S)] = \frac{\eta}{n} \sum_{\phi \in \Omega} I_{\phi}(S) \cdot p(\phi).$$

Recall that the true expected truncated influence spread is

$$\mathbb{E}[\Gamma(S)] = \sum_{\phi \in \Omega} \Gamma_{\phi}(S) \cdot p(\phi).$$

Obviously, for any $\phi \in \Omega$, unless $I_{\phi}(S) = n$,

$$\frac{\eta}{n} \cdot I_{\phi}(S) < \min\{I_{\phi}(S), \eta\} = \Gamma_{\phi}(S).$$

Specifically, consider the case that $I_{\phi}(S) \leq \eta$ for all ϕ . Then, this estimator is biased with a discount η/n , which is extremely inaccurate when $\eta \ll n$. In practice, η is likely to be a fraction of n , since even a set of ten thousand seed nodes has been found to influence less than half population on many datasets [36]. These facts indicate that RR-sets are highly biased for estimating truncated influence spread. As a consequence, the state-of-the-art algorithms [10–12, 17, 28] for influence maximization that utilize RR-sets [13] cannot provide theoretical guarantees for truncated influence maximization. In turn, this means that these algorithms cannot be fashioned to solve ASM with approximation guarantees. To address this issue, we propose a novel sampling approach that generates *multi-root reverse reachable (mRR)* sets which can estimate the truncated influence spread efficiently

and effectively. The algorithm utilizing mRR-sets is referred to as TRIM³. We rigorously show that TRIM can provide strong theoretical guarantees for truncated influence maximization and thus ASTI instantiated with TRIM is guaranteed to approximate ASM within a constant ratio.

4.2.3 Multi-Root Reverse Reachable Set

If we generate n correlated RR-sets such that (i) they start from n distinct nodes, and (ii) the materialization of each edge is consistent in all the RR-sets, then merging these RR-sets (with duplicates removed) as well as the edge statuses forms a realization sample. Based on this observation, if we generate k ($k < n$) correlated RR-sets using the same rule, then merging them as a k -root RR-set is likely to estimate the truncated influence spread more accurately compared against a vanilla RR-set. To explain how *multi-root reverse reachable* (mRR) set works, we first introduce its definition.

Definition 4.3 (Random mRR-set). *Let Φ be a random realization of G sampled from the realization space and K be a size- k node set selected uniformly at random from V . A random mRR-set is the set of nodes in Φ that can reach K . (That is, for each node v in the mRR-set, there is a directed path in Φ from v to some node in K .)*

By definition, the key difference between an mRR-set and an RR-set is that the former has multiple roots whereas the latter has one single root only. Similar to the generation of RR-sets, a random mRR-set can be generated by:

1. Choose a set of k nodes $K \subseteq V$ uniformly at random;
2. Perform a stochastic reverse breadth first search (BFS) that starts from K and follows the incoming edges of each node. Insert into R all nodes that are traversed during the stochastic BFS.

A natural question is how to decide the size of k for truncated spread estimation? The setting of k yields a tradeoff between efficiency and accuracy in that a larger k provides more accurate estimation but takes more computational resources. Through

³TRuncated Influence Maximization.

the aforementioned analysis of RR-set, we find that the high-efficiency of RR-set comes from its “binary” property. In particular, a random RR-set R estimates the influence spread of any node set S as n if $R \cap S \neq \emptyset$, and as 0 otherwise. To avoid maintaining the edge statuses, our mRR-set estimator shall retain this binary property. That is, it estimates the truncated influence spread of S as η if and only if S intersects this mRR-set, and as 0 otherwise. For a given k -RR-set R , if a node $v \in R$, then v can reach at least one of the k starting nodes. Then, its influence spread is estimated to be at least n/k and thus its estimated truncated influence spread is at least $\min\{n/k, \eta\}$. By setting $n/k \geq \eta$, the estimated truncated influence spread is η .

On the other hand, to improve the accuracy, k should be set as large as possible. So we choose $k = n/\eta$. However, n/η is not an integer in general. To address this issue, we adopt a randomized rounding approach. To generate a mRR-set, we randomly choose a set K of nodes such that its size k equals $\lfloor \frac{n}{\eta} \rfloor + 1$ with probability $\frac{n}{\eta} - \lfloor \frac{n}{\eta} \rfloor$, and equals $\lfloor \frac{n}{\eta} \rfloor$ otherwise. Then, the expectation of k is n/η . However, we note that when $k = \lfloor \frac{n}{\eta} \rfloor + 1$, the possible value of the estimated truncated influence spread is no longer binary (i.e., 0 or η). To address such a new challenge, we define an estimator $\tilde{\Gamma}(S)$ as $\tilde{\Gamma}(S) = \eta$ if and only if $S \cap R \neq \emptyset$, and $\tilde{\Gamma}(S) = 0$ otherwise. At the first glance, it seems that the relationship between $\mathbb{E}[\Gamma(S)]$ and $\mathbb{E}[\tilde{\Gamma}(S)]$ is unclear. Fortunately, the following theorem shows that under the above setting of k such that $\mathbb{E}[k] = n/\eta$, the ratio of $\mathbb{E}[\tilde{\Gamma}(S)]$ and $\mathbb{E}[\Gamma(S)]$ is in the range of $[1 - 1/e, 1]$.

Theorem 4.2. *Let $k^\perp = \lfloor \frac{n}{\eta} \rfloor$ and $r = n/\eta - k^\perp$ be the integer and fractional part of n/η , respectively. For any mRR-set, if we sample k nodes such that $k = k^\perp + 1$ with probability r and $k = k^\perp$ otherwise, then*

$$(1 - 1/e)\mathbb{E}[\Gamma(S)] \leq \mathbb{E}[\tilde{\Gamma}(S)] \leq \mathbb{E}[\Gamma(S)]. \quad (4.7)$$

Theorem 4.2 states that $\tilde{\Gamma}$ is a biased but sufficiently accurate estimator of the expected truncated influence spread $\mathbb{E}[\Gamma(S)]$. In fact, this estimator also works for any residual graph G_i . Specifically, let $\tilde{\Gamma}(S \mid S_{i-1})$ be the estimated truncated spread of S in G_i with respect to η_i , the lowered target corresponding to graph G_i . Recall that $\eta_i = \eta - (n - n_i)$. We have the following corollary.

Corollary 4.1. *In the residual graph G_i , let $k^\perp = \lfloor \frac{n_i}{\eta_i} \rfloor$ and $r = n_i/\eta_i - k^\perp$ be the integer and fractional part of n_i/η_i , respectively. For each mRR-set, if we sample k nodes such that $k = k^\perp + 1$ with probability r and $k = k^\perp$ otherwise, then*

$$(1 - 1/e)\mathbb{E}[\Gamma(S \mid S_{i-1})] \leq \mathbb{E}[\tilde{\Gamma}(S \mid S_{i-1})] \leq \mathbb{E}[\Gamma(S \mid S_{i-1})]. \quad (4.8)$$

Furthermore, for any two sets $S, S' \subseteq V_i$, it holds that

$$\frac{\mathbb{E}[\Gamma(S \mid S_{i-1})]}{\mathbb{E}[\Gamma(S' \mid S_{i-1})]} \geq (1 - 1/e) \frac{\mathbb{E}[\tilde{\Gamma}(S \mid S_{i-1})]}{\mathbb{E}[\tilde{\Gamma}(S' \mid S_{i-1})]}. \quad (4.9)$$

Now, we can construct a $(1 - 1/e)(1 - \varepsilon)$ -approximate greedy policy using the estimator $\tilde{\Gamma}$ built upon mRR-sets.

Remark. It is worth pointing out that our randomized rounding approach for choosing k is critical for achieving the above approximation bound. Specifically, if we fix k to be $\lfloor \frac{n}{\eta} \rfloor$, following the proof methodology of Theorem 4.2, we may derive that the ratio of $\mathbb{E}[\tilde{\Gamma}(S)]$ to $\mathbb{E}[\Gamma(S)]$ will be in the range of $[1 - 1/\sqrt{e}, 1]$. On the other hand, if we fix k to be $\lfloor \frac{n}{\eta} \rfloor + 1$, the ratio of $\mathbb{E}[\tilde{\Gamma}(S)]$ to $\mathbb{E}[\Gamma(S)]$ will be in the range of $[1 - 1/e, 2]$. Both settings yield much coarser bounds than our setting that uses a smart randomized rounding approach.

4.2.4 The Design of TRIM

Algorithm 8 presents the details of TRIM that can return a $(1 - 1/e)(1 - \varepsilon)$ -approximate solution for truncated influence maximization for any input graph G_i and error threshold ε . TRIM is similar in spirit to OPIM-C which is the state-of-the-art algorithm for influence maximization [17]. Specifically, OPIM-C uses two disjoint groups of random RR-sets, among which one group is used to derive the solution and the other is used to verify its quality. We customize TRIM by utilizing one group of mRR-sets, which would be more efficient for selecting a singleton seed set as pointed out in [28]. In a nutshell, TRIM starts from a small number of mRR-sets and iteratively increases the mRR-set number until a satisfactory solution is identified. Next, we discuss the details of TRIM.

In the mRR-set sampling stage (Lines 6 and 12), each mRR-set is started from a random set K of nodes whose size k is an independent random number. Recall that

Algorithm 8: TRIM(G_i, ε)

Input: Graph G_i and error threshold ε .

Output: A $(1 - 1/e)(1 - \varepsilon)$ -approximate solution v^* for truncated influence maximization.

```

1  $\delta \leftarrow \varepsilon / (100(1 - 1/e)(1 - \varepsilon)\eta_i)$ , and  $\hat{\varepsilon} \leftarrow 99\varepsilon / (100 - \varepsilon)$ ;
2  $\theta_{\max} \leftarrow 2n_i (\sqrt{\ln(6/\delta)} + \sqrt{\ln n_i + \ln(6/\delta)})^2 \cdot \hat{\varepsilon}^{-2}$ ;
3  $\theta_o \leftarrow \theta_{\max} \cdot \hat{\varepsilon}^2 / n_i$ ;
4  $T \leftarrow \lceil \log_2 \frac{\theta_{\max}}{\theta_o} \rceil + 1$ ;
5  $a_1 \leftarrow \ln(3T/\delta) + \ln n_i$ , and  $a_2 \leftarrow \ln(3T/\delta)$ ;
6 generate a set  $\mathcal{R}$  of  $\theta_o$  random mRR-sets;
7 for  $t \leftarrow 1$  to  $T$  do
8   find  $v^* \leftarrow \arg \max_{v \in V_i} \Lambda_{\mathcal{R}}(v)$ ;
9    $\Lambda^l(v^*) \leftarrow (\sqrt{\Lambda_{\mathcal{R}}(v^*)} + 2a_1/9 - \sqrt{a_1/2})^2 - a_1/18$ ;
10   $\Lambda^u(v^o) \leftarrow (\sqrt{\Lambda_{\mathcal{R}}(v^*)} + a_2/2 + \sqrt{a_2/2})^2$ ;
11  if  $\frac{\Lambda^l(v^*)}{\Lambda^u(v^o)} \geq 1 - \hat{\varepsilon}$  or  $t = T$  then return  $v^*$ ;
12  double the size of  $\mathcal{R}$ ;
    
```

k is $\lfloor \frac{n_i}{\eta_i} \rfloor + 1$ with probability $\frac{n_i}{\eta_i} - \lfloor \frac{n_i}{\eta_i} \rfloor$ and $\lfloor \frac{n_i}{\eta_i} \rfloor$ otherwise. Given a set $\mathcal{R}$ of random mRR-sets, we say that a node v *covers* a mRR-set $R \in \mathcal{R}$ if $v \in R$, and we define the *coverage* of v in $\mathcal{R}$, denoted as $\Lambda_{\mathcal{R}}(v)$, as the number of mRR-sets in $\mathcal{R}$ that are covered by v . Based on the mRR-sets generated, TRIM identifies the node $v^* \in V_i$ that covers the largest number of mRR-sets in $\mathcal{R}$ (Line 8). Let v^o be the optimal node such that $\Delta(v^o \mid S_{i-1}) = \max_{v \in V_i} \Delta(v \mid S_{i-1})$. Then, $\Lambda_{\mathcal{R}}(v^o)$ is bounded by $\Lambda_{\mathcal{R}}(v^*)$. According to Lemma A.5 in Appendix, with high probability, $\Lambda^l(v^*)$ (Line 9) is a lower bound on the expected coverage of v^* in $\mathcal{R}$, which indicates that

$$\frac{\eta_i \Lambda^l(v^*)}{|\mathcal{R}|} \leq \mathbb{E}[\tilde{\Gamma}(v^* \mid S_{i-1})]. \quad (4.10)$$

Similarly, with high probability, $\Lambda^u(v^o)$ (Line 10) is an upper bound on the expected coverage of v^o in $\mathcal{R}$. Thus,

$$\frac{\eta_i \Lambda^u(v^o)}{|\mathcal{R}|} \geq \mathbb{E}[\tilde{\Gamma}(v^o \mid S_{i-1})]. \quad (4.11)$$

In addition, by Equation (4.9) in Corollary 4.1, we know that

$$\frac{\Delta(v^* \mid S_{i-1})}{\Delta(v^o \mid S_{i-1})} \geq (1 - 1/e) \frac{\mathbb{E}[\tilde{\Gamma}(v^* \mid S_{i-1})]}{\mathbb{E}[\tilde{\Gamma}(v^o \mid S_{i-1})]}. \quad (4.12)$$

Combining Equations (4.10)–(4.12), we can derive a quantitative relationship between $\Delta(v^* | S_{i-1})$ and $\Delta(v^\circ | S_{i-1})$ such that with high probability

$$\Delta(v^* | S_{i-1}) \geq \frac{\Lambda^l(v^*)}{\Lambda^u(v^\circ)} \cdot (1 - 1/e) \cdot \Delta(v^\circ | S_{i-1}). \quad (4.13)$$

Therefore, the final guarantee is $(1 - 1/e)\Lambda^l(v^*)/\Lambda^u(v^\circ)$. Note that in our stopping condition of $\Lambda^l(v^*)/\Lambda^u(v^\circ) \geq 1 - \hat{\varepsilon}$ (Line 11), we use $\hat{\varepsilon}$ (defined in Line 1) to correct the error on Equations (4.10) and (4.11) (with low failure probability). This proves the $(1 - 1/e)(1 - \varepsilon)$ approximation ratio of $\Delta(v^* | S_{i-1})$.

4.2.5 Theoretical Analysis

Before we proceed to the theoretical analysis, we first present the hardness of ASM.

Lemma 4.1. *Given a probabilistic social network $G = (V, E)$ with $|V| = n$ and a threshold $\eta \in [1, n]$, for any $\xi > 0$, adaptive seed minimization cannot be approximated within a ratio of $(1 - \xi) \ln \eta$ in polynomial time unless $\text{NP} \subseteq \text{DTIME}(n^{O(\log \log n)})$.*

Approximation Guarantee. Theorem 4.1 indicates that any α -approximation greedy policy π could achieve an approximation ratio of $\frac{(\ln \eta + 1)^2}{\alpha}$. We examine the potential of TRIM to serve the role of such a policy. To cope with the randomness of seed selection algorithms (due to sampling), we use the notion of *expected approximation guarantee*, which considers the *average case*. We first obtain the approximation ratio of TRIM for each round of seed selection.

Lemma 4.2. *For the i -th round of seed selection in G_i , TRIM returns a $(1 - 1/e)(1 - \varepsilon)$ -approximate solution to the optimum.⁴*

Combining Theorem 4.1 and Lemma 4.2, we obtain the approximation guarantee of ASTI.

Theorem 4.3. *ASTI with the instantiation of TRIM achieves an expected approximation ratio of $\frac{(\ln \eta + 1)^2}{(1 - 1/e)(1 - \varepsilon)}$.*

⁴Here, α -approximation indicates that $\mathbb{E}[\frac{1}{\Delta(v^* | S_{i-1})}] \leq \frac{1}{\alpha} \cdot \frac{1}{\Delta(v^\circ | S_{i-1})}$, which is required by Theorem 4.1 for a randomized algorithm through a detailed check of the proof of Theorem 40 in [41].

Time Complexity. The time complexity of TRIM is dominated by the procedure for generating mRR-sets. Intuitively, this is based on (i) how much time is used for generating a random mRR-set, and (ii) how many mRR-sets are generated. In what follows, we show their relationship. In particular, for the i -th round of seed selection in G_i , let OPT_i (resp. v°) be the optimum (resp. optimal node) of $\mathbb{E}[\tilde{\Gamma}(v \mid S_{i-1})]$, i.e., $\text{OPT}_i = \mathbb{E}[\tilde{\Gamma}(v^\circ \mid S_{i-1})] = \max_v \mathbb{E}[\tilde{\Gamma}(v \mid S_{i-1})]$. (Note that v^* maximizes $\Lambda_{\mathcal{R}}(v)$, v° maximizes $\mathbb{E}[\tilde{\Gamma}(v \mid S_{i-1})]$, and v° maximizes $\Delta(v \mid S_{i-1})$.) We first show the expected time used for generating a random mRR-set in the following lemma.

Lemma 4.3. *For the i -th round of seed selection in G_i , the expected time complexity for generating a random mRR-set is $O(\frac{\text{OPT}_i}{\eta_i} m_i)$.*

Now, we present the following lemma that gives the expected number of mRR-sets generated by TRIM. The proof is similar to that of OPIM-C [17].

Lemma 4.4. *For the i -th round of seed selection in G_i , the expected number of mRR-sets TRIM generated is $O(\frac{\eta_i \ln n_i}{\varepsilon^2 \text{OPT}_i})$.⁵*

Finally, we provide the expected time complexity of TRIM in the following lemma.

Lemma 4.5. *For the i -th round of seed selection in G_i , TRIM achieves an expected time complexity of $O(\frac{m_i + n_i}{\varepsilon^2} \ln n_i)$.*

At the first glance, the expected time complexity of TRIM is counterintuitive. In particular, the expected root size of n_i/η_i in the i -th round is increasing with i . It seems that the time complexity of TRIM is more likely to increase with i . However, Lemma 4.5 just tells us the opposite. This is due to either the residual graph G_i being reduced significantly (Lemma 4.3) or the mRR-set size being reduced considerably (Lemma 4.4). Overall, the time complexity of TRIM in each round can be independent of the number of initially selected nodes. There are at most η rounds in total, we can derive the expected time complexity of ASTI instantiated with TRIM.

Theorem 4.4. *ASTI with the instantiation of TRIM has an expected time complexity of $O(\frac{\eta(m+n)}{\varepsilon^2} \ln n)$.*

⁵In general, it is $O(\frac{\eta_i \ln(n_i/\varepsilon)}{\varepsilon^2 \text{OPT}_i})$. Here, we assume that $\varepsilon \in \Omega(\frac{1}{\text{poly}(n_i)})$.

Algorithm 9: TRIM-B(G_i, ε, b)

Input: Graph G_i , error threshold ε , and batch size b .

Output: A $\rho_b(1 - 1/e)(1 - \varepsilon)$ -approximate solution S_b with size- b for truncated influence maximization, where $\rho_b = 1 - (1 - 1/b)^b$.

```

1  $\delta \leftarrow \varepsilon / (100(1 - 1/e)(1 - \varepsilon)\eta_i)$ , and  $\hat{\varepsilon} \leftarrow 99\varepsilon / (100 - \varepsilon)$ ;
2  $\theta_{\max} \leftarrow 2n_i \left( \sqrt{\ln \frac{6}{\delta}} + \sqrt{(\ln \binom{n_i}{b} + \ln \frac{6}{\delta}) / \rho_b} \right)^2 / (b\hat{\varepsilon}^2)$ ;
3  $\theta_o \leftarrow \theta_{\max} \cdot b\hat{\varepsilon}^2 / n_i$ ;
4  $T \leftarrow \lceil \log_2 \frac{\theta_{\max}}{\theta_o} \rceil + 1$ ;
5  $a_1 \leftarrow \ln(3T/\delta) + \ln \binom{n_i}{b}$ , and  $a_2 \leftarrow \ln(3T/\delta)$ ;
6 generate a set  $\mathcal{R}$  of  $\theta_o$  random mRR-sets;
7 for  $t \leftarrow 1$  to  $T$  do
8   find  $S_b \leftarrow \text{Greedy}(\mathcal{R})$ ;
9    $\Lambda^l(S_b) \leftarrow (\sqrt{\Lambda_{\mathcal{R}}(S_b)} + 2a_1/9 - \sqrt{a_1/2})^2 - a_1/18$ ;
10   $\Lambda^u(S_b^o) \leftarrow (\sqrt{\Lambda_{\mathcal{R}}(S_b)/\rho_b} + a_2/2 + \sqrt{a_2/2})^2$ ;
11  if  $\frac{\Lambda^l(S_b)}{\Lambda^u(S_b^o)} \geq \rho_b(1 - \hat{\varepsilon})$  or  $t = T$  then return  $S_b$ ;
12  double the size of  $\mathcal{R}$ ;
    
```

4.3 Extensions of Batch Size

TRIM selects one node in each round until at least η users are influenced. Therefore, the seed selection phase in ASTI instantiated by TRIM can be quite time consuming due to that the marginal (truncated) spread of a singleton node set is potentially small which may (i) involve in many rounds to achieve the target η , and (ii) generate a large number of mRR-sets for constructing an α -approximate solution in each round. To mitigate the enormous overhead, we propose a batched version of TRIM, referred to as TRIM-B⁶ algorithm, to accelerate the node selection process of ASTI.

