

# 控制与决策

Control and Decision

## 考虑新技能学习机制的软件项目调度人工蜂群算法

申晓宁, 姚铨滨, 徐继勇, 宋丽妍, 王玉芳

引用本文:

申晓宁, 姚铨滨, 徐继勇, 宋丽妍, 王玉芳. 考虑新技能学习机制的软件项目调度人工蜂群算法[J]. *控制与决策*, 2023, 38(3): 790–796.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2021.1319>

---

## 您可能感兴趣的其他文章

### Articles you may be interested in

#### [混合离散人工蜂群算法求解含不相关并行机的分布式柔性流水线调度](#)

Hybrid discrete artificial bee colony algorithm for distributed flexible flowline scheduling with unrelated parallel machines  
*控制与决策*. 2023, 38(3): 779–789 <https://doi.org/10.13195/j.kzyjc.2021.1438>

#### [基于刺激-响应分工机制的人工蜂群算法](#)

Artificial bee colony algorithm based on stimulus-response labor division  
*控制与决策*. 2022, 37(4): 881–891 <https://doi.org/10.13195/j.kzyjc.2020.1346>

#### [基于IABC算法的舰载机着舰调度](#)

Scheduling of landing for carrier-based aircraft based on improved artificial bee colony algorithm  
*控制与决策*. 2022, 37(7): 1810–1818 <https://doi.org/10.13195/j.kzyjc.2020.1767>

#### [面向建材装备集团制造的分布式多项目资源调度](#)

Distributed multi-project resource scheduling oriented to manufacturing of building materials equipment group  
*控制与决策*. 2021, 36(9): 2133–2142 <https://doi.org/10.13195/j.kzyjc.2019.1802>

#### [基于16方向24邻域改进蚁群算法的移动机器人路径规划](#)

Mobile robots path planning based on 16-directions 24-neighborhoods improved ant colony algorithm  
*控制与决策*. 2021, 36(5): 1137–1146 <https://doi.org/10.13195/j.kzyjc.2019.0600>

# 考虑新技能学习机制的软件项目调度人工蜂群算法

申晓宁<sup>1,2,3†</sup>, 姚铖滨<sup>1</sup>, 徐继勇<sup>1</sup>, 宋丽妍<sup>4</sup>, 王玉芳<sup>1,2,3</sup>

- (1. 南京信息工程大学 自动化学院, 南京 210044;
2. 南京信息工程大学 江苏省大气环境与装备技术协同创新中心, 南京 210044;
3. 南京信息工程大学 江苏省大数据分析技术重点实验室, 南京 210044;
4. 南方科技大学 广东省类脑智能计算重点实验室, 广东 深圳 518055)

**摘要:** 考虑新技能的学习机制, 建立软件项目调度问题的数学模型. 该模型融入员工对新技能的学习、新技能熟练度的增长、投入度的自适应变化以及已有技能熟练度变化等实际因素, 通过寻找最佳员工任务分配方案, 最小化软件项目的工期和成本. 为求解该模型, 提出一种引入问题启发信息的离散人工蜂群算法. 将多元学习策略应用于引领蜂阶段, 在保证种群多样性的同时, 加强算法全局搜索能力. 在跟随蜂阶段采用一种基于启发信息的变异机制, 保留最优个体中契合度较高的员工信息, 并根据不同个体目标值的优劣采用相异的变异方式, 针对性地进行搜索, 以增强算法的局部寻优能力. 实验结果表明, 与已有算法相比, 所提算法在不同规模的软件项目调度问题中均能够搜索到更优的分配方案.

**关键词:** 软件项目调度; 新技能学习机制; 熟练度; 人工蜂群算法; 多元学习; 启发信息

中图分类号: TP301.6

文献标志码: A

DOI: 10.13195/j.kzyjc.2021.1319

开放科学(资源服务)标识码(OSID):



**引用格式:** 申晓宁, 姚铖滨, 徐继勇, 等. 考虑新技能学习机制的软件项目调度人工蜂群算法[J]. 控制与决策, 2023, 38(3): 790-796.

## Artificial bee colony algorithm for software project scheduling considering new skills learning mechanism

SHEN Xiao-ning<sup>1,2,3†</sup>, YAO Cheng-bin<sup>1</sup>, XU Ji-yong<sup>1</sup>, SONG Li-yan<sup>4</sup>, WANG Yu-fang<sup>1,2,3</sup>

- (1. School of Automation, Nanjing University of Information Science and Technology, Nanjing 210044, China;
2. Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology, Nanjing University of Information Science and Technology, Nanjing 210044, China;
3. Jiangsu Key Laboratory of Big Data Analysis Technology, Nanjing University of Information Science and Technology, Nanjing 210044, China;
4. Guangdong Provincial Key Laboratory of Brain-inspired Intelligent Computation, Southern University of Science and Technology, Shenzhen 518055, China)

**Abstract:** Considering the learning mechanism of new skills, a mathematical model of the software project scheduling problem is established. The model integrates some practical factors such as learning of new skills, increase of the new and existing skill proficiencies, and adaptive changes of dedications. Both duration and cost of the software project are minimized by finding the best assignment of employees to tasks. To solve the model, a discrete artificial bee colony algorithm incorporating heuristic information is proposed. A multi-learning strategy is applied to the employed bees phase to enhance the global search ability of the algorithm while maintaining the population diversity. In the onlooker bees phase, a mutation mechanism based on heuristic information is adopted, where information of the employees with higher fit in the optimal individual is retained, and distinct mutation operators are employed on different individuals based on their objective values to improve the local search ability. Experimental results show that compared with the existing methods, the proposed algorithm can find a better allocation in software project scheduling problems with increasing scales.

