

支持工人能力模糊度量和角色协同的软件众包任务分配*

马 华, 陈跃鹏, 黄卓轩, 唐文胜, 娄小平
(湖南师范大学 信息科学与工程学院, 湖南 长沙 410081)

摘要: 现有众包任务分配研究缺乏考虑对工人能力不确定性的度量,也未从众包平台的角度实现任务和工人间多对多模式下的协同分配。由此,提出支持工人能力模糊度量和角色协同的软件众包任务分配方法。该方法结合工人的历史表现和任务的需求期望,以模糊区间数评估工人的多属性能力匹配度,使用模糊层次分析法计算工人的综合胜任能力。引入基于角色协同理论,将多对多型任务分配建模为涉及一组任务与相关工人的组合优化问题,综合考虑任务权重、工人数量、潜在冲突等约束,以提高任务分配的效率和成功率,并给出一种基于 CPLEX 的问题求解方法。仿真实验结果表明,在确保获得全局任务集最佳完成质量的前提下,该方法能高效准确实现多对多模式下的任务分配。

关键词: 软件众包;任务分配;工人能力度量;基于角色协同

中图分类号: TP311 **文献标志码:** A **文章编号:** 1001-2486(2022)05-124-10

Software crowdsourcing tasks assignment supporting fuzzy measurement of workers' qualification and role collaboration

MA Hua, CHEN Yuepeng, HUANG Zhuoxuan, TANG Wensheng, LOU Xiaoping

(College of Information Science and Engineering, Hunan Normal University, Changsha 410081, China)

Abstract: The existing researches on crowdsourcing task assignment don't involve in measuring the uncertainty of workers' qualification, and don't achieve the collaborative assignment in many-to-many mode between tasks and workers from the angle of crowdsourcing platform. Thus, an assignment approach of software crowdsourcing tasks was proposed supporting the fuzzy measurement of workers' qualification and role collaboration. Integrating the past performance of workers and the expectation of tasks' demands, this approach employed fuzzy interval numbers to evaluate the multiple attributes ability matching degree, and aggregated the comprehensive qualification via the fuzzy analytic hierarchy process method. By introducing the role-based collaboration theory, the many-to-many software crowdsourcing tasks assignment was formulated as a combinatorial optimization problem related to a task set and a worker group, and the constraints, including the tasks' weights, quantity of workers and potential conflicts, were used to enhance the efficiency and success rate of tasks assignment. A solution based on the CPLEX package was presented to solve the problem. Simulation experiments show that this method can efficiently and accurately realize the collaborative allocation of crowdsourcing tasks under the premise of ensuring the best completion quality of global tasks.

Keywords: software crowdsourcing; tasks assignment; measurement of workers' qualification; role-based collaboration

近年来,以开源众包、任务中国、亚马逊 Mechanical Turk 和 TopCoder 等为代表的软件众包平台发展迅速。依托它们,传统的封闭式、高成本、低效率的软件开发过程逐渐呈现出开放式、高性价比、高效率的团体间多人协作的新范式^[1]。随着大量众包任务被不断发布,越来越多的软件技术人员以众包工人的角色加入软件众包的协同开发过程。综合考虑工人的历史表现和任务的属性特征以准确度量众包工人的综合能力,实现众包任务和工人间多对多模式下的分配优化,对于

确保众包平台获得最佳的整体任务完成质量,具有重要的理论价值和现实意义。

软件众包平台通常采用在线竞争机制挑选优秀的众包工人来承接任务。而工人工作能力的准确度量是众包任务分配前首先需考虑的关键因素,它直接影响到任务的最终完成质量^[2]。现有研究尚缺乏对工人能力的全面分析,未综合考虑众包任务与工人能力的匹配程度对分配结果的影响^[3],现有的基于单一值的描述机制也无法客观反映工人实际能力的不确定性。同时,现有研究

* 收稿日期:2020-11-05

基金项目: 国家自然科学基金资助项目(62077014);湖南省自然科学基金资助项目(2020JJ4440);湖南省教育厅科学研究项目优秀青年资助项目(18B037)

作者简介: 马华(1979—),男,湖南宁远人,教授,博士,硕士生导师,E-mail:huama@hunnu.edu.cn;
陈跃鹏(通信作者),男,硕士研究生,E-mail:chenyuepeng@smail.hunnu.edu.cn

主要采用基于匹配和基于规划的任务分配模型,关注的是面向单个工人的一对一型和一对多型分配问题,尚未涉及面向众包平台的多对多型分配问题^[3-4]。

多对多型分配问题从软件众包平台的角度考虑如何将一组相关任务分配给具有协作关系的工人,以获得最佳的全局任务完成质量。由此,多对多型任务分配本质上可视为一个典型的基于角色协同^[5](role-based collaboration, RBC)问题。作为RBC的核心模型,E-CARGO^[6-7]使用一组核心概念建立组角色分配相关的数学模型,之前基于E-CARGO的任务分配优化研究^[8]可为解决多对多型众包任务分配提供重要的求解范式。

基于此,本文提出支持工人能力模糊度和角色协同的软件众包任务分配方法。该方法通过结合工人的历史表现和任务的需求期望,采用模糊区间数^[9]评估工人的多属性能力匹配度,以模糊层次分析法计算工人的综合胜任能力,并引入RBC理论和E-CARGO模型对多对多型任务分配问题进行系统的形式化建模和约束分析,给出了一种有效的问题求解方法,以确保获得最优的整体任务完成质量。

1 相关工作

1.1 众包工人能力的综合度量

众包工人能力度量的传统策略包括基于黄金标准数据的策略^[10]、多阶段动态众包质量评估策略^[11]、基于熵的众包质量评估策略^[12]和基于投票的一次性策略^[13]等。并且,基于历史数据对工人进行信誉评估,有助于全面了解工人的综合能力。芮兰兰等^[14]引入惩罚激励因子来处罚恶意工人和激励理性工人提高任务完成质量。借鉴滴滴出行软件中司机服务的星级评价,冉家敏等^[15]将空间众包中工人的历史等级评价转化为服务质量评价,综合当前评价和历史评价来计算工人得分。针对众包全景拼接系统特点,李沁雅等^[16]通过量化街景图像对全景地图构建的贡献度来确定图像质量,以此评估工人能力,并将其与工人酬劳相结合来激励参与者。

现有研究主要使用单一的确定值描述工人的能力。但是,软件技术人员在实施不同项目时,可能受到物质或精神激励、情绪、知识储备和技术成熟度等诸多影响,工作表现并不稳定。从而,在度量工人的综合能力时,不能忽视其具有的不确定性本质。