4.3.1 Batched Version of TRIM

Algorithm 9 shows the details of the TRIM-B algorithm. TRIM-B generalizes TRIM by selecting a fixed number of b seeds in each round, where b is an input parameter to determine the batch size. Specifically, TRIM-B first generates a small number of random mRR-sets and then uses a greedy algorithm for maximum coverage [35] to identify a size- b seed set S_b to cover mRR-sets with an approximation guarantee of $\rho_b = 1 - (1 - 1/b)^b$ (Line 8). If S_b meets the condition (Line 11), TRIM-B terminates; otherwise, the number

⁶TRuncated Influence Maximization in the Batched model.

of mRR-sets is doubled until a qualified S_b is derived. Consequently, the approximation ratio of TRIM-B is $\rho_b(1 - 1/e)(1 - \varepsilon)$. Note that when the batch size b is 1, TRIM-B degenerates to TRIM.

The major differences in the design between TRIM-B and TRIM are as follows. First, in TRIM-B, the definitions of variables $\theta_{\max}$ and θ_o are involved with ρ_b and b for generalization, as shown in Line 2 and Line 3, respectively. Second, to obtain the upper bound on the coverage of the optimal solution S_b^o in $\mathcal{R}$, the coverage of $\Lambda_{\mathcal{R}}(S_b)$ is divided by ρ_b (Line 10). Third, the ratio in the stop condition is updated to be $\rho_b(1 - \hat{\varepsilon})$ (Line 11).

4.3.2 Theoretical Analysis

The theoretical analysis of TRIM-B can be obtained by generalizing the properties of TRIM.

Approximation Guarantee. To establish the overall approximation guarantee, we first analyze the approximation ratio of TRIM-B in each round of seed selection.

Lemma 4.6. *For the i -th round of seed selection in G_i , TRIM-B returns a $\rho_b(1 - 1/e)(1 - \varepsilon)$ -approximate solution, where $\rho_b = 1 - (1 - 1/b)^b$.*

Combining Theorem 4.1 and Lemma 4.6, we obtain the approximation guarantee of TRIM-B.

Theorem 4.5. *ASTI with the instantiation of TRIM-B achieves an expected approximation ratio of $\frac{(\ln \eta + 1)^2}{\rho_b(1 - 1/e)(1 - \varepsilon)}$.*

Remark. Note that there exists a gap between the optimal policy in the sequential model and the optimal policy in the batched model, which is known as the *adaptivity gap* [41]. Adaptivity gap quantifies the performance difference between the optimal adaptive policy and the optimal non-adaptive policy. To explain, a size- b seed set is selected as a batch ($b \geq 1$) in TRIM-B without observing the realization of any seed therein. This selection is a non-adaptive process compared to that of $b = 1$ in TRIM. As a consequence, there exists an adaptivity gap between the two algorithms if the batch size $b > 1$. However, to the best of our knowledge, this adaptivity gap remains unknown in viral marketing

applications, which makes it hard to quantify the difference between the optimal policy in the sequential model and that in the batched model. Meanwhile, the existing bound of adaptivity gap of $(1 - 1/e)$ in [53] is not applicable to adaptive seed minimization. It holds only if the nodes in social graph G are *independent*, which, however, is not true.

Time Complexity. The time complexity of TRIM-B depends on three factors: (i) the time for generating a random mRR-set, (ii) the number of mRR-sets generated, and (iii) the time to derive a size- b seed set. The expected time used for generating a random mRR-set is given in Lemma 4.3. We now show the number of mRR-sets generated.

Lemma 4.7. *For the i -th round of seed selection in G_i , the expected number of mRR-sets TRIM-B generates is $O\left(\frac{\eta_i \ln \binom{n_i}{b}}{\varepsilon^2 \text{OPT}_{b,i}}\right)$, where $\text{OPT}_{b,i}$ denotes the maximum expected truncated spread among all the size- b seed sets in G_i .*

On the other hand, the greedy algorithm for identifying the size- b seed set runs in time linear to the total size of its input [35], i.e., $\sum_{R \in \mathcal{R}} |R|$. Meanwhile, the total number of mRR-sets examined in all the iterations is within twice of that in the last iteration. According to Wald's equation [55], the expected time complexity of the greedy procedure is $O(\mathbb{E}[|\mathcal{R}|] \cdot \mathbb{E}[|R|])$, which is dominated by that for generating mRR-sets. Consequently, by Lemma 4.3 and Lemma 4.7, the expected time used in the i -th round of TRIM-B is $O\left(\frac{b(m_i + n_i) \ln n_i}{\varepsilon^2}\right)$. There are at most $O(\eta/b)$ rounds in total. Based on the analysis above, the expected time complexity of TRIM-B is given in the following theorem.

Theorem 4.6. *ASTI with the instantiation of TRIM-B achieves an expected time complexity of $O\left(\frac{\eta \cdot (m+n)}{\varepsilon^2} \ln n\right)$.*

4.4 Experiments

This section evaluates the performance of the proposed algorithms against the state of the art. All the experiments are conducted on a Linux machine with an Intel Xeon 2.6GHz CPU and 64GB RAM. For fair comparison, we first randomly generate 20 possible realizations for each dataset, each of which is generated strictly following the definitions of the influence propagation models in the literature. We then measure the performance of each algorithm on those 20 realizations and report the average performance.

Figure 4.2: Degree distribution of tested datasets.

Table 4.2: Dataset details. ($K = 10^3, M = 10^6$)

Dataset	n	m	Type	Avg. deg.	LWCC size
NetHEPT	15.2K	31.4K	undirected	4.18	6.80K
Epinions	132K	841K	directed	13.4	119K
Youtube	1.13M	2.99M	undirected	5.29	1.13M
LiveJournal	4.85M	69.0M	directed	28.5	4.84M

4.4.1 Experimental Setting

Datasets. The experiments are conducted on four datasets, i.e., NetHEPT, Epinions, Youtube, and LiveJournal. NetHEPT [29] represents the academic collaboration networks of "High Energy Physics - Theory" area. The rest of the three are real-life social networks from [51]. Table 4.2 summarizes the details of the four datasets. Note that an undirected edge is transformed into two directed edges. There does exist any isolated node in the four tested datasets. Furthermore, the number of nodes in the largest weakly connected component (LWCC) indicates that nodes are highly interconnected, especially for the three social networks. As shown in Figure 4.2, all the four datasets have a power law degree distribution. The largest dataset that has been used for adaptive seed minimization in the literature contains 75k nodes and 500k edges [22], which is far smaller than LiveJournal. To the best of our knowledge, LiveJournal with millions of nodes and edges is the largest dataset ever tested in adaptive seed minimization experiments.

Algorithms. We evaluate six algorithms: ASTI, ASTI-2, ASTI-4, ASTI-8, ADAPT-IM and ATEUC [24]. ASTI- b is ASTI instantiated by TRIM-B with the batch sizes of b . (Note that ASTI is the version with a batch size of 1.) ADAPT-IM is modified from

the ADAPTIM-1 method proposed in [2] for the adaptive influence maximization problem. It iteratively runs a non-adaptive influence maximization algorithm (i.e., EPIC [2]) to select the node that maximizes the expected marginal influence spread on the residual graphs, until the desired threshold is reached. ADAPTIM differs from our ASTI algorithm in that it greedily selects the node to maximize the influence spread instead of the truncated influence spread. The batch size of ADAPTIM is set to 1 by default. As introduced in Section 2.3.1, ATEUC is the state of the art for the non-adaptive seed minimization problem. By comparing ASTI with ATEUC, we aim to prove the advantage of adaptivity over non-adaptivity in terms of the effectiveness. Meanwhile, three batched algorithms, i.e., ASTI-2, ASTI-4, ASTI-8, are compared with both ASTI and ATEUC to study how the batch size would affect the efficiency and effectiveness. For ADAPTIM, we obtain the source code of ADAPTIM-1 from the authors [2] with some necessary modifications (e.g., stop condition). For the other five algorithms, we implement them in C++ strictly following the algorithm description and compile them with the same optimization options.

Parameter Settings. In our experiments, all the algorithms are tested under both the *Independent Cascade (IC)* model and the *Linear Threshold (LT)* model. Following the common setting in the literature [10, 33], we set the approximation parameter $\varepsilon = 0.5$ for the five adaptive algorithms. For those parameters in ATEUC, we use the values recommended in [24]. Following a large body of existing work [4, 10–12, 17], for each dataset, we set the edge probability $p(\langle u, v \rangle) = \frac{1}{\text{indeg}_v}$ where indeg_v is the in-degree of node v .

The performance metrics measured include the number of seeds selected and the corresponding running time. To better understand the performance of the algorithms, we design the *large η setting* of the threshold for NetHEPT, Epinions, and Youtube, i.e., $\frac{\eta}{n} = \{0.01, 0.05, 0.1, 0.15, 0.2\}$, where n is the number of nodes in the social network. Observing that around $2K$ nodes are required on LiveJournal under the *large η setting* which is not convenient for exhibition, we thus use a tailored *small η setting*, i.e., $\frac{\eta}{n} = \{0.01, 0.02, 0.03, 0.04, 0.05\}$ for LiveJournal.

Table 4.3: Improvement ratio of ASTI over ATEUC

	η/n	0.01	0.05	0.1	0.15	0.2
IC Model	NetHEPT	N/A	40.8%	43.8%	43.0%	43.7%
	Epinions	N/A	N/A	50.7%	N/A	65.7%
	Youtube	0.0%	24.3%	N/A	37.5%	41.7%
	LiveJournal	N/A	43.0%	34.9%	N/A	33.0%
LT Model	NetHEPT	N/A	N/A	N/A	44.3%	47.5%
	Epinions	N/A	N/A	N/A	N/A	N/A
	Youtube	0.0%	39.5%	54.1%	N/A	47.9%
	LiveJournal	N/A	N/A	N/A	N/A	N/A

N/A: ATEUC does not meet the threshold for some realizations.

4.4.2 Results under the IC model

Seed Size vs. Threshold. Figure 4.3 reports the number of seeds selected by the six algorithms for different thresholds η under the IC model. As can be seen, ASTI selects far fewer seed nodes than ATEUC does, especially when the threshold η becomes larger. In general, ATEUC selects around 30%–40% more nodes than ASTI does on all the four datasets. In particular, with a threshold $\eta/n = 0.2$ on dataset Epinions, ASTI selects 116.95 seed nodes on average while ATEUC needs 193.8 seed nodes (i.e., 65.7% more nodes). For the sake of clarity, Table 4.3 shows the exact improvement ratio of ASTI over ATEUC on the number of seed nodes for the corresponding five thresholds under both the IC and LT model. Note that there exist many points (indicated by N/A) where the actual number of nodes activated by the seed set returned by ATEUC does not reach the required threshold under some realizations. This is because ATEUC selects a node set S such that $\mathbb{E}[I(S)] \geq \eta$ but may influence fewer than η nodes under some realizations, whereas our adaptive algorithms always ensure that at least η nodes are influenced by the returned node set under every realization. We shall explore this in more detail in Section 4.4.4. These facts support the superiority of adaptive algorithms over non-adaptive algorithms. We also observe that the number of nodes selected by ADAPTIM is close to that of ASTI, which indicates that ADAPTIM is *empirically* effective in seed minimization. However, it does not provide any approximation guarantees in terms of the number of nodes selected. Another interesting observation is that ASTI-2, ASTI-4, and ASTI-8 slightly increase the number of seed nodes selected compared with ASTI

Figure 4.3: Number of seed nodes vs. threshold under the IC model.

and still select nodes far less than ATEUC does for most of the cases. This confirms that adaptive algorithms by utilizing the information of partial realizations are more effective than non-adaptive algorithms.

Running Time vs. Threshold. Figure 4.4 presents the results of running time against the threshold under the IC model. As the results show, ATEUC runs faster than the other five adaptive algorithms on the four datasets when the threshold η is large. The main reason is that adaptive algorithms involve multiple rounds of seed selection whereas only one round is required for non-adaptive algorithms. Observe that the running time of ATEUC generally decreases with the increase of the threshold η , unlike the results of the five adaptive algorithms. The reason lies in the design of ATEUC. Specifically,

Figure 4.4: Running time vs. threshold under the IC model.

ATEUC selects two seed set candidates S_u and S_l , which are taken as the upper bound and lower bound on the number of seed nodes in the optimal solution. Only when the condition $|S_u| \leq 2|S_l|$ is satisfied, the candidate set S_u is returned as the solution; otherwise ATEUC will continue to refine S_u and S_l [24]. The larger the threshold, the more seed nodes are required, and the more easily this stop condition is met, which explains the unique running time pattern of ATEUC. We also observe that ADAPTIM runs around 10–20 times slower than ASTI for all cases. Particularly, ADAPTIM cannot finish within 72 hours when $\eta/n = 0.05$ under the IC model on the LiveJournal dataset (see Figure 4.4(d)). This demonstrates that ADAPTIM is significantly inferior to ASTI in terms of computational overheads. The reason behind this is that ASTI selects the

node to maximize the expected marginal *truncated* spread, while ADAPTIM attempts to maximize the expected marginal influence spread. Specifically, recall that the expected number of mRR-sets generated by ASTI is proportional to η_i/OPT_i . Meanwhile, the expected number of RR-sets generated by ADAPTIM is proportional to n_i/OPT'_i , where OPT'_i is the maximum expected marginal influence spread in the i -th round of seed selection in G_i . For the last few rounds of seed selection, we have $\text{OPT}'_i \approx \text{OPT}_i \approx \eta_i \ll n_i$, which indicates that the number of mRR-sets generated by ASTI is much smaller than the number of RR-sets generated by ADAPTIM. Consequently, ASTI runs remarkably faster than ADAPTIM. As such, ASTI is more preferable than ADAPTIM, as the former provides significantly better efficiency and approximation guarantees than the latter, while offering similar empirical effectiveness. Note that the batched algorithms, i.e., ASTI-2, ASTI-4, and ASTI-8, reduce the running time significantly, to around 30%, 10%, and 5% of ASTI, which makes them quite competitive with ATEUC in terms of the efficiency, not to mention ADAPTIM. In addition, as explained earlier, the terminal condition $|S_u| \leq 2|S_l|$ in ATEUC is easier satisfied when the threshold η is larger, and hence, ATEUC runs faster along with the increase of η . On the other hand, the running times of the adaptive algorithms increase with η . Therefore, ASTI-4 and ASTI-8 outperform ATEUC on datasets Epinions and Youtube when η is relatively small, but when the threshold $\eta/n = 0.2$, the running times of all three algorithms become similar, as shown in Figures 4.4(b) and 4.4(c). Recall that ASTI-8 selects far fewer seed nodes than ATEUC does. Therefore, ASTI-8 strikes a good balance between efficiency and effectiveness in the current setting. We also observe that the running time of ASTI-8 fluctuates from $\eta/n = 0.01$ to $\eta/n = 0.05$ on datasets Epinions and Youtube. This is due to the combined effects of the threshold and the batch size. In these cases, it needs no more than 8 nodes to reach the thresholds. Consequently, ASTI-8 finishes selecting seed nodes within just one round. However, when η/n increases from 0.01 to 0.05, the root size of mRR-sets decreases. As a consequence, it takes relatively less time to generate a random mRR-set in practice, which leads to the decrease in running time.

4.4.3 Results under the LT model

Seed Size vs. Threshold. Figure 4.5 reports the number of nodes selected by different algorithms under the LT model. In general, the results show similar trends to those

Figure 4.5: Number of seed nodes vs. threshold under the LT model.

observed in Figure 4.3. Similarly, ADAPTIM selects a close number of nodes as ASTI does on the four datasets, with negligible difference. ATEUC requires around 40% more nodes than the five adaptive algorithms do. Details are displayed in Table 4.3. In addition, we also observe that ASTI-8 selects more nodes than ATEUC for several settings (e.g., $\eta/n = 0.01$ on the Epinions and Youtube datasets). Through a careful analysis, we find that (i) all the algorithms select less nodes under the LT model than those under the IC model, and (ii) ASTI-8 selects 8 seed nodes in a batch with influence spread much higher than the requirements. These observations clearly tell us that there is a tradeoff in the setting of batch size. Increasing the batch size will speed up the algorithms but may result in more nodes selected.

Figure 4.6: Running time vs. threshold under the LT model.

Running Time vs. Threshold. Figure 4.6 shows the results of running time for different thresholds under the LT model. The conclusions we summarize for Figure 4.4 are generally applicable to Figure 4.6 as well. The major differences lie in two aspects: (i) the running time under the LT model is shorter than that under the IC model under the same setting as it takes less time to generate a random mRR-set under the LT model than that under the IC model (as mentioned and analyzed in previous work [17, 33]), which is consistent with the results in Figure 4.5, (ii) ASTI-4 outperforms ATEUC on Epinions and ASTI-8 outperforms ATEUC on both Epinions and Youtube for all cases under the LT model. This fact indicates (i) the batched version of ASTI is more scalable than ATEUC does, and (ii) when the batch size b is well-calibrated, ASTI can

Figure 4.7: Spread for 20 realizations on NetHEPT.

beat ATEUC in both efficiency and effectiveness.

4.4.4 Discussions on Related Spreads

Discussions on Spread Distribution. As discussed previously, non-adaptive algorithms may find solutions with influence spread far away from the requirement (i.e., either under-qualified or over-qualified). Figure 4.7 reports the spread distribution of 20 realizations achieved by the ASTI and ATEUC algorithms on the NetHEPT dataset under the IC and LT models, respectively. The solid (red) line in the figure represents the spread threshold (153) required. As shown, ATEUC fails to reach the threshold for 5 and 6 realizations under the IC and LT models, respectively, with corresponding percentages of 25% and 30%. In addition, for 5 and 6 realizations under the IC and LT models, respectively, the seed nodes selected by ATEUC produce influence spread much higher (over 50%) than the requirement. In contrast, ASTI meets the spread requirement for all the realizations under both the IC and LT models. Moreover, the spread produced by ASTI is generally kept close to the requirement. The spread exceeds the requirement by more than 50% for only 2 realizations under the LT model. These two over-qualified exceptions are due to that the last seed node selected achieves much higher spread than the gap to reach η , which is rare to happen in practice. These observations indicate that non-adaptive algorithms are unreliable for seed minimization.

Discussions on Influence Spread. Figure 4.8 reports the spread of the tested algorithms under the IC model (results under the LT model are similar). For the most

parts, all the algorithms achieve a comparable spread on the four datasets. The major differences lie in $\eta/n = 0.01$ on Epinions and Youtube. As observed, ASTI-8 (resp. ATEUC) achieves the largest (resp. smallest) spread among all algorithms. This is because the batch size b is relatively large with regard to the small threshold, owing to which the spread of the 8-size seed set selected by ASTI-8 significantly overshoots $0.01n$ on Epinions and Youtube. Another interesting observation is that the spread achieved by ATEUC is slightly larger than each of the other five adaptive algorithms as the threshold becomes larger (not quite noticeable in the figure). This is because ATEUC selects considerably more seeds than the adaptive algorithms do, resulting in a larger spread at the cost of an excessive number of seeds. This is also supported by the results in Table 4.3.

Discussions on Marginal Truncated Spread. To explore the property of the marginal truncated spread, we record the marginal spread of each seed node selected by adaptive algorithms under the 20 realizations sampled. Figure 4.9 shows the result of each realization with $\eta/n = 0.2$ on corresponding datasets (or $\eta/n = 0.05$ on the LiveJournal dataset) under the IC model. (The result under the LT model is similar.) In general, the marginal spread diminishes along the index of the seed node, which is consistent with the property of submodularity as expected. Note that the spread fluctuation is due to the randomness of the tested realizations, i.e., in some particular realizations, some seed node selected later may influence more nodes than some seed node selected earlier.

4.5 Summary

This research studies the problem of adaptive seed minimization, and proposes algorithms that provide both strong theoretical guarantees and superior empirical effectiveness. Our approach is based on a novel ASTI framework instantiated by a truncated influence maximization algorithm TRIM, which has a provable approximation guarantee. The core of our TRIM algorithm is an elegant sampling method that generates random multi-root reverse reachable (mRR) sets for estimating the truncated influence spread. We also extend TRIM into its batched version TRIM-B to further improve the efficiency of seed selection. With extensive experiments on real data, we show that our solutions

Figure 4.8: Spread vs. threshold under the IC model.

considerably outperform the state of the art for seed minimization under both the IC and LT diffusion models.

(a) NetHEPT

(b) Epinions

(c) Youtube

(d) LiveJournal

Figure 4.9: Marginal truncated spread under the IC model.

Chapter 5

Adaptive Target Profit Maximization

Profit maximization (PM) is the variant of influence maximization by incorporating the monetary factor on seed selection. The target profit maximization (TPM) problem generalizes the PM problem. As mentioned in Introduction (Chapter 1), the vanilla PM problem overlooks one fact that even though the advertisers have the full knowledge about the whole network, they are likely to only have access to a fraction of users [8], which is quite common in marketing applications. Motivated by this fact, TPM is proposed to circumvent the potential limitation that the advertisers might only have access to a small proportion of the networks. In this chapter, we study TPM in *adaptive* setting, where the seed users are selected through multiple batches, such that the selection of a batch exploits the knowledge of actual influence in the previous batches. To acquire an overall understanding, we study the adaptive TPM problem under both the *oracle model* and the *noise model*, and propose ADG and ADDATP algorithms to address them with strong theoretical guarantees, respectively. In addition, to better handle the sampling errors under the noise model, we propose the idea of *hybrid error* based on which we design a novel algorithm HATP that boosts the efficiency of ADDATP significantly. We conduct extensive experiments on real social networks to evaluate the performance, and the experimental results strongly confirm the superiorities and effectiveness of our solutions.

