控制与决策
Control and Decision

# 基于项集归减的高维频繁高效用项集挖掘多目标优化方法

张磊, 李柳, 杨海鹏, 孙翔, 程凡, 孙晓燕, 苏喻

引用本文:
张磊,李柳,杨海鹏,孙翔,程凡,孙晓燕,苏喻. 基于项集归减的高维频繁高效用项集挖掘多目标优化方法[J]. 控制与决策, 2023, 38(10): 2832−2840.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2021.1900

---

# 基于项集归减的高维频繁高效用项集挖掘多目标优化方法

张 磊[1], 李 柳[1], 杨海鹏[1], 孙 翔[1], 程 凡[2†], 孙晓燕[3], 苏 喻[4,5]

(1. 安徽大学 计算机科学与技术学院，合肥 230039；2. 安徽大学 人工智能学院，合肥 230039；
3. 中国矿业大学 信息与电气工程学院，江苏 徐州 221116；4. 合肥师范学院 计算机学院，合肥 230001；
5. 合肥综合性国家科学中心 人工智能研究院，合肥 230071)

**摘 要:** 频繁高效用项集挖掘是数据挖掘的一项重要任务,挖掘到的项集由支持度和效用这2个指标衡量. 在一系列用于解决这类问题的方法中,进化多目标方法能够提供1组高质量解以满足不同用户的需求,避免传统算法中支持度和效用的阈值难以确定的问题. 但是已有多目标算法多采用0-1编码,使得决策空间的维度与数据集中项数成正比,因此,面对高维数据集会出现维度灾难问题. 鉴于此,设计一种项集归减策略,通过在进化过程中不断对不重要项进行归减以减小搜索空间. 基于此策略,进而提出一种基于项集归减的高维频繁高效用项集挖掘多目标优化算法(IR-MOEA),并针对可能存在的归减过度或未归减到位的个体提出基于学习的种群修复策略用以调整进化方向. 此外还提出一种基于项集适应度的初始化策略,使得算法在进化初期生成利于后期进化的稀疏解. 多个数据集上的实验结果表明,所提出算法优于现有的多目标优化算法,特别是在高维数据集上.

**关键词:** 多目标优化；进化算法；归减策略；修复策略

**中图分类号:** TP273 **文献标志码:** A

**DOI:** 10.13195/j.kzyjc.2021.1900 **开放科学(资源服务)标识码(OSID):**

**引用格式:** 张磊, 李柳, 杨海鹏, 等. 基于项集归减的高维频繁高效用项集挖掘多目标优化方法 [J]. 控制与决策, 2023, 38(10): 2832-2840.

# An itemset reduction based multi-objective evolutionary algorithm for mining high-dimensional frequent and high utility itemsets

*ZHANG Lei*[1], *LI Liu*[1], *YANG Hai-peng*[1], *SUN Xiang*[1], *CHENG Fan*[2†], *SUN Xiao-yan*[3], *SU Yu*[4,5]

(1. School of Computer Science and Technology，Anhui University，Hefei 230039，China；2. School of Aritificial Intelligence，Anhui University，Hefei 230039，China；3. School of Information and Control Engineering，China University of Mining and Technology，Xuzhou 221116，China；4. School of Computer，Hefei Normal University，Hefei 230001，China；5. Institute of Artificial Intelligence，Hefei Comprehensive National Science Center，Hefei 230071，China)

**Abstract:** Frequent and high utility itemset mining is an important task in data mining, and the mined itemsets are measured by two metrics, support and utility. Among a series of methods used to solve such problems, evolutionary multi-objective methods provide a set of high-quality solutions to meet the needs of different users, as well as avoiding the problem of difficulty in determining the thresholds of support and utility in traditional algorithms. The existing multi-objective algorithms are encoded with 0-1 and the dimensionality of the decision space is proportional to items in the dataset. As a result, the curse of the dimensionality problem can occur in high-dimensional datasets. Therefore, this paper designs an itemset reduction strategy to reduce the search space by reducing the unimportant items. According to this strategy, the paper proposes a high-dimensional frequent and high utility multi-objective evolutionary algorithm for itemset mining based on itemset reduction (IR-MOEA), where a learning-based population restoration strategy is proposed to adjust the evolutionary direction for over-reduced or under-reduced individuals. In addition, an initialization strategy is proposed to generate sparse solutions that facilitate evolution. Finally, experimental results on datasets show that this algorithm outperforms the existing state-of-the-art multi-objective optimization algorithms for mining frequent and high utility itemsets, especially on high-dimensional datasets.

**Keywords:** multi-objective optimization problems；evolutionary algorithm；reduced strategy；repairing strategy

**收稿日期:** 2021-11-04；**录用日期:** 2022-05-17.

**基金项目:** 国家自然科学基金项目(61976001, 62076001, 61876184)；安徽省教育厅高校优秀人才支持计划重点项目(gxyqZD2021089)；安徽省自然科学基金项目(2008085QF309)；安徽省高校协同创新项目(GXXT-2020-050).

**责任编委:** 巩敦卫.

†通讯作者. E-mail: chengfan@mail.ustc.edu.cn.

*本文附带电子附录文件,可登录本刊官网该文“资源附件”区自行下载阅览.

## 0 引 言

项集挖掘(itemset minging, IM)的目标是在给定数据集中找到一些新颖的、难以直接观察的、有用的项集. 频繁项集挖掘(frequent itemset minging, FIM)是项集挖掘领域中的一个重要研究方向[1], 可用于发现项集间的关联关系, 从而用于决策. 若一个项集(itemset)的支持度, 即其在数据集中出现频率大于或等于用户指定的阈值min_supp, 则可称该项集为频繁项集. 在过去的几十年里, 研究人员提出了许多优秀的频繁项集挖掘算法[2-3]. 频繁项集广泛应用于购物篮分析[4]、个性化推荐[5]以及判断基因序列与疾病的关系[6]等领域. 但是由于频繁项集挖掘没有考虑项的附加内容(如购买数量和利润), 会造成部分信息的遗漏. 对此, Yao等[7]提出高效用项集挖掘(high utility itemset minging, HUIM)这一概念. 当项集的权重即效用值大于或等于用户指定的阈值min_util时称为高效用项集. 此类算法大多通过提出特殊存储结构和剪枝策略以高效挖掘高效用项集[8,9-13]. HUIM虽然弥补了FIM存在的遗漏部分信息等不足, 但是挖掘到的高效用项集可能存在频繁度低这一问题.