1.2 众包任务的分配方法

众包模型主要可分为两类:①基于社会网络的复杂任务群智协同众包模型^[4]。它不要求发包者自己分解一个复杂任务,而是由接收任务的工人通过组织其合作者来一起完成任务。可免除发包者的任务分解负担,但增加了模型相关的计算量和对工人社会属性深度挖掘的复杂性,实际应用时仍有诸多问题需解决。②基于任务分解的复杂任务众包模型^[17]。它要求发包者具备良好的任务分解能力,也要求工人提交的结果具有良好的兼容性。综合考虑可行性和时效性,实际众包应用平台大多采用该模型。

基于任务分解的复杂任务众包模型又可细分为两类:①基于匹配的任务分配模型^[3]。它使用二分图匹配,根据任务与工人的时间、空间、状态等特征,并结合工人的信誉值或可信度等,为每名工人分配一项任务,即一对一型任务分配。Mechanical Turk、猪八戒网等传统众包平台以及滴滴出行、美团众包等空间众包平台均属此类。②基于规划的任务分配模型^[3]。它类似于旅行商问题模型,研究众包平台如何为一个众包工人分配合理的任务执行顺序和方案,本质上是针对一对多型的任务分配问题,即为一个工人分配多个任务以达到最优的工作组合。考虑到工人的最晚工作时间约束,结合树分解和深度优先搜索方法^[18]可获得任务分配最优方案。

随着越来越多的任务被同时发布,众包平台可能存在大量具有相似特点和需求的相关任务。此时的任务分配问题呈现出“多对多”特征,即众包平台如何将多个相关任务分配给合适的多个工人,这是一个NP难问题。不同于传统的一对一或一对多视角,它需要从众包平台整体的角度来考虑如何实现多个任务与工人间的组合优化^[3],以帮助众包平台获得全局最佳的任务完成质量。

2 问题描述

假定软件众包平台中某一版块新发布了一个软件开发项目,分解该项目后得到 n 个待分配任务,以 $R = \{r_0, r_1, \dots, r_{n-1}\}$ 表示任务集(task set),假设在报酬激励下有 m 个候选工人申请参加该项目,以 $A = \{a_0, a_1, \dots, a_{m-1}\}$ 表示工人组(worker group)。则该多对多型软件众包任务分配问题的特点包括:

1) 每个 $r_j(0 \leq j < n)$ 需要1个或多个 a_i 参与完成。例如,项目发包人希望招募多名 a_i 同步完成子系统开发任务或相应的黑盒测试任务。每个

任务相当于一个角色(或岗位),它需要多名工人合作承担该角色(或岗位)相应的职责。

2) 不失一般性,假定每个 $a_i (0 \leq i < m)$ 在同一个项目中只可申请完成一个 r_j 。如果允许 a_i 在任务不冲突、精力足够的前提下同时申请多个 r_j ,只需在 A 中复制多个 a_i 参与后续分配过程即可。

3) 同一项目中的不同任务的重要性不同。显然,一些关键任务会相比非关键任务对于项目的完成具有更重要的影响。因此,在任务分配时应为关键任务分配能力更强的优秀工人。

4) 考虑到现实世界中众包工人与任务之间可能存在的各种冲突因素,例如完成任务的时间冲突、合作习惯的冲突以及地理位置的冲突等^[1],导致某些工人无法在一起高效地合作开展工作,即工人之间可能存在协作冲突,这些工人不能被分配参与同一任务或项目。

5) 结合众包平台上的历史数据,从技能水平、任务完成效率、任务完成质量、平台活跃程度 4 个指标对 a_i 的能力进行综合评价。其中,技能水平反映工人的软件设计、开发、测试等方面的专业能力熟练程度;任务完成效率体现工人完成任务的时效性;任务完成质量体现工人最终提交的成果的优劣程度;平台活跃程度反映工人的社交和协作意愿。以上每项指标可由多个子指标来进一步刻画。例如,工人技能水平可从岗位(如前端工程师、开发工程师)、应用类型(如移动应用、微信应用)、开发语言(如 Java、HTML5)等方面进行精细度量。通常而言,参与任务的工人的综合能力越强,该任务的最终完成质量越高。

6) 为确保获得理想的众包任务完成质量并控制好项目成本开支,发包者可分别对参与 r_j 的 a_i 的各个能力指标设定需求期望(或偏好)^[19]。需求期望体现了任务对工人能力的阈值要求。例如,一个工人曾独立完成过高难度的任务,但其平台活跃度和任务完成效率偏低,则该工人并不适合参加对协作性、实时性要求较高的任务。现实中,工人的综合能力越强,通常其雇佣成本也越高,因此,有必要遵循“工人能力够用即可”原则来挑选工人,通过合理设定各个任务的需求期望,以兼顾项目的整体经费开支。

综上,多对多型软件众包任务分配问题,就是在满足给定约束的前提下,基于工人的多指标能力评估和任务需求期望分析,建立 n 个任务和 m 个候选者之间的映射关系,为各个待分配任务寻找综合能力尽可能优秀的候选工人,并确保为所

有任务挑选出来的工人的全局综合胜任度最高,从而争取整个项目获得最优的完成质量。

3 基于模糊区间数的众包工人能力度量

3.1 基本定义

定义 1 区间数。设 b^L, b^H 是两个实数,如果 $B = [b^L, b^H], 0 \leq b^L \leq b^H \leq 1$,则称 B 为一个区间数。 b^L 和 b^H 分别称为区间的下限和上限。

定义 2 众包工人的综合能力。它表示为 $U_i = \{U_i^0, U_i^1, U_i^2, U_i^3\}$ 。其中, $U_i^0 = [U_i^{0L}, U_i^{0H}]$ 表示技能水平;类似地, $U_i^1 = [U_i^{1L}, U_i^{1H}]$ 、 U_i^2, U_i^3 分别表示工人的历史任务完成效率、历史任务完成质量、在软件众包平台上的活跃程度。

定义 3 任务的需求期望。发包者可定义各个 r_j 对工人能力的需求期望 $D_j = \{D_j^0, D_j^1, D_j^2, D_j^3\}$ 。其中, $D_j^0 = [D_j^{0L}, D_j^{0H}]$ 表示技能水平期望;类似地, D_j^1, D_j^2 和 D_j^3 分别表示任务完成效率期望、完成质量期望和活跃度期望。

