涉密论文□　公开论文□

浙 江 大 学

本科生毕业论文

题目	协作与竞争并存的社交网络影响最大化问题研究
学生姓名	刘明锐
学生学号	3170105696
指导教师	王灿
年级与专业	17级计算机科学与技术（求是科学班）
所在学院	计算机科学与技术学院
递交日期	2021 年 6 月 2 日

浙江大学本科生毕业论文（设计）承诺书

1. 本人郑重地承诺所呈交的毕业论文（设计），是在指导教师的指导下严格按照学校和学院有关规定完成的。

2. 本人在毕业论文（设计）中除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得浙江大学或其他教育机构的学位或证书而使用过的材料。

3. 与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

4. 本人承诺在毕业论文（设计）工作过程中没有伪造数据等行为。

5. 若在本毕业论文（设计）中有侵犯任何方面知识产权的行为，由本人承担相应的法律责任。

6. 本人完全了解浙江大学有权保留并向有关部门或机构送交本论文（设计）的复印件和磁盘，允许本论文（设计）被查阅和借阅。本人授权浙江大学可以将本论文（设计）的全部或部分内容编入有关数据库进行检索和传播，可以采用影印、缩印或扫描等复制手段保存、汇编本论文（设计）。

作者签名：　　导师签名：

签字日期：　　年　月　日　　签字日期　　年　月　日

致谢

感谢父母对我的养育之恩和对我一直以来的关心支持，他们对我的鼓励是我最大的前进动力。

感谢王灿老师，从 2016 年 NOI 现场到现在毕业论文近 5 年，始终在悉心教导着我。在此向王老师表达由衷的敬意。

感谢史麒豪、陈诗翰、陈靖邦学长，从整个研究到写论文的过程中一直在给予我指导，在研究方向上对我进行帮助。

感谢吴迪同学在论文文字问题和描述问题上给予我的指导和建议，以及对中英文对应部分提出的意见。

感谢浙江大学程序设计竞赛队的每一位队员、教练，在集训队一起训练、熬夜的时光我终生难忘。特别感谢我的两位队友，很荣幸正式在集训队的三年都在 Legilimens 队，祝愿每一位竞赛队的成员前途无量。

感谢求计 16 的学长们在课程学习上给我提供的细心指导；感谢 17、18 级的同学们与我一同学习，考试前穿梭于寝室之间、在机房复习、在咖啡厅刷夜的经历依然历历在目；

快乐的大学时光不仅让我学会了很多知识，也让我学会了很多学习与做人的方法；感谢在浙江大学结识的每一位同学，这致谢的篇幅写不下你们的名字，更写不出我对你们的无比谢意；今后各奔东西，只愿来日方长，后会有期。

感谢 Paradox Interactive，最近的几次更新激励了我按时完成本次毕业设计。

摘要

现代社交网络平台的茁壮发展为我们的生活带来了极大的变化，但同时也对社会发展带来了挑战：在我们的信息传播愈发便利的同时，虚假信息的传播也变得越来越容易。近几年来，学界对社交网络上的信息传播问题进行了深入的研究，通常在某种传播模型下，利用有限资源进行传播控制。主要研究方向分为普通的正面信息传播最大化问题和衍生的负面信息抑制最小化两类问题，然而这类研究通常只有一种信息传播，但在现实情况中，往往需要同时考虑最大化正面信息和最小化负面信息两个问题。本文结合影响最大化问题，谣言限制问题和负面信息问题，提出一个协作与竞争并存的独立级联模型，分析了该问题的复杂度。随后根据具体设置的不同，根据单调性与子模性的四种情况，提出了四种算法 PR-IMM、SA-IMM、RG-IMM、与 SA-RG-IMM。最终通过实验表明这些算法在真实数据上可以得到优秀的信息传播结果。

关键词：社交网络；独立级联模型；信息传播最大化；谣言限制；

Abstract

The thriving development of modern social network platforms has brought great changes to our lives, but at the same time it has also brought challenges to social development: while our information propagation has become more convenient, the propagation of false information has also become more and more convenient. In recent years, academia has conducted in-depth research on influence maximization problem, usually about uding limited resources to maximaize influence. The main research direction is divided into two types: the maximization of positive information propagation and the minimization and suppression of the negative information. However, this type of research usually has only one type of information propagating on the network, but in reality, it is often necessary to consider maximizing positive information and minimizing negative information at the same time. This paper combines the problem of influence maximization, rumors restriction and negative information, and proposes an independent cascade model in which collaboration and competition coexist, and analyzes the complexity of the problem. Then according to different settings, the four conditions of monotonicity and sub-modularity, four algorithms PR-IMM, SA-IMM, RG-IMM, and SA-RG-IMM are proposed. Finally, we conducted experiments show that these algorithms can obtain excellent information dissemination results on real data.

Keywords: Social Network; Indepedent Cascade Model; Influence Maximization; Rumor Restriction;

目录

第一部分　毕业论文

第一部分

毕业论文

1 绪论

1.1 研究背景

近几年来的科技进步，特别是交流方式方面的进步，大大地提升了社交网络的发展，信息交流与信息传播的速度也越来越快，覆盖面广泛提高，在生活中的重要性也愈发显著。今天的各种数字社交平台几乎完全掌控了我们的社交生活，我们不论是日常交流，还是广播想法，都会使用这一类的社交平台，包括国内的QQ、微信，国外的 facebook、twitter 等等。但如此强大的社交网络总会有两面性：在其提高了我们传播信息和广告推销的效率的同时，它也提高了虚假消息和垃圾广告推送给我们的可能性。这两者的例子在今天屡见不鲜，正面例子如社交平台给我们的生活与企业的发展带来了极大的便利；反面的例子有新冠疫情爆发带来了大量虚假信息，比如各类土方治疗、疫情谣言在网络上疯狂传播[1]，这些恶意信息给社会带来了多方面的影响。由于社交网络的重要性，如何设计算法利用有限资源来获取或限制传播一直都是研究的热点。

为了获取有效信息的传播，制止无效信息的扩散，首先我们要了解社交网络上信息传播的主要方式。最初的社交网络由电子邮件构成，此时的信息传播以点对点为主，通话以单个邮件的形式建立。今天，信息传播主要是瀑布形扩散的，包括论坛、微博、朋友圈一类扩散性的信息传播，使得今天的信息传播问题变得更加复杂。如今单个信息源在适当条件下可能影响到的个人以指数级扩张，使得分析的难度愈发增加。

在这一现实情况下，学术界对如何在合理的模型内保证正面信息传播、抑制负面信息传播进行了大量的研究。[2] 在保证正向信息方面，通常是视为选择一定量的种子用户，通过用户互相之间的传播达成尽可能地扩散指定广告或产品的效果。在减少负面信息方面，通常是选择有限的用户阻断或者令其获取辟谣信息，来阻挡负面信息的传播。总的来说，都是一个选择用户进行影响，以达到效果最大化的问题。

一般研究的模型下，社交网络被视为一张有着边权的有向图，节点表示一个社交网络中的用户，用户之间的有向边表示着他们的关系，如邮件关系，朋友圈等等，权值代表这一关系的强度。用户之间的信息传播由一些阈值或者变化触

发。通过这样的抽象建模，我们可以将社交网络中的信息传播问题理解成一个图上传播的概率问题。特别地说，还有更加细节的模型，用以对应不同的情况，比如独立级联（Independent Cascade Model）模型[3] 中用户一旦第一次接触到信息就会触发传播；线性阈值（Linear Threshold Model）模型[4] 中用户收到足够的信息影响就会出发传播等等。在实际的应用中，应该根据情况选择符合的建模。

1.2 本文研究目标

目前已有的大部分信息传播模型通常都只考虑一类信息的传播，要么都是正面信息进行增强，要么都是负面信息进行抑制。本文提出了一个新的信息传播模型，模型中一个正面信息 C_a 与一个负面信息 C_r 同时进行传播，平台通过初始时选择 k 个关键节点，普通信息传播到增强节点时会发生变化：正面信息会变为 C_a^+，拥有更强的传播概率；负面信息会变为 C_r^-，不再具有反面信息的效果，但仍然正常传播干扰负面信息。四种信息在传播到一个节点后就不会改变状态；同时传播到一个节点时，根据优先级确定顺序，共有 $4! = 24$ 种可能的顺序。

基于 Tang 等人[5] 的 IMM 算法，我们在这个模型下，根据不同优先级的性质确定了单调子模 (M-S)，单调非子模 (M-nS)，非单调子模 (nM-S)，非单调非子模 (nM-nS) 四种情况，分别提出了四种算法：PR-IMM，SA-IMM，RG-IMM，SA-RG-IMM。

我们在已有的 facebook、twitter 数据集[6] 和论文合作网络[7] 上进行实验，和对照算法的实验结果表明我们的算法可以得到良好的正面信息增强、负面信息抑制效果。

1.3 本文结构安排

第二章：介绍本文前置内容，包括问题本身的背景以及问题算法研究的进程。

第三章：给出本文主要问题模型与对应算法及证明。

第四章：通过真实数据上的实验，评估算法效果。

第五章：总结本文，给出该问题未来的发展方向。

2 相关工作

2.1 信息传播模型

通常将社交网络的传播模型抽象为一张有向图 $G(V,E)$。点集 V 代表用户，有向边集 E 代表信息的传递关系，如发送邮件、圈子群发等。每个节点有一个目前接受信息的状态，用来代表用户接受与传播的信息类型。在用户从信息上游接受某个信息后，本文称这个用户目前被激活，处于该信息对应的状态。在这个模型中，认为一个节点变为某种信息的激活状态后，就不能再改变状态。一个接受了信息的节点可以继续影响后续节点，使得信息在社交网络上不断传播。具体的传播方法取决于传播模型，这里本文介绍最常见的两种传播模型：独立级联模型（Independent Cascade Model）[3] 和线性阈值模型（Linear Threshold Model）[4]。

2.1.1 独立级联模型

独立级联模型（Independent Cascade Model）是一种传播模型[3]，其定义一个节点传播的时刻为第一次被激活，既成功接收到信息的时刻。一个节点 u 激活某个后续节点 v 是一个完全独立的随机事件，其成功概率为边权 $P_{u,v}$，与其他随机量无关，又信息在图上级联传播，因此称为独立级联模型。每当某个节点 u 在某个时刻被激活时，其仅在下个时刻尝试激活每个后续节点各一次。具体来说：

1. 初始激活集合 S 中的节点在时刻 0 被激活。

2. 在时刻 t，所有在 $t-1$ 时刻被激活的节点尝试激活其每个后继节点，每次激活成功概率为 $P_{u,v}$。

3. 整个事件最终必然收敛：在某个时刻没有新节点被激活时，整个过程必然结束。

由于激活的独立性，可以得到在每个时刻内不同节点激活事件的互相顺序是无关的。同样可以发现，所有边是否可以成功激活后续节点是一个互相独立的事件，因此整个随机部分可以在传播前全部先确定。这是本文后续算法所使用到的基本性质之一。

2.1.2 线性阈值模型

同样地，线性阈值模型（Linear Threshold Model）是一种传播模型[4]，其在传播开始前对每个节点分配一个随机值 $\theta_v \in (0,1)$，称为该节点的阈值。每当某个节点 u 被激活时，其对其后续节点 v 做 $P_{u,v}$ 的贡献。当一个节点的贡献积累大于等于阈值 θ_v 时，其进入激活状态。具体来说：

1. 初始激活集合 S 中的节点被激活。

2. 统计被激活的节点对每个后继节点的影响 $P_{u,v}$，计算该节点是否被激活。

3. 当没有新的节点被激活时，过程结束。

线性阈值模型和独立级联模型相比，最大的性质区别的无时序性，无顺序性，且传播过程本身无随机性。这些优秀的性质使得线性阈值模型的处理十分简单，可以方便地处理传播过程。

2.2 影响最大化问题

影响最大化问题是社交网络信息传播领域的基本问题，其模型最早由 Kempe 等[8] 在 2003 年提出，这一模型希望找到一组 k 个用户节点，将这些用户作为信息的初始激活集合，使得信息最终影响到的节点数量最大。Kempe 等[8] 证明了不论在独立级联模型还是线性阈值模型下，这个问题都是 NP-hard 的，同时还证明了在初始激活集合 S 下，计算期望最终影响的节点数量 $\sigma(S)$ 是 #P-hard 的。然而，利用这两种模型的单调性和子模性，Nemhauser 等[3] 证明了一个简单的贪心算法有近似比 $(1-1/e-\epsilon)$。这个模型可以轻易地扩展到每个用户带有代价 $c(u)$ 的情况，要求选取用户总代价小于预算 $\sum_{u\in S} c(u) \leq B$ 下最大化影响。通过在普通和最大化效率两种贪心可以获得近似比 $(\frac{1-1/e}{2}-\epsilon)$ 的算法[9][10][11]。近年来影响最大化问题的理论复杂度与实际效均得到了很大的提高，期望复杂度从一开始 Kempe 等[8] 的 $O(krn\theta)$，到 Borgs 等[12] 的 RIS 算法 $O(k(m+n)\log^2 n/\epsilon^2)$，以及下文提到的 Tang 等[5] 的 IMM 算法 $O(k(m+n)\log n/\epsilon^2)$。

2.3 谣言限制问题

谣言限制问题从影响最大化问题衍生而来，由 Kempe 等[13] 于 2005 年提出模型，目标找到一组 k 个用户节点，阻断负面信息到这些点的传播，以最小化谣言影响到的节点集合。Bharathi 等[14] 提出了多种谣言同时传播时一个近似比 $(1-1/e-\epsilon)$ 的算法，而 Borodin 等[4] 讨论了多种谣言传播在线性阈值模型下的情况。在谣言限制模型中，通常认为选中的用户节点产生一个竞争性的辟谣信息传播，这样的模型保持了原问题的单调子模性。在采样方面，Tong 等[15][16][17] 基于独立级联模型，提出了 RBR、HMP 等采样算法。除此之外，Budak 等[18] 证明了在当前节点状态是一个概率值的情况下，原问题仍然是子模的。

2.4 IMM 算法

IMM 算法是一个 Tang 等基于鞅论的影响最大化算法。Tang 等[5] 提出了一个与之前最好的理论复杂度相同，但实际效果更好的算法：IMM 算法在 $O((k+l)(n+m)\log n/\epsilon^2)$ 时间复杂度下以至少 $1-1/n^l$ 的概率返回一个 $1-1/e-\epsilon$ 的近似解。IMM 最大的提升在于运用鞅论的方法，得到了一个最优解 OPT 的下界，保证很高概率不小于 $OPT(1-1/e)/(1+\epsilon')^2$。在 $\epsilon=0.5, l=1$ 下，IMM 算法的效甚至高于目前最好的启发式算法，且仍然拥有一个理论复杂度保证。下文中使用的算法就是在新模型上基于 IMM 算法提出的。

3 研究方案

3.1 模型与问题

3.1.1 协作与竞争并存的影响最大化问题

下面本文结合影响最大化问题[19]，谣言限制问题[18] 和负面信息问题[20]，提出一个协作与竞争并存的独立级联模型（Complementary&Competitive IC model，C^2IC）如下：

该模型假设有两种不同的信息级联在社交网络上传播：一个盟友（正面）信息级联 C_a 和一个对手（负面）信息级联 C_r。该问题希望选取 k 个节点作为增强

节点，目的是最大化正面信息的传播，最小化负面信息的传播。其中，盟友信息级联代表**协作**信息，对手信息级联代表**竞争**信息，两种信息级联在正常情况下和普通独立级联模型传播的行为一致，它们只在传播到增强节点时发生变化，具体来说：

正面信息增强当一个 C_a 信息传播到一个增强节点时，该节点变为 C_a^+ 状态而不是 C_a 状态，并传播增强的 C_a^+ 正面信息。C_a^+ 信息在传播时，边权（本次传播的概率）变为 $p_{uv}^+ \geq p_{uv}$。这一增强模型源于影响最大化问题中，增强节点会修改普通信息以增强其传播概率。

负面信息抑制当一个 C_r 信息传播到一个增强节点时，该节点变为 C_r^- 状态而不是 C_r 状态，并传播抑制的 C_r^- 信息。C_r^- 信息不视为正面或负面信息，但可以通过和其他信息互相排斥来减少负面信息的传播。这一抑制模型源于负面信息中，抑制节点在收到原版信息时会产生抑制信息，抑制原版信息的传播。

