

引用格式:刘院英,郭景峰,蒋建伟. 求解影响最大化问题的一种混合算法[J]. 科学技术与工程, 2018, 18(7): 179—184

Liu Yuanying, Guo Jingfeng, Jiang Jianwei. A hybrid algorithm for influence maximization[J]. Science Technology and Engineering, 2018, 18(7): 179—184

# 求解影响最大化问题的一种混合算法

刘院英<sup>1</sup> 郭景峰<sup>2</sup> 蒋建伟<sup>1</sup>  
(河北经贸大学信息技术学院<sup>1</sup>, 石家庄 050061;燕山大学信息科学与工程学院<sup>2</sup>, 秦皇岛 066004)

**摘 要** 影响最大问题定义为在社会网络中寻找  $K$  个初始节点,使得信息传播过程结束后,网络中受影响的节点个数最多。针对传统贪心算法时间复杂度太高的缺点,基于遗传算法提出了 IM\_GA 算法。为了提高算法的运行速度,在计算种子集合的影响范围时,采用其对网络的预期影响力作为衡量指标。为了克服 IM\_GA 算法的早熟现象,在遗传算法中周期性地加入模拟退火算法,提出 IM\_GA\_SA 算法。通过实验验证算法的有效性。

**关键词** 社会网络 影响最大化 传播模型 遗传算法 模拟退火算法

**中图法分类号** TP312; **文献标志码** A

当某人购买某个产品后,他会向自己的朋友推荐,他的朋友还会向朋友的朋友推荐,促使这些人购买这种产品,这就是“病毒式”营销<sup>[1]</sup>。病毒式营销是一种极其重要的营销方式,它能达到一传十,十传百的营销效果。在商家进行营销时,通常会选择一定数量的客户作为初始客户,向这些客户推荐或者免费赠送产品;通过这些初始客户试用后的病毒式营销来达到宣传的目的。为了达到更好的营销效果,需要甄别选择初始客户,这就是社会网络领域研究的影响最大化问题;它定义为在社会网络中寻找  $K$  个初始节点,通过信息在网络中的“病毒式”传播,最终使得网络中受影响的节点个数最多<sup>[2]</sup>。其评价指标为算法运行时间和种子集合的影响范围<sup>[2]</sup>。

Kempe 等<sup>[3]</sup>第一次详细研究了影响最大化问题,证明了该问题是 NP-难问题。他们提出采用自然贪心爬山算法 Greedy 解决该问题;同时证明当传播模型是具有非负性、单调性、子模性时,采用 Greedy 算法得到的解是最优解的  $(1 - 1/e)$  ( $e$  是自然常数)。Leskovec 等<sup>[4]</sup>通过优化 Greedy 算法,提出 CELF 算法。该算法利用传播模型的子模特性,减少了影响范围函数的调用次数。实验表明其速度是 Greedy 算法的 700 倍,但在影响范围上没有取得更好的效果。陈等<sup>[5]</sup>将 IC 模型下的种子影响范围等价于以概率  $(1 - p)$  移除网络中的每一条后种子

的可达节点个数,提出 NewGreedy 算法,然后又结合 CELF 算法和 NewGreedy 算法,提出 MixGreedy 算法。MixGreedy 算法的速度较 Greedy 算法有很大提高,但是影响范围相近。

以上算法存在的共同问题是计算量大,时间复杂度高,不适合用于中大型社会网络。为了解决上述问题,本文提出基于遗传算法以及模拟退火算法的影响最大化方法。

## 1 传播模型

研究社会网络时,通常用图  $G = (V, E)$  来表示网络,其中  $V$  代表所有节点的集合,  $E$  是所有边的集合。节点表示的是网络中的用户,这些用户可以是个人,也可以是团队、集体。边表示的是用户之间的关系,这些关系可以是个人用户之间的亲朋好友关系、敌对关系、论文引用关系、微博上的关注粉丝关系等;也可以是团体之间的合作关系,对抗关系等。边可以有方向的,信息的传播只能从边的起点到达终点;也可以是无向的,信息可以在边的两个顶点之间相互流动。

