Quasi Group Role Assignment With Agent Satisfaction in Self-Service Spatiotemporal Crowdsourcing

Qian Jiang, Dongning Liu, *Senior Member, IEEE*, Haibin Zhu, *Senior Member, IEEE*, Baoying Huang, Naiqi Wu, *Fellow, IEEE*, and Yan Qiao, *Senior Member, IEEE*

***Abstract*—Quasi group role assignment (QGRA) presents a novel social computing model designed to address the burgeoning domain of self-service spatiotemporal crowdsourcing (SSC), specifically for tackling the photographing to make money problem (PMMP). Nevertheless, the application of QGRA in practical scenarios encounters a significant bottleneck. QGRA provides optimal assignment strategies under conditions where both the number of crowdsourced tasks and workers remain stable. However, real-world crowdsourcing applications may necessitate the phased integration of new tasks. With the rapid increase in the number of tasks, a set of residual tasks inevitably exists that are difficult to complete. To maximize the completion of crowdsourced tasks, workers may be assigned low-yield or even unprofitable tasks. Given the reluctance of crowdsourcing workers to be overstretched for these tasks, along with the inherent characteristics of self-service crowdsourcing tasks, this can lead to the failure of the assignment scheme. To tackle the identified challenges, this article proposes the QGRA with agent satisfaction (QGRAAS) method. Initially, it sheds light on a creative satisfaction filtering algorithm (SFA), which is engineered to perform optimal task assignments while actively optimizing the profitability of crowdsourcing workers. This approach ensures the satisfaction of workers, thereby fostering their loyalty to the platform. Concurrently, in response to the phased changes in the crowdsourcing environment, this article incorporates the concept of bonus incentives. This aids decision-makers in achieving a tradeoff between the operational costs and task completion rates. The robustness and practicality of the proposed solutions are confirmed through simulation experiments.**

Manuscript received 23 October 2023; revised 23 March 2024 and 17 June 2024; accepted 19 June 2024. Date of publication 11 July 2024; date of current version 2 October 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62072120, in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant RGPIN2018-04818, and in part by the Science and Technology Development Fund (FDCT), Macau SAR under Grant 0015/2020/AMJ, Grant 0018/2021/A1, Grant 0083/2021/A2, and Grant 0011/2023/RIA1. *(Corresponding author: Yan Qiao.)*

Qian Jiang, Baoying Huang, Naiqi Wu, and Yan Qiao are with Macao Institute of Systems Engineering and Collaborative Laboratory for Intelligent Science and Systems, Macau University of Science and Technology, Macao 999078, China (e-mail: qjiang.ieee@gmail.com; baoyhuang@126.com; nqwu@must.edu.mo; yqiao@must.edu.mo).

Dongning Liu is with the School of Computer Science and Technology, Guangdong University of Technology, Guangzhou 510006, Guangdong, China, and also with the Faculty of Data Science, City University of Macau, Macao 999078, China (e-mail: liudn@gdut.edu.cn).

Haibin Zhu is with the Collaborative Systems Laboratory (CoSys Laboratory), Nipissing University, North Bay, ON P1B 8L7, Canada (e-mail: haibinz@nipissingu.ca).

Digital Object Identifier 10.1109/TCSS.2024.3417959

***Index Terms*—Adaptive collaboration (AC), group role assignment (GRA), photographing to make money problem (PMMP), QGRA with agent satisfactions (QGRAAS), quasi-GRA (QGRA), role-based collaboration (RBC), self-service spatiotemporal crowdsourcing (SSC).**

Nomenclature

$\mathcal{R}$	Set of roles.
$\mathcal{A}$	Set of agents.
L	Role range vector.
Q	Qualification matrix.
T	Role assignment matrix.
Σ	Group performance.
t_{start}	Start time of the assignment scheme.
t_{end}	End time of the assignment scheme.
t_{interval}	Time interval for reassignment.
$\overline{Q}$	Capability matrix.
$\mathcal{T}$	Workable role vector.
$\mathcal{T}^a$	Workable agent vector.
$\lozenge$	Role profit vector.
Λ^a	Agent income level vector.
Ξ	Agent cost matrix.
τ^a	Agent satisfaction threshold vector.
u_{task}	Number of new tasks introduced during the reassignment phases.
u_{worker}	Number of new workers introduced during the reassignment phases.
$\gimel$	Role task number vector.
ν	Task incentive bonus.
$\mathcal{V}$	Role incentive bonus vector.
ψ	Agent satisfaction matrix.
$\mathcal{M}$	Workable capability matrix.
$\mathcal{R}_w$	Workable role set.
$\mathcal{A}_w$	Workable agent set.
Q_w	Workable qualification matrix.
L_w	Workable role range vector.
T_w	Workable role assignment matrix.
ρ	Task completion rate expected by the decision-makers.
step	Value representing the step size in searching for the optimal value of ν.
SFA	Satisfaction filtering algorithm.

I. Introduction

THE photographing to make money problem (PMMP) is a booming paradigm of self-service spatiotemporal crowdsourcing (SSC) [1]. Its rapid evolution can be attributed to the widespread proliferation of mobile internet. PMMP seeks to offer enterprises business inspection and information collection services through the active photo contributions of crowdsourcing workers.

In recent years, numerous applications associated with PMMP, such as Foap [2], Meituan Paidian [3], and Antzb [4] have emerged. Addressing the PMMP brings up two main challenges. First, owing to the self-service nature of PMMP, some tasks may not be completed by the assigned workers due to limited profit. Second, to appropriately enhance the task completion rate, decision-makers necessitate an adaptive assignment strategy for frequent redistribution of crowdsourcing tasks. In essence, SSC fundamentally represents an adaptive collaboration (AC) problem [1], [5], [6], [7] with an uncertain task completion rate. To address this, the quasi-group role assignment (QGRA) can be leveraged, as it has a natural match with AC in modeling.