3.2 基于历史评价的工人能力模糊度量计算

众包平台通常以五星制评价工人的历史表现,为便于计算,将五星评分转换为 $[0, 1.0]$ 内的数值。一星对应 0.2,二星对应 0.4,依此类推,五星对应 1.0。基于这些数据计算 a_i 的能力区间数:历史评价只有 1 个值 v 时, $U_i^k = [v, v]$;历史评价有 2 个值 v_1 和 v_2 时, $U_i^k = [v_1, v_2], v_1 \leq v_2$;历史评价数量为 3~5 时,令 v_1 和 v_2 分别表示最小和最大值,则 $U_i^k = [v_1, v_2]$;历史评价数量大于等于 6 时,使用云模型(cloud model)方法^[20]将各评价指标对应的单值评价转换为区间数。本文为各评价指标建立评价云模型,以识别它们的区间范围。因原始数据量大小决定了所生成的云模型的准确性,结合以往研究,选取阈值为 6。

以技能水平指标为例,介绍 $U_i^0 = [U_i^{0L}, U_i^{0H}]$ 的上限和下限值的计算方法。令 $V_i = \{e_1^i, e_2^i, \dots, e_\tau^i\}$ 为 a_i 的技能水平单值评价集合, τ 是评价总数, V_i 被视为云滴送入逆向云生成器(reverse cloud generator)^[21]。从而计算得到 a_i 的技能水平云模型的 3 个数字特征:

$$\begin{cases} C_i^{Ex} = \frac{1}{|V_i|} \times \sum_{k=1}^{|V_i|} e_k^i \\ C_i^{En} = \sqrt{\frac{\pi}{2}} \times \frac{1}{|V_i|} \sum_{k=1}^{|V_i|} |e_k^i - C_i^{Ex}| \\ C_i^{HE} = \sqrt{\left| \frac{1}{|V_i| - 1} \sum_{k=1}^{|V_i|} (e_k^i - C_i^{Ex})^2 - (C_i^{En})^2 \right|} \end{cases} \quad (1)$$

其中, C_i^{Ex} 、 C_i^{En} 和 C_i^{HE} 分别表示 a_i 的技能水平云模型的期望、熵和超熵。

基于 a_i 的技能水平云模型,可求得 U_i^0 为:

$$\begin{cases} U_i^{OL} = C_i^{Ex} + C_i^{En} + C_i^{HE} \cdot \gamma \\ U_i^{OH} = C_i^{Ex} - C_i^{En} - C_i^{HE} \cdot \gamma \end{cases} \quad (2)$$

其中, γ 是 C_i^{HE} 的影响系数,建议其取值范围为 $[0.1, 0.2]$ ^[21]。类似可求得其他指标的区间数。

3.3 任务需求期望导向的工人能力匹配度计算

通过计算区间数可能度,可得到工人 a_i 的各项能力评价与任务 r_j 的需求期望的匹配度。以技能水平为例,令 $U_i^0 = [U_i^{OL}, U_i^{OH}]$ 表示 a_i 的技能水平区间数, $D_j^0 = [D_j^{OL}, D_j^{OH}]$ 表示 r_j 对工人的技能水平期望区间数,记 $l_{U_i^0} = [U_i^{OH}, U_i^{OL}]$, $l_{D_j^0} = [D_j^{OH}, D_j^{OL}]$,则 $U_i^0 \geq D_j^0$ 的可能度^[22] 为:

$$P(U_i^0 \geq D_j^0) = \frac{\min\{l_{U_i^0} + l_{D_j^0}, \max\{U_i^{OH} - D_j^{OL}, 0\}\}}{l_{U_i^0} + l_{D_j^0}} \quad (3)$$

由式(3)可计算工人与每个任务在各个评价指标的可能度,并最终得到工人与任务两两之间关于评价指标匹配度的模糊决策矩阵 \mathbf{P} , $P_{i,j,k}$ 表示第 i ($0 \leq i < m$) 个候选工人的能力对应第 j ($0 \leq j < n$) 个任务的期望在第 k 个评价指标 ($0 \leq k < 4$) 的可能度。

3.4 工人能力评价指标权重的模糊层次分析

在实时型或协作频繁型应用等场景下,发包者对工人的各个能力指标的需求存在差异性。模糊层次分析 (fuzzy analytic hierarchy process, FAHP) 既可减少主观设定权重的偏差,又可避免计算量大且精度不高的问题,适用于解决多指标综合问题^[22]。令模糊判断矩阵 $\mathbf{B} = [b_{i,j}]_{n \times n}$, 满足 $0 \leq b_{i,j} \leq 1$, n 是工人能力指标个数。 $b_{i,j}$ 是第 i 个能力指标与第 j 个能力指标的重要性之比。采用由 0.1、0.3、0.5、0.7、0.9 构成的五级互补标度^[21] 确定 $b_{i,j}$ 。

矩阵 \mathbf{B} 满足 $b_{i,j} + b_{j,i} = 1$, $b_{i,i} = 0.5$, \mathbf{B} 为模糊互补判断矩阵。当满足 $b_{i,j} = b_{i,k} - b_{j,k} + 0.5$ 时, \mathbf{B} 为模糊一致性矩阵。令 b_i 为模糊互补矩阵中第 i 行之和,按式(4)^[23] 将 \mathbf{B} 转换为模糊一致性矩阵。

$$z_{i,j} = (b_i - b_j) / [2(n-1)] + 0.5 \quad (4)$$

仅 $\mathbf{Z} = [z_{i,j}]_{n \times n}$ 是模糊一致性矩阵, $n=4$ 。计算 \mathbf{Z} 中每行的和,并进行标准化处理,按式(5)获得工人能力指标的权重向量 $\mathbf{W} = [w_1, w_2, \dots, w_4]$ 。

$$w_k = (\sum_{t=1}^4 z_{k,t} + 1) / 12 \quad (5)$$

3.5 基于能力模糊度量的工人综合胜任度计算

基于能力指标的权重向量,最终可获得各个 a_i 完成 r_j 的综合胜任度,计算方法如式(6)所示。

$$Q_{i,j} = \sum_{k=0}^3 w_k \cdot P_{i,j,k} \quad (6)$$

$Q_{i,j}$ 表示工人 a_i 对于任务 r_j 的综合胜任能力,该值越大,说明此工人对于该项任务的胜任力越强,则将来可以获得更高的任务完成质量。

4 基于角色协同的众包任务分配模型

4.1 基本模型

借鉴 E-CARGO 模型^[6-8],多对多型软件众包任务分配问题可抽象为: $\sum :: = \langle E, C, O, R, A, G \rangle$ 。其中: E 表示一个涉及多个工人和多个任务的问题环境 environment; C 是 E 中抽象概念的类 (class) 集合; O 是与 C 相关的具体对象 (object) 集合; R 是待分配的任务集合,它对应 RBC 中的 role; A 是候选的工人集合,它对应 RBC 中的 agent; G 是工作组 (group),即由任务分配算法建立的工人团队。令 $m = |A|$ 表示候选工人的数量, $n = |R|$ 表示待分配的众包任务的数量。模型中的关键组件包括:

1) 工人综合能力评估矩阵 \mathbf{Q} : 它表示工人 a_i 对于待分配任务 r_j 的胜任程度。工人的综合能力可通过工人的技能水平、历史任务完成效率和完成质量、平台活跃程度等因素^[1] 来衡量, \mathbf{Q} 的计算方法如式(7)所示。

2) 任务分配矩阵 $\mathbf{T}: T_{i,j} \in \{0,1\}$ ($0 \leq i < m, 0 \leq j < n$), 表示工人 a_i 是否被分配到任务 r_j , 取值为 1 时表示 a_i 被分配了 r_j 。在基于角色协同的任务分配问题中, $Q_{i,j} > 0$ 是 $T_{i,j} = 1$ 的充分条件,否则无解,即能力值为 0 的工人不应被分配任务。

3) 工作组性能 ρ : 所有被分配了任务的工人形成一个工作组 G 。 ρ 表示 G 中所有工人的能力值总和。 ρ 越大,表示所有任务的完成质量越高。对于一组软件众包任务,需要最大化发挥工人的能力,以确保所有任务的 ρ 最高。因此,一个任务并不一定被分配给一个胜任力最强的工人。

4.2 约束定义

1) 任务的权重向量 $\mathbf{W}: W_j \in [0,1]$, 表示任务 r_j 的权重值,它满足 $\sum_{j=0}^{n-1} W_j = 1$ 。

2) 任务的工人数量需求向量 $L: L_j$ 表示任务 r_j 需要的最少工人数量。如果 $L_j \geq 1$, 表示任务 r_j 需要多个工人协作来完成此项众包任务。

3) 工人相关的冲突矩阵 C^A : 工人与工人之间可能存在多种冲突因素, 比如工人间的合作习惯冲突、合作地理位置冲突等, 从而导致某些工人不能被分配完成同一项任务。 C^A 为一个 $m \times m$ 矩阵, 且 $C_{i_1, i_2}^A \in \{0, 1\}$, $C_{i_1, i_2}^A = 1$ 表示工人 a_{i_1} 与工人 a_{i_2} 存在冲突。

4) 工人与任务的冲突矩阵 C^R : 待分配的众包任务与工人之间可能冲突, 比如任务的完成时间与工人的可用时间冲突、任务的地理位置需求与工人常居位置的冲突等, 从而导致某个工人不适合参加某个任务。 C^R 为一个 $m \times n$ 矩阵, 且 $C_{i, j}^R \in \{0, 1\}$, $C_{i, j}^R = 1$ 表示工人 a_i 与任务 r_j 存在冲突。

为提高任务分配的效率和成功率, 在任务分配时需要避开以上可能存在冲突的分配组合。

4.3 目标函数

基于以上定义, 可以得到基于角色协同的众包任务分配问题的目标函数:

$$\max \rho = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} Q_{i, j} \cdot C_{i, j}^R \times T_{i, j} \cdot W_j \quad (7)$$

它应该满足以下条件:

$$T_{i, j} \in \{0, 1\} \quad 0 \leq i < m, 0 \leq j < n \quad (8)$$

$$\sum_{i=0}^{m-1} T_{i, j} = L_j \quad 0 \leq j < n \quad (9)$$

$$\sum_{j=0}^{n-1} T_{i, j} \leq 1 \quad 0 \leq i < m \quad (10)$$

$$C_{i_1, i_2}^A \times T_{i_1, j} \times T_{i_2, j} \leq 1 \quad (11)$$

$$0 \leq i_1, i_2 < m, i_1 \neq i_2, 0 \leq j < n$$

$$C_{i_1, i_2}^A \times T_{i_1, j_1} \times T_{i_2, j_2} \leq 1$$

$$0 \leq i_1, i_2 < m, i_1 \neq i_2, 0 \leq j_1, j_2 < n, j_1 \neq j_2 \quad (12)$$

$$C_{i, j}^R \in \{0, 1\} \quad 0 \leq i < m, 0 \leq j < n \quad (13)$$

$$C_{i, i}^A \in \{0, 1\} \quad 0 \leq i < m \quad (14)$$

其中: 约束(8)表示工人是否被分配; 约束(9)表示每个任务被分配的工人数量满足该任务所要求的人数限制; 约束(10)表示工人只能承接一项任务; 约束(11)和约束(12)表示存在冲突的两个工人不能被分配在同一个任务或同一个项目中; 约束(13)和约束(14)表示一个工人与其他工人或任务间只存在有冲突与无冲突的两种状态。求解基于角色协同的众包任务分配问题就是要建立一个可以获得最大工作组性能 ρ 的候选工人集合。

5 众包任务分配问题的求解

5.1 求解过程

以上众包任务分配问题的求解过程如下:

Step 1: 从众包平台获取工人能力评价的相关原始数据、发包者对工人的能力期望 D 、任务的下限向量 L 以及冲突矩阵 C^A 和 C^R 等。

Step 2: 将工人能力评价的原始数据按 3.2 节介绍的方法转换为区间数 U 。

Step 3: 比较工人能力与待分配任务的需求期望区间数, 按 3.3 节的公式计算工人与任务间关于评价指标匹配度的三维模糊决策矩阵 P 。

Step 4: 基于 3.4 节的介绍, 使用 FAHP 方法计算工人能力指标的权重向量 W 。

Step 5: 使用 3.5 节的式(6)和 W 计算得到各个 a_i 完成 r_j 的综合胜任度, 从而获得候选工人的综合能力值矩阵 Q 。

Step 6: 根据给定的 Q, C^A 和 C^R , 对目标函数求解, 得到 ρ 的最大值和任务分配矩阵 T 。

Step 7: 如果 T 可用, 则 T 为最终的分配方案; 否则无解, 此时众包平台通知任务发包者对任务要求进行调整, 直至获得可行的最优解。

5.2 基于 CPLEX 的求解方法

借鉴基于角色协同的任务分配研究^[7-8], 采用 IBM CPLEX 优化包求解, 实现步骤如下:

Step 1: 确定与 CPLEX 接口的核心数据结构。CPLEX 需要目标函数系数、约束系数、右侧约束值以及上下限等输入参数。使用 Q, L 和 T 定义 CPLEX 中的线性规划问题, Q 是目标函数系数, T 是变量, $Q_{i, j} \in [0, 1], T_{i, j} \in \{0, 1\}$ 。

Step 2: 使用工人与任务间的冲突矩阵 C^R 对 Q 中的值进行提前修正, $Q_{i, j} = Q_{i, j} \cdot C_{i, j}^R$ 。

Step 3: 设定目标函数和约束表达式。众包任务分配问题的目标函数由矩阵 Q 和 T 的一维数组形式和 L 的线性表达式组成。首先, 将 Q 和 T 变换成一维阵列: $X_{i \times n + j} = T_{i, j}$ 和 $V_{i \times n + j} = Q_{i, j}$ 。然后, 添加优化目标, 调用以下方法: `lloIntVar[] X = cplex. intVarArray (m * n, 0, 1); cplex. addMaximize(cplex. scalProd(X, V)`。最后, 迭代添加约束表达式, 以 Java 代码为例, 具体实现方法见算法 1。

Step 4: 调用 CPLEX 的 `cplex. solve()` 方法, 根据目标函数和约束条件表达式来计算最大化的 ρ 值。

算法 1 添加约束表达式

Alg.1 Adding constraint expressions

输入:工人数 m ,任务数 n ,转换后的一维阵列 X 和 V ,
 工人数量需求向量 L ,工人相关的冲突矩阵 CA
 输出:添加约束后的 CPLEX 对象

```

/* 添加约束(9) */
for(int i=0; i<n; i++){//迭代 n 个任务
  lloLinearNumExpr reqC = cplex.linearNumExpr();
  for(int j=0; j<m; j++){//迭代 m 个工人
    reqC.addTerm(1, X[i+j*n]); //换算承担任务否
  }
  cplex.addEq(reqC, L[i]); //添加任务 i 的人数限制
}
/* 添加约束(10) */
for(int i=0; i<m; i++){//迭代 m 个工人
  lloLinearNumExpr uniC = cplex.linearNumExpr();
  for(int j=0; j<n; j++){//迭代 n 个任务
    uniC.addTerm(1, X[n*i+j]); //换算承担任务否
  }
  cplex.addLe(uniC, 1.0); //添加工人 i 的唯一约束
}
/* 添加约束(11)和约束(12) */
for(int i=0; i<m*m; i++){//迭代 C 中每个冲突值
  int row = i/m; int col = i%m; //换算出行、列值
  if(row >= col) continue; //仅检测上三角矩阵
  if(1 == CA[i]){//找到冲突位置
    lloLinearNumExpr cf = cplex.linearNumExpr();
    for(int j=0; j<n; j++){//找出有冲突的 2 个工人
      cf.addTerm(1, X[row*n+j]);
      cf.addTerm(1, X[col*n+j]);
    }
    cplex.addLe(cf, 1); //限定 2 个工人不同时被选
  }
}

```

表 1 候选工人的能力评价

Tab.1 Evaluations of candidate workers' qualification

A	U			
	u_0	u_1	u_2	u_3
a_0	[0.52,0.63]	[0.51,0.60]	[0.48,0.55]	[0.42,0.51]
a_1	[0.48,0.59]	[0.41,0.52]	[0.44,0.52]	[0.38,0.45]
a_2	[0.55,0.68]	[0.48,0.58]	[0.50,0.61]	[0.41,0.48]
a_3	[0.51,0.66]	[0.56,0.65]	[0.48,0.62]	[0.45,0.55]
a_4	[0.47,0.57]	[0.45,0.55]	[0.43,0.51]	[0.35,0.49]
a_5	[0.55,0.66]	[0.52,0.65]	[0.49,0.63]	[0.47,0.57]
a_6	[0.52,0.66]	[0.51,0.55]	[0.51,0.61]	[0.51,0.58]
a_7	[0.58,0.69]	[0.55,0.65]	[0.55,0.60]	[0.55,0.60]
a_8	[0.43,0.54]	[0.41,0.48]	[0.38,0.47]	[0.39,0.48]
a_9	[0.41,0.55]	[0.38,0.50]	[0.35,0.51]	[0.36,0.48]
a_{10}	[0.58,0.68]	[0.55,0.63]	[0.55,0.63]	[0.48,0.56]
a_{11}	[0.54,0.69]	[0.51,0.66]	[0.52,0.65]	[0.52,0.61]

发包者为任务设定的需求期望如表 2 所示。

表 2 各个任务对工人能力的需求期望

Tab.2 Demands for workers' qualification of tasks

R	U			
	u_0	u_1	u_2	u_3
r_0	[0.40,1.00]	[0.45,1.00]	[0.40,1.00]	[0.39,1.00]
r_1	[0.35,1.00]	[0.40,1.00]	[0.45,1.00]	[0.35,1.00]
r_2	[0.44,1.00]	[0.38,1.00]	[0.42,1.00]	[0.45,1.00]
r_3	[0.38,1.00]	[0.35,1.00]	[0.35,1.00]	[0.30,1.00]
r_4	[0.30,1.00]	[0.40,1.00]	[0.38,1.00]	[0.40,1.00]

由式(3)计算工人能力与任务需求的匹配度模糊决策矩阵。第 1 个工人的匹配度模糊决策子矩阵数据示例如表 3 所示。再使用权重 W 加权求和,获取的如表 4 所示的工人综合胜任度矩阵 Q 。

表 3 第 1 个工人的匹配度 $P_{0,j,k}$

Tab.3 Matching degree $P_{0,j,k}$ of worker #1

R	U			
	u_0	u_1	u_2	u_3
r_0	0.324	0.234	0.224	0.171
r_1	0.368	0.290	0.161	0.216
r_2	0.284	0.310	0.200	0.094
r_3	0.342	0.338	0.278	0.266
r_4	0.407	0.290	0.246	0.159

