电 子 科 技 大 学

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

# 硕士学位论文

**MASTER THESIS**


论文题目 网络中影响力传播的最大化

学 科 专 业 计算机应用技术

学 号 **201221060451**

作 者 姓 名 陈雪峰

指 导 教 师 向艳萍 教授

分类号 _____ 密级 _____

UDC [注1] _____

# 学　位　论　文

## 网络中影响力传播的最大化

（题名和副题名）


陈雪峰

（作者姓名）

指导教师　向艳萍　教授

电子科技大学　成都

(姓名、职称、单位名称)


申请学位级别　硕士　学科专业　计算机应用技术

提交论文日期　2015.03　论文答辩日期　2015.05

学位授予单位和日期　电子科技大学　2015.06

答辩委员会主席 _____

评阅人 _____

注 1：注明《国际十进分类法 UDC》的类号。

# INFLUENCE MAXIMIZATION IN NETWORKS

A Master Thesis Submitted to

University of Electronic Science and Technology of China


Major: **Computer Applied Technology**

Author: **Chen Xuefeng**

Advisor: **Professor Xiang Yanping**

School: **School of Computer Science & Engineering**

# 独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，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，也不包含为获得电子科技大学或其它教育机构的学位或证书而使用过的材料。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。

作者签名：_____ 日期： 年 月 日

# 论文使用授权

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权电子科技大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

(保密的学位论文在解密后应遵守此规定)

作者签名：_____ 导师签名：_____

日期： 年 月 日

# 摘 要

影响力传播的最大化问题的目标是在网络中寻找影响力最大的$K$个点，使得从这$K$个点传播出的影响力（如信息、想法、观点等）期望影响到的节点数是最大的。该问题是社交网络的基础研究问题，有助于推动影响力传播的分析和建模。同时，该问题也存在很大的应用价值，如口碑营销。

影响力传播的最大化问题已经吸引了大量的研究工作，但是，之前的研究工作都忽视了两个重要影响因子，一个是影响力传播过程中的新颖性衰变现象，即重复的被影响会减少影响力对于用户的作用；另一个是节点失效而导致的影响力丢失现象。

本文将研究这两个影响因子对影响力传播的最大化问题的作用，主要工作如下：

(1) 在真实数据集中分析了新颖性衰变对于影响力传播过程的影响并设计了一个拟合函数量化这个影响，还定义了新颖性衰变下影响力传播的最大化问题。不同于普通的影响力传播模型，新颖性衰变下传播模型中的点集影响计算公式既不是单调也不是子模的，这意味着普通的贪婪算法和已知的计算点集影响力的方法不再适用。为此，本文提出了一个限制性贪婪算法和动态剪枝优化，还提出了两个新的计算点集影响力的方法。在4个真实数据集上的实验结果验证了提出的方法的有效性和高效性。

(2) 在分析节点失效对于影响力传播的最大化问题的作用基础上，本文把节点失效下影响力传播的最大化问题定义为一个限制性非线性优化问题。由于普通的贪婪算法无法解决这个新的影响力传播的最大化问题，本文提出了一个限制性模拟退火算法，还通过高效估算点集影响力丢失的方法提高该算法的效率。本文还提供了在4个真实数据集上的实验结果作详细说明。

**关键词：**影响力传播的最大化，社交网络，新颖性衰变，节点失效

# ABSTRACT

The problem of influence maximization is to select a set of $K$ nodes from a social network so that the spread of influence (e.g., information, idea, opinion etc) is maximized over the network. Influence maximization is a fundamental research problem in social networks; it is able to promote the analysis and modeling of influence propagation. And influence maximization is useful in a large number of applications such as word-of-mouth marketing.

Influence maximization has attracted significant research work. However, previous research work overlooks two important factors. One is novelty decay phenomenon in influence propagation, i.e., repeated exposures will have diminishing influence on users. Another one is influence loss due to node failure. This thesis will study the effects of the two important factors on influence maximization. The main work is as follows:

(1) Analyze the effect of novelty decay on influence propagation on real-life datasets, develop a fitting function to characterize the effect, and formulate the problem of influence maximization with novelty decay. Different from the conventional influence propagation model, the influence function with novelty decay is neither monotone nor submodular; it means that the greedy algorithm and the existing methods for computing the influence of seed set are inapplicable. To this end, this thesis develops a restricted greedy algorithm and a dynamic pruning optimization, proposes two methds for computing the influence of seed set. Experiments conducted on four real-life datasets demonstrate the efficiency and effectiveness of the proposed approaches.

(2) Based on the analysis of the effects of node failure on influence maximization, this thesis defines the problem of influence maximization with node failure as a constrained nonlinear optimization problem. Since the greedy algorithm can not solve the new type of influence maximization problem, this thesis proposes a constrained simulated annealing algorithm and further improve its performance through efficiently estimating the influence loss. This thesis also provides experimental results over four real-life datasets in support.

**Keywords:** Influence Maximization, Social Networks, Novelty Decay, Node Failure

# 目 录

# 第一章 绪论

## 1.1 研究背景及意义

近几年来，作为社交网络研究领域的基础问题，影响力传播的最大化问题吸引了大量的研究和关注[1-10]。该问题的目标是在网络中寻找影响力最大的$K$个点，使得从这$K$个点传播出的影响力（如信息、想法、观点等）期望影响到的节点数是最大的。影响力传播的最大化问题在社交网络（如脸谱网、微信、人人网、新浪微博等）中存在广泛的应用，其中最典型的就是病毒营销（或口碑营销）。

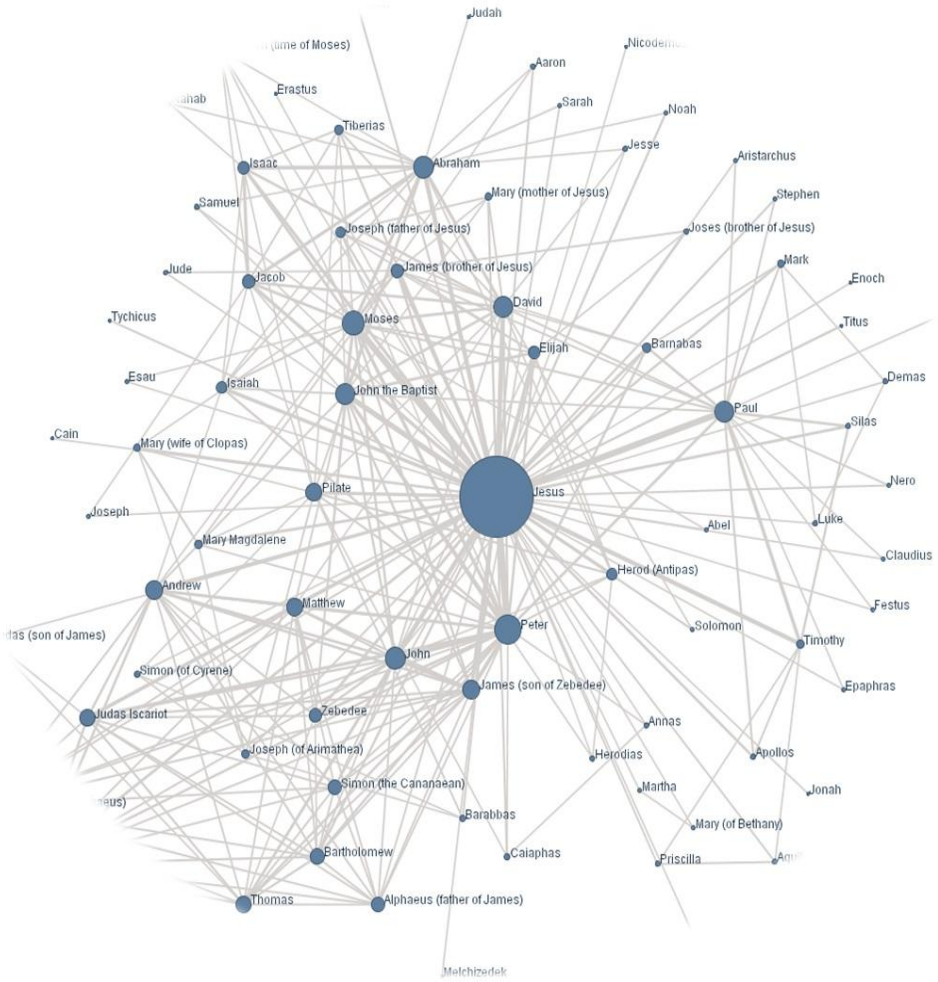


图1-1 一个社交网络的例子

一个社交网络的例子如图 1-1 所示，它由成千上万的用户（圆圈所示，圆圈大小表示该用户受关注的程度）和用户间建立起来的相互连接关系（如朋友关系）组成，相互连接的用户间可以观察到对方发布的信息，并将该信息继续发布，这

个过程使得影响力可以在网络中不断地传播。在此基础上，病毒营销致力于在社交网络中选择一部分有影响力的用户接受某个产品，以期通过口碑影响力的传播方式让社交网络中尽量多的用户接受该产品[11-12]。例如，一家新开张的中餐馆为了吸引更多的客户，在脸谱网上选择一部分有影响力（粉丝多）的客户作为免费试吃对象，并希望他们能把该店的口碑（如菜品好，服务好等）传遍脸谱网。

除了社交网络的迅速发展和普及，病毒营销能够在实际推广产品中取得成效的原因还在于，相比于陌生人的购买行为和意见，人们更倾向于关心朋友的购买行为，相信朋友提供的信息，体现容易被朋友影响的特性。一个明确的市场调查发现，在购买家用电器前，68%的用户咨询了其亲戚或者朋友，比上网搜索相关产品信息的用户多了一半[32]。

病毒营销比较成功的一个案例是邮件服务（如 Hotmail、Yahoo、Gmail 等）的推广，Hotmail 曾经仅在传统营销上花费了 50,000 美元就能在 18 个月内将用户数量从 0 激增到 12,000,000[51]。同时，Hotmail 用户数量的增长超过了很多媒体公司（如 CNN、AOL 等）。到 2000 年中旬，Hotmail 已经拥有了超过 66,000,000 的用户群，且每天有 270,000 的新账号被注册[52]。事实说明，病毒营销的市场价值和前景是非常巨大的。

影响力传播的最大化问题在有影响力传播性质的其他网络（如交通网络、水分布网络等）中也存在很大的应用前景。比如，水分布网络，如图 1-2 所示，它由很多关键地点（如河流交汇点、水塔等）及关键地点间相互连接通道（如河道、供水管道等）组成，使得水中的污染物可以随着水流在整个水分布网络中传播。为了监控水质情况，需要选择部分关键地点安放传感器，所以，在水分布网络中选择合适关键地点安放传感器使得其能最大限度监测水质情况成为了一个很有价值的研究问题[4,39-41]。其中，Leskovec 等人发现博客网络（一种社交网络）和水分布网络有相似结构, 并参考解决博客网络中影响力传播的最大化问题的算法提出了一个选择合适关键地点安放传感器的高效方法[4]。

此外，影响力传播的最大化问题的研究有助于推进网络中影响力传播的建模，影响力传播规律和相关影响因子的分析，寻找影响力传播源头等课题的研究。

为解决影响力传播的最大化问题，Kempe 等人提出了两个基本的、且被广泛研究的传播模型：线性阈值模型（linear threshold model）和独立级联模型（independent cascade model）[3]，在此基础上，Chen 等人[18]和 Liu 等人[17]分别考虑时间因子，提出了包含时间因子的独立级联模型；Chen 等人还建立了包含积极影响力和消极影响的独立级联模型[25]。此外，Goyal 等人提出了可以跟踪和学习用户历史数据的信用分布（credit distribution）模型[43]。这些模型都很好地揭示了网络中影响力传

播的过程和规律，为进一步影响力传播的分析打下基础。

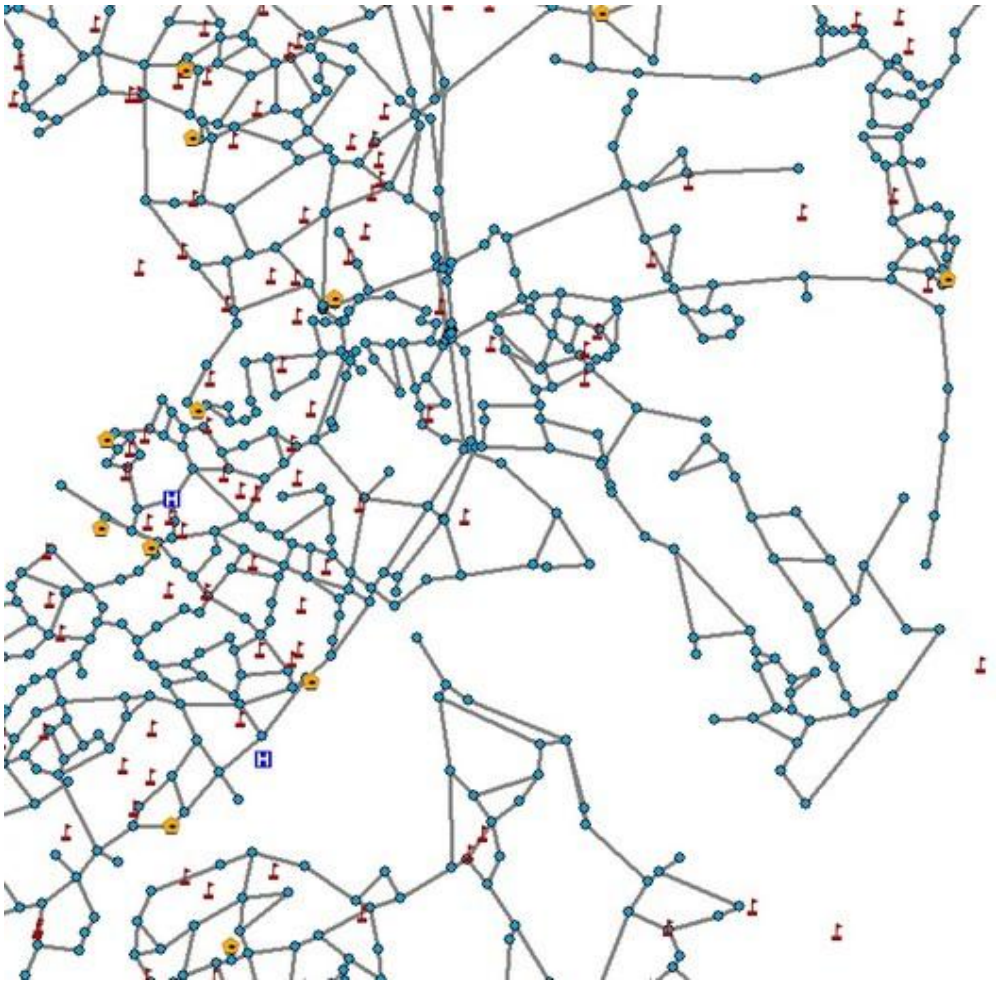

图1-2 一个水分布网络的例子

值得一提的是，He 等人在考虑竞争的影响力（如积极的影响力和消极的影响力）同时传播的基础上，提出了社交网络中阻碍影响力传播的最大化问题，即寻找社交网络中最合适的 $K$ 个点，以期最大限度地限制其他影响力的传播[42]。由于社交网络影响力传播模型和传染病传播模型非常相似，传染病的防疫问题可以借鉴社交网络中阻碍影响力传播的最大化问题的相关分析和解决方法。

## 1.2 主要研究内容

从上面的介绍可以看出，影响力传播的最大化问题的研究对于有影响力传播性质的网络的建模分析，及相关应用问题的解决有重要意义。但已有的研究都忽略了两个重要的影响因子：新颖性衰变和节点失效导致的影响力丢失。本文分别分析这两个影响因子对影响力传播的最大化问题的作用，并提出两个重要子问题

——新颖性衰变下的影响力传播的最大化问题和节点失效下的影响力传播的最大化问题建模分析，并提出相应的解决方法。

本文的具体研究内容包括：

(1) 基于传统的影响力传播模型和真实的数据集，分析新颖性衰变（节点失效）对影响力传播过程的影响，并建立相应的影响力传播模型。

(2) 标准化定义新颖性衰变（节点失效）下影响力传播的最大化问题。

(3) 分析并证明新定义问题的相关性质。

(4) 根据新问题的影响力传播模型提出（或改进）高效的方法计算点集的影响力。

(5) 根据新问题的相关性质提出（或改进）高效的选取影响力最大的点集的策略。

(6)编程实现提出（或改进）方法，并通过真实的数据集中做模拟实验，验证其高效性。

## 1.3 论文的组织结构

本文共分为五章，具体的组织结构如下：

第一章 绪论。首先介绍了影响力传播的最大化问题的研究背景、研究意义，然后介绍了本文的主要研究内容并给出了本文的组织结构。

第二章 相关研究工作与背景知识。首先，全面、详细地介绍了影响力传播的最大化问题的相关研究工作；然后，对本文用到的一些关键技术做了简单介绍，包括目前最好的快速计算点集影响力的方法之一的影响力传播路径方法（ISP），以及解决限制性非线性优化问题的限制性模拟退火算法。

第三章 新颖性衰变下影响力传播的最大化问题。首先，讨论了新颖性衰变对影响力传播过程的影响并标准化新颖性衰变函数；然后，建立了新颖性衰变下影响力传播模型，定义了新颖性衰变下影响力传播的最大化问题；接着，提出了高效的选出影响力最大点集的贪婪算法和动态剪枝优化，以及计算点集影响力的方法；最后，通过 4 个真实数据集上的实验结果验证了提出方法的高效性。

第四章 节点失效下影响力传播的最大化问题。首先，分析节点失效对影响力传播的最大化问题的影响，并标准化定义节点失效下影响力传播的最大化问题；然后，在新问题性质的基础上提出了限制性模拟退火算法及其优化策略；最后，在 4 个真实数据集上实验验证了提出方法的高效性。

第五章 总结与展望。对本文中影响力传播最大化问题的研究进行了总结和概括，指出本文的主要研究成果以及下一步的研究方向。

## 1.4 本章小结

本章首先介绍了影响力传播的最大化问题的概念及重要意义，然后列出了本文的主要研究内容和组织结构。

# 第二章 相关研究工作与背景知识

本章首先介绍影响力传播的最大化问题的相关研究工作，然后介绍后文用到的一些关键技术和背景知识。主要符号表如表 2-1 所示，

表2-1 主要符号表

| 数学符号 | 表达含义 |
|---|---|
| $G=\{V,E\}$ | 网络（一个有向图），包含点的集合和边的集合 |
| $N$ | 网络$G$中点的数量 |
| $M$ | 网络$G$中边的数量 |
| $S$ | 种子点集合 |
| $K$ | 种子点集合$S$包含点的数量 |
| $A$ | 失效点的集合 |
| $R$ | 失效点集合$A$包含点的数量 |
| $f(n)$ | 新颖性衰变函数 |
| $P_{uv}$ | 点$u$激活点$v$的概率 |
| $T_{uv}$ | 在边$(u,v)\in E$上期望的影响延迟时间 |
| $P_{con}$ | 影响传播路径$P$连通的概率 |
| $P_{blo}$ | 影响传播路径$P$阻断的概率 |
| $\sigma(S)$ | 期望被种子点集合$S$激活的点的数量 |
| $AP(s,v)$ | 点$v$被点集$s$激活的概率 |
| $PP(u,v)$ | 从$u$到$v$的影响力传播路径 |
| $PP_{ND}(v,S)$ | 从种子点集合$S$到点$v$的影响力传播路径集合 |
| $PP_{ND,\theta,C}(v,S)$ | $PP_{ND}(v,S)$中前$c$条短的且影响力不小于$\theta$的影响力传播路径集合 |
| $PP_{ND,\theta,c}(S)$ | 所有属于$PP_{ND}(v,S)$，且$u\in S_p$的路径组成的集合,其中$S_p$是所有可能被种子点集合$S$影响到的点组成的集合 |
| $\eta$ | 影响力丢失的阀值 |
| $L_{\Omega}(S,\lambda)$ | 联合空间$\Omega=(S,\lambda)$的惩罚函数 |

## 2.1 相关研究工作

Richardson 等人最早开始研究网络中影响力传播的最大化问题，他们用一个概率框架定义这个问题，并给出了一个采用马尔可夫随机场的解决方法[1-2]。接着，Kempe 等人把这个问题定义为一个离散优化问题并证明出它是一个 *NP-hard* 问题，该定义广泛地被后续的研究采纳[3]。Kempe 等人定义的影响力传播的最大化问题如下所示，

**定义** 2-1（影响力传播的最大化问题）：给定一个有向图$G=\{V,E\}$和一个正整数$K$，寻找一个种子点集$S\subseteq V$使得期望被种子点集$S$影响到的节点的数量最大，如

$$S=\arg\max_{S\subseteq V,|S|=K}\sigma(S) \tag{2-1}$$

其中，$\sigma(S)$是期望被种子点集合$S$激活的点的数量。为了求解定义 2-1 中的问题，首先需要对网络中影响力传播模型建模，然后才能在建立好的模型基础上设计有效的点集影响力的计算方法，最后还需要利用高效的选取影响力最大点集的策略在网络中找到影响力最大的点集。

可见，Kempe 等人定义的影响力传播的最大化问题包含以下三个子问题：网络中影响力传播模型建模、点集影响力的计算方法、选取影响力最大点集的策略。同时，影响力传播的相关因子的分析推动了大量该问题的分支问题的研究。

### 2.1.1 网络中影响力传播模型建模

Kempe 等人采用了两个基本的、且被广泛研究的传播模型：线性阈值模型和独立级联模型作为基本网络中影响力传播模型来研究影响力传播的最大化问题，还证明了这个两个模型的等价性[3]。

阀值模型最早被 Granovetter 等人提出[44]，其中的代表模型之一就是线性阈值模型。线性阈值模型中，节点只有激活和未激活两种状态，激活节点对其相邻的未激活节点有可线性叠加的影响力（总和不超过 1），一个未激活节点被激活的条件是受到来自其相邻激活节点的影响力总和超过一个阀值。该模型的影响力传播过程是：在第 *t* 步的所有未激活的节点被 *t-1* 步时已激活的所有节点尝试激活，当且仅当满足激活条件时，相应未激活节点被激活，直到第 *t* 步被新激活的节点数为 0，传播过程结束（不可能有新节点被激活为止）。该模型中，阀值代表的是未激活节点接受某种影响力的潜在倾向，这个取值与实际情况相关，在缺少这部分信息的情况下，通常给每个阀值一个[0,1]的随机取值（或者所有阀值等于 1/2）。

文献[13-14]基于线性阈值模型提出了解决影响力传播的最大化问题的改进方法，但近些年来，绝大部分影响力传播的最大化问题的研究和高效的解决方法集中在独立级联模型上[5-10]。

独立级联模型是基于相互作用的粒子系统的[45]，它被 Goldenberg 等人首先引入到营销相关的研究中[46-47]。这个模型中，节点只有激活和未激活两种状态，且节点只能由未激活状态转变成激活状态，不能反向转变，两个相邻节点间存在一个影响概率。该模型的影响力传播过程是：在第 *t* 步被激活的节点有且仅有一次机会以它与目标节点间的影响概率激活其相邻的未激活节点，直到第 *t* 步没有新的节点被激活，传播过程结束。