The remainder of the chapter is organized as follows. First we present the problem definition in Section 5.1. Afterwards, we study this problem in both oracle model and

Table 5.1: Frequently used notations

Notation	Description
T	the set of target seed nodes
k	the number of elements in T , i.e., $k = T $
S_i	the seed set selected in the i -th iteration
T_i	the seed set candidate in the i -th iteration
$I_{G_i}(u_i)$	the number of nodes activated by node u_i on G_i
$\pi^f, \pi^r, \pi^{\text{opt}}$	a front greedy policy, a rear greedy policy, and an optimal policy

noise model and analyze the corresponding approximation guarantees respectively in Section 5.2. To boost the efficiency, we propose an optimized version algorithm by adopting the idea of *hybrid error*, and also derive the approximation guarantee and time complexity accordingly. Last, we evaluate our proposed algorithms in Section 5.4.1 and conclude the work in Section 5.5.

5.1 Preliminary

In this section, we mainly present the definition of adaptive TPM. Table 5.1 summarizes the notations that are frequently used in this chapter.

Given a social network $G = (V, E)$, we consider a target user set $T \subseteq V$ and each user $u \in T$ is associated with a cost $c(u)$. For any set $S \subseteq T$, we denote $\rho(S)$ as the expected profit of S , which is defined as

$$\rho(S) := \mathbb{E}[I(S)] - c(S), \quad (5.1)$$

where $\mathbb{E}[I(S)]$ is the expected spread of S and $c(S) = \sum_{u \in S} c(u)$. As $\rho(\cdot)$ is a positive linear combination of a monotone submodular function (i.e., $\mathbb{E}[I(\cdot)]$) and a modular function (i.e., $c[\cdot]$), we know that $\rho(\cdot)$ is still submodular, which, however, may not be monotone. We assume that the expected profit of the target set T is nonnegative, i.e., $\rho(T) \geq 0$. Then, the target profit maximization (TPM) problem aims to select a subset $S \subseteq T$ to maximize expected profit.

Conventionally, seed nodes are non-adaptively selected without any knowledge of realization happened in the actual influence propagation process. Meanwhile, the adaptive

strategy selects each node based on the real-time feedback from the realization observed, which is proved to be more effective [2, 6, 9]. Specifically, the adaptive strategy first *selects* a node u_1 from the target set $T_0 = T$, and then *observes* how node u_1 activates other nodes on $G_1 = G$. Those activated nodes are removed from G_1 and T_0 , resulting in the residual graph G_2 and set T_1 from which the next node u_2 will be selected. This process is repeated until a solution S is produced.

In our study, we aim to study seed selection strategies (also known as *policies*) for adaptive target profit maximization. Let $S_\phi(\pi)$ be the seed set selected by the policy π under realization ϕ . Then, the profit of policy π under realization ϕ is $\rho_\phi(S_\phi(\pi)) = I_\phi(S_\phi(\pi)) - c(S_\phi(\pi))$. Now, we give the expected profit of policy over all possible realizations.

Definition 5.1 (Expected Profit of Policy). *The expected profit $\Lambda(\pi)$ of policy π is defined as*

$$\Lambda(\pi) := \mathbb{E}_{\Phi \sim \Omega}[\rho_\Phi(S_\Phi(\pi))] = \sum_{\phi \in \Omega} \rho_\phi(S_\phi(\pi)) \cdot p(\phi), \quad (5.2)$$

where $p(\phi)$ is the probability of ϕ sampled from Ω .

Based on the definition of expected profit of policy, adaptive target profit maximization is defined as follows.

Definition 5.2 (Adaptive TPM). *Given a probabilistic graph $G = (V, E)$ and a target user set $T \subseteq V$ with nonnegative expected profit where each node $u \in T$ is associated with a cost $c(u)$ for seed selection, the adaptive target profit maximization problem asks for a policy π^* that maximizes the expected profit over all possible realizations, i.e.,*

$$\pi^* := \arg \max_{\pi} \Lambda(\pi). \quad (5.3)$$

5.2 Adaptive Double Greedy Algorithm

In this section, we first introduce the *double greedy* method which is widely adopted to address the profit maximization problem in the literature. Then we extend double greedy to its adaptive version based on which we develop ADG algorithm and ADDATP algorithm to address the adaptive target profit maximization problem under the *oracle model* and *noise model* respectively. As what follows, we then present the analyses of the approximation guarantees of ADG and ADDATP.

Algorithm 10: Double Greedy [43]

```

1 Initialize  $S \leftarrow \emptyset, T \leftarrow V$ ;
2 foreach node  $u \in V$  do
3    $z^+ \leftarrow f(u \mid S)$ ;
4    $z^- \leftarrow -f(u \mid T \setminus \{u\})$ ;
5   if  $z^+ \geq z^-$  then  $S \leftarrow S \cup \{u\}$ ;
6   else  $T \leftarrow T \setminus \{u\}$ ;
7 return  $S (= T)$ ;
    
```

5.2.1 Double Greedy

Profit maximization is an application of *unconstrained submodular maximization (USM)* for which Buchbinder et al. [43] propose a *double greedy* algorithm as shown in Algorithm 10. The double greedy algorithm is widely used to address the profit maximization problem in the literature [26, 48, 49].

Algorithm 10 presents the pseudocode of double greedy. For any submodular function $f(\cdot)$, double greedy works as follows. It maintains two sets S and T initialized with empty set and ground set V , respectively. It then checks each node $u \in V$ one by one (in arbitrary sequence) to decide on selecting or abandoning u . To this end, it calculates the marginal gain of selecting it conditioned on current seed set S , denoted as z^+ , and the marginal gain of abandoning it conditioned on current candidate set T , denoted as z^- . If $z^+ \geq z^-$, node u will be selected and added to S . Otherwise, u is removed from T . When all nodes in V have been checked, the resultant S and T are equal and either of them is returned. The rationale behind double greedy is quite straightforward, i.e., the node is selected only if it could bring larger marginal gain by keeping it than that by removing it.

5.2.2 Adaptive Double Greedy under the Oracle Model

5.2.2.1 Description of ADG

Under the oracle model, we assume that the expected spread of any node set is accessible in $O(1)$ time. Based on this assumption, we design *adaptive double greedy (ADG)*. The decision on seed selection of ADG relies on the conditional expected marginal profit.

Algorithm 11: ADG

Input: Social graph G , a target seed set T with k nodes

Output: Selected seed node set S_k

```

1 Initialize  $S_0 \leftarrow \emptyset$ ,  $T_0 \leftarrow T$ ,  $G_1 \leftarrow G$ ;
2 for  $i \leftarrow 1$  to  $k$  do
3   if  $u_i$  is activated then
4      $T_i \leftarrow T_{i-1} \setminus \{u_i\}$ ,  $S_i \leftarrow S_{i-1}$ ,  $G_{i+1} \leftarrow G_i$ ;
5     continue;
6    $\rho_f \leftarrow \Delta_{G_i}(u_i \mid S_{i-1})$ ;
7    $\rho_r \leftarrow -\Delta_{G_i}(u_i \mid T_{i-1} \setminus \{u_i\})$ ;
8   if  $\rho_f \geq \rho_r$  then
9      $S_i \leftarrow S_{i-1} \cup \{u_i\}$ ,  $T_i \leftarrow T_{i-1}$ ;
10    Observe the node set  $A(u_i)$  activated by  $u_i$ ;
11    Update  $G_i$  into  $G_{i+1}$  by removing  $A(u_i)$ ;
12  else  $T_i \leftarrow T_{i-1} \setminus \{u_i\}$ ,  $S_i \leftarrow S_{i-1}$ ,  $G_{i+1} \leftarrow G_i$ ;
13 return  $S_k$ ;
```

Definition 5.3 (Conditional Expected Marginal Profit). *Given a node u and a node set S , the conditional expected marginal profit of u conditioned on S on graph G is defined as*

$$\Delta_G(u \mid S) := \rho_G(S \cup \{u\}) - \rho_G(S). \quad (5.4)$$

Then, we have $\Delta_G(u \mid S) = \mathbb{E}[I_G(u \mid S)] - c(u)$ if $u \notin S$, and $\Delta_G(u \mid S) = 0$ otherwise.

Algorithm 11 presents the pseudocode of ADG. First, two sets S_0 and T_0 are initialized with empty set and the target set T respectively (Line 1), where T contains k nodes. In the i -th iteration, if the examining node u_i is activated already, it just checks the next node (Lines 3–5). Otherwise, it calculates the conditional expected marginal profit $\Delta_{G_i}(u_i \mid S_{i-1})$ (Line 6), denoted as *front profit* ρ_f . Meanwhile, it also calculates $-\Delta_{G_i}(u_i \mid T_{i-1} \setminus \{u_i\})$ (Line 7), denoted as *rear profit* ρ_r . If $\rho_f \geq \rho_r$, it would (i) insert u_i into S_i as one seed node, (ii) observe the set of nodes $A(u_i)$ activated by u_i , and (iii) update G_i into G_{i+1} by removing all nodes in $A(u_i)$. Otherwise, u_i is removed from T_{i-1} . This process is repeated until all k nodes in T are examined and then the resultant S_k is returned. Note that when ADG terminates, S_k and T_k contain exactly the same nodes, i.e., $S_k = T_k$.

As shown, the decision on seed selection in ADG is fully determined by the value of ρ_f and ρ_r . In a nutshell, for each node $u \in T$, u will be selected only if selecting u could bring more marginal profit than abandoning it.

5.2.2.2 Approximation Guarantee

In this section, we would explore the approximation guarantee that ADG could achieve. First, we show the relation between the front profit ρ_f and the rear profit ρ_r in Algorithm 11.

Lemma 5.1. *If u_i is inactive, we have $\rho_f + \rho_r \geq 0$.*

Proof of Lemma 5.1. Given a residual graph G_i and the corresponding node u_i , $\rho_f = \mathbb{E}[I_{G_i}(u_i \mid S_{i-1})] - c(u_i)$ and $\rho_r = c(u_i) - \mathbb{E}[I_{G_i}(u_i \mid T_{i-1} \setminus \{u_i\})]$. Then we have $\rho_f + \rho_r = \mathbb{E}[I_{G_i}(u_i \mid S_{i-1})] - \mathbb{E}[I_{G_i}(u_i \mid T_{i-1} \setminus \{u_i\})]$. Since $S_{i-1} \subseteq T_{i-1} \setminus \{u_i\}$ and $u_i \in T \setminus (T_{i-1} \setminus \{u_i\})$, based on the property of submodularity, we have $\rho_f + \rho_r \geq 0$, which completes the proof. $\square$

Before the following discussion, we first define some vital operations on policy for further usage.

Definition 5.4 (Policy Truncation [9]). *For any adaptive seeding policy π , the policy truncation $\pi_{[i]}$ denotes an adaptive policy that performs exactly the same as π , except that $\pi_{[i]}$ only evaluates the first i ($i \leq n$) nodes in a given node sequence.*

Definition 5.5 (Policy Concatenation [9]). *For any two adaptive seeding policies π and π' , the policy concatenation $\pi \oplus \pi'$ denotes an adaptive policy that first executes the policy π , and then executes π' as if from a fresh start without any knowledge about π .*

Definition 5.6 (Policy Intersection). *For any two adaptive seeding policies π and π' , the policy intersection $\pi \otimes \pi'$ denotes an adaptive policy that executes the intersection part of policy π and π' .*

Note that under any given realization ϕ , policy concatenation shows that $S_\phi(\pi \oplus \pi') = S_\phi(\pi) \cup S_\phi(\pi')$ and policy intersection shows that $S_\phi(\pi \otimes \pi') = S_\phi(\pi) \cap S_\phi(\pi')$. Let π^{opt} be the optimal policy for adaptive TPM. Let π^f (resp. π^r) be the *front greedy* (resp. *rear greedy*) policy of ADG that executes as same as ADG and selects a set S_i (resp. T_i) of nodes after the i -th iteration, i.e., $S(\pi_{[i]}^f) = S_i$ (resp. $S(\pi_{[i]}^r) = T_i$). Define policy $\pi^\circ = (\pi^{\text{opt}} \oplus \pi^f) \otimes \pi^r$ and its truncation as $\pi_{[i]}^\circ = (\pi^{\text{opt}} \oplus \pi_{[i]}^f) \otimes \pi_{[i]}^r$. Let S°

be the solution obtained by π^{opt} , and S_i° be the temporal seed set obtained by $\pi_{[i]}^{\text{opt}}$, i.e., $S_i^\circ = S(\pi_{[i]}^\circ) = (S^\circ \cup S_i) \cap T_i$. Then, we have $\Lambda(\pi_{[0]}^f) = 0$, $\Lambda(\pi_{[0]}^r) = \rho(T) \geq 0$, $\Lambda(\pi_{[0]}^\circ) = \Lambda(\pi^{\text{opt}})$, and $\Lambda(\pi_{[k]}^f) = \Lambda(\pi_{[k]}^r) = \Lambda(\pi_{[k]}^\circ)$, where $k = |T|$. First of all, we bound the profit achievement of policies π^f and π^r on residual graph G_i in the following lemma.

Lemma 5.2. *For the i -th iteration of ADG, we have*

$$\begin{aligned} & \rho_{G_i}(S_{i-1}^\circ) - \rho_{G_i}(S_i^\circ) \\ & \leq \rho_{G_i}(S_i) - \rho_{G_i}(S_{i-1}) + \rho_{G_i}(T_i) - \rho_{G_i}(T_{i-1}). \end{aligned} \tag{5.5}$$

Based on Lemma 5.2, we further derive the following lemma for the expected profit of policies π^f , π^r , and policy π° .

Lemma 5.3. *For the i -th iteration of ADG, we have*

$$\Lambda(\pi_{[i-1]}^\circ) - \Lambda(\pi_{[i]}^\circ) \leq \Lambda(\pi_{[i]}^f) - \Lambda(\pi_{[i-1]}^f) + \Lambda(\pi_{[i]}^r) - \Lambda(\pi_{[i-1]}^r).$$

Lemma 5.3 establishes the relation on expected profit of policy between ADG and the optimal policy π^{opt} for each iteration. Based on Lemma 5.3, the approximation guarantee of ADG is derived as follows.

Theorem 5.1. *ADG achieves the approximation ratio of $1/3$.*

Proof of Theorem 5.1. From Lemma 5.3, we have

$$\begin{aligned} & \Lambda(\pi_{[0]}^\circ) - \Lambda(\pi_{[k]}^\circ) \\ & = \sum_{i=0}^k (\Lambda(\pi_{[i-1]}^\circ) - \Lambda(\pi_{[i]}^\circ)) \\ & \leq \sum_{i=0}^k (\Lambda(\pi_{[i]}^f) - \Lambda(\pi_{[i-1]}^f) + \Lambda(\pi_{[i]}^r) - \Lambda(\pi_{[i-1]}^r)) \\ & = \Lambda(\pi_{[k]}^f) - \Lambda(\pi_{[0]}^f) + \Lambda(\pi_{[k]}^r) - \Lambda(\pi_{[0]}^r). \end{aligned}$$

Recalling that $\Lambda(\pi_{[0]}^f) = 0$, $\Lambda(\pi_{[0]}^r) = \rho(T) \geq 0$, $\Lambda(\pi_{[0]}^\circ) = \Lambda(\pi^{\text{opt}})$, and $\Lambda(\pi_{[k]}^f) = \Lambda(\pi_{[k]}^r) = \Lambda(\pi_{[k]}^\circ)$, we obtain that $\Lambda(\pi_{[k]}^f) = \Lambda(\pi_{[k]}^r) \geq \Lambda(\pi^{\text{opt}})/3$. $\square$

5.2.3 Adaptive Double Greedy under the Noise Model

As well-known, computing the exact expected spread of any node set is #P-hard [21]. In this section, we try to estimate the expected spread of any seed set by taking the sampling error into account. In general, there are two forms of sampling error, i.e., *relative error* and *additive error*. Considering that we need to estimate the expected marginal spreads of $2k$ node sets during the whole process, it can be quite intricate utilizing relative error. To explain, for those with small expected marginal spreads, only a trivial amount of estimation error would be allowed, which is rather difficult to achieve by existing methods for spread estimation. Motivated by this, we adopt the additive error instead and propose the ADDATP¹ algorithm. Algorithm 12 presents the details of ADDATP.

5.2.3.1 Description of AddATP

The design principle of ADDATP follows that of ADG except that ρ_f and ρ_r in ADDATP are not accessible but to be estimated. The estimation reliability is guaranteed by *Hoeffding Inequality* [56]. Specifically, ADDATP first initializes S_0 with $\emptyset$ and T_0 with target set T (Line 1). In each iteration on residual graph G_i , it first checks if the current node u_i is activated. If u_i is activated, it skips current node and starts the next iteration immediately. Otherwise, ADDATP initializes the additive error parameter $\zeta_i = \zeta_0$ and the probability parameter $\delta_i = 1/(kn)$ (Line 6) for the following evaluations. Notice that a proper initialization value ζ_0 could speed up the performance empirically because some nodes can be decided within relatively small number of samples. To estimate the expected marginal spreads, ADDATP adopts the commonly used *reverse influence sampling (RIS)* [13] technique, referred to as *RR sets*. Specifically, ADDATP generates θ RR sets into $\mathcal{R}_1$ and $\mathcal{R}_2$ to calculate $\tilde{\rho}_f$ (Line 10) and $\tilde{\rho}_r$ (Line 11) respectively, where θ is determined by ζ_i and δ_i , $\tilde{\rho}_f$ and $\tilde{\rho}_r$ are the estimations of $\mathbb{E}[I_{G_i}(u_i | S_{i-1})] - c(u_i)$ and $c(u_i) - \mathbb{E}[I_{G_i}(u_i | T_{i-1} \setminus \{u_i\})]$. By Hoeffding Inequality [56], we have $\rho_f \in [\tilde{\rho}_f - n_i \zeta_i, \tilde{\rho}_f + n_i \zeta_i]$ and $\rho_r \in [\tilde{\rho}_r - n_i \zeta_i, \tilde{\rho}_r + n_i \zeta_i]$ with high probability. Then, the stopping conditions C_1 and C_2 are checked (Line 12), where C_1 and C_2 are defined as

$$C_1: (|\tilde{\rho}_f - \tilde{\rho}_r| \geq 2n_i \zeta_i) \vee (\tilde{\rho}_f \leq -n_i \zeta_i) \vee (\tilde{\rho}_r \leq -n_i \zeta_i),$$

$$C_2: n_i \zeta_i \leq 1.$$

¹Algorithms with additive error for adaptive targeted profit maximization

Algorithm 12: ADDATP

Input: Social graph G , a target seed set T with k nodes

Output: Selected seed node set S_k

```

1 Initialize  $S_0 \leftarrow \emptyset$ ,  $T_0 \leftarrow T$ ;
2 for  $i \leftarrow 1$  to  $k$  do
3     if  $u_i$  is activated then
4          $T_i \leftarrow T_{i-1} \setminus \{u_i\}$ ,  $S_i \leftarrow S_{i-1}$ ,  $G_{i+1} \leftarrow G_i$ ;
5         continue;
6      $\zeta_i \leftarrow \zeta_0$ ,  $\delta_i \leftarrow 1/(kn)$ ;                                     //  $\zeta_0 \geq 1/n$ 
7     while true do
8          $\theta \leftarrow \frac{1}{2\zeta_i^2} \ln \frac{8}{\delta_i}$ ;
9         Generate  $\theta$  RR sets as  $\mathcal{R}_1$  and  $\mathcal{R}_2$ , respectively;
10         $\tilde{\rho}_f \leftarrow \text{Cov}_{\mathcal{R}_1}(u_i \mid S_{i-1}) \cdot \frac{n_i}{\theta} - c(u_i)$ ;
11         $\tilde{\rho}_r \leftarrow -\text{Cov}_{\mathcal{R}_2}(u_i \mid T_{i-1} \setminus \{u_i\}) \cdot \frac{n_i}{\theta} + c(u_i)$ ;
12        if  $C_1$  or  $C_2$  then
13            if  $\tilde{\rho}_f \geq \tilde{\rho}_r$  then
14                 $S_i \leftarrow S_{i-1} \cup \{u_i\}$ ,  $T_i \leftarrow T_{i-1}$ ;
15                Observe the node set  $A(u_i)$  activated by  $u_i$ ;
16                Update  $G_i$  into  $G_{i+1}$  by removing  $A(u_i)$ ;
17            else  $T_i \leftarrow T_{i-1} \setminus \{u_i\}$ ,  $S_i \leftarrow S_{i-1}$ ,  $G_{i+1} \leftarrow G_i$ ;
18            break;
19         $\zeta_i \leftarrow \zeta_i/\sqrt{2}$ ,  $\delta_i \leftarrow \delta_i/2$ ;
20 return  $S_k$ ;
    
```

If the stopping condition C_1 is met, it indicates that the current estimations $\tilde{\rho}_f$ and $\tilde{\rho}_r$ are accurate enough to help make the right decision with high probability. If C_2 is observed instead, it indicates that (i) the expected spread of current node is too close to the judgement bar to distinguish, and (ii) the profit loss is insignificant if wrong decision is made. The purpose of this condition is to avoid the unnecessarily prohibitive sampling overhead for sufficient accurate of spread estimations. When one of the stopping conditions is met, the decision is made accordingly (Lines 13–17). Otherwise, ζ_i (resp. δ_i) is divided by $\sqrt{2}$ (resp. 2) (Line 19) to generate more samples for more accurate estimations. This process terminates when all k nodes in T are checked through.