显然，这一模型有两种特殊的退化情况：

正面信息增强模型（C^+IC）：社交网络上只有 C_a 和增强节点的情况。

负面信息抑制模型（C^-IC）：社交网络上只有 C_r 和增强节点的情况。

3.1.2 具体建模

通常使用一张有向图 $G(V,E)$ 代表社交网络，点集 V 代表用户，其中有一些属于 S_a，代表 C_a 信息的初始种子节点；有另一些属于 S_r，代表 C_r 信息的初始种子节点，其他两者均不属于。边集 E 代表传播关系，每条边上有两个权值 p_{uv}^+ 和 p_{uv}，$p_{uv}^+ \geq p_{uv}$，代表这条传播关系成功的概率。同时，还有一个优先级顺序，来代表四种不同信息 C_a，C_r，C_a^+，C_r^- 在同时传播到同一个点时的顺序关系。最后，该问题要求在图上的非初始种子节点中选择 k 个增强节点。在第 0 轮，S_a 中的节点激活为 C_a 信息，S_r 中的节点激活为 C_r 信息。

1. 在每一轮中，找到恰在上一轮被激活的节点，根据优先级顺序，从优先级最高的信息到最低的依次尝试往后传播。

2. 每次传播时，若出边指向的节点已经被某个信息激活，则不再尝试传播；否则根据边上信息对应的传播权值，随机得到本次激活是否成功。

3. 当上一轮不再有被新激活的节点时，整个传播过程结束。

3.1.3 目标函数

本文定义在增强节点集合为 S 时，正面信息 C_a 与 C_a^+ 所能传播到的期望节点数为 $\sigma_a(S)$，负面信息 C_r 所能传播到的期望节点数为 $\sigma_r(S)$。

$$f(S) = \sigma_a(S) - \sigma_a()$$

为增强节点提高的期望正面节点数。

$$g(S) = \sigma_r() - \sigma_r(S)$$

为增强节点降低的期望负面节点数。

$$h(S) = \lambda f(S) + (1-\lambda)g(S)$$

为目标函数，其中 $\lambda \in [0,1]$ 为可变参数。

问题 1.3.1 (协作与竞争并存的影响最大化 (C^2IM)) 在 C_a 信息初始节点为 S_a，C_r 信息初始节点为 S_r 的情况下，本问题要求找出一组大小为 k 的增强节点集合 S，使得提高正面节点数和降低负面节点数的混合平均值最大，即

$$S^* = \arg \max_{S \subseteq V \setminus S_a \setminus S_r, |S|=k} h(S).$$

同样地，我们可以得到本问题的两种退化情况：

问题 1.3.2 (正面信息最大化 (C^+IM)) 在 C_a 信息初始节点为 S_a 下，本问题要求找出一组大小为 k 的增强节点集合 S，使得提高的正面节点数最大，即

$$S^* = \arg \max_{S \subseteq V \setminus S_a, |S|=k} f(S).$$

问题 1.3.3 (负面信息最小化 (C^-IM)) 在 C_r 信息初始节点为 S_r 下，本问题要求找出一组大小为 k 的增强节点集合 S，使得降低的负面节点数最大，即

$$S^* = \arg \max_{S \subseteq V \setminus S_r, |S|=k} g(S).$$

下面给出问题的复杂度及其证明。

定理 1.3.1 上述三个问题均至少是 *NP-hard* 的。

证明 1.3.1 尝试将 *NP-hard* 的最大 k 覆盖问题[21] 归约到上述问题。

最大 k 覆盖问题给出一个数字 k 和 m 个非空集合 $C = \{c_1, c_2, ..., c_m\}$，其中每个 C_i 都是 $E = \{e_1, e_2, ..., e_n\}$ 的子集。该问题希望得到一个 C 的大小为 k 的子集 C' 以最大化 $|\cup_{c_i \in C'} c_i|$。

构造一张有向图 $G(V, E)$，对于每个 c_i 和每个 e_j，在 V 中构造一个对应点，为便于说明用同样符号表示。对于每个 $e_j \in c_i$，加入对应边 (e_j, c_i)。最后加入一个超级源 u，并对于每个 c_i，加入对应边 (u, c_i)。这显然是一个多项式复杂度的规约。

对于正面信息最大化问题，令每条边 (u, v) 有 $p + uv = 0, p^+_{uv} = 1$，且设 u 为唯一 C_a 信息初始节点，则该图上的正面信息最大化问题对应到原最大 k 覆盖问题。

对于负面信息最小化问题，令每条边 (u, v) 有 $p + uv = p^+_{uv} = 1$，且设 u 为唯一 C_r 信息初始节点，优先级为 $C_r^- \succ C_r$，则该图上的负面信息最小化问题对应到原最大 k 覆盖问题。

对于协作与竞争并存的影响最大化问题，每个正面信息最大化问题可以归约到 $\lambda = 1$ 的情况，每个负面信息最大化问题可以归约到 $\lambda = 0$ 的情况。

综上所述，上述三个问题均至少为 *NP-hard*。

下面给出目标函数的复杂度及其证明。

定理 1.3.2 上述三个函数均至少是 *#P-hard* 的。

证明 1.3.2 *Kempe* 等[8] 证明了独立级联模型下单点能够影响的期望节点数 $\sigma(s)$ 是 *#P-hard* 的，尝试将其归约到上述问题。

对于一张有向图 $G(V, E)$ 和单点 s，加入点 u, v，边 $(u, v), (v, s)$，设每个原有边的 $p^+_{ij} = p + ij$，并设 $p^+_{uv} = p + uv = 1$。这显然是一个多项式复杂度的规约。

对于增强节点提高的期望正面节点数 f，设 u 为 C_a 信息初始节点，$p^+_{vs} = 1, p + vs = 0$，则 $\sigma(s) = f(v)$。

对于增强节点降低的期望负面节点数 g，设 u 为 C_r 信息初始节点，$p^+_{vs} = p + vs = 1$，则 $\sigma(s) = g(v) - 1$。

对于上述问题的加权平均 h，每个 f 可以归约到 $\lambda = 1$ 的情况，每个 g 可以归约到 $\lambda = 0$ 的情况。

综上所述，上述三个函数均至少为 *#P-hard*。

3.1.4 优先级顺序

四种不同信息 C_a，C_r，C_a^+，C_r^- 的优先级顺序对问题的性质影响很大，本文根据算法需要利用的单调性和子模性两个性质，将 $4! = 24$ 种优先级顺序分为单调子模 (M-S)，单调非子模 (M-nS)，非单调子模 (nM-S)，非单调非子模 (nM-nS) 四种情况。本文将在下一章将讨论具体分类情况与证明。

3.2 算法与证明

3.2.1 反向采样

定义一张社交网络的采样如下：

对于每一条边，根据其权值 p_{uv}，以 p_{uv} 的概率将这条边加入采样，以 $1-p_{uv}$ 的概率拒绝这条边。最后，称所有采样内的边为”存活”的边，这些边构成原社交网络的一个子图。

定理 1.3.3 一种信息 C 在一张社交网络上的期望传播节点数 θ_C 等于其在一个采样 g 上传播节点数的期望，即 $\theta_C = \mathbb{E}[H_{g,C}]$，其中 $H_g(C)$ 是信息 C 在采样 g 上的传播节点数。

证明 1.3.3 *Kempe* 等[8] 给出了该定理的详细证明，此处不再阐述。

根据上述定理，我们有 $h(S) = \mathbb{E}[h_g(S)]$，即原图上的目标函数等于采样上目标函数的期望。

定义一张社交网络的反向采样如下：

等概率随机一个初始节点 u，随后从 u 开始反向 BFS，每次遇到一条边，根据其权值 p_{uv}，以 p_{uv} 的概率将这条边加入采样，以 $1 - p_{uv}$ 的概率拒绝这条边。最后，所有采样内的点为可能对 u 产生影响的点，所有采样内的边为”存活”的边，这些边构成原社交网络的一个子图。

设 $gain(S, R_u)$ 为在初始节点为 u 的反向采样 R 上，点 u 对目标函数的贡献，有 $h(S) = \mathbb{E}[h_g(S)] = \sum_u \mathbb{E}[gain(S, R_u)] = n\mathbb{E}[gain(S, R)]$，即原图上的目标函数等于反向采样上贡献的期望乘点数。因此，我们可以使用反向采样来估算原图上的目标函数。

在这个模型中，由于一条边有两条权值，以 p_{uv} 的概率将这条边以" 存活" 形式加入采样，以 $p^+_{uv} - p_{uv}$ 的概率将这条边以" 增强存活" 形式加入采样，以 $1 - p^+_{uv}$ 的概率拒绝这条边。在反向采样时，尝试首先只走“存活”边，找到最早

算法 1 单边采样

输入: 边 (u, v)

输出: 边状态 $\in$ { 存活，增强存活，拒绝 }

```
if 本边之前已经采样过 then
    return 之前的结果
end if
r = random(0, 1)
if r ≤ p_uv then
    return 存活
else
    if r ≤ p^+_uv then
        return 增强存活
    else
        return 拒绝
    end if
end if
```

可以到达自身的种子节点。这是这个节点被某种信息激活的最晚时间/最长距离，增加增强节点只会减少这个最晚时间/最长距离。我们称这个距离称为界限范围 $LimitDistance$。

接下来在界限范围内尝试走所有“存活”边和“增强存活”边，由于界限范围内才有可能影响这个节点的状态，这些走过的点以及它们之间的边构成一个反向采样。

算法 2 界限范围

输入: 带边权图 $G(V,E)$，顺序优先级 $Rank$，初始点集 S_a, S_r，初始节点 u

输出: 界限范围 $LimitDistance$ 反向采样 R_u，$gain(v, R_u)$

Q 为一队列，初始只包含 u
while Q 非空 **do**
    $t = Q.front()$
    $Q.pop_front()$
    for 每个 t 的前驱 v **do**
        单边采样 (v,u)
        if (v,u) 增强存活或拒绝 **then**
            Continue
        end if
        if v 未被本次 while 访问过 **then**
            $Dis(v) = Dis(t) + 1$
            $Q.push\_back(v)$
            if $v \in S_a$ 或 $v \in S_r$ **then**
                return $LimitDistance = Dis(v)$
            end if
        end if
    end for
end while
return $LimitDistance = n$

算法 3 反向采样（PR-sketch-Generation）

输入: 带边权图 $G(V,E)$，顺序优先级 $Rank$，初始点集 S_a, S_r，初始节点 u，界限范围 $LimitDistance$

输出: 反向采样 R_u

Q 为一队列，初始只包含 u
R_v 为一空集合
while Q 非空 **do**
    $t = Q.front()$
    $Q.pop_front$
    $R_v.push(t)$
    if $LimitDistance \leq Dis(t)$ **then**
        Continue
    end if
    for 每个 t 的前驱 v **do**
        单边采样 (v,u)
        if (v,u) 拒绝 **then**
            Continue
        end if
        if v 未被本次 while 访问过 **then**
            $Dis(v) = Dis(t) + 1$
            $Q.push\_back(v)$
        end if
    end for
end while
return R

当优先级顺序符合单调子模性时，每个点 v 作为增强节点目标函数的增加量 $gain(v, R_u)$ 恰好可以一次性进行计算。一个点 v 作为增强节点使得目标函数增加共有两种情况：

1.u 原本处于 C_r 状态，且在通过“存活”边传播到 u 的 C_r 最短传播路径中，存在一条包含 v 的路径，使得 u 变为 C_r^- 状态。

2.u 原本不处于 C_a 或 C_a^+ 状态，且 C_a 能通过“存活”边传播到 v，并从 v 通过“存活”或“增强存活”边传播到 u，且后者路径上任何一点的达到时间都不晚于其他信息到达同一点时间，使得 u 变为 C_a^+ 状态。

因此，可以先正向模拟出每个点原本在什么时刻接收什么信息，可以得到第一种情况；在此之上反向运行特殊最长路，包含 k 个点的路径 (a, b) 长度是 $\min_{v_i \in (a,b)} [v_i] - k + i$，可以得到第二种情况。

算法 5 给出一个优先级为 $C_a^+ \succ C_a \succ C_r^- \succ C_r$ 时该算法的伪代码，复杂度 $O(|R_e| \log |R_v|)$。

当优先级顺序非单调或非子模时，我们无法快速计算目标函数增加量，因此因此使用一个启发式的正向模拟解决这个问题。具体来说，我们每次将一个点变为增强节点时，它会改变其最终激活状态，而这会改变其后续节点的激活状态，以此类推。

算法 7 给出一个该算法的伪代码。

3.2.2 优先级分类与证明

首先介绍算法需要利用的单调性和子模性两个性质。

单调性指函数随输入增加而不减，即 $f(S \cup u) \leq f(S)$，其中 S 为图上一点集，u 为 S 外一点。

子模性指函数增益随输入增加而不增，即 $f(S \cup u) - f(S) \leq f(T \cup u) - f(T)$，其中 $T \subset S$ 为图上两点集，u 为 S 外一点。

在四种不同信息的 $4! = 24$ 种情况中，单调子模 (M-S)，单调非子模 (M-nS)，非单调子模 (nM-S)，非单调非子模 (nM-nS)，详细情况见表 1。

由于单调性和子模性都有乘正系数保持性质和相加保持性质，又 $\theta(C) = \mathbb{E}[H_g(C)]$ 本质是对所有可能采样的加权平均，可得 $\theta(C)$ 的单调/子模性取决于所有 g 是否单调/子模。

算法 4 正向模拟

输入: 反向采样 R，顺序优先级 $Rank$，初始点集 S_a, S_r，初始节点 u，界限范围 $LimitDistance$

输出: 每个点的 Dis 与 $Status$，代表接收时刻和接收信息类型

Q 为一单调队列，按照顺序优先级排序，初始元素为 $R \cap (S_a \cup S_r)$
初始化 $gain(v, R) = 0$
初始化 $Dis(v) = n$
$Status(v \in S_a) = C_a, Status(v \in S_r) = C_r$
$Dis(v \in S_a) = 0, Dis(v \in S_r) = 0$
while Q 非空 **do**
    $t = Q.front()$
    $Q.pop_front$
    if $t \notin R$ **then**
        Continue
    end if
    for 每个 t 的后继 v **do**
        单边采样 (u, v)
        if (u, v) 增强存活或拒绝 **then**
            Continue
        end if
        if v 未被本次 while 访问过 **then**
            $Status(v) = Status(t)$
            $Dis(v) = Dis(t) + 1$
            $Q.push\_back(v)$
        end if
    end for
end while

算法 5 计算增加量（GainComputation）

输入: 反向采样 R，顺序优先级 $Rank$，初始点集 S_a, S_r，初始节点 u，界限范围 $LimitDistance$

输出: 每个点 v 对应的 $gain(v, R)$

运行正向模拟
初始化 $Dis'(v) = 0$
$Dis'(u) = LimitDistance$
Q 为一单调队列，按照 Dis' 从大到小排序，初始元素为 u
while Q 非空 **do**
    $t = Q.front()$
    $Q.pop_front$
    if $t \notin R$ **then**
        Continue
    end if
    if $Dis(t) \leq Dis'(t)$ 且 $Status(t) = C_a$ **then** $gain(v, R) = gain(C_a) - gain(Status(u))$
    end if
    for 每个 t 的前驱 v **do**
        单边采样 (v, u)
        if (v, u) 拒绝 **then**
            Continue
        end if
        if v 未被本次 while 访问过 **then**
            $Dis'(v) = min(Dis'(t) - 1, Dis(v))$
            $Q.push\_back(v)$
        end if
    end for
end while
return $gain()$