线性阈值模型以受影响者（未激活节点）为中心考虑影响力的传播过程，而独立级联模型以影响发起者（激活节点）为中心考虑影响力的传播过程，根据两种模型的等价性，研究者可以根据自己的实际需要灵活选用某个模型。

独立级联模型中，节点之间的拓扑关系和影响概率对影响力传播的最大化问题的研究非常重要，Gomez-Rodriguez 等人分析了节点间的拓扑关系与影响力传播的关系[15]，Goyal 等人提出了从历史数据中挖掘节点间相互影响概率的方法[16]，但是它们不属于影响力传播的最大化问题的研究重点，而属于社交网络另外的基础研究问题。此外，大量影响力传播的最大化问题的分支问题对以上两个基础模型做了相应的拓展，具体在后文中描述。

### 2.1.2 点集影响力的计算方法

点集影响力的计算问题被证明是*#P-hard*[6]。

目前，被广泛使用和认可的基础方法是蒙特卡洛模拟法（Monte Carlo）[3]，该方法采用随机数生成的方式来模拟种子点集影响力在网络中的传播过程。但由于蒙特卡洛模拟法的随机性，为了获得较为准确的模拟结果，往往需要大量的实验次数。因此，蒙特卡洛模拟法虽然比较准确，但是非常耗时，不适用于较大的网络。

为了弥补蒙特卡洛模拟法的不足，一系列的启发式算法被设计并提出。Chen 等人根据网络中节点度与节点影响力的关系，提出了度折扣法（Degree Discount）[5]。度折扣法的基本思想是直接选取网络中出度最大的 *K* 个点作为种子点集，这个方法的直观解释为，影响力最大的 *K* 个点可组成影响力较大的点集。度折扣法在速度上较蒙特卡洛模拟法有很大提高，但精度也较差，因为影响力最大的 *K* 个点可能存在很多相同的目标节点，这 *K* 个点组成的点集往往不是影响力最大的点集。接着，他们又提出了 PMIA 方法，PMIA 利用由最大影响路径组合成的最大影

响树（Maximum Influence Arborescence）估算点集的影响力，由于需要保存每个节点的最大影响树，PMIA 的空间复杂度比较高[6]。Jung 等人将影响力排序技术融入到点集影响力估算中，提出了基于影响力排序的影响力估算方法（IRIE） [9]。Liu 等人提出影响力传播路径(Influence Spreading Path)的概念，并在此基础上提出了快速计算点集影响力的 ISP 算法，实验结果表明该算法在速度和精度上都优于 PMIA，空间复杂度比 PMIA 低很多，具有更强的扩展性（可解决大规模网络中影响力传播的最大化问题)，是目前最好的算法之一[17]。依照影响力传播路径的思想，Kim 等人提出了独立路径算法（IPA）及其并行算法，IPA 和 ISP 算法基本思想一致，并行算法则提高了 IPA 的速度及其可扩展性[10]。

PMIA 和 ISP 算法是影响力传播的最大化问题中计算点集最常用的两大类方法，PMIA 从未激活节点层面考虑，先找到未激活节点的最大影响树，再计算未激活节点期望被激活的数量；ISP 从激活节点层面考虑，先找到从激活节点出发的影响力传播路径，再计算激活节点期望激活的未激活节点的数量。可见，ISP 算法更适用于普遍采用的独立级联模型。

### 2.1.3 选取影响力最大点集的策略

从网络中选取影响力最大的点集是一个离散优化问题，目前普遍使用的方法是 Kempe 等人提出的贪婪算法（Greedy Algorithm），该算法的基本思想是：每次迭代从网络中选择一个边际影响力最大的点放入点集（使点集的影响力增长最大的点），直到点集中点的数量达到 $K$ 为止（选出所有点）[3]。他们还证明了在影响力传播的最大化问题中贪婪算法能得到（1-1/e） 的近似解。为了提高贪婪算法的速度，Leskovec 等人利用点集影响力计算公式的单调（Monotonous）和子模（Submodular）的性质，提出了成本效益的惰性向前选择（CELF）优化[4]。成本效益的惰性向前选择优化的基本思想是，把前几轮循环中计算出的结果作为上界（理论上可证明），根据该上界在本轮循环中优先遍历影响力增量上界较大的点，并排除增量上界较小的点，这个优化可以减少每轮循环遍历的点数，提高了贪婪算法的速度，同时理论上保证了贪婪算法解的质量（不会丢失最优解）。在此基础上，Goyal 等人对成本效益的惰性向前选择优化进行了改进，提出了更高效的 CELF++优化[48]。

另一个选取影响力最大点集的策略是 Jiang 等人引入并改进的模拟退火算法（Simulated Annealing），该算法的基本思想是：首先随机生成一个点集（含 $K$ 个点），然后迭代地生成新的点集（从网络中随机选取一个其他点去随机替换上一步点集中某一点），刚开始以很大的概率接受影响力降低的点集，绝对接受影响力增

加的点集，并逐渐降低接受影响力降低的点集的概率，直到点集影响力达到稳定（当控制温度很低时点集影响力变化很小）为止。他们还证明了模拟退火算法在影响力传播的最大化问题中的收敛性[8]。模拟退火算法在迭代次数很多的情况下才能得到较优解，相比于包含 CELF 的贪婪算法时间复杂度高很多，但模拟退火算法每个迭代中都保存有一个完整解（*K* 个点的点集），对于处理对点集有限制条件的问题有很大优势。

同时，并行和分布式技术的成功引入进一步提高了从网络中选取影响力最大的点集方法的效率。为了更快地求解影响力传播的最大化问题，Wang 等人利用社区探测技术将大的社交网络划分为若干小的子网络[7]，在此基础上，Song 等人提出了更有效的分治方法（Divide-and-Conquer）[18]；Liu 等人采用 GPU 的并行处理技术加速了求解过程[19]；Liu[20]等人采用分布式技术提高了他们提出的 ISP 算法的效率。

### 2.1.4 分支问题

Kempe 等人定义的影响力传播的最大化问题及其影响力传播模型[3]比较理想化，没有考虑网络及实际应用中存在的重要影响因子。近年来，在影响力传播的最大化问题的深入研究中，大量重要影响因子被考虑并融入到问题中，产生了很多有价值的分支问题和相应的解决方法。

时间因子在影响力传播最大化问题中非常重要[17, 21-22]。一方面，在影响力传播模型的建模分析中需要考虑时间的影响，比如用户接受某个信息并转发该信息往往有一定的时间延迟；另一方面，影响力传播最大化问题的实际应用中往往会遇到时间限制，比如病毒式营销中，考虑到产品的更新换代及市场竞争问题，商家都希望新产品能更快地占领市场。Chen 等人[18]和 Liu 等人[17]分别对时间限制下影响力传播的最大化问题进行了研究，他们都是在独立级联模型上加入时间因子，建模分析了时间限制下影响力传播过程。

网络是一个极其复杂的环境，传播的影响力中既有合作的，也有竞争的；既有积极的，也有消极的。因此，影响力属性作为一个重要因子被考虑到影响力传播的最大化问题的研究中[23-25]。Bharathi 等人建模分析了多种影响力竞争情况下影响力的传播过程，并采用博弈的思想提出了让己方影响力传播最大化的策略[20]。接着，Borodin 等人在线性阈值模型下分析了影响力竞争传播过程，并从理论上给出了可得到的近似解的上限[24]。Chen 等人在独立级联模型上建模分析了考虑消极影响力的影响力传播的最大化问题，并提出了结合 PMIA 的启发式算法[25]。

其他影响因子方面，Li 等人考虑不同节点间关系（友好或者敌对），分析了影

响力传播的最大化问题[26]。Nguyen 等人定义了预算限制条件下影响力传播的最大化问题，还借鉴背包问题的解决方法，提出了高效的求解算法[27]。Lu 等人将客户受到影响和接受影响分为两个过程，提出了效益最大化的影响力传播问题[28]。Li[29]等人定义了地理位置限制下影响力传播的最大化问题，并提出了两个贪婪算法解决该问题。

## 2.2 影响力传播路径方法

正如前文所述，影响力传播路径方法（ISP）[17]是目前最好的快速计算点集影响力的方法之一。该方法提出了一种特殊的结构——影响力传播路径来表示节点间的影响关系。一条影响力传播路径 $PP(u,v)$ 是从一个种子点 $u$（$u \in S$）到非种子节点 $v$（$v \in V-S$）且不包含重复点的简单路径。可见，一条影响力传播路径实际上是种子点 $u$ 激活非种子节点 $v$ 的一种可能的方式，那么，当且仅当这条影响力传播路径上所有点被激活，种子点 $u$ 通过这条影响力传播路径激活了非种子节点 $v$，则，这条影响力传播路径对非种子节点 $v$ 的影响力（概率）$P_{PP(u,v)}$ 是这条路径上所有边上的影响力（概率）的累乘之积，如 $P_{PP(u,v)}=\prod_{(u',v')\in PP(u,v)} P_{u'v'}$，其中 $P_{u'v'}$ 表示这条路径中某条边上的影响力。接着，在假设所有以节点 $v$ 为终点的影响力传播路径相互独立情况下，种子点集 $S$ 对非种子节点 $v$ 期望影响力（激活概率）为 $AP(S,v)=1-\prod_{u\in S}(1-P_{PP(u,v)})$，其中 $1-P_{PP(u,v)}$ 是某条影响力传播路径影响非种子节点 $v$ 失败的概率。因此，种子点集 $S$ 总的期望影响力（激活的节点数）为 $S$ 对所有非种子点期望影响力的累加之和，即 $\sigma(S)=\sum_{v\in V-S} AP(S,v)$。

## 2.3 限制性模拟退火算法

限制性模拟退火算法（Constrained Simulated Annealing-CSA）扩展了普通的模拟退火算法，用以解决限制性非线性优化问题[31]。普通的非线性优化问题的解空间只有可行解空间，所以模拟退火算法只需要搜索可行解空间就能获得最优解，它不需要考虑不可行解存在的情况。然而在限制性非线性优化问题中，解空间不仅包含了可行解空间，还包含了不可行解空间，则最优解可能需要算法搜索部分不可行解空间才能找到。为此，限制性模拟退火算法在普通的模拟退火算法的基础上提供了一个能在可行解空间和不可行解空间穿插搜索的策略。该搜索策略以一个惩罚函数作为基础，这个惩罚函数融合了目标函数和带有惩罚系数的限制条件函数，使得限制性模拟退火算法能够在惩罚系数较小的时候访问部分不可行解空间，跳过极值点，最终找到最优解。

## 2.4 本章小结

本章首先从影响力传播的最大化问题的各个子问题和分支问题出发，详细总结了其相关研究工作，然后介绍了目前最好的快速计算点集影响力的方法之一的影响力传播路径方法（ISP），还简单描述了能解决限制性非线性优化的限制性模拟退火算法。

# 第三章 新颖性衰变下影响力传播的最大化问题

本章将讨论新颖性衰变下影响力传播的最大化问题。本章首先结合近期研究成果和数据分析调查新颖性衰变在影响力传播过程中的影响；然后，建立新颖性衰变下影响力传播模型并在此基础上标准化定义新颖性衰变下影响力传播的最大化问题，该问题被证明是 *NP-hard*；接着，基于定义问题的性质提出选出影响力最大点集的贪婪算法及计算点集影响力的方法；最后，通过 4 个真实数据集中的实验结果，验证了提出的方法的高效性。

## 3.1 新颖性衰变在影响力传播过程中的影响

### 3.1.1 影响力新颖性衰变的近期相关研究工作

已被研究的影响力传播的最大化问题都假设节点（用户）间的影响力（概率）是稳定的[1-10]，但现实生活中，影响力（如信息、想法、观点等）的新颖性会随着它重复出现次数的增加而产生衰变，从用户角度来说，就是用户往往不容易接受第一印象就不好的事物，更不喜欢成为重复信息的传播者。

一些研究[32-33]已经发现重复推送同一个影响力给某个用户，会导致该影响力对该用户的影响效果产生衰变。例如，在推特网（Twitter）上，一个用户第一次看到某个新信息并转发该信息的概率大于该用户第一次见到并忽略该信息而在以后看到并转发该信息的概率，而且随着该用户看到并忽略该信息次数的增加，该用户转发该信息的概率不断降低。这种现象被我们称为新颖性衰变。直观来说，信息的新颖性随着它出现的次数不断下降。

新颖性衰变现象的调查中，Steeg 等人发现在社交网络中多次推送同一个故事给某个用户，该用户投票给（喜欢）该故事的概率只有少量的增长，这意味着人们不喜欢接受重复的信息。他们还发现重复信息的真实影响力远小于独立级联模型中计算出的结果[33]；Wu 等人发现用户对新颖事物的关注度在重复传播中逐渐减少且该趋势近似于一个延伸型指数函数[34]；Leskovec 等人发现在推荐网络中，用户购买某个产品的概率随着该产品被推荐次数的增加而增长，但是很快达到一个极值[32]。然而，这些工作并没有用函数形式具体分析新颖性衰变对影响力传播过程的影响，更没有考虑新颖性衰变下影响力传播的最大化问题。

为了直观地说明在影响力传播的最大化问题中新颖性衰变现象，举例如下：在图 3-1 中，4 个用户被 4 条有向边连接起来，每条边表明了用户间相互影响的关

系，且包含了两个重要参数，一个是影响（激活）概率 $P$，另一个是期望影响延迟时间 $T$，例如，用户 $V_1$ 在 2 个时间间隔后影响 $V_3$ 的概率是 0.7。假设种子点集 $S=\{V_1,V_2\}$，在不考虑新颖性衰变的情况下，$V_3$ 被 $S$ 激活的概率是 $0.1+(1-0.1)\times0.7$，其中 0.1 是 $V_3$ 被 $V_1$ 激活的概率，$(1-0.1)\times0.7$ 是 $V_3$ 被 $V_2$ 激活的概率。然而，如果考虑到新颖性衰变，$V_3$ 被 $V_1$ 激活失败后，$V_3$ 被 $V_2$ 激活的概率将会变小（小于 $(1-0.1)\times0.7$）。那么，新颖性衰变在影响力传播的最大化问题中是一个重要影响因子。

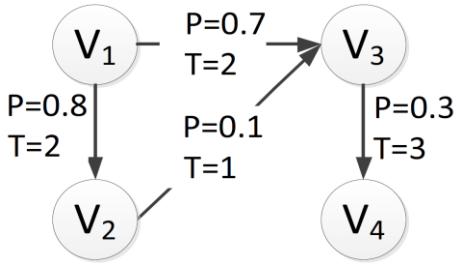


图3-1 一个包含影响概率 $P$ 和影响延迟时间 $T$ 的社交网络

### 3.1.2 新颖性衰变函数

基于前文对新颖性衰变的描述，本节结合两个真实数据集形式化分析影响力传播模型中新颖性衰变函数。



为了简化描述，假设一个用户的 $n$ 个朋友（相邻节点）被某个信息影响（激活）了，则该用户接收到该信息 $n$ 次。令 $TP_n$ 表示当某用户 $n$ 个朋友被影响后，该用户也被影响的概率，$p_n$ 表示该用户被他第 $n$ 个朋友信息影响的概率。$TP_n$ 与 $TP_{n-1}$ 的关系如公式(3-1)所示：

$$TP_n=TP_{n-1}+(1-TP_{n-1})\times p_n \tag{3-1}$$

接着，我们用公式(3-2)计算 $p_n$，

$$p_n=(TP_n-TP_{n-1})/(1-TP_{n-1}) \tag{3-2}$$

为了形式化表示新颖性衰变函数 $f(n)$，我们将新颖性因子从 $p_n$ 中分离出来，即 $f(n)=p_n/p_1$，这里 $p_1=TP_1$ 是某用户第一次接收到信息并被影响的平均概率。为进一步探索作为新颖性衰变函数普通表达形式的指数型函数 $f(n)=\gamma^{n-1}$[34]，我们采用最小二乘法在 Digg 和 Flickr 两个数据集中估算参数 $\gamma$。

（1）Digg 数据集

Digg 数据集包含关于 2009 年 6 月在 Digg（www.digg.com）首页推广的故事

的信息[35]，这个数据集包含 279,634 个节点和 1,731,658 条边。如果用户 $v$ 将用户 $u$ 列为他的朋友，$v$ 可以看到 $u$ 的活动。这个数据集还列出了记录用户在某个时候投票给某个故事的 Digg-votes，共有 139,409 个不同用户在 3,553 个不同流行故事上的 3,018,197 个投票。

如图 3-2 所示, 用户投票给某个故事的概率 $TP_n$ 在他足够数量（>25）朋友投票给该故事后达到稳定点。该结果说明前文中讨论的新颖性衰变的存在性，否则，$TP_n$ 应该随着 $n$ 的增大不断增长并最终达到 1.0。

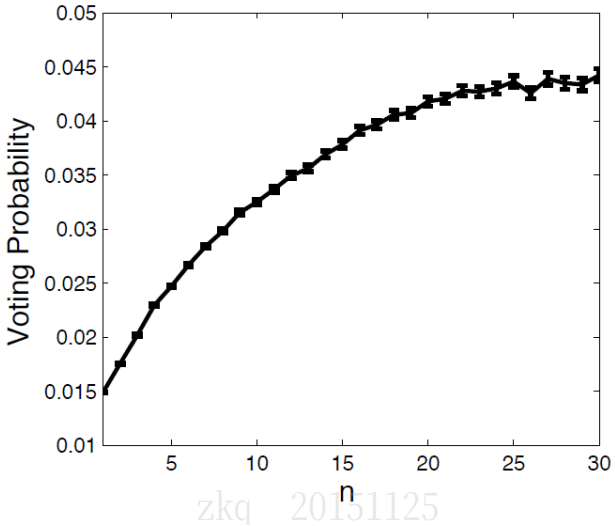


图3–2 Digg中用户在 $n$ 个朋友投票后投票的概率

利用公式(3-2)，我们得到 $p_n$，并求出新颖性衰变函数 $f(n)$ 随 $n$ 变化情况和最好的拟合函数如图 3-3 所示。可见 $f(n)$ 很符合指数型函数形式，其中最好的拟合函数是 $f(n)=0.2969^{n-1}$，该拟合函数达到最小的方差（SSE=0.1941）。

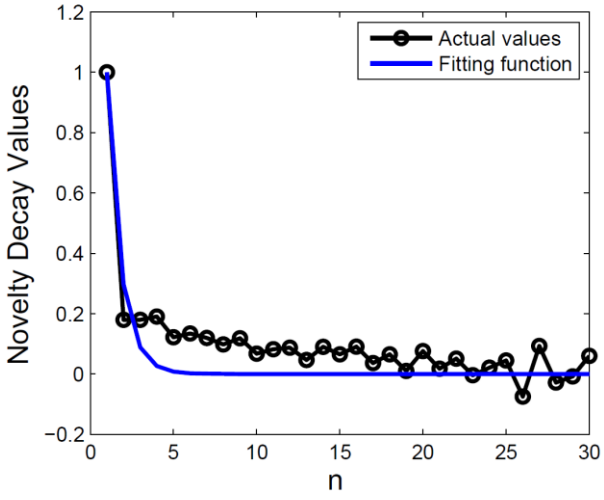


图3-3 Digg中新颖性衰变情况及它的拟合函数

（2）Flickr 数据集

Flickr 数据集包含一个大型的朋友关系网络和一系列来自于 Flickr（www.flickr.com）的流行图片的标记记录[36]。如果用户 $v$ 将用户 $u$ 列为他的朋友，$v$ 可以看到 $u$ 标记喜爱图片的行为。为了调查影响力在 Flickr 网络中传播的普遍情况，我们只考虑活跃用户（至少有 5 个标记的用户）和流行图片（至少被 100 个用户标记的图片），最终得到 222,038 个活跃用户，14,727,117 条用户间连接及 3,125 张流行图片。

如图 3-4 所示，用户标记喜欢图片的概率 $TP_n$ 随着 $n$ 的增大不断增长并逐渐达到稳定状态（$n$>23）。该结果再次说明前文中讨论的新颖性衰变的存在性。

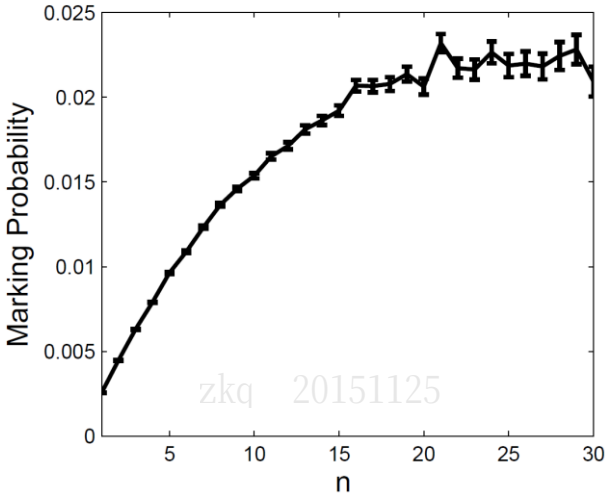


图3-4 Flickr中用户在 $n$ 个朋友标记后标记的概率

新颖性衰变函数 $f(n)$ 随 $n$ 变化情况及其最好的拟合函数如图3-5所示。$f(n)$ 依然很符合指数型函数形式，其中最好的拟合函数是 $f(n)=0.8918^{n-1}$，该拟合函数达到最小的方差（SSE=2.7570）。

## 3.2 新颖性衰变下影响力传播的最大化问题定义及其性质

本节在考虑时间延迟的独立级联模型的基础上进行扩展，使得新的模型融入了新颖性衰变因子，在此模型基础上，定义了新颖性衰变下影响力传播的最大化问题，还分析了该问题的相关性质。

### 3.2.1 新颖性衰变下独立级联模型

在考虑时间延迟的独立级联模型中，每个节点有两个状态：激活和未激活。影响力传播过程中，节点可以由未激活状态转换为激活状态，但不能从激活状态

转换为未激活状态。如图 3-1 所示，每条边上面关联了两个参数，一个是两点间的影响（激活）概率 $P_{uv}$，另一个是两点间期望影响延迟时间 $T_{uv}$。接下来，我们在考虑时间延迟的独立级联模型的基础上扩展出新颖性衰变下独立级联模型。

