

# 基于 IWOA 群智感知中数量敏感的任务分配方法

蒋伟进<sup>1,2,3</sup>, 张婉清<sup>1,2</sup>, 陈萍萍<sup>2,3</sup>, 陈君鹏<sup>2,3</sup>, 孙永霞<sup>2,3</sup>, 刘 权<sup>1</sup>

(1. 湖南工商大学前沿交叉学院, 湖南长沙 410205; 2. 新零售虚拟现实技术湖南省重点实验室, 湖南长沙 410205;  
3. 湖南工商大学计算机学院, 湖南长沙 410205)

**摘 要:** 随着移动终端的大规模普及, 群智感知技术以其高效且成本低廉的优势逐渐取代现有的静态传感器, 成为一种新兴的数据收集方式. 如何在保证质量、降低成本的前提下, 把感知任务分配给最佳执行用户以实现用户任务完成数量的最大化, 是数量敏感任务分配问题研究的重点. 基于此, 提出一种结合非线性递减收敛因子、最优局部抖动以及动态位置更新三种操作的改进鲸鱼优化算法(Improved Whale Optimization Algorithm, IWOA), 并将其用于求解所提出的任务分配问题. 对数量敏感的任务分配问题进行建模, 根据用户与任务间的适应程度, 定义空间匹配度与技能匹配度. 在用户执行任务的过程中考虑到用户的学习能力, 引入技能更新机制对用户已有技能进行及时更新, 以此提高任务分配的效率. 综合考虑预算、用户在线时长以及感知任务完成质量, 对最大化任务完成数量的任务分配问题进行合理定义, 并从为任务选择最佳执行用户的角度出发, 设计一种基于优先级的用户选择策略, 以实现在保证感知任务基本完成质量的前提下, 降低任务分配的成本. 在最优任务分配方案的求解过程中, 利用改进算法对每次迭代初始的任务序列进行不断优化, 经过有限次迭代即可得到最终结果. 将改进算法与其他优化算法在相同环境下进行对比实验, 结果表明改进算法在求解任务分配问题时具有更高的性能.

**关键词:** 数量敏感; 任务分配; 改进的鲸鱼优化算法; 技能更新机制; 优先级

中图分类号: TP393

文献标识码: A

文章编号: 0372-2112(2022)10-2489-14

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.12263/DZXB.20210198

## Quantity Sensitive Task Allocation Method Based on IWOA in Group Intelligence Perception

JIANG Wei-jin<sup>1,2,3</sup>, ZHANG Wan-qing<sup>1,2</sup>, CHEN Ping-ping<sup>2,3</sup>, CHEN Jun-peng<sup>2,3</sup>, SUN Yong-xia<sup>2,3</sup>, LIU Quan<sup>1</sup>

(1. College of Frontier Intersection, Hunan University of Technology and Business, Changsha, Hunan 410205, China;

2. Key Laboratory of Hunan Province for New Retail Virtual Reality Technology, Changsha, Hunan 410205, China;

3. School of Computer Science, Hunan University of Technology and Business, Changsha, Hunan 410205, China)

**Abstract:** With the large-scale popularity of mobile terminals, group intelligence sensing technology gradually replaces the existing static sensors with its advantages of high efficiency and low cost, becoming an emerging data collection method. How to assign perception tasks to the best performing users under the premise of ensuring quality and reducing costs to maximize the number of user tasks completed is the focus of the research on quantity sensitive task allocation. Based on this, a solution based on the improved whale optimization algorithm(IWOA) that combines the three operations of nonlinear decreasing convergence factor, optimal local jitter, and dynamic position update is put forward, which is used to solve the proposed task allocation problem. First, modeling the quantity sensitive task allocation problem, and then defining the spatial matching degree and skill matching degree according to the degree of adaptation between the user and the task. Taking into account the user's learning ability during the user's task execution, the skill update mechanism is introduced to update the user's existing skills in a timely manner, so as to improve task allocation effectiveness. Secondly, comprehensively considering the budget, the user's online time and the perceived task completion quality, and reasonably defining the task allocation problem that maximizes the number of tasks completed. In addition, from the perspective of selecting the best per-

收稿日期: 2021-02-02; 修回日期: 2021-08-04; 责任编辑: 孙瑶

基金项目: 国家自然科学基金面上项目(No. 61772196); 湖南省自然科学基金面上项目(No. 2020JJ4249); 湖南省社会科学基金重点项目(No. 2016ZDB006); 湖南省社会科学成果评审委员会课题重点项目(湘社评 19ZD1005); 湖南省研究生科研创新项目(No. CX20211108, No. CX20211102)

forming user for the task, designing a user selection strategy based on user's priority to reduce the cost of task allocation while ensuring the quality of the perceived task is basically completed. Then, in the process of solving the optimal task allocation plan, the improved algorithm is used to continuously optimize the initial task sequences of each iteration, and the final result can be obtained after a limited number of iterations. Finally, the improved algorithm is compared with other optimization algorithms in the same environment, and the results show that the improved algorithm has higher performance in solving task allocation problem.

**Key words:** quantity sensitive; task allocation; improved Whale Optimization Algorithm; skill update mechanism; priority

## 1 引言

任务分配是群智感知领域中研究的一个基本问题,其目的是在随机环境下找到一种满足若干约束条件的最优任务分配方案,属于 NP (Non-deterministic Polynomial) 难题. 数量敏感的任务分配问题通常要考虑更多的因素,比如用户有限的工作时间、任务分配的预计成本以及任务所需的基本感知数据质量等. 如何在保证感知数据基本质量、降低任务分配成本的前提下,把感知任务分配给最佳执行用户以最大化用户在有限工作时间内的任务完成数量,是数量敏感任务分配问题研究的重点.

在相关研究中发现,根据用户与任务某些属性的匹配程度把任务分配给最佳用户不但能提升所获得感知数据的质量,还能在一定程度上降低任务分配的成本. 对此,李卓等人<sup>[1]</sup>认为在感知数据质量能够满足任务基本需要的前提下,用户执行任务付出的代价越少意味着其对应成本越低,因此更容易得到任务分配者的认可,并在此基础上提出了位置相关的在线多任务分配算法;杜扬等人<sup>[2]</sup>则将任务所覆盖区域划分成若干个互不冲突的子区间,并求取各个子区间上基于地理位置的最佳用户,最后把这些最佳用户的并集作为最终的任务分配结果;Estrada 等人<sup>[3]</sup>提出了一种群智感知系统中基于位置的任务分配服务计算框架,该框架依赖于4个部分即招募算法、排队方案、任务委托机制以及信誉管理组件,仿真结果表明,所提出的框架可以最大程度地提高信息质量、减少执行任务的预算和响应时间;Gong 等人<sup>[4]</sup>根据已知的用户行驶路径设计了4种不同的在线任务分配算法,其中 B-DBA 算法同时考虑了用户行驶距离与任务完成质量,并在此基础上结合了生物启发式搜索功能,有效提升了任务分配的性能.

除了考虑用户与任务之间的位置匹配度,Zhao 等人<sup>[5]</sup>从人类的同质性出发,将社交网络中朋友之间的关系应用于任务分配场景,提出了一种基于好友关系的任务分配算法,并在合成数据集和真实数据集 Yelp 上分别进行了比较实验,从而验证了该分配机制的有效性和适用性;Tu 等人<sup>[6]</sup>希望根据任务的特征将其分配给合适的用户,以便在降低成本的同时增加感知数据

的质量,并基于此提出一种新颖的任务分配策略,通过捕获任务和用户之间的复杂交互,实现将任务动态地分配给一组用户;Zhu 等人<sup>[7]</sup>发现在面对一些复杂感知任务时,用户的完成度普遍偏低,考虑到用户之间的异质性,提出根据不同任务的特点,将其分别分配给具备特定技能的用户,从而降低任务分配的成本;Alsayneh 等人<sup>[8]</sup>则通过引入任务组合机制为每个用户生成个性化的任务摘要(称为复合任务)来改善任务分配,并提出了最适合用户的最优化以及最多样化的复合任务优化问题. 但是,按照文献[7,8]所提出的任务分配方法,如果发生用户所拥有技能与感知任务部分匹配的情况,则这些任务将无法被分配,从而导致任务分配效率低下,降低了系统整体的任务完成数量. 另外,由于用户在任务分配过程中具有一定的移动性,考虑到用户特有的学习能力,随着其完成任务数量的增加,用户对已有技能的熟练度也会逐渐提升并学习到一些新技能. 因此,在参照用户与任务间的匹配度进行任务分配时,用户的学习能力在技能提升方面所发挥的作用不可忽略,但文献[7,8]均未提及.

目前,大部分任务分配问题的关注点都集中在质量<sup>[9-11]</sup>或成本<sup>[12-14]</sup>两方面,同时也存在一些将两因素<sup>[15-17]</sup>进行联合考虑的研究. 然而,在实际情况中,用户用来完成任务的时间是有限的,因此,如何将任务合理地分配给用户使其在有限时间内完成较多的任务是群智感知系统中需要解决的问题,文献[18]对此进行了详细讨论. 此外,在现有针对任务分配的研究工作中虽然存在同时考虑两个方面的情况,但几乎没有研究对以上所提的三个方面进行同时优化.