FIM和HUIM分别只考虑了单个因素, 因此采用现有算法挖掘到的项集或是频繁不高效或是高效不频繁的, 这便造成了推荐的项集不能引起用户兴趣的问题. 对此, Zhang等[14]于2016年首次提出了一个通用的框架, 通过将支持度和效用加权为一个目标, 再用传统方法进行优化. 该方法虽然能够解决频繁高效用项集挖掘问题, 但是存在一个难点, 即算法中需提前给定频繁和高效用对应的阈值、支持度和效用的加权参数, 但是准确设置合适的阈值和参数本身是个困难问题. 为此, Zhang等[15]于2018年提出的FHUI-MOEA算法中, 首次将频繁高效项集挖掘与多目标优化相结合, 将项集支持度和效用作为2个目标放在一个框架中同时进行优化, 可在一次运行中为决策者推荐多个项集, 且可避免阈值难以确定这一问题. 在2019年, Cao等[16]在FHUI-MOEA的基础上提出了CP-MOEA算法, 该算法提出了一种基于封闭项集性质的多目标进化方法以进行项集的挖掘. 其中, 设计了两种基于闭集的更新策略用于提高频繁高效项集挖掘的质量. 另外, 2020年Tian等[17]提出了一个稀疏大规模多目标优化算法框架SparseEA, 并提出一个稀疏的初始化策略和2个高效的进化算子, 可用于解决频繁高效用项集挖掘多目标优化问题.

以上基于进化多目标优化的项集挖掘算法虽然可自适应地解决阈值和参数的设置问题, 但是这些算法多采用0-1编码, 使得决策空间的大小与项数呈几何级关系, 随着数据集的维度增大, 决策空间呈指数倍增长, 从而造成维度灾难问题, 不能挖掘到高质量解. 因此, 如何设计针对高维数据集进行频繁高效用项集挖掘的高效算法是当前研究所缺乏关注的. 基于此, 本文提出基于项集归减的高维频繁高效用项集挖掘多目标优化算法IR-MOEA. 算法中提出的归减策略可在进化过程中, 通过计算个体中项的关联度和重要度, 判断出待归减的比较不重要项来减小搜索空间. 主要内容如下.

1) 针对高维频繁高效用模式挖掘问题, 提出一种项集归减策略, 用于归减种群中每个个体的不相关项, 从而通过减小搜索空间来加速种群的收敛. 具体而言, 在归减阶段, 种群中非精英个体中的大部分项均有一定概率会被归减掉, 归减概率综合考虑了项在数据集中的重要度和项在种群中的重要度2个方面因素.

2) 在所提出归减策略的基础上, 本文提出一个基于项集归减的频繁项集挖掘多目标优化算法IR-MOEA. 在进化过程中使用所提出种群归减策略来减少搜索空间; 同时提出一个修复策略来修复可能存在的过度归减或未归减到位的项. 另外, 本文通过所提出基于项集适应度的初始化策略在进化初期生成1组稀疏解, 以便后期更好地进化.

3) 本文在8个数据集上与其他3个基于多目标优化的项集挖掘算法进行比较, 实验结果表明, IR-MOEA算法的结果优于另外3个对比算法, 尤其是在高维数据集上, 优势更为明显.

## 1 背景介绍

本节介绍了与本文相关的研究背景和频繁高效用项集挖掘问题的基本概念和定义.

事务数据集$D$由$m$条事务组成, 每个事务均有一个不重复的标识符, 可记作$T_{\mathrm{id}}(1 \leqslant \mathrm{id} \leqslant m)$. 事务由1组项组成, 不同事务包含的项可能不同, 假定数据集中所有事务包含的项共有$n$个, 这$n$个项记作$\{I_1, I_2, \ldots, I_n\}$. 在频繁高效用项集挖掘问题中, 事务中所有项均有对应的权重. 若事务中有项, 则对应权重为其效用, 若无, 则效用为0. 同一项在不同事务中的权重可能不同, 如表1中数据集$D$包含5个事务$\{T_1, T_2, T_3, T_4, T_5\}$, $\{a, b, c, d, e, f, g, h, i\}$9项, 事务$T_1$中项$c$的效用为3, 事务$T_2$中项$c$的效用为5.

项集是由1个及以上项组成的项集合, 若事务$T_i$包含项集$X$, 则$T_i$可称为项集$X$的一个支持事务,

**表1 数据集$D$示例**

| $T_{\text{id}}$ | Transaction | $W(T_i)$ |
|---|---|---|
| $T_1$ | $a$[1] $b$[0] $c$[3] $d$[0] $e$[4] $f$[0] $g$[5] $h$[0] $i$[4] | 17 |
| $T_2$ | $a$[0] $b$[0] $c$[5] $d$[1] $e$[0] $f$[2] $g$[0] $h$[0] $i$[3] | 11 |
| $T_3$ | $a$[4] $b$[1] $c$[1] $d$[0] $e$[2] $f$[0] $g$[2] $h$[0] $i$[2] | 12 |
| $T_4$ | $a$[0] $b$[1] $c$[0] $d$[1] $e$[0] $f$[1] $g$[0] $h$[0] $i$[2] | 5 |
| $T_5$ | $a$[0] $b$[3] $c$[1] $d$[0] $e$[0] $f$[0] $g$[0] $h$[0] $i$[1] | 5 |

项集$X$的所有支持事务的集合记作$D_X$. 项集$X$的支持度supp为数据集中其支持事务数与事务总数$m$的比值(如下式所示). 在频繁项集挖掘问题中,支持度被用于衡量项集是否频繁,若项集$X$的支持度$\text{supp}(X)$大于给定阈值min_supp,则可称其为频繁项集. 如表1中共有5条事务,假定$\text{min_supp}=0.5$, $X=\{c,i\}$, 易知$D_X=\{T_1,T_2,T_3,T_5\}$,代入下式得到$\text{supp}(X)=0.8>0.5$,因此$\{c,i\}$为频繁项集. 项集$X$的支持度公式如下所示:

$$\text{supp}(X)=\frac{|D_X|}{m}. \tag{1}$$

其中:$m$为数据集中事务总数,$D_X$为项集$X$的支持事务集合,支持度取值范围为$0\sim1$.

