软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2019,30(3):573−588 [doi: 10.13328/j.cnki.jos.005682]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于事件的社交网络上的双边偏好稳态规划*

成雨蓉[1], 王国仁[1], 李博扬[2], 袁 野[2]

[1](北京理工大学 计算机学院,北京 100081)

[2](东北大学 计算机科学与工程学院,辽宁 沈阳 110819)

通讯作者: 王国仁, E-mail: wanggrbit@126.com

摘 要: 在基于事件的社交网络中,一个经典的问题是为用户规划其感兴趣的事件.现有的工作仅仅考虑用户的喜好,仅从用户的角度出发,为其安排尽可能感兴趣的事件来参加.然而,从事件主办者的角度出发,他们亦希望为事件安排的用户尽可能有更大的影响力,用户的可靠性尽可能高,以保障事件能够顺利开展,并取得预期的效果.本质上来说,基于事件的社交网络上的规划问题是一个双向选择的问题,而现有的所有工作均未从用户和事件的双边偏好考虑问题.因此,提出一种双边偏好稳态规划问题来解决这种双向选择问题.该问题首次提出,因此现有工作中未有相关算法可供解决该问题.对比之前只考虑用户偏好的规划,在考虑用户和事件双边偏好时,面临着问题更复杂、约束条件更多的困难.因此,提出两种基础算法和一种改进算法来高效、高质量地解决这个问题,并用大量的实验验证所提出算法的高效性和有效性.

关键词: 基于事件的社交网络;双边偏好;稳态规划

中图法分类号: TP311

中文引用格式: 成雨蓉,王国仁,李博扬,袁野.基于事件的社交网络上的双边偏好稳态规划.软件学报,2019,30(3):573−588. http://www.jos.org.cn/1000-9825/5682.htm

英文引用格式: Cheng YR, Wang GR, Li BY, Yuan Y. Bilateral preference stable planning over event based social networks. Ruan Jian Xue Bao/Journal of Software, 2019,30(3):573−588 (in Chinese). http://www.jos.org.cn/1000-9825/5682.htm

Bilateral Preference Stable Planning over Event Based Social Networks

CHENG Yu-Rong[1], WANG Guo-Ren[1], LI Bo-Yang[2], YUAN Ye[2]

[1](School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China)

[2](School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China)

Abstract: In event based social networks (EBSNs), a typical problem is to plan interested events to users. Existing work only considers the users' preference to events, and plans the events that they are most possibly interested in. However, from the view of event holders, they also hope the users that are assigned to their events are with high influence and reliability. Consequentially, their events can be held successfully and achieve expected effects. Essentially, the planning problem over EBSNs is a bilateral selection problem. However, existing studies never consider the bilateral preference between events and users. Thus, this study proposes a bilateral preference stable planning problem to solve this bilateral selection problem. Since this study is the first to propose the bilateral preference planning problem, no existing algorithms can solve it. Compared with the existing planning problem which only considers the preference of users, the bilateral preference stable planning problem is more complex and contains more constraints. Thus, two baseline algorithms and two

* 基金项目: 中国博士后科学基金(2018M631358); 国家自然科学基金(61332006, 61332014, 61328202, U1401256, 61572119, 61622202); 中央高校基础科研业务费(N150402005)

Foundation item: China Postdoctoral Science Foundation (2018M631358); National Natural Science Foundation of China (61332006, 61332014, 61328202, U1401256, 61572119, 61622202); Fundamental Research Funds for the Central Universities (N150402005)

本文由智能数据管理与分析技术专刊特约编辑樊文飞教授、王国仁教授、王朝坤副教授推荐.

收稿时间: 2018-07-17; 修改时间: 2018-09-20; 采用时间: 2018-11-01

improved algorithms are proposed to efficiently and effectively solve this problem. Finally, extensive experiments are conducted to verify the efficiency and effectiveness of the proposed algorithms.

Key words: event based social network; bilateral preference; stable planning

近年来,基于事件的社交网络(event based social network)正逐步走入人们的生活.人们越来越喜欢用基于事件的社交网络平台来安排空闲时间,以参加一些自己感兴趣的活动(即事件),从而充实自己的业余生活[1].常见的基于事件的社交网络平台有 Meetup(http://www.meetup.com/),Plancast(http://plancast.com/)等.以 Meeup 为例,如图 1 所示:图 1(a)为 Meetup 首页,陈列用户所在位置附近正在举办或即将举办的事件;图 1(b)为其中一个事件的简介,包括事件的描述、参与人、举办人、举办地、举办时间等信息.可见基于事件的社交网络可以为用户提供一种从线上到线下(online to offline)的服务,允许用户在线上创建、管理以及加入不同的事件,并为用户制定个性化的线下参与事件的计划.截至 2018 年,仅 Meetup 一家网站就已经拥有 1 600 万用户,平均每个月举办 30 万个事件[2].

(a) Meetup 首页 (b) 事件简介及基本信息 (c) 用户基本信息

Fig.1 Homepage of Meetup and a brief introduction of an event

图 1 Meetup 首页、事件基本信息及用户基本信息

在基于事件的社交网络中,一个经典的问题是为用户安排其感兴趣的事件[2−6].用户在注册时会被要求选择自己的兴趣爱好,例如体育、音乐、编程等.如图 1(c)“Interests”栏中所有的标签所示.每个事件在创建时,亦会要求利用一些标签对其进行相应的描述.如图 1(b)“We’re about”栏中所有的标签所示.因此,可以用效用值(utility score)来衡量每个用户对每个事件的感兴趣程度,其计算方式为用户兴趣标签栏和事件描述标签栏中两组标签的相似度[2−6].该效用值越高,表示用户对相应的事件越感兴趣.此外,在为用户安排事件的时候,还应该考虑二者位置之间的距离.为此,用户可以提供一个行程预算值(travel budget),使得系统可以为其安排预算值之内的事件来参加.更进一步,当基于事件的社交网络平台为用户指定计划时,还应考虑到用户可能参加多个事件,这些事件举行的时间和地点彼此之间不应有冲突.平台整体的规划目标为使所有计划中的效用值之和最大.下面通过例 1 具体阐述现有工作[2−6]中的事件规划问题.

例 1:表 1 列出了基于事件社交网络中的每个用户对每个事件的效用值,以及每个事件的举办时间.

Table 1 Users’ utility to each event

表 1 用户对每个事件的效用值

	$u_1(16)$	$u_2(30)$	$u_3(17)$	$u_4(22)$	$u_5(25)$	举办时间
$e_1(1)$	0.9	0.1	0.5	0.5	0.1	10:00~12:00
$e_2(2)$	0.5	0.5	0.1	0.9	0.9	15:00~16:00
$e_3(3)$	0.1	0.9	0.9	0.1	0.5	10:30~12:05

表 1 中事件 e_1 与 e_3 的举办时间有重叠,可见两个事件是冲突的,不可被同时安排到同一个用户的计划中.表中用户后边附加的括号表示该用户的行程预算值,每个事件后边附加的括号表示每个事件参与人数的上限.所有用户的所在位置和所有事件的举办地如图 2 所示.图中坐标表示其经纬度,简单起见,这里用自然数表示.任

意两个位置之间的距离以欧式距离计算.一个用户的计划是其从当前位置出发,按照事件举办时间的顺序依次经过各个事件举办地,最后回到用户所在位置的过程.例如:用户 u_2 的计划即从 u_2 所在地出发,依次经过 e_3 和 e_2 所在的位置,最后回到 u_2 的初始所在地,如图中紫色的折线所示.在该计划中,用户 u_2 的总体距离开销为 $\sqrt{3^2+3^2}+\sqrt{11^2+1^2}+\sqrt{8^2+2^2}=\sqrt{18}+\sqrt{122}+\sqrt{68}\approx 23.53$,小于其预算值 30.图 2 中所有折线表示计划是文献[2−6]中的算法得出的规划方式,其总体效用值为 0.9+0.5+0.9+0.9+0.9+0.5=4.6.