因此,针对现有研究中的不足,本文综合考虑了质量、成本以及任务完成数量,从用户与任务间的属性匹配程度出发对数量敏感的任务分配问题进行了完整建模,并在此基础上提出了一种改进的鲸鱼优化算法(Improved Whale Optimization Algorithm, IWOA). 本文的主要贡献如下:

(1) 由于用户与任务之间地理位置以及技能信息存在着差异,引入了空间匹配度与技能匹配度,在技能匹配度中,认为感知任务可以被技能种类与其内容部分匹配的用户执行;

(2)考虑到用户的学习能力,对用户在执行任务过程中的技能更新方式进行建模,提出了相应的技能更新机制;

(3)结合质量、成本以及任务完成数量三方面,定义了最大化任务完成数量的任务分配问题,并给出了相关优化模型;

(4)将任务分配问题的求解转变成成为任务选择合适的执行用户,在此基础上设计了一种基于优先级的用户选择策略,通过为任务序列中的任务依次选择对应优先级最高的用户,以实现在保证感知任务基本完成质量的前提下,降低任务分配的总成本;

(5)对基本鲸鱼优化算法(Whale Optimization Algorithm, WOA)进行改进并将其运用于所提出的任务分配问题,通过改进算法对每次迭代初始的任务序列进行不断优化,从而最大化系统中用户的任务完成数量;

(6)将 IWOA 与 WOA<sup>[19]</sup>、离散布谷鸟搜索算法(Discrete Cuckoo Search Algorithm, DCSA)<sup>[20]</sup>以及改进混合灰狼优化算法(Improved hybrid Grey Wolf Optimizer, IGWO)<sup>[21]</sup>算法在最大任务完成数量、总成本、数据质量三方面进行比较以验证改进鲸鱼优化算法的搜索性能。

## 2 数量敏感的任务分配模型

假设群智感知系统中有  $m$  个用户以及  $n$  个任务,  $U = \{u_1, u_2, \dots, u_m\}$  与  $S = \{s_1, s_2, \dots, s_n\}$  分别代表用户集与任务集. 其中, 用户  $u_i = \langle \text{Pos}_i, \text{Las}_i, \text{Tec}_i \rangle, i = 1, 2, \dots, m$ ,  $\text{Pos}_i$  为用户  $i$  所在位置,  $\text{Las}_i$  为用户  $i$  可工作时长,  $\text{Tec}_i = \langle \text{TecKind}_i, \text{TecProf}_i \rangle$  为用户  $i$  所具备技能的相关信息,  $\text{TecKind}_i$  为技能种类集,  $\text{TecProf}_i$  为技能熟练度集且  $0 < \text{TecProf}_i < 1$ ; 任务  $s_j = \langle \text{Pos}_j, \text{Con}_j, \text{Adr}_j \rangle, j = 1, 2, \dots, n$ ,  $\text{Pos}_j$  为任务  $j$  的发出位置,  $\text{Con}_j$  为任务  $j$  所需的以关键字形式存储的感知内容集,  $\text{Adr}_j$  为任务  $j$  需要获得的数据量. 为了简化表达, 令  $T_i$  表示已经分配给用户  $u_i$  的任务集,  $T_0$  表示待分配的任务集。

### 2.1 空间匹配度

假设用户与任务之间的位置关系如图 1 所示. 分析图 1 可知, 用户  $u_1$  与任务  $s_1$  距离最近, 与任务  $s_n$  距离最远. 因此, 用户  $u_1$  如果执行任务  $s_n$ , 则会比执行任务  $s_1$  花费更长的时间, 并需要更高的移动成本. 针对这种情况, 引入空间匹配度这一概念, 利用用户与任务间的距离作为衡量匹配度的重要指标。

用户与任务间的距离远近体现了用户与任务间的空间匹配程度<sup>[22]</sup>. 距离用户越近的任务, 由于耗费更少的移动时间, 用户执行该任务的可能性就越大, 从而降低了用户的移动成本, 用户与该任务间的空间匹配程度就越好; 距离用户较远的任务, 不但会增加用户的移

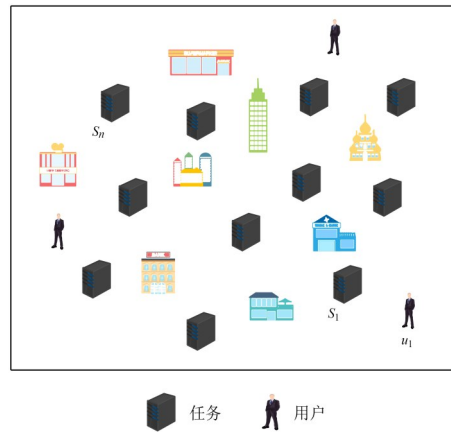


图1 用户-任务分布情况

动成本, 还会降低用户的执行效率, 因此, 用户与该任务间的空间匹配程度就越差. 结合实际情况, 给出用户与任务间的空间匹配度为

$$\text{spaMa}_{ij} = \frac{d_{\max} - \text{dist}_{ij}}{d_{\max}}, \text{dist}_{ij} \leq d_{\max} \quad (1)$$

其中,  $d_{\max}$  为用户  $u_i$  与各任务间的最远距离;  $\text{dist}_{ij}$  为用户  $u_i$  与任务  $s_j$  间的距离. 通过式(1)可得: 当用户距离任务越近, 用户与任务间的空间匹配度就越高; 当用户距离任务越远, 用户与任务间的空间匹配度就越低<sup>[23]</sup>; 当用户刚好位于任务所在位置时, 用户与任务之间完全匹配, 对应的空间匹配度为 1. 在用户前往任务感知区域的过程中, 随着与任务之间的距离越来越远, 用户与任务间的空间匹配度会逐渐升高, 用户想要执行该任务的意向就越来越大, 因此, 用户在移动的过程中会主动提高速度, 降低移动时间. 但是, 长距离的移动仍会损耗用户大量体力, 虽然用户的移动速度呈现上升趋势, 但是移动速率却逐渐下降. 图 2 显示了用户的移动速度随空间匹配度的变化趋势。

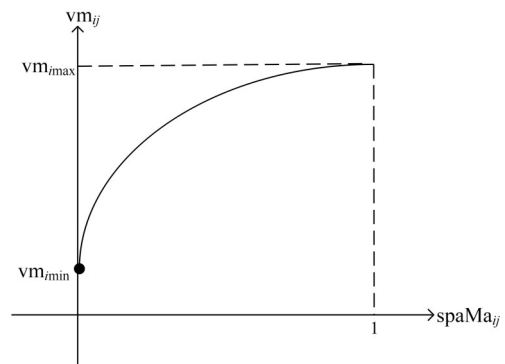


图2 用户移动速度的变化趋势

具体的移动速度变化函数为

$$\text{vm}_{ij} = \left( 1 + a \sqrt{\text{spaMa}_{ij}} \right) \times \text{vm}_{i \min} \quad (2)$$

其中,  $vm_{ij}$  为用户  $u_i$  前往任务  $s_j$  过程中的移动速度;  $vm_{i\min}$  为用户的基本移动速度;  $vm_{i\max} = (1+a) \times vm_{i\min}$  为用户能达到的最大移速,  $a \geq 1$  为空间匹配度系数.

## 2.2 技能匹配度

通过对用户所具备的技能信息以及任务所需要的感知内容进行分析可得, 技能种类与感知内容之间重合的部分越多, 用户对任务的完成情况越好, 所获取的数据质量也会更高. 同时, 用户对重合技能的熟练度决定了用户执行任务的速度, 越高的熟练度意味着用户对任务的执行力越强, 因此, 能够在短时间内获取更多高质量的感知数据. 所以, 用技能匹配度来表示用户和任务之间的适应程度. 根据用户与不同任务之间的技能匹配度, 依次把任务分配给最佳用户, 不但能减少用户执行任务所花费的时间, 还能有效利用用户的在线时间, 最大化完成任务的数量<sup>[24]</sup>. 技能信息与感知内容的匹配示意图如图3所示.

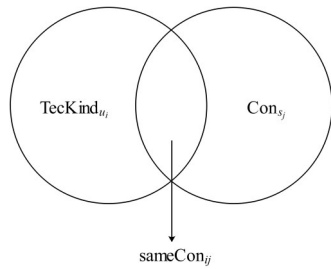


图3 技能信息与感知内容匹配示意图

图3中,  $sameCon_{ij} = TecKind_{u_i} \cap Con_{s_j}$  为技能重合集, 表示用户技能种类集和任务感知内容集重合的部分. 技能重合集中元素越多, 意味着技能种类集和感知内容集的重合度越高. 因此, 定义技能重合度为

$$TecCod_{ij} = \frac{|sameCon_{ij}|}{|Con_{s_j}|} \quad (3)$$

由于用户对任务的执行力受到技能重合度和重合技能熟练度两方面的影响, 同时考虑这两种因素, 技能匹配度为

$$TecMa_{ij} = \lambda_1 \times TecCod_{ij} + \lambda_2 \times \sum_{q=1}^w (\eta_q \times TecProf_{iq}) \quad (4)$$

其中,  $\lambda_1$  为技能重合度系数,  $\lambda_2$  为重合技能熟练度系数, 两者之间的关系为  $\lambda_1 + \lambda_2 = 1$ ;  $w = |sameCon_{ij}|$  为技能重合集的元素个数;  $\eta_q$  为执行感知任务过程中重合技能  $q$  的重要程度,  $\sum_{q=1}^w \eta_q = 1$ . 基于技能匹配度, 用户执行任务的速度为

$$ve_{ij} = (1 + b_1 \sqrt{TecMa_{ij}}) \times ve_{i\min} \quad (5)$$

其中,  $b_1 \geq 1$  为技能匹配度系数;  $ve_{i\min}$  为用户的基本执行速度.

