

控制与决策

Control and Decision

基于多标签k近邻方法实现元启发式算法的排名推荐

崔建双, 尚天泽, 杨帆, 蔚婧文

引用本文:

崔建双,尚天泽,杨帆,蔚婧文. 基于多标签k近邻方法实现元启发式算法的排名推荐[J]. *控制与决策*, 2022, 37(5): 1289–1298.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2020.1470>

您可能感兴趣的其他文章

Articles you may be interested in

[基于地标特征和元学习方法推荐最适用优化算法](#)

Recommending best suitable metaheuristic based on landmarking feature and meta-learning approach

控制与决策. 2021, 36(5): 1223–1231 <https://doi.org/10.13195/j.kzyjc.2019.0993>

[考虑卸载顺序约束的成品油二次配送车辆路径问题](#)

Vehicle routing problem of refined oil secondary distribution considering unloading sequence constraints

控制与决策. 2020, 35(12): 2999–3005 <https://doi.org/10.13195/j.kzyjc.2018.1756>

[超启发式交叉熵算法求解模糊分布式流水线绿色调度问题](#)

Hyper-heuristic cross-entropy algorithm for green distributed permutation flow-shop scheduling problem with fuzzy processing time

控制与决策. 2021, 36(6): 1387–1396 <https://doi.org/10.13195/j.kzyjc.2019.1681>

[带不相关并行机和有限缓冲MHFS调度的混合启发式算法](#)

Hybrid heuristic algorithm for multi-stage hybrid flow shop scheduling with unrelated parallel machines and finite buffers

控制与决策. 2021, 36(3): 565–576 <https://doi.org/10.13195/j.kzyjc.2019.0835>

[基于动态行为选择的和声搜索算法](#)

Harmony search algorithm based on dynamic behavior selection

控制与决策. 2021, 36(3): 577–588 <https://doi.org/10.13195/j.kzyjc.2019.0597>

基于多标签 k 近邻方法实现元启发式算法的排名推荐

崔建双^{1,2†}, 尚天泽, 杨帆, 蔚婧文

(北京科技大学 经济管理学院, 北京 100083)

摘要: 设计并研究一种基于多标签 k 近邻方法 (multi-label k -nearest neighbor, ML- k NN) 推荐元启发式算法的实现框架. 应用多标签 k 近邻分类学习技术, 实现最佳元启发式算法的排名推荐. 为了验证效果, 以多模式资源约束项目调度问题 (MRCPS) 为优化对象, 选取不同规模的数百个算例分别提取问题基本特征和地标特征; 选用遗传、粒子群、禁忌搜索、蜂群和蚁群 5 种元启发式算法, 使用 ML- k NN 建立元推荐模型; 利用海明损失、单错误率、覆盖率、排位损失和平均准确率 5 个指标对推荐效果做出分析和评价. 实验结果表明, 基于 ML- k NN 方法推荐元启发式算法效果突出, 其中基于地标特征的单错误率指标为 18.4%, 平均准确率达到 88.9%. 相对于 k NN 方法, ML- k NN 取得了更好的推荐结果. 此外, ML- k NN 方法可以实现对所有备选算法的排名推荐, 该研究结论有望推广应用于其他组合优化问题的优化算法推荐.

关键词: 多标签 k 近邻; 元启发式算法; 元学习; 问题特征; 地标特征; 算法推荐

中图分类号: TP273

文献标志码: A

DOI: 10.13195/j.kzyjc.2020.1470

引用格式: 崔建双, 尚天泽, 杨帆, 等. 基于多标签 k 近邻方法实现元启发式算法的排名推荐 [J]. 控制与决策, 2022, 37(5): 1289-1298.

Ranking recommendation to implement meta-heuristic algorithm based on multi-label k -nearest neighbor method

CUI Jian-shuang^{1,2†}, SHANG Tian-ze, YANG Fan, YU Jing-wen

(School of Economics and Management, University of Science and Technology Beijing, Beijing 100083, China)

Abstract: This paper designs and studies the implementation framework of a recommendation meta-heuristic algorithm based on the multi-label k -nearest neighbor (ML- k NN). The multi-label k -nearest neighbor classification learning technology is applied to implement the best meta-heuristic algorithm ranking recommendation. In order to verify the effect, the multi-modal resource-constrained project scheduling problem (MRCPS) is taken as the optimization object, and hundreds of examples of different scales are selected to extract landmarking features and problem basic features respectively; five meta-heuristic algorithms (genetics, particle swarm, tabu search, bee colony and ant colonies) are selected; the ML- k NN is applied to establish a meta-recommendation model; and the Hamming loss, single error rate, coverage rate, ranking loss and average accuracy rate are used to analyze and evaluate the recommendation effect. The experimental results show that the meta-heuristic algorithm based on ML- k NN recommendation is effective, among which, the single error rate of the ML- k NN based on landmarking features is 18.4%, and the average precision is 88.9%. The ML- k NN had been acquired the better recommendation effect in relative with the single label k NN. In addition, the ML- k NN method is able to achieve the ranking recommendations for all alternative algorithms. The research conclusions are expected to be extended to other combinatorial optimization algorithms.

Keywords: multi-label k -nearest neighbor method; meta-heuristic algorithm; meta-learning; problem feature; landmarking feature; algorithm recommendation

0 引言

现实中许多组合优化问题具有规模大、变量多、NP 难的特点, 为此, 人们多采用元启发式算法进行求解^[1]. 这类算法的随机机制虽不能确保获得最优解,

但较高的搜索效率和广泛的适用性奠定了其推广应用的基础. 多年以来, 先后涌现出了遗传、模拟退火、禁忌搜索、蚁群、粒子群、蜂群、细菌觅食、人工鱼群等多种元启发式算法, 与此同时也衍生出大量的改

收稿日期: 2020-10-26; 录用日期: 2021-02-10.

基金项目: 国家自然科学基金项目 (71871017).

责任编委: 巩敦卫.

†通讯作者. E-mail: cuijs@manage.ustb.edu.cn.

进算法. 目前来看, 由于相关算法理论上没有大的突破, 人们更多地侧重于应用方面的研究, 通过算法改进或定制来解决现实优化问题. 鉴于用户对各种算法的适用性和应用效果缺乏足够的认知和经验, 其成果大多具有主观随意性和盲目性, 难免“取短舍长”, 造成定制成本居高不下、效果差强人意等缺陷.