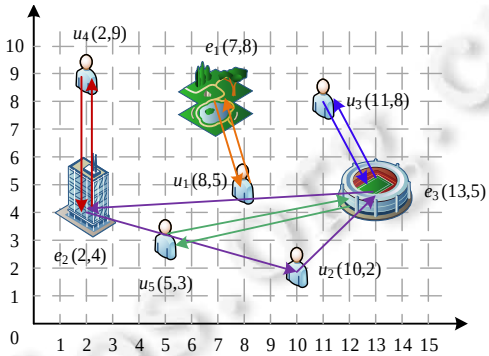


Fig.2 Location of users and events

图 2 用户所在位置及事件举办地点

现有的工作在为用户制定计划的过程中均忽略了一个重要的问题,那就是作为事件举办者,他们对不同质量的用户也是有偏向性和选择性的.例如,有些用户经常缺席安排给他们的事件,这些经常缺席的用户会严重影响事件举办的效果.以一个参观景区的事件为例,如果原计划中的用户大量缺席,会使得按时来参加该事件的用户无法享受原定的折扣景区票价,从而降低按时参加事件的用户的参与满意度,影响事件举办者的信誉.久而久之,会影响平台上事件举办者和参与者双方对于参与事件的积极性,造成平台的用户流失.另一方面而言,对于事件举办者来说,他们也更喜欢请在社交媒体中影响力更大的用户来参加他们的事件,从而扩大事件的影响力,促进事件长久高质量地举办下去.因此,基于事件的社交网络上规划问题本质上应该是一个用户和事件双边选择的问题,所安排的计划应该令事件举办者和用户双方都尽量感到满意,而现有的工作恰恰忽略了这一点.如果例 1 中的事件对用户的偏好效用值见表 2,则图 2 所示的计划中,安排给事件 e_1 的用户 u_1 是其效用值最低的用户;安排给事件 e_2 的用户 u_2 和 u_4 是其效用值最低的两个用户;而安排给事件 e_3 的用户除 u_5 外,另两个用户 u_2 和 u_3 也是 e_3 效用值最低的两个用户.可见,用现有工作[2−6]中的方法无法兼顾事件和用户双方的偏好效用.

Table 2 Events' utility to each user

表 2 事件对每个用户的效用值

	e_1(1)	e_2(2)	e_3(3)
u_1(16)	0.1	0.5	0.5
u_2(30)	0.3	0.3	0.1
u_3(17)	0.5	0.9	0.3
u_4(22)	0.7	0.1	0.7
u_5(25)	0.9	0.7	0.9

根据上文所述,在基于事件的社交网络规划中,考虑事件与用户双边偏好比只考虑用户的单边偏好的规划在实际应用中更合理.因此,如何定义一种新的能够考虑用户与事件双边偏好的规划问题,是本文面临的挑战之一.另外,由于现有工作从未考虑过此类双边偏好的规划问题,因此现有的考虑单边规划的算法能够提供的参考性有限.而考虑双边偏好的规划显然比只考虑单边偏好的规划问题更复杂,约束条件更多,从而更加难以解决.为此,本文提出了基于事件的社交网络上的双边偏好稳态规划问题,可以同时兼顾用户和事件双方对彼此之间的偏好效用,弥补现有工作的不足.本文提出了两种基础算法和一种改进算法,以解决该问题.最后,通过一些真实数据集上的实验,验证了本文提出方法的有效性和高效性.

本文第 1 节对基于事件的社交网络上的双边偏好稳态规划问题进行定义.第 2 节对相关工作加以总结.第 3 节详细介绍两种基础算法,并分析其正确性和复杂度.第 4 节分析基础算法的不足,并提出一种改进的方法.第 5 节通过真实数据集上的实验验证本文提出算法的高效性和有效性.第 6 节对全文加以总结,并提出对未来工作的设想.

1 问题定义

本节给出基于事件的社交网络上的双边偏好稳态规划问题的正式定义.设基于事件的社交网络平台上的用户集为 $U=\{u_i\}$,共包含 n 个用户;事件集为 $E=\{e_j\}$,共包含 m 个事件.每个用户 $u_i\in U$ 以二元组 $\langle \boldsymbol{l}_{u_i},B_i\rangle$ 表示,其中,$\boldsymbol{l}_{u_i}$ 为向量,表示 u_i 的当前位置,B_i 为 u_i 的行程预算值.每个事件 $e_j\in E$ 以四元组 $\langle \boldsymbol{l}_{e_j},\eta_j,t_j^s,t_j^t\rangle$ 表示.其中,$\boldsymbol{l}_{e_j}$ 为向量,表示事件 e_j 的举办地;η_j 为事件 e_j 参与人数的上界,即事件 e_j 最多可以被分配多少人;t_j^s 表示事件 e_j 举办的开始时间,t_j^t 表示结束时间.对于每一个用户 u_i 而言,他对每个事件 e_j 的感兴趣程度,以一个效用值 $p_u(u_i,e_j)$来表示;对于每个事件 e_j 而言,它对平台上的每个用户 u_i 亦有一个选择偏好,以一个效用值 $p_e(e_j,u_i)$来表示.$p_u(u_i,e_j)$,$p_e(e_j,u_i)\in[0,1)$.如果 $p_u(u_i,e_j)=0$,表示该用户对该事件丝毫不感兴趣或无法参加该事件;如果 $p_e(e_j,u_i)=0$,表示事件 e_j 的举办者已拉黑用户 u_i.本文中,设定 u_i 对每个事件的偏好顺序是严格的,即没有两个效用值是相同的;同理,e_j 对每个用户的偏好顺序也是严格的.

当基于事件的社交网络平台为用户制定计划时,必须在一个特定时间段内为用户制定预计的行程计划.为了简单起见,本文假设该时间段为 1 天,即,本文所制定的计划是为用户提前制定一天的计划.

设全局计划$\mathcal{P}$是为每个用户制定一天个性化计划的集合,即$\mathcal{P}=\{P_i:P_i\subseteq E,1\leqslant i\leqslant n\}$.每个用户的计划中不能包含彼此冲突的事件,也就是说,如果 P_i 中的一个事件 e_k 在另一个事件 e_h 之前开始,那么 e_k 也要在 e_h 开始之前结束.例如在例 1 中,事件 e_1 和 e_3 就是冲突的,因为 e_3 在 e_1 还没有结束的时候就已经开始了.如果事件 e_j 被匹配给了用户 u_i,记为 $M(u_i,e_j)$;此匹配中,$M(u_i)=e_j$,$M(e_j)=u_i$,u_i 对 e_j 的偏好记做 $p_M(u_i)$,e_j 对 u_i 的偏好,记做 $p_M(e_j)$.

在某个用户的计划中,可能会有多个事件,那么这个用户参加该计划中的所有事件的总行程开销 D_i 是参加每个事件的行程之和.本文使用欧拉距离计算各个位置之间的距离.

定义 1(阻塞对). 对于一对用户或事件(u_i,e_j)中,如果 u_i 和 e_j 不在$\mathcal{P}$中,而 u_i 比起$\mathcal{P}$中的 $M(u_i)$更喜欢 e_j,而 e_j 比起$\mathcal{P}$中的 $M(e_j)$更喜欢 u_i,那么就称(u_i,e_j)为$\mathcal{P}$中$(u_i,M(u_i))$和$(M(e_j),e_j)$的阻塞对.

例如在例 1 中,如果 $M(u_3,e_3)$在$\mathcal{P}$中,根据表 1 和表 2,那么(u_5,e_2)就是一个阻塞对,因为 u_5 比起计划中的 e_3 更喜欢 e_2,而 e_3 比起计划中的 u_3 也更喜欢 u_5.

基于以上预备知识,现在对基于事件的社交网络上双边偏好稳态规划问题进行定义.

定义 2(基于事件的社交网络上双边偏好稳态规划问题). 在基于事件的社交网络平台上,双边偏好稳态规划问题是指找到一个满足下列适当的全局计划$\mathcal{P}$.

(1) 用户的计划中没有彼此冲突的事件,即 $\forall i\forall e_k\neq e_h\in P_i,t_{e_k}^s<t_{e_h}^s\Rightarrow t_{e_k}^t<t_{e_h}^t$;

(2) 用户的行程开销要在其预算之内,即$\forall i,D_i\leqslant B_i$;

(3) 事件的参加人数要少于其人数上界,即$\forall j|\{P_i:e_j\in P_i\}\neq\eta_j$;

(4) $\mathcal{P}$中不存在阻塞对.

根据定义 **2** 中的表述,最终的规划中,用户和事件双方不会同时更喜欢另一个事件和用户,会达到一个稳定的状态.从经济学角度看,在存在竞争的关系中,稳定平衡的状态是最合理的,也是最符合事物发展规律的[7].

2 相关工作

本节从 3 个角度总结与本文相关的现有工作:(1) 基于位置的社交网络;(2) 基于事件的社交网络;(3) 稳定婚姻问题及其变种.