6 案例分析

假定当前环境下需要 9 名工人完成 5 个任务,用 $R = \{r_0, r_1, \dots, r_4\}$ 表示任务集,假设有 12 名工人参加竞标,用 $A = \{a_0, a_1, \dots, a_{11}\}$ 表示候选工人集。用 $U = \{u_0, u_1, u_2, u_3\}$ 表示工人综合能力的评价指标集, u_0 代表技能水平、 u_1 代表任务完成效率、 u_2 代表任务完成质量、 u_3 代表平台活跃程度。在计算各工人的综合能力后,采用基于角色协同的任务分配方法来最优化分配任务。假定由 FAHP 方法计算得到的权重值为 $W = [0.15, 0.25, 0.50, 0.10]$ 。从众包平台获取工人的原始评价数据,转换为区间数值,如表 1 所示。

表 4 工人的综合能力评估值

Tab.4 Comprehensive qualification values of workers

R	A					
	a_0	a_1	a_2	a_3	a_4	a_5
r_0	0.236	0.164	0.269	0.301	0.169	0.309
r_1	0.230	0.159	0.267	0.299	0.163	0.307
r_2	0.229	0.157	0.264	0.296	0.163	0.304
r_3	0.301	0.235	0.331	0.358	0.238	0.365
r_4	0.273	0.203	0.302	0.333	0.209	0.341

R	A					
	a_6	a_7	a_8	a_9	a_{10}	a_{11}
r_0	0.273	0.324	0.105	0.134	0.325	0.336
r_1	0.272	0.317	0.101	0.132	0.322	0.333
r_2	0.272	0.318	0.101	0.129	0.320	0.331
r_3	0.337	0.382	0.182	0.203	0.384	0.389
r_4	0.309	0.356	0.149	0.174	0.358	0.366

现实中可能存在多种冲突,假定工人与任务间的冲突约束 C^R 如下所示:

$$C^R = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

接下来,利用 CPLEX 优化包求解,设 $L = [2, 1, 3, 2, 1]$,得到最优解:

$$T = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

根据以上结果,可得到每个待分配任务对应

的众包工人组合,如表 5 所示。

表 5 任务分配结果

Tab.5 Task assignment result

任务编号	工作者组合
r_0	a_2, a_7
r_1	a_5
r_2	a_6, a_{10}, a_{11}
r_3	a_1, a_4
r_4	a_0

综上可知,通过评估候选工人的综合能力,建立基于角色协同的任务分配模型,求解后获得多对多型任务分配方案的目标函数最大值为 2.569。

7 实验分析

为验证所提方法,采用仿真数据进行实验分析。使用 64 位 Windows 10、Eclipse、Java 语言。CPU 为 Intel Core i5-9600K、4.30 GHz,内存为 16 GB DDR 4、3 200 MHz,IBM CPLEX Optimization Studio 版本 12.7。借鉴任务分配问题常用的实验设定^[6,8],取 $n = \lfloor m/3 \rfloor$ 或 $n = \lfloor m/4 \rfloor$, m 从 10 变化到 60、步长为 5,执行 50 次取平均值。

7.1 数据预处理时间分析

运用所提方法进行任务协同分配前,需搜集众包平台上工人能力的原始评价数据并将它们转换为区间数,再与任务需求期望区间数进行可能度计算,继而通过权重合成后获得工人的综合能力评估矩阵 Q 。实验分析以上数据预处理过程的时间消耗。考虑到现实中能力指标值很低的工人较少可能竞争到众包任务,设定工人能力和任务需求期望的随机范围为 $[0.3, 1.0]$,设定 $n = \lfloor m/3 \rfloor$ 或 $n = \lfloor m/4 \rfloor$ 、冲突概率为 0.1,实验结果如图 1 所示。

由图 1 可知,随着工人数量和任务数量的不断增多,数据预处理所消耗的时间逐渐增加,呈现近似于线性增长的趋势。在任务数为 20、候选工人数为 60 的最大数据规模下,多对多型软件众包任务分配问题的数据预处理时间低于 0.07 ms,能够具有较理想的处理性能。

7.2 执行耗时分析

实验设定 $n = \lfloor m/3 \rfloor$ 、冲突概率为 0.1。考虑到每个 $L_j > 1$ 的任务可等价于 $L_j = 1$ 的多个相同任务,统一设定 $L_j = 1, j \in \{1, 2, \dots, n\}$ 。每个步骤中初始化 Q, C^A, C^R ,对比本文方法与

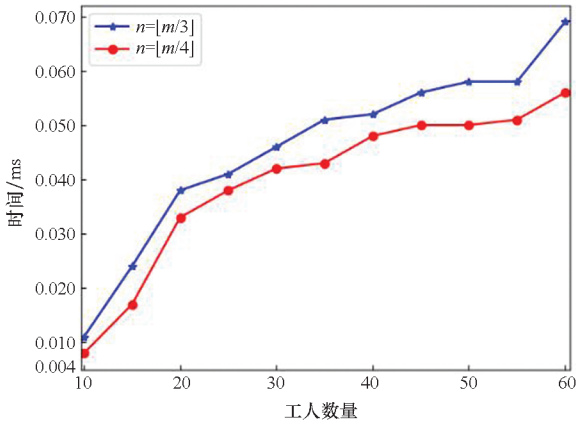


图1 数据预处理时间分析

Fig. 1 Analysis of data preprocess time

贪心方法、穷举方法的执行时间,结果如表6所示。

表6中,N/A表示在30 min无法获得最优解。由表5、表6可知,随着工人数量的增加,3种方法的耗时随之明显增加,尤其是穷举方法的

执行耗时呈指数增长。在 $n = \lfloor m/3 \rfloor$ 的实验中,当工人数小于等于35、任务数小于等于11时,贪心算法的执行性能明显优于本文方法;但是,随着问题规模的增大,当工人数大于等于40、任务数大于等于13以后,在某些特殊的随机数据情况下,贪心方法的执行时间可能急剧增加,在30 min内无法获得优化解。针对 $m = 40, n = 13$ 的情况下进行了深入分析,在多个特殊的随机数据样例(原始数据发布于百度云)下贪心方法与本文方法的耗时对比如表7所示。由表7可知,在问题复杂度较大(即工人数和任务数具有一定规模)时,贪心方法在不同数据样例情况下的执行时间具有不确定性,其原因是,在一些特殊冲突矩阵 C^A 和工人综合能力评估矩阵 Q 场景下,贪心方法在寻找优化解过程中为了避免冲突情况要进行回溯,而随机生成的冲突矩阵可能导致回溯过于频繁,从而使整体执行时间急剧增加。