为了解决上述缺陷, 相关专家进一步提出了各种解决方案, 这些方案以追求更高效、更精确、智能化地解决优化问题为宗旨, 形成了目前在优化算法领域的几大热门研究方向: 1) 自适应(self-adaptive)元启发式: 致力于把元启发式算法赋予智能机制以提升自动解决问题的能力, 如自适应调整参数、自适应调整组件功能、自适应改变搜索策略等^[2-4]; 2) 构建通用的元启发式算法软件框架: 目的是促进软件重用、减少开发工作, 这样的框架能够评估和比较不同的算法, 转换即用型算法, 乃至设计合成新的算法^[5]; 3) 解决大规模优化问题: 事实上, 当问题规模增加时, 大多数元启发式算法的性能会迅速下降, 因此, 高维优化问题的可伸缩性成为复杂优化算法方法及现有技术对并行体系结构的适应性的基本要求^[6-7]; 4) 混合(hybrid)元启发式方法: 通过启发式技术的杂交^[8], 改进算法解决问题的效率, 例如, 集合策略(ensemble)^[9]、算法合成(portfolios)^[10]等; 5) 超启发式(hyper-heuristics)技术^[11-12]: 在上层通过选择性地使用低层不同的启发式方法来实现与问题无关的搜索, 使得优化过程通用化、便利化; 6) 算法推荐方法和技术: 从众多可用算法中推荐最适用于当前问题的算法^[13-18], 最适用是指从计算效率到计算结果都表现出最佳的效果, 其中基于元学习的推荐方法成为当前研究热点之一.

基于元学习方法推荐算法源自机器学习研究领域, 其所推荐的算法也是指机器学习算法. 但在优化算法研究领域, 利用元学习方法推荐优化算法是否可行呢? 令人鼓舞的是, 基于元学习方法实现优化算法推荐已经取得了初步成果. Smith-Miles^[16]提供了一个将算法选择问题视为学习问题的统一框架, 以解决算法推荐这一带有普遍需求的问题. 崔建双等^[13-15]利用神经网络等元学习算法成功地实现了最佳元启发式算法的推荐. Kanda等^[17]使用基于标签排名算法的元学习方法来推荐求解旅行商问题的算法. Messelis等^[18]基于实证难度模型将问题特征映射到算法的性能来推荐最佳优化算法. Gutierrez-Rodríguez等^[19]通过定义一组表征不同问题算例的元特征, 使用多层感知分类器并结合元特征选择方

法, 预测出最适合于给定问题算例的元启发式算法. Chu等^[20]提出了使用元学习的自适应推荐模型推荐基于问题特征的群智能算法, 通过将问题特征映射到算法性能来自适应地选择算法.

综上, 大多数研究以推荐单一最佳算法为目标, 而现实中在相同资源条件下, 对于同一个优化问题, 多种优化算法都能够获得相同或相近的结果, 从效率和精度上区别不大. 此外, 对于有大量算例需要优化的情况, 如果能根据每个算例的具体特征对算法做出排名推荐, 则将有助于用户根据实际状况做出选择. 本文将在前期研究成果的基础上^[13-15], 继续深入探讨使用元学习方法和推荐元启发式算法的问题, 提出并验证一种基于ML- k NN方法对多个元启发式算法进行排名推荐的实现框架.

有关ML- k NN应用的研究文献并不多见, 相关研究大多用于大规模数据集的处理, 尚未见到将ML- k NN用于优化算法推荐的公开文献. Jiang等^[21]提出了一种采用模糊相似度和 k NN实现多标签文本分类的模型. Gonzalez-Lopez等^[22]使用基于MapReduce编程模型的分布式ML- k NN方法, 使得数据集的最大处理速度提高百倍. Zhang等^[23]提出了一种基于传统的 k NN的多标签惰性学习方法. Kanj等^[24]提出了使用不同的多标签分类方法来适应已编辑的数据集. 李志欣等^[25]对多标签学习的研究现状和进展进行了综述.

在本文ML- k NN方法建模中, 训练集由与标签集合 L 相关联的算例特征组成. 通过学习已知标签集的训练算例来预测未见算例的标签集合 L . 算例特征分别提取问题特征和地标特征两种特征集; 标签集合 L 的取得需要运行各优化算法于各个算例, 详见3.1节. 对于每个未见算例, 首先确定其在训练集中 k 个最近邻, 然后从这些近邻算例的标签集获得统计信息, 利用贝叶斯方法预测其标签集并做出排名推荐.

为了取得实证效果, 本文以MRCPSP为问题算例集, 从目前流行的十几种元启发式算法中随机选取遗传(GA)、粒子群(PSO)、禁忌搜索(TB)、蜂群(ABC)和蚁群(ACS)5种元启发算法参与实验. 本文研究的是算法推荐问题, 重点关注的是推荐效果, 而非某种算法的绝对性能表现. 关于算法参数的选择对算法性能的影响, 本文处理的手段是每种算法经实验确定适应问题的典型参数, 之后在所有算例中运行的算法参数都不变动. 算法参数的变动可视为一个新算法. 另外, 本文提出的算法推荐框架预留了扩充优化算法的空间. 在确定算例集和元启发式算法之后, 通

过提取两类特征和多标签集合建立训练集,使用ML- k NN训练建模并用于对新算例做出推荐. 推荐效果采用海明损失、单错误率、覆盖率、排位损失、平均准确率5项指标做出评价. 实验结果显示,各项评价指标均能取得令人满意的结果.

1 ML- k NN算法

使用ML- k NN解决多标签分类问题有两种途径^[25]:一是问题转化法,将问题分解成一个或多个 k NN问题,继而构造多个子分类器,最终集成子分类器,得到一个完整的多标签分类器;二是扩展改进法,通过改进传统的分类方法用于多标签分类问题. 本文所用方法属于后者.

1.1 ML- k NN实现机制

ML- k NN是基于概率的有监督分类方法,其大体思路是:训练样本集中的每个样本的多标签是已知的,通过子样本集多折交叉训练来获得分类器模型. 对于每个未见样本,统计其在训练集中的 k 个最近邻的标签集分类信息,通过贝叶斯方法,使用训练集的先验概率向量和后验概率矩阵中的数据得到预测概率大小,并按照概率大小排序给出分类标签值,进而实现未见样本的类型排序.