2.1 基于位置的社交网络

近年来,从线上到线下(online to offline)的服务逐步走入人们的日常生活,其中最火热的一类服务就是基于位置的社交网络[8−17].基于位置的社交网络服务的现有工作也会推荐或安排用户到各个事件(或地点,如旅店、商场等)中去,但这些工作更关注如何能够使用户各自的效用值最大,即面向用户个人的推荐.而基于事件的社交网络更关注于平台上所有用户和事件的总体效用值,即基于事件的社交网络工作更关注于令平台上所有用户对计划都很满意.更进一步而言,现有的基于位置的社交网络上的工作从未涉猎过事件与用户之间的双向选择问题,仅仅是关注用户单方面的偏好.

2.2 基于事件的社交网络

基于事件的社交网络首先由 Liu 等人在文献[1]中分析了来自于两个著名的基于事件的社交网络平台——Meetup 和 Plancust 数据的特征,并提出了基于事件的社交网络数据模型及其相应的性质.Zhang 等人[14]和 Du 等人[15]利用机器学习的方法,根据基于事件的社交网络平台上的历史数据,为用户推荐相关的事件.Pham 等人在文献[16]中提出了一种通用的异构图模型表示基于事件的社交网络数据,并提出了一种通用的训练方式解决基于事件的社交网络上的 3 种推荐问题,即:为用户推荐事件群组、为事件群组推荐标签以及为事件推荐用户.以上工作也均为关注用户的个性化、个体化推荐,而不是平台的全局规划.此外,Feng 等人在文献[17]中提出了一个找到最有影响力的覆盖集问题,即找到 k 用户,他们所拥有技能的并集能够覆盖一个技能要求集合,并组织一个能够使得在社交网络上影响力最大的事件.从本质上来说,该问题是解决最大影响力问题(influence maximization problem)[18]和团队构成问题(team formation problem)[19]所构成的综合问题.

一组更相关的基于事件的社交网络的工作是关于社交事件组织问题(social event organization problem)[5]及其变种,它是指为一组用户安排一系列事件,并能够保证整个平台的效用值是最大的.She 等人在文献[3,4]中考虑了事件之间的冲突情况,并做出了规避了冲突的事件安排;在文献[6]考虑了用户的行程预算问题,并从简单将用户和事件匹配起来的问题扩展到为用户安排参与所有事件的合理行程.Cheng 等人在文献[2]中进一步考虑了事件的参与人数下界和动态的事件规划问题.以上基于事件的社交网络上的工作从未涉猎过事件与用户之间的双向选择问题,仅仅是关注用户单方面的偏好.

2.3 稳定婚姻问题

稳定婚姻问题是指:设有 n 个男人和 n 个女人,将这些男人和女人进行配对,使得该配对方案中没有阻塞对(定义 1).文献[20,21]是两篇非常不错的相关综述.最基础的稳定婚姻问题要求匹配双方(男人和女人)的数量相同,且为一对一匹配.第 1 类变种为:可以允许男人和女人的偏好顺序不是严格的,即,在某个男人(或女人)的偏好列表中,可以对某些女人(或男人)的喜好是相同的.第 2 类变种将一对一匹配扩展为多对多匹配[22].这两种变种与本文强相关,但稳定婚姻的基础问题及此二变种问题,均不考虑匹配双方(即本文中的用户和事件)之间的相关时空信息限制(事件之间的时间冲突,事件与用户的地理位置及用户对行程的预算等).也就是说,以上问题是本文所研究问题的特例.因此,解决这些已有的稳定婚姻及其变种问题的现有方法不能直接应用来解决本文的问题.第 3 类变种为稳定室友问题[23],即给定 $2n$ 个人,他们对其他 $2n-1$ 个人均给出一个偏好列表,稳定室友问题是找到一种没有阻塞对的稳定匹配.该变种与本文的问题差别较大,因此亦不能用来解决本文的问题.

3 基础算法

本节介绍双边偏好稳态规划问题的两个基础算法,分别是事件优先的算法和用户优先的算法.

3.1 事件优先的算法

事件优先算法的主要思路是优先考虑事件对用户的偏好,让每个事件 e_i 优先选择其喜好的用户,如果当前被选择的用户因与其现有事件(比如 e_j)冲突或预算不足的原因无法加入其计划中,则要看用户对事件 e_i 和 e_j 更偏好哪个,令其选择他更偏好的那个事件,释放另一个事件,以便于其他用户可以对其选择.该算法的伪代码见算法 1.

算法 1. 事件优先算法.

输入:事件对用户的偏序 *PE*,用户对事件的偏序 *PU*,事件集 *E*,用户集 *U*;

输出:全局用户安排的事件$\mathcal{P}$.

1: 所有事件处于 *active* 状态;
2: **While** (存在处于 *active* 状态的事件)
3: 取出 *active* 状态的事件 e_i
4: **For** (每一个在 $PE(e_i)$的用户 u_j) //$PE(e_i)$已经按照其偏好对用户排序
5: **If** (e_i 被安排的用户等于人数上界)
6: **break**;
7: **End If**
8: *conflict_events*=∅;
9: **For** (在$\mathcal{P}(u_j)$中的每个事件 e_k)
10: **If** (e_i 与 e_k 在时间上冲突)
11: **If** (e_k 在 $PU(u_j)$的顺序优于 e_i 在 $PU(u_j)$中的顺序) //保障没有阻塞对
12: **break**;
13: **Else**
14: 将 e_k 从 $PU(u_j)$中移除,将 e_k 加入 *conflict_events*,将 e_k 的状态变为 *active*;
15: **End If**
16: **End For**
17: 将 e_i 加入到$\mathcal{P}(u_j)$中;
18: **While** ($\mathcal{P}(u_j)$的行程开销超过 u_j 的行程预算)
19: 从$\mathcal{P}(u_j)$中删除偏爱程度最差的事件 e_k;
20: **If** (e_i 与 e_k 是同一个事件)
21: **Break**;
22: **End**
23: 将 e_k 加入 *conflict_events*;
24: **End While**
25: **If** (e_i 被从$\mathcal{P}(u_j)$的行程中删除)
26: **For** (在 *conflict_events* 中的每个事件 e_k)
27: **If** (加入 e_k 不会使$\mathcal{P}(u_j)$的行程开销超过 u_j 的行程预算)
28: 将 e_k 加入$\mathcal{P}(u_j)$,将 e_k 的状态变为 *inactive*;
29: **End If**
30: **End For**
31: **End If**
32: **End For**
33: 将 e_i 的状态变为 *inactive*;
34: **End While**

算法 1 首先为每个事件设置一个是否活跃的状态:如果是活跃的,记为 *active*,表示该事件依然可以被安排到用户的计划中;否则记为 *inactive*,表示该事件已不可能被安排给任何用户(第 1 行).只要有事件是 *active* 的,则通过第 2 行~第 34 行的方式对其进行用户安排.每次取出一个 *active* 的事件,记为 e_i(第 3 行),根据其对事件的偏好列表 $PE(e_i)$来选择用户.$PE(e_i)$已经按照偏好效用值排序好,令偏好效用值高的用户排在前端,每次选最前端的用户 u_j 进行处理(第 4 行).如果 e_i 被安排的用户数量已经达到了其人数上界,则证明该事件无法被安排给更多的

用户,于是跳出循环(第5行~第7行);否则声明一个集合 *conflict_events* 来存储与 e_i 冲突的事件,并初始化为空(第8行).对于 e_i 选出的当前效用值最高的用户 u_j,首先判断 $\mathcal{P}(u_j)$ 中是否有与 e_i 冲突的事件,如果有(设为 e_k),则判断在 $PU(u_j)$ 中 u_j 更喜欢 e_i 还是 e_k:如果用户更喜欢 e_k,则证明 e_i 无法加入到 $\mathcal{P}(u_j)$ 中,并跳出循环(第10行~第13行);否则将 e_k 从 $\mathcal{P}(u_j)$ 中移除,并加入与 e_i 冲突的 *conflict_events* 集合中,并将 e_k 的状态变回 *active*(第14行~第16行).以上证明 e_i 与 $\mathcal{P}(u_j)$ 中的所有事件不冲突,于是将其加入 $\mathcal{P}(u_j)$ 中(第17行).如果加入 e_i 会使得 $\mathcal{P}(u_j)$ 的行程开销超过 u_j 的行程预算,则不断地删除 $\mathcal{P}(u_j)$ 偏好最差的事件(第18行~第24行).如果 e_i 被从 $\mathcal{P}(u_j)$ 中删除,则证明 e_i 不能加入 $\mathcal{P}(u_j)$ 中,那么判断 *conflict_events* 中的每个事件(依然按照效用值从大到小的顺序)是否有合适的事件 e_k 加入 $\mathcal{P}(u_j)$:如果可以,则加入 e_k 并将其状态更新为 *inactive*(第25行~第32行).最后,将 e_i 的状态变位 *active*(第33行).下面用例2对算法1的过程进行说明.