在频繁高效用项集挖掘中,项集是否高效由效用值衡量,若效用值大于给定阈值min_util,则可认为其是一个高效用项集. 项集$X$的效用值$\text{util}(X)$为包含项集的所有事务中$X$对应的效用和与数据集总效用和的比值(如下式所示). 如表1数据集中,假定效用阈值$\text{min_util}=0.5$, $X=\{c,i\}$,代入如下项集$X$的效用值公式:

$$\text{util}(X)=\frac{\sum\limits_{T_i\subseteq D,I_j\subseteq X}W(T_i,I_j)}{\sum\limits_{T_i\subseteq D}W(T_i)}, \tag{2}$$

得到$\text{util}(X)=(W(X,T_1)+W(X,T_2)+W(X,T_3)+W(X,T_5))/\text{util}(D)=(7+8+3+2)/50=0.4$,因此$\{c,i\}$为低效用项集. 按同样的计算方式,若$X=\{a,c,e,g,i\}$,得到$\text{util}(X)=0.56$,则$\{a,c,e,g,i\}$为高效用项集. 式(2)中:$W(T_i,I_j)$为事务$T_i$中项$I_j$的效用;$W(T_i)$为事务$T_i$中所有项的效用和,效用取值范围为$0\sim1$.

频繁高效项集可用上述2个指标(式(1)和式(2))衡量,为了避免阈值难以设定,Zhang等[15]首次将这2个指标当作2个目标,并将频繁高效项集挖掘问题转化为多目标优化问题,再进行项集挖掘. 问题定义如下:

$$\max\ F(X)=\{\text{supp}(X),\text{util}(X)\}^{\text{T}}. \tag{3}$$

其中:$X$为一个项集,$\text{supp}(X)$和$\text{util}(X)$分别为$X$的支持度和效用值. 实验结果表明,进化多目标方法适用于解决频繁高效用项集挖掘问题,其后出现的更多基于MOEA的频繁高效用算法进一步显示了良好的性能.

上述基于进化多目标优化的方法,均采用0-1编码,这使得项集挖掘的时间复杂度和空间复杂度会随着数据集规模(维度)呈指数倍增长. 本文研究的是高维频繁高效用项集挖掘问题,因此,为了更快更好地进行项集挖掘,本文引用了事务加权效用(transaction weighted utility, TWU)这一指标调整搜索方向[18]. 项集$X$的TWU为$X$的支持事务中所有项的效用之和(如下式所示),项集$X$对应的TWU不低于其效用值. 因此,若项集$X$的TWU值较低,其为低效用项集的概率便比较大,可利用此性质对搜索方向作出一定调整. TWU的计算公式如下所示,表1中:项$i$对应$\text{TWU}(\{i\})=W(T_1)+W(T_2)+W(T_3)+W(T_4)+W(T_5)=17+11+12+5+5=50$,项$f$对应$\text{TWU}(\{f\})=11+5=16$,则包含项$i$的项集比包含项$f$的项集为高效用项集的概率高,应重点对前者进行搜索. 项集$X$的TWU定义如下:

$$\text{TWU}(X)=\sum_{T_i\subseteq D,X\subseteq T_i}W(T_i), \tag{4}$$

其中$W(T_i)$为事务$T_i$中所有项的效用和.

## 2 基于归减的高维频繁高效用项集挖掘多目标优化方法

本节将详细介绍所提出针对高维数据集进行频繁高效用项集挖掘的IR-MOEA算法,按顺序依次阐述整个算法的框架、所提出初始化策略、项集归减策略和对应的项集修复策略.

### 2.1 算法框架

所提出IR-MOEA算法采用NSGA-II[19]框架,用于高效地挖掘频繁高效用项集. 算法的总体框架如算法1所示,主要包括以下步骤:1)利用所提出初始化策略生成一个规模为$N$的种群(line 1),并定义了一个指标$II$计算不同个体中各项的重要度,用于引导种群进化(line 2). 2)对初始化生成的种群进行gen次更新(line 3):为了更好更快地挖掘到频繁高效用项集,每隔radix代使用所提出项集归减策略来归减种群的每个个体中的无关项(lines 4 ~ 5). 随后针对部分过度归减或未归减到位的个体通过所提出项集修复策略进行修复(line 6). 其余代的进化中使用NSGA-II中的进化算子对种群进行交叉变异(line 9). 在得到每一代更新过的种群后,通过NSGA-II中的环境选择策略对原种群以及更新后的种群进行选择得到新种群(line 12),再将新种群代入指标$II$

项集重要度公式,即可得到新种群中各项对应的重要度(line 13),并利用其引导后续的种群更新.

**算法1**　IR-MOEA算法框架.

输入:数据集度DB,种群规模$N$,进化代数gen,归减间隔radix,调整参数$\gamma$;

输出:最终种群$P$.

1. $P^1 \leftarrow$ Initialization(DB, $N$) //种群初始化策略
2. $II \leftarrow$利用DB和$P^1$的信息计算项重要度
3. for $t=1$ to gen do
4. if $t$ mod radix$\neq 0$ then
5. $P_1^t \leftarrow$ PopulationReduction($II, P^t$, DB, $\gamma$) //种群归减策略
6. $P_2^t \leftarrow\leftarrow$ PopulationRepair($II, P^t, N$) //种群修复策略
7. $P_u^t \leftarrow P^t \bigcup P_1^t \bigcup P_2^t$
8. else
9. $P_1^t \leftarrow$交叉变异
10. $P_u^t \leftarrow P^t \bigcup P_1^t$
11. end if
12. $P^{t+1} \leftarrow$对$P_u^t$进行环境选择
13. $II \leftarrow$利用DB和$P^{t+1}$的信息更新项重要度
14. $t++$
15. end for
16. 输出非支配解

### 2.2　种群初始化

高维频繁高效项集挖掘问题是一个稀疏的问题,因此,产生稀疏的解更有利于种群的收敛. 本文提出了一个稀疏初始化策略,主要分为2个步骤: 1) 确定个体项数; 2) 确定应选哪些项(具体见算法2). 第1个步骤中,首先确定个体项数$k$的上限(line 1),即数据集中最长事务的长度. 再循环$N$次生成$N$个个体(line 2),每次在0和上限范围内随机取值作为个体项数,以确保解的稀疏性(line 3);然后,根据项的支持度supp和效用值util计算出适应度值,并通过二元锦标赛[20]选出较好的$k$项(lines 4 ~ 7). 假定个体最大项数为7,在0 ~ 7范围内,随机生成$p_1$的个体项数假设为3,通过二元锦标赛不重复地从$a$、$b$、$c$、$d$、$e$、$g$选出$a$、$c$和$g$三项,便可得到个体$p_1=\{1,0,1,0,0,0,1,0,0\}$.

**算法2**　Initialization.

输入:数据集DB,种群规模$N$;

输出:初始种群$P$.