**Keywords:** software project scheduling; new skill learning mechanism; proficiency; artificial bee colony algorithm; multi-learning; heuristic information

收稿日期: 2021-07-28; 录用日期: 2021-12-30.

基金项目: 广东省重点实验室项目(2020B121201001); 国家自然科学基金项目(61502239, 62002148); 江苏省自然科学基金项目(BK20150924).

责任编辑: 林崇.

†通讯作者. E-mail: sxnystst@sina.com.

## 0 引言

软件项目调度(SPSP)是软件开发的重要环节,直接影响软件企业的经济效益和市场竞争能力.SPSP是指要找到一种合理的人力资源和任务的安排方式,最小化项目开发成本和持续时间<sup>[1]</sup>,同时满足人力资源和任务的相关约束.近年来,基于搜索的软件工程迅速发展,越来越多的国内外学者开始研究将软件项目调度问题建模为基于搜索的优化问题<sup>[1-3]</sup>.

自Alba等<sup>[1]</sup>提出软件项目调度问题后,很多文献对该问题进行了研究.从员工角度,García等<sup>[4]</sup>和Rodríguez等<sup>[5]</sup>将员工的已有技能水平分级,但员工的技能水平在项目过程中保持不变,与实际不符.对此,郭一楠等<sup>[6]</sup>提出考虑学习/遗忘的软件项目调度模型.从任务角度,Chang等<sup>[7]</sup>提出基于时间轴的模式,在若干离散时间单元内分别调度;Di等<sup>[8]</sup>将项目划分为工作包,将开发者依据技能分组.从目标角度,肖菁等<sup>[9]</sup>提出多项目并行的调度模型,但仅考虑了项目开发成本;Di等<sup>[8]</sup>引入时间碎片指标,以缩短空闲时间段.现有文献均将团队员工的技能集合覆盖项目的所有技能需求作为约束条件,但他们忽略了实际项目的完成可能还需要某些现有团队尚未具备的新技能.基于此,本文建立考虑新技能学习机制的软件项目调度模型(简称SPSP-NSL).

软件项目调度问题已被证明是NP-hard问题<sup>[10]</sup>,多数学者采用元启发式算法对该类问题进行求解.人工蜂群算法(artificial bee colony, ABC)是由Karaboga<sup>[11]</sup>提出的一种模拟蜜蜂采蜜行为的元启发式算法,该算法兼顾全局探索和局部开采,对于不同阶段可采用不同的策略,具有结构灵活、控制参数少、易于实现、寻优能力强的优点.目前许多文献对ABC算法的种群初始化<sup>[12]</sup>、混合搜索策略<sup>[13]</sup>及选择开采对象<sup>[14]</sup>等方面做了改进.

为了有效求解考虑新技能学习机制的软件项目调度模型,本文提出一种引入启发信息的离散人工蜂群算法(heuristic discrete artificial bee colony, HDABC).引领蜂阶段采用多元学习策略,增强算法的全局搜索能力;跟随蜂阶段采用基于启发信息的变异机制,加强算法局部寻优能力.与已有算法的对比结果表明,所提算法具有更高的求解精度.

## 1 考虑新技能学习机制的软件项目调度模型

本文所建软件项目调度模型SPSP-NSL中的新

技能学习机制包括如何选择员工学习新技能、新技能熟练度的增长及其投入度的自适应变化.

### 1.1 符号和定义

软件项目调度问题中的员工和任务具有不同的属性,它们的符号定义如下:

$E_i$ : 员工  $i, i = 1, 2, \dots, M$ ;

$T_j$ : 任务  $j, j = 1, 2, \dots, N$ ;

$S$ : 项目需求的技能总数;

$\text{pro}_i^k(t)$ :  $t$ 时刻  $E_i$ 对技能  $k$ 的熟练度,  $\text{pro}_i^k(t) \in [0, C], C = 5^{[7]}$ ,  $\text{pro}_i^k(t) = 0$ 表示  $E_i$ 没有技能  $k$ ;

$\text{skill}_i$ :  $E_i$ 已有的技能集合,  $\text{skill}_i = \{k | \text{pro}_i^k(t_0) > 0, k = 1, 2, \dots, S\}$ ,  $t_0$ 表示项目初始时刻;

$E_i^{\text{LA}}$ :  $E_i$ 的学习能力;

$\text{SD}^k$ : 第  $k$ 个技能的难度,  $\text{SD}^k \in [0, 1]$ ,  $\text{SD}^k$ 越大表明该技能的难度越大;

$E_i^{\text{maxd}}$ :  $E_i$ 对项目的最大投入度;

$E_i^{n-s}$ :  $E_i$ 正常工作时间的月工资;

$E_i^{o-s}$ :  $E_i$ 的月加班工资;

$\text{req}_j$ :  $T_j$ 所需的技能集合;

$T_j^{\text{eff}}$ :  $T_j$ 所需的工作量,单位是人×月;

TPG: 项目中各任务的优先级关系有向图.