表6 $n = \lfloor m/3 \rfloor$ 时执行耗时分析

Tab. 6 Execution time analysis when $n = \lfloor m/3 \rfloor$

单位:ms

求解方法	工人数量										
	10	15	20	25	30	35	40	45	50	55	60
本文方法	2.62	2.86	3.46	4.56	7.60	10.94	20.0	29.1	33.94	45.9	66.9
贪心方法	0.02	0.04	0.08	0.09	0.35	0.61	N/A	N/A	N/A	N/A	N/A
穷举方法	0.10	46.93	3 651.00	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

对比结果可知,冲突约束下多对多型软件众包任务分配问题求解时,即使在工人数为60、任务数为20的最复杂情况下,本文基于CPLEX的求解方法仅耗时66.9 ms,始终能获得令人满意的执行性能。

设定 $n = \lfloor m/3 \rfloor, L_j = 1, j \in \{1, 2, \dots, n\}$, 冲突概率分别设置为0.1~0.4。结果如图2和表8所示。

由图2可知,随着问题规模的增加,命中最优解的概率逐渐下降,当问题规模达到一定程度后,命中率将急剧下降。 $n = \lfloor m/3 \rfloor$ 的实验中,冲突概率大于等于0.2时, m 大于等于40以后获得最优

表7 执行耗时对比

Tab. 7 Comparison of execution time

单位:ms

求解方法	样例数据编号					
	#1	#2	#3	#4	#5	#6
本文方法	12.0	12.0	29.0	11.0	23.0	43.0
贪心方法	2 638.6	5 369.3	19 901.0	119 145.5	22 139.1	853 762.0

7.3 命中最优解的概率分析

以穷举法获得的最优解为基准,在不同冲突概率情况下测试本文方法命中最优解的概率。实验

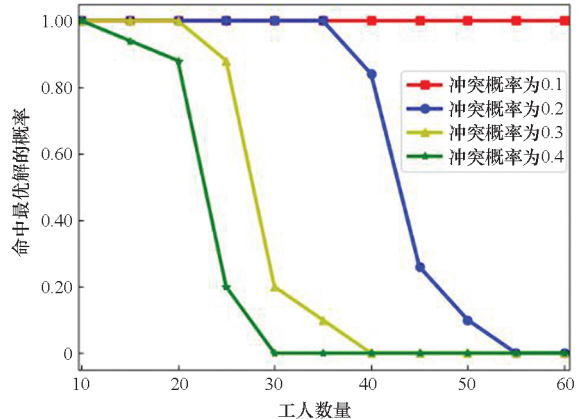


图2 命中最优解的概率

Fig. 2 Hit probability of optimal solution

表 8 $n = \lfloor m/3 \rfloor$ 情况下的精度分析
Tab. 8 Accuracy analysis when $n = \lfloor m/3 \rfloor$

冲突概率	工人数量											
	10	15	20	25	30	35	40	45	50	55	60	
0.1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
0.2	1.00	1.00	1.00	1.00	1.00	1.00	0.84	0.26	0.10	0	0	0
0.3	1.00	1.00	1.00	0.88	0.20	0	0	0	0	0	0	0
0.4	1.00	0.94	0.88	0.20	0	0	0	0	0	0	0	0

解的概率就明显不理想了。实验表明,在固定 n/m 比值的情况下,工人人数越多、冲突概率越大,则命中最优解的概率越低。

7.4 求解精度分析

本实验以穷举法的结果为基准,对比本文方法与贪心方法的求解精度。实验设定任务数为 5,冲突概率为 0.1,工人数量以步长 5 在 [10, 60] 范围内变动。求解精度值的计算公式为:

$$F = 1 - (\rho^* - \rho) / \rho^* \quad (15)$$

式中, ρ^* 是穷举法获得的最优的工作组性能值, ρ 是其他方法获得的工作组性能值。

实验结果如图 3 和表 9 所示。由图 3 可知,本文方法在较大问题规模情况下均能准确获得最优解,求解精度始终为 1。而贪心方法虽然执行性能较高,但无法保证始终获得最优解,尤其

在问题规模偏小时,其准确率明显不理想。图 3 中,当候选工人增多时,贪心方法的精度得以明显提高。

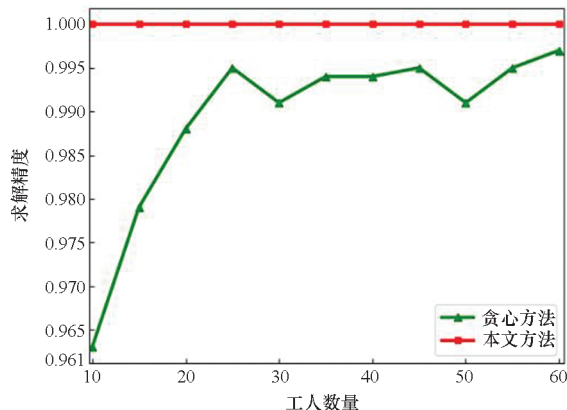


图 3 精度对比

Fig. 3 Comparison of accuracy

表 9 精度对比
Tab. 9 Comparison of accuracy

求解方法	工人数量											
	10	15	20	25	30	35	40	45	50	55	60	
本文方法	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
贪心方法	0.963	0.979	0.988	0.995	0.991	0.994	0.994	0.995	0.991	0.995	0.997	0.997

7.5 评价

总体来看,本文方法的整体表现明显优于其他方法,在一定规模的众包任务分配求解时,能保证良好的执行效率,并始终能够确保用户获得最优解。基于本文方法可以提高软件众包的任务完成质量,降低项目的开发风险。

8 结论

针对工人能力的不确定性和任务属性的多样性,提出一种众包工人综合胜任能力的模糊度量方法。它结合工人的历史记录和任务的需求期望,以模糊区间数表征众包工人的历史表

现,从技能水平、任务完成效率、任务完成质量和平台活跃程度 4 个指标评价工人的综合能力,计算多属性模糊能力匹配度,应用模糊层次分析法获得工人的综合胜任能力。引入 RBC 理论和 E-CARGO 模型,将涉及任务集与工人组的多对多任务分配问题形式化建模为基于角色协同的组合优化问题,该模型以工人综合胜任能力为基础,综合考虑任务的权重约束、工人数量需求约束、工人与任务间的冲突约束等,以提高任务分配的效率和成功率,并给出了一种基于 CPLEX 的问题求解方法。通过案例和实验分析了所提方法的可行性和有效性,所提方法可为

软件众包应用提供重要的理论参考和决策基础。未来研究中,将对工人综合能力的动态度和动态任务分配、包含同步任务的工人多角色协同分配等进行深入探讨。

参考文献(References)