例2:如例1中,事件和用户的位置如图2所示,用户对事件的偏好效用值见表1,事件对用户的偏好效用值见表2.根据算法1,所有用户的初始状态为 *active*.事件 e_1 首先选择其最喜好的用户 u_5,u_5 的计划中目前还没有事件,因此可以加入事件 e_1,此时 e_1 达到其用户数量上限,停止选择,状态变位 *inactive*.之后,事件 e_2 选择其最喜欢的用户 u_3,经判断,u_3 的计划中可以加入 e_2.事件 e_2 还可以有一个名额来选择其第2喜欢的用户 u_5,此时 u_5 已经参加了 e_1.由于 e_2 与 e_1 不冲突,且 e_2 的加入不会超过 u_5 的预算,因此 e_2 可以加入 u_5 的计划中.事件 e_2 选择完毕,状态变为 *inactive*.事件 e_3 进行选择,首先选择其最喜欢的用户 u_5,但 u_5 已经参加了事件 e_1,与事件 e_3 冲突.而在 u_5 的偏好列表中,他更喜欢 e_3 而不是 e_1,因此将 e_3 加入其计划中,把 e_1 移除计划,并将事件 e_1 的状态改回 *active*.此时 u_5 参加两个事件 e_2 和 e_3 不会超过其行程预算.e_3 继续考察第2喜欢的用户 u_4,u_4 的行程预算不支持往返 e_3,因此不能将 e_3 分配给 u_4.e_3 继续考察第3喜欢的用户 u_1,u_1 可以参加 e_3.e_3 继续考察第4喜欢的用户 u_3,u_3 的预算不支持同时参加 e_2 和 e_3.e_3 继续考察第5喜欢的用户 u_2,u_2 可以参加 e_3.此时 e_3 的参与人数达到上界,状态变位 *inactive*.由于 u_5 改去 e_3 而放弃 e_1,因此 e_1 现在是 *active* 的,因此令 e_1 选择当前最喜欢的用户 u_4.u_4 可以参加 e_1,e_1 的参与人数达到上界,变为 *inactive*.此时,所有的事件状态均已变为 *inactive*,算法结束.

算法正确性分析:算法在第4行保障当前事件所选择的用户是其最喜欢的,而一旦事件不得不放弃选择其最喜欢的用户时,算法的第11行保障该用户选择到的事件是其最喜欢的.因此,至少保障用户或事件一方满足其最喜欢的选择,因此不会出现阻塞对.同时,每当在计划中加入一个事件或者用户时,均会判断是否满足事件人数上界、用户行程开销、事件与计划中是否存在冲突这些所有约束条件是否满足.当且仅当所有的约束条件均满足时,才会插入该事件或者用户.因此,算法1可以满足定义2中的所有条件,所以算法1是正确的.

算法复杂度分析:算法在第2行需要遍历所有事件,时间复杂度为 $O(m)$;第4行~第16行需要每个事件遍历 $O(n)$ 个用户,其余的计算复杂度为 $O(1)$.因此,算法的时间复杂度为 $O(mn)$.空间复杂度也为 $O(mn)$.

3.2 用户优先算法

用户优先算法的主要思路是:优先考虑用户对事件的偏好,让每个用户优先选择其喜好的事件,如果当前被选择的事件因与其现有事件冲突或预算原因无法加入其计划中,则放弃此事件.如果事件是由于参加人数已满的原因而无法加入,则要看当前用户在该事件已匹配的用户的效用值是否是最低的:如果是,则放弃加入该事件;否则,用该用户替换其计划中效用值最低的用户,并将其释放另一个事件,以便其选择其他事件.

该算法的伪代码见算法2所示.

算法 2. 用户优先算法.

输入:事件对用户的偏序 *PE*,用户对事件的偏序 *PU*,事件集 *E*,用户集 *U*;

输出:全局用户安排的事件 $\mathcal{P}$.

1: 所有用户处于 *active* 状态;
2: **While** (存在处于 *active* 状态的用户)
3: 取出 *active* 状态的用户 u_i
4: **For** (每一个在 $PU(u_i)$ 的事件 e_j) //$PU(u_i)$ 已经按照其偏好对用户排序
5: **If** (u_i 的当前行程已超过预算或与当前计划中的事件冲突)

6: **break**;
7: **End If**
8: 事件 e_j 加入$\mathcal{P}(u_i)$中;
9: **If** (事件 e_j 的人数超过其上界)
10: **If** (u_i 在$\mathcal{P}(e_j)$的偏好效用值是最低的) //保障没有阻塞对
11: **break**;
12: **Else If**
13: 将$\mathcal{P}(e_j)$中效用值最低的用户 u_k 移除;
14: 将 u_k 的状态变为 *active*;
15: **End If**
16: **End If**
17: **End For**
18: 将 u_i 的状态变为 *inactive*;
19: **End While**

算法 2 首先为每个用户设置一个是否活跃的状态:如果是活跃的,记为 *active*,表示该用户依然可以参加其他事件;否则记为 *inactive*,表示该用户已无法参加更多事件(第 1 行).只要有用户是 *active* 的,则通过第 2 行~第 19 行的方式对其进行事件安排.每次取出一个 *active* 的用户,记为 u_i(第 3 行),根据其对事件的偏好列表 $PU(u_i)$来选择用户.$PU(u_i)$已经按照偏好效用值排序好,令偏好效用值高的事件排在前端,每次选最前端的事件 e_j 进行处理(第 4 行).如果加入 e_j 会使得 u_i 的当前行程已超过预算或与当前计划中的事件冲突,于是跳出循环(第 5 行~第 7 行);否则,将 e_j 加入到$\mathcal{P}(u_i)$中(第 8 行).如果加入事件 e_j 后,e_j 中所安排的人数超过了其上限,则要看当前用户在该事件已匹配用户的效用值是否是最低的:如果是,则放弃加入该事件;否则,用该用户替换其计划中效用值最低的用户,并将另一个事件释放,以便其选择其他事件(第 9 行~第 16 行).如此循环下去,直到$\mathcal{P}(u_i)$中无法再加入新的事件为止(第 4 行~第 17 行).此时,将 u_i 的状态变为 *inactive*(第 18 行).当所有用户的状态均变为 *inactive* 时,算法结束.下面用例 3 对算法 2 的过程加以说明.

例 3:例 1 中,事件和用户的位置如图 2 所示,用户对事件的偏好效用值见表 1,事件对用户的偏好效用值见表 2.根据算法 2,所有用户的初始状态为 *active*.用户 u_1 首先选择其最喜好的事件 e_1,将 e_1 加入$\mathcal{P}(u_1)$中;其次是 e_2,将 e_2 加入$\mathcal{P}(u_1)$中;之后考察 e_3,e_3 与 e_1 冲突,因此不能将 e_3 加入$\mathcal{P}(u_1)$中.u_1 选择完毕,状态变为 *inactive*.用户 u_2 选择事件,首先选择其最喜欢的事件 e_3,将 e_3 加入$\mathcal{P}(u_2)$中;其次是 e_2,将 e_2 加入$\mathcal{P}(u_2)$中;之后考察 e_1,e_1 与 e_3 冲突,因此不能将 e_3 加入$\mathcal{P}(u_2)$中.u_2 选择完毕,状态变为 *inactive*.用户 u_3 选择事件,首先选择其最喜欢的事件 e_3,将 e_3 加入$\mathcal{P}(u_3)$中;其次是 e_1,e_1 与 e_3 冲突,因此不能将 e_3 加入$\mathcal{P}(u_3)$中;之后考察事件 e_2,加入 e_2 会超过 u_3 的预算,因此不能将 e_2 加入$\mathcal{P}(u_3)$中.u_3 选择完毕,状态变为 *inactive*.用户 u_4 选择事件,首先选择其最喜欢的事件 e_2,由于此时$\mathcal{P}(e_2)$中已经有两个用户 u_1 和 u_2,加入 u_4 超过其人数上限,且在 e_2 的偏好列表中 u_4 的效用值比 u_1 和 u_2 都低,因此不能将 e_2 加入$\mathcal{P}(u_4)$中;其次是 e_1,由于此时$\mathcal{P}(e_1)$中已经有一个用户 u_1,加入 u_4 超过其人数上限,而 e_1 比起 u_1 更喜欢 u_4,因此将 e_1 加入到 u_4 中,并从$\mathcal{P}(u_1)$中删除 e_1,将 u_1 的状态变为 *active*;之后考察 e_3,e_3 与 e_1 冲突,因此不能将 e_3 加入$\mathcal{P}(u_4)$中.u_4 选择完毕,状态变为 *inactive*.用户 u_5 选择事件,首先选择其最喜欢的事件 e_2,将 e_2 加入$\mathcal{P}(u_5)$中;由于此时$\mathcal{P}(e_2)$中已经有两个用户 u_1 和 u_2,加入 u_5 超过其人数上限,又因为其中 e_2 最喜欢 u_5、最不喜欢 u_2,因此将 e_2 加入到 u_5 中,并从$\mathcal{P}(u_2)$中删除 e_2,将 u_2 的状态变为 *active*;其次是 e_3,将 e_1 加入$\mathcal{P}(u_5)$中;之后考察 e_1,e_1 与 e_3 冲突,因此不能将 e_1 加入$\mathcal{P}(u_5)$中.u_5 选择完毕,状态变为 *inactive*.此时,由于用户 u_1 和 u_2 的状态是 *active* 的,需要继续考察.对于用户 u_1 而言,此时他最喜欢的事件是 e_3,但加入 e_3 会超过 u_1 的预算,因此不能将 e_3 加入$\mathcal{P}(u_1)$中.u_1 选择完毕,状态变为 *inactive*.对于用户 u_2,还剩 e_1 可供选择,但 e_1 与 e_3 冲突,因此不能将 e_1 加入$\mathcal{P}(u_2)$中.u_2 选择完毕,状态变为 *inactive*.此时,所有用户的状态均为 *inactive*,算法终止.