假设员工的学习能力  $E_i^{\text{LA}} \sim N(\mu, \sigma^2)$ ,  $\mu = 0.5$ ,  $\sigma^2 = 0.15^{[15]}$ ,  $E_i^{\text{LA}}$ 越大表示其学习能力越强.  $E_i^{\text{maxd}} = 1, E_i^{\text{maxd}} > 1, E_i^{\text{maxd}} < 1$ 分别表示  $E_i$ 在该项目中全职工作,接受加班以及兼职工作.由  $\text{pro}_i^k(t)$ 可以进一步求出员工  $E_i$ 在  $t$ 时刻对任务  $T_j$ 的熟练程度<sup>[7]</sup>

$$E_{ij}^{\text{pro}}(t) = \prod_{k \in \text{req}_j \cap \text{skill}_i} \frac{\text{pro}_i^k(t)}{C},$$

且  $E_{ij}^{\text{pro}}(t) \in [0, 1]$ .

### 1.2 新技能的学习机制

本文采用让已有员工学习新技能的方式来满足项目的新技能需求,并提出一种新技能的学习机制.为了简化问题,以一个项目仅有一个新技能需求,且一个新技能只需一人学习为例.

#### 1.2.1 选择学习新技能的员工

考虑不同技能之间存在一定关联<sup>[16]</sup>,同时员工  $E_i$ 的每个已有技能  $k \in \text{skill}_i$ 在项目初始时刻  $t_0$ 都具有一定的初始熟练度  $\text{pro}_i^k(t_0)$ ,由此定义员工  $E_i$ 的技能  $k$ 与新技能  $z$ 之间的相似熟练度  $\text{rp}_i^{kz}$ 为

$$\text{rp}_i^{kz} = r^{kz} \cdot \text{pro}_i^k(t_0). \quad (1)$$

其中:  $r^{kz}$  为技能  $k$  与新技能  $z$  之间的关联程度<sup>[16]</sup>,  $r^{kz} \in [0, 1]$ ,  $r^{kz}$  越大, 表示两个技能越相关.

项目经理在选择一个员工去学习新技能时, 除了要考虑该员工已有技能与新技能  $z$  的相似熟练度较高以外, 还需顾及员工额外学习新技能会导致精力分散, 对项目的投入度将减少这一因素. 通常情况下, 如果一个员工对项目的总熟练度越高, 则他在项目投入度上的减小对项目工期产生的负面影响越大, 因此倾向于选择总熟练度较低的员工. 同时, 员工的学习能力  $E_i^{LA}$  越强, 越易掌握新技能. 综上, 定义员工  $E_i$  学习新技能  $z$  的合适度  $\alpha_i^z$  为

$$\alpha_i^z = E_i^{LA} \cdot \beta_i^z \cdot \frac{E_{\max}^{\text{ap}} - E_i^{\text{ap}}}{E_{\max}^{\text{ap}} - E_{\min}^{\text{ap}}}, \quad i = 1, 2, \dots, M. \quad (2)$$

其中:  $\beta_i^z = \max_{k \in \text{skill}_i} (r^{kz})$ , 表示员工  $E_i$  已有技能与新技能  $z$  相似熟练度的最大值, 该值越大,  $E_i$  对新技能  $z$  的相关知识储备越多;  $E_i^{\text{ap}} = \sum_{j=1}^N E_{ij}^{\text{pro}}(t_0)$  是  $E_i$  在  $t_0$  时刻对项目所有任务的总熟练度,  $E_{\max}^{\text{ap}} = \max_{i \in \{1, 2, \dots, M\}} E_i^{\text{ap}}$  和  $E_{\min}^{\text{ap}} = \min_{i \in \{1, 2, \dots, M\}} E_i^{\text{ap}}$  分别表示所有员工对项目总熟练度的最大和最小值. 选择  $\alpha_i^z$  最大的员工  $q$  学习新技能  $z$ , 即  $q = \arg \max_{i \in \{1, 2, \dots, M\}} \alpha_i^z$ .

### 1.2.2 新技能熟练度的增长

初始阶段是学习技能的基础知识, 技能熟练度增长速度较快; 当处于强化技能阶段时, 熟练度增长变缓; 最后, 熟练掌握此项技能, 熟练度趋于平稳值. 采用  $\tanh(\cdot)$  双曲正切函数来拟合新技能熟练度的上述增长趋势, 如下所示:

$$\text{pro}_q^z(t) = C \cdot \tanh\left(b_q^z(t - t_0) + \text{atanh}\left(\frac{\beta_q^z}{C}\right)\right), \quad t \geq t_0. \quad (3)$$

其中  $b_q^z = \frac{E_q^{LA}}{\text{SD}^z}$  表示员工  $E_q$  对新技能  $z$  的熟练度增长率<sup>[15]</sup>, 员工的学习能力  $E_q^{LA}$  越大, 技能难度  $\text{SD}^z$  越小, 则增长率  $b_q^z$  越大. 同时, 本文考虑已有技能与新技能间的知识关联, 将新技能熟练度增长曲线的起点值  $\text{pro}_q^z(t_0)$  定义为  $\beta_q^z$ , 即员工  $E_q$  已有技能与新技能  $z$  的相似熟练度最大值.