独立级联模型 IC<sup>[3]</sup> 是社会网络中最基本的信息传播模型,能较好地反映网络中的信息传播情况。在此模型中,假设节点只能处于活跃状态或者不活跃状态,其中活跃状态代表用户购买了某种产品或者接受了某种思想,不活跃状态表示用户未购买某种产品或拒绝接受某种思想。节点只能从不活跃状态变为活跃状态,反之不行。节点一旦活跃,会一直保持。网络中的任意边  $e(u, v)$  都有一个影响概率  $p(u, v)$ ,即节点  $v$  受  $u$  的影响变为活跃状态的概率

为 $p(u,v)$ 。活跃节点对其某一不活跃邻居的激活影响有且仅有一次。

独立级联模型的工作过程如下:在时间步 $t=0$ 时,一批初始节点被标注为活跃状态;当 $t \geq 1$ 时,上一时间步刚被激活的节点会以概率 $p$ 按照随机顺序去影响其所有处于不活跃状态的邻居节点。节点一旦被激活,又会在下一时间步重复上述过程。此过程会一直级联下去,直到没有新节点被激活为止。

2 基于遗传算法的影响最大化方法

Greedy 算法在选择种子过程中,需要进行 $K$ 次全局搜索。每一次搜索过程中,需要计算网络中除种子节点之外的所有节点加入种子集合时增加的影响力,然后选择影响力增量最大的节点加入种子集合。Greedy 算法的时间复杂度太高,为了降低时间复杂度,本文采用遗传算法,一次性地选择所有种子。

遗传算法 GA<sup>[6-9]</sup>是模拟生物在自然环境下进行遗传和进化过程而形成的随机搜索算法。该算法从代表问题可行解的一组初始解开始,按照适应度的大小优胜劣汰,然后通过交叉、变异等操作,逐代演化产生越来越好的近似解。

使用遗传算法时,需要设置染色体的编码方式、选择初始解、计算适应度、设置选择方式及交叉方式和变异方式等。下面结合影响最大化问题,介绍以上步骤的实现方法。

2.1 编码

遗传算法不能直接处理解空间的参数,必须把它们转化为由基因组成的染色体或者个体。染色体(个体)即问题的可行解,编码指的是染色体的表示方式。在本文中,基因用节点表示。由 $K$ 个节点组成的集合表示可行解,表示方法为 $[v_1, v_2, \cdots, v_K]$ 。其中, $v_i$ 表示解中的第 $i$ 个节点,用节点在社会网络图中的编号表示。解中节点的顺序并不影响解的影响范围。假设社会网络 $G$ 中有100个节点,用编号0~99表示,则包含5个节点的某一个可行解可以表示为 $[3, 6, 88, 54, 39]$ 。

2.2 候选种子集合

为了缩小搜索范围,加快算法收敛,需要设置候选种子集合。作为候选种子,节点应该具有较大的影响力。社会网络中存在大量度数为1的节点,这些节点影响力很小,因此,本文将度数大于1的节点组成候选种子集合。

2.3 初始解

初始解群是遗传过程中的第一代可行解组成的集合,可以表示为 $g = \{g_1, g_2, \cdots, g_M\}$ 。在影响最大

化问题中,每个初始解 $g_i$ 由 $K$ 个节点组成,每个节点均从候选种子集合中任意选择。

2.4 适应度

适应度用来表示个体对环境的适应能力,可以用来判断个体的优劣程度。适应度高的个体可获得更大的机会进入下一代,适应度低的个体则会逐渐灭绝。影响最大化问题中,可行解的适应度指的是解中所有节点的影响范围。求节点的影响范围时,需要结合相应的信息传播模型。当采用IC模型作为信息传播模型时,种子集合 $S$ 对所有 $m$ 步可达节点的预期影响力 $f(S)$ 可按式(1)得到。

$$f(S) = \sum_{l=1}^m \sum_{v \in T^l(S)} \left\{ 1 - \prod_{u \in \text{Nin}(v)} [1 - ap(u)p] \right\} \quad (1)$$