技能匹配度越高表示用户和任务之间的适应情况越好, 因此, 用户能够感知到更高质量的数据, 具体的质量生成函数<sup>[25]</sup>表示为

$$g_{ij} = \ln [b_2 (M_{i\min} + TecMa_{ij}) + 1] \quad (6)$$

其中,  $0 \leq M_{i\min} < 1$  为用户感知到的基本数据质量;  $b_2 \geq 1$  为质量生成系数. 由此定义用户的感知数据质量为

$$M_{ij} = \frac{e^{g_{ij}} - e^{-g_{ij}}}{e^{g_{ij}} + e^{-g_{ij}}} \quad (7)$$

其中,  $M_{ij}$  为用户  $u_i$  在执行任务  $s_j$  过程中的感知数据质量. 如图4所示, 在质量生成系数固定时, 用户的感知数据质量随质量生成函数值的增加而增加.

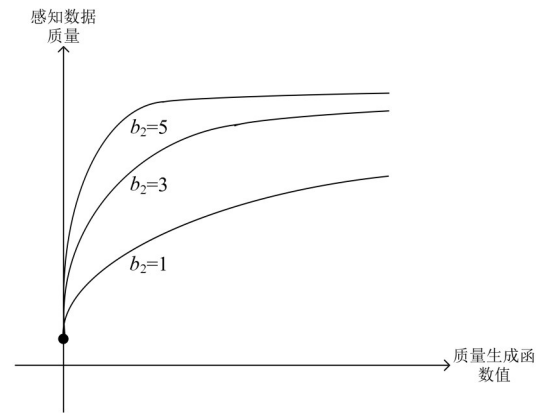


图4 用户感知数据质量与质量生成函数值的关系

## 2.3 技能更新机制

得益于人类特有的学习能力, 用户执行任务的过程也是技能不断提升的过程, 主要包括对已掌握技能地熟练应用和学习一种新技能两个方面. 因此, 设计一种基于用户学习能力的任务分配方案是十分有必要的.

在用户执行任务时, 已有的技能会被重复使用, 随着感知数据量的增多, 已有技能的熟练度也会获得相应提升<sup>[26]</sup>, 其更新公式如下:

$$TecProf_{iq'} = \frac{1}{\pi} \arctan \left( \beta \times (TecProf_{iq} + M_{ij}) \right) + TecProf_{iq} \quad (8)$$

其中,  $TecProf_{iq}$  为任务执行前用户对重合技能  $q$  的熟练度,  $\beta \geq 1$  为更新系数. 对于重合技能  $q$ , 当其熟练度达到 1 时, 表示该技能已被用户完全掌握, 并不再对其进行更新.

在实际的任务分配过程中, 用户可能执行感知内容与其技能种类部分重合的任务. 在此过程中, 用户不但能够更加熟练地应用已有技能, 还能学习到与该任务相关的其他技能, 不断完善自身的技能种类集, 具体

表示如下:

$$\text{TecKind}_{u_r} = \text{TecKind}_{u_i} \cup \text{Con}_{s_j} \quad (9)$$

其中,  $s_j \in T_i$  为用户任务集中待执行的任务;  $\text{TecKind}_{u_i}$  为任务  $s_j$  执行之前用户的技能种类集. 同时, 对于用户学习到的新技能, 其熟练度为  $\alpha (0 < \alpha < 1)$ .

### 2.4 最大化任务完成数量的任务分配问题

基于上述模型, 考虑到用户有限的工作时间, 在保证任务基本完成质量、降低成本<sup>[27]</sup>的前提下, 定义最大化任务完成数量的任务分配问题如下:

$$\begin{aligned} & \max \sum_{u_i \in U} |T_i| \quad (10) \\ \text{s.t.: } & M_{ij} \geq M_j, u_i \in U, s_j \in T_i \\ & \sum_{u_i \in U} (\text{exeC}_i + \text{movC}_i) \leq B \\ & \text{exeL}_i + \text{movL}_i \leq \text{Las}_i, u_i \in U \quad (11) \\ & T_0 \cap T_i = \emptyset \\ & T_i \cap T_j = \emptyset \end{aligned}$$

式(10)为优化问题的目标函数, 即最大化任务的完成数量. 式(11)的第一行表示用户对任务的感知数据质量不能低于其所需的基本数据质量. 式(11)的第二行表示系统中所有用户执行任务的总成本不得超过设定的预算, 其中:  $\text{exeC}_i = \sum_{s_j \in T_i} (c_1 \times \text{exeH}_{ij})$  为用户  $u_i$  的

执行成本,  $c_1$  为用户的单位执行成本,  $\text{exeH}_{ij} = \frac{\text{Adr}_j}{\text{ve}_{ij}}$  为用户  $u_i$  执行任务  $s_j$  花费的时间;  $\text{movC}_i = \sum_{s_j \in T_i} (c_2 \times \text{movH}_{ij})$  为用户  $u_i$  的移动成本,  $c_2$  为用户的单

位移动成本,  $\text{movH}_{ij} = \frac{\text{dist}_{ij}}{\text{vm}_{ij}}$  为用户  $u_i$  移动到任务  $s_j$  需要的时间. 式(11)的第三行表示用户  $u_i$  完成所有任务花费的总时间不得超过其在线时长, 其中:  $\text{exeL}_i = \sum_{s_j \in T_i} \text{exeH}_{ij}$  为用户  $u_i$  执行所有任务花费的总时间,  $\text{movL}_i = \sum_{s_j \in T_i} \text{movH}_{ij}$  为用户  $u_i$  移动到所有任务需要的总时间. 式(11)的第四行和第五行说明已分配的任务不会再分配给其他用户, 并且此任务将从待分配任务集中剔除.

**定理 1** 最大化任务完成数量的任务分配问题为 NP 难题

**证明** 在任务分配问题中, 当系统中只存在一个用户时, 优化目标将转换为求取该用户在有限资源约束条件下的最大任务完成数量. 考虑到旅行推销员问题 (Travelling Salesman Problem, TSP)<sup>[28]</sup> 中一个商品推销员需要在若干城市间移动并推销商品, 如何规划路线以实现其在最短路程内访问到所有城市是一个 NP

难题. 分析可得, 只存在一个用户的任务分配问题与该旅行推销员问题具有类似的情形, 而且最大化任务完成数量的任务分配问题通常存在多个用户, 相比之下会更加复杂. 因此, 最大化任务完成数量的任务分配问题为 NP 难题. 证毕

由于 NP 难题不存在能够在多项式时间内进行求解的算法. 因此, 结合所提任务分配问题的特点, 利用改进的鲸鱼优化算法以寻求问题的次优解.

## 3 用户选择策略

### 3.1 改进的鲸鱼优化算法

鲸鱼优化算法是受座头鲸捕食行为启发的一种新兴群智能优化算法<sup>[19]</sup>, 通过把问题的每个可行解视作不同位置的鲸鱼, 利用鲸鱼捕食过程中独特的位置更新机制达到对问题解的优化, 逐步得到最优解. 目前, 鲸鱼优化算法因其具有易于实现、控制参数少、鲁棒性强等优点在连续优化问题的求解方面得到了广泛应用<sup>[29]</sup>. 但由于其参数相对固定, 该算法的全局寻优能力弱, 易陷入局部搜索阶段, 出现“早熟”现象; 另外, 在鲸鱼进行位置更新时, 易受到种群中最优个体的影响, 从而降低了种群的多样性. 因此, 为了更好地用于最大化任务完成数量的任务分配问题, 需要对基本的鲸鱼优化算法进行改进以全面提高算法的搜索性能.