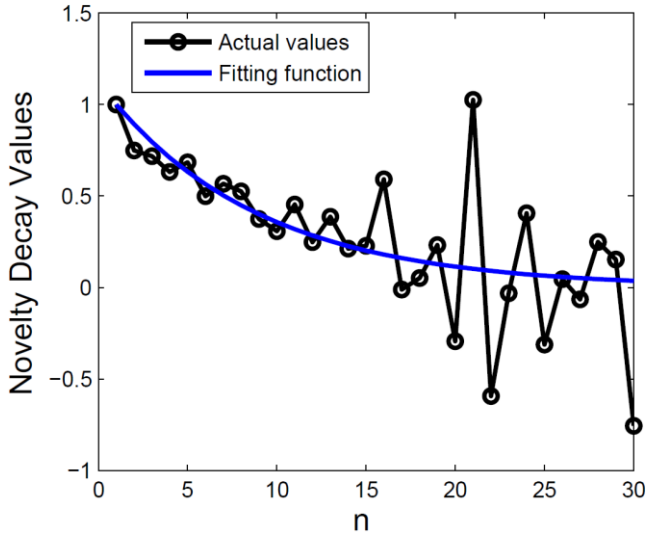


图3-5 Flickr中新颖性衰变情况及它的拟合函数

给定一个有向图 $G=\{V,E\}$，一个种子点集 $S\subseteq V$ 和一个新颖性衰变函数 $f(n)$，新颖性衰变下独立级联模型影响力传播过程如下所述。令 $A_t$ 为 $t\geq 0$ 时候被激活的点集，且 $A_0=S$。每个节点 $u\in A_t$ 有且仅有一次机会在时间 $t+T_{uv}$ 去尝试激活它相邻的未激活节点一次，且节点 $u$ 激活节点 $v$ 的概率为 $P_{uv}\times f(n)$，其中 $n$ 是 $v$ 已经被尝试激活的次数。当且仅当不存在已激活节点去激活其相邻未激活节点情况，影响力传播过程结束。因此，在影响力传播过程结束后，所有已激活节点总数就是种子点集的影响力 $\sigma(S)=\sum_{t=0}^{\infty}A_t$。

### 3.2.2 问题定义及其性质

基于新颖性衰变下独立级联模型，我们给出新颖性衰变下影响力传播的最大化问题的标准化定义。

**定义** 3-1（新颖性衰变下影响力传播的最大化问题）：给定一个有向图 $G=\{V,E\}$，一个新颖性衰变函数 $f(n)$ 和一个正整数 $K$，寻找一个种子点集 $S\subseteq V$ 使得在新颖性衰变下独立级联模中期望被种子点集 $S$ 影响到的节点的数量最大，如

$$S=\arg\max_{s\subseteq v,|S|\leq K}\{\sigma(S)\mid f(n)\} \tag{3-3}$$

**推论** 3-1：新颖性衰变下影响力传播的最大化问题是 *NP-hard*。

**证明**：给定一个普通的影响力传播的最大化问题的实例 $\varphi: G=\{V,E\}$ 和 $K$，我们可以构造一个新颖性衰变下影响力传播的最大化问题的实例 $\omega$，即通过设置新颖性衰变函数 $f(n)=1$。那么，$S$ 是 $\omega$ 的解当且仅当 $S$ 是 $\varphi$ 的解。因为普通的影响力传播的最大化问题已经被证明是 *NP-hard*[3]，所以新颖性衰变下影响力传播的最大化问题也是 *NP-hard*。■

但是，不同于普通的影响力传播的最大化问题，新颖性衰变下影响力传播的最大化问题的点集影响力计算公式 $\sigma(S)$ 是非单调（non-monotone）和非子模的（non-submdular）。为了方便证明，我们以图 3-1 的特殊网络为例并取 $f(n)=0.3^{n-1}$。

**推论** 3-2：新颖性衰变下影响力传播的最大化问题的点集影响力计算公式 $\sigma(S)$ 是非单调和非子模的。

**证明**：

1）非单调性：假设 $S_1=\{V_1\}$，$S_2=\{V_1,V_2\}$ 和 $S_3=\{V_1,V_2,V_3\}$，那么 $\sigma(S_1)=2.7204$，$\sigma(S_2)=2.3757$ 和 $\sigma(S_3)=3.3$。因为 $\sigma(S_3)>\sigma(S_1)>\sigma(S_2)$，所以 $\sigma(S)$ 是非单调的。

2）非子模性：假设 $S_1=\{V_1\}$，$S_2=\{V_1,V_2\}$，那么 $\sigma(S_1\cup\{V_3\})-\sigma(S_1)=0.3796$，和 $\sigma(S_2\cup\{V_3\})-\sigma(S_2)=0.9243$。因为 $S_1\subseteq S_2$ 和 $\sigma(S_1\cup\{V_3\})-\sigma(S_1)<\sigma(S_2\cup\{V_3\})-\sigma(S_2)$，所以 $\sigma(S)$ 是非子模的。■

## 3.3 贪婪算法及其优化

由于新颖性衰变下影响力传播的最大化问题的点集影响力计算公式是非单调和非子模的，传统的贪婪算法（Greedy Algorithm）[3]不能适用。我们参照用于解决社交网络中效益最大化问题（其点集影响力计算公式是非单调和子模的）的 U-Greedy 算法[28]提出了限制性贪婪算法（Restricted Greedy Algorithm），并在该算法基础上给出了一个优化策略。

### 3.3.1 限制性贪婪算法

限制性贪婪算法的伪代码见表 3-1，该算法初始化一个空的种子点集 $S$ 以及存储首轮点集的 $S_0$（空集）。在每次循环中选出带来点集影响力增量最大的候选点并存入该轮的种子点集中（lines 3-4），由于在新颖性衰变下影响力传播的最大化问题要求寻找的是节点数不多于 $K$ 的影响力最大的点集（$|S|\leq K$），算法最后选择并返回的是影响力最大的点集（lines 6-7）。尽管理论上算法可能返回点数量小于 $K$ 的点集，在实际应用中这种情况极少发生，主要原因是实际应用中相对于网络

中极大的点的数量，$K$ 的值往往很小。

表3-1 限制性贪婪算法的伪代码

**Input:** $G=\{V,E\}$, $T_{uv}$, $P_{uv}$, $f(n)$, $K$
**Output:** $S$
1. $S \leftarrow \Phi, S_0 \leftarrow \Phi$;
2. **for** $k \leftarrow 1$ **to** $K$ **do**
3. $u \leftarrow \arg\max_u(\sigma(S \cup \{u\}) - \sigma(S))$;
4. $S_k \leftarrow S_{k-1} \cup \{u\}$;
5. **end for**
6. $S \leftarrow$ the $S_k$ with maximum $\sigma(S_k)$ from $k=1$ to $k=K$;
7. **return** $S$;

## 3.3.2 动态剪枝优化

在限制性贪婪算法中（表 3-1 所示），$S_k$ 表示第 $k$ 轮中选择的点集，该算法在每轮都需要遍历所有候选点（$u \in V - S_{k-1}$）并计算点集影响力的增量，导致时间消耗非常大，而很多影响力很小的候选点是可以不用计算其影响力的增量而直接排除的。为了提高限制性贪婪算法的速度，我们根据新颖性衰变下影响力传播的最大化问题的点集影响力计算公式的特殊性质，提出了一个动态剪枝优化。

带有动态剪枝优化的限制性贪婪算法的伪代码见表 3-2。初始化操作中，一个空的种子点集 $S$ 以及存储首轮点集的 $S_0$（空集）被设置，被创建的优先级队列 $Q_k$ 用于存储第 $k$ 轮中遍历过的点及其加入到前一轮选出的点集 $S_k$ 得到的影响力总值（$Inf_k^u = \sigma(S_{k-1} \cup \{u\})$，lines 1-2）。在每轮中，该算法借助优先级队列以点集影响力值递减的方式访问候选点，一旦访问到的点的影响力值小于本轮已找到的最大影响力增量 max*MarInf*，停止本轮的访问（line 5）。

另一个有效的优化技术是记录 $u$ 在上一轮计算中得到的一个影响力增量的上界（line 6，$(Inf_{k-1}^u + \sigma(\{s_{k-1}\})) - \sigma(S_{k-1})$，相关证明在推论 3-3 中），如果这个影响力增量的上界也小于 max*MarInf*，该点也可以被忽视（line 7），否则，该点对于上一轮点集的影响力增量会被计算，并存储在 $Q_k$ 中（line 10），如果其影响力增量大于 max*MarInf*，max*MarInf* 会被更新（lines 11-14）。最后，我们获得本轮的种子点集和它的影响力值（lines 17-18）。

表3-2　带有动态剪枝优化的限制性贪婪算法的伪代码

**Input:** $G=\{V,E\},T_{uv},P_{uv},f(n),K$

**Output:** $S$

1. $S\leftarrow\Phi,S_0\leftarrow\Phi,\sigma(S_0)\leftarrow 0,s_0\leftarrow NULL;$
2. For every $v\in V$, calculate $\sigma(\{v\})$ and insert $(v,\sigma(\{v\}))$ into $Q_0;$
3. **for** $k\leftarrow 1$ to $K$ **do**
4. max*MarInf* $\leftarrow-\infty;$
5. **for** node $u\in V-S_{k-1}$, $\sigma(\{u\})\geq$ max*MarInf* **do**
6. **if** $u\in Q_{k-1}$ and $(Inf_{k-1}^{u}+\sigma(\{s_{k-1}\}))-\sigma(S_{k-1})<$max*MarInf*
7. continue;
8. **end if**
9. **else**
10. Calculate $Inf_k^u$ and insert $(u,Inf_k^u)$ into $Q_k;$
11. **if** $Inf_k^u-\sigma(S_{k-1})>$max*MarInf*
12. max*MarInf*$\leftarrow Inf_k^u-\sigma(S_{k-1});$
13. $s_k\leftarrow u;$
14. **end if**
15. **end else**
16. **end for**
17. $S_k\leftarrow S_{k-1}\cup\{s_k\};$
18. $\sigma(S_k)\leftarrow\sigma(S_{k-1})+$max*MarInf*;
19. **end for**
20. $S\leftarrow$the $S_k$ with maximum $\sigma(S_k)$ from $k=1$ to $k=K;$
21. **return** $S;$

由于动态剪枝优化只是剪掉了影响力小于 max*MarInf* 的节点，依然能保证限制性贪婪算法获得最优解，详细证明如下：

**推论** 3-3：动态剪枝优化能保证限制性贪婪算法获得最优解.

**证明**：新颖性衰变下影响力传播的最大化问题中点集影响力计算公式满足 $\sigma(S_1\cup S_2)\leq\sigma(S_1)+\sigma(S_2)$，$S_1,S_2\subseteq V$。对于节点$u$，它在上一轮点集的影响力增量 $MarInf^{k}(u)=\sigma(S_{k-1}\cup\{u\})-\sigma(S_{k-1})\leq\sigma(S_{k-1})+\sigma(u)-\sigma(S_{k-1})=\sigma(u)$，则，$\sigma(u)$ 为该节点影响力增量的一个上界。如果一个节点在上一轮中被访问 $MarInf^{k}(u)=\sigma(S_{k-1}\cup\{u\})-\sigma(S_{k-1})=\sigma(S_{k-2}\cup\{s_{k-1}\}\cup\{u\})-\sigma(S_{k-1})=\sigma((S_{k-2}\cup\{u\})$

$\cup\{s_{k-1}\})-\sigma(S_{k-1})\le Inf^{u}_{k-1}+\sigma(\{s_{k-1}\})-\sigma(S_{k-1})$，则，$Inf^{u}_{k-1}+\sigma(\{s_{k-1}\})-\sigma(S_{k-1})$ 也为该节点影响力增量的一个上界。那么，如果一个节点影响力增量的上界 $\sigma(u)$ 或者 $Inf^{u}_{k-1}+\sigma(\{s_{k-1}\})-\sigma(S_{k-1})$ 小于 max*MarInf*，该节点比当前最优点差，它不是最优解，动态剪枝优化剪掉这个节点不会丢失最优解。■

由于限制性贪婪算法需要在每一轮中遍历所有的点，其时间复杂度为 $O(KNT(\sigma(S)))$，其中 $T(\sigma(S))$ 是求解点集影响力 $\sigma(S)$ 的时间。同时，在实际应用中动态剪枝优化使得限制性贪婪算法每轮访问的点数远小于 $N$ 。

## 3.4 计算点集影响力的方法

求解新颖性衰变下影响力传播的最大化问题的另一个关键技术是计算点集影响力的方法。本文提出了一个基于模拟的方法和一个更有效的基于影响力传播路径的方法。

### 3.4.1 基于模拟的方法

蒙特卡洛模拟方法（Monte Carlo）已经被广泛的用作影响力传播的最大化问题的基准方法[3]，但是在新颖性衰变下影响力传播模型中，节点间相互影响的顺序必须被考虑，传统的蒙特卡洛模拟方法已经不适用，本文对其做了改良，得到了适用于新颖性衰变下影响力传播模型的基于模拟的方法。

基于模拟的方法的伪代码见表 3-3，该算法初始化了一个存储已激活节点的点集 $S_a$，一个记录节点 $u$ 已经被尝试激活失败的次数的数组 $NE(u)$，以及一个存储潜在激活新节点机会的最小优先级队列 $EH$，该队列的每个元素包含了尝试激活发生的时间延迟 $T$，激活的概率 $P$，目标节点 $u$，且以 $T$ 作为优先级排序的关键词。算法开始时，种子点集 $S$ 是激活状态，其他节点都是未激活状态，$S$ 包含的所有潜在激活新节点机会被存储到 $EH$ 中（lines 1-3）。当 $EH$ 非空时，该算法从 $EH$ 中取出最近可能激活新节点的机会，如果目标节点是非激活状态，则计算目标节点被激活的概率 $P\leftarrow E.P\times f(NE(u))$，其中 $f(NE(u))$ 是影响力衰变函数值，然后根据计算出的概率尝试激活目标节点（lines 6-8），如果目标节点被成功激活，则更新 $S_a$，并将激活节点包含的激活新节点机会存储到 $EH$ 中；否则，仅更新 $NE(u)$（lines 9-16）。直到 $EH$ 为空，即不存在潜在激活新节点机会时，该算法终止并返回激活节点的总数（line 18）。

在一次运行中，该算法最坏情况下需要遍历所有的点和边，所以它执行一次需要消耗时间为 $O(N+M)$。由于该算法的不确定性（模拟导致），需要进行多次模拟并求得平均值[3]，令模拟次数为 $N_{MC}$，则基于模拟的方法的时间复杂度为

$O(N_{MC}(N+M))$。

表3-3　基于模拟的计算点集影响力方法的伪代码

**Input:** $G=\{V,E\},T_{uv},P_{uv},f(n),S$
**Output:** $\sigma(S)$
1. Initialize the actived set $S_a \leftarrow S$;
2. Initialize an array $NE(u) \leftarrow 0$ for $u \in V-S$;
3. Initialize a min priority queue $EH$ and enqueue $E=(0,1,u)$ for $u \in S$;
4. **while** $EH \neq \Phi$ **do**
5. $E=$dequeue $(T,P,u)$ from $EH$;
6. **if** $u \notin S_a$
7. $P \leftarrow E.P \times f(NE(u))$;
8. Draw $flag$ from $Bernoulli(P)$;
9. **if** $flag=1$
10. $S_a \leftarrow S_a \cup \{u\}$;
11. Enqueue $E=(T_{uv}+E.T,P_{uv},v)$ into $EH$ for every $(u,v) \in E$;
12. **end if**
13. **else**
14. $NE(u) \leftarrow NE(u)+1$;
15. **end else**
16. **end if**
17. **end while**
18. **return** $|S_a|$;

## 3.4.2 基于影响力传播路径的方法

上文提出的基于模拟的方法比较准确，但非常耗时，且不适用于较大规模的网络。为此，我们参考影响力传播路径方法（ISP），提出了适用于新颖性衰变下影响力传播模型的基于影响力传播路径的方法来更快地估算点集的影响力。

给定一个种子点集 $S \subseteq V$，种子点集的影响力为 $\sigma(S)=\sum_{u \in V} AP_S(u)$，其中 $AP_S(u)$ 是 $u$ 被 $S$ 激活的概率。为了快速有效地估算 $AP_S(u)$，本文定义一个新颖性衰变下影响力传播路径（$PP_{ND}$）如下：

**定义** 3-2（新颖性衰变下影响力传播路径）：给定一个有向图$G=\{V,E\}$和一个种子点集$S\subseteq V$，一条在$G$中的路径 $h=(u_1\xrightarrow{e_1}u_2\xrightarrow{e_2}u_3\cdots\xrightarrow{e_{k-1}}u_k)$是新颖性衰变下影响力传播路径当且仅当$u_1\in S$且$u_k\notin S, k>1$。

因为一个节点不能被影响（激活）超过一次，一条新颖性衰变下影响力传播路径不能包含重复的节点。一条新颖性衰变下影响力传播路径长度是$Len(h)=\sum_{i=1}^{i=k-1}T_{e_i}$，其准确的影响力（激活目标节点的概率）是$\prod_{i=1}^{i=k-1}P(e_i)\times E(\tau^h(u_{i+1}))$，其中$E(\tau^h(u_{i+1}))$是该路径在其目标节点$u_{i+1}$的影响力衰变函数的期望值。

接着，我们讨论求解$E(\tau^h(u))$的方法。一条新颖性衰变下影响力传播路径$h$有两种状态：连通和阻断。路径$h$是连通的当且仅当$h$成功激活了除目标节点外其他节点，否则，$h$是阻断的。当$h$是连通时，它才有机会去激活其目标节点。路径$h$连通的概率为$P_{con}=\prod_{i=1}^{i=k-2}P(e_i)\times\tau^h(u_{i+1})$，则该路径阻断的概率为$P_{blo}=1-P_{con}$。如果该路径是第$c$条去尝试激活其目标节点$u_{i+1}$的路径，该路径的排序为$c$，即新颖性衰变函数 $f(n)$中$n$的取值。

假设$h_c$是以$u$作为目标节点的第$c$短的新颖性衰变下影响力传播路径，即有$c-1$条以$u$作为目标节点新颖性衰变下影响力传播路径比$h_c$更短。路径越短，其优先尝试激活目标节点的概率就越大，我们需要考虑更短的$c-1$条路径的所有状态（激活和未激活）去计算$E(\tau^{h_c}(u))$。例如，令$h_1$和$h_2$是第一和第二短的以$u$作为目标节点新颖性衰变下影响力传播路径，计算$E(\tau^{h_3}(u))$需要考虑 4 种情况：$E(\tau^{h_3}(u))=P_{blo}(h_1)\times P_{blo}(h_2)\times f(1)+P_{blo}(h_1)\times P_{con}(h_2)\times f(2)+P_{con}(h_1)\times P_{blo}(h_2)\times f(2)+P_{con}(h_1)\times P_{con}(h_2)\times f(3)$。

然后，我们说明寻找新颖性衰变下影响力传播路径的方法，对于给定的种子点集$S$，我们用$PP_{ND}(u,S)$表示从$S$出发到节点$u$的所有新颖性衰变下影响力传播路径的集合。由于路径的数量$|PP_{ND}(u,S)|$随着种子点集节点的数量成指数增长，寻找$PP_{ND}(u,S)$是非常耗时的，为此，本文给出了两个限制条件去排除那些影响力极小的路径。首先，我们剪掉那些影响力（对于目标节点激活概率）小于一个特定阀值$\theta>0$的路径，因为这些路径对于点集影响力的估算影响极小。其次，我们只保留$PP_{ND}(u,S)$中前$C$条短的路径，这是由于考虑新颖性衰变的情况下，排序靠后的路径对目标节点的影响力极小，可以忽略不计。满足以上两个限制条件并属于$PP_{ND}(u,S)$的路径集合表示为$PP_{ND,\theta,C}(u,S)$。

粗略来看，寻找$PP_{ND,\theta,C}(u,S)$类似于$K$条最短路径找寻问题（$K$ shortest path routing）。但是，寻找$PP_{ND,\theta,C}(u,S)$旨在寻找从多源到多目标的前$C$条短的带有$\theta$限制条件的路径，这意味着最先进的适用于$K$条最短路径找寻问题的算法[37-38]都不

适合用于寻找 $PP_{ND,\theta,C}(u,S)$，因为这些算法都是考虑单源问题，并且 $\theta$ 限制也不容易加入到这些算法中。为了弥补这个缺口，本文提出了改良的迪杰斯特拉算法（AD）来寻找 $PP_{ND,\theta,C}(u,S)$。如迪杰斯特拉算法，AD 也在每轮中采用贪婪寻找策略选取最短的路径去拓展，但 AD 只拓展满足 $\theta$ 限制条件的路径。本文进一步将 $PP_{ND,\theta,C}(u,S)$ 中路径的影响力的计算过程融合到寻找 $PP_{ND,\theta,C}(u,S)$ 中，以提高算法的效率。

表3-4 改良的迪杰斯特拉算法（AD）的伪代码

**Input:** $G=\{V,E\}, T_{uv}, P_{uv}, S, f(n), \theta, C$

**Output:** $PP_{ND,\theta,C}(u,S)$

1. $PP_{ND,\theta,C}(u,S) \leftarrow \Phi$;
2. Initialize two arrays $Count(u) \leftarrow 0$ and $PH_{con}(u) \leftarrow 0$ for $u \in V$;
3. Initialize a min priority queue $PH$ for paths metting the $\theta$ constraint, and enqueue $h=(0,1,path=\{u\})$ for $u \in V$;
4. **while** $PH \neq \Phi$ **do**
5. $h \leftarrow$ dequeue $(T, P_{con}, path)$ from $PH$;
6. Compute $E(\tau^h(u))$ according to $PH_{con}(u)$;
7. $P^h(u) \leftarrow P_{con} \times P_{wu} \times E(\tau^h(u))$;
8. **if** $Count(u) < C$ and $P^h(u) > \theta$ and $h$ is loopless
9. Insert $P^h(u)$ into $PP_{ND,\theta,C}(u,S)$;
10. $Count(u) \leftarrow Count(u)+1$;
11. Insert $P_{con}$ into $PH_{con}(u)$;
12. $P_{con} \leftarrow P^h(u)$;
13. Enqueue $h=(T+T_{uv}, P_{con}, path \cup \{v\})$ into $PH$ for $u \in V$;
14. **end if**
15. **end while**
16. **return** $PP_{ND,\theta,C}(u,S)$;

改良的迪杰斯特拉算法（AD）的伪代码如表 3-4 所示，AD 开始于初始化 $PP_{ND,\theta,C}(u,S)$，记录 $PP_{ND,\theta,C}(u,S)$ 中路径数量的 $Count(u)$，及记录 $PP_{ND,\theta,C}(u,S)$ 中路径连通概率的 $PH_{con}(u)$（line 1-2）。为了实现贪婪搜索策略，AD 初始化了一个最小优先级队列 $PH$ 用于存储候选的路径信息，$PH$ 的第一个元素是用于排序的该路径长度（延迟时间）$T$，第二个元素是该路径连通概率 $P_{con}$，第三个元素是路径

结构信息（经过的点） *path* （line 3）。在每一轮中，AD 从 *PH* 中选择长度最短的路径去拓展（line 5），且求解新颖性衰变下影响力传播路径 $h$ 激活其目标节点 $u$ 的概率为 $P^h(u)=P_{con}\times P_{wu}\times E(\tau^h(u))$（line 6-7）。如果 $h$ 满足以上提出的两个限制条件，AD 把它插入到 $PP_{ND,\theta,C}(u,S)$ 中，更新 $Count(u)$ 和 $PH_{con}(u)$，并获得一条新的候选路径存入 *PH* （line 8-14）。注意，最终目的是计算种子点集 $S$ 影响力 $\sigma(S)$，所以 AD 不需要存储具体的路径信息，只需要存储寻找 $PP_{ND,\theta,C}(u,S)$ 中获得的路径影响力 $P^h(u)$。