- [1] MAO K, CAPRA L, HARMAN M, et al. A survey of the use of crowdsourcing in software engineering [J]. *Journal of Systems and Software*, 2017, 126: 57–84.
- [2] GUO B, LIU Y, WANG L Y, et al. Task allocation in spatial crowdsourcing: current state and future directions[J]. *IEEE Internet of Things Journal*, 2018, 5(3): 1749–1764.
- [3] 马华, 陈跃鹏, 唐文胜, 等. 面向工作者能力评估的众包任务分配方法的研究进展综述[J]. *计算机应用*, 2021, 41(8): 2232–2241.
MA H, CHEN Y P, TANG W S, et al. Survey of research progress on crowdsourcing task assignment for evaluation of workers' ability [J]. *Journal of Computer Applications*, 2021, 41(8): 2232–2241. (in Chinese)
- [4] 蒋巍川, 蒋玖川. 社会网络中复杂任务的群智协同众包[J]. *中国计算机学会通讯*, 2018, 14(11): 25–30.
JIANG Y C, JIANG J C. Swarm intelligence collaboration crowdsourcing of complex tasks in social networks [J]. *Communications of the CCF*, 2018, 14(11): 25–30. (in Chinese)
- [5] ZHU H B, ZHOU M C. Role-based collaboration and its kernel mechanisms[J]. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 2006, 36(4): 578–589.
- [6] ZHU H B. Avoiding conflicts by group role assignment[J]. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2016, 46(4): 535–547.
- [7] ZHU H B. Maximizing group performance while minimizing budget [J]. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020, 50(2): 633–645.
- [8] MA H, TANG W S, ZHU H B, et al. Resource utilization-aware collaborative optimization of IaaS cloud service composition for data-intensive applications [J]. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021, 51(2): 1322–1333.
- [9] 武小年, 王青芝, 张楚芸, 等. 区间数多属性决策的云服务信任评估方法[J]. *小型微型计算机系统*, 2018, 39(7): 1492–1497.
WU X N, WANG Q Z, ZHANG C Y, et al. Trust evaluation method of cloud service based on interval multiple attribute decision making[J]. *Journal of Chinese Computer Systems*, 2018, 39(7): 1492–1497. (in Chinese)
- [10] OLESON D, SOROKIN A, LAUGHLIN G, et al. Programmatic gold: targeted and scalable quality assurance in crowdsourcing [C]//*Proceedings of the 11th AAAI Conference on Human Computation*, 2011: 43–48.
- [11] 高丽萍, 孙明达, 高丽, 等. 移动众包环境下一种多阶段质量感知的在线激励机制[J]. *小型微型计算机系统*, 2022, 43(5): 1102–1108.
GAO L P, SUN M D, GAO L, et al. Online incentive mechanism of multi-stage quality perception in mobile crowdsourcing environment[J]. *Journal of Chinese Computer Systems*, 2022, 43(5): 1102–1108. (in Chinese)
- [12] RAYKAR V C, YU S P. An entropic score to rank annotators for crowdsourced labeling tasks[C]//*Proceedings of the Third National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics*, 2011: 29–32.
- [13] HIRTH M, HÖBFELD T, TRAN-GIA P. Analyzing costs and accuracy of validation mechanisms for crowdsourcing platforms[J]. *Mathematical and Computer Modelling*, 2013, 57(11/12): 2918–2932.
- [14] 芮兰兰, 张攀, 黄豪球, 等. 一种面向众包的基于信誉值的激励机制[J]. *电子与信息学报*, 2016, 38(7): 1808–1815.
RUI L L, ZHANG P, HUANG H Q, et al. Reputation-based incentive mechanisms in crowdsourcing [J]. *Journal of Electronics & Information Technology*, 2016, 38(7): 1808–1815. (in Chinese)
- [15] 冉家敏, 倪志伟, 彭鹏, 等. 考虑空间众包工作者服务质量的众包任务分配策略及其萤火虫群优化算法求解[J]. *计算机应用*, 2021, 41(3): 794–802.
RAN J M, NI Z W, PENG P, et al. Task allocation strategy considering service quality of spatial crowdsourcing workers and its glowworm swarm optimization algorithm solution[J]. *Journal of Computer Applications*, 2021, 41(3): 794–802. (in Chinese)
- [16] 李沁雅, 吴帆, 陈贵海. 众包全景拼接系统下基于图像质量的酬劳分配策略[J]. *计算机学报*, 2021, 44(6): 1246–1257.
LI Q Y, WU F, CHEN G H. Quality-driven reward distribution in crowdsourced panoramic map construction[J]. *Chinese Journal of Computers*, 2021, 44(6): 1246–1257. (in Chinese)
- [17] 童咏昕, 袁野, 成雨蓉, 等. 时空众包数据管理技术研究综述[J]. *软件学报*, 2017, 28(1): 35–58.
TONG Y X, YUAN Y, CHENG Y R, et al. Survey on spatiotemporal crowdsourced data management techniques[J]. *Journal of Software*, 2017, 28(1): 35–58. (in Chinese)
- [18] 李洋, 贾梦迪, 杨文彦, 等. 基于树分解的空间众包最优任务分配算法[J]. *软件学报*, 2018, 29(3): 824–838.
LI Y, JIA M D, YANG W Y, et al. Optimal task assignment algorithm based on tree-decouple in spatial crowdsourcing[J]. *Journal of Software*, 2018, 29(3): 824–838. (in Chinese)
- [19] ZHAO H, XIAO M J, WU J, et al. Differentially private unknown worker recruitment for mobile crowdsensing using multi-armed bandits [J]. *IEEE Transactions on Mobile Computing*, 2021, 20(9): 2779–2794.
- [20] WANG G Y, XU C L, LI D Y. Generic normal cloud model[J]. *Information Sciences*, 2014, 280: 1–15.
- [21] MA H, ZHU H B, HU Z G, et al. Time-aware trustworthiness ranking prediction for cloud services using interval neutrosophic set and ELECTRE [J]. *Knowledge-Based Systems*, 2017, 138: 27–45.
- [22] SUN B Z, MA W M. An approach to consensus measurement of linguistic preference relations in multi-attribute group decision making and application [J]. *Omega*, 2015, 51: 83–92.
- [23] 徐泽水. 多属性决策的两种方差最大化方法[J]. *管理工程学报*, 2001, 15(2): 11–13, 4.
XU Z S. Two methods of maximizing deviations of multi-attribute decision making [J]. *Journal of Industrial Engineering and Engineering Management*, 2001, 15(2): 11–13, 4. (in Chinese)