ML- k NN具体实现过程如下:设给定样本 x 及其类别标签向量 $\mathbf{y}_x \in L$,其中 L 是所有标签的集合. 如果标签集合 L 中存在标签 l ,则意味着某个算例中确定 l 算法为最佳算法,即如果标签 $l \in L$,则其第 l 个分量 $\mathbf{y}_x(l)$ 赋值为1,否则为0. 令 $N(x)$ 表示训练集中样本 x 的多标签 k 近邻的集合,由此,可计算属于第 l 类别的计数向量为

$$\mathbf{C}_x(l) = \sum_{a=N(x)} \mathbf{y}_a(l), l \in L. \quad (1)$$

对于一个测试样本 t ,首先在训练集中标识其 k 个近邻 $N(t)$. H_1^t 代表标签 l 为样本 t 的最佳算法之一,即 H_1^t 代表样本 t 属于标签 l . H_0^t 代表标签 l 不是样本 t 的最佳算法,即 H_0^t 代表样本 t 不属于标签 l , $E_j^t(j \in 0, 1, \dots, k)$ 表示在 t 的 k 个近邻中恰好有 j 个样本带有标签 l . 因此,基于 \mathbf{C}_t ,由下式可以确定类别向量 \mathbf{y}_t :

$$\mathbf{y}_t(l) = \arg \max_{b \in (0,1)} P(H_b^l | E_{\mathbf{C}_t(l)}^l), l \in y. \quad (2)$$

使用贝叶斯方法,式(2)可展开为

$$\mathbf{y}_t(l) = \arg \max_{b \in (0,1)} \frac{P(H_b^l)P(E_{\mathbf{C}_t(l)}^l | H_b^l)}{P(E_{\mathbf{C}_t(l)}^l)} = \arg \max_{b \in (0,1)} P(H_b^l)P(E_{\mathbf{C}_t(l)}^l | H_b^l). \quad (3)$$

为了确定类别向量 \mathbf{y}_t ,所需要的信息是先验概率 $P(H_b^l)(j \in y, b \in (0,1))$ 和后验概率 $P(E_j^l | H_b^l)$

($j \in 0, 1, \dots, k$). 实际上,这些先验概率和后验概率都可以基于频率计数直接从训练集中计算获得,即前面所说的先验概率向量和后验概率矩阵.

上述过程的伪代码实现可由算法1描述.

算法1 ML- k NN算法.

$[\mathbf{y}_t, \mathbf{r}_t] = \text{ML-}k\text{NN}(T, k, L_T, s)$.

% 计算先验概率 $P(H_b^l)$

1) for $l \in y$ do

$$2) P(H_1^l) = \left(s + \sum_{i=1}^m \mathbf{y}_{m_i}(l) \right) / (s \times 2 + m),$$

$$P(H_0^l) = 1 - P(H_1^l).$$

3) end.

% 计算后验概率 $P(E_j^l | H_b^l)$

4) for $i \in (0, 1, \dots, t)$ do

5) $P(E_j^l | H_1^l) =$

$$(s + c[j]) / \left(s \times (k + 1) + \sum_{p=0}^k c[p] \right),$$

6) $P(E_j^l | H_0^l) =$

$$(s + c'[j]) / \left(s \times (k + 1) + \sum_{p=0}^k c'[p] \right);$$

7) end

% 计算 \mathbf{y}_t 和 \mathbf{r}_t

8) for $l \in y$ do

$$9) \mathbf{C}_x(l) = \sum_{a=N(x)} \mathbf{y}_a(l),$$

$$10) \mathbf{y}_t(l) = \arg \max_{b \in (0,1)} P(H_b^l)P(E_{\mathbf{C}_t(l)}^l | H_b^l),$$

$$11) \mathbf{r}_t(l) = P(H_b^l | E_{\mathbf{C}_t(l)}^l) =$$

$$\frac{P(H_1^l)P(E_{\mathbf{C}_t(l)}^l | H_1^l) / P(E_{\mathbf{C}_t(l)}^l)}{\sum_{b \in (0,1)} P(H_b^l)P(E_{\mathbf{C}_t(l)}^l | H_b^l)}.$$

12) end

算法1中:输入参数 T 是训练集, k 是近邻个数, L_T 是 T 对应的标签集, s 是控制强度平滑系数, m 是训练集个数, $\mathbf{y}_{m_i}(l)$ 是第 i 个训练集的标签个数. 输出自变量 \mathbf{y}_t 是类别向量, \mathbf{r}_t 是标签排名的实值向量(预测概率), $\mathbf{r}_t(l)$ 对应于样本 t 的后验概率 $P(H_1^l | E_{\mathbf{C}_t(l)}^l)$. 算法1的1)~3)行计算训练集的先验概率 $P(H_b^l)$; 4)~7)行计算训练集的后验概率 $P(E_j^l | H_b^l)$,在 l 的每次迭代中,使用 $c[j]$ 计数带有标签 l 的训练样本的数量,即其中 k 个近邻属于标签 l 的 j 个样本,相应地, $c'[j]$ 计数不属于标签 l 的训练样本的数量; 8)~12)行使用贝叶斯方法根据前面得到的先验概率和后验概率计算类别向量和实值向量.

使用算法1实现求解MPCSP问题的元启发式

算法排名推荐,具体步骤如下:

1) 设 $\text{Train}_t = (a_1, a_2, \dots, a_t)$ 为训练算例集,其中 $a_j (j \in 1, 2, \dots, t)$ 为算例的一组属性特征,各属性特征是条件独立的.

2) 基于训练算例获得分类器模型:

① 由属性特征矩阵计算训练算例样本两两之间的距离,此处采用欧几里得距离;

② 计算各标签类型的先验概率 $P(H_1^l)$ 和 $P(H_0^l)$;

③ 找出每一个算例的 k 个最近邻;

④ 逐一将每一个训练算例与其 k 个最近邻算例类型匹配/不匹配状况做数量统计,计算得到规模为 $L_1 \times (k+1)$ 的矩阵模型,其中行代表类型,列代表近邻次序,而第 l 行 ($l = 1, 2, \dots, L$) 第 1 列是不属于 l 类算例的统计结果;

⑤ 计算各标签类型的条件概率 $P(E_j^l|H_1^l)$ 和 $P(E_j^l|H_0^l)$ (算法1的第5)、6)行).

3) 基于分类器模型,采用贝叶斯方法,逐一计算测试算例的预测概率:

① 由测试属性特征矩阵计算测试算例和训练算例之间的距离,本文采用欧几里得距离;

② 从训练算例中找出每一个测试算例的 k 个训练算例最近邻;

③ 逐一将每一个测试算例与其 k 个近邻算例的类型状况做数量统计,得到关于测试算例的 k 个近邻的计数向量 $C_t(l)$ (算法1的第9)行);

④ 根据 $C_t(l)$ 中的数据选择对应的先验概率和后验概率,采用贝叶斯方法计算各测试算例的类别向量 y_t (算法1的第10)行)和标签排名的实值向量 r_t (算法1的第11)行).

1.2 ML- k NN分类结果评价指标

相对于单标签评价,多标签分类评价指标计算较为复杂^[20]. 假设有 m 个测试样本, Y_i 代表样本 i 实际标签集, Z_i 代表样本 i 预测标签集, M 代表所有标签总个数. 各项评价指标均是通过计算测试集中所有样本的实际标签集与预测标签集之间的平均误差来表示. 除了平均准确率越大越好之外,其余都是越小越好.