1. max_len $\leftarrow$计算最大事务长
2. for $i=1$ to $N$ do
3. $k \leftarrow$从1 ~ max_len中随机取一个数作为个体项数
4. for $j=1$ to $k$ do
5. $I_1, I_2 \leftarrow$随机选两项
6. $p_i \leftarrow p_i \bigcup$二元锦标赛($I_1, I_2$)
7. end for
8. end for

### 2.3　项集归减策略

本文提出了一个项集归减策略,提出此策略的动机在于: 一方面,$n$维数据集对应的搜索空间为$O(2^n)$,其会随着维度的增大呈指数倍增长,形成维度灾难;另一方面,在高维频繁高效用挖掘问题中,种群的每个个体在进化初期均有很大概率有冗余项或不相关项. 针对这两点,本文将非支配个体视为精英个体,通过所提出归减策略对种群中非精英个体进行调整来减小搜索空间,加速种群收敛.

归减部分的主要步骤如算法3所示. 首先,对种群中的个体进行非支配排序,根据排序结果将第1前沿面的个体存入精英个体集Elites中,其余视为非精英个体(line 1). 然后,对于种群内每个非精英个体(假定为$p_k$)进行归减(lines 2 ~ 3),第1步,找到$p_k$中所有非0项按TWU降序存入List(line 4),并将集合中最优项List(1)存入保留项集reList中(line 5),依次计算reList与$p_k$中未判断项(假定为List($j$))间的相关度Corr(reList, List($j$))(line 7),若相关度较高(line 8),表明{reList, List($j$)}为频繁项集的概率较高,则将List($j$)存入reList再与其他项进行相关度判断(line 9);若关联度较低且不为0,则需计算项List($j$)在个体中的翻转概率FP$^t$($p_k$, List($j$))(line 11),用于判断项是否需要被归减. 若翻转概率高需将此项归减(line 13),翻转概率低,则将此项加入保留项集reList中(line 15);若完全不相关,则直接对其进行归减(line 18). Corr(reList, List($j$))越大,项与项间相关度越高,FP$^t$($p_k$, List($j$))越大,项的重要度越高,因此,项被归减(由1置0)的概率随着这2个值的增大而降低.

**算法3**　PopulationReduction.

输入:项的重要度$II$,第$t$代种群$P^t$,数据集DB,调整参数$\gamma$;

输出:归减后的种群$P_1^t$.

1. Elites $\leftarrow$对$P^t$进行非支配排序得到的第1前沿面解
2. for $k=1$ to $|P^t|$ do
3. if $p_k \in$ Elites then
4. List $\leftarrow$个体$p_k$中的非0项(按TWU降序排列)
5. reList $\leftarrow$ List(1)
6. for $j=2$ to |List| do
7. 计算相关度Corr(reList, List($j$))
8. if Corr(reList, List($j$))$>0.5$ then

```
9.              reList = reList ⋃ List(j)
10.         else if Corr(reList, List(j)) > 0 then
11.             计算翻转概率 FP^t(p_k, List(j))
12.             if FP^t(p_k, List(j)) ⩾ rand( ) then
13.                 将 p_k 中项 List(j) 置 0
14.             else
15.                 reList ← reList ⋃ List(j)
16.             end if
17.         else
18.             将 p_k 中项 List(j) 置 0
19.         end if
20.       end for
21.    end if
22. end for
```

通过上述归减策略,能够有效地减少低相关项,从而调整搜索方向缩小搜索空间,但是在实现该策略时,有一个关键的挑战:个体中需要归减(删除)哪些项. 由于最优事务的大小是未知的,个体中具体归减的项数要根据后面的归减原则判断而得,其取值范围为$0 \sim (|\text{List}| - 1)$($|\text{List}|$为待归减个体$p_k$中非0项的数量). 为确定归减项制定以下依据:在确定好归减的比较次序后,根据项与项间的相关性以及项自身的重要性判断是否归减较差项. 项的归减次序由事务加权效用(TWU)确定. 若某一项的TWU较大,则意味着其具有较高效用的概率也较大. 基于这一原则,若2个项之间存在冲突,则优先保留TWU较大的项.

通过项的TWU确定归减次序后,介绍归减部分具体过程:首先找出种群中所有非精英解;然后依次比较非精英解中项与项间的相关性. 根据相关性可分为3种类型:1)若比较的两项相关度很高,表明它们为频繁项集的概率较高,则二者皆保留,并将其看作一个整体,再与其他项进行后续比较. 2)若与较优项的相关性低且并非完全不相关时,则需要对较差项自身重要性进行判断. 通过其重要度和当前进化代数可计算出一个由1翻转至0的概率,重要度越高翻转的概率越小,被归减的概率也越低;相应地,重要度越低,则其归减概率越高. 3)若相关度为0,则直接归减.

本文定义了一个新的指标$\text{Corr}(\text{reList}, \text{List}(j))$来衡量项间相关度. 形式上,个体中的保留项集reList和未判断项(假定为$\text{List}(j)$)的相关度$\text{Corr}(\text{reList}, \text{List}(j))$可定义为

$$\text{Corr}(\text{reList}, \text{List}(j)) = \frac{\text{supp}(\text{reList} \bigcup \text{List}(j))}{\text{supp}(\text{List}(j))}. \quad (5)$$

其中:reList为已判断且保留的项的集合,$\text{List}(j)$为未判断项集中TWU最高项.

在考虑项的归减时,除了项间关联度,还应考虑项自身重要度. 尽管一些待归减项与较优项相关性较低,但是它们自身有高重要性,很可能与其他项形成较优的组合,因此会按一定概率保留下来,保留的主要依据为个体$p_k$中项$\text{List}(j)$的重要度$II(p_k, \text{List}(j))$(如下式所示). 其考虑了2个方面的因素:项在种群中的重要度DS和项在数据集中的重要度SU. 具体如下所示:

$$II(p_k, \text{List}(j)) = \frac{\text{DS}(p_k, \text{List}(j)) + \text{SU}(\text{List}(j))}{2}. \quad (6)$$

其中:$\text{DS}(p_k, \text{List}(j))$为个体$p_k$中项$\text{List}(j)$在种群中的重要度,$\text{SU}(\text{List}(j))$为项$\text{List}(j)$在数据集中的重要度.

以DS衡量个体在种群中的重要度的依据如下:若一个个体支配的个体数量越多,则这个个体中所选项的重要度也会越高. 因此,个体$p_k$在整个种群中的重要度$\text{DS}(p_k, \text{List}(j))$由$p_k$支配的个体数量、种群规模和个体中项的状态计算而得,即

$$\text{DS}(p_k, \text{List}(j)) = \frac{\text{DS}(p_k) \times p_k(\text{List}(j))}{N}. \quad (7)$$

其中:$\text{DS}(p_k)$为个体$p_k$支配的个体数量;$N$为种群规模;$p_k(\text{List}(j))$为个体$p_k$中项$\text{List}(j)$的状态,值为1表示个体$p_k$中选中项$\text{List}(j)$,值为0表示未选择,取值范围为$0 \sim 1$,个体越重要对应的$\text{DS}(p_k, \text{List}(j))$越大.