在获得 $PP_{ND,\theta,C}(u,S)$ 后，我们可以计算节点 $u$ 被种子点集 $S$ 激活的概率 $AP_S(u)=1-\prod_{h\in PP_{ND,\theta,C}(u,S)}[1-P^h(u)]$，其中 $1-P^h(u)$ 是 $u$ 没有被新颖性衰变下影响力传播路径 $h$ 激活的概率。在此基础上，基于影响力传播路径的方法计算种子点集 $S$ 影响力的算法如表 3-5 所示。该算法首先获得所有从 $S$ 出发的新颖性衰变下影响力路径 $PP_{ND,\theta,C}(S)$，然后根据不同的目标节点将其划分为多个 $PP_{ND,\theta,C}(u,S)$（lines 2-3）。接着，对于每个被至少一条新颖性衰变下影响力路径影响的点计算其被激活概率 $AP_S(u)$，最后累加获得种子点集影响力 $\sigma(S)$（lines 4-8）。

表3-5 基于影响力传播路径的方法计算种子点集 $S$ 影响力的算法的伪代码

**Input:** $G=\{V,E\},T_{uv},P_{uv},f(n),S,PP_{ND,\theta,C}(u,S)$
**Output:** $\sigma(S)$
1. $\sigma(S)\leftarrow 0$;
2. Get $PP_{ND,\theta,C}(S)$ by AD;
3. Divede $PP_{ND,\theta,C}(S)$ into different $PP_{ND,\theta,C}(u,S)$;
4. **for** every $u$ with $|PP_{ND,\theta,C}(u,S)|>0$ **do**
5. $\quad AP_S(u)=1-\prod_{h\in PP_{ND,\theta,C}(u,S)}[1-P^h(u)]$;
6. $\quad \sigma(S)\leftarrow\sigma(S)+AP_S(u)$;
7. **end for**
8. **return** $\sigma(S)$;

令 $N_{\theta,C}=\max_{|S|\le K}|PP_{ND,\theta,C}(S)|$ 为始于种子点集 $S$ 的新颖性衰变下影响力路径数量的最大值，且平均影响力传播路径长度为 $\bar{L}$，AD 算法的时间复杂度为 $O(\bar{L}N_{\theta,C}\log N_{\theta,C})$，同时，计算 $\sigma(S)$ 也会使用 $O(\bar{L}N_{\theta,C})$ 的时间。那么，基于影响力传播路径的方法总的时间复杂度为 $O(\bar{L}N_{\theta,C}\log N_{\theta,C})$。由于在实际应用中 $O(\bar{L}N_{\theta,C}\log N_{\theta,C})<<O(N_{MC}(N+M))$，基于影响力传播路径的方法比基于模拟的方法更高效。

## 3.5 实验及结果分析

### 3.5.1 实验环境

本次实验在 6 核 Intel Pentium(R) Xeon(R), 2.66GHz，24GB 内存的 Windows 服务器上进行，所有算法都由 C++语言实现。

### 3.5.2 实验数据集

除了前文所述的两个真实数据集Digg和Flickr，本次实验还采用了2个公共可用的真实数据集作为补充，这些数据集包含了大量的节点（用户）和节点间的连接关系，它们的基本信息如表3-6所示。NetPHY数据集收集了从1991年到2003年arXiv上的“Physics”领域的论文合作关系[5]。Wiki是一个投票网络数据集，包含了维基百科（Wikipedia）从创建到2008年1月的投票数据[17]。

表3-6 数据集的基本信息

| 数据集名称 | Wiki | NetPHY | Digg | Flickr |
|---|---|---|---|---|
| 点的数量 | 7K | 37K | 279K | 222K |
| 边的数量 | 103K | 181K | 1,731K | 14,727K |
| 点的平均度 | 14.6 | 6.2 | 6.2 | 66.3 |

### 3.5.3 参数设置

本次实验采用被广泛使用的权重级联策略（weighted cascade policy）[3,5-10]设置两点间影响（激活）的概率 $P_{uv}=1/N_{in}(v)$，其中 $N_{in}(v)$ 是节点 $v$ 的入度。对于每条边上的影响力期望延迟时间 $T_{uv}$，本次实验使用几何分布方式自动生成[21]，其中几何分布生成参数设置为5/*(outdegree(v)*+5)，最大的 $T_{uv}$ 设定为15，即如果几何分布生成的 $T_{uv}>15$，重新按照几何分布生成一个1-15的正整数赋值给 $T_{uv}$。Digg和Flickr数据集中新颖性衰变函数在前文中已经分析得出（见章节3.1.2），而对于Wiki和NetPHY数据集，本次实验采用默认的新颖性衰变函数 $f(n)=0.3^{n-1}$。新颖性衰变函数对新颖性衰变下影响力传播的最大化问题的影响会在后文具体讨论。参考以前工作的实验方法[3]，本文也为基于模拟的实验方法设定模拟次数 $N_{MC}=20000$。为了使基于影响力传播路径的方法在运行时间和计算精度上有较好的平衡，本次实验设置影响力传播路径的影响力最小阀值 $\theta=0.001$，对于同一个目标节点保留的影响力传播路径条数上限 $C=5$，实验效果将在具体实验中描述。

### 3.5.4 实验中的方法

本次实验涉及到的方法可以根据新颖性衰变下影响力传播的最大化问题的性质分为两大类。一类是选取影响力最大点集的方法，另一类是点集影响力的计算方法。我们首先讨论选取影响力最大点集的方法，然后在最高效的方法上分析点集影响力的计算方法。

选取影响力最大点集的方法：

1) 限制性贪婪算法(Restricted Greedy Algorithm，如表3-1所述)，表示为RGA;
2) 带有动态剪枝优化（Dynamic Pruning Optimization，如表3-2所述）的RGA，表示为RGA-DP。

点集影响力的计算方法：

1) 基于模拟的方法(Simulation Based Method with Monte Carlo，如表3-3所述)，表示为MC；
2) 基于影响力传播路径的方法（Propagation Path based Method，如表3-4所述），表示为PPAN。
3) 未考虑新颖性衰变的经典蒙特卡洛方法[3]，表示为CIM-MC。
4) 基于（节点上）度的方法（Degree Based Method），即选择拥有最大出度的 $K$ 个节点组成种子点集[5]，表示为DE。

### 3.5.5 选取影响力最大点集方法的对比

为了更全面对比分析选取影响力最大点集的方法RGA和RGA-DP，点集影响力的计算方法MC和PPAN都被应用到实验中，运行时间超过5天的实验结果被省略。

图3-6展示了在数据集Wiki上RGA和RGA-DP解决新颖性衰变下影响力传播的最大化问题的运行时间。实验结果表明，随着种子点集数量 $K$ 的不断增大，选取影响力点集方法需要消耗更多的时间，这是由于 $K$ 的增大增加了选取影响力最大点集的方法的循环次数。当点集影响力的计算方法同时采用MC或PPAN时，RGA-DP都比RGA在速度上快了大约2个数量级，说明DP动态剪除影响力较小点的策略极大地提高了RGA的效率。

图3-7展示了在数据集NetPHY上RGA和RGA-DP解决新颖性衰变下影响力传播的最大化问题的运行时间。与数据集Wiki上实验结果相似，选取影响力点集方法消耗的运行时间随着种子点集数量 $K$ 增大。由于DP动态剪除影响力较小点的优化效果，无论点集影响力的计算方法同时采用MC还是PPAN，RGA-DP都比RGA在速度上快了大约2个数量级。

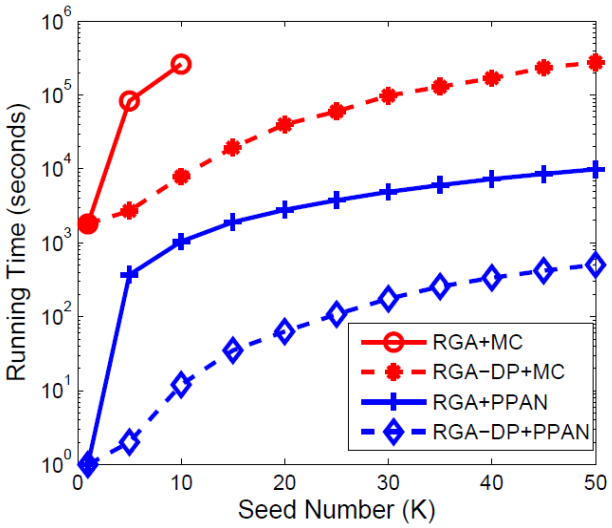


图 3-6 RGA 和 RGA-DP 在 Wiki 和 NetPHY 上运行时间对比

因此，在后面对比计算点集影响力方法的实验中，我们默认采用RGA-DP作为选取影响力最大点集的方法。注意，由于DP在理论上保证了RGA解的质量，对于解质量的对比实验可省略。此外，当选取影响力点集方法相同时，PPAN的速度比MC快了2-3个数量级，说明PPAN比MC在实际应用中效率高很多。

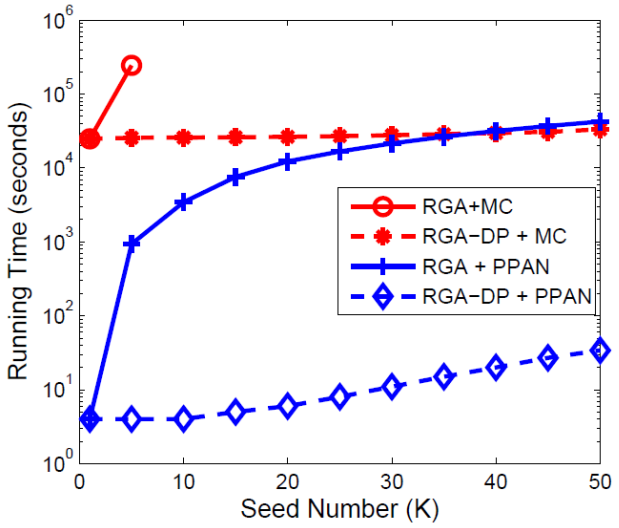


图 3-7 RGA 和 RGA-DP 在 Wiki 和 NetPHY 上运行时间对比

### 3.5.6 计算点集影响力方法的对比

本次实验在 4 个数据集上都从选出点集的影响力和运行时间上对以上列出的计算点集影响力的方法做了对比实验。运行时间超过 5 天的实验结果被省略，因此大数据集 Digg 和 Flickr 上没有 MC 和 CIM-MC 的实验结果。

图 3-8 给出了计算点集影响力的方法 MC、PPAN、CIM-MC 和 DE 在数据集

Wiki 上选出点集的影响力的对比结果，选出点集的影响力随着种子点集数量 $K$ 的增大而快速增长，这是因为 $K$ 越大种子点集包含的节点越多，点集的影响力也会越大。较其他的方法，MC 采用模拟的方式选出了拥有最大影响力的点集。PPAN 获得的点集的影响力和 MC 的非常接近，说明 PPAN 是非常有效的计算点集影响力的方法，另一方面，CIM-MC 和 DE 选出的点集的影响力明显低于 MC 和 PPAN，这说明解决普通影响力传播的最大化问题的计算点集影响力的方法没有考虑新颖性衰变问题，而不适用于新颖性衰变下影响力传播的最大化问题。

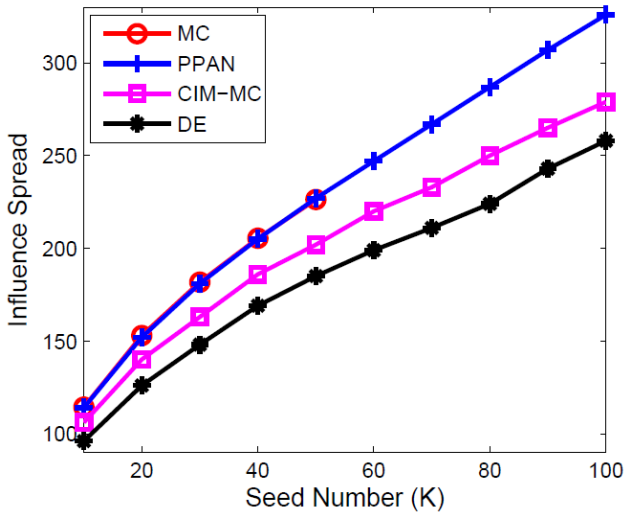


图 3-8 计算点集影响力方法在数据集 Wiki 上选出点集影响力对比

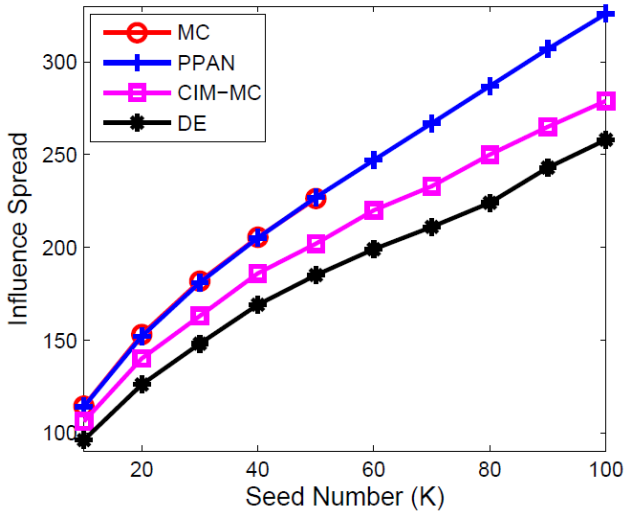


图 3-9 计算点集影响力方法在数据集 NetPHY 上选出点集影响力对比

图 3-9 给出了计算点集影响力的方法 MC、PPAN、CIM-MC 和 DE 在数据集 NetPHY 上选出点集的影响力的对比结果，同数据集 Wiki 上结果相似，选出点集的影响力随着种子点集数量 $K$ 的增大而快速增长，MC 选出的点集影响力是最大的

点集，同时 PPAN 选出的点集的影响力和 MC 的非常接近，而 CIM-MC 和 DE 选出的点集的影响力明显低于 MC 和 PPAN，并且 DE 相比其他方法，选出点集影响力小很多，说明这种直观的选择方式很难保证解的质量。

图 3-10 和图 3-11 分别给出了计算点集影响力的方法 PPAN 和 DE 在数据集 Digg 和 Flickr 上选出点集的影响力的对比结果，在两个数据集上，PPAN 选出点集的影响力都明显好于 DE 选出的点集。

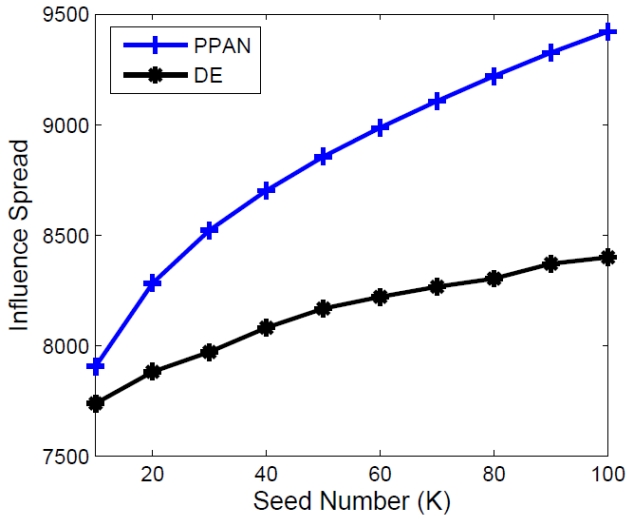


图 3-10 计算点集影响力方法在数据集 Digg 上选出点集影响力对比

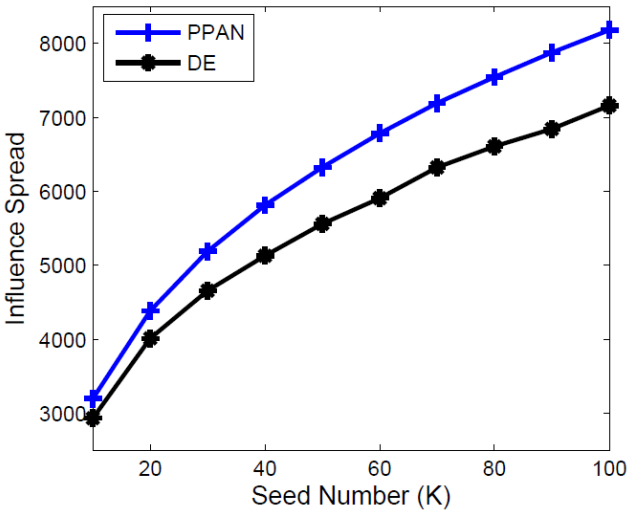


图 3-11 计算点集影响力方法在数据集 Flickr 上选出点集影响力对比

从以上 4 个数据集中的对比实验结果可见，PPAN 是非常有效的计算点集影响力的方法，它能基本达到 MC 的效果，而解决普通影响力传播的最大化问题的计算点集影响力的方法 CIM-MC 和 DE 不适用于新颖性衰变下影响力传播的最大化问题。接下来，本次实验从运行时间入手，对比分析计算点集影响力的方法的效

率。

由于 DE 运行时间极短（几秒内），同时 CIM-MC 和 MC 运行时间相差不大（都是耗时的模拟方法），DE 和 CIM-MC 实验结果不再展示。图 3-12 和图 3-13 分别在数据集 Wiki 和数据集 NetPHY 上对比了 MC 和 PPAN 在解决新颖性衰变下影响力传播的最大化问题的运行时间。在这两个数据集上，两个算法的运行时间都随着种子点集数量 $K$ 的增大而快速增长，这是因为 $K$ 的增大使选择影响力最大点集的策略的外层循环和计算点集影响力方法的计算时间都增加。另一方面，PPAN 的速度比 MC 快大约 3 个数量级，因此 PPAN 是高效且快速的计算点集影响力的方法。

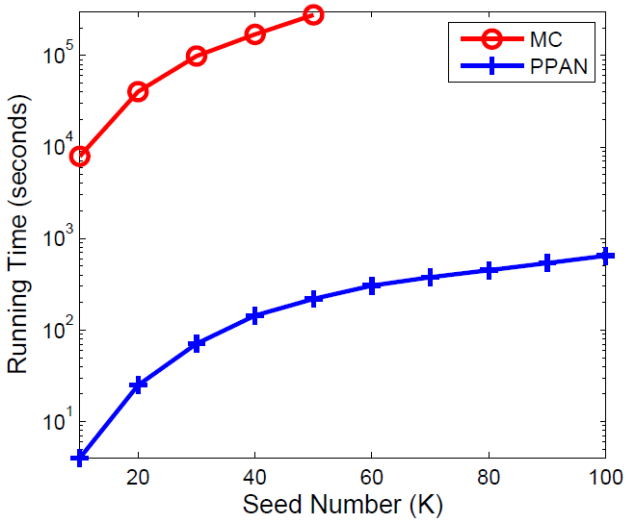


图3-12 计算点集影响力方法MC和PPAN在数据集Wiki上运行时间对比

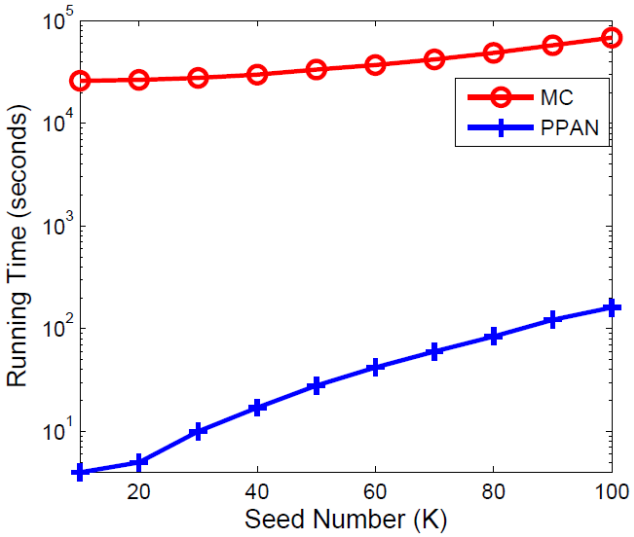


图3-13 计算点集影响力方法MC和PPAN在数据集NetPHY上运行时间对比

为了进一步测试计算点集影响力方法 PPAN 的可扩展性，图 3-14 和图 3-15 分别展示了在大规模数据集 Digg 和数据集 Flickr 上 PPAN 解决新颖性衰变下影响力传播的最大化问题的运行时间。同理，在这两个数据集上，PPAN 的运行时间都随着种子点集数量 $K$ 的增大而增长。同时，PPAN 能在几小时内求解问题，说明 PPAN 的可扩展性较好。

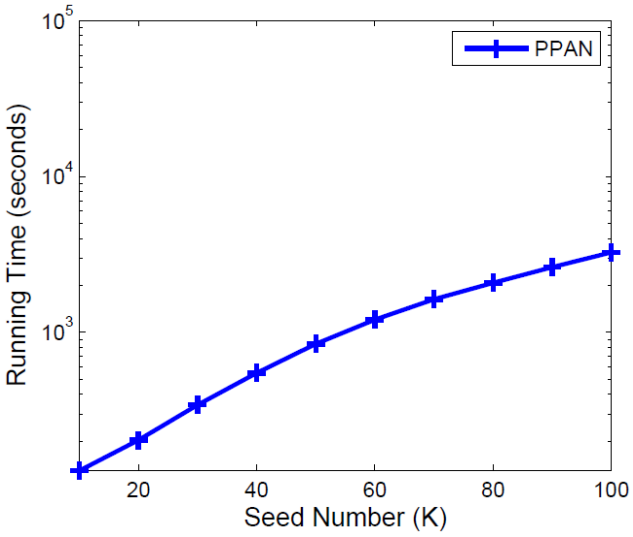


图3-14 计算点集影响力方法PPAN在数据集Digg上运行时间情况

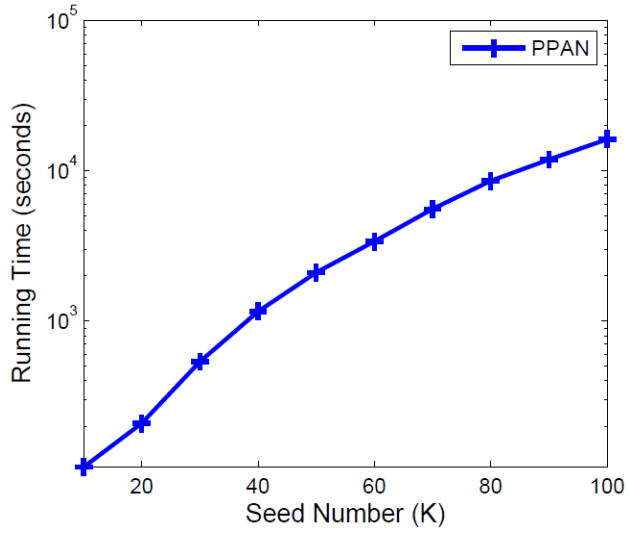


图3-15 计算点集影响力方法PPAN在数据集Flickr上运行时间情况

### 3.5.7 新颖性衰变函数对问题的影响

本次实验进一步在数据集 Wiki 和 NetPHY 上实验分析了新颖性衰变函数 $f(n)=\gamma^{n-1}$ 中 $\gamma$ 对于选出点集的影响力的影响。图 3-14 和图 3-15 分别展示了在 Wiki