算法 6 计算单点增加量

输入: 反向采样 R，顺序优先级 $Rank$，初始点集 S_a, S_r，初始节点 u，增强节点 v，界限范围 $LimitDistance$

输出: $gain(v, R)$

Q 为一队列，初始元素为 v
if $Status(v) = C_a$ **then**
    $Status(v) = C_a^+$
end if
if $Status(v) = C_r$ **then**
    $Status(v) = C_r^-$
end if
初始化 $gain(v, R) = 0$
$OldStatus = Status(u)$
while Q 非空 **do**
    $t = Q.front()$
    $Q.pop_front$
    if $t \notin R$ 或 t 已经被访问过 $|R|$ 次 **then**
        Continue
    end if
    if t 已经被访问过 $|R|$ 次 **then**
        $Status(t) = None$
    end if
    for 每个 t 的后继 w **do**
        单边采样 (u, w)
        if (u, w) 拒绝 **then**
            Continue
        end if
        重新计算 w 的接收时间和状态
        if w 状态发生改变 **then**
            $Q.push\_back(w)$
        end if
    end for
end while
return $gain(Status(u)) - gain(OldStatus)$

算法 7 计算增加量（GainComputation）

输入: 反向采样 R，顺序优先级 $Rank$，初始点集 S_a, S_r，初始节点 u，界限范围 $LimitDistance$

输出: $gain(v, R)$

1: **for** $v \in V$ **do**
2: $\quad gain' =$ 计算 v 增加量
3: $\quad gain+ = gain'$
4: **end for**
5: **return** $gain$

单调性

定理 1.3.4 当 $C_a \succ C_r \succ C_a^+$ 或 $C_a \succ C_r^- \succ C_a^+$ 或 $C_r^- \succ C_a^+ \succ C_r$ 或 $C_r^- \succ C_a \succ C_r$ 时，原函数非单调，否则原函数单调。

证明 1.3.4 显然，一个点变为增强节点会使得该点激活状态发生变化，若该点原本不会激活则原函数不变。我们考虑之后每个点的入边传入信息和时间发生的变化，共分为两种：

1. 某个边时间不变，传入信息优先级变高。此时唯一可能发生的事情是该节点不变，或从原本的信息变为该信息。这可能在增强节点使得 C_a 变为优先级更高的 C_a^+ 或者 C_r 变为优先级更高的 C_r^- 后的后续传播中发生。

2. 某个边时间不变，传入信息优先级变低。此时可能发生的事情是该节点不变，或变为该信息，或变为某个在新优先级和旧优先级之间的信息（在有这类信息同时传入的情况下）。我们称最后一种情况为让位。这可能在增强节点使得 C_a 变为优先级更低的 C_a^+ 或者 C_r 变为优先级更低的 C_r^- 时，后续的传播，或让位后后续传播中发生。

3. 某个边不再传入信息。此时该节点变为某个其他信息，或者不在会被激活。这可能在增强节点使得 C_r 变为优先级更高的 C_r^- 阻断 C_a^+ 在”增强存活”边上的传播时发生。

4. 某个边开始传入信息。此时该节点不变或变为该信息。这可能在增强节点使得 C_a 变为 C_a^+ 在”增强存活”边上的传播时发生。

按照这四种情况分别讨论：

1. 此时若传入信息成功激活该节点，则需要保证该信息对目标函数贡献不低于原本信息贡献。因此，$C_r^- \succ C_a^+ \succ C_r$ 和 $C_r^- \succ C_a \succ C_r$ 不能存在。

2. 此时若传入信息成功激活该节点或发生让位，则需要保证该信息对目标函数贡献不低于原本信息贡献。因此，$C_a \succ C_r \succ C_a^+$ 或 $C_a \succ C_r^- \succ C_a^+$ 不能存在。

3. 当 $C_r^- \succ C_a^+ \succ C_r$ 不存在时，阻断不会发生，因此不需要考虑这种情况。

4. 由于 C_a^+ 是贡献最高的状态之一，这种情况一定不会降低目标函数。

综上，当 $C_a \succ C_r \succ C_a^+$ 或 $C_a \succ C_r^- \succ C_a^+$ 或 $C_r^- \succ C_a^+ \succ C_r$ 或 $C_r^- \succ C_a \succ C_r$ 时，原函数非单调，否则原函数单调。

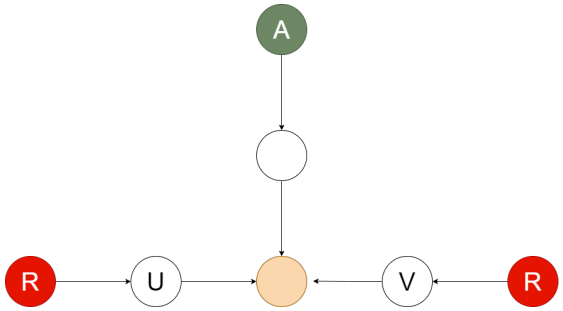

图 3.1: 单调性反例

如图 3.1 所示，A 是一个正面信息种子，R 是两个负面信息种子。当 $C_r^- \succ C_a \succ C_r$ 时，橙色节点和其后续节点初始为正面信息激活，但 U 或 V 变为增强节点后，橙色节点和其后续节点变为 C_r^-。由于橙色节点和其后续节点可以有任意多个，容易构造情况使得原图不符合单调性。同理，$C_r^- \succ C_a^+ \succ C_r$ 只需要一开始在空白节点上有增强节点即可，$C_a \succ C_r \succ C_a^+$ 或 $C_a \succ C_r^- \succ C_a^+$ 只需反转成 R 为正面信息种子，A 为负面信息种子，当 U 和 V 均变为增强节点后，橙色节点和其后续节点不满足单调性。由表 3.1 得知，这一反例覆盖了所有非单调情况。

子模性

定理 1.3.5 当优先级为下表中所示三者之一时，原函数子模。

证明 1.3.5 观察可以发现这三种优先级都符合：

1.$C_r^- \succ C_r$ 且中间没有其他信息等级。因此，当激活节点使得 C_r 变为 C_r^- 时，后续影响刚好是能够通过“存活”边传播到的，时间符合条件的所有 C_r 节点。

2.$C_a^+ \succ C_a$。因此，当激活节点使得 C_a 变为 C_a^+ 时，后续影响刚好是能够通过“存活”和“增强存活”边传播到的，时间符合条件的所有节点。

综上，我们可以得到每个 C_r 变为 C_r^- 能够影响到没有被其他 C_r^- 影响到的节点，每个 C_a 变为 C_a^+ 能够影响到没有被其他 C_a^+ 影响到的节点。这分别于 *Kempe* 等[8] 证明的普通独立级联模型下目标函数子模性一致，沿用其证明得证。

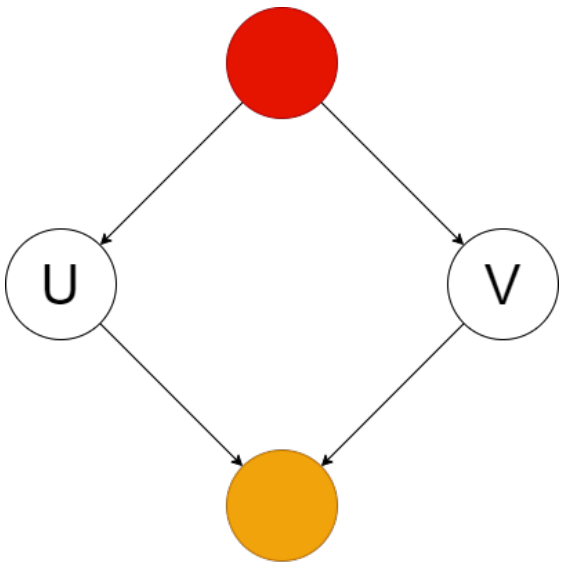

图 3.2: 子模性反例

如图 3.2 所示，红色节点为负面信息种子。当 $C_r \succ C_r^-$ 时，有 U 或 V 激活都不能改变橙色节点状态，但 U 和 V 同时激活可以。由于橙色节点和其后续节点可以有任意多个，容易构造情况使得原图不符合子模性。同理，当 $C_a \succ C_a^+$ 时，假设橙色节点后有任意多的“增强存活”边，则容易构造情况使得原图不符合子模性。另外，在图 3.1 中，若 $C_r^- \succ C_a \succ C_r$ 或 $C_r^- \succ C_a^+ \succ C_r$，则 V 变为增强节点会减少橙色节点贡献，但先增强 U 后增强 V 时 V 不会有贡献影响，不符合子模性。由表 3.1 得知，这些反例覆盖了所有非子模情况。

表 3.1: 不同优先级下的单调子模性

单调子模（M-S）	非单调子模（nM-S）
$C_a^+ \succ C_a \succ C_r^- \succ C_r$	
$C_a^+ \succ C_r^- \succ C_r \succ C_a$	
$C_r^- \succ C_r \succ C_a^+ \succ C_a$	
单调非子模（M-nS）	非单调非子模（nM-nS）
$C_a^+ \succ C_a \succ C_r \succ C_r^-$	$C_a^+ \succ C_r^- \succ C_a \succ C_r$
$C_a \succ C_a^+ \succ C_r^- \succ C_r$	$C_r^- \succ C_a^+ \succ C_a \succ C_r$
$C_a \succ C_a^+ \succ C_r \succ C_r^-$	$C_r^- \succ C_a \succ C_a^+ \succ C_r$
$C_a^+ \succ C_r \succ C_r^- \succ C_a$	$C_r^- \succ C_a \succ C_r \succ C_a^+$
$C_r \succ C_r^- \succ C_a^+ \succ C_a$	$C_a \succ C_r^- \succ C_a^+ \succ C_r$
$C_r^- \succ C_r \succ C_a \succ C_a^+$	$C_a \succ C_r^- \succ C_r \succ C_a^+$
$C_r \succ C_r^- \succ C_a \succ C_a^+$	$C_a \succ C_r \succ C_r^- \succ C_a^+$
$C_a^+ \succ C_r \succ C_a \succ C_r^-$	$C_a \succ C_r \succ C_a^+ \succ C_r^-$
$C_r \succ C_a^+ \succ C_a \succ C_r^-$	$C_r \succ C_a \succ C_r^- \succ C_a^+$
$C_r \succ C_a \succ C_a^+ \succ C_r^-$	$C_r^- \succ C_a^+ \succ C_r \succ C_a$
$C_r \succ C_a^+ \succ C_r^- \succ C_a$	

3.2.3 M-S 情况下的算法

假设已经得到了多次反向采样 $\mathcal{R} = \{R_i\}, i \in [\theta]$，那么可以每次贪心地选取增强节点，也就是选 u 使得 $\Phi(S) = \sum_{R \in \mathcal{R}} gain(A \cup \{u\}, R)$ 最大。注意到 Φ 的单调子模性与 h 一致，在本章中假设 h 具有单调子模性。

定理 1.3.6 当 h 具有单调子模性时，上述贪心算法会返回一个近似比 $1-1/e$ 的解。

证明 1.3.6 *Kempe* 等[8] 给出了该定理的详细证明，此处不再阐述。

随后通过运用在 IMM 算法[5] 中的采样方法，本文提出一个以 $1 - \frac{1}{n^\ell}$ 概率返回 $1 - 1/e - \varepsilon$ 近似比的算法。

总的来说，整个 PR-IMM 算法分为得到足够数量的反向采样样本（Sample-Generation）和利用样本进行贪心（GreedySelection）两个部分。假设目标函数 h

的最优值为 OPT，在采样部分的参数为

$$\alpha = \delta\sqrt{\ell \log n + \log 2}$$

$$\beta = \sqrt{\delta\left(\ell \log n + \log\binom{n}{k} + \log 2\right)}$$

$$\theta_i = (1 + \frac{\sqrt{2}}{3}\varepsilon) \cdot (\log\binom{n}{k} + \ell \log n + \log\log_2 n) \cdot \frac{2^i}{\varepsilon^2}$$

同时利用 IMM 算法[5] 中的分析，我们有如下的结果。

引理 1.3.1 假设贪心算法对 $\Phi(\cdot)$ 函数返回一个近似比为 δ 的解，则有 $1-n^{-\ell}$ 的概率，采样代码返回的 $\mathcal{I}$ 有

$$|\mathcal{R}| \geq 2n_v(\alpha+\beta)^2\varepsilon^{-2}/OPT \quad \textit{(Theorem.2 in[5])}.$$

同时，在上述等式成立时，有贪心算法以 $1-n^{-\ell}$ 的概率返回一个 S，符合

$$\lambda f(S) + (1-\lambda)g(S) \geq (\delta - \varepsilon) \cdot OPT \quad \textit{(Theorem.1 in[5])}.$$

算法 8 PR-IMM 算法

输入: 带边权图 $G(V,E)$，顺序优先级 $Rank$，初始点集 S_a, S_r，参数 ℓ, ε

输出: 一个原问题的解 S

1: 初始化 $\mathcal{R} = \emptyset$

2: $\ell' = \ell \cdot (1 + \log 2/\log n)$

3: $\langle \mathcal{R}, \{\Psi(u)\}_{u\in V}\rangle$ =SampleGeneration$(G, k, \ell', \varepsilon)$

4: S =GreedySelection$(S_a, S_r, \mathcal{R}, k, \{\Psi(u)\}_{u\in V})$

5: **return** S

利用上述两个定理，我们得到上述算法有近似比 $1-1/e-\varepsilon$。

接下来我们计算时间复杂度。称生成一个反向采样的复杂度为 EPT。Tang 等[5] 证明了采样复杂度为

$$O(\mathbb{E}[|\mathcal{R}|] \cdot EPT) = O\left(\frac{EPT}{OPT}(k+\ell)(n+m)\varepsilon^{-2}\log n\right).$$

同时,这里的贪心可以用最大覆盖问题贪心相同,得到时间复杂度为 $O\left(\sum_{R\in\mathcal{R}}|R|\right)$[21],和输入线性相关。综上所述，上述算法近似比为 $(1-1/e-\varepsilon)$，概率为 $1-2n^{-\ell'} = 1-n^{-\ell}$。我们得到如下定理。

定理 1.3.7 *PR-IMM* 算法在 $O\left(\frac{EPT}{OPT}(k+\ell)(n+m)\varepsilon^{-2}\log n\right)$ 时间复杂度内以 $1-1/n^{\ell}$ 概率返回一个 $(1-1/e-\varepsilon)$ 近似比的解。

算法 9 计算反向采样（SampleGeneration）

输入: 带边权图 $G(V,E)$，顺序优先级 $Rank$，初始点集 S_a, S_r，参数 ℓ, ε

输出: 一组采样 $\mathcal{R}$ 和采样中每个点能够达到的贡献 $\{\Psi(u)\}_{u\in V}$

初始化 $\mathcal{R} = \emptyset, LB = 1$
初始化 $\Psi(u) = 0$ for each $u \in V$
for $i = 1$ to $\log_2 n - 1$ **do**
    for $j = |\mathcal{R}|$ to θ_i **do**
        R =PR-sketch-Generation(G, S_a, S_r, v)
        $\{\psi(u,R)\}_{u\in V}$ =GainComputation(S_a, S_r, R)
        $\forall u \in V, \Psi(u) = \Psi(u) + \psi(u,R)$
        $\mathcal{R}.push\_back(R)$
    end for
    $S =$ GreedySelection$(S_a, S_r, \mathcal{R}, k, \{\Psi(u)\}_{u\in V})$
    if $n \cdot \Phi(S,\mathcal{R}) \geq (1+\sqrt{2}\cdot\varepsilon)\cdot n/2^i$ **then**
        $LB = n \cdot \Phi(S,\mathcal{R})/(1+\sqrt{2}\cdot\varepsilon)$
        break
    end if
end for
$\theta = \frac{2n(\alpha+\beta)^2}{LB\cdot\varepsilon^2}$
for $j = |\mathcal{R}|$ to θ **do**
    R =PR-sketch-Generation(G, S_a, S_r, v)
    $\{\psi(u,R)\}_{u\in V}$ =GainComputation(S_a, S_r, R)
    $\forall u \in V, \Psi(u) = \Psi(u) + \psi(u,R)$
    $\mathcal{R}.push\_back(R)$
end for
return $\langle \mathcal{R}, \{\Psi(u)\}_{u\in V} \rangle$

算法 10 贪心求解（GreedySelection）

输入: 初始点集 S_a, S_r，采样集合 $\mathcal{R}$，每个点的贡献 $\{\Psi(u)\}_{u\in V}$

输出: 一个原问题的解 S

Initial $S = \emptyset$
for $i = 1$ to k **do**
    $v = \arg\max_{v\in V\setminus S} \Psi(v)$
    $S = S \cup \{v\}$
    for 每个 $R \in \mathcal{R}$ 使得 $v \in R$ **do**
        如果非单调或非子模，重新计算每个点的贡献
        $\forall u \in V, \Psi(u) = \Psi(u) - \psi(u, R) + \phi(u, R)$
    end for
end forreturn S

3.2.4 M-nS 情况下的算法

当优先级符合上文中的要求，使得目标函数非子模时时，不能使用前文的算法。本文这里给出一个依赖数据的近似算法。

三明治近似算法

如果目标函数不是子模的，我们用三明治近似算法（Sandwich Approximation，SA)[22] 得到一个依赖数据的近似算法。具体来说我们尝试找到一个目标函数 $h(S)$ 的上界 $U(S)$ 和下界 $L(S)$，并保证这两个上下界是子模的。然后我们分别用普通贪心算法得到原问题的三个解 S_h，S_U 和 S_L。假设 S_U 和 S_L 对最大化 $U(S)$ 和 $L(S)$ 有 $1-1/e-\varepsilon$ 的近似比。设 $S_{sa} = \arg\max_{S\in\{S_h,S_U,S_L\}} h(S)$，利用三明治近似的证明[22] 我们有

$$h(S_{sa}) \geq \max\{\frac{L(S^*)}{h(S^*)}, \frac{h(S_U)}{U(S_U)}\} \cdot (1 - 1/e - \varepsilon) \cdot OPT,$$

注意到这一算法的近似比取决于 $U(S)$ 和 $L(S)$ 与 $h(S)$ 比多相近。因此接下来的目标是找到和 $h(S)$ 尽可能相近的 $U(S)$ 和 $L(S)$，同时还要保证子模性。

我们使用一个简单的上界 $U(S)$：将优先级改为 $C_a^+ \succ C_a \succ C_r^- \succ C_r$ 即可，其为所有其他优先级的一个上界，且具有子模性。

接下来我们尝试得到一个合理的下界。设 μ_{sv} 代表当 s 被选中时，v 对目标

函数的期望增加量。根据信息传播的独立性[8, 23]，我们有 $h(\{s\}) = \sum_v \mu_{sv}$.