#### 3.1.1 引入非线性递减收敛因子

在基本鲸鱼优化算法中, 系数向量  $A$  的变动受参数  $a$  的影响. 随着  $a$  的线性递减, 由式(13)可知,  $A$  是区间  $[-a, a]$  上的一个随机向量,  $|A|$  的大小控制着算法在探索阶段和利用阶段之间进行切换. 在算法运行前期, 为了维护种群的多样性, 应尽可能让  $a$  取较大的值, 在提升算法全局搜索能力的同时加快算法的收敛<sup>[30]</sup>; 在算法运行后期, 收敛因子  $a$  应取较小的值, 扩大当前最优解对种群中所有个体的影响, 从而增强算法的局部寻优能力<sup>[31]</sup>. 但是, 原始的收敛因子  $a$  是按照固定速率下降, 在算法迭代过程中不能很好地适应种群情况, 一旦在探索阶段没有找到种群最优解, 则在利用阶段很容易陷入局部最优, 降低算法的寻优精度. 因此, 本文采用两种非线性递减收敛因子结合随机参数的策略来控制算法执行过程中不同阶段的切换, 具体的计算公式如下:

$$a = \begin{cases} 2 - \frac{3}{2} \sqrt{\exp\left(\frac{t}{T_{\max}}\right) - 1}, & \text{rand}_1 \leq P_a \\ 2\left(\frac{t}{T_{\max}}\right)^2 - \frac{4t}{T_{\max}} + 2, & \text{rand}_1 > P_a \end{cases} \quad (12)$$

$$A = 2ar - a \quad (13)$$

其中,  $t$  为当前迭代数;  $T_{\max}$  为迭代的最大次数;  $P_a$  代表收敛因子指数递减的概率,  $P_a \in [0, 1]$ ;  $\text{rand}_1$  为  $[0, 1]$  范

围内的随机数,  $\text{rand}_1$  的取值决定了收敛因子按不同的方式递减;  $r$  为  $[0, 1]$  范围内的随机向量. 设  $T_{\max} = 500$ , 则式(12)中两种曲线的变化趋势如图5所示.

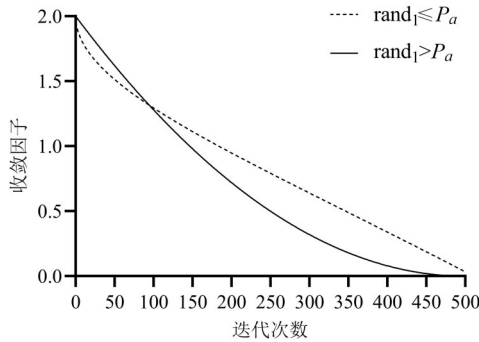


图5 两种非线性递减收敛因子的变化趋势

由图5可知,在算法迭代的过程中,两种非线性收敛因子的递减情况具有不同的针对性.当以指数形式递减时,因子 $a$ 的收敛速度会更加平缓,使算法能够在较长时间内进行全局搜索,降低了当前种群中最优个体对其他个体的影响,优化了下一代的种群质量.当收敛因子 $a$ 在多项式时间内递减时,算法会更快地进入到利用阶段,从而提高了局部收敛的速度,从整体上提升了算法的性能.为了使算法能够更好地平衡探索阶段和利用阶段,通过设置合适的 $P_a$ 值来控制收敛因子的递减方式,以实现在最佳时机进行不同阶段间的切换.

设 $P_a = 0.6$ ,代表在算法执行初期,种群中的个体选择全局搜索的可能性更大,维护了种群的多样性,增强了算法的全局寻优能力<sup>[32]</sup>,同时,在算法执行后期,也能随机地跳出局部最优,进一步扩大搜索范围,避免算法出现“早熟”现象.

### 3.1.2 最优局部抖动

在捕食期间,鲸鱼容易受到种群中最优个体的影响,并逐渐靠近最优个体.但是,在算法的整个迭代过程中,只有出现比当前最优个体适应度更高的个体时,才会替换当前的最优个体<sup>[33]</sup>.因此,最优个体一般保持不变,从而限制了算法的搜索范围.针对这种情况,将最优局部抖动思想融入鲸鱼优化算法中,在最优个体附近进行小范围的随机搜索,找到一个适应度值更高的个体,这样不仅有助于加快算法的全局收敛,还能避免算法过早地陷入局部最优<sup>[34]</sup>.

最优个体在邻域内产生局部抖动的公式为

$$X^{**}(t) = \begin{cases} X^*(t) + \text{rand}_2 \times X^*(t), & \text{rand}_2 < 0.5 \\ X^*(t), & \text{rand}_2 \geq 0.5 \end{cases} \quad (14)$$

其中,  $\text{rand}_2$  为  $[0, 1]$  区间上的随机数;  $X^*$  是当前鲸鱼群中最佳鲸鱼的位置向量;  $X^{**}(t)$  是优化后的新个体.

对于邻域内生成的新个体,采用贪心思想进行判

断,具体模型如下:

$$X^*(t) = \begin{cases} X^{**}(t), & f(X^{**}) < f(X^*) \\ X^*(t), & f(X^*) \leq f(X^{**}) \end{cases} \quad (15)$$

其中,  $f(x)$  为适应度函数. 若新个体的适应度值高于原最优个体,则进行最优个体的替换; 否则, 继续保留原最优个体, 不进行任何操作.

### 3.1.3 动态位置更新

当鲸鱼采取气幕攻击方式捕食猎物时, 会沿着特定形状的对数螺线来调整自身位置与最佳位置之间的距离<sup>[35]</sup>.  $b$  是决定对数螺线形状的参数, 由于  $b$  为常数, 鲸鱼每次更新位置都是按照相同的螺线进行移动, 这种单一的移动模式会缩小算法的搜索范围, 不利于算法全局收敛.

为了实现在迭代过程中鲸鱼能够依据外界情况动态地更新自身位置, 引入自适应螺旋参数, 将  $b$  设置成动态值, 并依赖于迭代次数. 随着迭代次数的增加,  $b$  值相应地进行变动, 随机地改变螺旋路径, 从而扩大鲸鱼的搜寻空间, 进一步提升算法在种群范围内的搜索能力. 动态螺旋位置更新公式如下:

$$\begin{cases} X(t+1) = D' \times e^{bl} \times \cos(2\pi l) + X^*(t) \\ b = 1 - \frac{5}{l} \cos \frac{\pi t}{T_{\max}} \end{cases} \quad (16)$$

$$D' = |X^*(t) - X(t)| \quad (17)$$

其中,  $X$  是鲸鱼的当前位置向量;  $D'$  表示当前鲸鱼与最佳鲸鱼之间的距离;  $l$  是  $[-1, 1]$  区间上的随机数.

参数  $b$  的设计是在螺旋线方程的基础上引入了迭代次数, 通过迭代次数的改变, 螺线形状会动态地进行调整. 在迭代初期, 螺线形状偏大, 鲸鱼能够在较大的范围内进行搜索, 尽可能多地遍历种群中的个体, 强化了算法全局寻优的能力; 在迭代后期, 螺线形状偏小, 鲸鱼会在最优个体的附近进行搜索, 从而加快了算法的收敛速度, 同时保证算法具有良好的寻优精度.

### 3.1.4 改进算法的具体步骤

基于改进鲸鱼优化算法的流程如图6所示, 完整步骤如下:

Step1 设置种群规模、最大迭代次数、 $P_a$  概率值等相关参数, 初始化鲸鱼种群;

Step2 计算种群中所有个体的适应度值, 并记录当前最优个体;

Step3 根据式(14)和式(15)对最优个体进行随机的局部抖动;

Step4 生成一个随机数  $p = \text{rand}(0, 1)$ , 若  $p \geq 0.5$ , 则根据式(16)更新个体位置, 若  $p < 0.5$ , 则执行Step5;

Step5 计算  $|A|$  值, 若  $|A| \geq 1$ , 则选择随机搜索方式更新个体位置, 若  $|A| < 1$ , 则通过包围锁定方式更新

个体位置;

Step6 计算新一代种群中所有个体的适应度值,更新最优个体,并更新参数  $\{a, A\}$ ;

Step7 判断迭代次数是否达到上限,若达到,则执行 Step8,否则,执行 Step3;

Step8 输出种群中的最优个体.

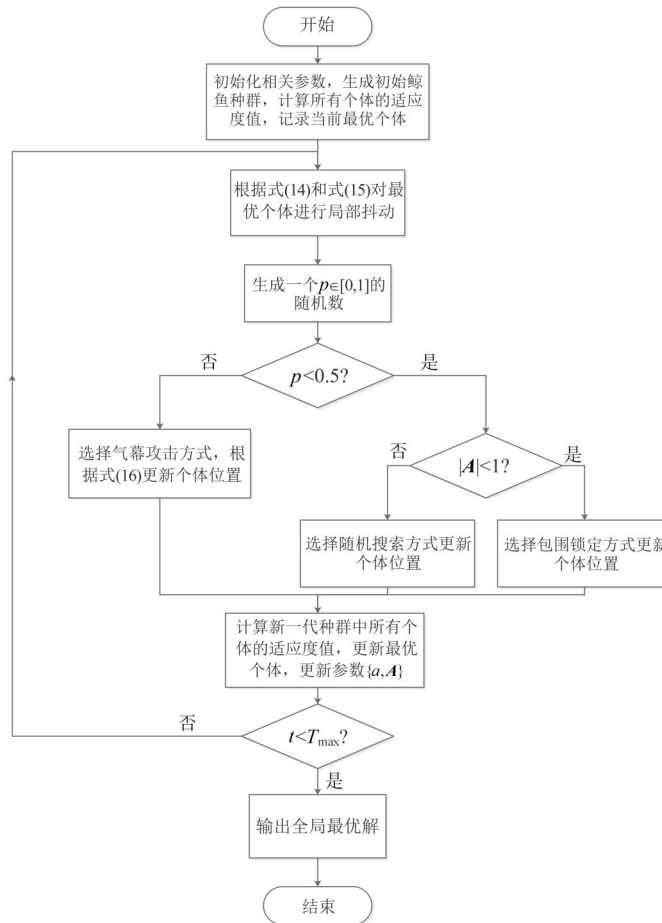


图6 基于改进鲸鱼优化算法的流程

### 3.2 基于优先级的用户选择策略

针对数量敏感任务分配问题的特点,结合改进的鲸鱼优化算法,从为任务选择合适的执行用户出发设计一种基于优先级的用户选择策略(Selecting Users of Priority-based Strategy, SUPS),主要包括任务序列优化和选择执行用户两个环节.