5.2.3.2 Theoretical Analysis of AddATP

In what follows, we analyze the approximation guarantee and time complexity of ADDATP.

Approximation Guarantee. First, we present the *Hoeffding Inequality* [56] based on which the estimation is reliable with high probability.

Lemma 5.4 (Hoeffding Inequality [56]). *Let X_i be an independent bounded random variable such that for each $1 \leq i \leq \theta$, $X_i \in [a_i, b_i]$. Let $X = \frac{1}{\theta} \sum_{i=1}^{\theta} X_i$ and $\mu = \mathbb{E}[X]$. Given $\zeta \in (0, 1)$, then*

$$\Pr[|X - \mu| \geq \zeta] \leq 2e^{-\frac{2\theta^2\zeta^2}{\sum_{i=1}^{\theta}(b_i - a_i)^2}}. \quad (5.6)$$

Similarly, to obtain the approximation ratio of ADDATP, we need to establish the relation between $\Lambda(\pi_{[i-1]}^{\circ}) - \Lambda(\pi_{[i]}^{\circ})$ and $\Lambda(\pi_{[i]}^f) - \Lambda(\pi_{[i-1]}^f) + \Lambda(\pi_{[i]}^r) - \Lambda(\pi_{[i-1]}^r)$ as the one in Lemma 5.3. We note that ADG is a *deterministic* algorithm while ADDATP is a *randomized* algorithm. To explain, ADG has access to the expected spread of any node set under the oracle model. Contrarily, under the noise model, ADDATP adopts the randomized *reverse influence sampling (RIS)* [13] technique to estimate the expected spread, which makes π^f , π^r , and π° all random polices. Consequently, we need to tackle the internal randomness of ADDATP.

Lemma 5.5. *For the i -th iteration of ADDATP, we have*

$$\begin{aligned} & \mathbb{E}_{\mathcal{A}}[\rho_{G_i}(S_{i-1}^{\circ}) - \rho_{G_i}(S_i^{\circ})] - (2 + 2/k) \\ & \leq \mathbb{E}_{\mathcal{A}}[\rho_{G_i}(S_i) - \rho_{G_i}(S_{i-1}) + \rho_{G_i}(T_i) - \rho_{G_i}(T_{i-1})]. \end{aligned} \quad (5.7)$$

Thus, for the expected profit of policy π^f , π^r , and π° in the i -th iteration, we have following lemma.

Lemma 5.6. *For the i -th iteration of ADDATP, we have*

$$\begin{aligned} & \mathbb{E}_{\mathcal{A}}[\Lambda(\pi_{[i-1]}^{\circ}) - \Lambda(\pi_{[i]}^{\circ})] - (2 + 2/k) \\ & \leq \mathbb{E}_{\mathcal{A}}[\Lambda(\pi_{[i]}^f) - \Lambda(\pi_{[i-1]}^f) + \Lambda(\pi_{[i]}^r) - \Lambda(\pi_{[i-1]}^r)], \end{aligned} \quad (5.8)$$

where the expectation $\mathbb{E}_{\mathcal{A}}[\cdot]$ is over the internal randomness of ADDATP and $k = |T|$.

Note that there are additional terms $2/k$ and 2 in (5.8). Specifically, $2/k$ is the compensation factor on the profit loss when ADDATP makes the wrong choice on u_i due to the failed spread estimation. 2 is the upper bound of the profit loss incurred when ADDATP terminates due to the stopping condition C_2 . Interested readers are referred to Appendix for more details.

Based on Lemma 5.6, we could derive the approximation ratio of ADDATP as follows.

Theorem 5.2. ADDATP could achieve the expected profit at least $\frac{\Lambda(\pi^{\text{opt}}) - (2k+2)}{3}$, where $\Lambda(\pi^{\text{opt}})$ is the expected profit of optimal policy π^{opt} and k is the number of nodes in T .

Proof of Theorem 5.2. According to Lemma 5.6, accumulating both sides of (5.8) from $i = 1$ to k gives

$$\begin{aligned} & \mathbb{E}_{\mathcal{A}}[\Lambda(\pi_{[0]}^{\circ}) - \Lambda(\pi_{[k]}^{\circ})] - \sum_{i=1}^k (2 + 2/k) \\ & \leq \mathbb{E}_{\mathcal{A}}[\Lambda(\pi_{[k]}^{\text{f}}) - \Lambda(\pi_{[0]}^{\text{f}}) + \Lambda(\pi_{[k]}^{\text{r}}) - \Lambda(\pi_{[0]}^{\text{r}})]. \end{aligned}$$

Thus, we have

$$\mathbb{E}_{\mathcal{A}}[\Lambda(\pi_{[k]}^{\text{f}})] \geq \frac{\Lambda(\pi^{\text{opt}}) - (2 + 2k)}{3},$$

which completes the proof. $\square$

Discussion. On particular social graphs, our algorithm could achieve an expected ratio of $(1 - \varepsilon)/3$. The main idea is to dynamically set the threshold of $n_i \zeta_i$ for stopping condition C_2 instead of fixing to 1. Specifically, in each iteration, we have (at most) a profit loss of $2/k$ due to a failure estimation of marginal expected spread and an extra profit loss of 2 if stopping condition C_2 , i.e., $n_i \zeta_i \leq 1$, occurs (see the proof of Lemma 5.5). Thus, we have (at most) a total profit loss of $2 + 2k$ for all k iterations. However, we note that it is unlikely that all k iterations meet C_2 . Thus, we could try to bound the actual profit loss within $\varepsilon \Lambda(\pi^{\text{opt}})$ by adjusting the error threshold dynamically as follows. For the i -th iteration, let η_i be the settled threshold of $n_i \zeta_i$, i.e., $C_2: n_i \zeta_i \leq \eta_i$, and ρ_i be the accumulated profit. Let $\tilde{\eta}_i$ be the indicator whether C_2 occurs such that $\tilde{\eta}_i = \eta_i$ if C_2 occurs and $\tilde{\eta}_i = 0$ if C_1 occurs. In the $(i + 1)$ -th iteration, we set $\eta_{i+1} = (\varepsilon \rho_i - 2 \sum_{j=1}^i \tilde{\eta}_j - 2)/2$ as long as $\varepsilon \rho_i \geq 2 \sum_{j=1}^i \tilde{\eta}_j + 2$, which ensures that $2 \sum_{j=1}^{i+1} \tilde{\eta}_j + 2 \leq \varepsilon \rho_i$. With this dynamic strategy, ADDATP could achieve an approximation ratio of $(1 - \varepsilon)/3$.

Time Complexity. As indicated in Algorithm 12, there are $O(\frac{1}{\zeta_i^2} \ln \frac{1}{\delta_i})$ random RR sets generated in the i -th iteration. There are at most $2 \lceil \log(n_i \zeta_0) \rceil$ rounds for the i -th iteration and $\zeta_0 \in [1/n, 1]$, thus δ_i is bounded by $O(\frac{1}{kn^3})$. Then the total number of RR sets is at most $O(\frac{\ln n}{\zeta_i^2})$. According to Lemma 4 in [10], the expected time for generating a random RR set on G_i , denoted as EPT, is $\text{EPT} \leq \frac{m_i}{n_i} \mathbb{E}[I_{G_i}(\{v_i^{\circ}\})]$ where v_i° is the node

with the largest expected spread on G_i . By Wald's equation [55], the expected time complexity of ADDATP is

$$O\left(\sum_{i=1}^k \left(\frac{\ln n}{\zeta_i^2} \cdot \frac{m_i \mathbb{E}[I(\{v_i^\circ\})]}{n_i}\right)\right) = O\left(kmn \mathbb{E}[I(\{v_1^\circ\})] \ln n\right),$$

since $n_i \zeta_i \geq \sqrt{2}/2$ and $\mathbb{E}[I(\{v_i^\circ\})] \leq \mathbb{E}[I(\{v_1^\circ\})]$ for any i .

Theorem 5.3. *The expected time complexity of ADDATP is $O(kmn \mathbb{E}[I(\{v^\circ\})] \ln n)$ where v° is the node with the largest expected spread on G .*

5.3 Optimization with Hybrid Error

In this section, we aim to optimize ADDATP and propose HATP in terms of efficiency. We first analyze the rationale of optimization and then conduct theoretical analysis on HATP towards its approximation guarantee and time complexity.

5.3.1 Rationale of Optimization

ADDATP in Section 5.2 considers the additive error on spread estimation, which may suffer from efficiency issues. Recall that for nodes with expected spreads close to the judgement bar, ADDATP uses the stopping condition $n_i \zeta_i \leq 1$ to avoid unnecessary computation overhead. However, when $\zeta_i = O(1/n_i)$, the number of RR sets required is $O(n_i^2 \ln n)$, which would incur prohibitive computation overhead. To tackle this issue, we propose to estimate the expected spread via a *hybrid error*, i.e., the combination of the additive error and relative error. The rationale of hybrid error is that for those nodes with large expected marginal spread, their estimation errors are easily to be bounded within relative error, while for those nodes with small expected marginal spread, their estimation errors are easily to be bounded within additive error. Thus, the expected marginal spread for every node can be efficiently estimated utilizing hybrid error. This estimation reliability using hybrid error is guaranteed by *Relative+Additive Concentration Bound* as follows.

Algorithm 13: HATP

Input: Social graph G , a size- k set T , the threshold ε

Output: Selected seed node set S_k

```

1 Initialize  $S_0 \leftarrow \emptyset, T_0 \leftarrow T$ ;
2 for  $i \leftarrow 1$  to  $k$  do
3   if  $u_i$  is activated then
4      $T_i \leftarrow T_{i-1} \setminus \{u_i\}, S_i \leftarrow S_{i-1}, G_{i+1} \leftarrow G_i$ ;
5     continue;
6    $\varepsilon_i \leftarrow \varepsilon_0, \zeta_i \leftarrow \zeta_0, \delta_i \leftarrow 1/(kn)$ ; //  $\varepsilon_0 \geq \varepsilon, \zeta_0 \geq 1/n$ 
7   while True do
8      $\theta \leftarrow \frac{(1+\varepsilon_i/3)^2}{2\varepsilon_i\zeta_i} \ln(\frac{4}{\delta_i})$ ;
9     Generate  $\theta$  RR sets as  $\mathcal{R}_1$  and  $\mathcal{R}_2$ , respectively;
10     $f_{\text{est}} \leftarrow \text{Cov}_{\mathcal{R}_1}(u_i | S_{i-1}) \cdot \frac{n_i}{\theta}$ ;
11     $r_{\text{est}} \leftarrow \text{Cov}_{\mathcal{R}_2}(u_i | T_{i-1} \setminus \{u_i\}) \cdot \frac{n_i}{\theta}$ ;
12    if  $C'_1$  or  $C'_2$  then
13      if  $f_{\text{est}} + r_{\text{est}} \geq 2c(u_i)$  then
14         $S_i \leftarrow S_{i-1} \cup \{u_i\}, T_i \leftarrow T_{i-1}$ ;
15        Observe the node set  $A(u_i)$  activated by  $u_i$ ;
16        Update  $G_i$  into  $G_{i+1}$  by removing  $A(u_i)$ ;
17      else  $T_i \leftarrow T_{i-1} \setminus \{u_i\}, S_i \leftarrow S_{i-1}, G_{i+1} \leftarrow G_i$ ;
18      break;
19    if  $n_i\zeta_i \leq 1$  or  $f_{\text{est}} \geq 10n_i\zeta_i$  then  $\varepsilon_i \leftarrow \varepsilon_i/2$ ;
20    else if  $\varepsilon_i \leq \varepsilon$  or  $f_{\text{est}} \leq n_i\zeta_i$  then  $\zeta_i \leftarrow \zeta_i/2$ ;
21    else  $\varepsilon_i \leftarrow \varepsilon_i/\sqrt{2}, \zeta_i \leftarrow \zeta_i/\sqrt{2}$ ;
22     $\delta_i \leftarrow \delta_i/2$ ;
23 return  $S_k$ 
    
```

Lemma 5.7 (Relative+Additive Concentration Bound). *Let $X_1 - \mathbb{E}[X_1], \dots, X_\theta - \mathbb{E}[X_\theta]$ be a martingale difference sequence such that $X_i \in [0, 1]$ for each i . Let $X = \frac{1}{\theta} \sum_{i=1}^\theta X_i$ and $\mu = \mathbb{E}[X]$. Given $\varepsilon, \zeta \in (0, 1)$, then*

$$\Pr[X \geq (1 + \varepsilon)\mu + \zeta] \leq e^{-\frac{2\theta\varepsilon\zeta}{(1+\varepsilon/3)^2}}, \quad (5.9)$$

$$\Pr[X \leq (1 - \varepsilon)\mu - \zeta] \leq e^{-2\theta\varepsilon\zeta}. \quad (5.10)$$

Proof. According to the martingale concentration [11], we have $\Pr[X \leq (1 - \varepsilon)\mu - \zeta] \leq e^{-\frac{(\varepsilon\mu + \zeta)^2\theta}{2\mu}} \leq e^{-\frac{(2\sqrt{\varepsilon\mu\zeta})^2\theta}{2\mu}} = e^{-2\varepsilon\zeta\theta}$. Similarly, we have $\Pr[X \geq (1 + \varepsilon)\mu + \zeta] \leq e^{-h(\lambda)}$, where $h(\lambda) = \frac{(\lambda^2\theta)}{2(\lambda - \zeta)/\varepsilon + 2\lambda/3}$ and $\lambda = \varepsilon\mu + \zeta$. Let

$$\frac{dh(\lambda)}{d\lambda} = \frac{(2\lambda((\lambda - \zeta)/\varepsilon + \lambda/3) - (1/\varepsilon + 1/3)\lambda^2)\theta}{2((\lambda - \zeta)/\varepsilon + \lambda/3)^2} \triangleq 0.$$

Thus, $h(\lambda)$ achieves its minimum at $\lambda = \frac{2\zeta}{\varepsilon(1/\varepsilon+1/3)}$ such that $h(\lambda) = \frac{2\varepsilon\beta\theta}{(1+\varepsilon/3)^2}$. This completes the proof. $\square$

Based on the hybrid error, we propose HATP² algorithm, as shown in Algorithm 13.

5.3.2 Description of HATP

The design principle of HATP is similar to that of ADDATP except that HATP adopts *hybrid error* instead of *additive error*. The major differences between HATP and ADDATP lie in two aspects. First, the stopping conditions C_1 and C_2 (Line 12) have been updated accordingly in HATP as follows.

$$\begin{aligned} C'_1: & \left(\frac{f_{\text{est}} + r_{\text{est}} - 2n_i\zeta_i}{1+\varepsilon} \geq 2c(u_i) \right) \vee \left(\frac{r_{\text{est}} - n_i\zeta_i}{1+\varepsilon} \geq c(u_i) \right) \\ & \vee \left(\frac{f_{\text{est}} + r_{\text{est}} + 2n_i\zeta_i}{1-\varepsilon} \leq 2c(u_i) \right) \vee \left(\frac{f_{\text{est}} + n_i\zeta_i}{1-\varepsilon} \leq c(u_i) \right), \\ C'_2: & (\varepsilon_i \leq \varepsilon) \wedge (n_i\zeta_i \leq 1). \end{aligned}$$

Second, the error parameters ε_i and ζ_i are adjusted adaptively in HATP (Lines 19–21) instead of decreasing with a fixed ratio in ADDATP. Specifically, if the current addition error $n_i\zeta_i$ reaches the threshold or it is the one magnitude smaller than estimated f_{est} , we can infer that the expected marginal spread of u_i is much larger than the additive error. In such case, the relative error ε_i is halved (Line 19). Similarly, if the current relative error ε_i reaches the threshold or the additive error is larger than estimated f_{est} , we should halve the additive error $n_i\zeta_i$ (Line 20). Otherwise, both relative and additive errors are decreased by a factor of $\sqrt{2}$ (Line 21). This adaptive adjustment could boost the efficiency of HATP significantly.

5.3.3 Theoretical Analysis of HATP

Approximation Guarantee. To derive the approximation guarantee of HATP, we need a similar equation like (5.8) to bridge our solution with the optimal solution in each iteration. Toward this end, we have the following lemma.

²Algorithms with hybrid error for adaptive targeted profit maximization

Lemma 5.8. *For the i -th iteration of HATP, we have*

$$\begin{aligned} \mathbb{E}_{\mathcal{A}}[\Lambda(\pi_{[i-1]}^{\circ}) - \Lambda(\pi_{[i]}^{\circ})] &- \frac{2(1 + \varepsilon c(u_i))}{1 - \varepsilon} - \frac{2}{k} \\ &\leq \mathbb{E}_{\mathcal{A}}[\Lambda(\pi_{[i]}^f) - \Lambda(\pi_{[i-1]}^f) + \Lambda(\pi_{[i]}^r) - \Lambda(\pi_{[i-1]}^r)] \end{aligned}$$

where the expectation $\mathbb{E}_{\mathcal{A}}$ is over the internal randomness of HATP and ε is the threshold of relative error.

Lemma 5.8 establishes the relation between policies π^f , π^r and π° for each iteration, based on which, we have following theorem on the approximation of HATP.

Theorem 5.4. *HATP achieves the expected profit at least $\frac{\Lambda(\pi^{\text{opt}}) - 2(k + \varepsilon c(T))/(1 - \varepsilon) - 2}{3}$ for any $\varepsilon \in (0, 1)$, where $c(T)$ is the cost of T .*

Time Complexity. In the i -th iteration, there are $O(\frac{1}{\varepsilon \zeta_i} \ln \frac{1}{\delta_i})$ random RR sets, where $\delta_i = O(\frac{\varepsilon}{n})$. Thus, the expected time complexity of HATP is

$$O\left(\sum_{i=1}^k \left(\frac{\ln \frac{n}{\varepsilon}}{\varepsilon \zeta_i} \cdot \frac{m_i \mathbb{E}[I(\{v_i^{\circ}\})]}{n_i}\right)\right) = O\left(\frac{km \mathbb{E}[I(\{v_1^{\circ}\})]}{\varepsilon} \ln \frac{n}{\varepsilon}\right).$$

Theorem 5.5. *The expected time complexity of HATP is $O(\frac{km \mathbb{E}[I(\{v^{\circ}\})]}{\varepsilon} \ln \frac{n}{\varepsilon})$ where v° is the node with the largest expected spread on G .*

Note that HATP is approximately $O(\varepsilon n)$ more efficient than ADDATP. Usually, $\varepsilon = O(1)$, e.g., $\varepsilon = 0.1$, in the literature, HATP achieves a factor of $O(n)$ improvement on efficiency.

5.4 Experiments

In this section, we evaluate the performance of our proposed algorithms through extensive experiments. We measure the efficiency and effectiveness in real online social networks. Our experiments are deployed on a Linux machine with an Inter Xeon 2.6GHz CPU and 64GB RAM.

Table 5.2: Dataset details. ($K = 10^3, M = 10^6$)

Dataset	n	m	Type	Avg. deg
NetHEPT	15.2K	31.4K	undirected	4.18
Epinions	132K	841K	directed	13.4
DBLP	655K	1.99M	undirected	6.08
LiveJournal	4.85M	69.0M	directed	28.5

5.4.1 Experimental Setting

Datasets. Four online social networks are used in our experiments, namely NetHEPT, Epinions, DBLP, LiveJournal, as presented in Table 5.2. Among them, NetHEPT [29] is the academic collaboration networks of “High Energy Physics-Theory” area. The rest three datasets are real-life social networks available in [51]. In particular, LiveJournal contains millions of nodes and edges. For fair comparison, we randomly generate 20 possible realizations for each dataset, each of which is generated strictly following the definitions of the influence propagation models in the literature. We then report the average performance of each tested algorithm on the 20 possible realizations.

Algorithms. First, we try to evaluate the two proposed adaptive algorithms HATP and ADDATP. However, ADDATP encounters *out of memory* issue even on the smallest dataset NetHEPT, which reveals the serious efficiency problem of ADDATP. Therefore we omit ADDATP in our evaluation. To verify the advantage of adaptive algorithms over non-adaptive algorithms, we tailor HATP into a non-adaptive version, referred to as HNTP³, to address non-adaptive TPM. Meanwhile, we also include two extra non-adaptive algorithms proposed in the latest work for the non-adaptive profit maximization problem [26], i.e., *non-adaptive simple greedy* (NSG) and *non-adaptive double greedy* (NDG). Note that both NSG and NDG are heuristic algorithms with no theoretical guarantees [26]. For relatively fair comparison, we set the sample size of NSG and NDG as the largest number of samples generated in HATP for one iteration in all settings.

Parameter settings. To conduct a comprehensive evaluation, we design two different procedures to obtain a suitable target set T and the corresponding cost of each user in T . First, we follow the setting in [6, 7] where the cost of each user in T is based on the

³Algorithms with hybrid error for nonadaptive targeted profit maximization

expected spread of T . We use one of the state of the arts [11] for influence maximization to obtain the top- k influential users as the target seed set. We vary the target size k as $k = \{10, 25, 50, 100, 200, 500\}$. To determine the cost of each user $u \in T$, we (i) estimate the lower bound of T 's expected spread $\mathbb{E}[I(T)]$ as $\mathbb{E}^l[I(T)]$, and (ii) ensure that $c(T) = \mathbb{E}^l[I(T)]$. Under this condition, we design two cost settings, i.e., *degree-proportional* cost setting and *uniform* cost setting. In the degree-proportional setting, the cost of each node is proportional to its out-degree. In the uniform cost setting, the cost of each node is equal.