### 1.2.3 被选员工投入度的自适应变化

学习新技能的员工  $E_q$  需要花费时间去学习, 但人的精力有限, 导致  $E_q$  在  $t$  时刻对任务  $T_j$  的投入度  $d_{qj}(t)$  有所下降.  $d_{qj}(t)$  的自适应变化规律如下:

$$d_{qj}(t) = \frac{\text{pro}_q^z(t)}{C} x_{qj}, \quad j \in T_{\text{active\_set}}(t). \quad (4)$$

其中:  $x_{qj}$  表示项目调度方案中分配的  $E_q$  对  $T_j$  的投入度,  $T_{\text{active\_set}}(t)$  表示  $t$  时刻可执行任务集合.  $d_{qj}(t)$  的下降幅度随着  $E_q$  对新技能  $z$  掌握的熟练度  $\text{pro}_q^z(t)$  的提高而逐渐降低.

### 1.3 已有技能熟练度的变化

随着项目的推进, 员工反复使用已有技能, 或与同事不断交流, 他们已有技能熟练度会有小浮动的提升. 选用变化较为平缓的  $\ln(\cdot)$  函数来拟合员工已有技能的变化趋势. 员工  $E_i$  对已有技能  $k$  的熟练度  $\text{pro}_i^k(t)$  的变化函数如下所示:

$$\text{pro}_i^k(t) = \min\{C, \ln(b_i^k(t - t_0) + e^{\text{pro}_i^k(t_0)})\}, \quad t \geq t_0. \quad (5)$$

其中  $b_i^k = \frac{E_i^{LA}}{\text{SD}^k}$  表示员工  $E_i$  对技能  $k$  的熟练度增长率,  $\text{pro}_i^k(t_0)$  表示  $E_i$  对技能  $k$  的初始熟练度.

### 1.4 数学模型

所提模型的决策变量是员工-任务分配矩阵  $X = (x_{ij})_{M \times N}$ ,  $i = 1, 2, \dots, M$ ,  $j = 1, 2, \dots, N$ .  $x_{ij}$  表示员工  $E_i$  对任务  $T_j$  的投入度,  $x_{ij} \in [0, E_i^{\text{maxd}}]$ . 本文模型的目标函数和约束条件如下:

$$\min f = \omega_t \cdot t_{\text{dur}} + \omega_c \cdot \text{cost}. \quad (6)$$

$$\text{s.t.} \quad \sum_{i=1}^M x_{ij} > 0, \quad j = 1, 2, \dots, N; \quad (7)$$

$$\text{req}_j \subseteq \bigcup_{i \in \{1, 2, \dots, M\}} \{\text{skill}_i | x_{ij} > 0\}, \quad j = 1, 2, \dots, N; \quad (8)$$

$$\sum_{j \in T_{\text{active\_set}}(t)} x_{ij} \leq E_i^{\text{maxd}}, \quad i = 1, 2, \dots, M; \quad (9)$$

$$\sum_{i=1, i \neq q}^M \text{pro}_i^z = 0, \quad q \in \{1, 2, \dots, M\}. \quad (10)$$

式(6)中的目标对项目工期  $t_{\text{dur}}$  和成本  $\text{cost}$  加权求和, 设  $\omega_t = 10^{-1}$ ,  $\omega_c = 10^{-5}$ <sup>[17]</sup>; 式(7)表示每一个任务至少有一个员工参与; 式(8)表示分配到某任务中的员工技能集合必须包含任务所需技能; 式(9)表示每个员工对当前可执行任务的总投入度不得超过其最大投入度, 若超过, 则通过修复法处理不可行解<sup>[3]</sup>; 式(10)限制每个新技能  $z$  只需一人学习.

## 2 求解SPSP-NSL的HDABC算法

针对所提软件项目调度模型 SPSP-NSL, 对引领蜂和跟随蜂阶段分别改进, 提出一种引入启发信息的

离散人工蜂群算法HDABC.

## 2.1 HDABC算法框架

求解所提模型的HDABC算法框架如图1所示.

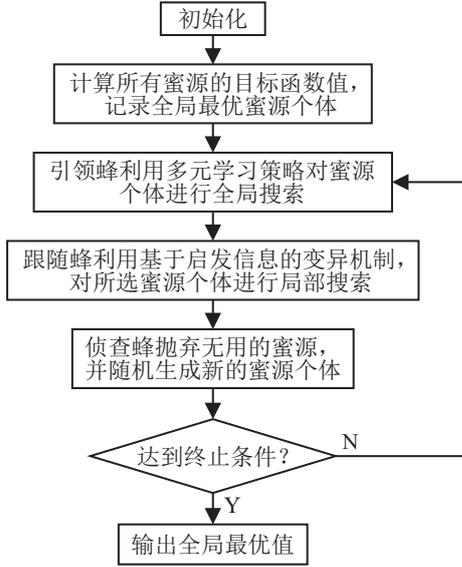


图1 HDABC算法框架

## 2.2 编码和解码

采用整数编码,对于包含  $M$  个员工、 $N$  个任务的软件项目调度问题,每个个体编码均为  $M \times N$  的矩阵  $Y = (y_{ij})_{M \times N}$ ,  $i = 1, 2, \dots, M, j = 1, 2, \dots, N$ . 其中:  $y_{ij} \in \{0, 1, 2, \dots, h\}$ ,  $h$  表示解的粒度,即将员工  $E_i$  的最大投入度  $E_i^{\max d}$  平均分为  $h$  份.  $y_{ij}$  表示  $E_i$  将自己最大投入度  $E_i^{\max d}$  的  $y_{ij}/h$  奉献给任务  $T_j$ . 本文设  $h$  为  $10^{[3]}$ . 解码时,通过比例  $y_{ij}/h$  将个体矩阵转化为员工-任务的分配矩阵  $X = (x_{ij})_{M \times N}$ ,即

$$x_{ij} = y_{ij}/h \cdot E_i^{\max d}. \quad (11)$$

## 2.3 引领蜂阶段的多元学习策略

所提算法在引领蜂阶段采用多元学习策略  $M\_Learning$ ,每一只引领蜂利用另一只引领蜂共享的蜜源信息对其所负责的蜜源进行开采,并对开采前与其开采后新生成的蜜源进行贪婪选择,留下适应度值较优的蜜源个体<sup>[11]</sup>.