#### 3.2.1 任务序列优化

在任务序列优化这一环节中,首先需要对问题中的每个用户以及各个任务进行二进制编码,然后根据问题的规模随机产生一定数量的初始任务序列.现对系统中的  $m$  个用户以及  $n$  个任务分别编码为  $[w_1, w_2, \dots, w_m]$  和  $[W_1, W_2, \dots, W_n]$ .其中,  $w_i \in \{1, 2, \dots, m\}$  表示用户号;  $W_j \in \{1, 2, \dots, n\}$  表示任务号.在进行任务分配的过程中,会对本次分配结果中完成任务数量最多的结果所对应的任务序列进行最优记录,并用其来优化下一代产生的任务序列<sup>[36]</sup>.假设

系统中有 2 个用户和 8 个任务,对应位置对  $W_k - w_k$  表示任务  $W_k$  分配给用户  $w_k$ .当  $w_k = 0$  时,意味着任务序列相同位置的任务还未被分配.随机给出这 8 个任务的 1 个初始序列如图 7 所示,此时还未对任务进行分配.

任务序列	5	3	7	2	4	8	1	6
分配结果	0	0	0	0	0	0	0	0

图7 初始任务序列

#### 3.2.2 选择执行用户

在初始任务序列生成后,系统会根据任务顺序依次为每个任务选择执行用户<sup>[37]</sup>.首先,从任务序列中的第一个任务开始计算每个任务对应可分配用户的优先级,在所有优先级中选择最高优先级对应的用户执行该任务.如果用户剩余的可工作时长不足以执行任一任务,将不再给该用户分配任务.当发现所有用户执行

任务的总成本快要超出预算时,则会停止任务分配,分配过程结束.

在整个任务分配过程中,质量和成本往往是需要考虑的两个重要因素,而用户执行任务的总成本通常和用户耗费的时间有着直接关系<sup>[38]</sup>.基于此,给出用户优先级的计算公式:

$$\zeta_{ij} = \frac{M_{ij}}{\text{exeH}_{ij} + \text{movH}_{ij}} \quad (18)$$

对于任务  $s_j$ ,  $\zeta_{ij}$  越大,说明其对应用户的优先级越高.在为任务选择执行用户时,高优先级的用户不仅能够感知到更高质量的数据,还能产生较少的总成本.

结合用户的有限工作时长以及预算约束,依次为任务序列中的每个任务选择对应优先级最高的执行用户.在用户完成某一感知任务后,依据该任务的相关信息对用户所在位置以及技能进行更新,并参照最新的用户信息选择下一个任务的执行用户<sup>[39]</sup>.当某个用户的在线时长不足以继续执行任务时,该用户将不会再被选择.直到所有用户的总成本即将超出预算,会强制用户退出系统,从而完成部分任务的分配.基于图7的任务序列,举例说明选择执行用户的流程.首先,在任务分配之前,根据式(18)得到系统中2个用户的优先级,如图8所示.

用户优先级	2	1
-------	---	---

图8 第一个用户优先级序列

分析图8,对于待分配的第一个任务,优先级最高的是用户2.因此,选择用户2作为第一个任务的执行用户.对图7中的分配结果进行更新,此时的任务分配情况如图9所示.

任务序列	5	3	7	2	4	8	1	6
分配结果	2	0	0	0	0	0	0	0

图9 第一次用户选择结果

基于第一次的用户选择结果,对用户2的所在位置以及技能进行更新,重新计算系统中所有用户的优先级,如图10所示.

用户优先级	1	2
-------	---	---

图10 第二个用户优先级序列

根据最新的用户优先级序列,选择用户1作为第二个任务的执行用户,分配结果如图11所示.

按照上述流程依次为任务序列中的剩余任务选择

任务序列	5	3	7	2	4	8	1	6
分配结果	2	1	0	0	0	0	0	0

图11 第二次用户选择结果

对应优先级最高的执行用户.在选择完第四个任务的执行用户后,发现用户2的在线时长已不能够继续执行任务,因此,用户2将不再参与任务分配,此时的用户选择情况如图12所示.

任务序列	5	3	7	2	4	8	1	6
分配结果	2	1	1	2	0	0	0	0

图12 第四次用户选择结果

在停止向用户2分配任务后,系统中只存在1个用户,则由用户1继续按照任务序列依次执行剩余任务.当执行完任务6时,所有用户的累计成本即将超出预算,此时,强制所有用户退出系统,结束任务分配,完整的分配结果如图13所示.

任务序列	5	3	7	2	4	8	1	6
分配结果	2	1	1	2	1	1	0	0

图13 最终用户选择结果

数量敏感的任务分配流程如图14所示,完整步骤如下:

Step1 初始化相关参数,对系统中的用户以及任务进行二进制编码;

Step2 随机生成一定数量的初始任务序列,根据SUPS依次为任务序列中的任务选择执行用户;

Step3 参照用户选择结果,分别计算所有任务序列对应的任务完成数,并记录最优解,即当前任务完成数最大的序列及其对应的分配结果;

Step4 利用最优解,结合改进的鲸鱼优化算法对所有任务序列进行优化;

Step5 再次根据SUPS依次为任务优化序列中的任务选择执行用户;

Step6 分别计算Step5中每种分配结果对应的任务完成数,对最优解进行更新;

Step7 判断迭代次数是否达到上限,若达到,则执行Step8,否则,执行Step4;

Step8 输出最优任务序列及其对应的分配结果.

### 3.2.3 复杂度分析

SUPS主要包含任务序列优化和选择执行用户两个

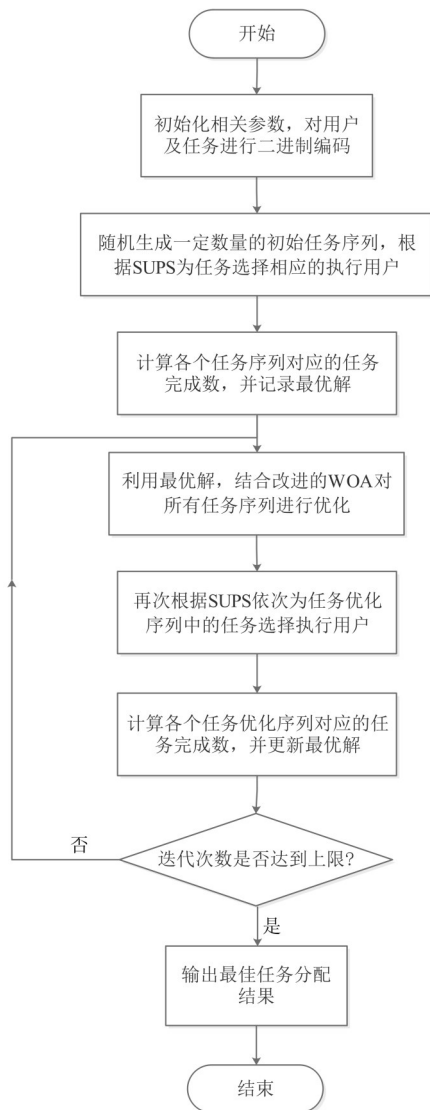


图 14 数量敏感的任务分配流程

阶段. 其中, 任务序列优化阶段的时间复杂度为  $O(ion)$ ,  $i$  为迭代次数,  $o$  为初始任务序列数,  $n$  为任务数; 选择执行用户阶段的时间复杂度为  $O(iomn)$ ,  $m$  为用户数. 由此可得, 数量敏感的任务分配过程总体的时间复杂度为  $O(ion) + O(iomn) = O(ion(1 + m))$ , 与迭代次数和初始任务序列数成正比关系, 在可接受的范围内.

#### 4 仿真实验与分析

本文在基本鲸鱼优化算法的基础上引入了非线性递减收敛因子、最优局部抖动以及动态位置更新三种改进操作, 将改进后的鲸鱼优化算法命名为 IWOA.

对照算法选用文献[19]中的 WOA、文献[20]中的 DCSA 以及文献[21]中的 IGWO. DCSA 是基于离散布谷鸟的搜索算法, 其根据布谷鸟寻找寄宿鸟巢的方式设计对应的全局搜索过程以及局部搜索过程以求得问题

的最优解, 而 IGWO 是一种改进的灰狼优化算法, 其特点为通过一个初始化算法生成初始种群, 并在每次迭代中选出最好的三个候选解, 由它们带领整个种群朝着最优解的方向移动. 这两种算法都为近年来很受欢迎的群体智能优化算法, 已有研究表明其性能优于粒子群算法和遗传算法. 因此, 将 IWOA 与上述算法进行对比研究, 可以更好地说明本文所提出的算法在求解任务分配问题时的可行性.