这样就得到了一个下界函数如下：$\underline{h}(S) = \sum_{v\in V} \max_{s\in S}\{\mu_{sv}\}$

引理 1.3.2 函数 $\underline{h}(S)$ 是 $h(S)$ 的一个下界函数且单调子模。

证明 1.3.7 设 $h(S, v)$ 为增强节点为 S 时 v 对目标函数的贡献。显然，对于 $s \in S$ 有 $h(S, v) \geq \mu_{sv}$。因此有 $h(S) = \sum_{v\in V} h(S, v) \geq \sum_{v\in V} \max_{s\in S} \mu_{sv} = \underline{h}(S)$。由此证明 $\underline{h}(S)$ 是 $h(S)$ 的下界函数。

随后我们可以利用 $\underline{h}(S)$ 是每个节点贡献的简单加和。对于点 v，假设其目前贡献来自于增强节点 $s_1 \in S$。当 s_2 加入 S 时，要么产生更大的贡献（即 $\mu_{s_2v} > \mu_{s_1v}$），或保持原贡献不变（即 $\mu_{s_2v} \leq \mu_{s_1v}$），由此可得单调性。在上一种情况中，$s_2$ 的贡献是 $\mu_{s_2v} - \mu_{s_1v} \leq \mu_{sv}$。由于这对任意 s_1, s_2 和 S 均成立，我们有 $\underline{h}(S \cup \{s_2\}) - \underline{h}(S) \geq \underline{h}(S \cup \{s_1, s_2\}) - \underline{h}(S \cup \{s_1\})$，由此可得子模性。

SA-IMM 算法

算法 11 中，第 1-3 行与 PR-IMM 算法相同。这里用函数 GreedySelectionUB 作为算法的贪心部分，以示区分。在第 4 行，我们用第 2 行得到的反向采样给出一个得到解 S_h 的方法。第 5-12 行给出一个得到下界解 S_L 的方法：

对于给定点 v，我们尝试做反向采样 $\mathcal{R}_v$.

我们不知道哪个 s 得到最大贡献，因此我们必须修改采样过程，以合理的误差范围得到合理的 s。

对于任意 s，我们有 $\mu_{sv} = \mathbb{E}[gain(s, R)]$

设 $F_v(s, \mathcal{R}_v) = \frac{\sum_{R\in\mathcal{R}_v} gain(s,R)}{|\mathcal{R}_v|}$

使用和上文相同的鞅论方法[5]，我们可以得到这里的采样同样符合原文中的结论。对于一段随机得到的反向采样 $\mathcal{R}_v$，其中 $\theta = |\mathcal{R}_v|$，对任意 $v \in S$ 和 $\mu_{sv} = \mathbb{E}[gain(s, R)]$，我们有

$$\Pr[\sum_{R\in\mathcal{R}_v} gain(s, R) - \theta\mu_{sv} \geq \delta\theta\mu_{sv}] \leq \exp(-\frac{\delta^2}{2 + \frac{2}{3}\delta}\theta\mu_{sv}),$$

对任意 $\delta > 0$ 成立，且

$$\Pr[\sum_{R\in\mathcal{R}_v} gain(s, R) - \theta\mu_{sv} \leq -\delta\theta\mu_{sv}] \leq \exp(-\frac{\delta^2}{2}\theta\mu_{sv}),$$

对任意 $0 < \delta < 1$ 成立。

接下来我们给出对任意 s 和 v 估计 μ_{sv} 时，使用反向采样集合 $\mathcal{R}_v$ 的误差范围。

算法 11 SA-IMM 算法

输入: 带边权图 $G(V, E)$，顺序优先级 $Rank$，初始点集 S_a, S_r，参数 $\ell, \varepsilon_1, \varepsilon_2$

输出: 三个原问题的解 $\langle S_U, S_h, S_L \rangle$ 初始化 $\mathcal{R} = \emptyset$

$\ell' = \ell \cdot (1 + \log 2 / \log n)$

$\langle \mathcal{R}, \{\Psi(u)\}_{u \in V} \rangle$ =SampleGeneration$(G, k, \ell', \varepsilon_1)$

S_U =GreedySelectionUB$(S_a, S_r, \mathcal{R}, k, \{\Psi(u)\}_{u \in V})$

S_h =GreedySelection$(S_a, S_r, \mathcal{R}, k, \{\Psi(u)\}_{u \in V})$

$\kappa = \frac{\varepsilon_2}{2 - \varepsilon_2}$

$\theta = (2 + \frac{2}{3}\kappa)(2 - \frac{1}{e} + \kappa)\frac{(\ell+1)\log n + \log 2}{\kappa^3(3 - \frac{1}{e})}$

对每个 v 初始化 $\mathcal{R}_v = \emptyset$

for each $v \in V$ **do**

  for $j = 0$ to θ **do**

    R =PR-sketch-Generation(G, S_a, S_r, v)

    $\mathcal{R}_v.push\_back(R)$

  end for

end for

S_L =GreedySelectionLB$(S_a, S_r, \{\mathcal{R}_v\}_{v \in V}, k)$

return $\langle S_U, S_h, S_L \rangle$

引理 1.3.3 若 $\mathcal{R}_v$ 符合

$$\theta = |\mathcal{R}_v| \geq (2 + \frac{2}{3}\kappa)\frac{(\ell + 1)\log n + \log 2}{\gamma \cdot \kappa^2},$$

则对有 $1 - \frac{1}{n^\ell}$ 的概率，对所有 $s \in V$ 都有

$$\textit{if} \quad \mu_{sv} \geq \gamma, |\mu_{sv} - F_v(s, \mathcal{R}_v)| < \kappa\mu_{sv},$$

$$\textit{if} \quad \mu_{sv} < \gamma, |\mu_{sv} - F_v(s, \mathcal{R}_v)| < \kappa\gamma.$$

证明 1.3.8 固定点 s。已知 $\mu_{sv} = \mathbb{E}[gain(s,R)]$，则 $|\mathcal{R}_v| \cdot F_v(\mathcal{R}_v)$ 可以被视为 $|\mathcal{R}_v|$ 次独立同伯努利分布 *(i.i.d. Bernoulli)* 变量，均值 μ_{sv}。若 $\mu_{sv} < \gamma$，我们有

$$\begin{aligned}
&\Pr[|F_v(s,\mathcal{R}_v) - \mu_{sv}| \geq \kappa\gamma] \\
&\quad = \Pr[\sum_{R\in\mathcal{R}_v} gain(s,R) - \theta\mu_{sv} \geq \theta\frac{\kappa\gamma}{\mu_{sv}}\mu_{sv}] \\
&\quad \leq 2\exp(-\frac{\frac{\kappa^2\gamma^2}{\mu_{sv}}}{2+\frac{2\kappa\gamma}{3\mu_{sv}}}|\mathcal{R}_v|) \\
&\quad = 2\exp(-\frac{3\kappa^2\gamma^2}{6\mu_{sv}+2\kappa\gamma}\theta) \\
&\quad < 2\exp(-\frac{3\kappa^2\gamma^2}{6\gamma+2\kappa\gamma}\theta) = \frac{1}{n^{\ell+1}}.
\end{aligned}$$

同时，若 $\mu_{sv} \geq \gamma$，我们有

$$\begin{aligned}
&\Pr[|F_v(s,\mathcal{R}_v) - \mu_{sv}| \geq \kappa\mu_{sv}] \\
&\quad = \Pr[\sum_{R\in\mathcal{R}_v} gain(s,R) - \theta\mu_{sv} \geq \theta\kappa\mu_{sv}] \\
&\quad \leq 2\exp(-\frac{\kappa^2\mu_{sv}}{2+\frac{2}{3}\kappa}\theta) \leq \frac{1}{n^{\ell+1}}.
\end{aligned}$$

两者结合，得证。

由上述定理可得

$$\begin{aligned}
&\underline{h}(S) = \sum_{v\in V}\max_{s\in S}\{\mu_{sv}\} \\
&= \sum_{v\in V_\gamma} h(s,v) + \sum_{v\in V\setminus V_\gamma}\mu_{sv} \\
&\geq \sum_{v\in V_\gamma}\frac{F_v(S,\mathcal{R}_v)}{1+\kappa} + \sum_{v\in V\setminus V_\gamma}(F_v(S,\mathcal{R}_v) - \kappa\gamma) \\
&\geq \sum_{v\in V}\frac{F(S,\mathcal{R}_v)}{1+\kappa} - n\kappa\gamma \\
&\geq \frac{1-1/e}{1+\kappa}\sum_{v\in V}F(S^*,\mathcal{R}_v) - n\kappa\gamma \\
&\geq \frac{1-1/e}{1+\kappa}(\sum_{v\in V_\gamma}(1-\kappa)h(S^*) + \sum_{v\in V\setminus V_\gamma}(h(S^*)-\kappa\gamma)) - n\kappa\gamma \\
&= (1-\frac{1}{e})\frac{1-\kappa}{1+\kappa}OPT - (\frac{1-1/e}{1+\kappa}+1)n\kappa\gamma
\end{aligned}$$

由于近似比保证和采样次数都取决于 κ 和 γ 两个变量，设 $\gamma = \frac{\gamma'}{n}OPT$，并设 κ 为一变量以符合

$$(1+\kappa)\varepsilon_2 = 2\kappa + (2 - \frac{1}{e} + \kappa)\gamma'\kappa.$$

我们可以得到一个式子简单的近似比 $\underline{h}(S) \geq (1 - \frac{1}{e} - \varepsilon_2)OPT$.

利用与 IMM 算法[5] 类似的下界估算方法，我们可以得到一个 OPT 的下界，以获得一个 γ 的估计来保证上述定理要求成立。然而，这会使每个 v 的采样次数乘上一个 n，则总采样会带有系数 n^2。这么做采样次数显然过大。

因此，考虑更加简单的情况：设 $\kappa = \frac{\varepsilon_2}{2(1-1/e))-\varepsilon_2}$。由于 $\kappa \in (0,1]$，我们有 $\varepsilon \in (0, 1 - \frac{1}{e}]$。随后有

$$\underline{h}(S) \geq (1 - \frac{1}{e} - \varepsilon_2)OPT - (\frac{1-1/e}{1+\kappa} + 1)n\kappa\gamma.$$

设 $\gamma = \frac{\kappa(3-1/e)}{(2-1/e+\kappa)}$，由于 $\kappa \in (0,1]$，我们有 $\gamma \in (0,1]$，因此

$$\underline{h}(S) \geq (1 - \frac{1}{e} - \varepsilon_2)OPT - \frac{(3-1/e)\kappa^2}{1+\kappa}n.$$

得到采样次数为

$$\theta = |\mathcal{R}_v| = (2 + \frac{2}{3}\kappa)(2 - \frac{1}{e} + \kappa)\frac{(\ell+1)\log n + \log 2}{\kappa^3(3-1/e)},$$

将 κ 替换为 $\frac{\varepsilon_2}{2-\varepsilon_2}$，得到总时间复杂度为 $O(EPTn\ell\varepsilon_2^{-3}\log n)$。

这里，我们假设 $\kappa\gamma = \frac{\kappa^2(3-1/e)}{(2-1/e+\kappa)}$。当 $\kappa = 1$ 时，$\gamma = 1$。上述定理有 $h(s,v) \leq \gamma = 1$.

当 $\kappa = 0$，有 $\gamma = 0$，$\kappa\gamma = 0$。此时 $h(s,v) \geq \gamma = 0$，我们需要采样无数次才能保证 $h(s,v)$ 绝对精确.

当 ε 很小时，相对误差量 $1 - 1/e - \varepsilon_2$ 和绝对误差量 $\frac{\varepsilon_2^2(3-1/e)}{(2-\varepsilon_2)^2}$ 都会很小。我们在实验中调整 ε_2 的值，观察效果变化。

定理 1.3.8 对任意 $\varepsilon_1 > 0$，$\varepsilon_2 \in (0, 1 - \frac{1}{e}]$ 和 $\kappa = \frac{\varepsilon_2}{2(1-1/e))-\varepsilon_2}$*SA-IMM* 算法有至少 $1 - 2/n^\ell$ 的概率返回一个 S_{sa}，且具有下述的两个算法保证：

$$h(S_{sa}) \geq \frac{h(S_U)}{U(S_U)} \cdot (1 - 1/e - \varepsilon_1) \cdot OPT,$$

$$h(S_{sa}) \geq \frac{L(S^*)}{h(S^*)} \cdot (1 - 1/e - \varepsilon_2) \cdot OPT - \frac{(3-1/e)\kappa^2}{1+\kappa}n.$$

时间复杂度是 $O(\max\{T_1, T_2\})$，其中 $T_1 = \frac{EPT}{OPT}(k+\ell)(n+m)\varepsilon_1^{-2}\log n$，$T_2 = EPTn\ell\varepsilon_2^{-3}\log n)$.