多元学习策略包括随机学习、精英学习和优值学习3种算子,它们的学习方法相同,均为将另一个蜜源的某行或列替换到当前蜜源的相应位置.3种学习策略的操作对象不同:随机学习是引领蜂利用任一不同于当前蜜源的随机蜜源进行学习.本文将蜜源个体按目标值从优到劣排序,排在前  $10\% \cdot SN$  ( $SN$  为蜜源数量)的个体作为精英蜜源<sup>[14]</sup>.精英学习是引领蜂利用任一不同于当前蜜源的精英蜜源进行学习.优值学习是引领蜂利用目标值优于当前蜜源的

随机蜜源进行学习,若当前蜜源本身是全局最优个体,则进行随机学习.

随机学习可以保证种群的多样性,但收敛速度较慢,而蜜源向精英个体或更优个体学习可以加快收敛,但易陷入局部最优.考虑3种学习方式的优缺点,将它们混合构成多元学习策略.为了提高算法的全局探索能力,为随机学习设置较大的选择概率  $P_r$ .由于精英学习和优值学习都能引导种群的搜索方向,为两者设置相同的选择概率  $P_e = P_s = \frac{1 - P_r}{2}$ .采用轮盘赌策略选择相应的学习算子,在保证种群多样性的同时,加快种群收敛.

## 2.4 跟随蜂阶段基于启发信息的变异机制

基本ABC算法中的跟随蜂搜索方向随机性较大,求解精度低.对此,本文利用启发信息,设计一种应用于跟随蜂阶段的变异机制  $H\_Mutate$ .

1) 选择变异个体:由于排序选择能够使得算法跳出局部最优且操作简单<sup>[14]</sup>,本文在跟随蜂阶段使用它选择变异蜜源.根据蜜源目标值排序位置的选择概率采用轮盘赌确定需要进行变异的蜜源  $YP$ ,使得优势个体被选择的概率不受目标值变化的影响.

2) 基于契合度的员工选择:员工  $E_i$  对任务  $T_j$  的契合度  $wf_{ij}$  定义如下:

$$wf_{ij} = \lambda \cdot E_{ij}^{pro}(t_0) - \eta \cdot E_i^{n-s} - \gamma \cdot E_i^{o-s}. \quad (12)$$

为了权衡式(12)中各项的相对重要程度,取  $\lambda = 10$ ,  $\eta = 10^{-4}$ ,  $\gamma = 10^{-5}$ .当  $wf_{ij} < 0$  时,令  $wf_{ij} = 0$ .由式(12), $E_i$  对  $T_j$  的熟练程度越高且工资越低,则  $E_i$  对  $T_j$  的契合度越高.对每个  $T_j$  依照  $wf_{ij}$  通过轮盘赌选择一个员工  $E_{u_j}$ ,并保留其序号,即  $index_j = u_j$ .

3) 考虑个体目标值优劣的变异:对于第1)步选择出的蜜源  $YP$ ,依照变异概率  $P_m = \frac{1}{M \cdot N}$  确定出需要变异的元素  $yp_{ij}$ .此时,若  $i = index_j$  (即员工  $i$  是步骤2)基于契合度选择出的员工),则  $yp_{ij}^{new} = y_{ij}^{best}$ ,其中  $y_{ij}^{best}$  表示最优蜜源  $Y^{best}$  中的元素,该操作保留了最优蜜源中契合度高的员工信息.若  $i \neq index_j$ ,当前蜜源  $YP$  的目标排序序号位于前  $50\%$ ,则  $yp_{ij}$  通过下式变异:

$$L = yp_{ij} + N(0, 1) \cdot yp_{ij},$$

$$yp_{ij}^{new} = \min(h, \text{round}(L) \cdot \text{sgn}(L)); \quad (13)$$

否则通过下式变异:

$$yp_{ij}^{new} = \min(h, \text{round}(yp_{ij} + \text{rand} \cdot (y_{ij}^{best} - yp_{ij}))). \quad (14)$$

其中:  $N(0, 1)$  是标准高斯分布,  $\text{round}(\cdot)$  是四舍五入取整函数,  $\text{sgn}(\cdot)$  是符号函数,  $\min(\cdot)$  是为了使个体中的元素值不超过上限值  $h$  (见 2.2 节), 随机数  $\text{rand} \in [0, 1]$ . 式 (13) 对适应度较好的蜜源进行高斯扰动, 以避免算法陷入局部最优. 式 (14) 令适应度较差的蜜源向最优蜜源学习, 以引导种群快速收敛.