和 NetPHY 上选出点集的影响力随新颖性衰变函数参数 $\gamma$ 变化的情况（$K=50$）。如预期效果一致，选出点集的影响力随 $\gamma$ 的增大而迅速增长，这是由于 $\gamma$ 越大，新颖性衰变越少，则种子点集可能激活的节点数越多，其影响力越大。此外，PPAN 选出的点集的影响力依然与 MC 选出的点集的影响力非常接近，再次说明 PPAN 是非常高效的计算点集影响力的方法。

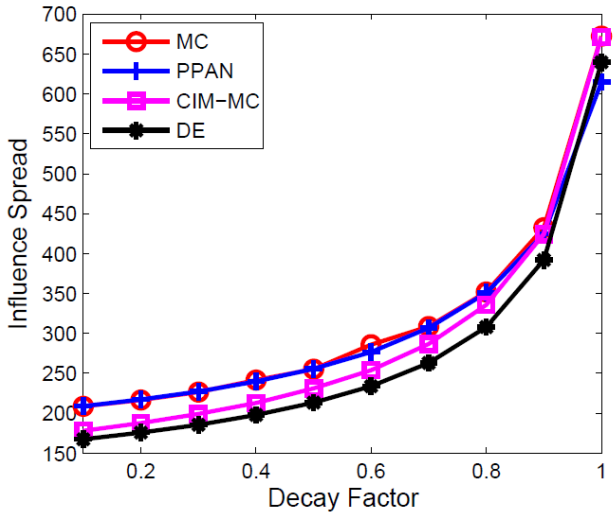


图3-14 在Wiki上选出点集的影响力随新颖性衰变函数参数 $\gamma$ 变化的情况

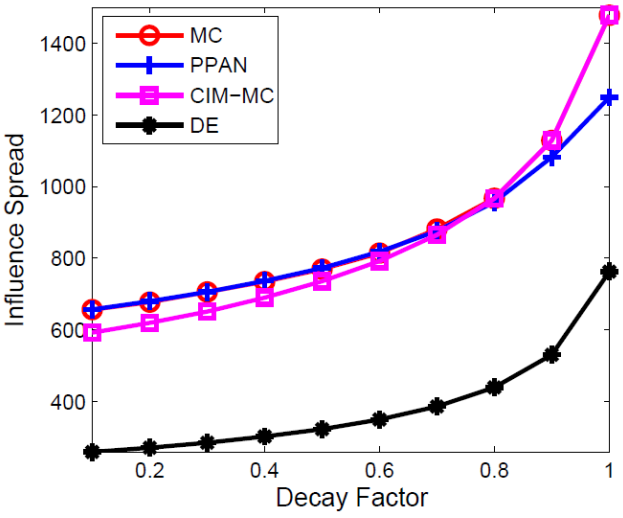


图3-15 在NetPHY上选出点集的影响力随新颖性衰变函数参数 $\gamma$ 变化的情况

## 3.6 本章小结

本章首先在相关研究工作基础上通过数据分析的方式讨论了新颖性衰变对影响力传播过程中的影响从而标准化新颖性衰变函数，接着建立了新颖性衰变下影响力传播模型并定义了新颖性衰变下影响力传播的最大化问题。为了在新影响力传播模型中解决新的问题，本章分析了新颖性衰变下影响力的传播模型和新定义问题的性质，还提出了高效的选出影响力最大点集的贪婪算法和动态剪枝优化，以及计算点集影响力的方法。最后，这些方法的高效性在 4 个真实数据集中的实验结果中得到了验证。

# 第四章 节点失效下影响力传播的最大化问题

本章将讨论节点失效下影响力传播的最大化问题。首先，分析节点失效对影响力传播的最大化问题的影响；然后，标准化定义节点失效下影响力传播的最大化问题，并证明该问题是 *NP-hard*；接着，提出解决该问题的限制性模拟退火算法（CSA）及其优化策略（CSA-Q）；最后，通过 4 个真实数据集中的实验结果，验证了我们提出的方法的高效性。

## 4.1 节点失效对影响力传播的最大化问题的影响

在理想的情况下，影响力传播的最大化问题中找到的种子点集能将某个影响力最大限度的传播到网络中，这种理想情况必须满足所有选中的种子点集能够正常的工作并将影响力传播出去。但是，在实际情况中被选中的种子节点有可能无法发挥期望的作用，比如，为推广一款新的产品某公司会选择一些在社交网络上有影响力的人作为免费试用对象。由于个人喜好（如试用者不喜欢该产品）或者偶然事件（如试用者声誉骤降）都可能导致产品无法像预期那样被最大限度的推广到网络中，在这种情况下，该公司会更倾向于选择更稳定的试用对象群（如影响力丢失在可容忍的范围内）。又如，在水分布网络中安放一定数量传感器并最大限度监测水质情况的问题[4]中，传感器很有可能出现失效的情况而导致水质监测范围达不到预期的效果，所以节点失效情况在这个问题中也必须考虑。

当考虑节点失效时，情况变得非常复杂，因为节点的失效在选择种子点集时是未知的。此外，预测每个节点失效的概率在实际应用中是极难的[49]，尤其是当网络规模很大的时候，因为单个节点失效的概率由很多不确定的因素（如环境，节点自身状况等）决定，而且它随着时间不断变化。在另一方面，根据分析网络上之前的数据，估算可能失效节点的数量是比较容易的。比如，在社交网络上推广一款新产品时，由于每个候选人的情况不同，获得每个候选人失效的概率是非常困难的，但估算可能失效节点的数量是相对容易的。同理，在水分布网络中，因为不同安置点环境差异很大，预测每个安置点传感器失效的概率也是很困难的，而估计可能失效节点的数量却较容易。因此，本文主要研究给定失效节点数量情况下影响力传播的最大化问题。

节点失效会对影响力传播的最大化产生很大的影响，尤其是当节点失效发生在选定的种子点集中（后文实验部分有详细说明），因此本文主要讨论发生在种子点集中节点失效的问题。如上文所述，在考虑节点失效时，我们往往会有一个影

响力丢失的容忍度，即影响力丢失不超过一个阀值。为了说明节点失效对于影响力传播的最大化的影响，本文举例如图 4-1 所示，给定 $K=3$，种子点集 $\{V_2,V_5,V_7\}$ 体现出最大的影响力（影响力值为 4.9012），而影响力的值排第二的种子点集是 $\{V_1,V_2,V_5\}$（影响力值为 4.864）。假设种子点集中有一个节点失效，在 $\{V_2,V_5,V_7\}$ 中 $V_2$ 失效造成最大影响力丢失（值为 2.4412），而在 $\{V_1,V_2,V_5\}$ 中 $V_1$ 失效造成最大影响力丢失（值为 1.2264）。给定一个影响力丢失阀值 1.5，影响力值排第二的 $\{V_1,V_2,V_5\}$ 才是最优解，因为这个解更加的稳定。

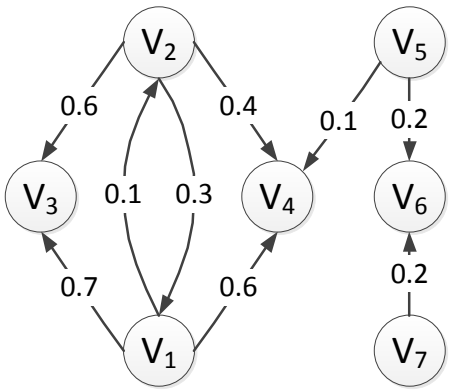


图4-1 一个包含影响概率的社交网络

## 4.2 节点失效下影响力传播的最大化问题的定义

当考虑种子点集的节点失效时，影响力传播的过程并没有受影响，因此经典的影响力传播模型（线性阀值模型和独立级联模型）都适用，由于两个模型的等价性[3]，本文仅选择更流行的独立级联模型来讨论。

种子点集中失效节点的集合被标记为 $A$，其节点数被标记为 $R=|A|$，则 $A\subseteq S$ 且 $R\le K$。影响力丢失表示为 $[\sigma(S)-\sigma(S-A)]$，其中 $\sigma(S-A)$ 是 $R$ 个节点失效后，剩余的种子点集的影响力，该影响力需要计算剩余 $K-R$ 个节点组成的点集的影响力，属于一个计算点集影响力的问题。种子点集中 $R$ 个节点失效有很多种情况，为了满足前文提到的影响力丢失的阀值，本文在定义问题时仅考虑最坏的情况，即剩余 $K-R$ 个节点组成点集的影响力最小，此时出现了最大影响力丢失 $MaxLoss=\sigma(S)-\min_{A\subseteq S}\sigma(S-A)$。

对于一个给定的网络，节点失效下影响力传播的最大化问题的目标是寻找一个包含 $K$ 个节点的种子点集 $S$ 使得 $S$ 的影响力 $\sigma(S)$ 最大化，且包含 $R$ 个节点的失效点集 $A$ 造成的影响力丢失不超过一个影响力丢失阀值 $\eta$（>0）。节点失效下影响力传播的最大化问题的标准化定义如下：

**定义** 4-1（节点失效下影响力传播的最大化问题）：给定一个有向图$G=\{V,E\}$，一个种子节点数$K$（正整数），一个失效节点数$R$（正整数）和影响力丢失阀值$\eta$（正数）寻找一个种子点集$S\subseteq V$使得

$$Objective: S=\arg\max_{s\subseteq V,|S|=K}\sigma(S)$$

$$Constraint: MaxLoss=\sigma(S)-\min_{A\subseteq S}\sigma(S-A)\le\eta \tag{4-1}$$

与普通的影响力传播的最大化问题一样，节点失效下影响力传播的最大化问题也是 *NP-hard*。证明如下：

**推论** 4-1：节点失效下影响力传播的最大化问题是 *NP-hard*。

**证明**：给定一个普通的影响力传播的最大化问题的实例$\varphi: G=\{V,E\}$和$K$，我们可以构造一个节点失效下影响力传播的最大化问题的实例$\omega$通过增加影响力丢失阀值$\eta\ge N$。由于$MaxLoss=\sigma(S)-\min_{A\subseteq S}\sigma(S-A)\le\eta$且$\sigma(S)\le N$，则任意$S$都满足影响力丢失不超过阀值的限制条件，那么，$S$是$\omega$的解当且仅当$S$是$\varphi$的解。因为普通的影响力传播的最大化问题已经被证明是 *NP-hard*[3]，所以节点失效下影响力传播的最大化问题也是 *NP-hard*。■

节点失效下影响力传播的最大化问题需要在给定网络中设置参数$K$,$R$和$\eta$。通常情况下，$K$与具体应用的预算有关，$R$由种子节点的稳定性决定，$\eta$取决于决策者的偏好，一般与收益的风险相关。在本章的实验部分，本文将讨论各参数取值对于节点失效下影响力传播的最大化问题的影响。

## 4.3 限制性模拟退火算法及其优化策略

本节首先说明常用于普通影响力传播的最大化问题的贪婪算法（Greedy Algorithm）[3]不适用于节点失效下影响力传播的最大化问题并给出一个可行的新贪婪算法（GreedyB），然后详细说明本文提出的一个限制性模拟退火算法（Constrained Simulated Annealing-CSA），最后给出提高限制性模拟退火算法的优化策略。

### 4.3.1 贪婪算法不适用性及可行的新贪婪算法

对于求解节点失效下影响力传播的最大化问题，很自然的想法是采用在影响力传播的最大化问题中广泛使用的贪婪算法，因为这个算法时间复杂度很低，而且在普通的影响力传播的最大化问题中能达到（1-1/e） 的近似解[3]。贪婪算法的基本思想是：每个迭代从网络中选择一个边际影响力最大的点放入点集（使点集的影响力增长最大的点），直到点集中点的数量达到$K$为止（选出所有点）。

但是，由于贪婪算法在选择点集的时候不会考虑节点失效导致的影响力丢失，它选择的点集可能会存在丢失大量影响力的风险，如果选择出的点集可能丢失的影响力大于影响力丢失阀值$\eta$，贪婪算法所选择出的点集不是节点失效下影响力传播的最大化问题的可行解。例如，用贪婪算法解决在图 4-1 所示的网络中节点失效下影响力传播的最大化问题（$K=3, R=1, \eta=1.5$）时，贪婪算法会首先选择$V_2$，然后选择$V_5$，最后选择$V_7$，从而得到能得到影响力最大的点集$\{V_2, V_5, V_7\}$。然而，正如前文所述，$\{V_2, V_5, V_7\}$的最大影响力丢失超过了阀值$\eta$，它并不是一个可行解。

为了保证在节点失效下影响力传播的最大化问题中找到可行解，本文提出一个回朔策略来扩展贪婪算法，得到的新贪婪算法表示为 GreedyB。该算法首先采用原贪婪算法的方式选出影响力最大的种子点集，接着不断更新种子点集直到种子点集满足影响力丢失的限制条件。为了简化操作，在每次更新中，有且仅有一个节点被更新。基于以上回朔策略的新贪婪算法能保证找到的解在节点失效下影响力传播的最大化问题中是可行的（可行解存在的情况下）。但是由于回朔策略只考虑了满足影响力丢失的限制条件，并没有关心解的质量（点集影响力大小），新贪婪算法的解的质量没有任何的理论保证。

参考文献[27]对于贪婪算法的扩展策略，另一个可能的方式是设计一个策略（或者目标函数）使得贪婪算法每次选择种子点时在点集影响力增量和影响力丢失增量之间做一个权衡（trade-off）。不过，由于影响力丢失对点集的变化非常敏感，无法在不完成所有点的选择之前（选出 $K$ 个点）计算出来，这个可能的方式是很难做出的。因此，本文着力从另外的选择影响力最大点集的方法（模拟退火算法）入手，提出解决节点失效下影响力传播的最大化问题的方法。

### 4.3.2 限制性模拟退火算法

本节将详细描述用于解决节点失效下影响力传播的最大化问题的限制性模拟退火算法（CSA），讨论其参数设定，并证明其渐进收敛性。

根据上一节分析可知，解决节点失效下影响力传播的最大化问题需要一个系统的能够直接获得并度量完整解的搜索策略。在选择影响力最大点集的策略中，模拟退火算法能够直接获得并度量完整解，但考虑到节点失效下影响力传播的最大化问题是一个限制性非线性优化问题，本文提出了考虑限制条件的限制性模拟退火算法[31,50]来解决节点失效下影响力传播的最大化问题。

为了保证和基本限制性模拟退火算法的一致性，我们首先将目标函数的最大化问题转换为最小化问题，然后将限制条件转换为等价的最大化等式 $Max[0, \sigma(S)-\min_{A\subseteq S}\sigma(S-A)-\eta]=0$，这个新等式成立的条件是当且仅当

$MaxLoss \le \eta$。因此，总的惩罚函数表示为：

$$L_{\Omega}(S,\lambda) = -\sigma(S) + \lambda \times Max[0, \sigma(S) - \min_{A \subseteq S} \sigma(S-A) - \eta] \quad (4\text{-}2)$$

其中$\Omega$是所有解和惩罚系数$\lambda$组成的联合空间，一个状态$D$是一个属于该联合空间的解，且$D=(S,\lambda) \in \Omega$。惩罚系数$\lambda$的选择取决于所解决的具体问题，本文取相对大的$\lambda$值，使得惩罚函数值在限制条件不满足情况下变得很大而不易被接受。

解决节点失效下影响力传播的最大化问题的限制性模拟退火算法（CSA）的伪代码如表 4-1 所示，总体上，CSA 在联合空间$\Omega$上对于每个温度都进行一定量的搜索，其间以降温调度决定的概率更新解，直到温度足够低的时候得到最优解。CSA 开始于初始化包含点集$S$和惩罚系数$\lambda$的解$D$和温度$T_0$（lines 1-2），通过计算点集影响力$\sigma(S)$和$\min_{A \subseteq S} \sigma(S-A)$得到解$D$的惩罚函数值$L_D(S,\lambda)$（line 4）。然后，CSA 更新$D$的邻居解集$N_D$，同时在$N_D$中随机产生一个新解$D'$，接着计算出解$D$和$D'$的惩罚函数之差$\Delta_L$（lines 5-7）。接下来，CSA 判断是否接受新解（lines 9-23），其接受的概率在子空间$\lambda$和$S$上是不同的。如果新解的改变发生在$S$上，CSA 直接接受有更小惩罚函数值的新解，因为这样的解的$\sigma(S)$更大；对于惩罚函数值更大的新解 CSA 仅以概率$\exp(-\Delta_L / T_t)$接受，这个概率值随温度降低而减小（lines 9-19）。如果新解的改变发生在$\lambda$上，CSA 直接接受有更大惩罚函数值的新解，因为这样的新解中$\lambda$在变大，推动着新解更加满足限制条件，另一方面，CSA 也以逐渐随温度降低而减小的概率$\exp(-\Delta_L / T_t)$接受有更小$\lambda$的新解（lines 20-32）。当 CSA 在温度$T_t$上搜索解$q$次后，更新温度同时开始新的搜索（lines 33-36）。当且仅当温度$T_t$达到终止温度$T_f$时，CSA 结束搜索过程，返回最优解（line 38）。

表4-1 限制性模拟退火算法（CSA）的伪代码

**Input:** $G=\{V,E\}, P_{uv}, K, R, \eta, T_0$ (initial temperature of CSA),
$T_f$ (termination temperature of CSA), $q$ (trial number)
**Output:** $S$
1. $t \leftarrow 0, T_t \leftarrow T_0, c \leftarrow 0$;
2. Initialize a solution $D \leftarrow (S,\lambda)$ including $\lambda = 0$,
   and a seed set $S \in V$ with $|S| = K$;
3. **while** $T_t > T_f$ **do**
4. Compute $L_D(S,\lambda)$;
5. Update a neighbor set $N_D$ for $D$;

6. Generate a trail point $D' \in N_D$ randomly;
7. Compute the change of $L$: $\Delta_L \leftarrow L_{D'} - L_D$
8. Set $c \leftarrow c+1$;
9. **if** $\lambda' = \lambda$
10. **if** $\Delta_L < 0$
11. $D \leftarrow D'$;
12. **end if**
13. **else**
14. Generate a random number $\theta \in (0,1)$;
15. **if** $\exp(-\Delta_L / T_t) > \theta$
16. $D \leftarrow D'$;
17 **end if**
18. **end else**
19. **end if**
20. **else**
21. **if** $S' = S$
22. **if** $\Delta_L > 0$
23. $D \leftarrow D'$;
24. **end if**
25. **else**
26. Generate a random number $\theta \in (0,1)$;
27. **if** $\exp(-\Delta_L / T_t) > \theta$
28. $D \leftarrow D'$;
29. **end if**
30. **end else**
31. **end if**
32. **end else**
33. **if** $c > q$;
34. Compute the trial temperature $\rho$;
35. $t \leftarrow t+1, T_t \leftarrow q \times \rho / \ln(t+1), c \leftarrow 0$;
36. **end if**
37. **end while**
38. **return** $S$;

如文献[31]所述，新解在变量$S$上改变的次数应是在变量$\lambda$上改变次数的 10 倍以上，对任意一个$S$其相邻点集有$K(N-K)$个，因此本文设定$\lambda$的离散取值为$\Lambda(\lambda)=\{0,\max\sigma(S)/[10\%K(N-K)],2\max\sigma(S)/[10\%K(N-K)],\cdots,\max\sigma(S)\}$。参数$\rho$是调节温度$T_f$的重要参数，其在 CSA 于某个温度上完成了$q$次搜索后会被更新（line 25）。同时，$\rho$也与 CSA 的渐进收敛性有关，因为它决定了 CSA 的迭代执行次数，本文设定$\rho=2\max\sigma(S)(1+\max\sigma(A))$，这个设定足以保证 CSA 的收敛性（证明在后文中）。剩余的问题是如何高效的估算$\max\sigma(S)$和$\max\sigma(A)$。

计算准确的$\max\sigma(S)$和$\max\sigma(A)$实际上是普通的影响力传播的最大化问题，是很耗时的，而 CSA 只需要较准确的估算$\max\sigma(S)$和$\max\sigma(A)$的上界，本文用$|S|$个单点点集的最大和$\max\sum_{u\in S}\sigma(u)$作为$\max\sigma(S)$上界的估算值，其中由点集影响力计算函数的子模性质，可得$\max\sum_{u\in S}\sigma(u)\geq\max\sigma(S)$，同理我们也可用$|A|$个单点点集的最大和$\max\sum_{u\in A}\sigma(u)$作为$\max\sigma(A)$的估算值。对于点集影响力的计算，本文采用前文提及的目前最好的快速计算点集影响力的方法之一的影响力传播路径方法（ISP）[17]。

接下来，本文证明 CSA 的渐进收敛性。增加惩罚系数$\lambda$到每个候选种子点集$S$中（如惩罚函数$L_\Omega(S,\lambda)$所示）后，CSA 可以解决节点失效下影响力传播的最大化问题。令$Z_{D,D'}$作为从解$D$的邻居解集$N_D$产生新解$D'$的概率，由于邻居点集是$S$和$\lambda$的联合，可得$Z_{D,D'}=1/[K(N-K)+|\Lambda|-1]$，其中$|\Lambda|$是$\lambda$的值域空间$\Lambda$的大小。

CSA 接受新解的概率在子空间$S$和$\lambda$中各不相同（如表 4-1 中 lines 9-23 所示），则这个概率求解为：

$$F_{D,D'}=\begin{cases}\exp(-\max[0,L_{D'}-L_D])/T_t, D'=(S',\lambda);\\ \exp(-\max[0,L_D-L_{D'}])/T_t, D'=(S,\lambda').\end{cases}$$

那么，从解$D$转移到新解$D'$的概率为：

$$I_{D,D'}=\begin{cases}Z_{D,D'}F_{D,D'}, D'\in N_D;\\ 1-\sum_{W\in N_D}Z_{D,W}F_{D,W}, D'=D;\\ 0, otherwise.\end{cases}$$

因为不同解之间的转移概率在不同的子空间中不同且这种转移在整个过程不断发生，CSA 算法的执行过程可以建模成一个不均匀的马尔可夫链（Markov chain），让$q$为在各种子空间中从任意解转移到最优解的最小转移次数的最大值，且$q$可以在邻居解集被合理构建的基础上获得。因此，在降温调度中（如表 4-1 中 lines 25 所示），$T_t=q\times\rho/\ln(t+1)$，且在前文已设定$\rho=2\max\sigma(S)(1+\max\sigma(A))$，可得如下推论。