3.2.5 nM-S 情况下的算法

首先我们考虑 nM-S 的情况。我们使用 Buchbinder 等提出的随机贪心（RandomGreedy）算法[24] 来代替原算法中的贪心部分。（如下算法 13）我们可以得到一个近似比 $(\frac{1}{e}-\varepsilon)$ 的解。对应的 RG-IMM 算法如下代码 12。

算法 12 RG-IMM 算法

输入： 带边权图 $G(V,E)$，顺序优先级 $Rank$，初始点集 S_a, S_r，参数 ℓ, ε

输出： 一个原问题的解 S

1: 初始化 $\mathcal{R}=\emptyset$ and let $\ell'=\ell\cdot(1+\log 2/\log n)$

2: $\langle \mathcal{R}, \{\Psi(u)\}_{u\in V}\rangle$ =SampleGeneration(G,k,ℓ',ε)

3: S =RandomGreedy$(S_a, S_r, \mathcal{R}, k, \{\Psi(u)\}_{u\in V})$

4: **return** S

Buchbinder 等[24] 证明了随机贪心（RandomGreedy）算法能够在非单调但子模的函数上保证 $\frac{1}{e}$ 的近似比。类似于 PR-IMM 算法，当 δ 设为 $\frac{1}{e}$ 时，我们可以得到 RG-IMM 算法的算法保证如下。

定理 1.3.9 *RG-IMM* 算法有 $1-1/n^{\ell}$ 的概率返回一个 $(1/e-\varepsilon)$ 近似比的解，时间复杂度 $O\left(\frac{EPT}{OPT}(k+\ell)(n+m)\varepsilon^{-2}\log n\right)$。

3.2.6 nM-nS 情况下的算法

类似地，我们可以使用 SA-IMM 算法，随后将所有贪心算法部分（包括 GreedySelectionUB 和 GreedySelectionLB）改为 RandomGreedy 算法，称为 SA-RG-IMM 算法。同样地，我们设 $\gamma=\frac{\kappa(2+1/e)}{1+1/e+\kappa}$，$\kappa=\frac{\varepsilon_2}{2/e-\varepsilon_2}$。

定理 1.3.10 对于任意 $\varepsilon_1>0, \varepsilon_2\in(0,\frac{1}{e}]$ 和 $\kappa=\frac{\varepsilon_2}{2/e-\varepsilon_2}$，*SA-RG-IMM* 算法有至少 $1-2/n^{\ell}$ 的概率返回一个解 S_{sa} 保证如下两个限制：

$$h(S_{sa})\geq\frac{h(S_U)}{U(S_U)}\cdot(1/e-\varepsilon_1)\cdot OPT,$$
$$h(S_{sa})\geq\frac{L(S^*)}{h(S^*)}\cdot(1/e-\varepsilon_2)\cdot OPT-\frac{(2+1/e)\kappa^2}{1+\kappa}n.$$

时间复杂度为 $O(\max\{T_1,T_2\})$，其中 $T_1=\frac{EPT}{OPT}(k+\ell)(n+m)\varepsilon_1^{-2}\log n$，$T_2=EPTn\ell\varepsilon_2^{-3}\log n)$。

算法 13 随机贪心（RandomGreedy）

输入: 初始点集 S_a, S_r，采样集合 $\mathcal{R}$，每个点的贡献 $\{\Psi(u)\}_{u\in V}$

输出: 一个原问题的解 S

初始化 $S=\emptyset$ and $S_0=\emptyset$
for $i=1$ to k **do**
    $\forall u\in V, \Psi'(u)=\Psi(u)$
    for 每个 $R\in\mathcal{R}$ 使得和 S_{i-1} 有交 **do**
        $\{\phi(u,R)\}_{u\in V}$ = GainUpdate(S_a, S_r, S_{i-1}, R)
        $\forall u\in V, \Psi'(u)=\Psi'(u)-\psi(u,R)+\phi(u,R)$
    end for
    初始化 $S_i=\emptyset$
    for $j=1$ to k **do**
        $v=\arg\max_{v\in V\setminus\{S_i\cup S_{i-1}\}}\Psi'(v)$
        $S_i=S_i\cup\{v\}$
    end for
    从 S_i 中随机等概率选择一个 u
    $S=S\cup\{u\}$
end for
return S

4 实验结果

4.1 实验数据

我们利用真实的网络数据来测试我们算法的效果。主要的数据来自于真实的社交网络和合作网络，包括来自 Stanford 的公开 facebook 等社交网络数据，和来自 arXiv.org 上的合作网络等。

对于本身是无向边的数据，我们拆成两条有向边来使用。本文当中不同算法的运行速度差别很大，因此很多数据来源于总社交网络上的部分子图。对于正面和负面信息的初始种子节点，我们使用 Tong 等提出的方法[25]，根据图的不同大小，选取度数在前 $[1/2, 1/4]$ 的节点中随机。对于边上的两个权值，我们首先使用 Tong 等[25] 使用的度数倒数方法 $p_{uv} = 1/d_u$，然后使用 Lin 等[19] 提出的方法，令 C_a^+ 对应的增强概率为 $p_{uv}^+ = 1 - (1 - p_{uv})^\beta$，其中 β 称为增强参数，本文选取 $\beta = 2$。

表 4.1: 数据集信息

数据集	数据集信息
ego-Facebook[6]	来自 Facebook 上朋友圈信息组成的数据。 这一数据来自于在 Facebook 应用上的一个自愿调查。
ego-Twitter[6]	来自 twitter 上朋友圈信息组成的数据。 这一数据来自 twitter 上爬虫得到的公开数据。
ego-Gplus[6]	来自 Google+ 上朋友圈信息组成的数据。 这一数据由采集用户手动分享朋友圈的信息组合而成。
Musae-git[26]	来自 Github 上合作关系的网络。 节点代表至少有 10 个项目的用户，边代表用户关注。
NetPHY[20]	从 arXiv.org 上爬虫得到的论文合作网络。 Phy 数据集是整个数据的物理方向部分。
DBLP[20]	从 dblp.org 上爬虫得到的论文合作网络。

4.2 实验方案

除了文中给出的四种算法之外，我们给出了一些基本的对照算法如下：

表 4.2: 数据信息

数据	来自数据集	点数	边数
facebook-698	ego-Facebook	61	1080
twitter-87771546	ego-Twitter	78	2246
twitter-90072587	ego-Twitter	129	3260
facebook-686	ego-Facebook	168	6624
facebook-0	ego-Facebook	333	10076
gplus-115625564993990145546	ego-Gplus	923	78800
facebook-combined	ego-Facebook	4039	176468
phy	NetPHY	37149	463014
musae-git	Musae-git	37700	578006
dblp	DBLP	613586	3980518

表 4.3: 算法

算法代号	算法描述
PR-IMM	对应 M-S 的优先级情况的算法
SA-IMM(ALL)	对应 M-nS 的优先级情况的算法，上界函数、原函数、下界函数三个结果
RG-IMM	对应 nM-S 的优先级情况的算法（不使用）
SA-RG-IMM(ALL)	对应 nM-nS 的优先级情况的算法，上界函数、原函数、下界函数三个结果
SA-IMM(U&L)	对应 M-nS 的优先级情况的算法，只跑上界函数和下界函数两个结果
SA-RG-IMM(U&L)	对应 nM-nS 的优先级情况的算法，只跑上界函数和下界函数两个结果
Greedy	用蒙特卡洛法进行期望的估算，并在此之上使用文中描述的贪心算法
Random	随机选取 k 个点的算法
None	不选取任何点的算法

我们选择数据、种子数量、k、λ、ϵ_2 五个可变参数作为自变量，分别进行实验。

4.3 实验结果

若没有额外说明，实验中保证 $\epsilon = \epsilon_2, \ell = 1$。

4.3.1 图大小-运行时间

在数据（图大小）作为自变量的实验中，种子数量默认为正负各有 10 个，$k = 25$，$\lambda = 0.5$，对 PR-IMM 算法和 SA-IMM 算法 $\epsilon = \epsilon_2 = 0.5$，对 SA-RG-IMM 算法 $\epsilon = \epsilon_2 = 0.2$。

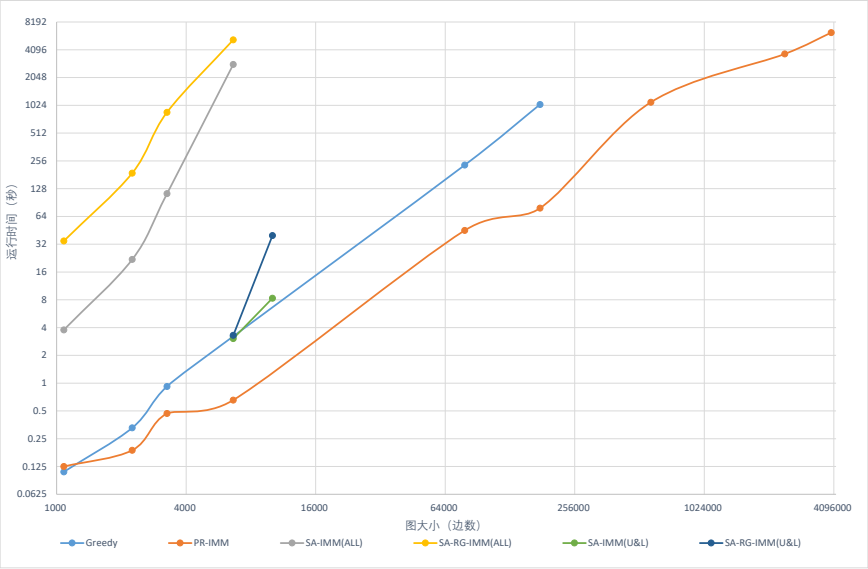

图 4.1: 图大小-运行时间对照实验

图 4.1 可以看到，可以看到作为最还原 IMM 算法原有过程的 PR-IMM 算法是最快的，其速度远远超过了蒙特卡洛贪心算法；SA-IMM 算法和 SA-RG-IMM 算法除去普通贪心后的速度和贪心算法相近，由于上界算法本身就是 PR-IMM 算法，大部分的时间花在了下界部分的采样上——这一采样比普通的采样多得

多。如果加上了普通贪心，SA-IMM 算法和 SA-RG-IMM 算法的速度甚至会比蒙特卡洛贪心算法更慢，主要是因为非单调或非子模的情况下，整个采样会使用上文提到的启发式算法，其复杂度比单调子模下的近线性算法慢得多。

4.3.2 图大小-运行效果

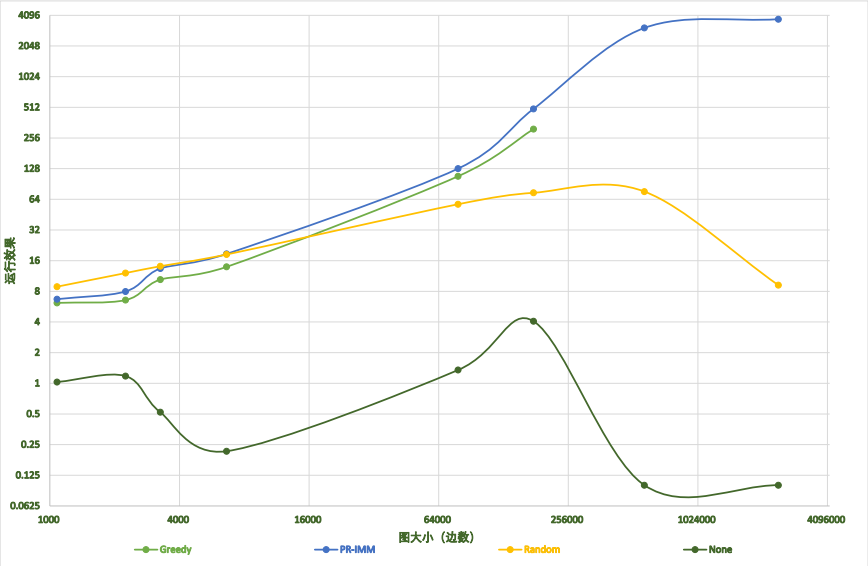

图 4.2: 图大小-运行效果 PR-IMM 算法对照实验

图 4.2 到图 4.4 可以看到，在 PR-IMM 算法方面，可以看到数据较大时可以和蒙特卡洛贪心算法和随机拉开一定差距。蒙特卡洛贪心在超过 64000 条边的时候不论是速度还是效果都跟不上 PR-IMM 算法，说明了 PR-IMM 的优势。SA-IMM 算法和 SA-RG-IMM 算法的三个返回结果效果互相比较相近，和蒙特卡洛贪心相比有一定优势。在较大数据上能看到下界函数返回的答案有一定劣势，说明 PR-IMM 算法在非单调子模的模型上也有不错的效果。

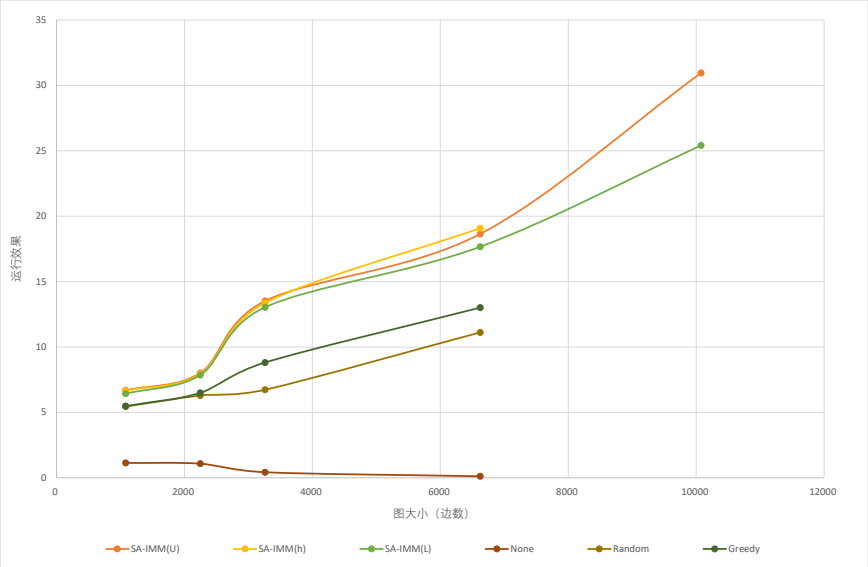

图 4.3: 图大小-运行效果 SA-IMM 算法对照实验

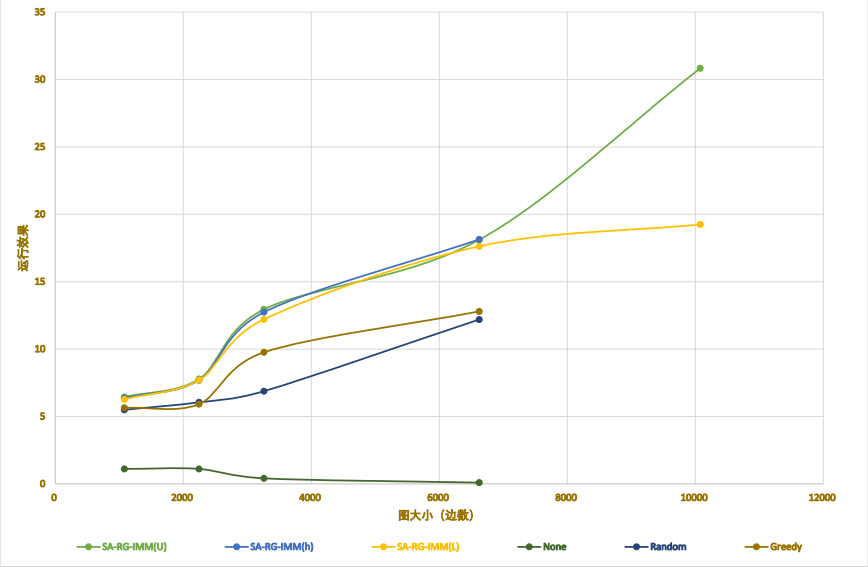

图 4.4: 图大小-运行效果 SA-RG-IMM 算法对照实验

4.3.3 种子数量-运行时间

在种子数量作为自变量的实验中，PR-IMM 算法的测试图为 463014 条边的 phy 数据，SA-IMM 算法和 SA-RG-IMM 算法的测试图为 10076 条边的 facebook-0 数据。其他非自变量参数和上一实验相同。