式(1)中, $ap(w)$ 为 $w$ 的活跃概率; $\text{Nin}(w)$ 表示 $w$ 的入边邻居集合; $T^l(S)$ 表示从集合 $S$ 出发经过 $l$ 步可达的节点集合; $p$ 为IC模型中边的激活概率;当 $u \in S$ 时, $ap(u) = 1$ 。

当 $m$ 取值为 $\infty$ 时,得到的便是种子集合对网络中所有节点的预期影响力。

2.5 保留最优解

为了保证遗传过程中的最优解不被破坏,在每轮计算了各个解的适应度之后,需要把此轮的最优解及其适应度保留下来,然后与上次得到的最优解进行比较,最终保留适应度大的解。

2.6 选择

选择的目的是把本次遗传过程中的优秀解遗传到下一代,它的操作建立在对解的适应度的评估基础上,即选择适应度大的解,淘汰适应度小的解。在本算法中,为了保证本次最优解能进入下一次遗传过程,首先将适应度最大的解直接复制到下一代。剩下的 $M-1$ 个解,按轮盘赌算法进行选择,每个解的选择概率参照式(2)进行。

$$p_{si} = f_i / \sum_{i=0}^{M-1} f_i \quad (2)$$

式(2)中, $f_i$ 表示第 $i$ 个解的适应度。

2.7 交叉

为了实现遗传过程中基因的重组,通过交叉操作把两个父代个体的某些基因进行互换从而产生新的个体。在本算法中,将 $M$ 个解随机分为 $M/2$ 对,然后在每一对的两个解中任意选择一个节点,以概率 $p_c$ 进行互换。

2.8 变异

变异操作是在交叉操作结束后,对个体中的某些基因进行变动,它的作用是保持个体的多样性。本算法中的变异操作,以概率 $p_b$ 将当前解中的某个节点替换为候选种子集合中的任意一个节点。

根据上述过程,提出基于遗传算法的影响最大化算法 IM\_GA,其具体描述如下。

算法 1 IM\_GA

输入:  $G=(V,E)$ , 种子节点个数  $K$ , 种群个数  $M$ , 交叉概率  $p_c$ , 变异概率  $p_b$ , 遗传终止次数  $r$

输出: 种子集合

```
(1) for node in V do
(2)     { 选择备选种子节点加入集合 C }
(3) 选择初始种群  $g = \{g_1, g_2, \dots, g_M\}$ , 每个  $g_i$  包含  $K$  个节点,  $f_{optimal} = 0$ 
(4) while  $r > 0$  do
(5)     { 令 Sumf 的初值为 0
(6)         for  $i = 1$  to  $M$  do
(7)             { 求解的适应度  $f(g_i)$ 
(8)                 Sumf 累加上  $f(g_i)$  }
(9)         令本次遗传过程最大的适应度为 maxf
(10)        令本次遗传过程适应度最大的解为 maxg
(11)        令  $g'$  为从  $g$  中去掉 maxg 所得集合
(12)        for  $g_i$  in  $g'$  do
(13)            { 计算  $g_i$  的选择概率  $p_i = f(g_i) / \text{Sumf}$  }
(14)        令  $g_{parent}$  为在  $g'$  中选择的  $M-1$  个解
(15)        将 maxg 加入  $g_{parent}$  集合中
(16)        随机将  $g_{parent}$  分成  $M/2$  对, 表示为  $\{(g_{11}, g_{12}), (g_{21}, g_{22}), \dots\}$ 
(17)        for  $i = 0$  to  $M/2$  do
(18)            { 选择  $g_{i1}$  中的一个节点
(19)                选择  $g_{i2}$  中的一个节点
(20)                以概率  $p_c$  进行互换 }
(21)        令交叉后的解集合为  $g_{parent}$ 
(22)        for  $i = 0$  to  $M$  do
(23)            { 随机选择  $g_i$  中的一个节点  $v$ 
(24)                随机从  $C$  中选择一个节点  $u$ 
(25)                以概率  $p_b$  将  $v$  替换成  $u$  }
(26)         $f_{optimal}$  取 maxf 和  $f_{optimal}$  中的最大值
(27)        令  $optimal$  为  $f_{optimal}$  对应的解
(28)         $r = r - 1$  }
(29) 输出最优解  $optimal$ 
```