Second, we follow the setting in the latest work [26] where the cost of each user in the graph G is predefined before we choose the target set T . Let $\lambda = c(V)/n$ be the ratio of cost to node number where $n = |V|$ and $c(V)$ is the cost of V . To get a proper size of T , we vary λ as $\lambda = \{200, 300, 400, 500\}$. Then the cost of each node in V also follows the degree-proportional cost setting and uniform cost setting respectively. We then adopt the two proposed methods, namely NDG and NSG to identify the target set T .

In a nutshell, in the first setting, we choose a target set T first and then assign a cost to each node in T based on the expected spread of T , while in the second setting, we assign a cost to each node first and then find a target set T based on the cost assignment. For the two settings, we set the input parameters $n_i\zeta_0 = 64$, $\varepsilon_0 = 0.5$, and its threshold $\varepsilon = 0.05$ in HATP and HNTP. Our experimental results indicate that by setting the initial additive error $n_i\zeta_0 = 64$, initial relative error $\varepsilon_0 = 0.5$, and its threshold $\varepsilon = 0.05$, the proposed algorithms could achieve a good balance between efficiency and efficacy and users are suggested to follow our setting. Following a large body of existing work in the literature [4, 10–12, 17], we set the propagation probability of each edge $e = (u, v)$ to $\frac{1}{d_{\text{in}}(v)}$, where $d_{\text{in}}(v)$ is the in-degree of node v .

5.4.2 Comparison of Profit

Degree-proportional Cost. Figure 5.1 reports the profits achieved by the four tested algorithms under the degree-proportional cost setting. In addition, the dark-blue line with cross mark ($\times$) named *Baseline* represents the estimated profit of the target set T . We observe that all four tested algorithms could improve the profit of baseline significantly. In particular, HATP shows superior advantage over the other three non-adaptive

Figure 5.1: Profit in degree-proportional cost.

algorithms. Specifically, profit achieved by our adaptive algorithm is around 10%–15% larger than those of non-adaptive algorithms on average, where the improvement percentage is as high as 20% on the Epinions dataset. This verifies the effectiveness of our solutions.

As with the non-adaptive algorithms, i.e., HNTTP, NSG, and NDG, they obtain quite comparable profits on the four datasets. In particular, we observe that the profit of NSG is slightly higher than the profits of HNTTP and NDG on datasets DBLP and LiveJournal, which implies the minor advantage of simple greedy over double greedy on large datasets.

Uniform Cost. Figure 5.2 shows the profit results under the uniform cost setting. At

Figure 5.2: Profit in uniform cost.

the first glance, the results follow the trend of Figure 5.1. However, there are two major distinctions on the profit results between the two cost settings. First, algorithms achieve around 50% more profits in the uniform cost setting than they do in the degree-proportional cost setting. This can be explained as follows. In the degree-proportional cost setting, the cost of each user is proportional to its out-degree which is highly correlated with its expected spread. In this regard, each user's cost is roughly proportional to its spread, which largely limits the profits the influential nodes could contribute in some degree. Contrarily, in the uniform cost setting where all users are assigned with the same cost, influential nodes can be more easily picked out to exert their influence and bring more profits.

Figure 5.3: Running time in degree-proportional cost.

The other notable distinction is that the gap of profit between adaptive algorithms and non-adaptive algorithms becomes smaller in the uniform cost setting. As aforementioned, those profitable nodes are easier to be identified under this setting. As a consequence, the overlap of selected seed sets between adaptive and non-adaptive algorithms expands, which weakens the adaptivity advantage slightly.

5.4.3 Comparison of Running Time

Degree-proportional Cost. Figure 5.3 presents the results of running time under the degree-proportional cost setting. Our approximation algorithms HATP and HNTTP run slower than the heuristic algorithms NSG and NDG. This is because HATP and HNTTP

Figure 5.4: Running time in uniform cost.

regenerate RR sets in all k iterations from scratch to ensure bounded sampling errors, as shown in Algorithm 13. On the contrary, NSG and NDG complete seed selection on one set of RR sets as they do not have any guarantee on profit estimation. Therefore, HATP and HNTTP generate around k times samples than NSG and NDG do.

Meanwhile, HNTTP runs slightly slower than HATP does. Recall that HNTTP is the non-adaptive version of HATP and there is no any graph update on G after each node selection. Contrarily, HATP would update current graph into residual graph by removing newly activated nodes in each iteration. We observe that generating an RR set on a smaller residual graph is faster than that on the original graph. Therefore, HATP spends less time than HNTTP on sampling.

Figure 5.5: Profits of HATP and NDG on LiveJournal.

Figure 5.6: Profits of HATP and NSG on LiveJournal.

Uniform Cost. Figure 5.4 displays the results of running time under the uniform cost setting. The result trend keeps consistent with that in degree-proportional cost setting. The major difference between the two is that the running time in Figure 5.3 is smaller than that in Figure 5.4 for each algorithm on the corresponding setting. This can be explained with the reasons aforementioned, i.e., profitable nodes in uniform cost setting are easier identified within less samples. Therefore, the corresponding running time becomes shorter.

5.4.4 Comparison of Profit with Predefined Cost

This section explores the profit improvement of HATP over NDG and NSG, following the setting that the cost of each user is predefined before we choose the target set T .

Specifically, after the cost of each user is set, we adopt NDG and NSG to derive the target seed set T respectively. As implied in Section 5.4.2, profit improvement exhibits similar characteristics on the four datasets. Therefore, this part of experiment is conducted only on the largest dataset LiveJournal.

Figure 5.5 presents the profits of HATP and NDG with various λ values under the two cost settings. Note that smaller λ value means larger seed set size k . The overall improvement of HATP over NDG is around 10% and 15% under the degree-proportional and uniform cost settings respectively. In particular, the improvement ratio can be up to 21.3% with $\lambda = 200$ in Figure 5.5(b). The advantage of HATP over NDG in Figure 5.5(b) gets more notable along the decrease of λ , which reveals that larger the target set size is, more effective the adaptive algorithms become. This can also explain why the advantage in Figure 5.5(b) is more obvious than that in Figure 5.5(a), since the target set size under the uniform cost setting is larger with the same value of λ . Figure 5.6 shows the profit improvement of HATP over NSG. The overall improvement of HATP over NSG is 5% or so, less significant compared with the results in Figure 5.5. This indicates that NSG might be more effective than NDG on profit maximization. Observe that the improvement is more notable in Figure 5.6(b) than that in Figure 5.6(a), which again verifies the fact that the advantage of adaptive algorithms over non-adaptive algorithms is more impressive when the target size gets larger.

5.5 Summary

We study the adaptive target profit maximization (TPM) problem which aims to identify a subset of nodes from the target node set to maximize the expected profit utilizing the adaptive advantage. To acquire an overall understanding on the adaptive TPM problem, we investigate this problem in both oracle model and noise model. We prove that adaptive double greedy could address adaptive TPM under the oracle model and achieve an approximation ratio of $\frac{1}{3}$. To address it under the noise model, we design ADDATP that achieves provable approximation guarantee. We later optimize ADDATP into HATP that has made remarkable improvement on efficiency. HATP is designed based on the concept of hybrid error that could efficiently handle different nodes with various expected

marginal spread by adjusting the relative error and additive error adaptively. To evaluate the performance of HATP, we conduct extensive experiments on real social networks, and the experimental results strongly confirm the superiorities and effectiveness of our approaches.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we study three problems on influence propagation in online social networks, all of which are originated from viral marketing and find a plethora of applications in real world.

Firstly, we study the adaptive influence maximization (IM) problem. We propose the first practical algorithm for this problem that could provide both the expected and worst-case approximation guarantee of $1 - e^{\alpha(\varepsilon-1)}$, where $\varepsilon \in (0, 1)$ is a user-specified parameter and $\alpha = 1 - (1 - 1/b)^b$ is set by the batch size b . Our approach is based on a novel framework ADAPTGREEDY. Specifically, ADAPTGREEDY is a general framework that enables us to construct strong approximation solutions for adaptive IM instantiated by any existing non-adaptive IM with expected approximation guarantee. To this end, we propose a novel non-adaptive IM algorithm EPIC which not only provides strong expected approximation guarantee, but also presents superior performance compared with the existing IM algorithms. Meanwhile, we clarify some existing misunderstandings in recent work and shed light on further study of the adaptive IM problem. We conduct experiments on real social networks to evaluate our proposed algorithms comprehensively, and the experimental results strongly corroborate the superiorities and effectiveness of our approach.

Secondly, we investigate the adaptive seed minimization (SM) problem. We propose a novel algorithm, ASTI, which addresses the adaptive seed minimization problem in $O\left(\frac{\eta(m+n)}{\varepsilon^2} \ln n\right)$ expected time and offers an approximation guarantee of $\frac{(\ln \eta + 1)^2}{(1-1/e)(1-\varepsilon)}$ in

expectation, where η is the targeted number of influenced nodes and $\varepsilon \in (0, 1)$ is a user-specified parameter. To the best of our knowledge, ASTI is the first algorithm that provides such an approximation guarantee without incurring prohibitive computation overhead. At the same time, we extend TRIM into a batched version TRIM-B that selects a batch of b nodes in each round, so as to accelerate seed selection. ASTI instantiated by TRIM-B provides an approximation guarantee of $\frac{(\ln \eta + 1)^2}{(1 - (1 - 1/b)^b)(1 - 1/e)(1 - \varepsilon)}$ with the same time complexity as TRIM. Finally, we conduct extensive experiments on a variety of datasets, and demonstrate the effectiveness and efficiency of ASTI over competing methods.

Thirdly, we explore the adaptive target profit maximization (TPM) problem. TPM is a generalization of profit maximization and to the best of our knowledge, we are the first to study the profit maximization problem in the adaptive setting. We study adaptive TPM under both the oracle model and the noise model, and propose ADG and ADDATP algorithms for these two models with strong theoretical guarantees, respectively. Moreover, we propose hybrid error to better handle the sampling errors for nodes with various expected spread, based on which we design algorithm HATP that boosts the efficiency of ADDATP significantly. Finally, we design a series of experiments on real social networks to evaluate the performance of proposed algorithms, and the experimental results strongly confirm the superiorities and effectiveness of our solutions.

6.2 Future Directions

During the study of the above three problems, we realize that there exist potential limitations in the current problem definitions and algorithms. To address those limitations, we propose the following three future directions. In particular, the proposals in section 6.2.1 and section 6.2.2 are applicable for all the three problems, and discussion in section 6.2.3 is specialized for the (target) profit maximization problem.

6.2.1 Sample Reuse without Bias

Recall that the fundamental challenge to address these three problems lies in seed selection. To identify the most suitable seed set, we adopt the random RR (resp. mRR)

set to evaluate the influence spread (resp. truncated influence spread) of each node set. Observe that in the design of these algorithms, we abandon all the samples on seed selection of previous rounds and regenerate samples for current round, as indicated in the pseudocode of the algorithms. The reason behind is that reusing the “old” samples in previous rounds could affect the result of seed selection in current round, thus incurring bias. To explain, each time a seed set is selected, the samples overlapped the selected seed set, denoted as *conflict samples*, would be removed. As a consequence, the coverage distribution of each node set candidate on these remaining samples is highly shaped by the coverage of the selected node set, which unavoidably incurs bias, compared to their exact coverage distribution in the residual graph.

Due to this fact, samples are regenerated for each seed selection round in our algorithm design. As a result, the total running times of the algorithms increase along the number of seed nodes selected, as the experimental results indicate. However, if we could fix this bias issue either by modifying the remaining samples or devising new sampling techniques accordingly to eliminate the potential bias, we could boost the efficiency of those algorithms remarkably by sample reuse. The key challenge to eliminate the bias is to restore the distribution of those random samples each time the conflict samples are discarded. In this regard, we could try to reconstruct those conflict samples instead of discarding them. Take one RR set R as example. First, we remove all the seed nodes selected in current round appeared in R and ignore the resultant isolated nodes disconnecting to the root of R , resulting in R' . Second, we take all the new leaf nodes in R' as root nodes to expand R by following the procedure of generating random RR set. By this method, we might could fix the distribution drift and eliminate the incurred bias.

6.2.2 Online Processing for Dynamic Social Graphs

In our study on these three problems in social networks, we assume that the corresponding social graphs are static and can be processed offline. However, in applications of real world, the graph is highly likely to keep dynamically updated. For example, in social networks, individuals could follow (resp. unfollow) some celebrities or make friends (resp. unfriend) with other users, or new (resp. old) users sign up (resp. quit) some existing social

networks. In this regard, the structures of social graphs are involving along the time. Thus it is critical to reflect the graph alterations for practicability. One straightforward idea is to rerun our algorithms from scratch each time one graph change is detected. However, this can be extremely inefficient due to the prompt graph updates in practice.

Motivated by this requirement, it would be significant of importance to explore an efficient approach to update the affected samples only according to the graph alteration instead of regenerating all samples. To this purpose, One possible way is to build some index structure on samples. For example, each time a sample is constructed, we insert its index into the queues of the nodes in this sample. Once a node is detected removed, we then only eliminate the samples whose indexes are shown in its queue. As such, any tiny graph modifications can be instantly responded. This technique could be really advantageous for the applications demanding online processing for dynamic social graphs.

6.2.3 Profit Maximization Beyond Non-negativity Assumption

With regard to the (target) profit maximization (PM) problem in the literature, algorithms with approximation guarantees are all based on assumptions that either the users in the whole networks or users in certain predefined target set could guarantee the non-negative expected profit so as to ensure the final approximation guarantees. This is due to the fact that the PM problem is an application of unconstrained submodular maximization (USM) and there is no efficient approximation algorithms for USM without any additional assumptions, as pointed out in Section 2.4. However, such kind of assumptions are either impractical or restrictive.

To relieve this requirement, one straightforward thinking is to remove such non-negativity assumptions and explore the theoretical guarantees of the (target) profit maximization problem in general case. To meet this objective, algorithms based on double greedy method are no longer effective as the theoretical guarantee of double greedy is built on such non-negativity assumption. On the contrary, some novel variants of greedy method might help. The idea of greedy method is to select the node providing the maximum marginal profits. However, one potential issue for greedy method is that it is too aggressive to select the profitable nodes in the beginning, which overlooks the fact that

the combination of some less profitable users might bring more profits than those greedily selected individuals. To alleviate this profit damage, we could adaptively calibrate the weights of profit and cost respectively during the process of seed selection by greedy method. In doing so, we might not derive the corresponding approximation ratio to the optimal solution, but could provide some inspirational insights to the profit maximization problem in social graph without any restrictions.

Appendix A

Proofs

We first focus on the proof of the general result in Theorem 3.1 as when the policy is restricted to run once, its worst-case and expected approximation guarantees are the same, which indicates that Theorem 3.1 degenerates to Lemma 3.1. We first introduce some definitions and lemmas, and then use them to prove Theorem 3.1. For convenience, we call any strategy that selects seed nodes based on partial observations as an *adaptive seeding policy*. To analyze the performance ratio of ADAPTGREEDY, we formally define the *expected influence spread* of any adaptive seeding policy in Definition A.1 and *Optimal Marginal Gain* in Definition A.2.

Definition A.1 (Expected Influence Spread). *Given any adaptive seeding policy π , let $S(\pi)$ denote the set of all seed nodes that would be selected by π . The expected influence spread of π is defined as $\Lambda(\pi) = \mathbb{E}_{\Phi \sim \Omega}[\mathbb{E}_{\tau}[I_w(S(\pi))]]$, where ϕ is a realization with respect to the distribution of the universe of realizations Ω , and τ is a possible execution trace of π under ϕ that returns a sequence of seed sets $\{S_1, \dots, S_k\}$.*

Definition A.2 (Optimal Marginal Gain). *For any adaptive policy π , define $G[\pi]$ as the graph generated by removing the nodes activated by π in G . Define $\Delta^*(\pi)$ as the maximum expected influence spread of any b seed nodes in $G[\pi]$, called the optimal marginal gain of π , where the expectation is again taken over the randomness of both realization ϕ and policy π .*

Lemma A.1. *For any adaptive seeding policy π and any $i \geq 1$, we have*

$$\Lambda(\pi_{[i]}) - \Lambda(\pi_{[i-1]}) \leq \Delta^*(\pi_{[i-1]}). \quad (\text{A.1})$$

Lemma A.1. By the definition, for a given ϕ and an execution trace τ of π , the sequence of residue graph $\{G_1, \dots, G_k\}$ is unique. Therefore, there exists a unique distribution of $G_i \sim \mathcal{G}_i$ satisfying

$$\begin{aligned} \Delta^*(\pi_{[i-1]}) &= \sum_{\phi, \tau} (\text{OPT}_b(G_i) \cdot \Pr[\phi, \tau]) \\ &= \sum_{G_i} (\text{OPT}_b(G_i) \cdot \Pr[G_i]) \\ &= \mathbb{E}_{G_i \sim \mathcal{G}_i} [\text{OPT}_b(G_i)]. \end{aligned} \tag{A.2}$$

Similarly,

$$\begin{aligned} \Lambda(\pi_{[i]}) - \Lambda(\pi_{[i-1]}) &= \sum_{\phi, \tau} ((I_w(S(\pi_{[i]})) - I_w(S(\pi_{[i-1]}))) \cdot \Pr[\phi, \tau] \\ &= \sum_{G_i} \left(\sum_{S_i} (\mathbb{E}[I_{G_i}(S_i)] \cdot \Pr[S_i | G_i]) \cdot \Pr[G_i] \right) \\ &= \mathbb{E}_{G_i \sim \mathcal{G}_i} [\mathbb{E}_{S_i} [\mathbb{E}[I_{G_i}(S_i)]]], \end{aligned} \tag{A.3}$$

where S_i is the size- b seed set chosen from the i -th residue graph G_i and $\mathbb{E}[I_{G_i}(S_i)]$ is the expected spread of S_i on G_i . In addition, for any S_i and G_i , it always hold that

$$\mathbb{E}[I_{G_i}(S_i)] \leq \text{OPT}_b(G_i). \tag{A.4}$$

Putting it all together completes the proof. $\square$ $\square$

Lemma A.2. *Given any adaptive seeding policy π and any $i \leq j$, we have*

$$\Delta^*(\pi_{[j]}) \leq \Delta^*(\pi_{[i]}). \tag{A.5}$$

Proof. Note that under a given ϕ and τ , the nodes activated by $\pi_{[i]}$ are always activated by $\pi_{[j]}$ due to $i \leq j$, which indicates that G_{j+1} is always a subgraph of G_{i+1} . As a result,

$$\begin{aligned} \Delta^*(\pi_{[j]}) &= \sum_{\phi, \tau} (\text{OPT}_b(G_{j+1}) \cdot \Pr[\phi, \tau]) \\ &\leq \sum_{\phi, \tau} (\text{OPT}_b(G_{i+1}) \cdot \Pr[\phi, \tau]) \\ &= \Delta^*(\pi_{[i]}). \end{aligned}$$

Hence the lemma follows. $\square$ $\square$

Intuitively, the above lemmas reveal an interesting submodular property of any adaptive policy π , i.e., the optimal marginal gain of π satisfies the “diminishing returns” properties under truncation. Using Lemma A.1 and Lemma A.2, we can build a quantitative relationship between ADAPTGREEDY’s expected influence spread and the expected influence spread of the optimal adaptive seeding policy, as shown by Lemma A.3.

Lemma A.3. *Let π^{ag} and π^{opt} denote the ADAPTGREEDY policy and the optimal adaptive seeding policy, respectively. For any $1 \leq i \leq r$, we have*

$$\Lambda(\pi_{[i]}^{\text{ag}}) - \Lambda(\pi_{[i-1]}^{\text{ag}}) \geq \frac{\alpha(1-\varepsilon_i)}{r} (\Lambda(\pi^{\text{opt}}) - \Lambda(\pi_{[i-1]}^{\text{ag}})), \quad (\text{A.6})$$

where $\alpha = 1 - (1 - 1/b)^b$.