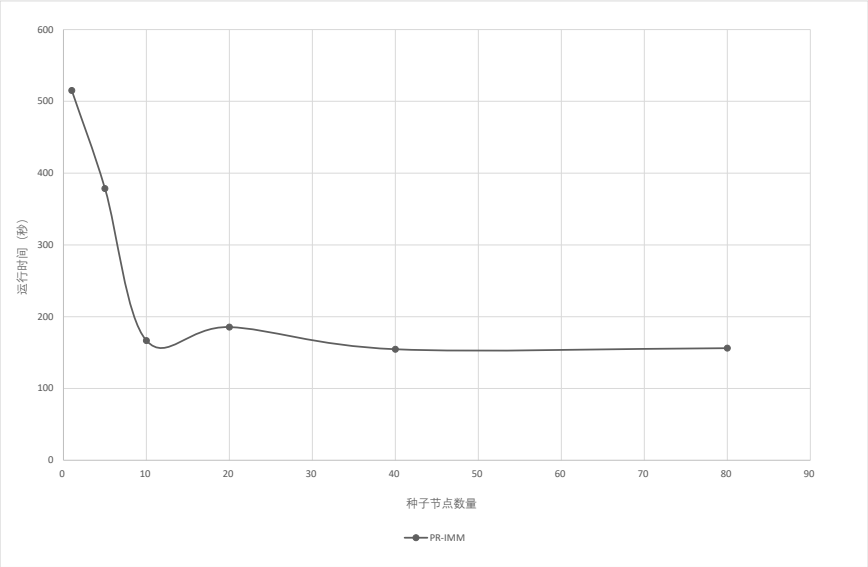

图 4.5: 种子数量-运行时间 PR-IMM 算法对照实验

图 4.5 到图 4.7 可以看到，由于 PR-IMM 算法采样计算贡献时是接近线性的，可以看到种子数量越少算法跑的越慢：这是因为一次反向采样的大小取决于最近的种子节点，因此种子数量少时采样会很大。SA-IMM 算法在种子数量非常低时也会很快，本文估计是启发式算法的效果，因为种子数量过低时实际能改变节点状态的增强节点位置非常少，大部分会被启发式很快判断无用。这里可以看出 SA-RG-IMM 算法明显比 SA-IMM 算法更慢。蒙特卡洛贪心的复杂度为模拟复杂度，因此种子越少跑的越快。

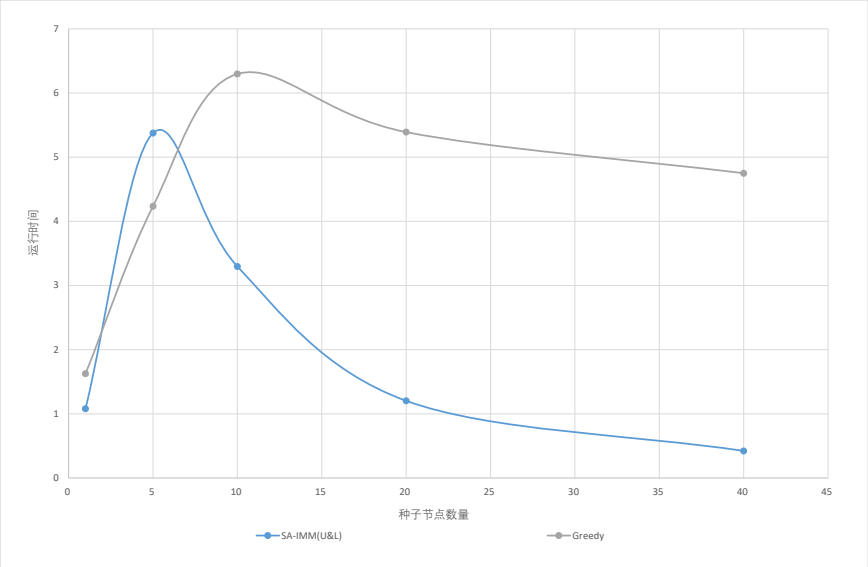

图 4.6: 种子数量-运行时间 SA-IMM 算法对照实验

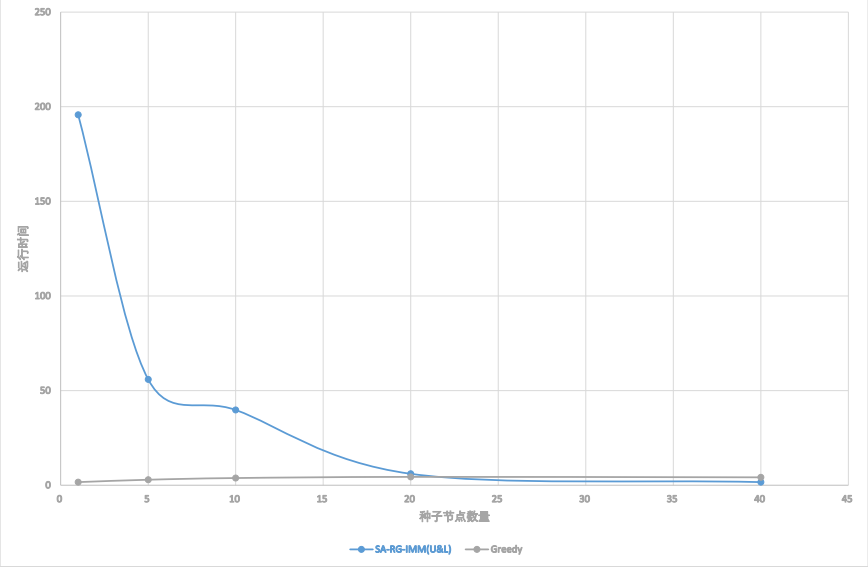

图 4.7: 种子数量-运行时间 SA-RG-IMM 算法对照实验

4.3.4 种子数量-运行效果

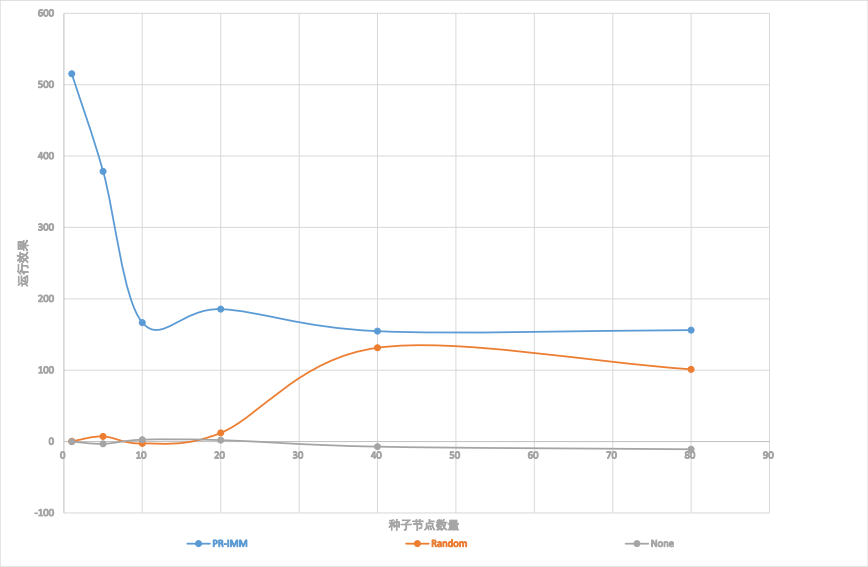

图 4.8: 种子数量-运行效果 PR-IMM 算法对照实验

图 4.8 到图 4.10 可以看到，种子数量越少时能够优化的空间越多，因此三张图大体上都有种子数量越多，运行效果越差的趋势。SA-IMM 算法和 SA-RG-IMM 算法在种子数提高时始终能够保持一定优势，而 PR-IMM 算法的图中，PR-IMM 和随机的差距逐渐减小，估计是在大图上种子数量多时优化已经到了一定的瓶颈。

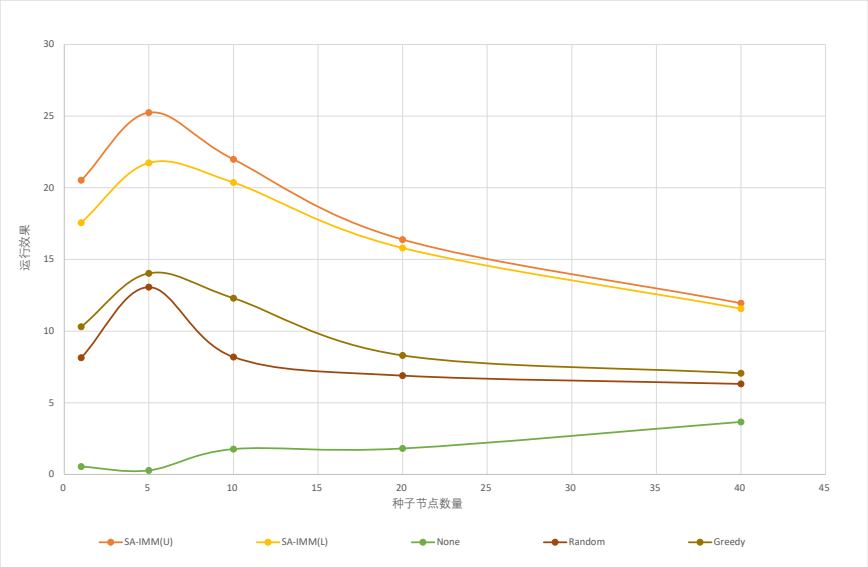

图 4.9: 种子数量-运行效果 SA-IMM 算法对照实验

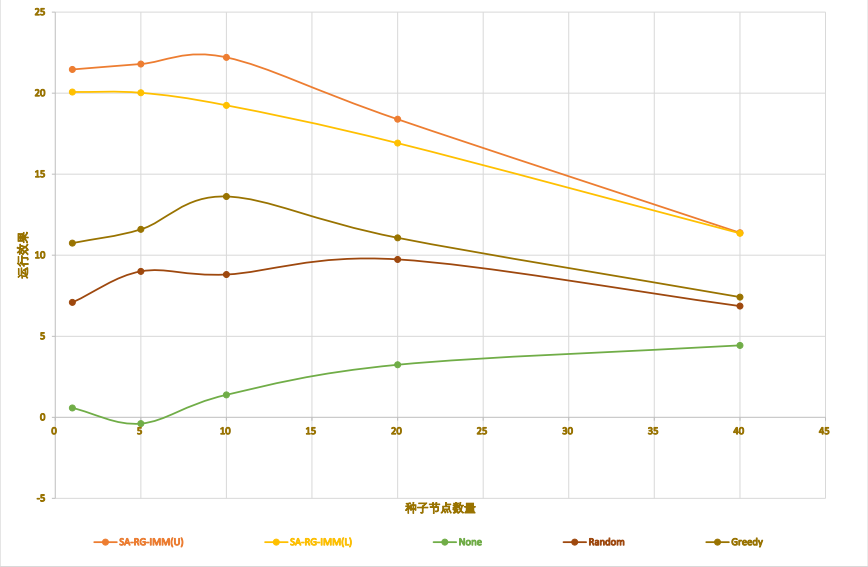

图 4.10: 种子数量-运行效果 RG-IMM 算法对照实验

4.3.5 k-运行时间

在 k 作为自变量的实验中，除去 k 之外，其他非自变量参数和上一实验相同。

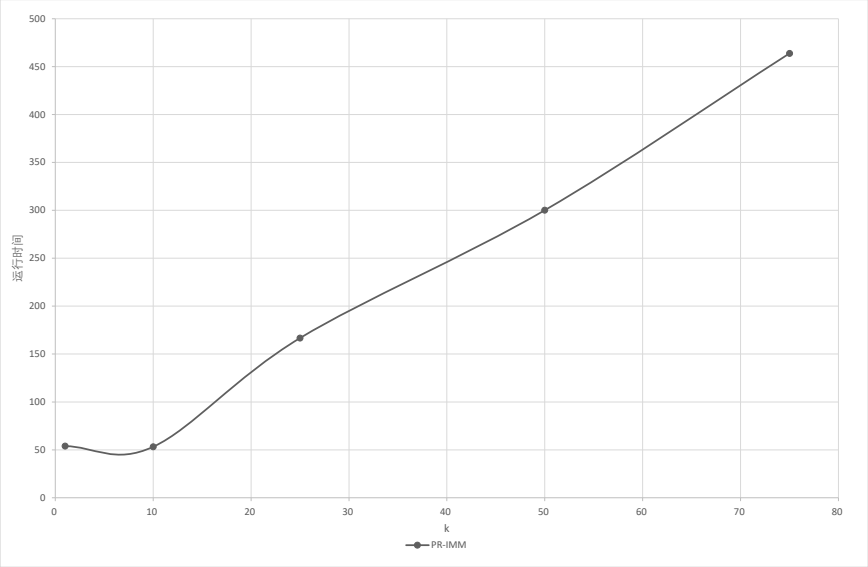

图 4.11: k-运行时间 PR-IMM 算法对照实验

图 4.11 到图 4.13 可以看到，总的来说，k 在 PR-IMM 算法中和复杂度线性相关，图上也确实呈现出如此。在 SA-IMM 算法和 SA-RG-IMM 算法方面，k 对于算法的时间影响不是很大，主要复杂度在于采样和计算贡献计算。

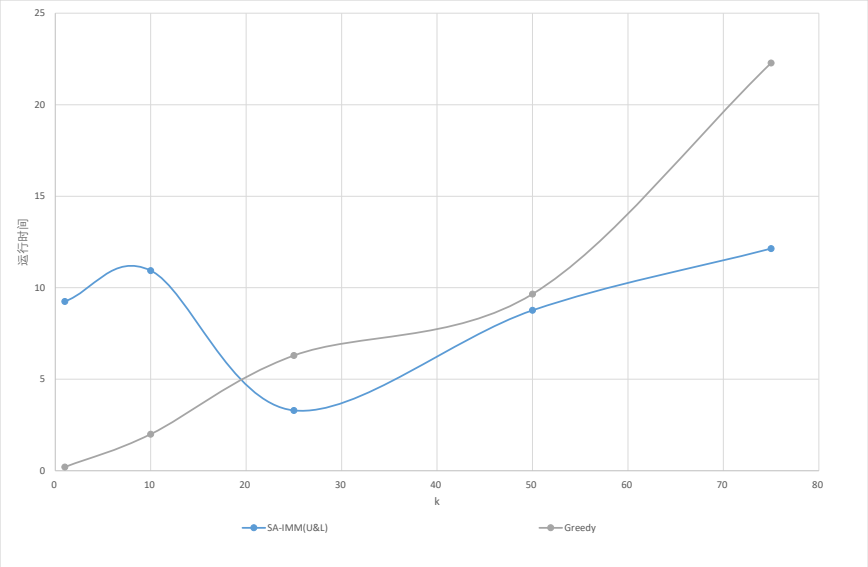

图 4.12: k-运行时间 SA-IMM 算法对照实验

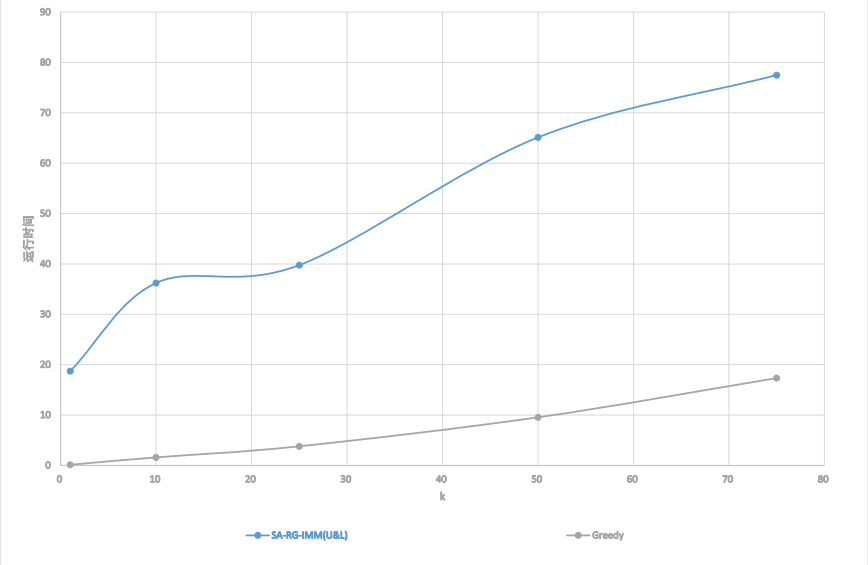

图 4.13: k-运行时间 SA-RG-IMM 算法对照实验

4.3.6 k-运行效果

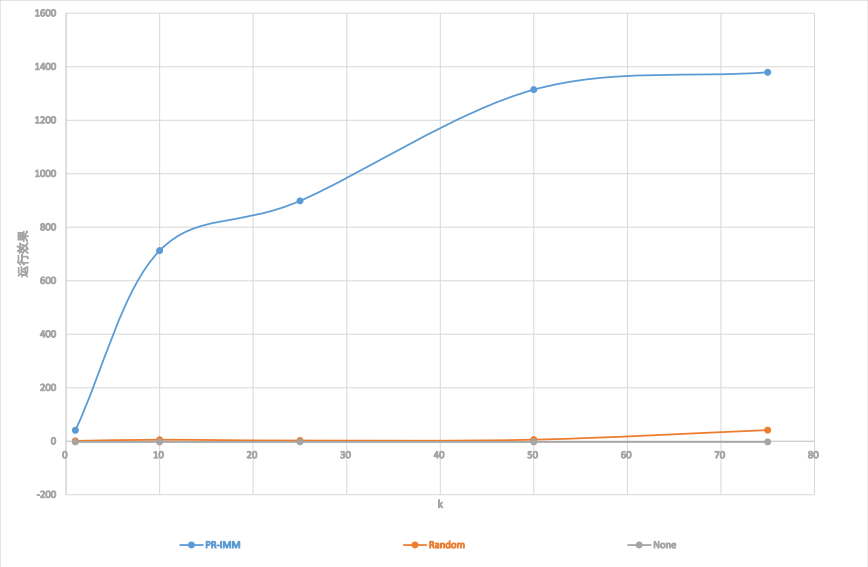

图 4.14: k-运行效果 PR-IMM 算法对照实验

图 4.14 到图 4.16 可以看到，k 直接代表了算法对图能影响的能力，因此三张图都表现出 k 越大，影响越高。三个算法在在 k 较高时达到一定的瓶颈，其中 PR-IMM 算法最为明显。同时可以看到 PR-IMM 算法在 k 变大时远超随机效果，而 SA-IMM 算法和 SA-RG-IMM 算法则和对照算法逐步上升。