首先, 为了验证 IWOA 的搜索性能, 分别将 WOA, DCSA 以及 IGWO 运用到本文所提出的任务分配策略, 并与 IWOA 在不同环境下进行性能对比. 其次, 设计一种基于 IWOA 从为用户分配任务角度出发的任务分配策略 (Assigning Tasks of Users-based Strategy, ATUS), 比较说明 SUPS 的有效性. 最后, 利用 SUPS 分别在考虑用户学习能力与不考虑用户学习能力的环境下进行仿真实验, 突出说明在任务分配过程中考虑用户学习能力的必要性. 运行实验的环境: Windows10 64bit; PyCharm 2016; 处理器 Intel i5-7200U; 主频 2.50 GHz; 内存 8.00 GB.

##### 4.1 参数设置

所有仿真实验的数据来源于 eil51 数据集, 相关参数设置如表 1 所示.

表 1 相关参数设置

参数	取值
$P_a$	[0, 1]
用户数	6
用户所在位置	[0-30, 0-30]
用户可工作时长/min	[200, 500]
用户技能熟练度	[0, 1]
用户基本移动速度/(m/s)	[1.1, 1.5]
用户基本执行速度	[1, 2]
用户基本感知数据质量	[0, 1]
任务数	20
任务所需数据量	[40, 70]
任务所在位置	[0-30, 0-30]
单位移动成本	0.7
单位执行成本	0.2
所有系数	1

##### 4.2 对比实验分析

分别将 IWOA, WOA, DCSA 以及 IGWO 应用到任务分配问题, 并在相同参数设置下将每个算法运行 50 次, 取 50 次结果的平均值作为最终结果. 仿真实验对比 IWOA, WOA, DCSA, IGWO 在不同预算或不同用户数环境下运行时能达到的最大任务完成数量, 其结果如图 15(a) 和图 15(b) 所示.

从图 15 可以看出, 各算法对应的最大任务完成数

量随预算或用户数的增加而增加,它们之间的关系呈正相关.其中,针对不同的预算或不同的用户数,IWOA得到的最大任务完成数量始终位于第一位,而IGWO和DCSA得到的最大任务完成数量则位居其后,且在预算相同时,两者之间相差不大.此外,观察图15可知,在预算较低或用户数较少时,IWOA仍能得到较高的任务完成数量,这表明IWOA在任务完成数量方面所表现的性能优于其他算法.

图16显示了IWOA,WOA,DCSA,IGWO在不同预

算或不同用户数环境下运行得到的任务分配结果所对应的总成本.可以看出,针对不同的预算或不同的用户数,四种算法得到的总成本均低于预算,并且在预算足够时,各个算法都将尽可能多地把任务分配给用户.在预算增加的同时,各个算法对应的总成本也会随之增加,但仍旧不超过预算.此外,由于IWOA得到的分配结果中任务完成数量最多,因此,相比于其他算法,IWOA具有更高的总成本,但因其在预算范围内,所以可以接受.

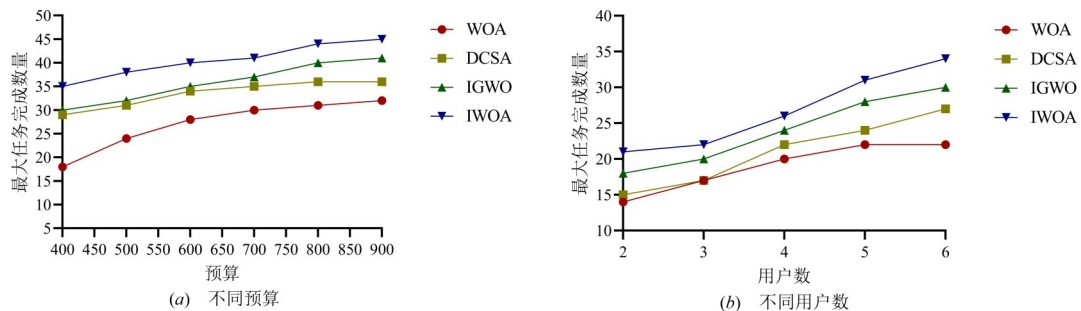


图15 最大任务完成数量

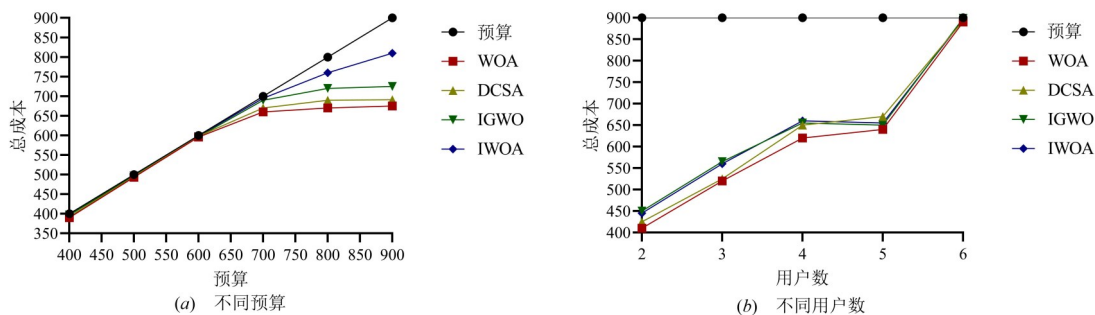


图16 总成本

IWOA,WOA,DCSA,IGWO在不同预算或不同用户数环境下运行获得的数据质量如图17所示.观察可得,由于预算或用户数的增加,每种算法对应的最大任务完成数量也随之增加,从而提升了感知任务所获得的整体数据质量.在预算相同的情况下,IWOA对应的数据质量最高,IGWO和DCSA对应的数据质量依次递减,且都高于WOA对应的数据质量.在用户数相同的情况下,IWOA和IGWO对应的数据质量之间差别不大,但和其余两种算法相比仍为最优.由此可见,将改进算法应用于任务分配问题所获得的数据质量具有显著提高.

为了说明SUPS的有效性,从为用户分配任务的角度出发设计一种ATUS来求解该任务分配问题.图18~图20显示了两种策略在不同预算或不同用户数环境下最大任务完成数量、总成本以及数据质量的变化情况.

观察可得,两种策略在不同环境下的最大任务完成数量、总成本以及数据质量均随预算或用户数的增加而增加.其中,SUPS的最大任务完成数量以及数据质量始终高于ATUS,即使在低预算条件下,SUPS也能获得较高的任务完成数量.此外,两种策略得到的任务分配结果所对应的总成本均低于预算,相比之下,SUPS需要的总成本更低,因此,SUPS在任务分配方面会比ATUS更有效.

为了体现考虑用户学习能力在任务分配过程中的必要性,结合SUPS,分别在考虑用户学习能力与不考虑用户学习能力的前提下执行IWOA,得到的最大任务完成数量、总成本以及数据质量在不同预算或不同用户数环境下的取值情况如图21~图23所示.