### 3 求解影响最大化问题的混合算法

遗传算法虽然能实现全局择优,但是通过观察上述遗传算法可以发现,在选择操作时不仅保留了上次的最优解,并且通过采用轮盘赌算法,使得本轮适应度大的解以较大概率在下一轮遗传过程中出现多次,而后面操作过程中的交叉和变异概率较小,所以非常容易导致解集合中的若干个解之间差别很小,这就是遗传算法中的“早熟现象”。

为了克服遗传算法的早熟现象带来的缺陷,在全局搜索更好的解,将模拟退火算法加入到遗传算法中。模拟退火算法<sup>[10]</sup>是基于金属退火原理而建立起来的一种组合优化方法,模拟的是固体降温的物理学过程。

文献[11]中利用模拟退火算法设计了影响最

大化算法 SA,该算法首先在社会网络中随机选择  $K$  个节点组成初始解  $A$  并计算此解的影响范围,然后更换解中的任意节点,得到新解  $A'$ ,并再次计算  $A'$  的影响范围。最后比较两个解的影响范围的大小,如果  $A'$  的影响范围大于  $A$ ,则采用新解,否则以一定概率让新解代替旧解。算法在经过若干次循环后最终得到了最优解。

综上所述,当遗传算法迭代到一定次数时,可以对当时的“最优解”采用模拟退火方法,退火过程结束后,选择遗传算法的“最优解”与模拟退火过程得到的“最优解”中影响范围最大者,代替遗传过程解集合中的部分解进入下一轮遗传模拟退火过程。

根据以上思路,提出求解影响最大化问题的混合算法 IM\_GA\_SA,其具体描述如下所示。

算法 2 IM\_GA

输入:  $G=(V,E)$ , 种子节点个数  $K$ , 种群个数  $M$ , 交叉概率  $p_c$ , 变异概率  $p_b$ , 遗传终止次数  $r$ , 收敛误差

输出: 种子集合

```
(1) 设置程序中用到的各种参数。
(2) 执行一定次数的遗传过程,将得到的最优解设置为  $GA_{optimal}$ 。
(3) 执行模拟退火过程,生成新的最优解  $SA_{optimal}$ 。
(4) 设置  $GA_{optimal}$  和  $SA_{optimal}$  中影响范围最大者为  $X$ 。
(5) 判断  $X$  是否达到收敛值,如果达到,转(6);否则用  $X$  随机替换掉上一轮遗传过程得到的解集中的若干个解,然后转入(2)继续执行。
(6) 输出最优解。
```

### 4 实验与分析

对 IM\_GA 算法及 IM\_GA\_SA 算法进行实验验证,并同已有的影响最大化算法进行比较。实验分别检验了算法在不同传播概率下的影响范围、算法的运行时间。

#### 4.1 实验设计

本文的所有程序代码均采用 python 语言编写,运行计算机的配置为: Intel (R) Core (TM) i5—4590 CPU 3.30 GHz,4 GB 内存。

##### 4.1.1 测试数据集

实验用到了四个真实的数据集。其中,数据集 CA-GrQc、NetHEPT 和 Ca-hepTH 均来源于 <http://www.arXiv.org>,给出的都是论文中作者之间的合作关系。数据集 CA-GrQc 包含的是广义相对论和量子宇宙论部分 1993~2003 年的数据。数据集 NetHEPT 和 Ca-hepTH 包含的都是高能物理——理论部分的数据。这三个数据集中,节点均代表作者,边则表示作者之间的论文合作关系。数据集 Facebook 来源于 <http://snap.stanford.edu/data/>,给出的是社交网站 Facebook 上的数据。四个数据集的统计信