算法正确性分析:算法在第 4 行保障当前用户所选择的事件是其最喜欢的,而一旦用户不得不放弃选择其

最喜欢的事件时,算法的第 10 行保障该事件选择到的用户是其当前最喜欢的.因此,至少保障用户或事件一方满足其最喜欢的选择,因此不会出现阻塞对.同时,每当在计划中加入一个事件或者用户时,均会判断是否满足事件人数上界、用户行程开销、事件与计划中是否存在冲突这些所有约束条件是否满足.当且仅当所有的约束条件均满足时,才会插入该事件或者用户.因此,算法 2 可以满足定义 2 中的所有条件,所以算法 2 是正确的.

算法复杂度分析:算法在第 2 行需要遍历所有用户,时间复杂度为 $O(n)$;第 4 行~第 17 行需要每个用户遍历 $O(m)$个事件,其余的计算复杂度为 $O(1)$.因此,算法的时间复杂度为 $O(mn)$.空间复杂度也为 $O(mn)$.

4 改进算法

本节首先在第 4.1 节对基础算法进行分析,指出基础算法的缺点.针对该缺点提出改进算法,在第 4.2 节中对其加以详述.

4.1 对基础算法的分析

两个基础算法均从事件(或用户的)单边优先的角度考虑,当遇到冲突或预算不足时,才通过考虑另一方的偏好来进行调整,以保证无阻塞对的出现.但这样可能会出现一种情况,即:用户和事件双方均没有选到各自较为喜欢的事件和用户,均以选择相对不喜欢的事件或用户从而达到稳态(即无阻塞对的出现).这样会使得虽然规划满足稳态条件,但总体的效用值会变得较低.那么是否有一种方式,可以满足时空约束条件、在达到稳定状态的前提下,得到尽可能高的效用值呢?本节便解决这一问题.其次,对于每个事件和用户,都要存储一个 $m\times n$ 的效用值表,当数据量大时,耗用大量的时间进行搜索,且需要大量内存来存储.因此,本节也介绍一种空间剪枝方法,来缩小效用值表格的存储.

4.2 改进算法描述

改进算法的主要思路是:综合了用户和事件双方对彼此的偏好,让每个用户(事件)优先选择其喜好的事件(用户),如果当前被选择的事件因与其现有事件冲突或预算原因无法加入其计划中,则放弃此事件.如果事件是由于参加人数已满的原因而无法加入,则要看当前用户在该事件已匹配的用户的效用值是否是最低的:如果是,则放弃加入该事件;否则,用该用户替换其计划中效用值最低的用户,并将其释放释放另一个事件,以便其选择其他事件.该算法的伪代码见算法 3.

算法 3. 改进算法.

输入:事件对用户的偏序 PE,用户对事件的偏序 PU,事件集 E,用户集 U;

输出:全局用户安排的事件$\mathcal{P}$.

```
将所有用户和事件的组合按照偏好加和递增排序;
For (每一个用户和事件的组合(e_i,u_j)
  If (事件 e_i 的人数超过其上界且事件 e_i 更偏爱当前分配的用户)
    break;
  End If
  conflict_events=∅;
    For (在P(u_j)中的每个事件 e_k)
      If (e_i 与 e_k 在时间上冲突)
        If (e_k 在 PU(u_j)的顺序优于 e_i 在 PU(u_j)中的顺序)   //保障没有阻塞对
        break;
  Else
          将 e_k 从P(u_j)中移除,将 e_k 加入 conflict_events;
      End If
    End For
```

15: 将 e_i 加入到$\mathcal{P}(u_j)$中;
16: **While** ($\mathcal{P}(u_j)$的行程开销超过 u_j 的行程预算)
17: 从$\mathcal{P}(u_j)$中删除偏爱程度最差的事件 e_k;
18: **If** (e_i 与 e_k 是同一个事件)
19: **Break**;
20: **End If**
21: 将 e_k 加入 *conflict_events*;
22: **End While**
23: **If** (e_i 被从$\mathcal{P}(u_j)$的行程中删除)
24: **For** (在 *conflict_events* 中的每个事件 e_k)
25: 将 e_k 加入$\mathcal{P}(u_j)$,将 e_k 的状态变为 *inactive*;
26: **End For**
27: **Else If** (事件 e_i 的人数超过其上界)
28: 找到目前事件 e_i 偏好程度最低的用户 u_k,并将 e_i 从$\mathcal{P}(u_k)$中删除;
29: **End If**
30: **End For**

算法 3 首先将所有可能的用户和事件组合按照互相偏序加和,按照和的递增排序,和越小意味着用户和事件之间的偏好程度越高(第 1 行).按顺序取出每一个组合,记(e_i,u_j)(第 3 行),如果 e_i 被安排的用户数量已经达到了其人数上界且更偏好于当前被安排的用户,则证明 u_j 无法被安排到该事件,于是跳出循环(第 3 行~第 5 行);否则,声明一个集合 *conflict_events* 来存储与 e_i 冲突的事件,并初始化为空(第 6 行).首先判断$\mathcal{P}(u_j)$中是否有与 e_i 冲突的事件,如果有(设为 e_k),则判断在 $PU(u_j)$中 u_j 更喜欢 e_i 还是 e_k:如果用户更喜欢 e_k,则证明 e_i 无法加入到$\mathcal{P}(u_j)$中,并跳出循环(第 8 行~第 10 行);否则,将 e_k 从$\mathcal{P}(u_j)$中移除,并加入与 e_i 冲突的 *conflict_events* 集合中(第 11 行~第 14 行).以上证明 e_i 与$\mathcal{P}(u_j)$中的所有事件不冲突,于是将其加入$\mathcal{P}(u_j)$中(第 15 行).如果加入 e_i 会使得$\mathcal{P}(u_j)$的行程开销超过 u_j 的行程预算,则不断的删除$\mathcal{P}(u_j)$偏好最差的事件(第 16 行~第 22 行).如果 e_i 被从$\mathcal{P}(u_j)$中删除,则证明 e_i 不能加入$\mathcal{P}(u_j)$中,那么判断 *conflict_events* 中的每个事件(依然按照效用值从大到小的顺序)是否有合适的事件 e_k 加入$\mathcal{P}(u_j)$:如果可以,则加入 e_k(第 23 行~第 26 行).最后,如果 e_i 分配的用户超过人数上界,则删除 e_i 中偏好程度最低的用户,并将 e_i 从$\mathcal{P}(u_k)$中删除(第 27 行~第 30 行).下面用例 4 对算法 3 的过程加以说明.