另一个因素SU综合考虑了项在数据集中的supp和util,用于衡量项在数据集中的重要度. 若$\text{List}(j)$为频繁高效用项,则包含$\text{List}(j)$的项集为频繁高效用项集的概率便越大. $\text{SU}(\text{List}(j))$公式如下所示:

$$\text{SU}(\text{List}(j)) = \frac{\text{supp}(\text{List}(j)) + \text{util}(\text{List}(j))}{2}. \quad (8)$$

其中:$\text{supp}(\text{List}(j))$和$\text{util}(\text{List}(j))$分别为项$\text{List}(j)$的支持度和效用值;$\text{SU}(\text{List}(j))$取值范围为$0 \sim 1$,项越重要对应的$\text{SU}(\text{List}(j))$越大.

以一个示例演示不同个体中所有项对应的重要度的计算. 假定种群的规模为4,个体$p_1$在种群中支配的个体数为3,将其代入式(7),可得到$p_1$中所选项对应的重要度$\text{DS} = 3/4 = 0.75$,未选项的DS默认为0. 假定9个项($a \sim i$)在数据集中的重要度$\text{SU} = \langle 0.72, 0.31, 0.63, 0.23, 0.42, 0.55, 0.44, 0.00, 1.00 \rangle$,代入式(6)可得到$II(p_1, a) = 0.73$, $II(p_1, b) = 0.16$等.

个体$p_k$中项$\text{List}(j)$的$II(p_k, \text{List}(j))$越大,其重要度越大,对应的归减可能性应越低. 但是不同的进化代数的最优个体与其他个体的差距并不是随代数均匀变化的,因此除重要度外,归减时还要考虑进化

代数. 基于此,本文提出一个综合考虑$II(p_k, \text{List}(j))$和进化代数$t$的指标:翻转概率$\text{FP}^t(p_k, \text{List}(j))$. 在进化早期,种群中的个体与最终个体相差较大,应作大幅度调整,所以翻转概率应作增大处理. 相应地,进化后期种群趋于收敛,翻转概率应作减小处理. 翻转概率如下式所示:

$$\text{FP}^t(p_k, \text{List}(j)) = \frac{\exp^{-\gamma \cdot t}}{1 + \exp^{II(p_k, \text{List}(j)) - \overline{II(p_k)}}}. \quad (9)$$

其中:$II(p_k, \text{List}(j))$为个体$p_k$中项$\text{List}(j)$的重要度,$\overline{II(p_k)}$为个体$p_k$中所有项的平均个体重要度,$\gamma$为调整因子,$t$为当前代数. 个体中项的重要度越小,进化代数越靠前,对应的翻转概率越大,被归减的概率便越高.

图1为个体$p_2 = \langle 1,1,0,0,1,1,1,0,1 \rangle$的归减过程,以演示归减策略的主要步骤. $p_2$中被选项$a$、$b$、$e$、$f$、$g$、$i$对应的$\text{TWU} = \langle 30, 17, 25, 26, 27, 50 \rangle$. 将被选项按TWU降序排列可得到项的相关度比较次序为$i$、$a$、$g$、$f$、$e$、$b$,如图1(a)所示. 图1(b)中:$\text{Corr}(\{i\}, a) = 0.5$属于第2类,需通过较差项$a$的重要度计算出其翻转概率$\text{FP}^t(p_2, a) = 0.2$,假定其小于随机数(如0.3),则对其进行保留;并将项集$\{i, a\}$看作一个整体与项$g$进行相关度比较,$\text{Corr}(\{i, a\}, g) = 0.8$属于第1类,相关度极高所以保留$g$,如图1(c)所示;图1(d)中:计算得到$\text{Corr}(\{i, a, g\}, f) = 0.25$属于第2类,且$\text{FP}^t(p_2, f) = 0.67$,假定其大于随机数(如0.5),则因$f$与其他项关联度和自身重要度均不高,对其进行归减;图1(e)中情况类似图1(c),保留项$e$;图1(f)中:$\text{Corr}(\{i, a, g, e\}, b) = 0$属直接归减. 因此,可得到归减后的个体$P_2' = \langle 1,0,0,0,1,0,1,0,1 \rangle$.

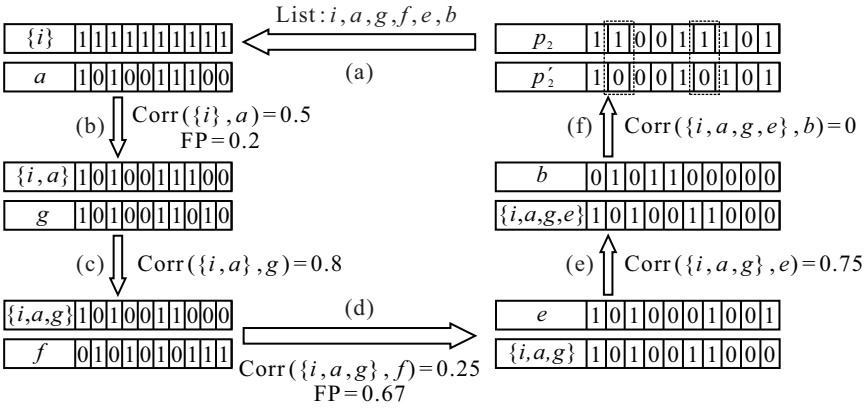


**图 1　种群中个体归减的示例**

### 2.4　项集修复策略

虽然所提出归减策略可大大提高搜索效率,但是可能存在过度归减或归减不到位的情况,从而导致产生低质量的解或陷入局部最优. 针对此问题,本文设计了一个基于学习的个体修复策略,对归减后的待修复个体进行进化指导. 对于随机不重复选取的2个父代个体,仅针对个体中0-1状态不同的项进行修复. 下文将详细介绍这一策略.

修复部分的主要思想如算法4所示. 随机不重复地选取2个个体$p_1$和$p_2$,通过适应度确定待修复个体,再找出个体中状态不同的项的集合List($p_1$中为1,$p_2$中为0或$p_1$中为0,$p_2$中为1的项). 之后计算List中项(假定为$\text{List}(j)$)对应的重要度$II_{\text{Rep}}(\text{List}(j))$来综合考虑项在整个种群中的重要度从而计算修复概率. 重要度越高,为1的概率便越大.