息如表 1 所示。

| 表 1 四个数据集的统计信息 |        |        |       |       |
|----------------|--------|--------|-------|-------|
| 数据集            | 节点数    | 边数     | 平均度   | 最大度   |
| CA-GrQc        | 5 242  | 14 496 | 5.53  | 81    |
| NetHEPT        | 15 235 | 31 399 | 4.12  | 64    |
| Ca-hepTH       | 9 876  | 25 974 | 5.26  | 65    |
| Facebook       | 4 039  | 88 234 | 43.69 | 1 045 |

4.1.2 对比算法

本实验将算法 IM\_GA、IM\_GA\_SA 与混合贪心算法 MixGreedy 及两个启发式算法 Random、Degree 进行对比。在 IM\_GA 和 IM\_GA\_SA 算法中,当采用式(1)计算解的预期影响力时,可通过设置  $m$  为不同值得到种子集合  $S$  对不同距离邻居的影响力。为了验证  $m$  为不同值时所选种子集合的性能,实验中将 IM\_GA 算法中的  $m$  分别设置为最小值 1 和最大值,相应得到  $S$  对最近邻居的影响力和对整个网络的影响力。当  $m$  设置值为 1 时,算法记为 IM\_GA1;当  $m$  设置为最大值时,算法记为 IM\_GAn。算法 IM\_GA\_SA 中,计算解的预期影响力时,设置  $m$  值为 1。另外,将交叉概率  $p_c$  设定为常用值 0.6,将变异概率  $p_b$  设定为常用值 0.1,收敛误差设置为  $10^{-7}$ 。

MixGreedy 是文献[5]中提出的一种改进的贪心算法,该算法的影响范围与贪心算法 Greedy 接近,但是运行速度却比 Greedy 快几百倍。实验中通常设置蒙特卡罗模拟次数  $R$  为 10 000。在此设置

下,在普通计算机上需要运行几十个小时才能得到 50 个种子。为了提高 MixGreedy 算法的运行速度,在本实验中以损失算法的影响范围为代价,设置  $R$  值为 100。Random 算法是一种基本的启发式算法,在此算法中,每次随机选择  $K$  个节点作为种子。Degree 也是一种启发式算法,该算法把数据集中的所有节点按照度数排序,然后选择排序靠前的  $K$  个节点。

4.2 影响范围分析

影响范围指的是信息传播过程结束后,网络中活跃节点的个数。种子集合的影响范围越大,表明算法的效果越好。实验中,计算各个算法得到的种子集合的影响范围时,均采用独立级联模型 IC 作为信息传播模型。由于 IC 模型的随机性,所以采用蒙特卡罗模拟法,设置重复次数为 1 000 次,然后取平均值作为影响范围。另外,应用 IC 模型时需要设置节点间的影响概率  $p$ 。实际应用中,影响概率的设定,可以通过对信息传播过程的历史数据进行挖掘分析得到。在本文的理论研究中,无法从数据集中得到其以往的传播数据,因此,为了一般性,将影响概率  $p$  分别设置为 0.01 和 0.05。

4.2.1  $p$  为 0.01 时的影响范围

图 1 给出了影响概率为 0.01 时,各个算法在数据集 CA-GrQc、NetHEPT、Ca-hepTH、Facebook 上的影响范围。图中横坐标表示种子集合的大小,从 0 递增到 50,递增值为 5,纵坐标表示种子集合的影响范围。图1(a)给出了数据集CA-GrQc上各算法的影响范