Proof. For any $1 \leq i \leq r$, we have

$$\begin{aligned} & \Lambda(\pi^{\text{opt}}) - \Lambda(\pi_{[i-1]}^{\text{ag}}) \\ & \leq \Lambda(\pi_{[i-1]}^{\text{ag}} \oplus \pi^{\text{opt}}) - \Lambda(\pi_{[i-1]}^{\text{ag}}) \end{aligned} \quad (\text{A.7})$$

$$\begin{aligned} & = \sum_{j=1}^r \left(\Lambda(\pi_{[i-1]}^{\text{ag}} \oplus \pi_{[j]}^{\text{opt}}) - \Lambda(\pi_{[i-1]}^{\text{ag}} \oplus \pi_{[j-1]}^{\text{opt}}) \right) \\ & \leq \sum_{j=1}^r \Delta^*(\pi_{[i-1]}^{\text{ag}} \oplus \pi_{[j-1]}^{\text{opt}}) \end{aligned} \quad (\text{A.8})$$

$$\leq \sum_{j=1}^r \Delta^*(\pi_{[i-1]}^{\text{ag}}) \quad (\text{A.9})$$

$$= r \cdot \Delta^*(\pi_{[i-1]}^{\text{ag}}), \quad (\text{A.10})$$

where (A.7) is due to the monotonicity of $I_w(\cdot)$, (A.8) is due to Lemma A.1, and (A.9) is due to Lemma A.2. On the other hand, we have

$$\begin{aligned} & \Lambda(\pi_{[i]}^{\text{ag}}) - \Lambda(\pi_{[i-1]}^{\text{ag}}) \\ & = \mathbb{E}_{G_i \sim \mathcal{G}_i} [\mathbb{E}_{S_i} [\mathbb{E}[I_{G_i}(S_i)]]] \\ & \geq \mathbb{E}_{G_i \sim \mathcal{G}_i} [\alpha(1 - \mathbb{E}[\xi(G_i)]) \cdot \text{OPT}_b(G_i)] \end{aligned} \quad (\text{A.11})$$

$$\geq \alpha(1 - \varepsilon_i) \cdot \mathbb{E}_{G_i \sim \mathcal{G}_i} [\text{OPT}_b(G_i)] \quad (\text{A.12})$$

$$= \alpha(1 - \varepsilon_i) \cdot \Delta^*(\pi_{[i-1]}^{\text{ag}}), \quad (\text{A.13})$$

where (A.11) is due to the reason that ADAPTGREEDY achieves an $\alpha(1 - \mathbb{E}[\xi(G_i)])$ expected approximation guarantee in each batch, and (A.12) is due to that $\mathbb{E}[\xi(G_i)] \leq \varepsilon_i$. The lemma follows by combining (A.10) and (A.13). $\square$ $\square$

Intuitively, Lemma A.3 reveals that the expected marginal gain of ADAPTGREEDY in each batch i “covers” a sufficiently large portion of the optimal expected influence spread, which leads to the proof of the ADAPTGREEDY’s approximation ratio:

Theorem 3.1. According to Lemma A.3, Equation (A.6) can be rewritten to

$$\Lambda(\pi^{\text{opt}}) - \Lambda(\pi_{[i]}^{\text{ag}}) \leq \left(1 - \frac{\alpha(1-\varepsilon_i)}{r}\right) \cdot \left(\Lambda(\pi^{\text{opt}}) - \Lambda(\pi_{[i-1]}^{\text{ag}})\right)$$

Moreover, for any x such that $0 \leq x \leq 1$, we have $1 - x \leq e^{-x}$. Therefore,

$$\begin{aligned} \Lambda(\pi^{\text{opt}}) - \Lambda(\pi_{[r]}^{\text{ag}}) &\leq e^{(\varepsilon_r-1)\alpha/r} \left(\Lambda(\pi^{\text{opt}}) - \Lambda(\pi_{[r-1]}^{\text{ag}})\right) \\ &\leq \dots \\ &\leq e^{\sum_{i=1}^r (\varepsilon_i-1)\alpha/r} \left(\Lambda(\pi^{\text{opt}}) - \Lambda(\pi_{[0]}^{\text{ag}})\right) \\ &= e^{\alpha(\varepsilon-1)} \cdot \Lambda(\pi^{\text{opt}}). \end{aligned}$$

Rearranging it completes the proof. $\square$ $\square$

Lemma 3.1. Assume that all the notations used for proof Theorem 3.1 are random variables by considering policy π as a random variable. Similar to proof in Lemma A.3, (A.10) still holds, i.e.,

$$\Lambda(\pi^{\text{opt}}) - \Lambda(\pi_{[i-1]}^{\text{ag}}) \leq r \cdot \Delta^*(\pi_{[i-1]}^{\text{ag}}). \quad (\text{A.14})$$

In addition, (A.13) will be slightly changed such that

$$\begin{aligned} &\Lambda(\pi_{[i]}^{\text{ag}}) - \Lambda(\pi_{[i-1]}^{\text{ag}}) \\ &= \mathbb{E}_{G_i \sim \mathcal{G}_i}[\mathbb{E}[I_{G_i}(S_i)]] \\ &\geq \mathbb{E}_{G_i \sim \mathcal{G}_i}[\alpha(1 - \xi(G_i)) \cdot \text{OPT}_b(G_i)] \\ &= \alpha(1 - X_i) \cdot \mathbb{E}_{G_i \sim \mathcal{G}_i}[\text{OPT}_b(G_i)] \end{aligned} \quad (\text{A.15})$$

$$= \alpha(1 - X_i) \cdot \Delta^*(\pi_{[i-1]}^{\text{ag}}), \quad (\text{A.16})$$

where (A.15) is by the definition of X_i . Combing (A.14) and (A.16) gives

$$\Lambda(\pi_{[i]}^{\text{ag}}) - \Lambda(\pi_{[i-1]}^{\text{ag}}) \geq \frac{\alpha(1 - X_i)}{r} \left(\Lambda(\pi^{\text{opt}}) - \Lambda(\pi_{[i-1]}^{\text{ag}}) \right).$$

Recursively, we have

$$\begin{aligned} \Lambda(\pi^{\text{opt}}) - \Lambda(\pi_{[r]}^{\text{ag}}) &\leq e^{(X_r-1)\alpha/r} \left(\Lambda(\pi^{\text{opt}}) - \Lambda(\pi_{[r-1]}^{\text{ag}}) \right) \\ &\leq \dots \\ &\leq e^{\sum_{i=1}^r (X_i-1)\alpha/r} \left(\Lambda(\pi^{\text{opt}}) - \Lambda(\pi_{[0]}^{\text{ag}}) \right) \\ &= e^{\alpha(X-1)} \cdot \Lambda(\pi^{\text{opt}}). \end{aligned}$$

Rearranging it completes the proof. $\square$ $\square$

In what follows, we show some useful martingale concentration bounds, i.e., the Chernoff-like bounds [11] and their variants [17].

Lemma A.4 ([11]). *Let $X_1 - \mathbb{E}[X_1], \dots, X_T - \mathbb{E}[X_T]$ be a martingale difference sequence such that $X_i \in [0, 1]$ for each i . Let $\bar{X} = \frac{1}{T} \sum_{i=1}^T X_i$. If $\mathbb{E}[X_i]$ is identical for every i , i.e., $\mathbb{E}[X_i] = \mathbb{E}[\bar{X}]$, then for any $\lambda \geq 0$, we have*

$$\Pr[\bar{X} > \mathbb{E}[\bar{X}] + \lambda] \leq \exp\left(-\frac{\lambda^2 T}{2\mathbb{E}[\bar{X}] + 2\lambda/3}\right), \quad (\text{A.17})$$

$$\Pr[\bar{X} < \mathbb{E}[\bar{X}] - \lambda] \leq \exp\left(-\frac{\lambda^2 T}{2\mathbb{E}[\bar{X}]}\right). \quad (\text{A.18})$$

Lemma A.5 ([17]). *Let $X_1 - \mathbb{E}[X_1], \dots, X_T - \mathbb{E}[X_T]$ be a martingale difference sequence such that $X_i \in [0, 1]$ for each i . Let $\bar{X} = \frac{1}{T} \sum_{i=1}^T X_i$. If $\mathbb{E}[X_i]$ is identical for every i , i.e., $\mathbb{E}[X_i] = \mathbb{E}[\bar{X}]$, then for any $\lambda \geq 0$, we have*

$$\Pr\left[\mathbb{E}[\bar{X}] \cdot T < \left(\sqrt{\bar{X}T + \frac{2\lambda}{9}} - \sqrt{\frac{\lambda}{2}}\right)^2 - \frac{\lambda}{18}\right] \leq e^{-\lambda}, \quad (\text{A.19})$$

$$\Pr\left[\mathbb{E}[\bar{X}] \cdot T > \left(\sqrt{\bar{X}T + \frac{\lambda}{2}} + \sqrt{\frac{\lambda}{2}}\right)^2\right] \leq e^{-\lambda}. \quad (\text{A.20})$$

Then, we introduce the following lemma that is used to prove Theorem 4.1.

Lemma A.6 ([41]). *If function Γ satisfies all the following conditions:*

- *there exists Q such that $\Gamma_\phi(V) = Q$ for all ϕ ;*

- Γ is integer-valued;
- Γ is self-certifying;
- Γ is strong adaptive monotone;
- Γ is strong adaptive submodular;

then an α -approximate greedy policy π achieves an approximation ratio of $\frac{(\ln \eta + 1)^2}{\alpha}$.

Proof of Theorem 4.1. Obviously, $\Gamma_\phi(V) = \eta$ for all ϕ and Γ is an integer-valued function. Now, we need to prove that for any $v \in V_i$, $\phi, \phi' \in \Omega_i$, and $j \leq i$

$$\Gamma_\phi(S_{i-1}) = \Gamma_\phi(V) \text{ if and only if } \Gamma_{\phi'}(S_{i-1}) = \Gamma_{\phi'}(V), \quad (\text{A.21})$$

$$\Gamma_\phi(v \mid S_{i-1}) \geq 0, \quad (\text{A.22})$$

$$\Delta(v \mid S_{j-1}) \geq \Delta(v \mid S_{i-1}), \quad (\text{A.23})$$

$$\Delta(v \mid S_{j-1}; S_{i-1}) \geq \Delta(v \mid S_{i-1}), \quad (\text{A.24})$$

where $\Delta(v \mid S_{j-1}; S_{i-1}) := \mathbb{E}_{\Phi \sim \Omega_i}[\Gamma_\Phi(v \mid S_{j-1})]$. Equation (A.21) represents self-certifying, Equation (A.22) describes strong monotonicity, Equations (A.23) and (A.24) capture strong adaptive submodularity.

Equation (A.21) obviously holds, i.e., if $\Gamma_\phi(S_{i-1}) = \Gamma_\phi(V) = \eta$, we must have $\Gamma_{\phi'}(S_{i-1}) = \eta = \Gamma_{\phi'}(V)$, and vice versa.

Equation (A.22) holds naturally as “selecting more nodes never hurts” the function Γ .

Next, we prove Equation (A.23). Let ϕ_i be a realization of G_i with probability $p(\phi_i)$ according to the influence propagation. Let $\Omega_j(\phi_i)$ be the subset realizations of Ω_j that are consistent with ϕ_i . That is, for every $\phi \in \Omega_j(\phi_i)$ and every edge $e \in E_i$, the statuses of e are the same in ϕ and ϕ_i such that both are either live or blocked. Then, for any ϕ_i ,

$$\sum_{\phi \in \Omega_j(\phi_i)} p(\phi) = p(\phi_i).$$

In addition, for any $\phi \in \Omega_i$, let $V_\phi(v \mid S_{i-1})$ be the set of nodes activated by v in G_i . Thus, $|V_\phi(v \mid S_{i-1})|$ is the spread of v in G_i under realization ϕ . As a consequence, the marginal truncated spread of v in G_i under ϕ is

$$\Gamma_\phi(v \mid S_{i-1}) = \min\{|V_\phi(v \mid S_{i-1})|, \eta_i\}.$$

Similarly, for any $\phi \in \Omega_j$, we have

$$\Gamma_\phi(v \mid S_{j-1}) = \min\{|V_\phi(v \mid S_{j-1})|, \eta_j\} \geq \Gamma_\phi(v \mid S_{i-1}),$$

where the inequality is due to $G_i \subseteq G_j$ and $\eta_i \leq \eta_j$. Therefore,

$$\begin{aligned} \Delta(v \mid S_{j-1}) &= \sum_{\phi \in \Omega_j} \Gamma_\phi(v \mid S_{j-1}) \cdot p(\phi) \\ &\geq \sum_{\phi \in \Omega_j} \Gamma_\phi(v \mid S_{i-1}) \cdot p(\phi) \\ &= \sum_{\phi_i} \sum_{\phi \in \Omega_j(\phi_i)} \Gamma_\phi(v \mid S_{i-1}) \cdot p(\phi) \\ &= \sum_{\phi_i} \Gamma_\phi(v \mid S_{i-1}) \cdot \sum_{\phi \in \Omega_j(\phi_i)} p(\phi) \\ &= \sum_{\phi_i} \Gamma_\phi(v \mid S_{i-1}) \cdot p(\phi_i) \\ &= \Delta(v \mid S_{i-1}). \end{aligned}$$

Finally, we prove Equation (A.24). For any $\phi \in \Omega_i$, we have

$$\Gamma_\phi(v \mid S_{j-1}) \geq \Gamma_\phi(v \mid S_{i-1}).$$

Taking the expectation over $\Phi \sim \Omega_i$ completes the proof. $\square$

Proof of Theorem 4.2. We prove the elementary version of Equation (4.7), i.e., for any given realization ϕ ,

$$(1 - 1/e)\Gamma_\phi(S) \leq \mathbb{E}[\tilde{\Gamma}_\phi(S)] \leq \Gamma_\phi(S),$$

where the expectation is only taken over the randomness of root size K .

Let $x = I_\phi(S)$ denote the number of nodes influenced by S under ϕ . Let $p(x)$ be the probability that none of the k nodes sampled can be influenced by S , which is given by

$$p(x) := \Pr[\tilde{\Gamma}_\phi(S) = 0] = \binom{n-x}{k} / \binom{n}{k} = \prod_{i=0}^{k-1} \frac{n-x-i}{n-i}.$$

Then, by the definition of $\tilde{\Gamma}_\phi(S)$, with probability $p(x)$, $\tilde{\Gamma}_\phi(S) = 0$; and with probability $1 - p(x)$, $\tilde{\Gamma}_\phi(S) = \eta$. As a consequence, we have

$$\mathbb{E}[\tilde{\Gamma}_\phi(S)] = \eta(1 - \mathbb{E}[p(x)]), \tag{A.25}$$

where the expectation on the right hand side is taken with respect to the randomness of k . Let $f(x)$ be the ratio of $\mathbb{E}[\tilde{\Gamma}_\phi(S)]$ to $\Gamma_\phi(S)$, which is given by

$$f(x) := \frac{\mathbb{E}[\tilde{\Gamma}_\phi(S)]}{\Gamma_\phi(S)} = \frac{\eta(1 - \mathbb{E}[p(x)])}{\min\{x, \eta\}}.$$

Now, we need to prove that $1 - 1/e \leq f(x) \leq 1$. We consider the following two scenarios:

(i) $x \geq \eta$, and (ii) $x < \eta$.

(i) $x \geq \eta$: In this case, $f(x) = 1 - \mathbb{E}[p(x)] \leq 1$. Meanwhile,

$$\begin{aligned} \mathbb{E}[p(x)] &= r \prod_{i=0}^{k^\perp} \frac{n-x-i}{n-i} + (1-r) \prod_{i=0}^{k^\perp-1} \frac{n-x-i}{n-i} \\ &= \left(\frac{r(n-x-k^\perp)}{n-k^\perp} + (1-r) \right) \prod_{i=0}^{k^\perp-1} \frac{n-x-i}{n-i} \\ &= \left(1 - \frac{rx}{n-k^\perp} \right) \prod_{i=0}^{k^\perp-1} \frac{n-x-i}{n-i}. \end{aligned}$$

As $1 - y \leq e^{-y}$ for any $0 \leq y \leq 1$, in the above equation,

$$r.h.s. \leq e^{-rx/(n-k^\perp)} \prod_{i=0}^{k^\perp-1} e^{-x/(n-i)} \leq e^{-(rx/n + k^\perp x/n)} = e^{-x/\eta}.$$

As $x \geq \eta$ by assumption, this implies that $f(x) \geq 1 - 1/e$.

(ii) $x < \eta$: In this case, $f(x) = \eta(1 - \mathbb{E}[p(x)]) / x$. Take the derivative,

$$f'(x) = \frac{\eta}{x^2} (\mathbb{E}[p(x)] - 1 - x\mathbb{E}[p'(x)]).$$

Let $g(x) = p(x) - 1 - xp'(x)$. Take the derivative, when $x > 0$,

$$g'(x) = p'(x) - p'(x) - xp''(x) = -xp''(x).$$

According to the definition of $p(x)$, we can get that

$$p''(x) = \sum_{i=0}^{k-1} \sum_{j=0, j \neq i}^{k-1} \frac{p(x)}{(n-x-i)(n-x-j)} \geq 0.$$

Thus, $g(x)$ decreases with x , which indicates that $g(x) \leq g(0) = 0$. This implies that $f'(x) \leq 0$. As a consequence,

$$f(x) \leq f(1) = \eta(1 - \mathbb{E}[p(1)]) = \eta(1 - \mathbb{E}[(n-k)/n]) = 1,$$

$$f(x) \geq f(\eta) = 1 - \mathbb{E}[p(\eta)] \geq 1 - 1/e,$$

where the last step above follows from the analysis for the case $x \geq \eta$, by considering the special case $x = \eta$.

Hence, the theorem is proved. $\square$

Proof of Corollary 4.1. Equation (4.8) follows directly from Theorem 4.2. By Equation (4.8),

$$\begin{aligned} \mathbb{E}[\Gamma(S \mid S_{i-1})] &\geq \mathbb{E}[\tilde{\Gamma}(S \mid S_{i-1})], \\ \frac{1}{\mathbb{E}[\Gamma(S' \mid S_{i-1})]} &\geq \frac{1 - 1/e}{\mathbb{E}[\tilde{\Gamma}(S' \mid S_{i-1})]}. \end{aligned}$$

Hence, Equation (4.9) holds. $\square$

Proof of Lemma 4.1. We consider the special case of the adaptive seed minimization problem in which the probability $p(e) = 1$ for each edge $e \in E$. In this case, for any node $v \in V$, the set of nodes influenced by v is the set of nodes that can be reached by v in G , denoting as the *cover set* S_v . Thus, for each node $v \in V$, its cover set S_v is deterministic. As a consequence, the adaptive seed minimization problem reduces to a *set cover* problem, i.e., aiming to find as few nodes as possible to cover at least η nodes. Feige [57] has shown that no polynomial time algorithm can approximate the optimal solution of set cover within a ratio of $(1 - \varepsilon) \ln \eta$ for any $\varepsilon > 0$ unless $\text{NP} \subseteq \text{DTIME}(n^{O(\log \log n)})$. Hence, lemma 4.1 holds on noting that ASM generalizes set cover. $\square$

Proof of Lemma 4.2. Let $\mathcal{E}$ be the following event:

$$\mathcal{E}(v^*): \mathbb{E}[\tilde{\Gamma}(v^* \mid S_{i-1})] \geq (1 - \varepsilon) \mathbb{E}[\tilde{\Gamma}(v^\circ \mid S_{i-1})].$$

Note that v^* is the seed node returned by the policy which is a random variable. Let U_t be the set of possible seed nodes selected (but not necessarily returned) by TRIM in the t -th iteration in which each node $u \in U_t$ has a probability $\Pr[u]$ such that $\sum_{u \in U_t} \Pr[u] = 1$, where $1 \leq t \leq T$. Let U_t^* denote the set of random seed nodes returned at the t -th iteration of TRIM, where $U_t^* \subseteq U_t$. Therefore, the event $\mathcal{E}(v^*)$ does not happen only if there exists a node $v_t^* \in U_t^*$ at iteration $t \in [1, T]$ satisfying that $\mathcal{E}(v_t^*)$ does not happen.