**推论** 4-2：降温调度设置 $T_t = q \times \rho / \ln(t+1)$ 满足以下属性：1) $T_t > T_{t+1}$，2) $\lim_{t\to\infty} T_t = 0$，3) $T_t \geq 2q/\ln(t+1) \times \max|L_{D'} - L_D|$。

证明：因为 $q$ 和 $\rho$ 是常数，且 $T_t$ 是随 $t$ 单调递减的函数，属性 1)和 2)是显然成立的。接着，我们从 $S$ 或者 $\lambda$ 在联合空间上发生改变的两种情况讨论，计算 $\max|L_{D'} - L_D|$：

a) 情况 1：$D' = (S', \lambda)$，则

$\max|L_{D'} - L_D| = |-\sigma(S') + \lambda \times \max[0, (\sigma(S') - \min\sigma(S' - A'))]$

$-[-\sigma(S) + \lambda \max[0, (\sigma(S) - \min\sigma(S - A))]|$（由于 $\sigma(S) - \sigma(S-A) \leq \sigma(A)$）

$\leq |\sigma(S) + \lambda \max\sigma(A')|$（由于 $\lambda \leq \max\sigma(S)$）

$\leq \max\sigma(S) + \max\sigma(S) \times \max\sigma(A')$

b) 情况 2：$D' = (S, \lambda')$，则

$\max|L_{D'} - L_D| = |\lambda' \times \max[0, (\sigma(S) - \min\sigma(S-A) - \eta)] - \lambda \times [-\sigma(S) + \lambda \times$

$\max[0, (\sigma(S) - \min\sigma(S-A) - \eta)]] \leq \lambda' \times \max\sigma(A)$（由于 $\lambda \leq \max\sigma(S)$）

$\leq \max\sigma(S) \times \max\sigma(A)$

因为 $T_t = q \times \rho / \ln(t+1) = 2q \times \max\sigma(S)(1 + \max\sigma(A)) / \ln(t+1)$，所以属性 3)也是成立的。■

给定一个转移概率 $I_{D,D'}$ 和满足推论 4-2 所述性质的递减 $T_t$，根据文献[31]的证明，可以证明出 CSA 的渐进收敛性，即可得如下推论：

**推论** 4-3：当 $t \to \infty$ 时，用于解决节点失效下影响力传播的最大化问题的 CSA 收敛于一个全局最优解。

本文采用快速计算点集影响力的影响力传播路径方法（ISP）去计算 CSA 中每次迭代循环都需要计算的 $\sigma(S)$ 和 $\sigma(S) - \sigma(A)$。令 ISP 需要访问的所有影响力传播路径组成的集合分别为 $PP(S)$ 和 $PP(S-A)$，且平均影响力传播路径长度为 $\bar{L}$，则，用深度优先算法访问这些路径集合需要使用的时间分别为 $O(\bar{L}|PP(S)|)$ 和 $O(\bar{L}|PP(S-A)|)$，那么，CSA 总的时间复杂度为 $O(qT\bar{L}(|PP(S)| + C(K,R)|PP(S-A)|))$，其中 $T$ 是降温调度（CSA 中 while 循环）总的调节次数，$C(K,R)$ 是 $K$ 个种子点中失效 $R$ 个点的组合方式的数量。

### 4.3.3 优化的限制性模拟退火算法

限制性模拟退火算法（CSA）中有大量对于惩罚函数的计算，而每次计算中都会考虑 $K$ 个种子点中失效 $R$ 个点的所有组合方式，这使得 CSA 非常耗时，尤其在 $C(K,R)$ 很大时。为了提高 CSA 的效率，本文提出了新的惩罚函数，并得到优化的限制性模拟退火算法（CSA-Q）。

在节点失效下影响力传播的最大化问题中，点集影响力计算公式$\sigma(S)$有单调（monotone）和子模的（submdular）性质，利用这些性质可以得到点集影响力丢失的上界如下，

**推论 4-4**：点集影响力丢失$\sigma(S)-\min_{A\subseteq S}\sigma(S-A)$的上界为$\max\sum_{i=1,u_i\in S}^{R}\sigma(u_i)$。

**证明**：因为$\sigma(S)$是子模的，可得$\min\sigma(S-A)\geq\min[\sigma(S)-\sigma(A)]=\sigma(S)-\max\sigma(A)$，那么$\sigma(S)-\min_{A\subseteq S}\sigma(S-A)\leq\sigma(S)-[\sigma(S)-\max_{A\subseteq S}\sigma(A)]=\max_{A\subseteq S}\sigma(A)$。同理，可得$\max_{A\subseteq S}\sigma(A)\leq\max\sum_{i=1,u_i\in S}^{R}\sigma(u_i)$，其中$\max\sum_{i=1,u_i\in S}^{R}\sigma(u_i)$是单点影响力最大的$R$个点组成的点集的影响力值，因此，$\sigma(S)-\min_{A\subseteq S}\sigma(S-A)\leq\max_{A\subseteq S}\sigma(A)\leq\max\sum_{i=1,u_i\in S}^{R}\sigma(u_i)$。■

根据推论 4-4，可以用点集影响力丢失的上界$\max\sum_{i=1,u_i\in S}^{R}\sigma(u_i)$作为其估算值，由此可得新的惩罚函数$L'_{\Omega}(S,\lambda)$如下，

$$L'_{\Omega}(S,\lambda)=-\sigma(S)+\lambda\times Max[0,\sigma(S)-\max\sum_{i=1,u_i\in S}^{R}\sigma(u_i)-\eta] \tag{4-3}$$

新的惩罚函数的运算量远远小于原来的惩罚函数，这是因为，一方面，新的惩罚函数不需要考虑$K$个种子点中失效$R$个点的所有组合方式，另一方面，所有单点的影响力值可以在预处理的时候一次性全部计算好并存储起来，然后重用。将新的惩罚函数用到 CSA 的算法框架（如表 4-1 所示）中，可以得到一个优化的限制性模拟退火算法（表示为 CSA-Q）。由于新的惩罚函数的运算量较原惩罚函数有很大降低，CSA-Q 的效率也会比 CSA 高很多，更重要的是，CSA-Q 用点集影响力丢失的一个上界去估算点集影响力丢失，在理论上保证了 CSA-Q 能找到可行解。证明如下：

**推论 4-5**：在节点失效下影响力传播的最大化问题中，CSA-Q 返回的是可行解。

**证明**：由新的惩罚函数（公式 4-3 所示）可知，CSA-Q 返回的解满足限制条件为$\max\sum_{i=1,u_i\in S}^{R}\sigma(u_i)\leq\eta$，结合推论 4-4，可得$\sigma(S)-\min_{A\subseteq S}\sigma(S-A)\leq\max\sum_{i=1,u_i\in S}^{R}\sigma(u_i)\leq\eta$。由此可知，CSA-Q 返回的是可行解。■

CSA-Q 只是将 CSA 中的惩罚函数替换成了一个新的惩罚函数，并且保证了找到可行解的性质，因此，CSA-Q 同样满足 CSA 的渐进收敛些（如推论 4-3 所示）。虽然理论上 CSA-Q 返回的解的质量会比 CSA 差一些，但大量实验表明（参照后文实验部分），在实际应用中 CSA-Q 与 CSA 返回的解的质量非常接近。

同理，CSA-Q 也采用 ISP 去计算点集影响力。CSA-Q 避免了 CSA 中每次计算点集影响力丢失考虑$K$个种子点中失效$R$个点的所有组合方式，同时可以一次性计算并存储所有单点的影响力，则 CSA-Q 的时间复杂度为$O(qT\bar{L}(|PP(S)|)+\bar{L}_u(|PP(u)|))$，其中$PP(u)$和$\bar{L}_u$分别是 ISP 计算单点影响力时需

要访问的所有点单影响力传播路径组成的集合和平均影响力传播路径长度。由于 $\bar{L}_u(|PP(u)|) << \bar{L}(|PP(S)|) << qT\bar{L}C(K,R)|PP(S-A)|$，CSA-Q 比 CSA 效率高很多。

## 4.4 实验及结果分析

### 4.4.1 实验环境

本次实验在 4 核 Intel i7-3770, 3.4GHz，8GB 内存的 Linux PC 上进行，所有算法都由 C++语言实现。

### 4.4.2 实验数据集

本次实验采用了4个不同大小的公共可用真实数据集，这些数据集的基本信息如表4-2所示。NetHEPT数据集收集了从1991年到2003年arXiv上的“High Energy Physics: Theory”领域的论文合作关系[5]。Wiki是一个投票网络数据集，包含了维基百科（Wikipedia）从创建到2008年1月的投票数据。Epinions是来自于Epinions.com上由相互信任关系建立起来的社交网络[17]。Amazon是从Amazon website上2003年3月收集到的消费关系数据[32]。

表4-2 数据集的基本信息

| 数据集名称 | NetHEPT | Wiki | Epinions | Amazon |
|---|---|---|---|---|
| 点的数量 | 15K | 7K | 75K | 262K |
| 边的数量 | 58K | 103K | 508K | 1,234K |
| 点的平均度 | 3.9 | 14.6 | 6.7 | 4.7 |

### 4.4.3 实验中的方法

本次实验实现了以下算法，并将它们在实验中的运行时间和找到的点集的影响力进行对比分析。

1) 限制性模拟退火算法(Constrained Simulated Annealing，如表4-1所述)，表示为CSA，该方法采用前文提及的目前最好的快速计算点集影响力的方法之一的影响力传播路径方法（ISP）[17]来计算点集影响力；
2) 优化的限制性模拟退火算法，即用公式4-3所示的惩罚函数替代CSA中原惩罚函数得到的新算法，表示为CSA-Q；
3) 适用于节点失效下影响力传播的最大化问题的贪婪算法GreedyB（如章节

4.3.1所述），由于该方法可以找到新问题的可行解，将该方法作为基准方法；

4) 随机选取法，即随机选取 $K$ 个点作为种子点集，该方法是用于对比普通影响力传播的最大化问题中选取影响力最大的点集策略的常用方法之一[17]。为了保证该方法能找到可行解，本文中的随机选取法需要选取到最大影响力丢失不超过影响力丢失阀值 $\eta$ 才能结束，表示为Random。

### 4.4.4 参数设置

本次实验采用被广泛使用的权重级联策略（weighted cascade policy）[3,5-10]设置两点间影响（激活）的概率 $P_{uv}=1/N_{in}(v)$，其中 $N_{in}(v)$ 是节点 $v$ 的入度。为了保证可行解的存在，本实验为影响力丢失阀值 $\eta$ 设定的值域为 $[\min_{A\subseteq V}\sigma(A),\max_{A\subseteq V}\sigma(A)]$，其中 $\min_{A\subseteq V}\sigma(A)$ 和 $\max_{A\subseteq V}\sigma(A)$ 都可以用ISP方法快速估算出来（如章节4.3.2所述）。根据推论4-2，初始化温度设置为 $T_0=2q\times\max\sigma(S)(\max(A)+1)$， $\max\sigma(S)$ 和 $\max(A)$ 同样可以采用ISP方法快速估算出来。参考文献[31]，考虑到 $T_t=T_0/\ln(t+1)$ 收敛速度太慢，不适合实际应用，本次实验采用较优的降温设置 $T_t=0.95^{t-1}T_0$ 以控制CSA降温。接下来，我们讨论CSA其他参数的设定。

如表4-1所示，每个温度下循环次数 $q$ 对CSA的运行时间影响很大。直观地，循环次数 $q$ 越小，CSA运行时间越短，但可能漏选较优点集，即选出的点集的影响力较小；相反，较大的循环次数 $q$ 会保证CSA选出点集的质量，却会导致较大的运行时间。同时， $q$ 也与点集的变化和 $\lambda$ 的更新紧密相关（表4-1中lines 34-35）。理想状态下，即当每个温度中解的更新策略最优时， $q$ 的取值为 $K+1$。而实际情况下，解的更新策略不是最优的，因此，我们需要调查 $q$ 的变化对CSA效果的影响，然后选出 $q$ 的较优取值。

图4-2和图4-3分别展示了在NetHEPT上对CSA和CSA-Q调节参数 $q$ 的选出点集影响力和运行时间的结果，其中其他参数设置为 $\eta=20,K=10,R=2,T_f=10^{-4}$。当 $q=10(K+1)$ 时，CSA和CSA-Q都可以在消耗较合理运行时间下，得到非常好（影响力很高）的解。由于，CSA-Q重用了很多之前步骤中点集影响力计算的结果，所以它的运行速度很快，运行时间曲线都基本和x轴重合，它的运行时间在 $q$ 增大时，变化不大。此外，取更大的 $q$ 对CSA和CSA-Q解质量提高非常小，同时增大了运行时间，因此， $q=10(K+1)$ 是一个非常好的设置，将被用在接下来的实验中。

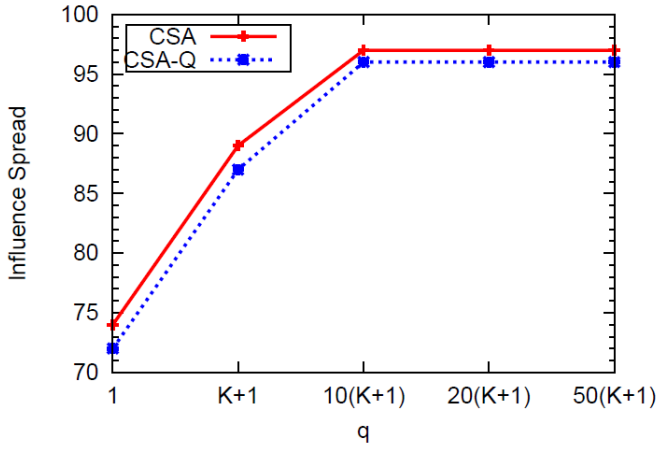


图4-2 在NetHEPT上CSA和CSA-Q调节参数 $q$ 的选出点集影响力的情况

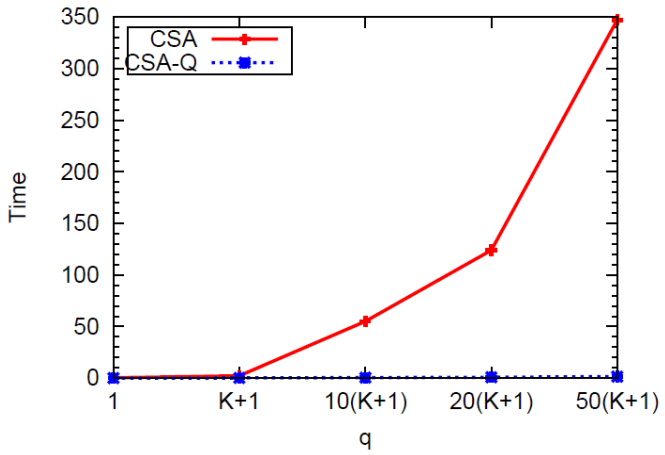


图4-3 在NetHEPT上CSA和CSA-Q调节参数 $q$ 的运行时间情况

参照对于参数 $q$ 调节的实验，我们对CSA终止温度 $T_f$ 做调节实验。根据温度调节函数 $T_t$ 的特性，$T_t$ 的值只能无限接近0，而不可能等于0，所以，终止温度 $T_f$ 越接近于0，CSA外层循环次数越多，即需要遍历更多个温度值，这样会增多CSA的运行时间，也会提高CSA获得的解的质量，这就需要在CSA的运行时间和解的质量上做一个权衡。

图4-4和图4-5分别展示了在NetHEPT上对CSA和CSA-Q调节参数 $T_t$ 的选出点集影响力和运行时间的结果，其中其他参数设置为 $\eta=20, K=10, R=2, q=10(K+1)$。可以观察得到，$T_f=10^{-4}$ 的设置下CSA和CSA-Q都可以在相对少的运行时间下获得质量很高的解。

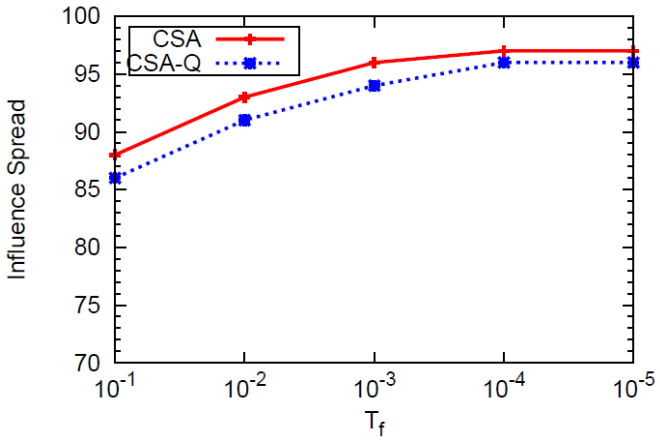


图4-4 在NetHEPT上CSA和CSA-Q调节参数$T_f$的选出点集影响力的情况

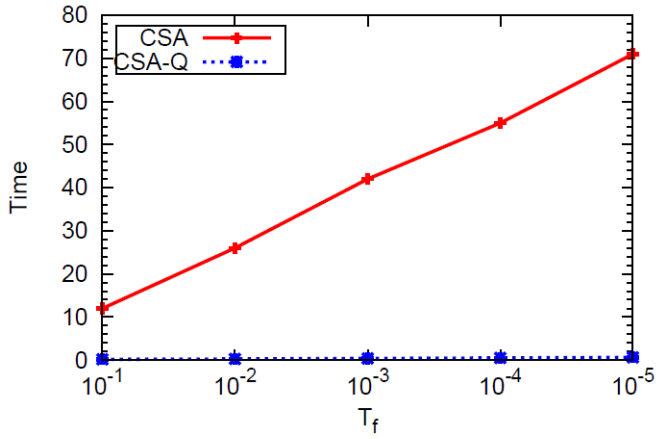


图4-5 在NetHEPT上CSA和CSA-Q调节参数$T_f$的运行时间情况

总之，本节通过实验的方法讨论了调节每个温度下循环次数$q$和终止温度$T_t$的设置。在NetHEPT上，取$q=10(K+1)$和$T_f=10^{-4}$时，CSA和CSA-Q都可以在运行时间和解的质量上得到一个很好的平衡，这些参数设置将会作为后续实验的默认设置。由于其他数据集中这些参数设置的实验结果与NetHEPT上的结果类似，本文不再一一说明。

### 4.4.5 节点失效对影响力丢失的影响

正如前文所述，不同数量的节点失效会导致不同程度的影响力丢失，本次实验在NetHEPT和Epinion两个数据集上（其他数据集效果类似），调查不同数量的节点失效对种子点集影响力丢失的影响。我们首先计算出无节点失效时种子点集（$K$个点）的影响力，然后计算出$R$个节点失效后（来自种子点集）剩下$K-R$个节点组成的点集的影响力，最后可以获得平均影响力丢失的比例

$Ratio = \{\sigma(S) - Ave[\sigma(S-A)]\}/\sigma(S)$，其中 $Ave[\sigma(S-A)]$ 是种子点集 $S$ 中任意组合的 $A$（$A \subseteq S$）点集失效后影响力的平均剩余值。图4-6和图4-7分别展示了在NetHEPT和Epinion两个数据集上，$K=10$，$R$ 取不同值时平均影响力丢失比例的情况。

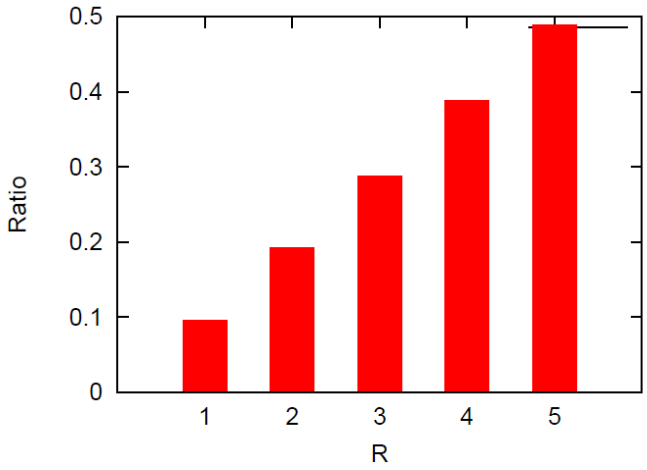


图4-6 在NetNEPT上，节点失效对点集影响力丢失的影响

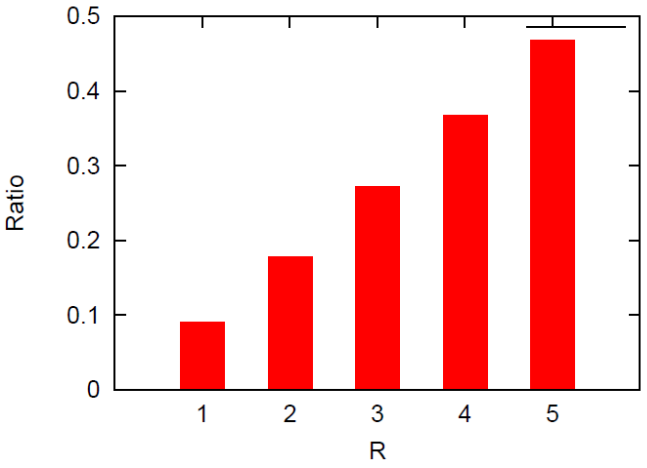


图4-7 在Epinion上，节点失效对点集影响力丢失的影响

从图 4-6 和图 4-7 中可观察到，在两个数据集中，$R$ 的变化对平均种子点集影响力丢失的影响非常大（$R=1$时，种子点集影响力几乎丢掉了 10%；$R=5$时，种子点集影响力差不多丢掉了 50%）。因此，种子点集中节点的失效会导致其影响力大量丢失，在影响力传播的最大化问题中考虑节点失效情况很有必要。

如前文所述，节点失效不仅可能发生在种子点集的节点中，还可能发生在非种子的点集的其他节点中。接下来，我们讨论不同情况的节点失效对问题的影响。

令$TR$为失效节点的总数量，$R$为种子点集中失效节点的数量，则，平均影响力剩余的比例$Ratio = Ave[\sigma(S-A)]/\sigma(S)$，其中$Ave[\sigma(S-A)]$是$TR$个节点失效后（$R$个失效节点来自种子点集$S$，$TR-R$个失效节点来自非种子点集$V-S$）剩余点集影响力的平均值。图 4-8 和图 4-9 分别展示了在 NetHEPT 和 Epinion 两个数据集上，平均影响力剩余比例随来自种子点集失效节点比例$R/TR$的变化情况。在两个数据集中，随着$R/TR$的增大，平均影响力剩余比例迅速减小，当$R/TR=1$时，平均影响力剩余比例都降到 50%左右，可见，来自种子点集的节点失效是导致选出点集影响力丢失的主要原因，所以在影响力传播的最大化问题中考虑种子点集的节点失效很有意义。