1) 海明损失 (Hamming loss, HL): 用于评价一个样本误分类的情况. 即该样本不属于(或属于)某类标签但被错分为另一类标签,从而导致误判. 该指标度量了预测结果标签序列与实际标签序列之间数值上的距离. 即

$$\text{HL} = \frac{1}{m} \sum_{i=1}^m \frac{Y_i \Delta Z_i}{M}, \quad (4)$$

其中 Δ 表示计算两个集合 Y_i 与 Z_i 的对称差,即按位置做异或运算.

2) 1-错误率 (one-error, OE): 用来评估在输出结果中排第一的标签并不属于实际标签集中的概率,类似于单标签分类问题中的评价指标错误率. 有

$$\text{OE} = \frac{1}{m} \sum_{i=1}^m m(\arg \max(Z_i) \in \bar{Y}_i);$$

$$m(x) = \begin{cases} 1, & x \text{ is true;} \\ 0, & x \text{ is false.} \end{cases} \quad (5)$$

3) 覆盖率 (coverage, CO): 用于考察在样本的标签排序序列中覆盖隶属于样本的所有标签所需的搜索深度情况. 有

$$\text{CO} = \frac{1}{m} \sum_{i=1}^m \max(\text{rank}(Z_i)) - 1. \quad (6)$$

4) 排位损失 (ranking loss, RL): 用于评价预测样本不相关标签的排位高于相关标签排位次数的平均值. 有

$$\text{RL} = \frac{1}{m} \sum_{i=1}^m \frac{1}{|Y_i| |\bar{Y}_i|} \times \left(\sum_a Z_i(l_a) < Z_i(l_b) | (l_a, l_b) \in Y_i \times \bar{Y}_i \right), \quad (7)$$

其中 \bar{Y}_i 是 Y_i 相对于所有类别标签集 L 的补集.

5) 平均准确率 (average precision, AP): 用于评估在样本的概念标签排序序列中,排在某个特定标签 $l \in Y_i$ 之前的标签占标签集合的平均比例. 有

$$\text{AP} = \frac{1}{m} \sum_{i=1}^m \frac{1}{|Y_i|} \sum_k \frac{l_k |Z_i(l_k) \leq Z_i(l), (l_k, l) \in Y_i}{Z_i(l)}. \quad (8)$$

2 基于ML- k NN的算法推荐框架

关于算法推荐, Rice^[26] 最早曾给出一个概念图和形式化的定义: 设 P 是问题的集合, A 是可用算法的集合, F 是问题可测特征的集合, Y 是算法性能测度的集合. 对于给定问题实例 $x \in P$, 算法推荐就是在已知问题特征 $f(x) \in F$ 情况下, 找到算法 $a \in A$, 使得 $S(f(x)) \rightarrow A : \max\{y(a(x))\} \in Y$.

参考 Rice 算法推荐的概念, 本文设计了基于 ML- k NN 排名推荐元启发式算法的实现框架, 如图 1 所示. 具体实现步骤如下:

1) 对问题算例集所有的算例以最小化目标值偏差为目标分别执行元启发式算法 a_1, a_2, \dots, a_n . 为了无偏起见, 每个算法执行所需资源一致, 每个算例独立执行 5 次后取均值.

2) 提取算例集的问题特征和地标特征.

3) 根据步骤 1) 中各算法执行的结果, 按照目标

值偏差从小到大进行排序,其中偏差最小(可能不止1个)的为1,其他为0,得到多标签集合.例如10010意味着算法1与算法4效果相同且为最好算法.

4) 根据特征及多标签集合,使用ML- k NN方法训练建模得到分类器.

5) 使用分类器做出最佳算法推荐和排名推荐,并依据评价指标做出评价.

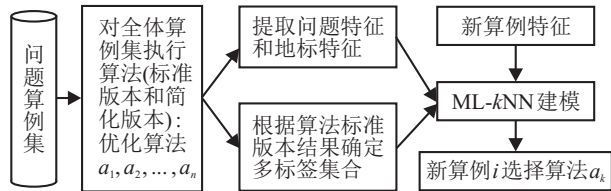


图1 基于ML- k NN排名推荐元启发式算法的实现框架

3 实验过程及结果分析

3.1 实验环境的设置

实验采用Matlab(2014b)编程实现.选取不同规模的350个MRCPS P算例作为实验数据集(活动数量分别为10、12、14、16、18、20、30).选用GA、PSO、ACS、ABC和TB共5种元启发式算法作为被推荐算法.为了确保各算法具有公平可比性,采用如下几项措施:1)使用同一台计算机(CPU Core i5,8G,64位);2)各优化算法在每个算例上均运行5次并对结果取均值;3)各优化算法均采用相同的编/解码格式;4)训练建模时均随机选取相同数量的训练和测试算例,并采用10折交叉验证;5)各优化算法均取6000次解为结束标准.

各优化算法的典型参数如下:1)GA:种群规模24,迭代次数250,交叉概率 $P_C = 0.85$,变异概率 $P_M = 0.01$,下一代种群选择采用轮盘赌方式;2)PSO:粒子个数30,迭代次数200, $c_1 = 0.8$, $c_2 = 0.7$, $\omega = 0.93 - 0.3 \times$ 迭代次数/最大迭代次数,取标准粒子群算法公式;3)TB:解的上限为6000,禁忌表长4,最佳邻域解保留个数6;4)ABC:总蜜蜂数20,迭代次数300,蜜蜂角色变换上限6,下一代蜂群选择采用轮盘赌方式;5)ACS: $\rho = 0.2$, $\alpha = 0.1$, $\beta = 1$, $Q = 40$,蚂蚁数量30,迭代次数200,信息素更新采用蚁周模型.

在元启发式算法中,其种群数量不易过大或过小:当种群数量过大时,单次迭代种群之间学习所耗时间过大;当种群数量过小时,不能体现群智能思想,无法进行种群之间的学习.故本文选择GA种群规模为24,PSO和ACS种群规模为30,ABC种群规模为20,在此基础上,为满足所有算法6000解这一要求,对4种群智能算法的迭代次数进行相应调整.TB

作为随机搜索算法,选择其解上限为6000次.而对于其他参数指标,本文根据各参数代表的含义进行相应调整和实验,选取其中效果最好时所对应的参数值.