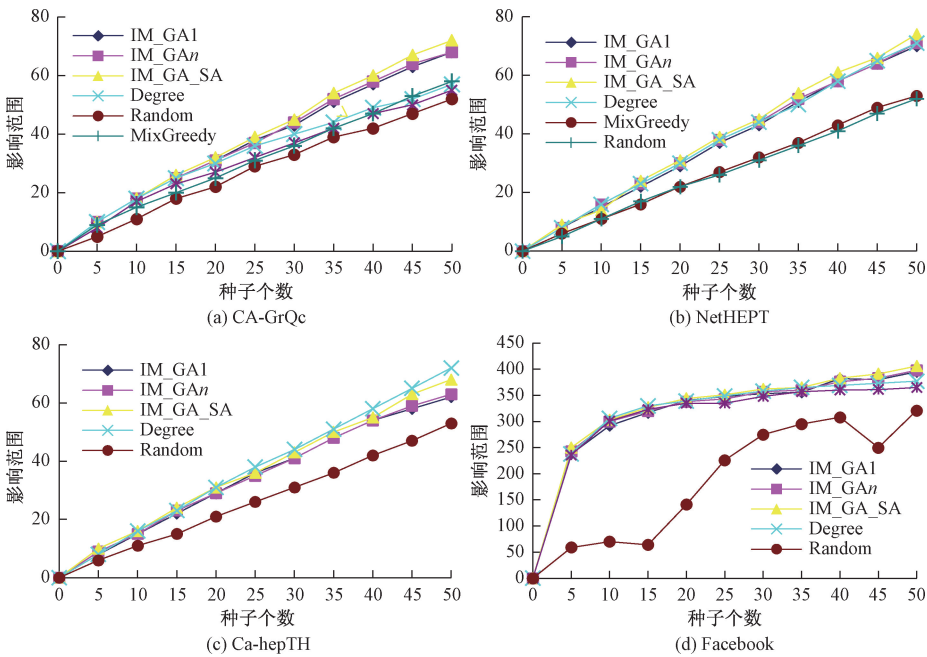


图 1  $p = 0.01$  时四个数据集上的影响范围

Fig. 1 Influence range on the four datasets with  $p = 0.01$

万方数据

围。从图中可见,随着种子个数逐渐增多,IM\_GA-SA 算法的影响范围变为最大,IM\_GA1 和 IM\_GAn 算法的影响范围基本保持一致,次之;Degree 算法的表现比较差;MixGreedy 算法的影响范围非常小;Random 算法的影响范围是所有算法中最小的。

图 1(b)给出了数据集 NetHEPT 上各算法的影响范围。从图中可见,IM\_GA-SA 算法的影响范围是最大的;Degree、IM\_GA1、IM\_GAn 算法的影响范围基本保持一致,次之;MixGreedy 算法和 Random 算法的影响范围非常接近,基本上一直是最小的。

图 1(c)给出了数据集 Ca-hepTH 上各算法的影响范围。由于 MixGreedy 算法运行太慢,所以在该数据集上没有运行此算法。从图中可见,随着种子个数逐渐增大, Degree 算法的影响范围逐渐优于其他算法;IM\_GA-SA 次之;在整个过程中,IM\_GA1 和 IM\_GAn 算法的影响范围基本保持一致;Random 算法的影响范围是最小的。

图 1(d)给出了数据集 Facebook 上各算法的影响范围。在该数据集上同样没有运行 MixGreedy 算法。从图中可见,在种子个数小于等于 35 时, Degree、IM\_GA1、IM\_GAn、IM\_GA-SA 算法的影响范围基本相同。当种子个数大于 35 之后,IM\_GA-SA 算法的影响范围最大,IM\_GA1 和 IM\_GAn 算法的影响范围基本相同,次之;Degree 算法的影响范围更小一些;Random 算法的影响范围一直是最小的。

通过对图 1 进行分析可以得出,本文提出的 IM\_GA 和 IM\_GA-SA 算法是有效的,这是由于这两个算法将  $K$  个种子作为一个整体考虑其影响力,提高了影响范围的精度;并且,由于 IM\_GA-SA 算法克服了 IM\_GA 算法的早熟现象,所以其影响范围大于 IM\_GA1 算法;在影响概率较小时,节点的影响力主要集中在其近邻区域,所以 IM\_GA1 和 IM\_GAn 算法的影响范围接近;启发式算法 Degree 选择的种子节点的影响力有可能叠加,影响范围没有理论保证,所以在四个数据集上的表现不同;MixGreedy 算法的蒙特卡罗模拟次数设置太小,精度受到影响,所以其影响范围很小;Random 算法随机选择节点,没有考虑其影响范围,所以性能一直最差。