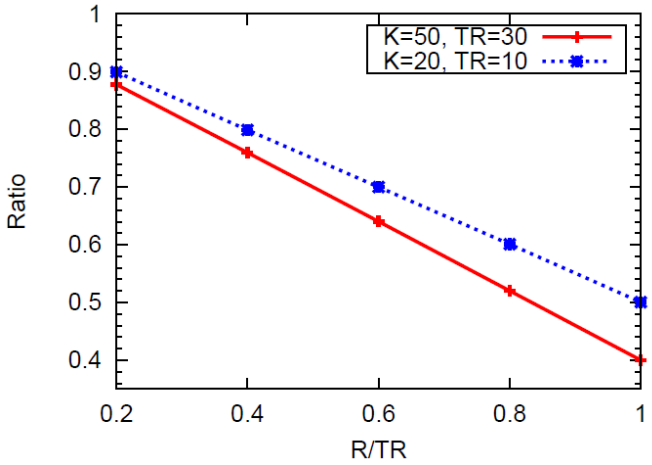


图4-8 在NetNEPT上平均影响力剩余比例随种子点集失效节点比例变化情况

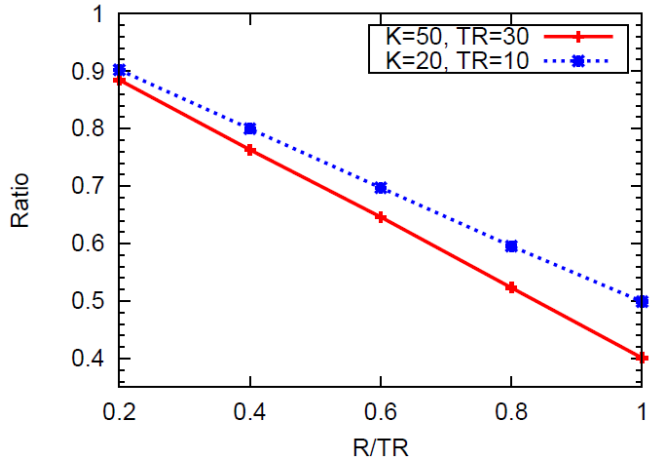


图4-9 在Epinion上平均影响力剩余比例随种子点集失效节点比例变化情况

### 4.4.6 影响力丢失阀值对问题的影响

本节在 4 个数据集上讨论影响力丢失阀值$\eta$对不同算法选出点集的影响力的影响。图 4-10、图 4-11、图 4-12 和图 4-13 分别展示了在数据集 NetNEPT、Wiki、Epinion 和 Amazon 给定$K=10, R=2$时，4 种算法选出的点集影响力的变化情况。较大的$\eta$会放松种子点集中节点失效的限制，这样就会使得那些影响力较大而影响力丢失风险也大的点集可能满足限制条件。正如期望的那样，所有算法选出的点集的影响力都随着$\eta$的增大而增大。

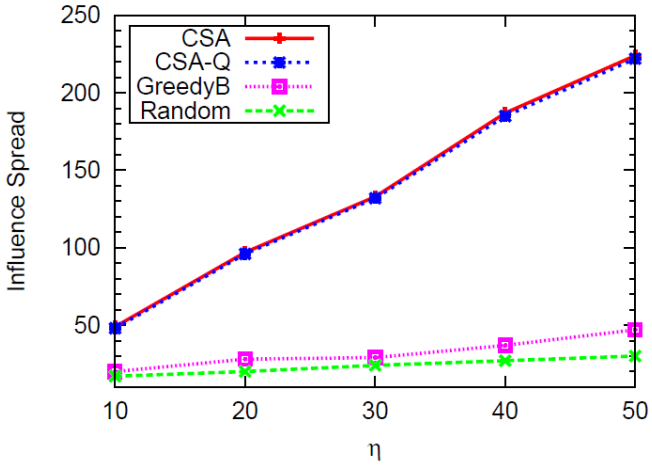


图4-10 在数据集NetNEPT上，影响力丢失阀值$\eta$对不同算法选出点集影响力的影响

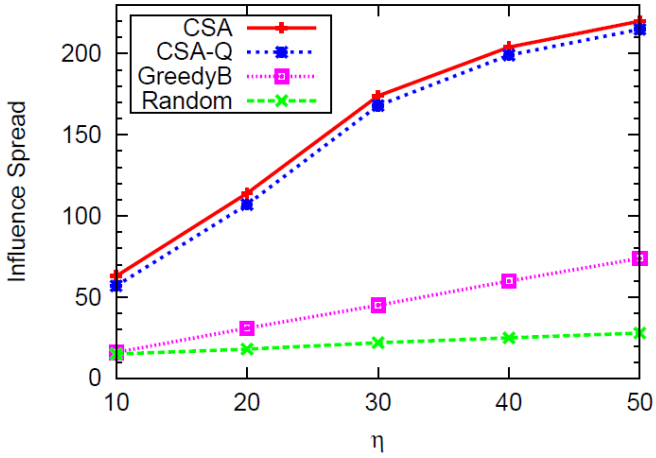


图4-11 在数据集Wiki上，影响力丢失阀值$\eta$对不同算法选出点集影响力的影响

同时，CSA 和 CSA-Q 在所有数据集上都比其他两个算法表现得更好，即选出的点集影响力较大。Random 只是随机选取满足限制条件的点集，它无法保证选出点集的质量（影响力的大小），因此，它选出的点集影响力都很小，而且随$\eta$的增

大变化不大。尽管 GreedyB 也保证了能选择出可行解，但是它在保证解满足限制条件替换解中节点的操作是随机的，也类似于 Random，无法保证选出点集的质量。

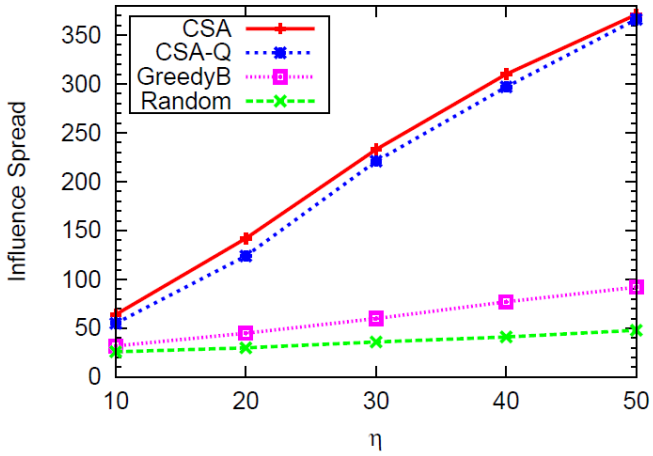


图4-12 在数据集Epinion上，影响力丢失阀值$\eta$对不同算法选出点集影响力的影响

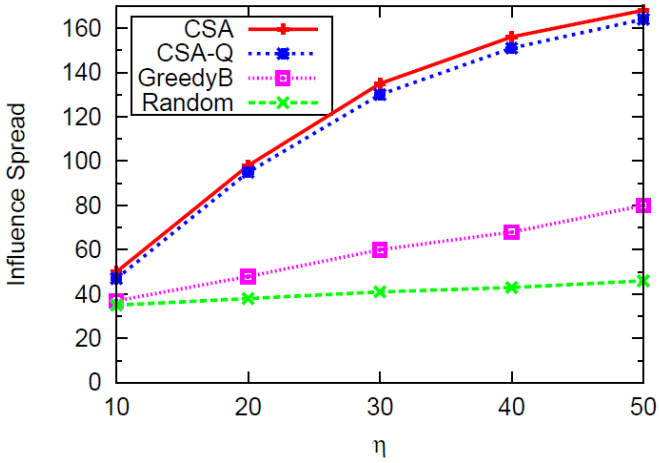


图4-13 在数据集Amazon上，影响力丢失阀值$\eta$对不同算法选出点集影响力的影响

随着$\eta$的不断增大，CSA 和 CSA-Q 选出的点集的影响力不断增长，这是由于$\eta$的增大减小了对于点集的限制条件，使得可行解的范围不断扩大，包含了更多影响力较大的点集且风险较大的点集。另一方面，CSA 和 CSA-Q 选出点集的影响力非常接近，它们的曲线甚至在 NetHEPT 这个数据集上重合了。这也说明了新的惩罚函数估算效果好，保证了 CSA-Q 能选择出足够好的解。

解决普通的影响力传播的最大化问题的 ISP 方法被标记为 ISP-IM。为了验证 CSA 算法框架的有效性，我们选取 Wiki 和 Amazon 数据集，做 CSA，CSA-Q 和 ISP-IM 的对比实验（如图 4-14 和图 4-15 所示，其中$K=10, R=2$）。可以观察出，

在两个数据集上，当$\eta$超过 50 时，CSA 和 CSA-Q 选出的点集影响力很接近 ISP-IM 选出的点集影响力，这个结果一方面说明了当影响力丢失阀值$\eta$非常大时（限制条件不存在），节点失效下影响力传播的最大化问题也就变成了普通的影响力传播的最大化问题，另一方面验证了 CSA 算法框架的有效性，即在普通的影响力传播的最大化问题中同样可以选择出质量较高的解。

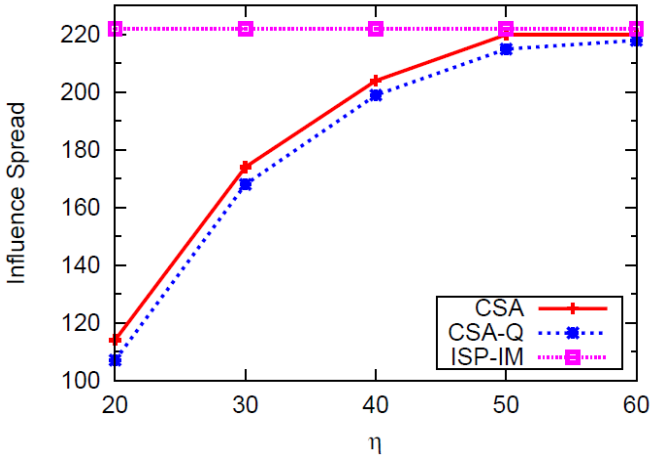


图4-14 在Wiki上，CSA,CSA-Q和ISP-IM选出点集影响力的对比

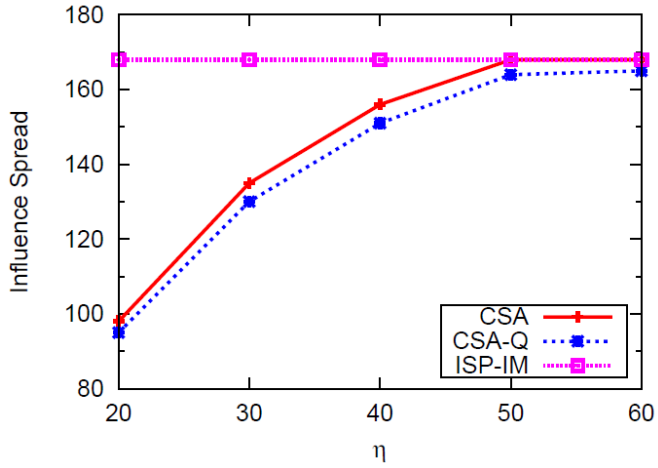


图4-15 在Amazon上，CSA,CSA-Q和ISP-IM选出点集影响力的对比

### 4.4.7 种子点集节点数量和失效节点数量对问题的影响

图 4-16、图 4-17、图 4-18 和图 4-19 分别展示了在数据集 NetNEPT、Wiki、Epinion 和 Amazon 上不同算法选出点集的影响力随失效节点的数量$R$变化的情况，其中$K=20,\eta=40$。如图所示，随着$R$增加，所有算法选出的点集的影响力都会降低。这是由于在种子点集节点数量$K$和影响力丢失阀值$\eta$保持不变，失效节点的

数量 $R$ 的增加意味着每个失效节点带来的影响力丢失必须减少，可行解的范围逐渐缩小到点集影响力较小而点集可能的影响力丢失也更小的点集范围内，因此算法选出的点集的影响力也会不断变小。

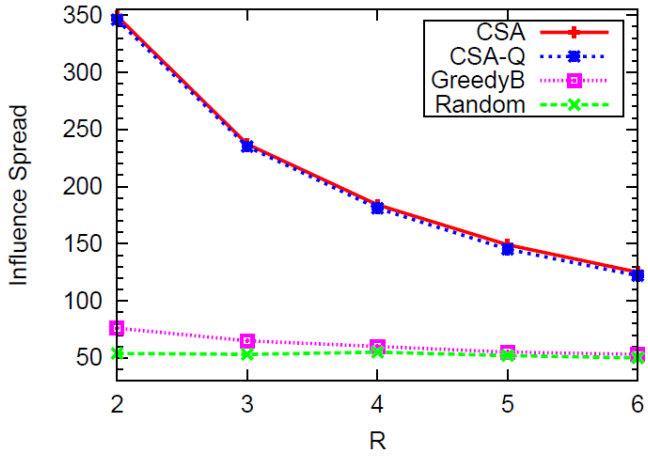


图4-16 在数据集NetNEPT上不同算法选出点集的影响力随失效点集数量 $R$ 的变化情况

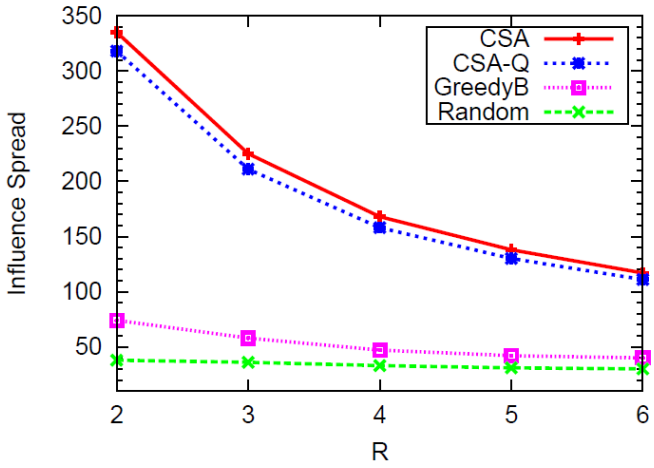


图4-17 在数据集Wiki上不同算法选出点集的影响力随失效点集数量 $R$ 的变化情况

此外，这些结果还说明了在选出的点集的影响力对比上，CSA 和 CSA-Q 都比 Random 和 GreedyB 好很多，而且 CSA 和 CSA-Q 选出的点集的影响力非常接近，尤其是在 NetHEPT 和 Amazon 两个数据集上。

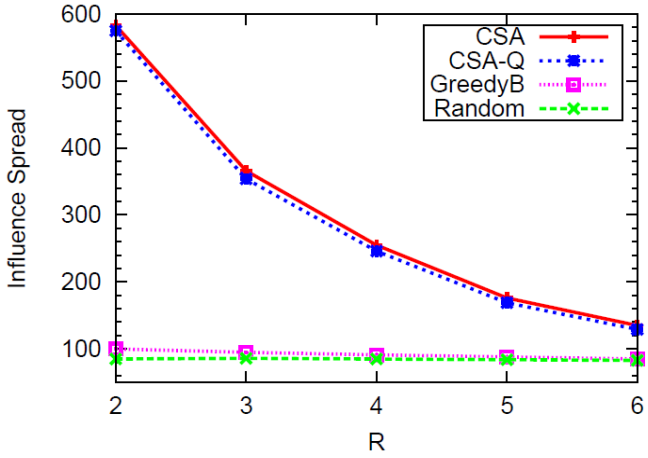


图4-18 在数据集Epinion上不同算法选出点集的影响力随失效点集数量 $R$ 的变化情况

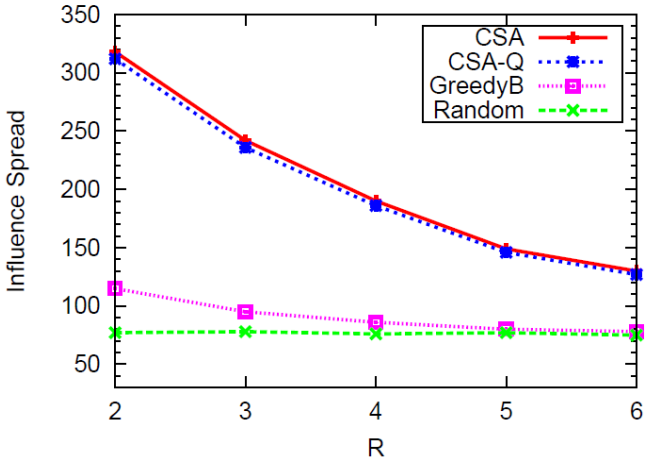


图4-19 在数据集Amazon上不同算法选出点集的影响力随失效点集数量 $R$ 的变化情况

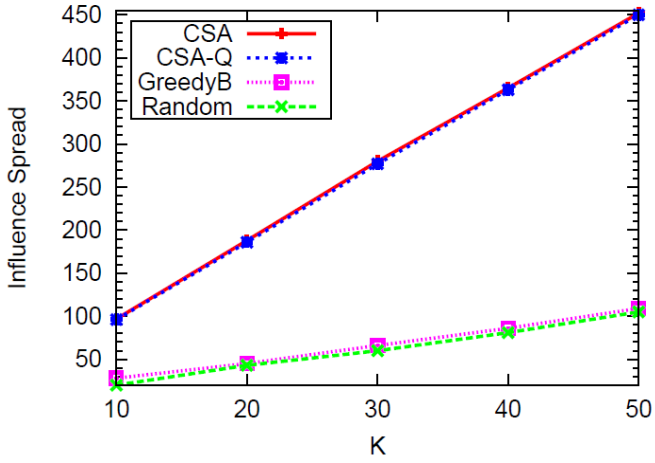


图4-20 在数据集NetNEPT上不同算法选出点集影响力随种子点集节点数 $K$ 的变化情况

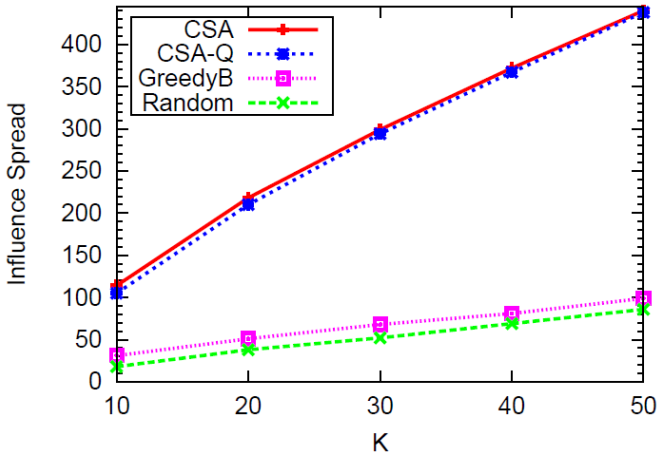


图4-21 在数据集Wiki上不同算法选出点集影响力随种子点集节点数 $K$ 的变化情况

图 4-20、图 4-21、图 4-22 和图 4-23 分别展示了在数据集 NetNEPT、Wiki、Epinion 和 Amazon 上，4 个算法选出点集的影响力随种子点集节点数量 $K$ 变化的情况，其中 $R=2,\eta=20$。这些结果说明，所有算法选出的点集的影响力随着 $K$ 的增大都会增长。主要原因是种子点集节点数量 $K$ 的增大扩大了种子点集的规模使得种子的点集的影响力获得提高。同时，图 4-9 再次说明了，CSA 和 CSA-Q 在选出的点集的影响力上好于 Random 和 GreedyB，而且 CSA-Q 选出点集影响力效果非常接近 CSA 选出的点集。

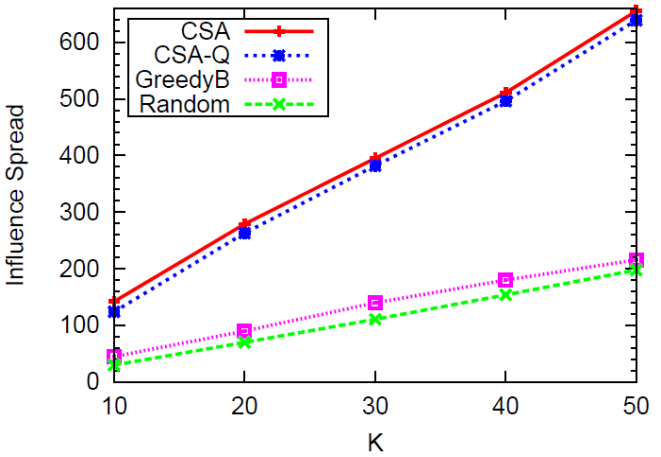


图4-22 在数据集Epinion上不同算法选出点集影响力随种子点集节点数 $K$ 的变化情况

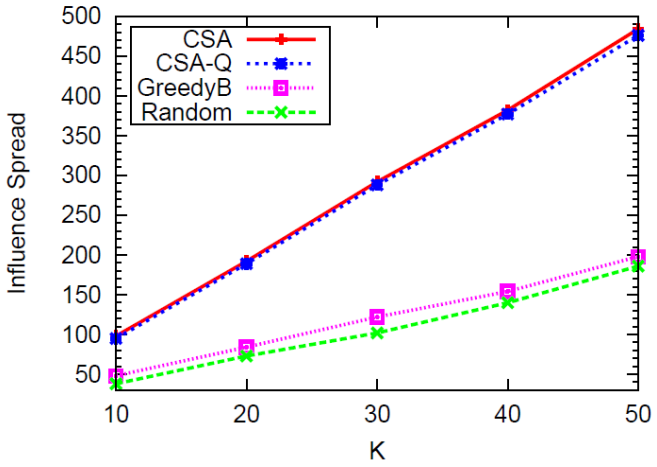


图4-23 在数据集Amazon上不同算法选出点集影响力随种子点集节点数 $K$ 的变化情况

### 4.4.8 算法运行时间的对比

表 4-3 对比了在 4 个数据集中，4 个不同算法在参数 $K, R, \eta$ 变化情况下的运行时间。对于 Random，运行时间很短时用毫秒（ms）或者秒（s）衡量，其他情况用分钟（mins）衡量，对于其他算法，默认都用分钟（mins）衡量，其中运行时间超过 1 天的情况，用*表示。