例 4:如例 1 中,事件和用户的位置如图 2 所示,用户对事件的偏好效用值见表 1,事件对用户的偏好效用值见表 2.根据算法 3,所有用户和事件的组合排序结果为(e_2,u_5),(e_3,u_5),(e_1,u_4),…,(e_1,u_3).因为 e_2 对 u_5 的偏序是 2,u_5 对 e_2 的偏序是 1,和为 3,(e_2,u_5)排在最前面;(e_3,u_5)的和也为 3,但是 $e_3>e_2$,所以排在第 2 个位置;(e_1,u_4)的和为 4,排在第 3 个位置;以此类推,(e_1,u_3)的和为 7,排在最后一个.首先考察组合(e_2,u_5),将 e_2 加入$\mathcal{P}(u_5)$中.接下来分别考察(e_3,u_5)和(e_1,u_4),分别将 e_3 和 e_2 加入$\mathcal{P}(u_5)$和$\mathcal{P}(u_4)$中.当考察(e_1,u_5)时,将 e_1 加入$\mathcal{P}(u_5)$中会超过 u_5 的预算,所以 e_1 无法加入到$\mathcal{P}(u_5)$中.接下来将 e_2 加入到$\mathcal{P}(u_3)$中,将 e_2 加入到$\mathcal{P}(u_3)$中.考察(e_1,u_3),e_1 已经达到人数上界,且更偏爱 u_4,所以 e_1 加入$\mathcal{P}(u_3)$中.考察(e_2,u_1),e_2 已经达到人数上界,但 e_2 更偏爱 u_1,将 e_2 加入到$\mathcal{P}(u_1)$中,将 e_2 从$\mathcal{P}(u_3)$中移除.下一组,将 e_3 加入到$\mathcal{P}(u_3)$中.e_3 无法加入到$\mathcal{P}(u_4)$中,因为将超过 u_4 的预算.e_1 无法加入到$\mathcal{P}(u_1)$中,因为 e_1 更加偏爱当前的 u_4.同理,e_2 无法加入到$\mathcal{P}(u_2)$中.接下来考察(e_3,u_1),由于加入 e_3 将超过 u_1 的预算,所以 e_3 无法加入到$\mathcal{P}(u_1)$中.将 e_3 加入到$\mathcal{P}(u_2)$中.由于将会超过 u_4 的预算,e_3 无法加入到$\mathcal{P}(u_4)$中.最后,e_1 无法加入到$\mathcal{P}(u_3)$中,因为 e_1 更加偏爱当前的 u_3.此时,所有的组合遍历完毕,算法终止.

算法正确性分析:算法在第 3 行保障当前事件所选择的用户是其最喜欢的,而一旦事件不得不放弃选择其最喜欢的用户时,算法的第 9 行保障该用户选择到的事件是其最喜欢的.因此,至少保障用户或事件一方满足其最喜欢的选择,因此不会出现阻塞对.同时,每当在计划中加入一个事件或者用户时,均会判断是否满足事件人数上界、用户行程开销、事件与计划中是否存在冲突这些所有约束条件是否满足.当且仅当所有的约束条件均

满足时,才会插入该事件或者用户.因此,算法 3 可以满足定义 2 中的所有条件,所以算法 3 是正确的.

算法复杂度分析:算法在第 2 行需要遍历所有可能组合,时间复杂度为 $O(mn)$;其余的计算复杂度为 $O(1)$.因此,算法的时间复杂度为 $O(mn)$.空间复杂度也为 $O(mn)$.

4.3 空间剪枝方法

不难发现,对于任意一个用户 u_i,他所能参加的最远的事件,其举办地与 u_i 所在位置的距离不能超过 $B_i/2$.因此,u_i 可以参加的事件,其举办地必须在以 u_i 所在位置 $\boldsymbol{l}_{u_i}$ 为圆心、$B_i/2$ 为半径所在的圆内.于是,在算法运行之前,可以对数据集进行预处理.对每个用户 u_i 来说,以 $\boldsymbol{l}_{u_i}$ 为圆心、$B_i/2$ 为半径画一个圆,将 $\boldsymbol{l}_{e_i}$ 不在圆内的每个事件 e_j、彼此的效用值 $p_u(u_i,e_j)$和 $p_e(e_j,u_i)$均置为 0,即从彼此效用值表中删除.利用这种方法可以大量减少两个效用值表的存储,并节省算法在遍历两个效用值表时所用的时间.

5 实验及分析

本节给出算法的实验结果,包括实验环境、算法的计算时间、内存消耗、获得的总体效用值.另外,将本文提出的稳态规划问题定义与现有工作[6]所提出的问题定义(即忽略定义 2 中的条件(4))及算法进行比较,证明本文提出问题定义及相应算法的必要性.

5.1 数据集及实验环境

本文算法用 C++及其 STL 库所实现.实验在一台服务器上运行,该服务器的操作系统为 Linux Fedora 16 (Linux3.6.11-4.fc16*86 62 GNOME 3.2.1),处理器为 Intel(R)Xeon(R) CPU E5-2620 0@2.00GHz,内存 64GB.内存开销利用 Linux 系统函数进行测试.

本实验所使用的数据集为 Meetup 数据集[1],数据集中,每个用户有一个标签文件和位置文件.标签文件记录用户在平台上注册时所选择的兴趣标签;位置文件记录用户位置的经纬度坐标.对于每个事件,有一个位置文件以及一个事件群组文件.在 Meetup 中,事件是由事件群所创建的,事件群文件记录着包含在群中的事件,标签文件记录着每个群组的兴趣标签.位置文件记录着每个事件举办地的经纬度.利用用户的表情文件、事件的标签文件、群组的标签文件,可以计算用户对事件感兴趣程度的效用值[1,3].事件对用户的感兴趣程度,由用户参加事件的历史记录所计算.设 e 为用户 u 参加过的历史事件,e 的标签集为$\{L_1,L_2,\ldots,L_n\}$,则每个拥有$\{L_1,L_2,\ldots,L_n\}$中至少其中一个表情的事件对 u 的好感度+1.统计所有历史事件后,将每个事件对用户的好感度进行归一化.其他参数的产生方式与文献[6]中相同.表 3 中总结了真实数据集中的各参数,其中,n 为数据集中的用户数量,m 为数据集中的事件数量,η为事件参与人数的上界,冲突率表示冲突事件占事件总数的百分比.

Table 3 Real datasets

表 3 真实数据集

城市名称	n	m	η的均值	冲突率
北京	113	16	50	0.25
奥克兰	569	37	50	0.25
夏威夷	2 967	817	50	0.25
阿拉斯加	2 394	4 156	50	0.25
香港	3 528	1 324	50	0.25
新加坡	9 893	4 257	50	0.25
温哥华	16 095	11 536	50	0.25

为了测试算法的可扩展性,从温哥华真实数据中截取了一定数量的用户和事件,生成几个合成数据集.表 4 列出了合成数据集中用户数量和事件数量的变化情况,其中加粗的为默认值.

本实验对比本文的算法 1~算法 3 以及文献[6]中的算法.以下,算法 1 记为事件稳定规划,算法 2 记为用户稳定规划,算法 3 记为改进稳定规划,文献[6]中的算法记为单边规划.算法 1~算法 3 均经过第 4.3 节空间剪枝方法的预处理.在真实数据集上,测试以上 4 个算法的总效用值、运行时间、所需内存以及“单边规划”算法所得结果

中阻塞对数量所占的百分比.在可扩展性测试中,测试以上 4 种算法在用户数量 n 和事件数量 m 变化时,运行时间和总效用值的变化情况.在无特殊说明的情况下,n 和 m 的值取默认值.

Table 4 Synthetic datasets

表 4 合成数据集

参数名称	设置值
n	20, **50**, 100, 200, 500
m	200, 500, 1000, **5000**

5.2 真实数据集上的实验结果

4 种算法在真实数据集上的实验结果见表 5~表 7.表 5 显示各算法在真实数据集上获得的总效用值.表 6 显示单边规划算法在真实数据集上运行结果中阻塞对所占的百分比.表 7 显示各算法在真实数据集上运行所需的时间开销和内存开销.