标签集合 L 由所有算法组成,标签 l 指标签集合 L 中的某一个元素位置(即某一个算法),若算法 $q \in L$,则该算例所对应算法的标签为1.一个算例可能会有多个标签为1的情况,即一个算例同时属于多个类,意味着该算例分别使用多个算法求解效果上没有区别.训练集是从不同规模的350个MRCPS P算例数据集中随机选择一部分,其余作为测试集,其测试/训练占比的选择依据将在参数分析部分阐述.在标签集合中,为了更加准确地体现平均性能,避免算法偏差值的波动带来推荐的不准确,本文在每个算例上将各算法运行5次之后取平均值作为多标签集合的依据.虽然元启发式算法运行所得结果有所波动,但是各算法运行所得偏差值在一定范围内比较稳定,只需几次运行取平均值就可以体现算法处理问题的能力;而且随着算例数量的增加,其处理时间会随之递增.实验发现,在同一算例运行10次甚至更多次时,同一算法取平均值效果较小、成本较大,于实验帮助不大,因此本文选择5次运行取平均值.

3.2 特征矩阵的获取

3.2.1 问题特征

MRCPS P是一类典型的NP难组合优化问题,文献[27]给出了该问题的模型定义.本文重点关注其特征的提取.问题特征能够以“离散”的形式表征其轮廓,并在问题数据集样本特征上反映出算法的性能^[28].

文献[12,28]提及4种类型的问题特征,本文使用其中的两类特征.第1类是源自问题的基本特征及统计特征,如变量维度、节点个数、标准差、偏度、峰度、距离、相关系数、平均熵等;第2类是地标特征或称基准特征^[14,29].本文结合第1类特征分别提取26个有关MRCPS P的基本特征和4个统计特征,如表1所示.实验结果表明,统计特征并没有带来结果的改善,初步判断是由于算例之间的相互独立性使得统计特征难以提供有价值的特征信息.计算表明,算例之间的相关系数普遍小于0.05,故本文仅取其中的26个基本特征.

3.2.2 地标特征

关于MRCPS P的地标特征,文献[14]给出了较为详细的描述,本文再次采纳这项技术用于最佳元启发式算法的排名推荐.地标特征是与算法密切相关的,这也是其与问题基本特征和统计特征的区别.

表1 MRCPSP问题特征列表

序号	特征	变量	特征描述
1	Anum	NumActivity	活动数量
2	SumRR	SumRR	可更新资源总量
3	SumNRR	SumNRR	不可更新资源总量
4	UBRRuse ₁	UBRR ₁	可更新资源1上限
5	UBRRuse ₂	UBRR ₂	可更新资源2上限
6	UBNRuse ₁	UBNR ₁	不可更新资源1上限
7	UBNRuse ₂	UBNR ₂	不可更新资源2上限
8	AvgRRuse	SumRR/Anum	可更新资源平均可用量
9	AvgNRuse	SumNRR/Anum	不可更新资源平均可用量
10	Suc ₁	Successor ₁	后继数量为1的活动数
11	Suc ₂	Successor ₂	后继数量为2的活动数
12	Suc ₃	Successor ₃	后继数量为3的活动数
13	TotalDur	SumallactsDur/Mnum	全体活动持续时间
14	AvDur	TotalDur/Anum	活动持续时间均值
15	AvgRRuse ₁	SumRRuse ₁ /(Anum×Mnum)	可更新资源1平均使用量
16	AvgRRuse ₂	SumRRuse ₂ /(Anum×Mnum)	可更新资源2平均使用量
17	AvgNRuse ₁	SumNRuse ₁ /(Anum×Mnum)	不可更新资源1平均使用量
18	AvgNRuse ₂	SumNRuse ₂ /(Anum×Mnum)	不可更新资源2平均使用量
19	RS ₁	$RS_1 = (R_1 - R_1^{min}) / (R_1^{max} - R_1^{min})$	可更新资源强度1,反映第1种可更新资源的稀缺状况
20	RS ₂	$RS_2 = (R_2 - R_2^{min}) / (R_2^{max} - R_2^{min})$	可更新资源强度2,反映第2种可更新资源的稀缺状况
21	NRS ₁	$NRS_1 = (NR_1 - NR_1^{min}) / (NR_1^{max} - NR_1^{min})$	不可更新资源强度1,反映第1种不可更新资源的稀缺状况
22	NRS ₂	$NRS_2 = (NR_2 - NR_2^{min}) / (NR_2^{max} - NR_2^{min})$	不可更新资源强度2,反映第2种不可更新资源的稀缺状况
23	RC ₁	$RC_1 = \bar{R}_1 / R_1$	可更新资源受限程度1,反映资源1稀缺程度
24	RC ₂	$RC_2 = \bar{R}_2 / R_2$	可更新资源受限程度2,反映资源2稀缺程度
25	NRC ₁	$NRC_1 = \overline{NR}_1 / NR_1$	不可更新资源受限程度1,反映资源1的稀缺程度
26	NRC ₂	$NRC_2 = \overline{NR}_2 / NR_2$	不可更新资源受限程度2,反映资源2的稀缺程度
27	SD	STDEVP	基本特征的标准差
28	Sk	SKEW	基本特征的偏度
29	Ku	KURT	基本特征的峰度
30	DD	PDIST	各算例与中心算例的欧氏距离

实验以算法执行一定时长的目标值偏差作为地标特征. 时间太长损失效率, 太短则难以获得足够的信息, 设计时应根据实际运行结果进行判断. 原则上是所有算法均呈现稳定状态之后的最小执行时间. 为此, 本文分别设计了1s、5s、10s、20s、30s

等5种运行时长作为地标特征进行分析比较. 首先分别编程实现5种优化算法, 然后运行这5种算法求解350个算例的各算法偏差值. 分析发现: 当运行时长小于10s时, 大部分算例的偏差值下降且呈现指数趋势; 当运行时长为10s时, 大部分算例的偏差值下降

表2 5种元启发式算法获取的地标特征(仅取10个样例)

序号	执行时长10s					执行时长30s				
	GA	PSO	TB	ABC	ACS	GA	PSO	TB	ABC	ACS
1	3.4	0.2	0	2.2	0	2.2	0	0	0.8	0
2	3.2	0.4	0	2.4	0	2	0	0	0.8	0
3	2.4	1.2	0	1.6	0	2.2	0.4	0	0.2	0
4	3.4	0.4	0.4	2	0.2	2.8	0	0	0.6	0
5	5.6	0	0	3.8	0	1.6	0	0	0.4	0
6	4.2	1.2	1	5.8	0.2	3.8	1.2	0	1.8	0
7	7.4	2.4	0.6	4.8	3	5.8	0.4	0.4	3.2	0.8
8	3.6	0.8	0.2	4.4	1.6	2.4	0.2	0	1.2	0.6
9	2.4	0	0	1	0.2	1.4	0.2	0	0.4	0
10	6.2	2.4	2	4.4	2.6	4.6	1.8	0.8	1.6	1.2

稳定;当运行时长为30s时,大部分算例的偏差值趋于稳定.因此提取地标特征时,每个算例分配的最小时长设定为10s,确切稳定时间设定为30s.