**算法4**　PopulationRepair.

输入: 种群归减重要度$II$,归减后的第$t$代种群$P_1^t$,种群规模$N$;

输出: 修复后的种群$P_2^t$.

```
1.  for i=1 to N/2 do
2.      P1, P2 ← 从 P1^t 中随机不放回地取2个个体
3.      Pb, Pw ← 根据适应度判断出待修复个体 // Pw 为待修复个体
4.      List ← 找2个体中0-1状态不同的项
5.      for j=1 to |List| do
6.          计算修复概率 RP^t(Pk, List(j))
7.          if RP^t(Pk, List(j)) ≥ rand( ) then
8.              调整 Pw(Ij) 的0-1状态
9.          end if
10.     end for
11. end for
```

具体操作如下:通过个体的supp和util计算适应度值,适应度值大的个体更优,较优个体可用于指导另一个体修复. 若某项在2个个体里同为0或同为1,则不需要改变它;若某项在2个个体中被选状态不同,则较差个体中的这些项需要按一定概率向较优个体学习. 本文定义了一个新的指标$II_{\text{Rep}}(\text{List}(j))$进一

步衡量每个项的重要度. 再将重要度结合进化代数计算修复概率$\mathrm{RP}^t(p_k,\mathrm{List}(j))$ . $II_{\mathrm{Rep}}(\mathrm{List}(j))$的公式如下所示:

$$II_{\mathrm{Rep}}(\mathrm{List}(j))=\frac{\dfrac{\sum\limits_{i=1}^{N}II(p_i,\mathrm{List}(j))}{N}+\dfrac{\mathrm{TWU}(\mathrm{List}(j))}{n}}{2}. \tag{10}$$

其中: $\mathrm{List}(j)$为待修复项, $II(p_i,\mathrm{List}(j))$为个体$p_i$中项$\mathrm{List}(j)$的重要度, $N$为种群中个体数量, $n$为数据集中数量. 将$II_{\mathrm{Rep}}(\mathrm{List}(j))$代入如下修复概率公式中便可计算得到$\mathrm{List}(j)$的修复概率$\mathrm{RP}^t(p_k,\mathrm{List}(j))$:

$$\begin{aligned}&\mathrm{RP}^t(p_w,\mathrm{List}(j))=\\&\alpha+(-1)^{\alpha}\cdot II_{\mathrm{Rep}}(\mathrm{List}(j))\cdot\exp^{\Psi(b,w,t)},\\&\Psi(b,w,t)=\\&\frac{1}{\sqrt{2}}\cdot\left\{\left(\Phi_{(b>w)}-\frac{1}{5}\right)\cdot\left[\alpha+(-1)^{\alpha}\cdot\frac{t}{200}\right]\right\}.\end{aligned} \tag{11}$$

其中: 待修复项为1和为0时调整公式存在部分区别, 由$\alpha$控制, $\alpha=\Phi_{(b>w)}$, 若$p_b$为0且$p_w$为1, 则$\alpha$为1, 反之, $\alpha$为0; $II_{\mathrm{Rep}}(\mathrm{List}(j))$为项$I_j$的修复重要度; $t$为当前进化代数. 代数越靠前, 修复重要度越高, 修复概率越大, 此项为1的概率越大.

以一个例子说明修复策略的主要步骤. $p_2$、$p_3$分别为第2个和第3个个体, 数据集中有9个项$(a\sim i)$. 假定$p_3$优于$p_2$, 则利用$p_3$引导$p_2$进化. 根据状态判断, 可找到3个待修复项$(e、f、g)$. 通过计算得到它们的$II_{\mathrm{Rep}}(\cdot)=\langle 0.24,0.372\,5,0.25\rangle$和TWU的次序为$\langle 4,3,7\rangle$. 代入式(11)后可得到修复概率$\mathrm{RP}^t(p_w,\mathrm{List}(j))$为$\langle 0.76,0.23,0.34\rangle$. 假定随机数为0.5, 只有项$e$翻转概率大于随机数, 则只需对其进行修复(从1修复为0).

## 3 实验和分析

本节中, 将IR-MOEA算法与3个现有的多目标优化算法在8个数据集(4个真实数据集和4个人工数据集)上进行了HV[20]的比较, 通过挖掘到的高质量解以验证算法在高维频繁高效用项集挖掘问题上的有效性. HV表示由解集中的个体与参考点在目标空间中所围成的超立方体的体积, 能够对解集的收敛性进行评价. 接下来将具体分析实验设置和实验结果, 并对算法中所提出策略作有效性验证.

### 3.1 实验设置

#### 3.1.1 对比算法

本文将IR-MOEA的性能与FHUI-MOEA[15]、CP-MOEA[16]和SparseEA[17]这3种代表性进化多目标算法进行比较. 其中, 前2个算法是现有基于EA的频繁高效项集挖掘算法. FHUI-MOEA首次提出将支持度和效用放于一个框架中转化为多目标问题, 并提出一个多目标算法以解决此问题. CP-MOEA设计了两种基于闭集的更新策略用于提高频繁高效项集挖掘的质量. SparseEA是新近提出的一个高效稀疏大规模进化算法, 而高维频繁高效用项集挖掘本身也是一种稀疏大规模优化问题, 对此, 本文将其引入作为第3个对比算法. 为了公平比较, 对于所有算法, 本文将规模和进化代数均设为统一值: 种群规模为100, 进化代数为200. 其余参数对所有对比算法均采用了原论文中推荐的参数值, 各算法的详细参数如表2所示. 所有算法的实验结果均为每个算法在每个数据集上独立运行20次, 记录最优值、平均值而得到的. 实验的运行环境为Intel (R) Core (TM) i5-4 460 CPU 8 GB RAM, Windows 10操作系统.

**表 2 算法和参数设置**

| No. | algorithm | parameter |
|---|---|---|
| 1 | FHUI-MOEA[15] | cp = 1.0, mp = 0.01 |
| 2 | CP-MOEA[16] | radix = 10, minconf = 0.8 |
| 3 | SparseEA[17] | cp = 1.0, mp = $1/n$ |
| 4 | IR-MOEA | radix = 5, $\gamma$ = 0.05 |

#### 3.1.2 数 据 集

本文在4个已知且被广泛使用的真实公共数据集和4个人工数据集上验证IR-MOEA算法性能. 数据集的维度为76～30 000. 表3为所采用数据集的详细特征(即事务数、总项数、平均项数、最大项数以及稀疏度). 其中数据集按总项数由小至大排列.