If TRIM stops at the iteration $t = T$, according to the setting of $\theta_{\max}$ and by [11], we have

$$\Pr[(t = T) \wedge \neg \mathcal{E}(v_t^*)] \leq \delta/3. \quad (\text{A.26})$$

If TRIM stops at the iteration $t < T$, for any node $v \in V_i$, we define two events $\mathcal{E}_1(v)$ and $\mathcal{E}_2(v)$ as

$$\begin{aligned} \mathcal{E}_1(v) : \mathbb{E}[\Lambda_{\mathcal{R}}(v)] &\geq (\sqrt{\Lambda_{\mathcal{R}}(v) + 2a_1/9} - \sqrt{a_1/2})^2 - a_1/18, \\ \mathcal{E}_2(v) : \mathbb{E}[\Lambda_{\mathcal{R}}(v)] &\leq (\sqrt{\Lambda_{\mathcal{R}}(v) + a_2/2} + \sqrt{a_2/2})^2. \end{aligned}$$

where $\mathbb{E}[\Lambda_{\mathcal{R}}(v)] = |\mathcal{R}| \cdot \mathbb{E}[\tilde{\Gamma}(v \mid S_{i-1})]/\eta_i$ is the expected coverage of v in $\mathcal{R}$. Then, if v is independent of $\mathcal{R}$, by Lemma A.5, we have

$$\Pr[\neg \mathcal{E}_1(v)] \leq \frac{\delta}{3Tn_i}, \quad (\text{A.27})$$

$$\Pr[\neg \mathcal{E}_2(v)] \leq \frac{\delta}{3T}. \quad (\text{A.28})$$

By a union bound that ensures all the n_i nodes satisfying Equation (A.27), we have

$$\begin{aligned} \Pr[\neg \mathcal{E}_1(v_t^*)] &= \sum_{u \in U_t^*} \Pr[\neg \mathcal{E}_1(u)] \cdot \Pr[u] \\ &\leq \sum_{u \in U_t} \Pr[\neg \mathcal{E}_1(u)] \cdot \Pr[u] \\ &\leq \sum_{u \in U_t} \delta/(3T) \cdot \Pr[u] \\ &= \delta/(3T) \end{aligned}$$

Meanwhile, v° is independent of $\mathcal{R}$ naturally. Thus, together with the fact that $\Lambda_{\mathcal{R}}(v^\circ) \leq \Lambda_{\mathcal{R}}(v^*)$, by Equation (A.28)

$$\Pr[\mathbb{E}[\Lambda_{\mathcal{R}}(v^\circ)] > \Lambda^u(v^\circ)] \leq \Pr[\neg \mathcal{E}_2(v^\circ)] \leq \delta/(3T).$$

As a consequence, when TRIM stops at $\Lambda^l(v_t^*)/\Lambda^u(v^\circ) \geq 1 - \hat{\varepsilon}$, if the event $\mathcal{E}(v_t^*)$ does not happen, then at least one of the events $\mathcal{E}_1(v_t^*)$ and $\mathcal{E}_2(v^\circ)$ does not happen. Thus, the event $\mathcal{E}(v_t^*)$ does not happen for all $t < T$ with probability at most:

$$\Pr\left[\bigvee_{t=1}^{T-1} \neg \mathcal{E}(v_t^*)\right] \leq (T-1)\left(\frac{\delta}{3T} + \frac{\delta}{3T}\right) \leq \frac{2\delta}{3}. \quad (\text{A.29})$$

Combining Equations (A.26) and (A.29) shows that the event $\mathcal{E}(v^*)$ holds with probability at least $1 - \delta$. Thus, together with the Equation (4.9) in Corollary 4.1, we have

$$\begin{aligned} \mathbb{E}\left[\frac{1}{\Delta(v^* \mid S_{i-1})}\right] &\leq \left(\frac{1 - \delta}{(1 - 1/e)(1 - \hat{\varepsilon})} + \delta \cdot \eta_i\right) \cdot \frac{1}{\Delta(v^\circ \mid S_{i-1})} \\ &\leq \frac{1}{(1 - 1/e)(1 - \varepsilon)} \cdot \frac{1}{\Delta(v^\circ \mid S_{i-1})}. \end{aligned}$$

Hence, the lemma is proved. $\square$

Proof of Lemma 4.3. For any node v , v is not visited by a random mRR-set R if and only if $v \notin R$. The probability for not visiting v under a realization ϕ is $p(x_v)$, where $x_v = I_\phi(v \mid S_{i-1})$ is the number of nodes that can be activated by v in G_i under ϕ . On the other hand, if a node is visited, all of its incoming edges will be examined. Let indeg_v denote the number of v 's incoming edges. Then, the expected time complexity for generating a random mRR-set is

$$\sum_{v \in V} \text{indeg}_v \cdot \mathbb{E}[1 - p(x_v)], \quad (\text{A.30})$$

where the expectation is over the randomness of both k and Φ . In addition, we already know that

$$\mathbb{E}[\eta_i(1 - p(x_v))] = \mathbb{E}[\tilde{\Gamma}(v \mid S_{i-1})] \leq \text{OPT}_i, \quad (\text{A.31})$$

Combining (A.30) and (A.31) gives

$$\sum_{v \in V} \text{indeg}_v \cdot \mathbb{E}[1 - p(x_v)] \leq \sum_{v \in V} \text{indeg}_v \frac{\text{OPT}_i}{\eta_i} = \frac{\text{OPT}_i}{\eta_i} m_i.$$

Hence, the lemma is proved. $\square$

Proof of Lemma 4.4. Let $\varepsilon_1 = \hat{\varepsilon}/2$ and ε_2 be the root of

$$\varepsilon_2 = \sqrt{\frac{a(2 + 2\varepsilon_2/3)}{\mathbb{E}[\Lambda_{\mathcal{R}}(v^*)]}},$$

where $a = c \ln(4n_i T / \delta)$ for any $c \geq 1$ and $\delta = 1/n_i$. Let

$$\theta^* := \frac{12\eta_i \ln(4n_i T / \delta)}{(1 - \hat{\varepsilon})\hat{\varepsilon}^2 \text{OPT}_i}.$$

As $\hat{\varepsilon} = O(\varepsilon)$, one can verify that $\theta^* = O\left(\frac{\eta_i \ln n_i}{\varepsilon^2 \text{OPT}_i}\right)$.¹ Define the events $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_4$ as follows:

$$\begin{aligned}\mathcal{E}_1 &= \{\Lambda_{\mathcal{R}}(v^\diamond) \geq (1 - \varepsilon_1) \mathbb{E}[\Lambda_{\mathcal{R}}(v^\diamond)]\}, \\ \mathcal{E}_2 &= \{\Lambda_{\mathcal{R}}(v^*) \leq \mathbb{E}[\Lambda_{\mathcal{R}}(v^*)] + \varepsilon_1 \mathbb{E}[\Lambda_{\mathcal{R}}(v^\diamond)]\}, \\ \mathcal{E}_3 &= \{\Lambda_{\mathcal{R}}(v^*) \geq (1 - \varepsilon_2) \cdot \mathbb{E}[\Lambda_{\mathcal{R}}(v^*)]\}, \\ \mathcal{E}_4 &= \{\Lambda_{\mathcal{R}}(v^*) \leq (1 + \varepsilon_2) \cdot \mathbb{E}[\Lambda_{\mathcal{R}}(v^*)]\}.\end{aligned}$$

Then, when a number of $|\mathcal{R}| = c\theta^*$ mRR-sets are generated, by Lemma A.4, it is easy to verify that any event $\mathcal{E}_j$ ($1 \leq j \leq 4$) does not happen with probability at most

$$\Pr[\neg \mathcal{E}_j] \leq (\delta/4)^c.$$

By the union bound, the probability that all the events $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_4$ happen is at least $1 - \delta^c$.

If the events $\mathcal{E}_1, \mathcal{E}_2$ happen,

$$\begin{aligned}\mathbb{E}[\Lambda_{\mathcal{R}}(S^*)] &\geq \Lambda_{\mathcal{R}}(v^*) - \varepsilon_1 \cdot \mathbb{E}[\Lambda_{\mathcal{R}}(v^\diamond)] \\ &\geq \Lambda_{\mathcal{R}}(v^\diamond) - \varepsilon_1 \cdot \mathbb{E}[\Lambda_{\mathcal{R}}(v^\diamond)] \\ &\geq (1 - \varepsilon_1) \cdot \mathbb{E}[\Lambda_{\mathcal{R}}(v^\diamond)] - \varepsilon_1 \cdot \mathbb{E}[\Lambda_{\mathcal{R}}(v^\diamond)] \\ &= (1 - \hat{\varepsilon}) \cdot \mathbb{E}[\Lambda_{\mathcal{R}}(v^\diamond)], \\ &= (1 - \hat{\varepsilon}) \cdot \frac{\text{OPT}_i}{\eta_i} |\mathcal{R}|.\end{aligned}$$

Thus, we have

$$\varepsilon_2 = \sqrt{\frac{a(2 + 2\varepsilon_2/3)}{\mathbb{E}[\Lambda_{\mathcal{R}}(v^*)]}} \leq \sqrt{\frac{(2 + 2\varepsilon_2/3)\hat{\varepsilon}^2}{12}} < \hat{\varepsilon}/2. \quad (\text{A.32})$$

In addition, let

$$\Lambda_l := (1 + \varepsilon_2) \cdot \mathbb{E}[\Lambda_{\mathcal{R}}(v^*)].$$

According to the definition of ε_2 , we have

$$\mathbb{E}[\Lambda_{\mathcal{R}}(v^*)] = \left(\sqrt{\Lambda_l + 2a/9} - \sqrt{a/2}\right)^2 - a/18.$$

¹Without loss of generality, we assume $\hat{\varepsilon} \leq 0.5$. If $\hat{\varepsilon} > 0.5$, TRIM achieves a higher approximation of 0.5 with $O\left(\frac{\eta_i \ln n_i}{\text{OPT}_i}\right)$ mRR-sets.

Since $a_1 \leq a$, if event $\mathcal{E}_4$ happens (i.e., $\Lambda_{\mathcal{R}}(v^*) \leq \Lambda_l$), then

$$\begin{aligned} & \Lambda^l(v^*) - \mathbb{E}[\Lambda_{\mathcal{R}}(v^*)] \\ & \geq (\sqrt{\Lambda_{\mathcal{R}}(v^*) + 2a/9} - \sqrt{a/2})^2 - a/18 - \mathbb{E}[\Lambda_{\mathcal{R}}(v^*)] \\ & \geq \Lambda_{\mathcal{R}}(v^*) - \Lambda_l. \end{aligned}$$

As a consequence, if event $\mathcal{E}_3$ also happens, we have

$$\Lambda^l(v^*) \geq \Lambda_{\mathcal{R}}(v^*) - \varepsilon_2 \mathbb{E}[\Lambda_{\mathcal{R}}(v^*)] \geq \frac{1 - 2\varepsilon_2}{1 - \varepsilon_2} \Lambda_{\mathcal{R}}(v^*). \quad (\text{A.33})$$

Similarly, let

$$\Lambda_u := (1 - \varepsilon_2) \cdot \mathbb{E}[\Lambda_{\mathcal{R}}(v^*)].$$

According to the definition of ε_2 , we have

$$\mathbb{E}[\Lambda_{\mathcal{R}}(v^*)] \geq (\sqrt{\Lambda_u + a/2} + \sqrt{a/2})^2.$$

Since $a_2 \leq a$, if event $\mathcal{E}_3$ happens (i.e., $\Lambda_{\mathcal{R}}(v^*) \geq \Lambda_u$), then

$$\begin{aligned} \frac{\Lambda^u(v^\circ)}{\Lambda_u} & \leq \frac{(\sqrt{\Lambda_{\mathcal{R}}(v^*) + a/2} + \sqrt{a/2})^2}{(1 - \varepsilon_2) \cdot \mathbb{E}[\Lambda_{\mathcal{R}}(v^*)]} \\ & \leq \frac{\Lambda_{\mathcal{R}}(v^*)}{(1 - \varepsilon_2) \cdot \Lambda_u}. \end{aligned}$$

As a consequence, we have

$$\Lambda^u(v^\circ) \leq \frac{\Lambda_{\mathcal{R}}(v^*)}{1 - \varepsilon_2}. \quad (\text{A.34})$$

Putting it all together of (A.32), (A.33) and (A.34), we have

$$\frac{\Lambda^l(v^*)}{\Lambda^u(v^\circ)} \geq \frac{(1 - 2\varepsilon_2)\Lambda_{\mathcal{R}}(v^*)}{1 - \varepsilon_2} \cdot \frac{1 - \varepsilon_2}{\Lambda_{\mathcal{R}}(v^*)} \geq 1 - \hat{\varepsilon}.$$

Therefore, when a number of $c\theta^*$ mRR-sets are generated, TRIM does not stop only if at least one of the events in $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_4$ does not happen, with probability at most δ^c .

Let t be the first iteration that the number of mRR-sets generated by TRIM reaches θ^* such that $2^{t-2} \cdot \theta_\circ < \theta^*$ and $2^{t-1} \cdot \theta_\circ \geq \theta^*$. From this iteration onward, the expected

number of mRR-sets further generated is at most

$$\begin{aligned}
 \sum_{z \geq t} \theta_{\circ} \cdot 2^{z-1} \cdot \delta^{2^{z-t}} &= 2^{t-1} \cdot \theta_{\circ} \sum_{z=0} 2^z \cdot \delta^{2^z} \\
 &\leq 2\theta^* \sum_{z=0} 2^{-2^z+z} \\
 &\leq 2\theta^* \sum_{z=0} 2^{-z} \\
 &\leq 4\theta^*.
 \end{aligned}$$

The first inequality is due to $2^{t-1} \cdot \theta_{\circ} < 2\theta^*$ and $\delta \leq 1/2$, and the second inequality is due to $-2^z + z \leq -z$. If the algorithm stops before the t -th iteration, there are at most θ^* random samples generated. Therefore, the expected number of random samples generated is less than $5\theta^*$, which is $O(\frac{\eta_i \ln n_i}{\varepsilon^2 \text{OPT}_i})$.

Hence, the lemma is proved. $\square$

Proof of Lemma 4.5. The time complexity of TRIM is determined by that for generating mRR-sets. By Wald's equation [55], the expected total time used for generating mRR-sets equals the expected number of mRR-sets generated, times the expected time used for generating one mRR-set. Thus, according to Lemmas 4.3 and 4.4, the expected time complexity of TRIM is $O(\frac{m_i+n_i}{\varepsilon^2} \ln n_i)$. $\square$

Proof of Lemma 4.6. Let S^* be the seed set returned by the batched policy with $|S^*| = b$ and S° be the corresponding optimal seed set in the i -th round. Let $\mathcal{E}_b$ be the following event:

$$\mathcal{E}_b(S^*): \mathbb{E}[\tilde{\Gamma}(S^* \mid S_{i-1})] \geq \rho_b(1 - \varepsilon) \mathbb{E}[\tilde{\Gamma}(S^{\circ} \mid S_{i-1})].$$

Let S_t^* be the generalized definition of v_t^* in Section 4.2.5. If S^* is returned at T -th iteration, based on the setting of T and by [11], we still have

$$\Pr[(t = T) \wedge \neg \mathcal{E}_b(S_t^*)] \leq \delta/3. \quad (\text{A.35})$$

If TRIM-B stops at the iteration $t < T$, for any node $S \subseteq V_i$ obtained by greedy method with $|S| = b$, we define two events $\mathcal{E}_{b,1}(S)$ and $\mathcal{E}_{b,2}(v)$ as

$$\begin{aligned}
 \mathcal{E}_{b,1}(S) : \mathbb{E}[\Lambda_{\mathcal{R}}(S)] &\geq (\sqrt{\Lambda_{\mathcal{R}}(S) + 2a_1/9} - \sqrt{a_1/2})^2 - a_1/18, \\
 \mathcal{E}_{b,2}(S) : \mathbb{E}[\Lambda_{\mathcal{R}}(S)] &\leq (\sqrt{\Lambda_{\mathcal{R}}(S)/\rho_b + a_2/2} + \sqrt{a_2/2})^2.
 \end{aligned}$$

where $\mathbb{E}[\Lambda_{\mathcal{R}}(S)] = |\mathcal{R}| \cdot \mathbb{E}[\tilde{\Gamma}(S \mid S_{i-1})]/\eta_i$ is the expected coverage of S in $\mathcal{R}$.

Based on Lemma A.5, we could have

$$\Pr[-\mathcal{E}_{b,1}(S)] \leq \frac{\delta}{3T \binom{n_i}{b}}. \quad (\text{A.36})$$

Similarly, by union bound for all $\binom{n_i}{b}$ candidates of size- b node set, we could immediately have

$$\Pr[-\mathcal{E}_{b,1}(S_t^*)] \leq \delta/(3T). \quad (\text{A.37})$$

Let $S_{\mathcal{R}}^\circ$ be the size- b seed set that could cover largest number of mRR-sets in $\mathcal{R}$. Since S is derived by Greedy method from $\mathcal{R}$, by the property of greedy method, we have $\Lambda_{\mathcal{R}}(S) \geq \rho_b \Lambda_{\mathcal{R}}(S_{\mathcal{R}}^\circ) \geq \rho_b \Lambda_{\mathcal{R}}(S^\circ)$. Then $\Lambda_{\mathcal{R}}(S)/\rho_b$ can be taken as the upper bound of $\Lambda_{\mathcal{R}}(S^\circ)$. Similarly, by Lemma A.5, we have following equation

$$\Pr[-\mathcal{E}_{b,2}(S^\circ)] \leq \delta/(3T). \quad (\text{A.38})$$

By following the analysis in Section 4.2.5, we acquire the fact that event $\mathcal{E}_b$ holds with at least $1 - \delta$ probability where $\delta = 1/n_i$. By Corollary 4.1, the expected approximation ratio of TRIM-B is at least

$$(1 - \delta) \cdot (1 - \varepsilon) \cdot \rho_b \cdot (1 - 1/e) = \rho_b(1 - 1/e)(1 - \varepsilon).$$

Hence, the lemma is proved. $\square$

Proof of Lemma 5.2. Consider the case that $\rho_f \geq \rho_r$. Then, we have $S_i = S_{i-1} \cup \{u_i\}$, $T_i = T_{i-1}$, and $S_i^\circ = S_{i-1}^\circ \cup \{u_i\}$. Thus, (5.5) becomes $-\Delta_{G_i}(u_i \mid S_{i-1}^\circ) \leq \Delta_{G_i}(u_i \mid S_{i-1}) = \rho_f$. If $u_i \notin S_{i-1}^\circ$, we have $-\Delta_{G_i}(u_i \mid S_{i-1}^\circ) \leq -\Delta_{G_i}(u_i \mid T_{i-1} \setminus \{u_i\}) = \rho_r \leq \rho_f$ due to the submodularity of $\rho_{G_i}(\cdot)$ and the fact that $S_{i-1}^\circ \subseteq T_{i-1} \setminus \{u_i\}$ and $u_i \notin T_{i-1} \setminus \{u_i\}$. If $u_i \in S_{i-1}^\circ$, we have $-\Delta_{G_i}(u_i \mid S_{i-1}^\circ) = 0 \leq \rho_f$ since $\rho_f + \rho_r \geq 0$ by Lemma 5.1. Thus, it holds that $-\Delta_{G_i}(u_i \mid S_{i-1}^\circ) \leq \rho_f$.

The other case of $\rho_f < \rho_r$ is analogous. Specifically, we have $S_i = S_{i-1}$, $T_i = T_{i-1} \setminus \{u_i\}$, and $S_i^\circ = S_{i-1}^\circ \setminus \{u_i\}$ and need to show that $\rho_{G_i}(S_{i-1}^\circ) - \rho_{G_i}(S_i^\circ) \leq \rho_r$. Again, if $u_i \notin S_{i-1}^\circ$, we have $\rho_{G_i}(S_{i-1}^\circ) - \rho_{G_i}(S_i^\circ) = 0 \leq \rho_r$ since $\rho_f + \rho_r \geq 0$. If $u_i \in S_{i-1}^\circ$, we have $\rho_{G_i}(S_{i-1}^\circ) - \rho_{G_i}(S_i^\circ) = \Delta_{G_i}(u_i \mid S_{i-1}^\circ \setminus \{u_i\}) \leq \Delta_{G_i}(u_i \mid S_{i-1}) = \rho_f < \rho_r$ since $S_{i-1} \subseteq S_{i-1}^\circ \setminus \{u_i\}$ and $u_i \notin S_{i-1}^\circ \setminus \{u_i\}$.

Hence, the lemma is proved. $\square$

Proof of Lemma 5.3. For simplicity, let $S_i := S_\phi(\pi_{[i]}^f)$, $T_i := S_\phi(\pi_{[i]}^r)$, and $S_i^\circ := S_\phi(\pi_{[i]}^\circ)$. By definition, we have

$$\Lambda(\pi_{[i]}^f) - \Lambda(\pi_{[i-1]}^f) = \sum_{\phi} (\rho_\phi(S_i) - \rho_\phi(S_{i-1})) \cdot p(\phi), \quad (\text{A.39})$$

$$\Lambda(\pi_{[i]}^r) - \Lambda(\pi_{[i-1]}^r) = \sum_{\phi} (\rho_\phi(T_i) - \rho_\phi(T_{i-1})) \cdot p(\phi), \quad (\text{A.40})$$

$$\Lambda(\pi_{[i]}^\circ) - \Lambda(\pi_{[i-1]}^\circ) = \sum_{\phi} (\rho_\phi(S_i^\circ) - \rho_\phi(S_{i-1}^\circ)) \cdot p(\phi). \quad (\text{A.41})$$

Let $\mathcal{G}_i$ be the distribution of i -th residual graph with respect to the ADG policy, where each residual graph $G_i \in \mathcal{G}_i$ has a probability of $\Pr[G_i]$. Then, the realization space Ω can be disjointedly partitioned with respect to each $G_i \in \mathcal{G}_i$. For each G_i , we denote the corresponding set of realizations as $\Omega(G_i)$, which implies that $\Pr[G_i] = \sum_{\phi \in \Omega(G_i)} p(\phi)$. Moreover, for any S and G_i , we have

$$\rho_{G_i}(S) = \sum_{\phi \in \Omega(G_i)} \rho_\phi(S) \cdot \frac{p(\phi)}{\Pr[G_i]}.$$

Then, (A.39)–(A.41) can be rewritten as

$$\begin{aligned} \Lambda(\pi_{[i]}^f) - \Lambda(\pi_{[i-1]}^f) &= \sum_{G_i} (\rho_{G_i}(S_i) - \rho_{G_i}(S_{i-1})) \cdot \Pr[G_i], \\ \Lambda(\pi_{[i]}^r) - \Lambda(\pi_{[i-1]}^r) &= \sum_{G_i} (\rho_{G_i}(T_i) - \rho_{G_i}(T_{i-1})) \cdot \Pr[G_i], \\ \Lambda(\pi_{[i]}^\circ) - \Lambda(\pi_{[i-1]}^\circ) &= \sum_{G_i} (\rho_{G_i}(S_i^\circ) - \rho_{G_i}(S_{i-1}^\circ)) \cdot \Pr[G_i]. \end{aligned}$$

Putting it all together with Lemma 5.2 completes the proof. □

Proof of Lemma 5.5. Let f_{est} and r_{est} be the estimations of $\mathbb{E}[I_{G_i}(u_i \mid S_{i-1})]$ and $\mathbb{E}[I_{G_i}(u_i \mid T_{i-1} \setminus \{u_i\})]$. For the i -th iteration of ADDATP and the j -th round estimations of expected marginal spreads (or profits), we define two events $\mathcal{E}_{i,j}^1$ and $\mathcal{E}_{i,j}^2$ as

$$\begin{aligned} \mathcal{E}_{i,j}^1 &: |\mathbb{E}[I_{G_i}(u_i \mid S_{i-1})] - f_{\text{est}}| \leq n_i \zeta_i, \\ \mathcal{E}_{i,j}^2 &: |\mathbb{E}[I_{G_i}(u_i \mid T_{i-1} \setminus \{u_i\})] - r_{\text{est}}| \leq n_i \zeta_i. \end{aligned}$$