4.2.2  $p$  为 0.05 时的影响范围

为了说明不同影响概率对种子集合传播范围带来的影响,在数据集 CA-GrQc 和 NetHEPT 上,验证 IM\_GA1、IM\_GAn、IM\_GA-SA、Degree、Random 等算法选择的种子在影响概率为 0.05 时的传播范围,结果如图 2 所示。图 2 中坐标轴的设置同图 1。

图 2(a)给出的是各个算法在数据集 CA-GrQc 上的影响范围。从图中可见,IM\_GA1、IM\_GAn、

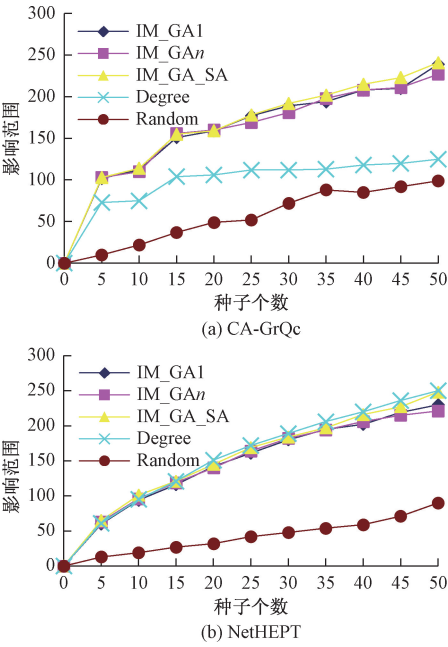


图 2  $p = 0.05$  时两个数据集上的影响范围  
Fig. 2 Influence range on the two datasets with  $p = 0.05$

IM\_GA-SA 算法的影响范围比较接近,其中,IM\_GA-SA 算法的影响范围略大;Degree 算法的影响范围小于上述三个算法,Random 算法的影响范围最小。

图 2(b)给出的是各个算法在数据集 NetHEPT 上的影响范围。从图中可见,随着种子个数的增加, Degree 算法的影响范围逐渐大于其他几个算法;IM\_GA-SA 算法的影响范围大于 IM\_GA1 和 IM\_GAn 算法;IM\_GA1 和 IM\_GAn 算法的影响范围基本相同;Random 算法的影响范围最小。

通过对图 2 进行分析可以得出,当影响概率取较大值时,本文提出的 IM\_GA 和 IM\_GA-SA 算法依然有效;启发式算法 Degree 的影响范围没有理论保证,所以在两个数据集上的表现不同;Random 算法选择种子时没有考虑节点的影响范围,所以性能依然很差。

4.3 运行时间分析

本节测试了不同算法在四个数据集上选择 50 个种子所用时间,实验结果如图 3 所示。需要注意的是,实验中 IM\_GA、IM\_GA-SA 和 MixGreedy 算法中的影响概率取值为 0.01。图 3 中,横坐标给出的是不同的数据集,纵坐标表示算法的运行时间并采用对数坐标。

从图 3 中可见,IM\_GAn 算法的运行时间比 IM\_GA1 算法长,其原因与  $m$  的取值有关。当  $m$  取最大值时,需要计算种子集合对整个网络的影响力,当  $m$  为 1 时,仅需要计算种子集合对最近邻居的影响力,IM\_GA1 算法缩小了计算的规模,使得运行时间减

少。在四个数据集中,IM\_GA\_SA 算法的运行时间均比 IM\_GA1 算法长,原因在于 IM\_GA\_SA 算法中需要循环执行 IM\_GA1 算法。虽然实验过程中设置了蒙特卡罗模拟次数仅为 100,但 MixGreedy 算法的运行时间依然是所有算法中最长的,需要几十个小时。其原因在于 MixGreedy 算法是 Greedy 算法的改进算法,由于 Greedy 算法的时间复杂度极高,所以 MixGreedy 算法的时间复杂度依然很高。启发式算法 Degree、Random 的运行速度非常快,在四个数据集上,它们均在不到 1 s 的时间就能执行完毕。