### 3 实验仿真与结果分析

为验证所提算法 HDABC 的性能, 在同一实验环境下, 设计两组实验: 1) 改进策略有效性的验证; 2) 所提算法 HDABC 的整体性能验证. 实验所用算例为 12 个采用文献 [1] 中算例生成器生成的软件项目调度随机算例和 3 个现实世界中的商业软件建设项目实例<sup>[2]</sup>. 本文算法蜜源数量  $SN = 100$ , 每个蜜源的最大未更新次数  $\text{limit} = D \cdot SN^{[18]}$ , 其中  $D$  表示个体的维

度,  $D = M \cdot N$ . 随机学习算子的选择概率  $P_r$  是新引入的参数, 通过实验分析取  $P_r = 0.6$ .

#### 3.1 改进策略的有效性验证

用随机学习、精英学习和优值学习 3 种策略分别替换 M\_Learning, 产生 3 种比较算法 HDABC-UR、HDABC-UE 和 HDABC-US, 用文献 [3] 的变异方式代替 H\_Mutate, 产生第 4 种比较算法 HDABC-H. 每种算法在不同算例上分别运行 30 次, 每次的终止条件为目标评价次数达到 50 000 次. 计算各算法搜索到的目标值的平均值  $\text{avg}$  和最优值  $\text{best}$ , 将  $\text{best}$  和  $\text{avg}$  的最好值加粗表示. 为显著对比 5 种算法的优劣, 引入显著水平为 0.05 的 Wilcoxon 秩和检验对实验结果进行统计测试. 其中: “+” 表示 HDABC 显著优于对比算法, “=” 表示两者无明显差异, “-” 表示 HDABC 显著劣于对比算法, 实验结果见表 1.

表 1 所提算法和替换单一策略算法的对比结果

测试算例	HDABC		HDABC-UR		HDABC-UE		HDABC-US		HDABC-H	
	best	avg	best	avg	best	avg	best	avg	best	avg
10E-10T-5S	<b>4.5093</b>	<b>4.9824</b>	4.6883	5.0805+	4.7728	5.1689+	4.6597	5.0901+	4.7259	5.1278+
10E-15T-5S	<b>9.3042</b>	<b>9.5555</b>	9.4572	9.6316+	9.3780	9.6248+	9.3960	9.6177+	9.4977	9.6973+
10E-15T-10S	5.7930	<b>5.9542</b>	5.8646	6.0464+	5.7778	5.9867=	<b>5.7238</b>	6.0189+	5.9949	6.1261+
15E-20T-5S	<b>9.9531</b>	<b>10.6135</b>	10.6364	10.7787+	10.2582	10.7023+	10.3081	10.6743=	10.4492	10.7886+
15E-20T-10S	9.8263	<b>10.1653</b>	10.1049	10.3847+	<b>9.7620</b>	10.2113=	9.9137	10.2380=	10.1854	10.5045+
15E-30T-5S	<b>11.3442</b>	<b>11.8065</b>	11.8085	13.2474+	11.4833	12.5648+	11.3472	12.2970+	11.4557	12.7667+
15E-30T-10S	<b>13.2970</b>	<b>14.7506</b>	13.7411	15.3520+	13.4255	15.5578+	13.3850	15.5251+	13.8857	16.1399+
20E-30T-5S	<b>12.1490</b>	<b>15.2273</b>	14.1795	16.4780+	12.7535	16.1232+	12.2780	15.8732+	12.8835	16.4071+
20E-30T-10S	<b>14.9195</b>	17.5998	17.8379	18.1980+	17.1377	<b>17.5303</b> -	16.5674	17.5663=	16.7602	18.1133+
20E-40T-10S	<b>16.5835</b>	<b>17.7539</b>	17.1519	18.4596+	17.5230	17.9979+	17.7336	18.0751+	17.9782	18.4437+
25E-40T-10S	18.0161	<b>18.3155</b>	17.9817	18.9348+	<b>17.3031</b>	18.4316=	17.9835	18.5595+	18.5646	19.0076+
30E-40T-10S	<b>18.0125</b>	<b>18.8036</b>	18.1791	19.2998+	18.1469	18.8905=	18.1148	18.9496=	19.0230	19.4903+
Real-1	<b>6.9502</b>	<b>7.2813</b>	7.1583	7.4633+	7.1508	7.3651+	6.9973	7.3482+	7.4307	7.6756+
Real-2	<b>1.5279</b>	<b>1.5650</b>	1.5792	1.6575+	1.5297	1.5708=	1.5349	1.5725=	1.6346	1.6896+
Real-3	<b>1.9119</b>	<b>1.9899</b>	1.9202	2.0271+	1.9267	2.0723+	1.9225	2.0335+	1.9927	2.0885+
总计 +/ = / -			15/0/0		9/5/1		10/5/0		15/0/0	

由表 1 和统计测试结果可知, HDABC 在绝大多数算例中显著优于其他对比算法. 究其原因, HDABC-UR 在引领蜂阶段只采用随机学习, 种群多样性较好, 收敛性能较差. HDABC-UE 和 HDABC-US 分别只采用精英学习和优值学习, 易造成种群多样性的流失. 所提算法采用 3 种学习机制相融合的多元学习策略 M\_Learning, 随机学习算子的被选概率较高, 有效地增加了种群的多样性; 同时, 小概率的精英学习和优值学习算子能够引导种群向正确的方向移动, 进一步提高了算法的搜索能力. 在变异机制方面, 文

献 [3] 的变异方式随机性较大, 而所提基于启发信息的变异算子 H\_Mutate, 通过排序选择保证优势个体的选择概率, 利用启发信息保留最优个体中契合度高的员工信息, 同时根据当前蜜源个体的目标值优劣, 选择不同的变异方式, 加强对算法搜索方向的引导, 提高了算法局部寻优能力.