Table 5 Results of total utility over real datasets

表 5 真实数据集上对总效用值的测试结果

城市名称	单边规划	事件稳定规划	用户稳定规划	改进稳定规划
北京	212.6	218.9	210.3	219.8
奥克兰	2 348.3	2 439.7	2 436.8	2 442.2
夏威夷	11 649.8	1 237.8	12 297.2	12 416.7
阿拉斯加	10 108.5	10 198.3	10 239.6	10 367.3
香港	18 538.2	17 936.5	18 773.2	18 933.2
新加坡	50 175.2	47 682.3	46 811.0	52 927.8
温哥华	31 357.0	69 384.7	69 001.0	70 789.0

Table 6 Results of percentage of block pairs over real datasets

表 6 真实数据集上单边规划阻塞对数量百分比

城市名称	单边规划阻塞对数量百分比(%)
北京	39.31
奥克兰	61.14
夏威夷	32.35
阿拉斯加	26.38
香港	52.32
新加坡	68.73
温哥华	47.43

Table 7 Resultsof time and memory cost over real datasets

表 7 真实数据集上对时间和内存开销的测试结果

城市名称		北京	奥克兰	夏威夷	阿拉斯加	香港	新加坡	温哥华
单边规划	计算时间(s)	0.25	0.37	2	10	5	37	114
	消耗内存(MB)	98.72	2 543.3	12 347.9	11 325.4	18 970.3	47 250.8	69 472.7
事件稳定规划	计算时间(s)	0.13	1	64	214	383	10 088	43 931
	消耗内存(MB)	96.10	0.18	0.02	23.40	40.95	368.43	1 630.93
用户稳定规划	计算时间(s)	0.21	2.47	85.63	227	452	10 458	44 295
	消耗内存(MB)	97.85	0.18	0.02	25.40	41.37	379.46	1 689.25
改进稳定规划	计算时间(s)	0.09	0.15	12	33	27	395	1 436
	消耗内存(MB)	104.35	0.29	0.18	42.50	51.37	368.43	1 689.25

由表 5 可以看出:改进稳定规划算法所获得的总效用值是最高的,在温哥华数据集上,其效用值甚至比单边规划算法高出 2 倍以上.事件稳定规划算法和用户稳定规划算法所获得的总效用值在数据集北京、香港、新加坡上比单边规划低.这说明现有的单边规划算法不考虑事件与用户双方的效用偏好,它得到的总效用值是比较随机的.而考虑了双边效用偏好的算法在大部分情况下,也可以获得较高的效用值.事件稳定规划算法和用户稳定规划两个算法偶尔得到效用值低的情况,是它们在双方均取较低效用值的情况下所达到的稳态规划.而改进

算法较好地规避了这一情况,因此在保证稳态规划下,也可以获得较高的总效用值,在数据量大的情况下尤为显著.该实验证明了本文提出稳态规划的合理性和必要性.

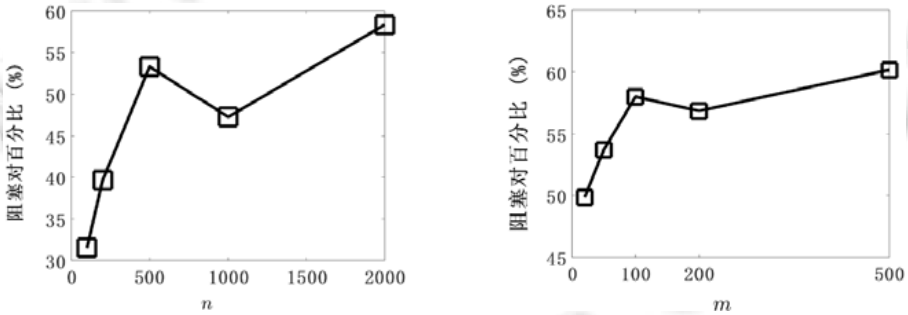
由表 6 可以看出:单边规划算法在真实数据集上所得到的计算结果中,阻塞对的数量不在少数,在数据集奥克兰和新加坡上,甚至高达 61.14%和 68.73%,在 3 个稳态规划算法中,阻塞对的数量为 0.根据表 5 和表 6 二者综合的结果可以看出:单边规划算法既不能保证事件和用户双方共同达到满意,在总效用值上也不如考虑双方偏好效用的稳态算法.由此可以进一步的证明,本文提出的稳态规划问题的合理性和必要性.

由表 7 可以看出,单边规划算法和双边稳态算法的内存开销差别不大.而在时间开销上,双边稳态规划算法比单边算法要高一些.这是因为,双边稳态算法需要一定的时间来保障匹配结果中没有阻塞对的存在,因此需要更多的时间开销.另外,虽然本文提出的 3 种算法在时间复杂度上是一样的,但平均来说,改进稳态算法比两个基础算法所花费的时间要低一些.这是因为改进算法同时考虑双方共同的选择,因此当遇到阻塞对出现时,所需调整的空间就小一些.因此,改进算法在实验中的时间开销会比两个基础算法要小,但比单边规划算法要大一些.

5.3 合成数据集上的实验结果

本节分析本文提出的 3 种算法在合成数据集上的可扩展性,并与单边规划算法进行比较.实验结果如图 3 所示.当用户数量 n 变化时,m 取默认值 5 000;当事件数量 m 变化时,n 取默认值 50.

图 3 所示为单边规划算法在合成数据集上阻塞对所占的百分比,可以看出:随着 n 与 m 变大,单边规划算法在合成数据集上阻塞对所占的百分比基本上在增高,当然也存在下降的情况.这是因为单边规划算法完全不考虑双方稳态平衡的问题,产生阻塞对的情况基本上是随机的.而随着数据量的增大,产生阻塞对的概率就会增大,因此阻塞对百分比基本会呈现上升趋势.

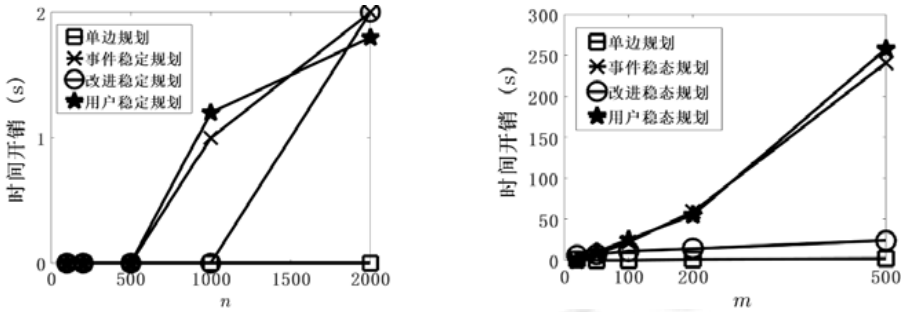


(a) 阻塞对百分比随用户数量的变化　　(b) 阻塞对百分比随事件数量的变化

Fig.3 Percentage of block pairs in the results of unilateral planning algorithms over synthetic datasets

图 3 单边规划算法在合成数据集上阻塞对所占的百分比

图 4 所示为时间开销随用户数量和事件数量的变化情况,从图中可以看出,4 种算法的时间开销均随用户数量和事件数量的增加而增加.该现象显然是合理的,因为随着数据量的增加,算法在检索遍历事件和用户之间效用值的过程所耗费的时间就要增加.这是因为双边稳态算法需要一定的时间来保障匹配结果中没有阻塞对的存在,因此需要更多的时间开销.另外,单边规划的时间开销比较低,随着用户和事件数量的增大而增加的幅度也比较小.事件稳态规划和用户稳态规划两种基础算法所耗费的时间最长,二者之间的差别不大.改进稳态规划的时间开销介于单边规划和两个基础算法之间.这是因为改进算法同时考虑双方共同的选择,因此当遇到阻塞对出现时,所需调整的空间就小一些.

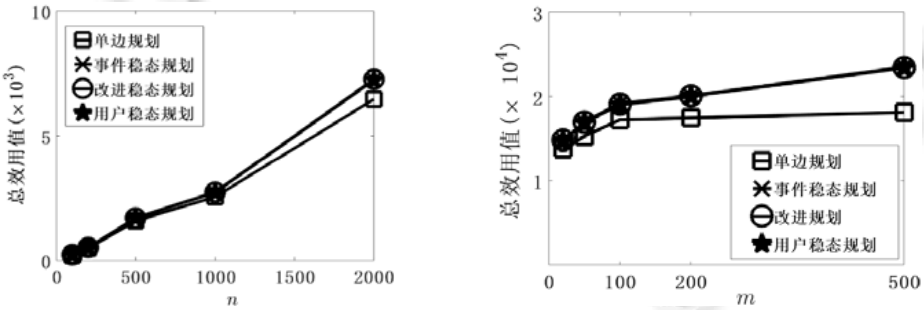


(a) 时间开销随用户数量的变化 (b) 时间开销随事件数量的变化

Fig.4 Time cost with reference to the number of users and events

图 4 时间开销随用户数量和事件数量的变化

图 5 所示为 4 种算法的总效用值随用户数量和事件数量的变化情况,可以看出,4 种算法的总效用值均随用户数量和事件数量的增加而增加.该现象显然是合理的,因为随着数据量的增加,形成匹配的用户和事件的数量就会增加,因此总体的效用值必然增加.改进稳定规划算法所获得的总效用值是最高的;事件稳定规划算法和用户稳定规划算法所获得的总效用值次之,但并不明显;单边规划的总效用值最低.由此可见:现有的单边规划定义和算法无论在考虑事件和用户的双方满意度上,还是总体效用值上,均不如考虑双边稳态的定义和算法达到的效果好,因此也佐证了本文提出的双边稳态问题的合理性.