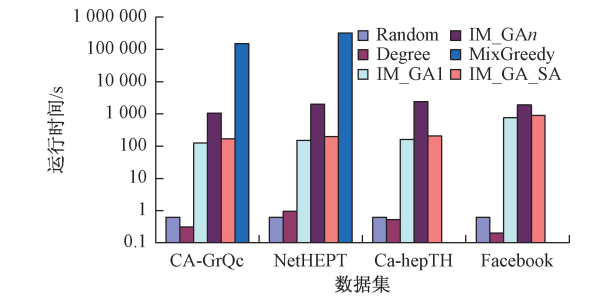


图 3 各算法在四个数据集上的运行时间  
Fig. 3 Running time of each algorithm on the four datasets

5 结 语

将遗传算法的求解过程与影响最大化问题结合在一起,提出基于遗传算法的影响最大化算法 IM\_GA;为了克服 IM\_GA 算法的早熟现象,将模拟算法加入遗传算法中,提出解决影响最大化问题的混合算法 IM\_GA\_SA。在多个数据集上的仿真实验结果验证了所提算法在影响范围和运行时间方面的有效性。

参 考 文 献

1 Mahajan V, Muller E, Bass F M. New product diffusion models in marketing: a review and directions for research. Diffusion of

Technologies and Social Behavior. HK: Springer Berlin Heidelberg, 1991:1—26

2 Domingos P, Richardson M. Mining the network value of customers. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York:ACM, 2001:57—66

3 Kempe D, Kleinberg J. Maximizing the spread of influence through a social network. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York:ACM, 2003:137—146

4 Leskovec J, Krause A, Guestrin C, et al. Cost-effective outbreak detection in networks. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York:ACM, 2007:420—429

5 Chen W, Wang Y, Yang S. Efficient influence maximization in social networks. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Paris: DBLP, 2009:199—208

6 熊志辉, 李思昆, 陈吉华. 遗传算法与蚂蚁算法动态融合的软硬件划分. 软件学报, 2005; 16(4): 503—512

Xiong Zhihui, Li Sikun, Chen Jihua. Hardware/software partitioning based on dynamic combination of genetic algorithm and ant algorithm. Journal of Software, 2005; 16(4):503—512.

7 Deb K. A fast elitist multi-objective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 2000; 6(2):182—197

8 梁亚澜, 聂长海. 覆盖表生成的遗传算法配置参数优化. 计算机学报, 2012; 35(7): 1522—1538

Liang Yalan, Nie Changhai. The optimization of configurable genetic algorithm for covering arrays generation. Chinese Journal of Computers, 2012; 35(7):1522—1538

9 Shiba T, Tsuchiya T, Kikuno T. Using artificial life techniques to generate test cases for combinatorial testing. Computer Software and Applications Conference, 2004. New York IEEE, 2004:72—77

10 Carlier J, Pinson E. An algorithm for solving the job-shop problem. Informs, 1989;35(2):164—176

11 Jiang Q, Song G, Cong G, et al. Simulated annealing based influence maximization in social networks. AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2011:127—132

A Hybrid Algorithm for Influence Maximization

LIU Yuan-ying<sup>1</sup>, GUO Jing-feng<sup>2</sup>, JIANG Jian-wei<sup>1</sup>

(College of Information Technology, Hebei University of Economics and Business<sup>1</sup>, Shijiazhuang 050061, China;  
College of Information Science and Engineering, Yanshan University<sup>2</sup>, Qinhuangdao 066004, China)

[Abstract] Influence maximization is the problem of finding a small subset of  $K$  nodes in a social network that could maximize the number of nodes influenced. Aiming at the problem in the traditional Greedy algorithm with high time complexity, an IM\_GA algorithm is proposed based on the genetic algorithm. To improve the running speed, expected influence spread is used as the measurement in the computation of seed set's influence range value. In order to avoid precocious phenomena, annealing algorithm joining simulated to genetic algorithm periodicaly, An IM\_GA\_SA algorithm is put forward. The effectiveness of the algorithms is verified by experiments.

[Key words] social network influence maximization diffusion model genetic algorithm simulated annealing algorithm