可以看出,在任务分配问题中,无论是否考虑用户学习能力,其需要的总成本均不超过预算.随着预算或用户数的增加,这两种情况对应的最大任务完成数

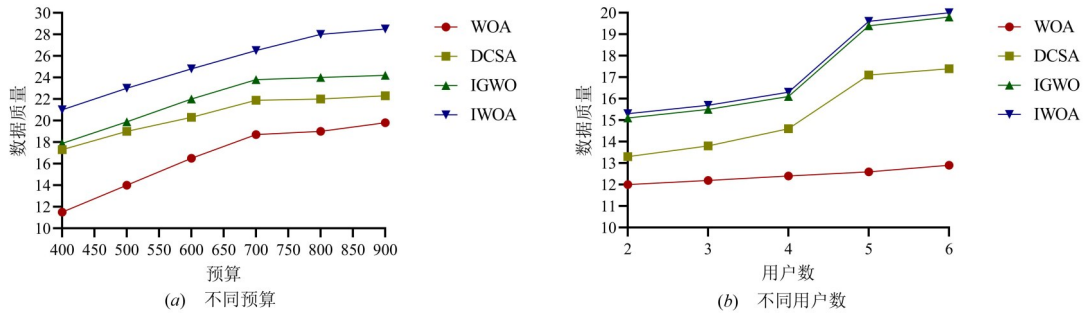


图 17 数据质量

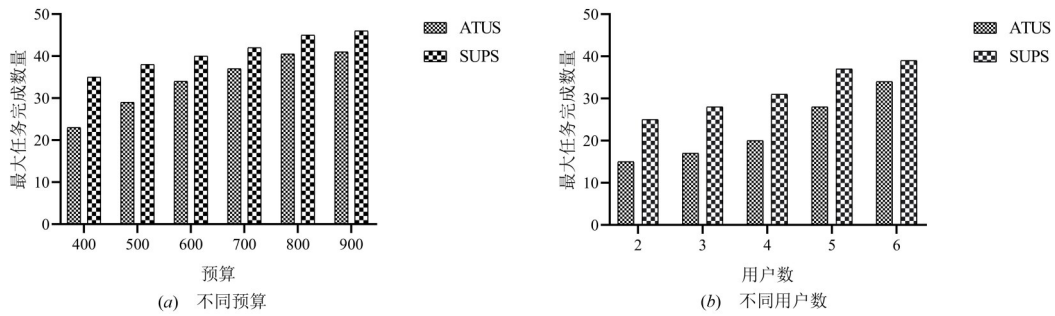


图 18 最大任务完成数量

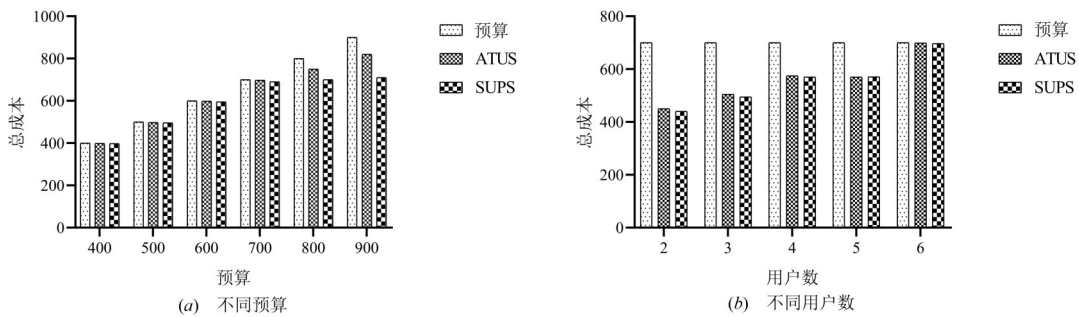


图 19 总成本

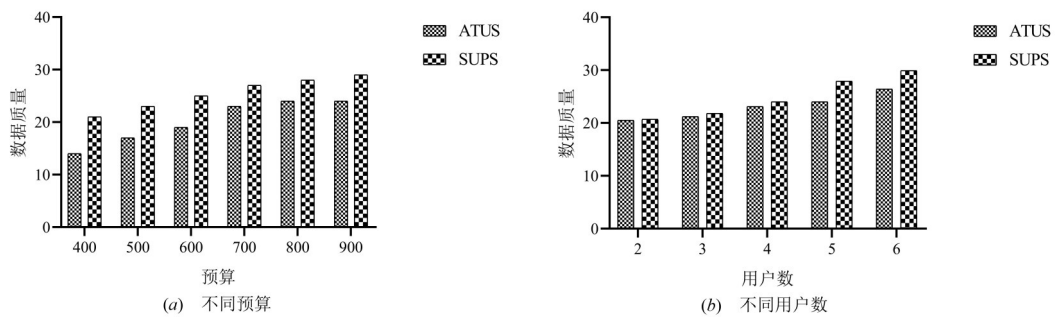


图 20 数据质量

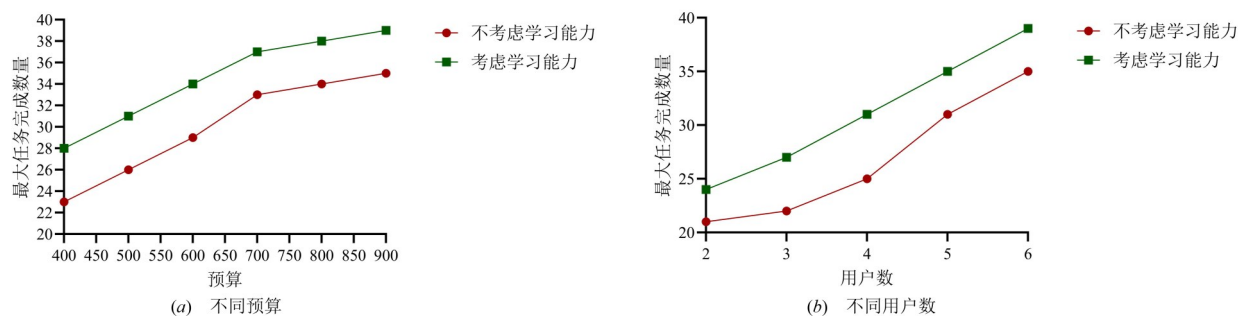


图21 最大任务完成数量

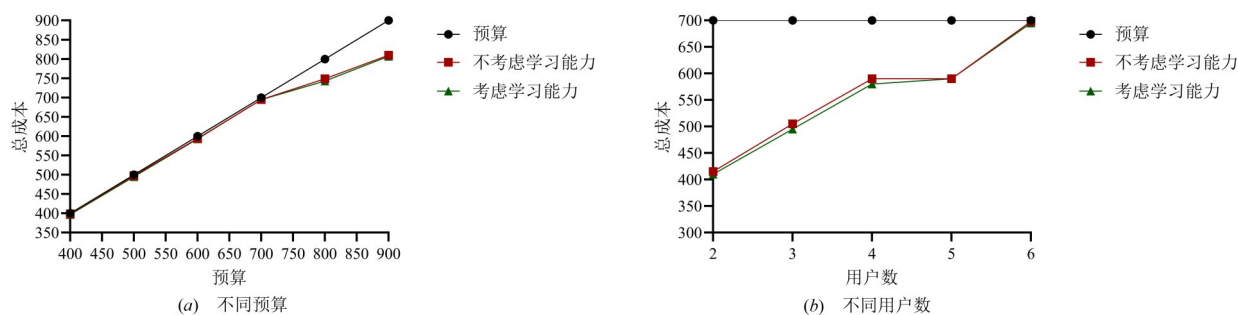


图22 总成本

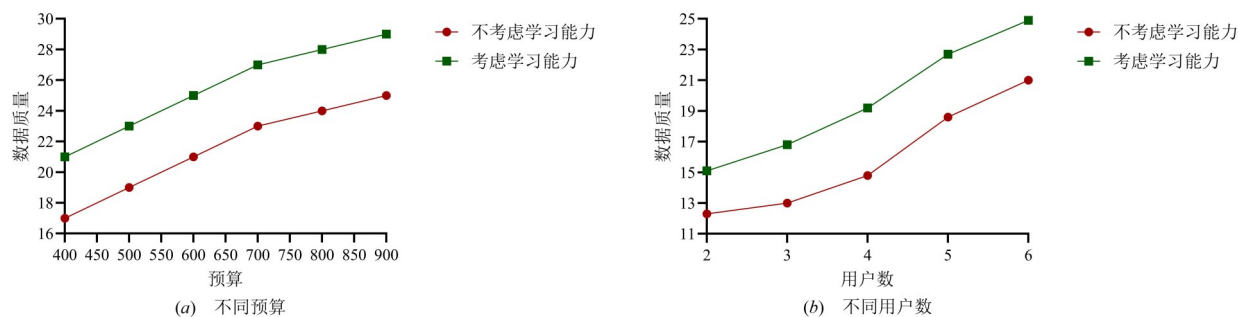


图23 数据质量

量以及数据质量均随之增加。然而,在整个变化过程中,考虑用户学习能力相比不考虑用户学习能力将获得更多的任务完成数量以及更高的数据质量。因此,在任务分配过程中,考虑用户学习能力是非常关键的。

## 5 总结

针对数量敏感的任务分配问题,本文提出了一种结合非线性递减收敛因子、最优局部抖动、动态位置更新三种操作的改进鲸鱼优化算法。考虑到用户与任务间的不同适应程度对感知数据质量产生的影响,引入了空间匹配度与技能匹配度的概念。由于用户在执行任务的过程中具备一定的学习能力,对用户技能的提升方式进行建模,并提出相应的技能更新机制。在此

基础上,定义了最大化任务完成数量的任务分配问题,给出了相应的优化模型。为了使问题能够更好地解决,将原本分配任务给用户的求解方式转换为任务选择最佳的执行用户,提出一种基于优先级的用户选择策略。最后,把改进的鲸鱼优化算法应用到所提出问题,通过仿真实验得出,改进算法比其他优化算法具有更高的寻优精度,并且实用性能较好。由于本文所提出的改进算法只适用于静态场景,如何根据实际情况设计动态的任务分配算法是接下来研究工作的重心。

## 参考文献

- [1] 李卓,徐哲,陈昕,等. 面向移动群智感知的位置相关在线多任务分配算法[J]. 计算机科学, 2019, 46(6):