在实验环境的设置中,为得到稳定的算法偏差值,各算法在每个算例上运行5次取平均值,因为地标特征运行的是算法的简化版本,所以确定地标特征为各算法在每个算例上运行5次取平均值.表2列出了部分样本算例的地标特征数据,这些数值是各算法在每个算例上运行5次后所取的偏差均值.

表3 参数($k/s/r$)组合变化下各评价指标最大值

评价指标	HL	OE	CO	RL	AP
评价值	0.145	0.02	0.613	0.044	0.972
$k/s/r$	10/0.02/0.11	12/0.38/0.11	10/0.02/0.11	12/0.38/0.11	12/0.38/0.11

由表3可见,参数($k/s/r$)的组合以(10/0.02/0.11)或(12/0.38/0.11)两组典型数值效果较好且参数相差较小.又因为45000组组合数据虽然均为10折交叉验证,但偶然性依然存在,故在后续实验中对已经选择的两组典型数值进行再次实验验证,最终选择参数组合典型值 $k = 12, s = 0.38, r = 0.11$.后续实验发现:固定参数 s 和 r 时, k 选择10~20时评价指标最好,当 k 太小或太大时评价指标较差;固定参数 k 和 r 时, s 选择0.01~0.5时评价指标最好,超过0.5时 s 越大评价指标越差;固定参数 k 和 s 时, r 选择0.1~0.4时评价指标最好,超过0.5时 r 越大评价指标越差.而且虽然其中单个指标取得了较好成效,但可能造成其他评价指标更坏,故参数选择时一般需要指明以哪个评价指标为重点,当需要综合考虑各评价指标时,需要专门进行参数选择方法实验.

3.4 实验结果及分析

3.4.1 基于问题基本特征的实验结果及分析

按照1.1节所描述的实现步骤,将训练集的问题特征矩阵和多标签集代入ML- k NN方法进行训练建模,然后使用测试数据验证模型效果.所有参与评估的多标签评价指标均采用相同的数据集划分和10折交叉验证.表4展示了在问题基本特征下各评价指标的结果.

表4 问题特征下各评价指标数值

评价指标	问题特征
HL	0.2355
OE	0.1915
CO	2.0300
RL	0.1099
AP	0.8807

3.3 参数分析

在ML- k NN建模环节,近邻值 k 、平滑系数 s 和训练/测试算例占比 r 这三个参数有可能会对预测结果产生影响.为此,对3个参数数值进行组合实验,测试对HL、OE、CO、RL和AP这5项评价指标的影响.选择 k 值从1~30,步长以1变化;平滑系数 s 从0.01~1,步长以0.02变化;测试/训练占比 r 从0.1~0.5,步长以0.015变化,共测试45000组组合数据.参数($k/s/r$)组合变化下各评价指标最大值如表3所示.

由表4可知,本文提出的ML- k NN算法在350数据集下其5个评价指标均取得了不错的效果,从而表明了算法推荐可以实现多标签分类并进行排名推荐.实验结果同时还显示:评价指标OE为0.1915,表明第1个标签预测准确率较高,可以很好实现单标签实验中的要求;RL为0.1099,表明在进行预测排名中排位损失很小,具有实际推荐意义.

在单标签实验中,一般采取错误率作为评价指标,而多标签中的OE为预测结果中最前端的标签不属于该算例实际标签集的算例个数占所有算例的比例,即多标签中的OE与单标签中的错误率概念相同.因此,本文将ML- k NN和 k NN两种实验下的错误率结果进行对比.在 k NN实验过程中采取与ML- k NN完全相同的问题特征,并且其标签来源与多标签集合均为各算法偏差均值,唯一不同之处是: k NN中的标签集合选择各算法偏差值最小的标签,如果偏差值相同则选择其中靠前的标签;ML- k NN的标签集合选择各算法偏差值最小的标签,如果偏差值相同则选择所有相同的标签作为标签集合.

实验结果发现,采用问题特征和ML- k NN方法的错误率为19.15%,采用问题特征和 k NN方法的错误率为37.52%.基于问题特征和ML- k NN方法进行推荐比基于问题特征和 k NN方法进行推荐的错误率小18个百分点,几乎减少了一倍.这些实验结果表明,基于ML- k NN方法实现了最佳算法推荐和多种算法的排名推荐,其结果是可接受的.

3.4.2 基于地标特征的实验结果及分析

与基于问题特征的实验步骤类似,基于地标特征实验按照1.1节所描述的实现步骤,将训练集的地标特征矩阵和多标签集代入ML- k NN方法中进行训

练建模,然后使用测试数据验证模型效果.所有参与评估的多标签评价指标均采用相同的数据集划分和10折交叉验证测试.表5展示了在两组地标特征下各评价指标的结果.

表5 不同地标特征下各评价指标数值

评价指标	执行时长10s	执行时长30s
HL	0.2365	0.1810
OE	0.2135	0.1843
CO	1.8925	1.8425
RL	0.1110	0.0861
AP	0.8635	0.8887

分析表5中的数据,执行时长为30s时的各项指标均优于时长为10s时的结果,这表明较长的地标特征的提取时间有利于获得更多的地标信息.此外,对比表4和表5的最后一列数据,使用地标特征(30s)做算法推荐的效果要好于使用问题基本特征做算法推荐的效果,究其原因很可能是由于问题的基本特征是与算法无关而独立提取的,而地标特征是与算法密切相关的.事实上,迄今为止,问题的基本特征与算法性能表现之间的关联关系仍是一个开放的课题.地标

特征能够更好地契合问题与算法之间的关联关系,直接反映出问题的内在特征.图2给出了3组不同特征下各项指标的归一化对比效果.由图2可见,地标特征(30s)的所有指标均优于其他两组特征的指标.

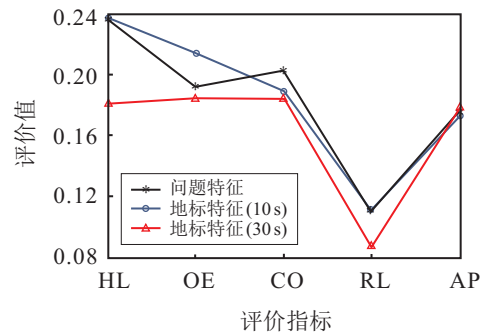


图2 3组不同特征下各项指标对比

3.5 算法排名推荐

上述实验结果表明,基于地标特征的推荐效果较好,故本节将以执行时长30s的地标特征作为特征集进一步展开算法排名实验.表6是根据训练集的地标特征矩阵和多标签集,采用ML-kNN方法进行实验得到的训练集先验概率和后验概率(为便于展示取近邻数 $k=1$).