**表 3 8个数据集的信息**

| datasets | # trans | # items | ave.len | max.len | sparsity |
|---|---|---|---|---|---|
| Chess | 3 196 | 76 | 37 | 37 | 0.486 8 |
| Connect_50 % | 16 890 | 119 | 23 | 23 | 0.193 3 |
| Pumsb_10 % | 4 900 | 7 117 | 74 | 74 | 0.010 4 |
| Retail_10 % | 3 300 | 12 479 | 10.3 | 74 | 0.000 8 |
| Synthesis 1 | 500 | 15 000 | 100 | 122 | 0.006 7 |
| Synthesis 2 | 500 | 20 000 | 150 | 184 | 0.007 5 |
| Synthesis 3 | 500 | 25 000 | 200 | 214 | 0.008 0 |
| Synthesis 4 | 500 | 30 000 | 300 | 331 | 0.010 0 |

### 3.2 实验结果

#### 3.2.1 HV值的对比分析

表4为IR-MOEA与3个对比算法在8个维度不同数据集上项集挖掘的HV的最大值和平均值的比较结果. 本文采用了Wilcoxon rank sum test分析算法性能的统计差异($\alpha=0.05$), 表中的符号“$+,-,\approx$”分别为对比算法实验结果显著地优于、差于以及接近于IR-MOEA所得结果. 由表4可见, 在绝大多数数据

集上IR-MOEA的效果均优于其他算法,不仅在20次运行中的HV最优值为4个算法中最高的,其稳定性和平均HV同样也是最优的(retail数据集太过稀疏,以至于很难找到高质量解). 高维数据集对应的搜索空间极大,搜索到高质量解的难度随数据集维度的增长呈指数倍增长,而IR-MOEA中提出的项集归减策略可缩小搜索空间调整进化方向,使得算法性能得到提高,挖掘到高质量项集. 同时,所提出初始化可在进化初期得到1组稀疏解,对后面的进化有一定提升作用. 另外,虽然SparseEA是一个高效稀疏大规模进化算法,但所提出初始化和对应进化算子的特性不能适用于有极低稀疏度且事务数较少的数据集,因此,在后6个稀疏度为0.01上下的数据集中,经过200代的进化均不能进化出有效解,造成对应HV值为0.

**表 4　IR-MOEA与3个对比算法在真实网络上HV值的对比**

| dataset | Metric | FHUI-MOEA | CP-MOEA | SparseEA | IR-MOEA |
|---|---|---|---|---|---|
| Chess | $HV_{max}$ | 0.179 9 | 0.218 1 | 0.206 6 | **0.227 0** |
| | $HV_{avg}$ | 0.164 2 − | 0.218 1≈ | 0.177 2 − | **0.218 6** |
| Connect_50% | $HV_{max}$ | 0.204 4 | 0.318 7 | 0.258 5 | **0.327 3** |
| | $HV_{avg}$ | 0.170 5 − | 0.302 8 − | 0.155 4 − | **0.311 7** |
| Pumsb_10% | $HV_{max}$ | 0.123 9 | 0.157 6 | 0 | **0.163 0** |
| | $HV_{avg}$ | 0.106 9 − | 0.154 0 − | 0 − | **0.158 4** |
| Retail_10% | $HV_{max}$ | **0.015 2** | **0.015 2** | 0 | **0.015 2** |
| | $HV_{avg}$ | **0.015 2**≈ | **0.015 2**≈ | 0 − | **0.015 2** |
| Synthesis 1 | $HV_{max}$ | 0.190 3 | 0.182 6 | 0 | **0.325 6** |
| | $HV_{avg}$ | 0.168 5 − | 0.177 7 − | 0 − | **0.297 6** |
| Synthesis 2 | $HV_{max}$ | 0.038 6 | 0.047 3 | 0 | **0.067 9** |
| | $HV_{avg}$ | 0.033 4 − | 0.044 8 − | 0− | **0.058 4** |
| Synthesis 3 | $HV_{max}$ | 0.124 9 | 0.318 2 | 0 | **0.419 2** |
| | $HV_{avg}$ | 0.110 2 − | 0.317 9 − | 0− | **0.378 5** |
| Synthesis 4 | $HV_{max}$ | 0.053 3 | 0.104 6 | 0 | **0.164 4** |
| | $HV_{avg}$ | 0.049 6 − | 0.103 4 − | 0 − | **0.133 4** |
| +, −, ≈ | | 0 / 7 / 1 | 0 / 6 / 2 | 0 / 8 / 0 | |

### 3.2.2　策略有效性分析

本文验证IR-MOEA所提出算法的策略有效性,即归减策略、修复策略和初始化策略. 表5为3个算法运行20次平均HV的排名对比,其中,IR-MOEA-redp、IR-MOEA-rep、IR-MOEA-init分别为不使用归减和修复策略的IR-MOEA算法、不使用初始化策略的IR-MOEA算法和不使用修复策略的IR-MOEA算法. 由表5可见,完整算法的实验结果明显优于去掉部分策略的对比算法,在8个数据集中均排名第1,IR-MOEA-redp的效果最差,且其对应的HV远低于其他算法,表明不使用归减和修复策略的算法面对高维数据集无法有效地进行项集挖掘,从而可验证,所提出归减策略是十分有效的. 总结而言,在IR-MOEA中同时使用这3个策略,可得到最优结果,进而验了证所提出策略的有效性. 除此之外,作者还展示了各种算法在8个数据集上的非支配解对比图以及参数敏感性分析,限于篇幅,此略.

**表 5　IR-MOEA与3个变种算法在真实网络上HV值的对比**

| dataset | IR-MOEA-redp | IR-MOEA-rep | IR-MOEA-init | IR-MOEA |
|---|---|---|---|---|
| Chess | 0.211 2 (3) | 0.217 6 (2) | 0.211 0 (4) | **0.218 6 (1)** |
| Connect_50% | 0.284 2 (4) | 0.302 4 (2) | 0.288 1 (3) | **0.311 7 (1)** |
| Pumsb_10% | 0.082 2 (4) | 0.144 3 (3) | 0.155 7 (2) | **0.158 4 (1)** |
| Retail_10% | 0.009 2 (4) | **0.015 2 (1)** | **0.015 2 (1)** | **0.015 2 (1)** |
| Synthesis 1 | 0.019 9 (4) | 0.294 8 (2) | 0.236 5 (3) | **0.297 6 (1)** |
| Synthesis 2 | 0.017 2 (4) | 0.056 3 (2) | 0.054 7 (3) | **0.058 4 (1)** |
| Synthesis 3 | 0.259 1 (4) | 0.375 3 (2) | 0.336 6 (3) | **0.378 5 (1)** |
| Synthesis 4 | 0.068 5 (3) | 0.127 8 (2) | 0.049 6 (4) | **0.133 4 (1)** |
| averank | 3.750 | 2.000 | 2.875 | **1.000** |