(a) 总效用值随用户数量的变化 (b) 总效用值随事件数量的变化

Fig.5 Total utility with reference to the number of users and events

图 5 总效用值随用户数量和事件数量的变化

6 结论及展望

本文重点研究社交网络中为用户规划感兴趣事件这一经典问题,从用户和事件两个角度进行偏好考虑,提出一种双边偏好稳态规划算法,解决了社交网络中这一双向选择难题.为了验证该双边偏好算法的有效性,设计了两种基础算法与一种改进算法,并通过实验进行了对比.实验结果显示,双边稳态规划对比现有的单边规划无论从总效用值上,还是满足事件与用户双方偏好上,均没有双边稳态规划的效果好.从而为主办者和参与者提供了一种高效、高质量的双向匹配事件推送模式,有效解决了这一经典问题中的双向选择难题,有助于提高社交平台的用户满意度与价值提升.

未来工作中,可以考虑事件与用户中各个参数的动态变化的可能,提供支持动态变化的双边稳态规划算法.

References:

[1] Liu X, He Q, Tian Y, Lee WC, McPherson J, Han J. Event basedsocial networks: Linking the online and offline social worlds. In: Proc. of the 18th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. Beijing: ACM Press, 2012. 1032−1040.

[2] Cheng Y, Yuan Y, Chen L, Giraud-Carrier C, Wang G. Complexevent-participant planning and its incremental variant. In: Proc. of the 2017 IEEE 33rd Int'l Conf. on Data Engineering. San Diego: IEEE, 2017. 859−870.

[3] She J, Tong Y, Chen L, Cao CC. Conflict-aware eventparticipantarrangement. In: Proc. of the 2015 IEEE 31st Int'l Conf. on Data Engineering. Seoul: IEEE, 2015. 735−746.

[4] She J, Tong Y, Chen L, Cao CC. Conflict-aware event-participant arrangement and its variant for online setting. IEEE Trans. on Knowledge and Data Engineering, 2016,28(9):2281−2295.

[5] Li K, Lu W, Bhagat S, Lakshmanan LV, Yu C. On social eventorganization. In: Proc. of the 20th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM Press, 2014. 1206−1215.

[6] She J, Tong Y, Chen L. Utility-aware social event-participantplanning. In: Proc. of the 2015 ACM SIGMOD Int'l Conf. on Management of Data. Melbourne: ACM Press, 2015. 1629−1643.

[7] Li SJ, Zhao HY. Foundation of Economics. Beijing: Tsinghua University Press, 2015 (in Chinese).

[8] Minkov E, Charrow B, Ledlie J, Teller S, Jaakkola T. Collaborativefuture event recommendation. In: Proc. of the 19th ACM Int'l Conf. on Information and Knowledge Management. Toronto: ACM Press, 2010. 819−828.

[9] Liao G, Zhao Y, Xie S, Yu PS. An effective latent networksfusion based model for event recommendation in offline ephemeral social networks. In: Proc. of the 22nd ACM Int'l Conf. on Information & Knowledge Management. San Francisco: ACM Press, 2013. 1655−1660.

[10] Khrouf H, Troncy R. Hybrid event recommendation using linkeddata and user diversity. In: Proc. of the 7th ACM Conf. on Recommender Systems. Hong Kong: ACM Press, 2013. 185−192.

[11] Sun YC, Chen CC. A novel social event recommendation method based on social and collaborative friendships. In: Proc. of the Int'l Conf. on Social Informatics. Kyoto: Springer-Verlag, 2013. 109−118.

[12] Chen C, Zhang D, Guo B, Ma X, Pan G, Wu Z. Tripplanner: Personalized trip planning leveraging heterogeneous crowdsourced digitalfootprints. IEEE Trans. on Intelligent Transportation Systems, 2015,16(3):1259−1273.

[13] Lu EHC, Chen CY, Tseng VS. Personalized trip recommendation with multiple constraints by mining user check-in behaviors. In: Proc. of the 20th Int'l Conf. on Advances in Geographic Information Systems. Redondo Beach: ACM Press, 2012. 209−218.

[14] Zhang W, Wang J, Feng W. Combining latent factor model with location features for event-based group recommendation. In: Proc. of the 19th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. Chicago: ACM Press, 2013. 910−918.

[15] Du R, Yu Z, Mei T, Wang Z, Wang Z, Guo B. Predicting Activity Attendance in Event-based Social Networks: Content, Context and Social Influence. Seattle: ACM Press, 2014. 425−434.

[16] Pham TAN, Li X, Cong G, Zhang Z. A general graph-basedmodel for recommendation in event-based social networks. In: Proc. of the 2015 IEEE 31st Int'l Conf. on Data Engineering. Seoul: IEEE, 2015. 567−578.

[17] Feng K, Cong G, Bhowmick SS, Ma S. In search of influentialevent organizers in online social networks. In: Proc. of the 2014 ACM SIGMOD Int'l Conf. on Management of Data. Snowbird: ACM Press, 2014. 63−74.

[18] Kempe D, Kleinberg J, Tardos E. Maximizing the spread ofinfluence through a social network. In: Proc. of the 9th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. Washington: ACM Press, 2003. 137−146.

[19] Lappas T, Liu K, Terzi E. Finding a team of experts in social networks. In: Proc. of the 15th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. Paris: ACM Press, 2009. 467−476.

[20] Iwama K, Miyazaki S. A survey of the stable marriage problem and its variants. In: Proc. of the Int'l Conf. on Informatics Education and Research for Knowledge-Circulating Society. Kyoto: IEEE, 2008. 131−136.

[21] Gusfield D, Irving RW. The Stable Marriage Problem: Structure and Algorithms. Cambridge: MIT Press, 1989.

[22] Malhotra VS. On the stability of multiple partner stable marriages with ties. In: Proc. of the European Symp. on Algorithms. Beilin: Springer-Verlag, 2004. 508−519.

[23] Kujansuu E, Lindberg T, M¨akinen E. The stable room mates problem and chess tournament pairings. Divulgaciones Matemáticas, 1999,7(1):19−28.

[24] Qiao SJ, Han N, Zhou JL, Li RH, Jin CQ, Gutierrez LA. SocialMix: A familiarity-based and preference-aware location suggestion approach. Engineering Applications of Artificial Intelligence, 2018,68:192−204.

[25] Qiao SJ, Han N, Gao YJ, Li RH, Huang JB, Guo J, Gutierrez LA, Wu XD. A fast parallel community discovery model on complex networks through approximate optimization. IEEE Trans. on Knowledge and Data Engineering, 2018,30(9):1638−1651.

[26] Qiao SJ, Han N, *et al*. TraPlan: An effective three-in-one trajectory prediction model in transportation networks. IEEE Trans. on Intelligent Transportation Systems, 2015,16(3):1188−1198.

[27] Qiao SJ, Shen DY, Wang XT, Han N, Zhu W. A self-adaptive parameter selection trajectory prediction approach via hidden Markov models. IEEE Trans. on Intelligent Transportation Systems, 2015,16(1):284−296.

附中文参考文献:

[7] 李世杰,赵岩红.经济学基础.北京:清华大学出版社,2015.

成雨蓉(1989－),女,山西清徐人,博士,CCF 专业会员,主要研究领域为图数据查询处理与分析,时空众包数据分析,社交网络数据分析.

李博扬(1992－),男,学士,主要研究领域为社交网络数据分析,时空数据分析,机器学习.

王国仁(1966－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为不确定数据管理,数据密集型计算,可视媒体数据管理与分析,非结构化数据管理,分布式查询处理与优化技术(主要包括传感器网络和 P2P 对等计算),生物信息学.

袁野(1981－),男,博士,教授,博士生导师,CCF 专业会员,主要研究领域为云计算,大数据管理(包括图数据管理,不确定数据管理,数据隐私保护),P2P 计算.

www.jos.org.cn