表6 先验概率和后验概率

算法标签	先验概率		后验概率			
	$P(H_1^1)$	$P(H_0^1)$	$P(E_0^1 H_1^1)$	$P(E_1^1 H_1^1)$	$P(E_0^1 H_0^1)$	$P(E_1^1 H_0^1)$
GA	0.040	0.960	0.735	0.265	0.949	0.051
PSO	0.799	0.201	0.154	0.846	0.373	0.627
TB	0.793	0.207	0.160	0.840	0.376	0.624
ABC	0.407	0.593	0.508	0.492	0.646	0.354
ACS	0.449	0.551	0.189	0.811	0.811	0.189

在表6中: $P(H_1^1)$ 表示标签为1的个数占训练集算例所有标签的比例; $P(H_0^1)$ 表示标签为0的个数占训练集算例所有标签的比例; $P(E_0^1|H_1^1)$ 表示算例在标签为1的条件下,其最近邻算例标签为0的概率; $P(E_1^1|H_1^1)$ 表示算例在标签为1的条件下,其最近邻算例标签为1的概率; $P(E_0^1|H_0^1)$ 表示算例在标签为0的条件下,其最近邻算例标签为0的概率; $P(E_1^1|H_0^1)$ 表示算例在标签为0的条件下,其最近邻算例标签为

1的概率.

对测试算例提取地标特征组成特征向量后,基于训练集训练出先验概率和后验概率,通过ML-kNN方法根据特征向量从训练集选取 k 个最近邻;然后,根据最近邻的标签集合,采用贝叶斯方法得到类别向量;最后,基于类别向量进行预测排名,如表7所示,其计算方法见1.1节式(3).

表7 测试集预测算法排名(仅取3个样例)

算法	近邻标签			真实标签			实际排名			类别向量			预测排名		
	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
GA	0	0	0	0	0	0	5	5	5	0.031	0.031	0.031	5	5	5
PSO	0	1	1	0	1	1	1	1	2	0.623	0.843	0.843	1	1	1
TB	0	1	1	1	1	1	2	2	3	0.619	0.838	0.838	2	2	2
ABC	1	1	0	0	1	1	4	3	4	0.488	0.488	0.350	3	3	4
ACS	0	0	1	0	0	1	3	4	1	0.159	0.159	0.778	4	4	3

由表7可知,在3个样本中,实际排名是通过比较5种算法偏差值大小所得,其中偏差值越小算法越准确,当算法偏差值相同时,运行时间较少的算法排名在前.预测排名是根据类别向量的大小进行排名,类别向量越大算法越优秀.第1个样本实现了较准确的排名推荐,准确度达到0.6;第2个样本实现了准确的排名推荐,准确度达到1;第3个样本实现了较准确的排名推荐,准确度为0.4.实验表明:算法排名推荐可以体现各推荐算法的优劣和对算例的适应性,实现对算法推荐的合理准确推荐;同时也表明算法排名推荐是可行的,可以针对不同问题算例对各推荐算法的效果进行直观展示.

3.6 对未见算例的预测与分析

为了验证推荐效果,本文从算例数据集中随机选取一组35个未见算例(未在训练集和测试集使用的算例),以相同的资源成本运行这些算例于每一个元启发式算法并进行算法排名;然后,分别根据每一个算例的问题或地标特征,采用已经建立的分器对其进行算法推荐和排名推荐.实验中所有参与评估的多标签评价指标均采用相同的数据集划分和10折交叉验证.表8展示了在两组特征下各评价指标的结果数值.

表8 未参与训练算例在两组特征下各评价指标数值

评价指标	问题特征	地标特征(30s)
HL	0.257	0.177
OE	0.235	0.176
CO	1.743	1.600
RL	0.096	0.064
AP	0.864	0.907

由表8可知,各评价指标均符合预期,相对于基于问题特征的各项评价指标,基于地标特征(30s)所得评价指标得到了更好的效果,说明基于地标特征(30s)更能体现各算例的不同点,可以更准确地区分不同算例.例如,基于地标特征(30s)中的OE为0.176,RL为0.064,获得了更准确的元启发式算法排名推荐.

4 结论

在算法研究领域,算法选择与推荐是一项具有挑战性的前沿课题.其目标是从众多可用算法中选出最适用于当前问题的算法,实现问题-算法的精准对接,以提升解决问题的效率.而以往进行算法推荐采用的均是单标签方法.

相对于单标签元学习方法,ML- k NN方法有其独有特色:1)可以克服单标签方法的算法“歧视”和训练模型偏差,采用更切合实际的标签集合;2)单标签

推荐方法只能推荐一个最适用算法,而基于ML- k NN方法可以对算法进行排名推荐;3)对于某些易解算例,往往几种算法均可实现最优解,此时采用多标签方法允许用户根据实际情况做出灵活选择.

接下来将进一步研究更多的多标签元学习方法的表现形式和建模方法,目标是以更高效的方式获得更精确的推荐效果.

参考文献(References)

- [1] Boussaïd I, Lepagnot J, Siarry P. A survey on optimization metaheuristics[J]. *Information Sciences*, 2013, 237: 82-117.
- [2] Li K, Fialho Á, Kwong S, et al. Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition[J]. *IEEE Transactions on Evolutionary Computation*, 2014, 18(1): 114-130.
- [3] 郭一楠, 刘丹丹, 程健, 等. 自适应混合变异文化算法[J]. *电子学报*, 2011, 39(8): 1913-1918.
(Guo Y N, Liu D D, Cheng J, et al. Adaptive cultural algorithm adopting mixed mutation[J]. *Acta Electronica Sinica*, 2011, 39(8): 1913-1918.)
- [4] 郭一楠, 程健, 曹媛媛, 等. 基于混沌知识迁移的多种群粒子群文化算法[J]. *控制理论与应用*, 2011, 28(9): 1056-1062.
(Guo Y N, Cheng J, Cao Y Y, et al. Multi-population particle swarm cultural algorithms adopting chaotic knowledge migration[J]. *Control Theory & Applications*, 2011, 28(9): 1056-1062.)
- [5] Ochoa G, Hyde M, Curtois T, et al. Hy flex: A benchmark framework for cross-domain heuristic search[C]. *European Conference on Evolutionary Computation in Combinatorial Optimization*. Heidelberg: IEEE, 2012: 136-147.
- [6] 梁静, 刘睿, 于坤杰, 等. 求解大规模问题协同进化动态粒子群优化算法[J]. *软件学报*, 2018, 29(9): 2595-2605.
(Liang J, Liu R, Yu K J, et al. Dynamic multi-swarm particle swarm optimization with cooperative coevolution for large scale global optimization[J]. *Journal of Software*, 2018, 29(9): 2595-2605.)
- [7] Shen X N, Guo Y N, Li A M. Cooperative coevolution with an improved resource allocation for large-scale multi-objective software project scheduling[J]. *Applied Soft Computing*, 2020, 88: 106059.
- [8] Wang Z M, Luo Q F, Zhou Y Q. Hybrid metaheuristic algorithm using butterfly and flower pollination base on mutualism mechanism for global optimization problems[J]. *Engineering with Computers*, 2021, 37(4): 3665-3698.
- [9] Wu G H, Mallipeddi R, Suganthan P N. Ensemble strategies for population-based optimization algorithms — A survey[J]. *Swarm and Evolutionary Computation*, 2019, 44: 695-711.