#### 3.2 所提 HDABC 算法的性能验证

为验证所提算法 HDABC 在求解 SPSP-NSL 时的性能, 将其与 (1+1)EA<sup>[3]</sup> 和 RLS<sup>[3]</sup>、Pop-EA<sup>[3]</sup>、TSaABC<sup>[13]</sup> 和 ABCMSSCE<sup>[18]</sup> 这 5 个具有代表性的算

法分别在12个随机算例和3个实例上进行求解.所有对比算法均使用与HDABC相同的个体编码解码方式,终止条件均为目标评价次数达到50000次,其

余参数与原文献相同.将它们分别应用于本文建立的模型,并与所提算法进行比较.每种算法在不同算例上分别运行30次,实验结果见表2.

表2 HDABC与对比算法的实验结果

测试算例	HDABC		(1+1)EA		RLS		Pop-EA		TSaABC		ABCMSSCE	
	best	avg	best	avg	best	avg	best	avg	best	avg	best	avg
10E-10T-5S	<b>4.5093</b>	<b>4.9824</b>	5.8404	6.0901+	5.8164	6.1870+	5.4404	6.0306+	5.7545	5.9224+	5.5401	5.7970+
10E-15T-5S	<b>9.3042</b>	<b>9.5555</b>	9.8368	10.2134+	9.9528	10.4057+	9.9379	10.1840+	10.0846	10.2062+	9.7087	10.0488+
10E-15T-10S	<b>5.7930</b>	<b>5.9542</b>	6.8266	7.3077+	6.7714	7.4370+	6.8388	7.1420+	7.0605	7.2509+	6.0520	6.6043+
15E-20T-5S	<b>9.9531</b>	<b>10.6135</b>	11.0837	11.3871+	11.1712	11.5192+	11.0006	11.2621+	11.3018	11.4946+	10.4364	10.9720+
15E-20T-10S	<b>9.8263</b>	<b>10.1653</b>	10.7455	11.3109+	10.9122	11.5706+	10.8344	11.1922+	11.2642	11.5409+	10.3439	10.6759+
15E-30T-5S	<b>11.3442</b>	<b>11.8065</b>	15.8153	16.2903+	16.0114	16.4985+	15.2258	16.2458+	17.7620	18.2062+	11.4126	12.1133=
15E-30T-10S	<b>13.2970</b>	<b>14.7506</b>	18.9577	19.5818+	19.0365	19.9679+	18.7599	19.7275+	21.4815	22.1116+	13.4856	15.7115+
20E-30T-5S	<b>12.1490</b>	<b>15.2273</b>	16.9638	17.6011+	17.0848	17.8257+	16.9741	17.7372+	19.6173	19.9189+	12.1691	15.6169=
20E-30T-10S	<b>14.9195</b>	<b>17.5998</b>	18.1879	18.8653+	18.3362	19.0093+	18.6695	19.3333+	21.0403	21.7438+	16.7861	17.8222=
20E-40T-10S	<b>16.5835</b>	<b>17.7539</b>	18.1144	18.7259+	18.1614	18.7781+	18.0544	18.9841+	21.9009	22.3754+	17.2956	17.8172=
25E-40T-10S	18.0161	<b>18.3155</b>	18.1102	18.5761+	18.1379	18.7052+	18.4415	19.1738+	22.1454	22.5728+	<b>17.6622</b>	18.5204+
30E-40T-10S	<b>18.0125</b>	<b>18.8036</b>	18.1886	19.1180+	18.1475	19.0215+	18.6334	19.9209+	22.8514	23.2384+	18.3426	18.8464=
Real-1	<b>6.9502</b>	<b>7.2813</b>	8.6701	9.3933+	8.7835	9.6247+	8.4284	9.0556+	9.0675	9.3625+	7.3244	7.8604+
Real-2	<b>1.5279</b>	<b>1.5650</b>	1.7140	1.8708+	1.6669	1.9044+	1.7523	1.8568+	1.8603	1.8890+	1.5614	1.6203+
Real-3	<b>1.9119</b>	<b>1.9899</b>	2.3375	2.5421+	2.3833	2.7500+	2.3394	2.5176+	2.5217	2.5912+	2.0468	2.2046+
总计 +/ = / -			15/0/0		15/0/0		15/0/0		15/0/0		10/5/0	

由表2可知,所提算法HDABC与其余5种算法相比,在不同规模的算例上均取得了最佳的平均值avg,在除算例25E-40T-10S外的其余所有算例上均取得了最佳的best. Wilcoxon秩和检验结果表明,HDABC在绝大多数算例中的求解精度均显著优于其他对比算法.所提算法的优越性能得益于多种改进策略的综合使用.引领蜂阶段采用的多元学习策略维护了种群多样性,强化了算法全局搜索能力.跟随蜂阶段采用的变异机制为不同类型的个体选择相异的变异方式,在个体附近采用适合其特征的方式进行挖掘,提升了算法的局部搜索能力.此外,排序选择能够获得优势个体在算法后期仍然保持较大的选择概率,而不受种群中个体目标值差距缩小的影响,从而进一步提高算法的寻优能力.