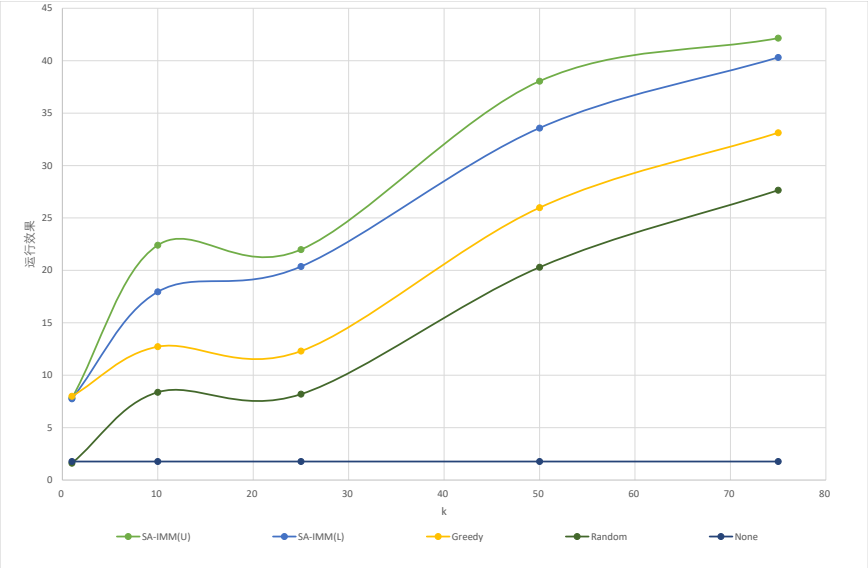

图 4.15: k-运行效果 SA-IMM 算法对照实验

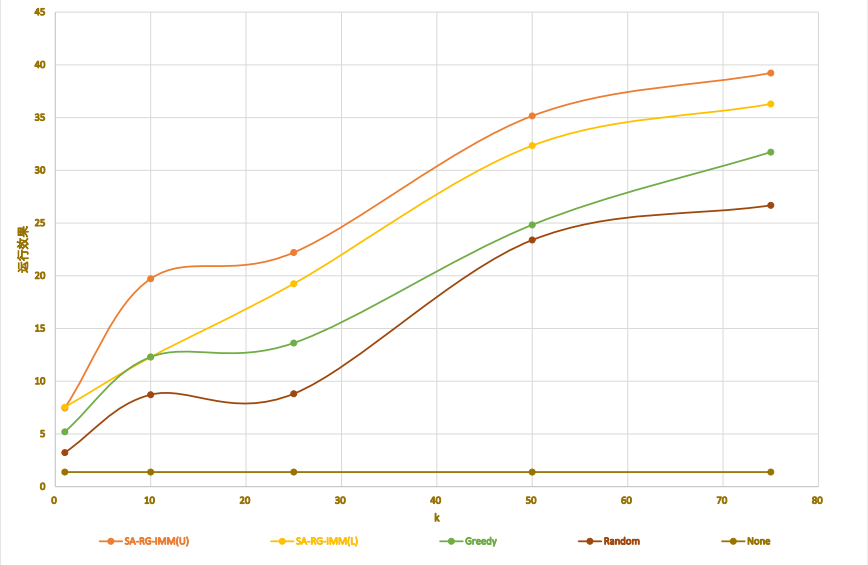

图 4.16: k-运行效果 RG-IMM 算法对照实验

4.3.7 λ-运行时间

在 λ 作为自变量的实验中，除去 λ 之外，其他非自变量参数和上一实验相同。

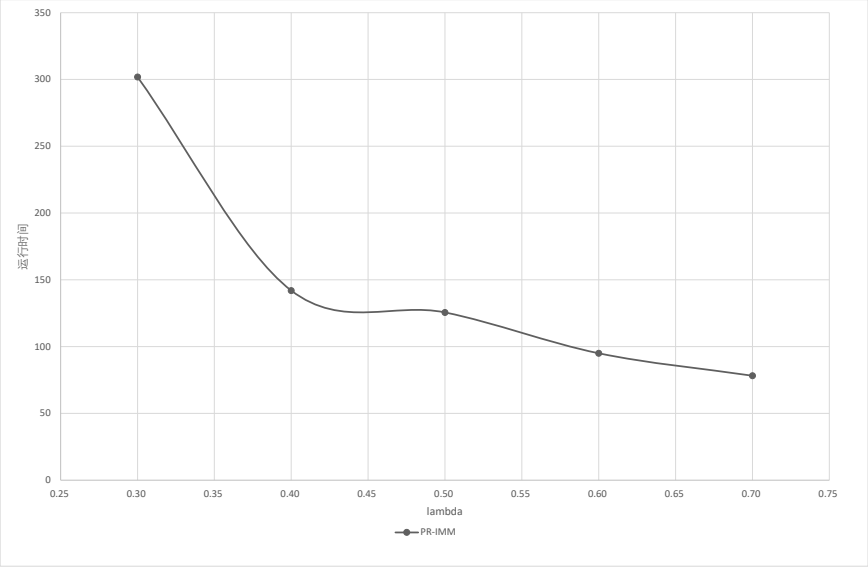

图 4.17: λ-运行时间 PR-IMM 算法对照实验

图 4.17 到图 4.19 可以看到，当 λ 很小时，负面信息抑制占据主要贡献。这样的影响使得 OPT 更小，估算也更加困难，因此三个算法都呈现 λ 越小时间越长的趋势。

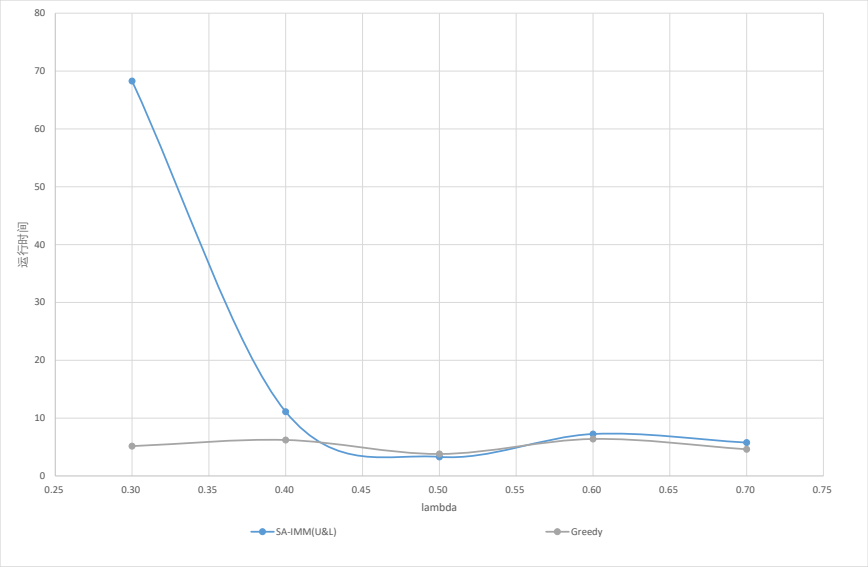

图 4.18: λ-运行时间 SA-IMM 算法对照实验

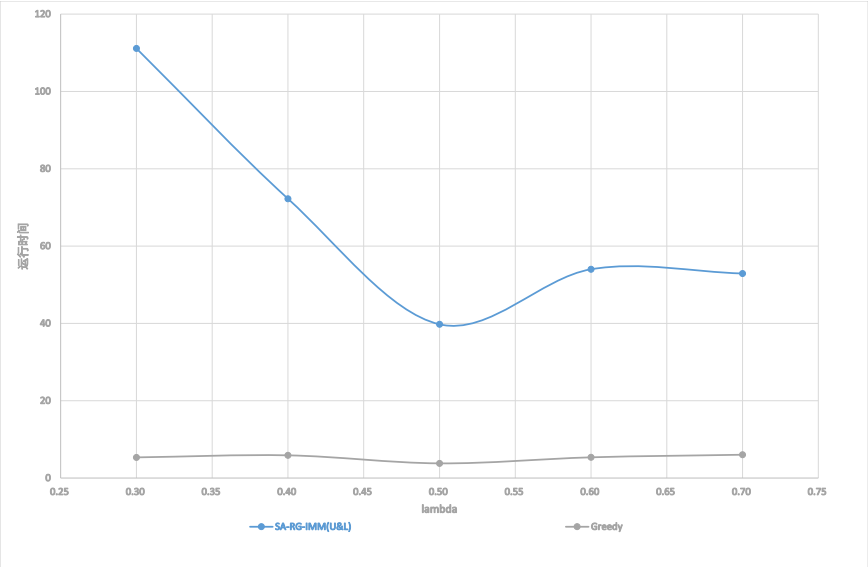

图 4.19: λ-运行时间 SA-RG-IMM 算法对照实验

4.3.8 λ-运行效果

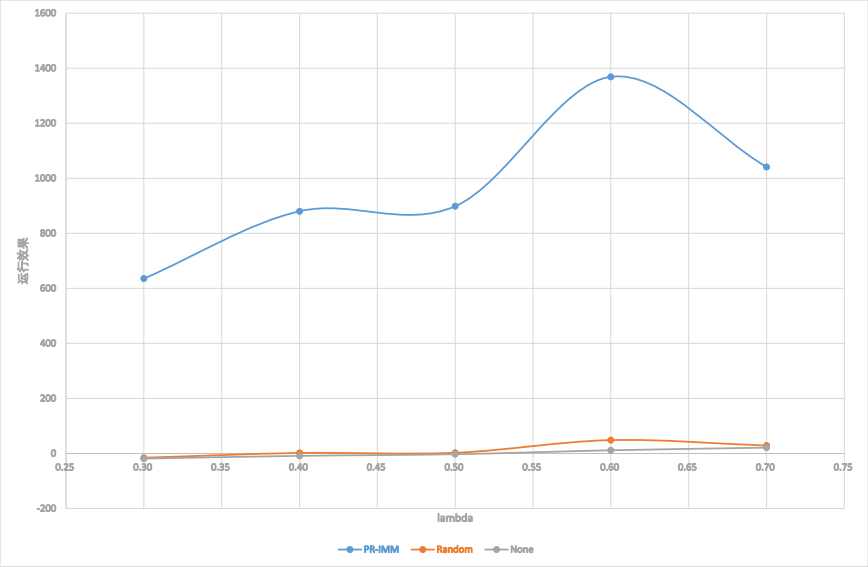

图 4.20: λ-运行效果 PR-IMM 算法对照实验

图 4.20 到图 4.22 可以看到，如上所述，λ 越小，负面信息抑制占据主要贡献。这样的影响使得 OPT 更小，估算也更加困难，因此三个算法都呈现时间越长的趋势。

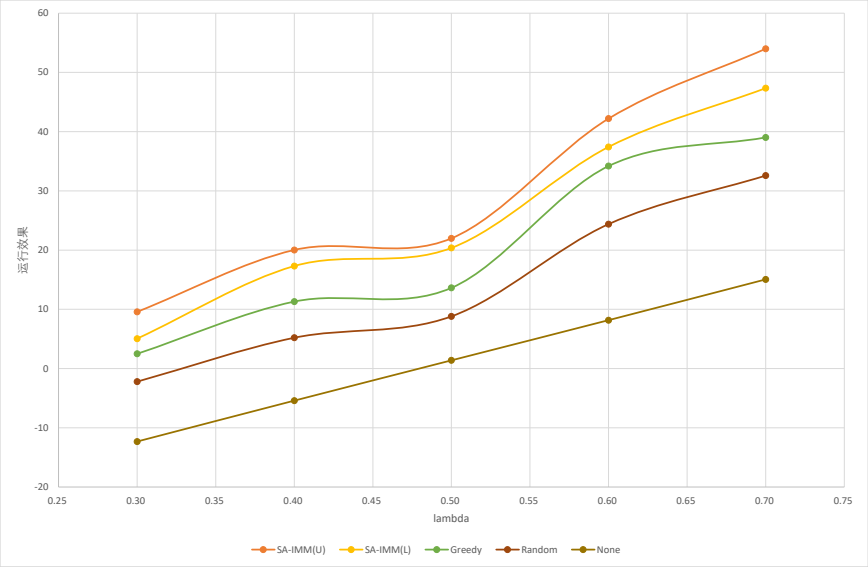

图 4.21: λ-运行效果 SA-IMM 算法对照实验

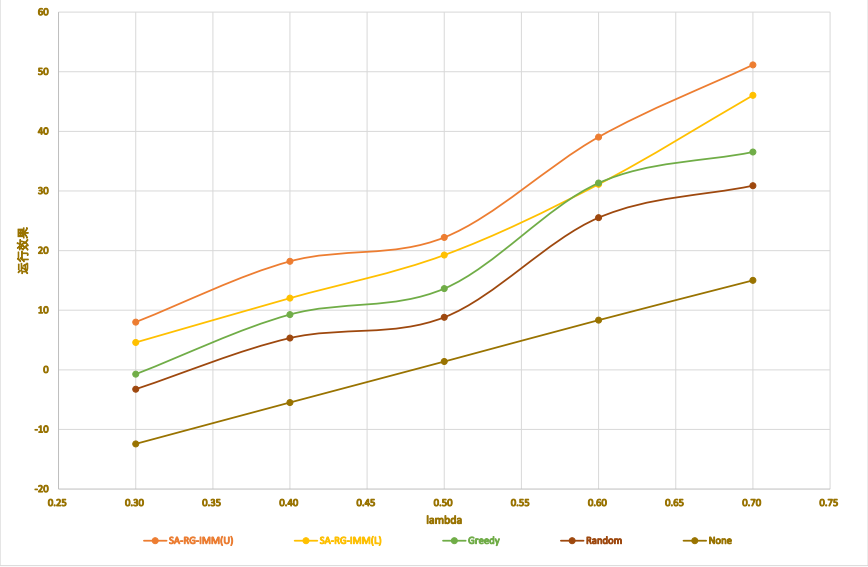

图 4.22: λ-运行效果 RG-IMM 算法对照实验

4.3.9 ϵ_2-运行时间

在 ϵ_2 作为自变量的实验中，除去 ϵ_2 之外，其他非自变量参数和上一实验相同。

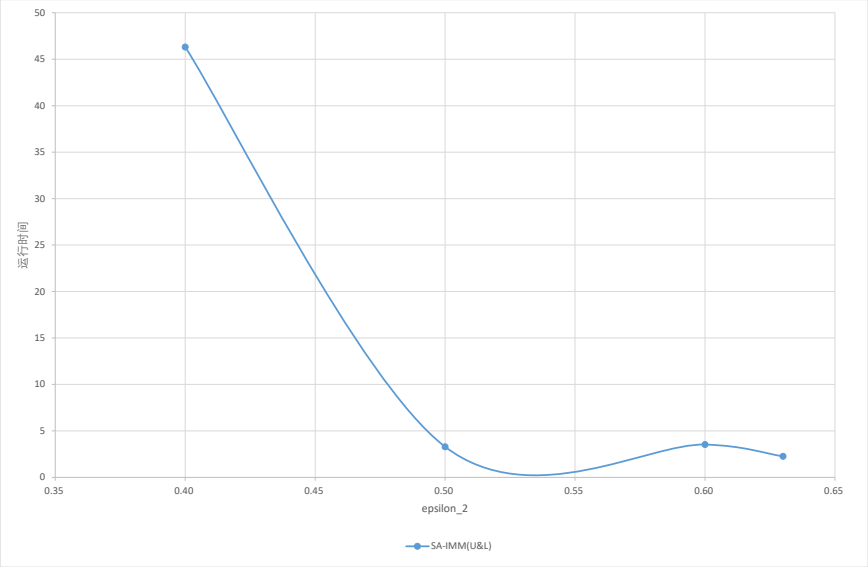

图 4.23: ϵ_2-运行时间 SA-IMM 算法对照实验

图 4.23 到图 4.24 可以看到，ϵ_2 只对 SA-IMM 算法和 SA-RG-IMM 算法有效。两张图中我们可以明显看到 ϵ_2 的减小对运行时间的巨大变化，符合我们对其时间复杂度分析的预期。