- [10] Xu L, Hutter F, Hoos H H, et al. SATzilla: Portfolio-based algorithm selection for SAT[J]. *Journal of Artificial Intelligence Research*, 2008, 32: 565-606.
- [11] Burke E K, Gendreau M, Hyde M, et al. Hyper-heuristics: A survey of the state of the art[J]. *Journal of the Operational Research Society*, 2013, 64(12): 1695-1724.
- [12] 王万良, 徐昶, 赵燕伟, 等. 基于超启发式算法的选址-路径问题研究[J]. *浙江工业大学学报*, 2019, 47(6): 604-610.
(Wang W L, Xu C, Zhao Y W, et al. Research on location-routing problem based on hyper-heuristic algorithm[J]. *Journal of Zhejiang University of Technology*, 2019, 47(6): 604-610.)
- [13] 崔建双, 刘晓婵, 杨美华, 等. 基于元学习推荐的优化算法自动选择框架与实证分析[J]. *计算机应用*, 2017, 37(4): 1105-1110.
(Cui J S, Liu X C, Yang M H, et al. Meta-learning based optimization algorithm selection framework and its empirical study[J]. *Journal of Computer Applications*, 2017, 37(4): 1105-1110.)
- [14] 崔建双, 吕玥, 徐子涵. 基于地标特征和元学习方法推荐最适用优化算法[J]. *控制与决策*, 2021, 36(5): 1223-1231.
(Cui J S, Lyu Y, Xu Z H. Recommending best suitable metaheuristic based on landmarking feature and meta-learning approach[J]. *Control and Decision*, 2021, 36(5): 1223-1231.)
- [15] 崔建双, 车梦然. 基于多分类支持向量机的优化算法智能推荐系统与实证分析[J]. *计算机工程与科学*, 2019, 41(1): 153-160.
(Cui J S, Che M R. An intelligent recommendation system for optimization algorithms based on multi-classification support vector machine and its empirical analysis[J]. *Computer Engineering & Science*, 2019, 41(1): 153-160.)
- [16] Smith-Miles K A. Cross-disciplinary perspectives on meta-learning for algorithm selection[J]. *ACM Computing Surveys*, 2009, 41(1): 1-25.
- [17] Kanda J, de Carvalho A, Hruschka E, et al. Meta-learning to select the best meta-heuristic for the traveling salesman problem: A comparison of meta-features[J]. *Neurocomputing*, 2016, 205: 393-406.
- [18] Messelis T, De Causmaecker P. An automatic algorithm selection approach for the multi-mode resource-constrained project scheduling problem[J]. *European Journal of Operational Research*, 2014, 233(3): 511-528.
- [19] Gutierrez-Rodríguez A E, Conant-Pablos S E, Ortiz-Bayliss J C, et al. Selecting meta-heuristics for solving vehicle routing problems with time windows via meta-learning[J]. *Expert Systems with Applications*, 2019, 118: 470-481.
- [20] Chu X H, Cai F L, Cui C, et al. Adaptive recommendation model using meta-learning for population-based algorithms[J]. *Information Sciences*, 2019, 476: 192-210.
- [21] Jiang J Y, Tsai S C, Lee S J. FSKNN: Multi-label text categorization based on fuzzy similarity and k nearest neighbors[J]. *Expert Systems with Applications*, 2012, 39(3): 2813-2821.
- [22] Gonzalez-Lopez J, Ventura S, Cano A. Distributed nearest neighbor classification for large-scale multi-label data on spark[J]. *Future Generation Computer Systems*, 2018, 87: 66-82.
- [23] Zhang M L, Zhou Z H. ML-KNN: A lazy learning approach to multi-label learning[J]. *Pattern Recognition*, 2007, 40(7): 2038-2048.
- [24] Kanj S, Abdallah F, Denux T, et al. Editing training data for multi-label classification with the k -nearest neighbor rule[J]. *Pattern Analysis and Applications*, 2016, 19(1): 145-161.
- [25] 李志欣, 卓亚琦, 张灿龙, 等. 多标记学习研究综述[J]. *计算机应用研究*, 2014, 31(6): 1601-1605.
(Li Z X, Zhuo Y Q, Zhang C L, et al. Survey on multi-label learning[J]. *Application Research of Computers*, 2014, 31(6): 1601-1605.)
- [26] Rice J R. The algorithm selection problem[J]. *Advances in Computers*, 1976, 15: 65-118.
- [27] van Peteghem V, Vanhoucke M. An experimental investigation of metaheuristics for the multi-mode resource-constrained project scheduling problem on new dataset instances[J]. *European Journal of Operational Research*, 2014, 235(1): 62-72.
- [28] 曾子林, 张宏军, 张睿, 等. 基于元学习思想的算法选择问题综述[J]. *控制与决策*, 2014, 29(6): 961-968.
(Zeng Z L, Zhang H J, Zhang R, et al. Summary of algorithm selection problem based on meta-learning[J]. *Control and Decision*, 2014, 29(6): 961-968.)
- [29] Balte A, Pise N, Kulkarni P. Meta-learning with landmarking: A survey[J]. *International Journal of Computer Applications*, 2014, 105(8): 47-51.

作者简介

崔建双(1961—), 男, 副教授, 博士, 从事智能优化算法、项目优化调度、商务数据分析等研究, E-mail: cuijs@manage.ustb.edu.cn;

尚天泽(1993—), 男, 硕士生, 从事智能优化算法、项目优化调度、机器学习算法的研究, E-mail: shangtianze1@163.com;

杨帆(1996—), 女, 硕士生, 从事智能优化算法、项目优化调度、机器学习算法的研究, E-mail: 15822813229@163.com;

蔚婧文(1995—), 女, 硕士生, 从事智能优化算法、组合优化问题的研究, E-mail: 1055069323@qq.com.

(责任编辑: 李君玲)