Regarding the event $\mathcal{E}_{i,j}^1$ or $\mathcal{E}_{i,j}^2$, we know that $\delta_i = \frac{1}{2^{j-1}kn}$ and $\theta = \frac{\ln(8/\delta_i)}{2\zeta_i^2}$. Thus, by Lemma 5.4, we have

$$\Pr[\neg\mathcal{E}_{i,j}^1 \vee \neg\mathcal{E}_{i,j}^2] \leq \Pr[\neg\mathcal{E}_{i,j}^1] + \Pr[\neg\mathcal{E}_{i,j}^2] \leq 4e^{-2\theta\zeta_i^2} = \frac{1}{2^j kn}.$$

As a result, we have

$$\Pr\left[\bigvee_{j=1}^{\infty} (\neg\mathcal{E}_{i,j}^1 \vee \neg\mathcal{E}_{i,j}^2)\right] \leq \sum_{j=1}^{\infty} \frac{1}{2^j kn} = \frac{1}{kn}. \quad (\text{A.42})$$

In what follows, we assume that events $\mathcal{E}_{i,j}^1$ and $\mathcal{E}_{i,j}^2$ happen for every j . Thus, we have $\tilde{\rho}_f - n_i\zeta_i \leq \rho_f \leq \tilde{\rho}_f + n_i\zeta_i$ and $\tilde{\rho}_r - n_i\zeta_i \leq \rho_r \leq \tilde{\rho}_r + n_i\zeta_i$. We first consider the case that C_1 occurs. If $\tilde{\rho}_f \geq \tilde{\rho}_r$, we have $\tilde{\rho}_f - \tilde{\rho}_r \geq 2n_i\zeta_i$ or $\tilde{\rho}_r \leq -n_i\zeta_i$ since $\tilde{\rho}_f + \tilde{\rho}_r \geq 0$ (using the same argument in the proof of Lemma 5.1). For the former, $\rho_f - \rho_r \geq \tilde{\rho}_f - \tilde{\rho}_r - 2n_i\zeta_i \geq 0$, while for the latter, $\rho_r \leq 0$ and hence $\rho_f \geq \rho_r$ as $\rho_f + \rho_r \geq 0$ by Lemma 5.1. Thus, $\rho_f \geq \rho_r$ always hold if $\tilde{\rho}_f \geq \tilde{\rho}_r$. With a similar analysis, $\rho_f < \rho_r$ also holds if $\tilde{\rho}_f < \tilde{\rho}_r$. Then, when C_1 occurs, by Lemma 5.2, we have

$$\begin{aligned} & \rho_{G_i}(S_{i-1}^\circ) - \rho_{G_i}(S_i^\circ) \\ & \leq \rho_{G_i}(S_i) - \rho_{G_i}(S_{i-1}) + \rho_{G_i}(T_i) - \rho_{G_i}(T_{i-1}). \end{aligned}$$

Now, we consider the case that C_2 occurs. Again, if $\tilde{\rho}_f \geq \tilde{\rho}_r$, we have $\rho_r \leq \tilde{\rho}_r + n_i\zeta_i \leq \tilde{\rho}_f + n_i\zeta_i \leq \rho_f + 2n_i\zeta_i \leq \rho_f + 2$. On the other hand, one can verify that $\rho_{G_i}(S_{i-1}^\circ) - \rho_{G_i}(S_i^\circ) \leq \max\{\rho_f, \rho_r\} \leq \rho_f + 2 = \rho_{G_i}(S_i) - \rho_{G_i}(S_{i-1}) + 2$ and $\rho_{G_i}(T_i) - \rho_{G_i}(T_{i-1}) = 0$. Similarly, if $\tilde{\rho}_f < \tilde{\rho}_r$, we have $\rho_{G_i}(S_{i-1}^\circ) - \rho_{G_i}(S_i^\circ) \leq \max\{\rho_f, \rho_r\} \leq \rho_r + 2 \leq \rho_{G_i}(T_i) - \rho_{G_i}(T_{i-1}) + 2$ and $\rho_{G_i}(S_i) - \rho_{G_i}(S_{i-1}) = 0$. Then,

$$\begin{aligned} & \rho_{G_i}(S_{i-1}^\circ) - \rho_{G_i}(S_i^\circ) - 2 \\ & \leq \rho_{G_i}(S_i) - \rho_{G_i}(S_{i-1}) + \rho_{G_i}(T_i) - \rho_{G_i}(T_{i-1}), \end{aligned} \quad (\text{A.43})$$

which also holds for the case that C_1 occurs.

On the other hand, suppose that at least one of the events $\mathcal{E}_{i,j}^1$ and $\mathcal{E}_{i,j}^2$ does not happen. We know that $\rho_{G_i}(S_{i-1}^\circ) - \rho_{G_i}(S_i^\circ) \leq \max\{\rho_f, \rho_r\} \leq n$ and $\rho_{G_i}(S_i) - \rho_{G_i}(S_{i-1}) + \rho_{G_i}(T_i) - \rho_{G_i}(T_{i-1}) \geq \min\{\rho_f, \rho_r\} \geq -n$. Thus, it always holds that

$$\begin{aligned} & \rho_{G_i}(S_{i-1}^\circ) - \rho_{G_i}(S_i^\circ) - 2n \\ & \leq \rho_{G_i}(S_i) - \rho_{G_i}(S_{i-1}) + \rho_{G_i}(T_i) - \rho_{G_i}(T_{i-1}), \end{aligned} \quad (\text{A.44})$$

Combining (A.42), (A.43), and (A.44) completes the proof. $\square$

Proof of Lemma 5.6. We first fix a random seed for ADDATP such that ADDATP becomes a deterministic algorithm. Using a similar argument in the proof of Lemma 5.3, we have

$$\begin{aligned}\Lambda(\pi_{[i]}^f) - \Lambda(\pi_{[i-1]}^f) &= \sum_{G_i} (\rho_{G_i}(S_i) - \rho_{G_i}(S_{i-1})) \cdot \Pr[G_i], \\ \Lambda(\pi_{[i]}^r) - \Lambda(\pi_{[i-1]}^r) &= \sum_{G_i} (\rho_{G_i}(T_i) - \rho_{G_i}(T_{i-1})) \cdot \Pr[G_i], \\ \Lambda(\pi_{[i]}^o) - \Lambda(\pi_{[i-1]}^o) &= \sum_{G_i} (\rho_{G_i}(S_i^\circ) - \rho_{G_i}(S_{i-1}^\circ)) \cdot \Pr[G_i].\end{aligned}$$

Then, taking the expectation on both sides over the internal randomness of the ADDATP algorithm and combining with Lemma 5.5 complete the proof. $\square$

Proof of Lemma 5.8. The proof is analogous to those of Lemmas 5.5 and 5.6. The key step is to figure out the bound on profit loss when C_2' occurs, i.e., $n_i \zeta_i \leq 1$ and $\varepsilon_i \leq \varepsilon$, and the estimation is sufficiently accurate (with probability $1 - 1/(kn)$), i.e., $\frac{f_{\text{est}} - n_i \zeta_i}{1 + \varepsilon_i} - c(u_i) \leq \rho_f \leq \frac{f_{\text{est}} + n_i \zeta_i}{1 - \varepsilon_i} - c(u_i)$ and $c(u_i) - \frac{r_{\text{est}} + n_i \zeta_i}{1 - \varepsilon_i} \leq \rho_r \leq c(u_i) - \frac{r_{\text{est}} - n_i \zeta_i}{1 + \varepsilon_i}$. If $f_{\text{est}} + r_{\text{est}} \geq 2c(u_i)$, one can verify that $\rho_r \leq \frac{f_{\text{est}} - n_i \zeta_i}{1 + \varepsilon} - c(u_i) + \frac{2(n_i \zeta_i + \varepsilon c(u_i))}{1 - \varepsilon} \leq \rho_f + \frac{2(1 + \varepsilon c(u_i))}{1 - \varepsilon}$. Therefore, we have $\rho_{G_i}(S_{i-1}^\circ) - \rho_{G_i}(S_i^\circ) \leq \max\{\rho_f, \rho_r\} \leq \rho_f + \frac{2(1 + \varepsilon c(u_i))}{1 - \varepsilon} = \rho_{G_i}(S_i) - \rho_{G_i}(S_{i-1}) + \frac{2(1 + \varepsilon c(u_i))}{1 - \varepsilon}$ and $\rho_{G_i}(T_i) - \rho_{G_i}(T_{i-1}) = 0$. Similarly, if $f_{\text{est}} + r_{\text{est}} < 2c(u_i)$, we have $\rho_{G_i}(S_{i-1}^\circ) - \rho_{G_i}(S_i^\circ) \leq \max\{\rho_f, \rho_r\} \leq \rho_r + \frac{2(1 + \varepsilon c(u_i))}{1 - \varepsilon} \leq \rho_{G_i}(T_i) - \rho_{G_i}(T_{i-1}) + \frac{2(1 + \varepsilon c(u_i))}{1 - \varepsilon}$ and $\rho_{G_i}(S_i) - \rho_{G_i}(S_{i-1}) = 0$. Following the same arguments in the proofs of Lemmas 5.5 and 5.6, the lemma is proved. $\square$

Publications

1. **Keke Huang**, Jing Tang, Xiaokui Xiao, Aixin Sun, and Andrew Lim “Efficient Approximation Algorithms for Adaptive Target Profit Maximization”, IEEE International Conference on Data Engineering (**ICDE**), 2020
2. Jing Tang*, **Keke Huang*** (Co-first author), Xiaokui Xiao, Laks V.S. Lakshmanan, Xueyan Tang, Aixin Sun, and Andrew Lim “Efficient Approximation Algorithms for Adaptive Seed Minimization.”, Proceedings of the ACM SIGMOD International Conference on Management of Data (**SIGMOD**), 2019.
3. Kai Han, Yuntian He, **Keke Huang**, Xiaokui Xiao, Shaojie Tang, Jingxin Xu, Liusheng Huang “Best Bang for the Buck: Cost-Effective Seed Selection for On-line Social Networks.”, IEEE Transactions on Knowledge and Data Engineering (**TKDE**), 2019.
4. **Keke Huang**, Sibow Wang, Glenn Bevilacqua, Xiaokui Xiao, and Laks V.S. Lakshmanan. “Revisiting the Stop-and-Stare Algorithms for Influence Maximization.”, Proceedings of the VLDB Endowment (**PVLDB**), 2017

Papers under submission

1. **Keke Huang**, Jing Tang, Kai Han, Xiaokui Xiao, Wei Chen, Aixin Sun, Xueyan Tang, and Andrew Lim “Efficient Approximation Algorithms for Adaptive Influence Maximization”, Submitted to *The International Journal on Very Large Data Bases* (**VLDBJ**), 2020

References

- [1] L. Sun, W. Huang, P. S. Yu, and W. Chen, “Multi-round influence maximization,” in *Proc. ACM KDD*, pp. 2249–2258, 2018.
- [2] K. Han, K. Huang, X. Xiao, J. Tang, A. Sun, and X. Tang, “Efficient algorithms for adaptive influence maximization,” *Proc. VLDB Endowment*, vol. 11, no. 9, pp. 1029–1040, 2018.
- [3] Statista, “Most famous social network sites worldwide as of April 2019,” April 2019.
- [4] D. Kempe, J. Kleinberg, and E. Tardos, “Maximizing the spread of influence through a social network,” in *Proc. ACM KDD*, pp. 137–146, 2003.
- [5] P. Zhang, W. Chen, X. Sun, Y. Wang, and J. Zhang, “Minimizing seed set selection with probabilistic coverage guarantee in a social network,” in *Proc. ACM KDD*, pp. 1306–1315, 2014.
- [6] D. Arthur, R. Motwani, A. Sharma, and Y. Xu, “Pricing strategies for viral marketing on social networks,” in *WINE*, pp. 101–112, 2009.
- [7] W. Lu and L. V. S. Lakshmanan, “Profit maximization over social networks,” in *12th IEEE International Conference on Data Mining, ICDM*, pp. 479–488, 2012.
- [8] A. Badanidiyuru, C. Papadimitriou, A. Rubinstein, L. Seeman, and Y. Singer, “Locally adaptive optimization: Adaptive seeding for monotone submodular functions,” in *Proc. SODA*, pp. 414–429, 2016.
- [9] D. Golovin and A. Krause, “Adaptive submodularity: Theory and applications in active learning and stochastic optimization,” *Journal of Artificial Intelligence Research*, vol. 42, no. 1, pp. 427–486, 2011.

REFERENCES

- [10] Y. Tang, X. Xiao, and Y. Shi, “Influence maximization: Near-optimal time complexity meets practical efficiency,” in *Proc. ACM SIGMOD*, pp. 75–86, 2014.
- [11] Y. Tang, Y. Shi, and X. Xiao, “Influence maximization in near-linear time: A martingale approach,” in *Proc. ACM SIGMOD*, pp. 1539–1554, 2015.
- [12] H. T. Nguyen, M. T. Thai, and T. N. Dinh, “Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks,” in *Proc. ACM SIGMOD*, pp. 695–710, 2016.
- [13] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier, “Maximizing social influence in nearly optimal time,” in *Proc. SODA*, pp. 946–957, 2014.
- [14] H. Li, S. S. Bhowmick, A. Sun, and J. Cui, “Conformity-aware influence maximization in online social networks,” *VLDB Journal*, vol. 24, no. 1, pp. 117–141, 2015.
- [15] J. Tang, X. Tang, and J. Yuan, “Influence maximization meets efficiency and effectiveness: A hop-based approach,” in *Proc. IEEE/ACM ASONAM*, pp. 64–71, 2017.
- [16] J. Tang, X. Tang, and J. Yuan, “An efficient and effective hop-based approach for influence maximization in social networks,” *Social Network Analysis and Mining*, vol. 8, no. 10, 2018.
- [17] J. Tang, X. Tang, X. Xiao, and J. Yuan, “Online processing algorithms for influence maximization,” in *Proc. ACM SIGMOD*, 2018.
- [18] K. Han, Y. He, X. Xiao, S. Tang, F. Gui, C. Xu, and J. Luo, “Budget-constrained organization of influential social events,” in *Proc. IEEE ICDE*, pp. 917–928, 2018.
- [19] K. Han, C. Xu, F. Gui, S. Tang, H. Huang, and J. Luo, “Discount allocation for revenue maximization in online social networks,” in *Proc. ACM MobiHoc*, pp. 121–130, 2018.
- [20] W. Chen, C. Wang, and Y. Wang, “Scalable influence maximization for prevalent viral marketing in large-scale social networks,” in *Proc. ACM KDD*, pp. 1029–1038, 2010.

REFERENCES

- [21] W. Chen, Y. Yuan, and L. Zhang, “Scalable influence maximization in social networks under the linear threshold model,” in *Proc. IEEE ICDM*, pp. 88–97, 2010.
- [22] S. Vaswani and L. V. S. Lakshmanan, “Adaptive influence maximization in social networks: Why commit when you can adapt?,” arXiv preprint, <https://arxiv.org/abs/1604.08171>, 2016.
- [23] A. Goyal, F. Bonchi, L. Lakshmanan, and S. Venkatasubramanian, “On minimizing budget and time in influence propagation over social networks,” *Social Network Analysis and Mining*, vol. 3, no. 2, pp. 179–192, 2013.
- [24] K. Han, Y. He, X. Xiao, S. Tang, J. Xu, and L. Huang, “Cost-effective seed selection for online social networks,” arXiv preprint, <https://arxiv.org/abs/1711.10665>, 2017.
- [25] Y. Zhu, D. Li, R. Yan, W. Wu, and Y. Bi, “Maximizing the influence and profit in social networks,” *IEEE Trans. Comput. Social Systems*, vol. 4, no. 3, pp. 54–64, 2017.
- [26] J. Tang, X. Tang, and J. Yuan, “Profit maximization for viral marketing in online social networks: Algorithms and analysis,” *IEEE Transaction on Knowledge and Data Engineering.*, vol. 30, no. 6, pp. 1095–1108, 2018.
- [27] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.
- [28] K. Huang, S. Wang, G. Bevilacqua, X. Xiao, and L. V. S. Lakshmanan, “Revisiting the stop-and-stare algorithms for influence maximization,” vol. 10, pp. 913–924, 2017.
- [29] W. Chen, Y. Wang, and S. Yang, “Efficient influence maximization in social networks,” in *Proceedings of the 15th ACM SIGKDD*, pp. 199–208, 2009.
- [30] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. M. VanBriesen, and N. S. Glance, “Cost-effective outbreak detection in networks,” in *Proceedings of the 13th ACM SIGKDD*, pp. 420–429, 2007.

REFERENCES

- [31] Y. Li, J. Fan, Y. Wang, and K. Tan, “Influence maximization on social graphs: A survey,” *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 10, pp. 1852–1872, 2018.
- [32] W. Chen, L. V. S. Lakshmanan, and C. Castillo, *Information and Influence Propagation in Social Networks*. Morgan & Claypool Publishers, 2013.
- [33] A. Arora, S. Galhotra, and S. Ranu, “Debunking the myths of influence maximization: An in-depth benchmarking study,” in *in SIGMOD*, pp. 651–666, 2017.
- [34] A. Goyal, W. Lu, and L. V. S. Lakshmanan, “SIMPAT: an efficient algorithm for influence maximization under the linear threshold model,” in *11th IEEE International Conference on Data Mining, ICDM*, pp. 211–220, 2011.
- [35] V. V. Vazirani, *Approximation Algorithms*. Springer, 2003.
- [36] H. T. Nguyen, M. T. Thai, and T. N. Dinh, “Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks.” arXiv preprint, <https://arxiv.org/abs/1605.07990v3>, 22 Feb 2017.
- [37] S. Lei, S. Maniu, L. Mo, R. Cheng, and P. Senellart, “Online influence maximization,” in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 645–654, 2015.
- [38] N. Chen, “On the approximability of influence in social networks,” *SIAM Journal on Discrete Mathematics*, vol. 23, no. 3, pp. 1400–1415, 2009.
- [39] S. Goldberg and Z. Liu, “The diffusion of networking technologies,” in *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pp. 1577–1594, 2013.
- [40] C. Long and R. C.-W. Wong, “Minimizing seed set for viral marketing,” in *Proc. IEEE ICDM*, pp. 427–436, 2011.
- [41] D. Golovin and A. Krause, “Adaptive submodularity: Theory and applications in active learning and stochastic optimization.” arXiv preprint, <https://arxiv.org/abs/1003.3967>, 2017.

REFERENCES

- [42] U. Feige, V. S. Mirrokni, and J. Vondrák, “Maximizing non-monotone submodular functions,” *SIAM J. Comput.*, vol. 40, no. 4, pp. 1133–1153, 2011.
- [43] N. Buchbinder, M. Feldman, J. Naor, and R. Schwartz, “A tight linear time $(1/2)$ -approximation for unconstrained submodular maximization,” in *IEEE, FOCS*, pp. 649–658, 2012.
- [44] A. Asadpour, H. Nazerzadeh, and A. Saberi, “Stochastic submodular maximization,” in *Proc. WINE*, pp. 477–489, 2008.
- [45] A. Badanidiyuru and J. Vondrák, “Fast algorithms for maximizing submodular functions,” in *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, SODA*, pp. 1497–1514, 2014.
- [46] A. A. Ageev and M. Sviridenko, “An 0.828-approximation algorithm for the uncapacitated facility location problem,” *Discrete Applied Mathematics*, vol. 93, no. 2-3, pp. 149–156, 1999.
- [47] M. X. Goemans and D. P. Williamson, “Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming,” *Journal of the ACM*, vol. 42, no. 6, pp. 1115–1145, 1995.
- [48] Z. Lu, V. O. K. Li, and Q. Shuai, “Price competition of spreaders in profit-maximizing sponsored viral marketing,” *IEEE Trans. Comput. Social Systems*, vol. 5, no. 4, pp. 931–941, 2018.
- [49] G. A. Tong, W. Wu, and D. Du, “Coupon advertising in online social systems: Algorithms and sampling techniques,” *CoRR*, vol. abs/1802.06946, 2018.
- [50] B. Liu, X. Li, H. Wang, Q. Fang, J. Dong, and W. Wu, “Profit maximization problem with coupons in social networks,” in *Algorithmic Aspects in Information and Management AAIM*, pp. 49–61, 2018.
- [51] J. Leskovec and A. Krevl, “SNAP Datasets: Stanford large network dataset collection.” <http://snap.stanford.edu/data>, 2014.

REFERENCES

- [52] J. Tang, K. Huang, X. Xiao, L. V. S. Lakshmanan, X. Tang, A. Sun, and A. Lim, “Efficient approximation algorithms for adaptive seed minimization,” in *Proceedings of SIGMOD*, pp. 1096–1113, 2019.
- [53] Y. Chen and A. Krause, “Near-optimal batch mode active learning and adaptive submodular optimization,” in *Proc. ICML*, pp. 160–168, 2013.
- [54] Y. Chen, H. Shioi, C. F. Montesinos, L. P. Koh, S. Wich, and A. Krause, “Active detection via adaptive submodularity,” in *Proc. ICML*, pp. 55–63, 2014.
- [55] A. Wald, *Sequential Analysis*. Wiley, 1947.
- [56] W. Hoeffding, “Probability inequalities for sums of bounded random variables,” *Journal of the American Statistical Association*, vol. 58, pp. 13–30, 1963.
- [57] U. Feige, “A threshold of $\ln n$ for approximating set cover,” *Journal of the ACM*, vol. 45, no. 4, pp. 634–652, 1998.