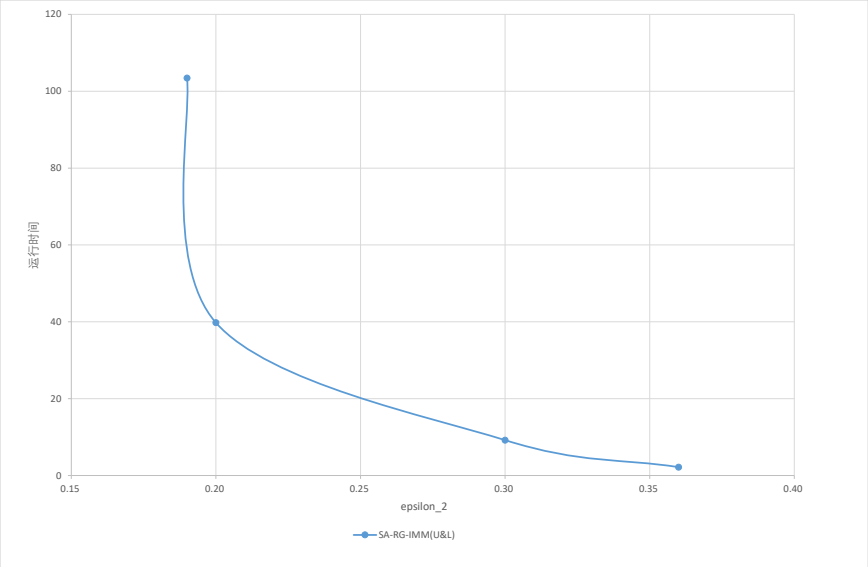

图 4.24: ϵ_2-运行时间 SA-RG-IMM 算法对照实验

4.3.10 ϵ_2-运行效果

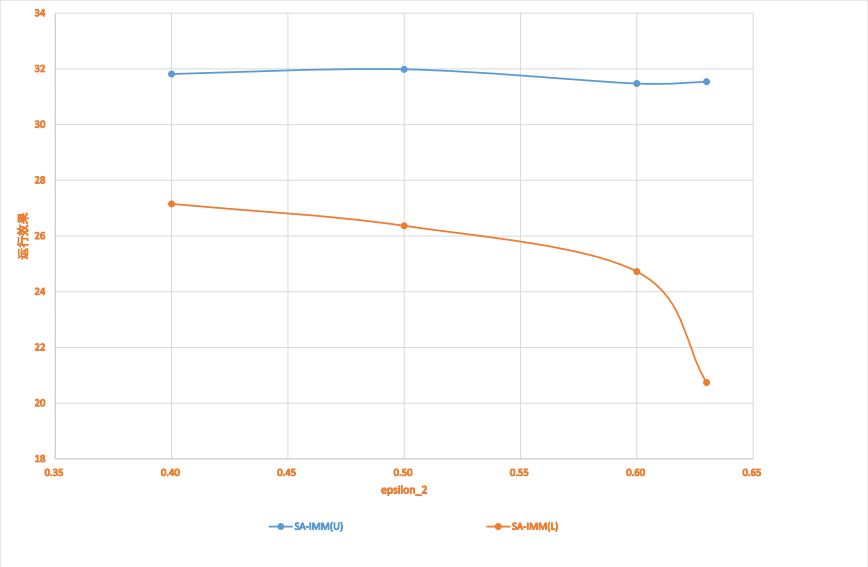

图 4.25: ϵ_2-运行效果 SA-IMM 算法对照实验

图 4.25 到图 4.26 可以看到，同样地，两张图中我们可以明显看到 ϵ_2 的减小对运行效果的影响。注意到两者的上界算法本质就是 PR-IMM 算法，因此 ϵ_2 对其没有影响。

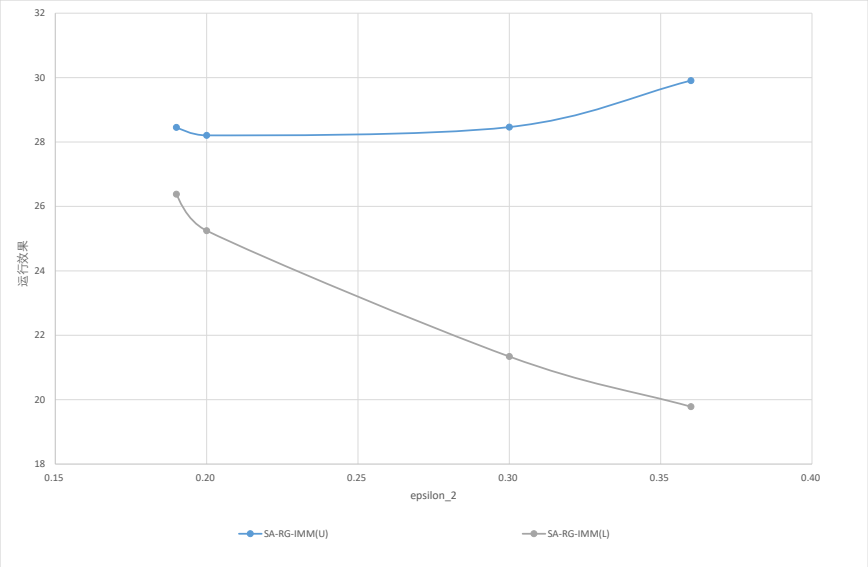

图 4.26: ϵ_2-运行效果 SA-RG-IMM 算法对照实验

5 结论

本文首先在影响最大化问题[19]，谣言限制问题[18] 和负面信息问题[20] 基础上进行结合，得到了一个新模型。在这一模型上进行了详细的数学分析，根据单调性和子模性是否成立，给出了具体证明，并得到了不同优先级下的不同算法，分析了其复杂度。最终通过真实数据上的实验，验证了提出的算法确实可以达到优于对照算法的优化效果。

本文当中的问题还有很多发展的方向，首先本文给出的模型虽然较为复杂，但仍然只有两类初始信息，可以考虑更加复杂的模型、引入不定量的信息，使其贴合实际情况；本文给出的算法部分还有优化的空间，包括在非单调或非子模情况下，目前的实验结果完全无法和单调子模的结果相比较。本文使用的启发式计算贡献有使用其他高级算法代替的可能，部分优先级也可能有比启发式计算好的多的方法。最后，本文虽然使用了大量真实数据，但由于特殊模型导致需要一些启发式的参数来配合数据使用，在未来可以通过分析具体数据的方法，使用机器学习获取更好的传播参数，以取得更贴合实际的数据。

6 参考文献

[1] TASNIM S, HOSSAIN M M, MAZUMDER H. Impact of Rumors and Misinformation on COVID-19 in Social Media[J]. Journal of Preventive Medicine and Public Health, 2020, 53(3): 171-174.

[2] LI Y, FAN J, WANG Y, et al. Influence Maximization on Social Graphs: A Survey[J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 30(10): 1852-1872. DOI: 10.1109/TKDE.2018.2807843.

[3] NEMHAUSER G L, WOLSEY L A, FISHER M L. An analysis of approximations for maximizing submodular set functions—I[J]. Mathematical programming, 1978, 14(1): 265-294.

[4] BORODIN A, FILMUS Y, OREN J. Threshold models for competitive influence in social networks[C]//International workshop on internet and network economics. [S.l. : s.n.], 2010: 539-550.

[5] TANG Y, SHI Y, XIAO X. Influence maximization in near-linear time: A martingale approach[C]//Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. [S.l. : s.n.], 2015: 1539-1554.

[6] LESKOVEC J, MCAULEY J. Learning to Discover Social Circles in Ego Networks[C/OL]// PEREIRA F, BURGES C J C, BOTTOU L, et al. Advances in Neural Information Processing Systems:vol. 25. [S.l.]: Curran Associates, Inc., 2012. https://proceedings.neurips.cc/paper/2012/file/7a614fd06c325499f1680b9896beedeb-Paper.pdf.

[7] CHEN W, WANG Y, YANG S. Efficient Influence Maximization in Social Networks[C/OL]// Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Paris, France: Association for Computing Machinery, 2009: 199-208. https://doi.org/10.1145/1557019.1557047. DOI: 10.1145/1557019.1557047.

[8] KEMPE D, KLEINBERG J, TARDOS É. Maximizing the Spread of Influence through a Social Network[C/OL]//Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Washington, D.C.: Association for Computing Machinery, 2003: 137-146. https://doi.org/10.1145/956750.956769. DOI: 10.1145/956750.956769.

[9] KHULLER S, MOSS A, NAOR J S. The budgeted maximum coverage problem[J]. Information processing letters, 1999, 70(1): 39-45.

[10] KRAUSE A, GUESTRIN C. A note on the budgeted maximization of submodular functions[M]. [S.l.]: Citeseer, 2005.

[11] LESKOVEC J, KRAUSE A, GUESTRIN C, et al. Cost-effective outbreak detection in networks[C]//Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. [S.l. : s.n.], 2007: 420-429.

[12] BORGS C, BRAUTBAR M, CHAYES J, et al. Maximizing social influence in nearly optimal time[C]//Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms. [S.l. : s.n.], 2014: 946-957.

[13] KEMPE D, KLEINBERG J, TARDOS É. Influential nodes in a diffusion model for social networks[C]//International Colloquium on Automata, Languages, and Programming. [S.l. : s.n.], 2005: 1127-1138.

[14] BHARATHI S, KEMPE D, SALEK M. Competitive influence maximization in social networks[C]//International workshop on web and internet economics. [S.l. : s.n.], 2007: 306-311.

[15] TONG G, WU W, GUO L, et al. An efficient randomized algorithm for rumor blocking in online social networks[J]. IEEE Transactions on Network Science and Engineering, 2017.

[16] TONG G A, DU D Z. Beyond uniform reverse sampling: A hybrid sampling technique for misinformation prevention[C]//IEEE INFOCOM 2019-IEEE Conference on Computer Communications. [S.l. : s.n.], 2019: 1711-1719.

[17] TONG A, DU D Z, WU W. On misinformation containment in online social networks[C]// Advances in neural information processing systems. [S.l. : s.n.], 2018: 341-351.
[18] BUDAK C, AGRAWAL D, EL ABBADI A. Limiting the spread of misinformation in social networks[C]//Proceedings of the 20th international conference on World wide web. [S.l. : s.n.], 2011: 665-674.
[19] LIN Y, CHEN W, LUI J C. Boosting information spread: An algorithmic approach[C]// Data Engineering (ICDE), 2017 IEEE 33rd International Conference on. [S.l. : s.n.], 2017: 883-894.
[20] CHEN W, COLLINS A, CUMMINGS R, et al. Influence maximization in social networks when negative opinions may emerge and propagate[C]//Proceedings of the 2011 siam international conference on data mining. [S.l. : s.n.], 2011: 379-390.
[21] VAZIRANI V V. Approximation algorithms[M]. [S.l.]: Springer Science & Business Media, 2013.
[22] LU W, CHEN W, LAKSHMANAN L V. From competition to complementarity: comparative influence diffusion and maximization[J]. Proceedings of the VLDB Endowment, 2015, 9(2): 60-71.
[23] YAN R, LI Y, WU W, et al. Rumor blocking through online link deletion on social networks[J]. ACM Transactions on Knowledge Discovery from Data (TKDD), 2019, 13(2): 1-26.
[24] BUCHBINDER N, FELDMAN M, NAOR J, et al. Submodular maximization with cardinality constraints[C]//Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms. [S.l. : s.n.], 2014: 1433-1452.
[25] TONG G, WU W, DU D Z. Distributed Rumor Blocking With Multiple Positive Cascades[J]. IEEE Transactions on Computational Social Systems, 2018, 5(2): 468-480. DOI: 10.1109/TCSS.2018.2818661.
[26] ROZEMBERCZKI B, ALLEN C, SARKAR R. Multi-scale Attributed Node Embedding[Z]. 2019arXiv: 1909.13021 [cs.LG].

作者简历

姓名：刘明锐

性别：男

民族：汉

出生年月：1999-5-24

籍贯：江苏句容

2014.09-2017.06 上海市格致中学

2017.09-2021.06 浙江大学攻读学士学位

获奖情况：

2017 年第四十二届国际大学生程序设计竞赛区域赛 (青岛) 冠军

2018 年第十五届浙江省大学生程序设计竞赛冠军

2018 年第三届中国大学生程序设计竞赛区域赛 (吉林) 季军

2018 年参加第四十二届国际大学生程序设计竞赛世界总决赛

2017 年第四十二届国际大学生程序设计竞赛区域赛 (北京) 金奖

2018 年第四十三届国际大学生程序设计竞赛区域赛 (徐州) 金奖

2019 年第四十四届国际大学生程序设计竞赛区域赛 (徐州) 金奖

2019 年第四十四届国际大学生程序设计竞赛区域赛 (沈阳) 金奖

2017 年第四十四届国际大学生程序设计竞赛东大陆总决赛金奖

2019 年第四十四届国际大学生程序设计竞赛东大陆总决赛金奖

2017 年第二届中国大学生程序设计竞赛区域赛 (哈尔滨) 金奖

2019 年第二届中国大学生程序设计竞赛区域赛 (秦皇岛) 金奖

2017 年第二届中国大学生程序设计竞赛总决赛金奖

2016 年中国信息学奥林匹克竞赛 NOI 全国决赛银牌

2017-2018 年浙江大学基础学科拔尖人才一等奖学金

2018-2019 年浙江大学基础学科拔尖人才一等奖学金

本科生毕业论文（设计）任务书

一、题目：

二、指导教师对毕业论文（设计）的进度安排及任务要求：

根据开题报告中提出的研究计划，学生要在3月完成相关论文的阅读和学习，4月中旬完成算法设计，在5月上旬完成算法实现和实验，并在5月中下旬完成毕业论文撰写工作，完成研究计划中提出的协作与竞争并存的社交网络影响最大化问题研究。

起讫日期　2020　年　6　月　6日　至　2021　年　6　月　6　日

指导教师（签名）_____ 职称 _____

三、系或研究所审核意见：

负责人（签名）_____

年　月　日

本科生毕业论文（设计）考核

一、指导教师对毕业论文（设计）的评语：

刘明锐同学在其毕业论文《协作与竞争并存的社交网络影响最大化问题研究》中，通过较为全面的相关文献调研与综述，确定了研究方案，提出了对社交网络中协作和竞争并存的信息传播以及有限资源进行传播控制进行了建模，提出了该模型上的多个不同算法，通过编程与实验验证了模型的有效性，达到了导师的要求。毕业论文语句通顺，逻辑严密，条理清楚，有较好的创新性和理论价值,达到了本科生毕业论文的水平。

指导教师（签名）_____

年　月　日

二、答辩小组对毕业论文（设计）的答辩评语及总评成绩：

成绩比例	文献综述（10%）	开题报告（15%）	外文翻译（5%）	毕业论文质量及答辩（70%）	总评成绩
分值					

负责人（签名）_____

年　月　日