#### 4 结论

本文建立了考虑新技能学习机制的软件项目调度模型,引入了4个实际因素:1)项目开发中存在新技能需求;2)如何选择适当的员工学习新技能;3)新技能熟练度的增长及员工投入度的自适应变化;4)已有技能熟练度的变化.提出一种引入启发信息的离散人工蜂群算法求解所提模型,采用了2种改进

策略:1)在引领蜂阶段引入多元学习策略,提升算法的全局搜索能力;2)在跟随蜂阶段设计基于启发信息的变异机制,增强对局部搜索方向的引导.实验结果表明,所提算法能在不同规模的问题中均求得较好的员工-任务分配方案.

#### 参考文献(References)

- [1] Alba E, Chicano JF. Software project management with GAs[J]. Information Sciences, 2007, 177(11): 2380-2401.
- [2] Chen W N, Zhang J. Ant colony optimization for software project scheduling and staffing with an event-based scheduler[J]. IEEE Transactions on Software Engineering, 2013, 39(1): 1-17.
- [3] Minku L L, Sudholt D, Yao X. Improved evolutionary algorithm design for the project scheduling problem based on runtime analysis[J]. IEEE Transactions on Software Engineering, 2014, 40(1): 83-102.
- [4] García-Nájera A, Gomez M. A multi-objective genetic algorithm for the software project scheduling problem[C]. Proceedings of the Mexican International Conference on Artificial Intelligence. Tuxtla Gutiérrez, 2014: 13-24.
- [5] Rodríguez D, Ruiz M, Riquelme J C, et al. Multi-objective simulation optimisation in software

- project management[C]. Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation. Dublin, 2011: 1883-1890.
- [6] 郭一楠, 季俊华, 吉建娇, 等. 考虑学习/遗忘特性的软件项目调度多目标优化方法[J]. 控制与决策, 2018, 33(2): 203-210.  
(Guo Y N, Ji J H, Ji J J, et al. Multi-objective software project scheduling optimization method with the learning and forgetting effect[J]. Control and Decision, 2018, 33(2): 203-210.)
- [7] Chang C K, Jiang H, Di Y, et al. Time-line based model for software project scheduling with genetic algorithms[J]. Information and Software Technology, 2008, 50(11): 1142-1154.
- [8] Di Penta M, Harman M, Antoniol G. The use of search-based optimization techniques to schedule and staff software projects: An approach and an empirical study[J]. Software: Practice and Experience, 2011, 41(5): 495-519.
- [9] 肖菁, 吴洲. 基于时间轴的软件多项目任务调度遗传算法[J]. 计算机科学, 2012, 39(12): 133-138.  
(Xiao J, Wu Z. Software multi-project scheduling genetic algorithms based on a time-line model[J]. Computer Science, 2012, 39(12): 133-138.)
- [10] Crawford B, Soto R, Johnson F, et al. A max-min ant system algorithm to solve the software project scheduling problem[J]. Expert Systems with Applications, 2014, 41(15): 6634-6645.
- [11] Karaboga D. An idea based on honey bee swarm for numerical optimization[R]. Kayseri: Erciyes University, 2005.
- [12] Xiang W L, An M Q. An efficient and robust artificial bee colony algorithm for numerical optimization[J]. Computers and Operations Research, 2013, 40(5): 1256-1265.
- [13] Song X Y, Zhao M, Yan Q F, et al. A high-efficiency adaptive artificial bee colony algorithm using two strategies for continuous optimization[J]. Swarm and Evolutionary Computation, 2019, 50: 100549.
- [14] 孔德鹏, 常天庆, 戴文君, 等. 基于排序选择和精英引导的改进人工蜂群算法[J]. 控制与决策, 2019, 34(4): 781-786.  
(Kong D P, Chang T Q, Dai W J, et al. An improved artificial bee colony algorithm based on the ranking selection and the elite guidance[J]. Control and Decision, 2019, 34(4): 781-786.)
- [15] Shen X N, Minku L L, Marturi N, et al. A Q-learning-based memetic algorithm for multi-objective dynamic software project scheduling[J]. Information Sciences, 2018, 428: 1-29.
- [16] Otero L D, Centeno G, Ruiz-Torres A J, et al. A systematic approach for resource allocation in software projects[J]. Computers & Industrial Engineering, 2008, 56(4): 1333-1339.
- [17] Shen X, Zhang M. A differential evolution-based memetic algorithm for project scheduling problems[J]. International Journal of Innovative Computing and Applications, 2015, 6(3/4): 229-239.
- [18] 王志刚, 尚旭东, 夏慧明, 等. 多搜索策略协同进化的人工蜂群算法[J]. 控制与决策, 2018, 33(2): 235-241.  
(Wang Z G, Shang X D, Xia H M, et al. Artificial bee colony algorithm with multi-search strategy cooperative evolutionary[J]. Control and Decision, 2018, 33(2): 235-241.)

### 作者简介

申晓宁(1981—), 女, 教授, 博士, 从事计算智能、多目标优化、调度技术等研究, E-mail: sxnystsyt@sina.com;

姚铨滨(1998—), 男, 硕士生, 从事群智能计算、调度技术的研究, E-mail: 1270089446@qq.com;

徐继勇(1998—), 男, 硕士生, 从事群智能计算、调度技术的研究, E-mail: 853724424@qq.com;

宋丽妍(1986—), 女, 助理研究员, 博士, 从事软件缺陷预测、工作量估计等研究, E-mail: songly@sustech.edu.cn;

王玉芳(1979—), 女, 副教授, 博士, 从事生成调度、智能调度算法等研究, E-mail: qing\_0325@163.com;

(责任编辑: 齐 霖)