- 102-106.
- LI Z, XU Z, CHEN X, et al. Location-related online multi-task assignment algorithm for mobile crowd sensing[J]. *Computer Science*, 2019, 46(6): 102-106. (in Chinese)
- [2] 杜扬, 黄河, 孙玉娥, 等. 地理位置相关移动感知系统任务分配问题研究[J]. *计算机研究与发展*, 2014, 51(11): 2374-2381.
- DU Y, HUANG H, SUN Y, et al. A location-based task assignment mechanism for mobile phone sensing[J]. *Journal of Computer Research and Development*, 2014, 51(11): 2374-2381. (in Chinese)
- [3] ESTRADA R, MIZOUNI R, OTROK H, et al. A crowdsensing framework for allocation of time-constrained and location-based tasks[J]. *IEEE Transactions on Services Computing*, 2020, 13(5): 769-785.
- [4] GONG W, ZHANG B X, LI C. Location-based online task assignment and path planning for mobile crowdsensing[J]. *IEEE Transactions on Vehicular Technology*, 2019, 68(2): 1772-1783.
- [5] ZHAO B X, WANG Y J, LI Y S, et al. Task allocation model based on worker friend relationship for mobile crowdsourcing[J]. *Sensors(Basel, Switzerland)*, 2019, 19(4): 921.
- [6] TU J Z, YU G X, WANG J, et al. CrowdWT: Crowdsourcing via joint modeling of workers and tasks[J]. *ACM Transactions on Knowledge Discovery from Data*, 2021, 15(1): 1-24.
- [7] ZHU W P, GUO W Z, YU Z Y, et al. Multitask allocation to heterogeneous participants in mobile crowd sensing[J]. *Wireless Communications and Mobile Computing*, 2018, 2018: 7218061.
- [8] ALSAYASNEH M, AMER-YAHIA S, GAUSSIER E, et al. Personalized and diverse task composition in crowdsourcing[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2018, 30(1): 128-141.
- [9] GAO X F, HUANG H W, LIU C L, et al. Quality inference based task assignment in mobile crowdsensing[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2021, 33(10): 3410-3423.
- [10] PAN Q X, PAN T W, DONG H B, et al. An online task assignment based on quality constraint for spatio-temporal crowdsourcing[J]. *IEEE Access*, 2019, 7: 170292-170303.
- [11] TU J Y, CHENG P, CHEN L. Quality-assured synchronized task assignment in crowdsourcing[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2021, 33(3): 1156-1168.
- [12] SONG S W, LIU Z D, LI Z J, et al. Coverage-oriented task assignment for mobile crowdsensing[J]. *IEEE Internet of Things Journal*, 2020, 7(8): 7407-7418.
- [13] CHEN J Y, YANG J S. Maximizing coverage quality with budget constrained in mobile crowd-sensing network for environmental monitoring applications[J]. *Sensors(Basel, Switzerland)*, 2019, 19(10): 2399.
- [14] EL-ASHMAWI W H, ALI A F. A modified salp swarm algorithm for task assignment problem[J]. *Applied Soft Computing*, 2020, 94: 106445.
- [15] HASSAN U UL, CURRY E. Efficient task assignment for spatial crowdsourcing: A combinatorial fractional optimization approach with semi-bandit learning[J]. *Expert Systems With Applications*, 2016, 58: 36-56.
- [16] WU P K, NGAI E W T, WU Y Y. Toward a real-time and budget-aware task package allocation in spatial crowdsourcing[J]. *Decision Support Systems*, 2018, 110: 107-117.
- [17] MIAO C Y, YU H, SHEN Z Q, et al. Balancing quality and budget considerations in mobile crowdsourcing[J]. *Decision Support Systems*, 2016, 90: 56-64.
- [18] LIU W B, YANG Y J, WANG E, et al. Prediction based user selection in time-sensitive mobile crowdsensing[C]// 2017 14th Annual IEEE International Conference on Sensing, Communication, and Networking. San Diego: IEEE, 2017: 1-9.
- [19] MIRJALILI S, LEWIS A. The whale optimization algorithm[J]. *Advances in Engineering Software*, 2016, 95: 51-67.
- [20] 杨正清, 周朝荣, 袁姝. 移动群智感知系统中基于离散布谷鸟搜索算法的任务分配[J]. *计算机应用*, 2019, 39(9): 2778-2783.
- YANG Z Q, ZHOU Z R, YUAN S. Task assignment based on discrete cuckoo search algorithm in mobile crowd sensing system[J]. *Journal of Computer Applications*, 2019, 39(9): 2778-2783. (in Chinese)
- [21] 姚远远, 叶春明. 求解作业车间调度问题的改进混合灰狼优化算法[J]. *计算机应用研究*, 2018, 35(5): 1310-1314.
- YAO Y Y, YE C M. Solving Job-Shop scheduling problem using improved hybrid grey wolf optimizer[J]. *Application Research of Computers*, 2018, 35(5): 1310-1314. (in Chinese)
- [22] 范泽军, 沈立炜, 彭鑫, 等. 基于约束的空间众包多阶段任务分配[J]. *计算机学报*, 2019, 42(12): 2722-2741.
- FAN Z J, SHEN L W, PENG X, et al. Multi stage task allocation on constrained spatial crowdsourcing[J]. *Chinese Journal of Computers*, 2019, 42(12): 2722-2741. (in Chinese)
- [23] JIANG N, XU D, ZHOU J, et al. Toward optimal partici-

- pant decisions with voting-based incentive model for crowd sensing[J]. *Information Sciences*, 2020, 512: 1-17.
- [24] WANG T X, WEI X L, TANG C G, et al. Efficient multi-tasks scheduling algorithm in mobile cloud computing with time constraints[J]. *Peer-to-Peer Networking and Applications*, 2018, 11(4): 793-807.
- [25] LADUMOR D P, TRIVEDI I N, JANGIR P, et al. A whale optimization algorithm approach for unit commitment problem solution[C]//*Proceedings of National Conference Advancement in Electrical & Power Electronics Engineering(AEPEE 2016)*. Morbi: AEPEE, 2016: DOI: 10.13140/RG.2.1.1290.2003.
- [26] WANG W D, GAO H, LIU C H, et al. Credible and energy-aware participant selection with limited task budget for mobile crowd sensing[J]. *Ad Hoc Networks*, 2016, 43: 56-70.
- [27] 郭兴海, 计明军, 温都苏, 等. “最后一公里”配送的分布式多无人机的任务分配和路径规划[J]. *系统工程理论与实践*, 2021, 41(4): 946-961.  
GUO X H, JI M J, WEN D S, et al. Task assignment and path planning for distributed multiple unmanned aerial vehicles in the “last Mile” [J]. *Systems Engineering-Theory & Practice*, 2021, 41(4): 946-961. (in Chinese)
- [28] XIE X F, LIU J M. Multiagent optimization system for solving the traveling salesman problem(TSP) [J]. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 2009, 39(2): 489-502.
- [29] 蔡雨岑, 杜鹏楨. 基于平衡鲸鱼优化算法的无人车路径规划[J]. *控制与决策*, 2021, 36(11): 2647-2655.  
CAI Y C, DU P Z. Path planning of unmanned ground vehicle based on balanced whale optimization algorithm[J]. *Control and Decision*, 2021, 36(11): 2647-2655. (in Chinese)
- [30] GAO H, SHI Y J, PUN C M, et al. An improved artificial bee colony algorithm with its application[J]. *IEEE Transactions on Industrial Informatics*, 2019, 15(4): 1853-1865.
- [31] CHI M W. An improved wolf pack algorithm[C]//*AI-IPCC' 19: Proceedings of the International Conference on Artificial Intelligence, Information Processing and Cloud Computing*. Sanya: ACM, 2019: 1-5.
- [32] IBRAHIM R A, EWEEES A A, OLIVA D, et al. Improved salp swarm algorithm based on particle swarm optimization for feature selection[J]. *Journal of Ambient Intelligence and Humanized Computing*, 2019, 10(8): 3155-3169.
- [33] LI Z Y. An improved genetic algorithm for solving packing problem[C]//*Proceedings of the 2016 4th International Conference on Machinery, Materials and Computing Technology & Advances in Engineering Research*. Paris: Atlantis Press, 2016: 1815-1820.
- [34] WANG F, ZHANG H, LI K S, et al. A hybrid particle swarm optimization algorithm using adaptive learning strategy[J]. *Information Sciences*, 2018, 436/437: 162-177.
- [35] GHAREHCHOPOGH F S, GHOLIZADEH H. A comprehensive survey: Whale optimization algorithm and its applications[J]. *Swarm and Evolutionary Computation*, 2019, 48: 1-24.
- [36] TONG Y X, WANG L B, ZHOU Z M, et al. Flexible online task assignment in real-time spatial data[J]. *Proceedings of the VLDB Endowment*, 2017, 10(11): 1334-1345.
- [37] 陈思光, 陈佳民, 赵传信. 基于深度强化学习的云边协同计算迁移研究[J]. *电子学报*, 2021, 49(1): 157-166.  
CHEN S G, CHEN J M, ZHAO C X. Deep reinforcement learning based cloud-edge collaborative computation offloading mechanism[J]. *Acta Electronica Sinica*, 2021, 49(1): 157-166. (in Chinese)
- [38] 李博扬, 成雨蓉, 王国仁, 等. 新型时空众包平台中的在线三维稳定匹配问题[J]. *软件学报*, 2020, 31(12): 3836-3851.  
LI B Y, CHENG Y R, WANG G R, et al. 3D-online stable matching problem for new spatial crowdsourcing platforms[J]. *Journal of Software*, 2020, 31(12): 3836-3851. (in Chinese)
- [39] 王乐乐, 眭泽智, 蒲志强, 等. 一种改进RRT的多机器人编队路径规划算法[J]. *电子学报*, 2020, 48(11): 2138-2145.  
WANG L L, SUI Z Z, PU Z Q, et al. An improved RRT algorithm for multi-robot formation path planning[J]. *Acta Electronica Sinica*, 2020, 48(11): 2138-2145. (in Chinese)

#### 作者简介



蒋伟进 男, 1964年出生, 湖南桃江人. 博士, 二级教授. 研究方向为云计算安全、边缘计算、群智感知、社会计算等.  
E-mail: jwjndt@163.com



张婉清(通讯作者) 女, 1997年出生, 河南洛阳人. 硕士研究生. 研究方向为群智感知、移动边缘计算.  
E-mail: 727149424@qq.com