## 4　结　论

为了缩小搜索空间来解决高维项集挖掘问题,本文提出了一个项集归减策略,以减少对无关或低相关度项集的探索. 在此基础上提出了一个基于项集归减的项集挖掘算法框架(IR-MOEA),在进化阶段每隔若干代,利用所提出项集归减和修复策略提高算法搜索效率. 另外,算法通过基于项的适应度进行种群初始化,可在进化初期得到1组高质量稀疏解,有利于后续的项集挖掘. 在真实以及合成数据集上的实验结果表明,所提出算法能够取得较好的结果,表明所提出的IR-MOEA算法能够很好地处理高维项集挖掘问题. 所提出的项集归减方法虽然能够处理高维项集问题,但是现实世界中还存在部分大规模高维数据集,在未来将进一步探索如何设计面向大规模高维数据集的项集挖掘多目标优化高效算法.

## 参考文献(References)


[1] Agrawal R, Srikant R. Fast algorithms for mining association rules in large databases[C]. VLDB'94: Proceedings of the 20th International Conference on Very Large Data Bases. Santiago, 1994: 487-499.

[2] 邱剑锋, 武梦雨, 储建军, 等. 面向健康体检数据的多目标Top-$k$频繁模式挖掘方法[J]. 控制与决策, 2023, 38(1): 190-200.
(Qiu J F, Wu M Y, Chu J J, et al. A multi-objective Top-$k$ frequent pattern mining approach oriented for health examination data[J]. Control and Decision, 2023, 38(1): 190-200.)

[3] Han J W, Pei J, Yin Y W. Mining frequent patterns without candidate generation[C]. SIGMOD'00: Proceedings of the ACM SIGMOD International Conference on Management of Data. Dallas, 2000: 1-12.

[4] Hoseini M S, Shahraki M N, Neysiani B S. A new

algorithm for mining frequent patterns in Can Tree[C]. The 2nd International Conference on Knowledge-Based Engineering and Innovation. Tehran, 2015: 843-846.

[5] Kim J K, Cho Y H, Kim W J, et al. A personalized recommendation procedure for internet shopping support[J]. Electronic Commerce Research and Applications, 2002, 1(3/4): 301-313.

[6] Reiter L T, Potocki L, Chien S, et al. A systematic analysis of human disease-associated gene sequences in Drosophila melanogaster[J]. Genome Research, 2001, 11(6): 1114-1125.

[7] Yao H, Hamilton H J, Butz C J. A foundational approach to mining itemset utilities from databases[C]. Proceedings of the SIAM International Conference on Data Mining. Philadelphia, 2004: 482−486.

[8] Yun U, Ryang H, Ryu K H. High utility itemset mining with techniques for reducing overestimated utilities and pruning candidates[J]. Expert Systems with Applications, 2014, 41(8): 3861-3878.

[9] Tseng V S, Wu C W, Shie B E, et al. UP-Growth: An efficient algorithm for high utility itemset mining[C]. KDD'10: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Washington, DC, 2010: 253-262.

[10] Liu M C, Qu J F. Mining high utility itemsets without candidate generation[C]. CIKM'12: Proceedings of the 21st ACM International Conference on Information and Knowledge Management. Maui, 2012: 55-64.

[11] Duong Q H, Liao B, Fournier-Viger P, et al. An efficient algorithm for mining the top-$k$ high utility itemsets, using novel threshold raising and pruning strategies[J]. Knowledge-Based Systems, 2016, 104: 106-122.

[12] 黄坤, 吴玉佳, 李晶. 基于差集的高效用项集挖掘方法[J]. 电子学报, 2018, 46(8): 1804-1814.
(Huang K, Wu Y J, Li J. Mining high utility itemsets using diffsets[J]. Acta Electronica Sinica, 2018, 46(8): 1804-1814.)

[13] Lin J C W, Djenouri Y, Srivastava G, et al. A predictive GA-based model for closed high-utility itemset mining[J]. Applied Soft Computing, 2021, 108: 107422.

[14] Zhang L, Luo P, Chen E H, et al. Revisiting bound estimation of pattern measures: A generic framework[J]. Information Sciences, 2016, 339: 254-273.

[15] Zhang L, Fu G L, Cheng F, et al. A multi-objective evolutionary approach for mining frequent and high utility itemsets[J]. Applied Soft Computing, 2018, 62: 974-986.

[16] Cao H, Yang S S, Wang Q R, et al. A closed itemset property based multi-objective evolutionary approach for mining frequent and high utility itemsets[C]. IEEE Congress on Evolutionary Computation. Wellington, 2019: 3356-3363.

[17] Tian Y, Zhang X Y, Wang C, et al. An evolutionary algorithm for large-scale sparse multiobjective optimization problems[J]. IEEE Transactions on Evolutionary Computation, 2020, 24(2): 380-393.

[18] Julander C R. Basket analysis: A new way of analysing scanner data[J]. International Journal of Retail & Distribution Management, 1992, 20(7): 10-18.

[19] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197.

[20] Zitzler E, Thiele L. Multiobjective optimization using evolutionary algorithms a comparative case study[C]. Parallel Problem Solving from Nature — PPSN V. Amsterdam, 1998: 292−301.

## 作者简介

张磊(1986−), 男, 副教授, 博士生导师, 从事数据挖掘、社交网络分析及进化多目标优化等研究, E-mail: zl@ahu.edu.cn;

李柳(1997−), 女, 硕士生, 从事频繁高效用项集挖掘及多目标优化的研究, E-mail: li54548@163.com;

杨海鹏(1996−), 男, 博士生, 从事社团检测及多目标优化的研究, E-mail: haipengyang@qq.com;

孙翔(1997−), 男, 硕士生, 从事子集选择及多目标进化优化的研究, E-mail: sunx1997@163.com;

程凡(1979−), 男, 教授, 博士生导师, 从事智能优化、计算视觉及进化多目标优化等研究, E-mail: chengfan@mail.ustc.edu.cn;

孙晓燕(1978−), 女, 教授, 博士生导师, 从事智能优化和控制等研究, E-mail: xysun78@126.com;

苏喻(1984−), 男, 副教授, 博士, 从事机器学习、自然语言理解、数据挖掘与推荐系统等研究, E-mail: yusu@hfnu.edu.cn.