表4-3 4个数据集中4个算法运行时间的对比

| 数据集名 | $K$ | $R(\eta)$ | Random | GreedyB | CSA | CSA-Q |
|---|---|---|---|---|---|---|
| NetHEPT | 10 | 2(20) | 9ms | 168 | 55 | 0.5 |
| | | 2(30) | 8ms | 168 | 72 | 0.6 |
| | | 2(50) | 8ms | 164 | 125 | 1.6 |
| | | 5(50) | 22ms | 682 | 201 | 0.8 |
| | 20 | 2(20) | 60ms | 1358 | 679 | 2.1 |
| | | 2(40) | 10ms | 1440 | 885 | 3.5 |
| | | 4(40) | 98ms | * | 1305 | 2.4 |
| | | 6(40) | 208ms | * | * | 1.2 |
| | 30 | 2(20) | 259ms | * | * | 4,5 |
| | | 8(80) | 27s | * | * | 5 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | 40 | 2(20) | 525ms | * | * | 5.5 |
| | | 10(100) | 65mins | * | * | 7 |
| | 50 | 2(20) | 980ms | * | * | 8.6 |
| | | 10(100) | 790mins | * | * | 11.4 |
| | | 20(200) | * | * | * | 13.2 |
| Wiki | 10 | 2(20) | 20ms | 512 | 157 | 3.3 |
| | | 2(30) | 18ms | 462 | 242 | 4.1 |
| | | 2(50) | 16ms | 400 | 305 | 6 |
| | | 5(50) | 52ms | 1520 | 271 | 3.8 |
| | 20 | 2(20) | 210ms | * | 954 | 12 |
| | | 2(40) | 165ms | * | 1062 | 19.5 |
| | | 4(40) | 315ms | * | 1398 | 13 |
| | | 6(40) | 615ms | * | * | 8.5 |
| | 30 | 2(20) | 1.5s | * | * | 27 |
| | | 8(80) | 30s | * | * | 34.5 |
| | 40 | 2(20) | 2s | * | * | 36 |
| | | 10(100) | 70mins | * | * | 48 |
| | 50 | 2(20) | 2.8s | * | * | 52 |
| | | 10(100) | 870mins | * | * | 55.3 |
| | | 20(200) | * | * | * | 59.6 |
| | 10 | 2(20) | 33ms | 2850 | 191 | 5.5 |
| | | 2(30) | 30ms | 2772 | 316 | 5.7 |
| | | 2(50) | 26ms | 2560 | 504 | 6.5 |
| | | 5(50) | 81ms | * | 476 | 6.3 |
| | 20 | 2(20) | 500ms | * | 1216 | 7.8 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Epinion | 30 | 2(40) | 386ms | * | * | 11.2 |
| | | 4(40) | 1.2s | * | * | 8.6 |
| | | 6(40) | 2.5s | * | * | 6.3 |
| | | 2(20) | 4s | * | * | 11.5 |
| | | 8(80) | 29s | * | * | 13.8 |
| | 40 | 2(20) | 10s | * | * | 14.6 |
| | | 10(100) | 69mins | * | * | 18 |
| | 50 | 2(20) | 18s | * | * | 18.2 |
| | | 10(100) | 840mins | * | * | 19.8 |
| | | 20(200) | * | * | * | 23.9 |
| Amazon | 10 | 2(20) | 22ms | 616 | 117 | 8 |
| | | 2(30) | 20ms | 600 | 159 | 9 |
| | | 2(50) | 20ms | 582 | 198 | 12.1 |
| | | 5(50) | 55ms | 982 | 462 | 8.8 |
| | 20 | 2(20) | 43ms | * | * | 12.5 |
| | | 2(40) | 33ms | 1250 | * | 17 |
| | | 4(40) | 84ms | * | * | 13.6 |
| | | 6(40) | 158ms | * | * | 11.8 |
| | 30 | 2(20) | 88ms | * | * | 19.4 |
| | | 8(80) | 30s | * | * | 22.5 |
| | 40 | 2(20) | 108ms | * | * | 23.3 |
| | | 10(100) | 70mins | * | * | 27.8 |
| | 50 | 2(20) | 130ms | * | * | 28 |
| | | 10(100) | 870mins | * | * | 32.2 |
| | | 20(200) | * | * | * | 37.5 |

尽管 Random 在大部分情况下运行很快（仅消耗小于 1 分钟的运行时间），但是它选出的点集的影响力非常小（参照前文），因此，Random 不适用于节点失效下影响力传播的最大化问题。此外，Random 同样需要计算选出的点集影响力的最大丢失，所以，当 $K, R$ 取值较大时，Random 需要消耗很多的运行时间，比如，在 NetHEPT 中，$K=50, R=10$ 时 Random 需要运行 790 分钟，这相比于 CSA-Q 仅需要的 11.4 分钟长了很多。

GreedyB 需要运行大量的时间来获得可行解，在很多情况下，它的运行时间甚至超过了 1 天。这是由于 GreedyB 的回朔策略需要置换很多次节点来找到可行解，同时每次置换都需要计算选出的点集影响力的最大丢失，这些操作是非常耗时的。

CSA 和 CSA-Q 的运行时间都比 GreedyB 少，这主要是由于 CSA 框架的系统搜索策略比 GreedyB 的回朔策略好。特别地，CSA-Q 比 CSA 速度快了很多，即使在 $K$ 或者 $R$ 增大的情况下，CSA-Q 运行时间的增长也比 CSA 慢很多，比如，在 Amazon 上，$R=2, \eta=20$ 时，当 $K$ 从 30 增长到 50 时，CSA-Q 的运行时间仅增长了不到 10 分钟。这说明 CSA-Q 中新的惩罚函数对于 CSA 的加速效果非常好。

给定 $K$ 和 $R$（如 $K=10, R=2$），$\eta$ 增长时，Random 和 GreedyB 消耗的时间在减少，而 CSA 和 CSA-Q 消耗的时间在增多。这是由于在放松限制条件时，Random 和 GreedyB 更容易找到可行解，相反，CSA 和 CSA-Q 所需要的循环次数变得更多，运行时间更久。

给定 $K$ 和 $\eta$（如 $K=20, \eta=40$），$R$ 增长时，仅 CSA-Q 运行时间在减少，其他 3 个算法的运行时间在增长。这是因为 $R$ 的增长促使 CSA-Q 去搜索那些影响力较低而影响力丢失风险较小的点集，而这些点集的影响力传播路径相对较少，也就省下来 CSA-Q 中用 ISP 方法计算点集影响力的时间。但是对于 CSA，$R$ 的增长使它不得不计算更多的种子点集中节点失效的可能情况，而这部分增加的时间远远大于省下的计算点集影响力的时间，所以 CSA 的运行时间会增长。至于 Random 和 GreedyB 运行时间的增长，是由于 $R$ 的增长将限制条件提高了，减小了可行解的空间，使得它们需要消耗更多时间去更新点集来找到可行解。

除了 Random 和 CSA-Q，其他两个算法在大部分情况下都不能在一天内获得解。而且 Random 在 $K$ 和 $R$ 比较大的情况下（如 $K=50, R=20$）也不能在一天内找到可行解。相反，CSA-Q 在表中所列的所有情况下都能在 1 小时内找到较优的解，这是因为 CSA-Q 在计算选出的点集影响力的最大丢失时，不是穷举所有节点失效在种子点集中的可能情况，而是重用之前计算出的单点影响力来快速求解选出的点集影响力的最大丢失，这个优化极大地提高了 CSA-Q 的速度。因此，对比

的 4 个算法中，在解决节点失效下影响力传播的最大化问题时，CSA-Q 是最高效的。

### 4.4.9 优化的限制性模拟退火算法的可扩展性

为了进一步说明优化的限制性模拟退火算法（CSA-Q）的可扩展性，本节在更复杂的情况（$K$ 和 $R$ 取值更大）下，在 4 个数据集上对比不同算法选出点集的影响力。由于 CSA 和 GreedyB 在这些复杂情况下耗时超过一天，仅对 CSA-Q 和 Random 选出点集的影响力做对比。如表 4-4 所示，在 CSA-Q 在 4 个数据中选出点集的影响力都远超过 Random。同时，参考表 4-3，可见 CSA-Q 在复杂情况下运行时间也很短。因此，CSA-Q 的可扩展性好于其他 3 个算法。

表4-4 4个数据集中CSA-Q和Random在复杂情况下选出点集影响力的对比

| 算法名称 | $K$ | $R(\eta)$ | NetHEPT | Wiki | Epinion | Amazon |
|---|---|---|---|---|---|---|
| Random | 40 | 10(100) | 90 | 75 | 165 | 158 |
| | 50 | 10(100) | 315 | 283 | 334 | 351 |
| CSA-Q | 40 | 10(100) | 112 | 93 | 196 | 192 |
| | 50 | 10(100) | 377 | 343 | 402 | 456 |

## 4.5 本章小结

本章首先讨论了节点失效对影响力传播的最大化问题的影响，然后标准化定义了节点失效下影响力传播的最大化问题。在分析了常用的贪婪算法不适用于新定义的问题后，提出了解决新问题的限制性模拟退火算法，还证明了该算法的渐进收敛性。为了提高限制性模拟退火算法的速度，结合计算点集影响力公式的子模属性设计了一个新的惩罚函数，并证明了它能保证解的可行性。最后，在 4 个真实数据集中的实验结果验证了提出的方法的高效性。

# 第五章 总结与展望

## 5.1 本文的主要成果

本文主要成果在于分析了新颖性衰变和节点失效对影响力传播的最大化问题的影响，并定义了两个新的问题，即新颖性衰变下影响力传播的最大化问题和节点失效下影响力传播的最大化问题。为了解决这两个新问题，本文分析了这两个问题的性质，然后提出了新的方法和相应的优化，最后通过在真实数据集中的实验结果验证了新方法和优化策略的有效性和高效性。

在新颖性衰变下影响力传播的最大化问题中，本文结合相关的研究工作和真实数据集中的数据分析调查了新颖性衰变对于影响力传播的影响，并得出了可量化分析新颖性衰变的新颖性衰变函数。然后，本文将新颖性衰变函数融合到独立级联模型中，建立了新颖性衰变下影响力传播模型，还在此模型基础上，标准化定义了新颖性衰变下影响力传播的最大化问题。该问题被证明是 *NP-hard* 的，且它的目标函数是非单调和非子模的，因此本文提出了限制性贪婪算法和适用于该算法的动态剪枝优化。接着，由于普通影响力传播模型的计算影响力点集的方法不适用于新颖性衰变下影响力传播模型，本文提出了基于影响力传播路径的方法，还给出改良的迪杰斯特拉算法来快速寻找影响力传播路径。最后，4 个真实数据集上的实验结果验证了贪婪算法、动态剪枝优化及基于影响力传播路径的方法的高效性，还说明了普通影响力传播的最大化问题的方法不适用于新颖性衰变下影响力传播的最大化问题。

在节点失效下影响力传播的最大化问题中，本文首先分析了节点失效对于影响力传播的最大化问题的影响，然后把节点失效下影响力传播的最大化问题标准化定义为一个限制性非线性优化问题。该问题被证明是 *NP-hard* 的，且它的限制条件导致了广泛用于影响力传播的最大化问题的贪婪算法不适用于解决该问题。为此，本文提出了限制性模拟退火算法，还证明了该算法的渐进收敛性。接着，本文利用计算点集影响力公式的子模属性，为限制性模拟退火算法设计了一个新的惩罚函数，极大地提高了该算法的速度。最后，在 4 个真实数据集上的实验详细讨论了限制性模拟退火算法的参数设置，还通过与其他算法对比，说明了限制性模拟退火算法及其新的惩罚函数在解决节点失效下影响力传播的最大化问题的高效性。

## 5.2 下一步的研究工作

新颖性衰变下影响力传播的最大化问题和节点失效下影响力传播的最大化问题的分析和解决，不仅是网络中影响力传播最大化问题的重要研究成果，还推动了具有影响力传播属性的社交网络的其他研究方向的发展。下一步的研究工作将从以下几个方面展开：

1) 对于新颖性衰变下影响力传播的最大化问题，由于网络和节点属性的不同，还可能存在除指数函数形式外，其他形式的新颖性衰变函数，在下一步的研究工作中，可以收集相关数据并分析其他形式的新颖性衰变函数。然后，可以对新颖性衰变下影响力传播模型做新的拓展。同时，也可以在选取影响力最大点集策略和点集影响力计算方法上做进一步研究和优化，以提高这些方法的效率。
2) 对于节点失效下影响力传播的最大化问题，本文主要的贡献点在于提出了一个选取影响力最大点集的策略，即限制性模拟退火算法，下一步的研究工作可以根据节点失效对于影响力传播的影响，设计更高效的点集影响力（或点集影响力丢失）的计算方法。
3) 除了影响力传播的最大化问题外，新颖性衰变广泛存在于具有影响力传播性质的社交网络中，比如，推荐系统中，新产品的新颖性会随着用户被推荐的次数增加而减少。因此，下一步的研究工作可以讨论新颖性衰变在社交网络其他研究方向的影响。
4) 节点失效及其带来的影响力丢失是社交网络中的普遍现象，下一步的研究工作可以进一步探讨在社区探测、影响力传播模型分析等社交网络热点研究问题中节点失效及其带来的影响力丢失的作用。

# 致 谢

转眼间，硕士生涯的三年时间即将结束，这也意味着我在成电七年求学的经历将画上句号。这段欢乐与充实的时光使我的学习和研究能力都得到了极大地提高，并且热爱上了研究。在此，我要衷心地感谢多年来给予我指导、关心、帮助的老师、同学、朋友及亲人们。

首先，我要感谢我的导师向艳萍老师。向艳萍老师在学习、科研和日常生活中都给予了我极大的帮助与耐心的指导，为我的学术研究提供了良好环境，还让我加入到其海外研究合作中。向艳萍老师不仅知识渊博、治学严谨，而且和蔼可亲，更具有极强的敬业精神。向艳萍老师的教诲将成为我一生用之不竭的财富。

其次，我要特别地感谢曾一锋老师。在学术研究中，曾一锋老师不仅给予我非常多的指导与帮助，还给予我很多自由发挥的空间，锻炼了我的各方面的研究能力。曾一锋老师一丝不苟、精益求精的治学态度使我肃然起敬，诲人不倦的教育作风让我获益匪浅。

再次，我要感谢一直以来给予我关心和帮助的其他老师、同学和朋友们。他们的帮助使我生活和学习更加顺利和快乐。

最后，我要感谢我的父母及家人，他们的鼎力支持和无私奉献是我能够顺利完成学业的基石，他们的鼓励和期盼更是我现在乃至将来学习、生活和工作的支柱与动力。

# 参考文献

[1] P. Domingos and M. Richardson. Mining the network value of customers. In KDD, pp. 57–66, 2001.

[2] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In KDD, pp. 61–70, 2002.

[3] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In KDD, pp. 137–146, 2003.

[4] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In KDD, pp. 420–429, 2007.

[5] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In KDD, pp. 199–208, 2009.

[6] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In KDD, pp. 1029–1038, 2010.

[7] Y. Wang, G. Cong, G. Song, and K. Xie. Community-based greedy algorithm for mining top-K influential nodes in mobile social networks. In KDD, pp. 1039–1048, 2010.

[8] Q. Jiang, G. Song, G. Cong, Y. Wang, W. Si, and K. Xie. Simulated annealing based influence maximization in social networks. In AAAI, pp. 127–132, 2011.

[9] K. Jung, W. Heo, and W. Chen. Irie: Scalable and robust influence maximization in social networks. In ICDM, pp. 918–923, 2012.

[10] J. Kim, S.-K. Kim, and H. Yu. Scalable and parallelizable processing of influence maximization for large-scale social networks. In ICDE, pp. 266–277, 2013.

[11] F. Bass. A new product growth model for consumer durables. Management Science, vol. 15, pp. 215–227, 1969.

[12] V. Mahajan, E. Muller, and F. M. Bass. New product diffusion models in marketing: a review and directions for research. Journal of Marketing, vol. 54, no. 1, pp. 1–26, 1990.

[13] W. Chen, Y. Yuan, and L. Zhang. Scalable influence maximization in social networks under the linear threshold model. In ICDM, pp. 88–97, 2010.

[14] A. Goyal, W. Lu, and L. V. S. Lakshmanan. Simpath: an efficient algorithm for influence maximization under the linear threshold model. In ICDM, pp. 211-220, 2011.

[15] M. Gomez-Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. In KDD, pp. 1019-1028, 2010

[16] A. Goyal, F. Bonchi, and L. Lakshmanan. Learning influence probabilities in social networks. In WSDM, pp. 241-250, 2010.

[17] B. Liu, G. Cong, D. Xu, and Y. Zeng. Time constrained influence maximization in social networks. In ICDM, pp. 439–448, 2012.

[18] G. Song, X. Zhou, Y.Wang, and K. Xie. Influence maximization on large-scale mobile social network: A divide-and-conquer method. IEEE Transactions on Parallel and Distributed Systems (TPDS), 2014.

[19] X. Liu, M. Li, S. Li, S. Peng, X. Liao, and X. Lu. Imgpu: Gpu-accelerated influence maximization in large-scale social networks. IEEE Transactions On Parallel and Distributed Systems (TPDS), vol. 25, no. 1, pp. 136–145, 2014.

[20] B. Liu, G. Cong, Y. Zeng, D. Xu, and C. Y. Meng. Influence spreading path and its application to the time constrained social influence maximization problem and beyond. IEEE Transactions on Knowledge and Data Engineering (TKDE), vol. 26, no. 8, pp. 1904–1917, 2014.

[21] W. Chen, W. Lu, and N. Zhang. Time-critical influence maximization in social networks with time-delayed diffusion process. In AAAI, pp. 592–598, 2012.

[22] M. Gomez-Rodriguez and B. Schlkopf. Influence maximization in continuous time diffusion networks. In ICML, 2012.

[23] S. Bharathi, D. Kempe, and M. Salek. Competitive influence maximization in social networks. Internet and Network Economics, pp. 306-311, 2007.

[24] A. Borodin, Y. Filmus, and J. Oren. Threshold models for competitive influence in social networks. Internet and Network Economics, pp. 539-550, 2010.

[25] W. Chen, A. Collins, R. Cummings, et al. Influence Maximization in Social Networks When Negative Opinions May Emerge and Propagate. In SDM, pp. 379-390, 2011.

[26] Y. Li, W. Chen, Y. Wang, and Z. Zhang. Influence diffusion dynamics and influence maximization in social networks with friend and foe relationships. In WSDM, pp. 657-666, 2013.

[27] H. Nguyen, R. Zheng. On Budgeted Influence Maximization in Social Networks. IEEE Journal on Selected Areas in Communications, 2013, 31(6): 1084-1094.

[28] W. Lu and L. V. S. Lakshmanan. Profit Maximization over Social Networks. In ICDM, pp. 479-488, 2012.

[29] G. Li, S. Chen, J. Feng, K. l. Tan, and W. S. Li. Efficient locationaware influence maximization. In SIGMOD, pp. 87-98, 2014.

[30] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of

state calculations by fast computing machines. Journal of Chemical Physics, vol. 21, no. 6, pp. 1087–1092, 1953.

[31] B. W. Wah and T. Wang. Simulated annealing with asymptotic convergence for nonlinear constrained global optimization. In Principles and Practice of Constraint Programming, pp. 461–475, 1999.

[32] J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. ACM Transactions on the Web (TWEB), vol. 1, no. 1, pp. 5, 2007.

[33] G. Ver Steeg, R. Ghosh, and K. Lerman. What stops social epidemics. In ICWSM, pp. 377-384, 2011.

[34] F. Wu and B. A. Huberman. Novelty and collective attention. The National Academy of Sciences, vol. 104, no. 45, pp. 17599–17601, 2007.

[35] K. Lerman and R. Ghosh. Information Contagion: An empirical study of the spread of news on digg and twitter social networks. In ICWSM, pp. 90-97, 2010.

[36] M. Cha, A. Mislove and K. P. Gummadi. A measurement-driven analysis of information propagation in the flickr social network. In WWW, pp. 721-730, 2009.

[37] Yen, J. Y. Finding the k shortest loopless paths in a network. Management Science 17(11):712–716, 1971.

[38] D. Eppstein. Finding the k shortest paths. SIAM Journal on computing 28(2):652–673, 1998.

[39] J. Berry, W. E. Hart, C. A. Phillips, J. G. Uber, and J.-P. Watson. Sensor placement in municipal water networks with temporal integer programming models. Journal of water resources planning and management, 132(4):218-224, 2006.

[40] A. Krause, J. Leskovec, C. Guestrin, J. VanBriesen, and C. Faloutsos. Efficient sensor placement optimization for securing large water distribution networks. Journal of Water Resources Planning and Management, 134(6):516-526, 2008.

[41] J. Xu, M. P. Johnson, P. S. Fischbeck, M. J. Small, and J. M. VanBriesen. Robust placement of sensors in dynamic water distribution systems. European Journal of Operational Research, 202(3):707-716, 2010.

[42] X. He, G. Song, W. Chen, and Q. Jiang. Influence blocking maximization in social networks under the competitive linear threshold model. In SDM, pp. 463-474, 2012.

[43] A. Goyal, F. Bonchi, and L. V. S. Lakshmanan. A data-based approach to social influence maximization. In VLDB, pp. 73-84, 2011.

[44] M. Granovetter. Threshold models of collective behavior. The American Journal of Sociology, 83(6):1420-1443, 1978.

[45] R. Durrett, R. Durrett, R. Durrett, and R. Durrett. Lecture notes on particle systems and percolation. Wadsworth & Brooks/Cole Advanced Books & Software, 1988.

[46] J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. Marketing letters, 12(3):211-223, 2001.

[47] J. Goldenberg, B. Libai, and E. Muller. Using complex systems analysis to advance marketing theory development: Modeling heterogeneity effects on new product growth through stochastic cellular automata. Academy of Marketing Science Review, 9(3):1-18, 2001.

[48] A. Goyal, W. Lu, and L. V. Lakshmanan. Celf++: optimizing the greedy algorithm for influence maximization in social networks. In WWW, pp. 47-48, 2011.

[49] B. Hajian and T. White. On measurement of influence in social networks. In ASONAM, 2012, pp. 101–105.

[50] T.Wang. Global optimization for constrained nonlinear programming. PhD Thesis, University of Illinois at Urbana-Champaign, 2001.

[51] S. Jurvetson. What exactly is viral marketing? Red Herring, 78:110–112, 2000.

[52] P. Bronson. Hotmale. Wired Magazine, 6(12), 1998.

# 攻硕期间取得的研究成果

[1] X. Chen, Y. Zeng, G. Cong, S. Qin, Y. Xiang and Y. Dai. On Information Coverage for Location Category Based Point-of-Interest Recommendation. In AAAI Conference on Artificial Intelligence (AAAI), pp. 37–43, 2015.

[2] Y. Zeng, X. Chen, X. Cao, S. Qin, M. Cavazza and Y. Xiang. Optimal Route Search with the Coverage of Users' Preferences. Accepted by International Joint Conference on Artificial Intelligence (IJCAI), 2015.

[3] S. Feng, X. Chen, G. Cong, Y. Zeng, Y. M. Chee, and Y. Xiang. Influence maximization with novelty decay in social networks. In AAAI Conference on Artificial Intelligence (AAAI), pp. 37–43, 2014.

[4] X. Chen, Y. Zeng, Y.-S. Ong, C. S. Ho, and Y. Xiang. A study on like-attracts-like versus elitist selection criterion for human-like social behavior of memetic mulitagent systems. In IEEE Congress on Evolutionary Computation (CEC), pp. 1635–1642, 2013.

[5] Y. Pan, Y. Zeng, Y. Xiang, L. Sun and X. Chen. Time-Critical Interactive Dynamic Influence Diagram. International Journal of Approximate Reasoning (IJAR), 57: 44-63, 2015.

[6] X. Chen, Y. Zeng, G. Cong, S.Qin and Y. Xiang. Maximizing Influence under Influence Loss Constraint in Social Networks. Submitted to Expert Systems with Applications (ESWA).