The QGRA model [1] is derived from its environments—classes, agents, roles, groups, and objects (E-CARGO) [8] metamodel and group role assignment (GRA) submodel. Cooperated with its role-based collaboration (RBC) [9], [10], [11] methodology, it has become an efficient computational method for solving SSC-related issues [1]. Still, there exist nontrivial challenges for the practical application of the QGRA model.

QGRA model utilizes the capability matrix $\overline{Q}$ to properly quantify the relationship between crowdsourcing workers' satisfaction and task completion rate. This method is suitable for a relatively stable crowdsourcing environment, implying a stable number of tasks and workers. Nonetheless, real-world applications often require the periodic integration of new tasks and participants. Owing to the self-service characteristics of crowdsourcing tasks, the accumulation of historical tasks tends to increase concomitantly with the uptick in task volume. Under these circumstances, to maximize the number of completed tasks, there exists a propensity to assign low-profit or even unprofitable tasks to overstretched crowdsourcing workers. This practice can detrimentally impact worker satisfaction levels and may even lead to the failure of the task assignment strategy.

To tackle the aforementioned challenges, this article first proposes the QGRA with agent satisfactions (QGRAAS) model. This model primarily incorporates a novel satisfaction filtering algorithm (SFA), which is designed to aid the QGRAAS model in executing optimal assignments while striving to maximize crowdsourcing workers' profit. This further increases the satisfaction of workers, thereby encouraging their loyalty to the crowdsourcing platform.

Meanwhile, to accommodate phased alterations in the crowdsourcing environment, this article integrates the concept of bonus incentives. Existing bonus incentive schemes [12], [13], both personalized and generic, primarily focus on elevating task completion rates, with less emphasis on the consequential cost implications and returns for decision-makers. In contrast, our proposed bonus incentive framework aims to assist decision-makers in achieving a judicious tradeoff between operational expenses and task completion rates, thereby leading to assignment strategies that meet their satisfaction.

Last, an increase in the number of crowdsourcing workers and tasks could potentially lead to a decrease in solution speed. To counter this, have optimized our solution strategy and implemented the Kuhn–Munkres (KM) algorithm [8] in lieu of the IBM ILOG CPLEX optimization package (CPLEX) [14], thereby further accelerating the solution speed.

In summary, the contributions of this article are listed as follows.

1) This article improves the QGRA model for solving SSC-related problems, i.e., QGRAAS, by considering the satisfaction of agents. By maximizing the satisfaction of crowdsourcing workers to the greatest extent possible, this article introduces the SFA method.
2) This article proposes to use a bonus incentive mechanism to assist decision-makers in striking a balance between the costs and the completion rate of crowdsourcing tasks.
3) This article further improves the QGRAAS model by applying the AC method to provide solutions to continuous and dynamic real-world problems, i.e., SSC of tasks.
4) The proposed method moves one more step on the track for the QGRA model to approach practical application.

The organization of this article is as follows: In Section II, we delineate the foundational QGRA model. Section III delves into a real-world scenario pertaining to PMMP. Subsequently, Section IV formalizes the PMMP using the QGRAAS model. Section V elucidates the algorithms tailored for QGRAAS. Comprehensive simulation-based comparative experiments are presented in Section VI, while Section VII reviews related research. Finally, Section VIII concludes the article and highlights avenues for future research.

II. Preliminaries

QGRA serves as an effective unified mathematical modeling method [1] that aptly formalizes SSC-related issues. The priority of utilizing QGRA is determining the roles (i.e., tasks) and the agents (i.e., task executors). To clarify the definition of the QGRA, we use nonnegative integers m ($= |\mathcal{A}|$, where $|\mathcal{A}|$ denotes the cardinality of set $\mathcal{A}$) to articulate the size of the agent set $\mathcal{A}$, and n ($= |\mathcal{R}|$) the size of the role set $\mathcal{R}$. Here, $i \in A = \{0, 1, \ldots, m-1\}$ and $j \in R = \{0, 1, \ldots, n-1\}$ represent the indices of agents and roles, respectively. The key definitions of QGRA are as follows.

Definition 1: A role [15], [16] is defined as $r \overset{\text{def}}{=} <id, ®>$.

Note: *id* denotes the identifier of *r*, e.g., *role j* ($j \in R$); ® symbolizes the set of requirements or properties for agents to perform *r*.

In SSC, it is easy to take the crowdsourcing tasks as roles because they are independent and straightforward. More pertinently, we reasonably divide these tasks into clusters to form the roles. Such an operation can increase roles' profits, thereby improving the agents' likelihoods to perform these roles. Besides, taking task clusters as roles in the PMMP can also convert the many-to-many (M-M) problem to a one-to-many (1-M) problem. This conversion can considerably reduce the time complexity of the distributing process in SSC. In addition, because the tasks in the SSC scenarios are independent, the clustering of tasks does not introduce new constraints. In the context of the QGRA model, roles refer to clustered crowdsourcing tasks, obtained by the role awareness method propounded in [1]. To validate the efficacy of our role awareness method, we will conduct a comparison with M-M assignment algorithms in the experimental section.

Definition 2: An agent [17], [18] is defined as $a \overset{\text{def}}{=} <id, ©>$.

Note: *id* is the identification of *a*, e.g., *Agent i* ($i \in A$); ◎ expresses the agents' performances about ®, as elucidated in Definition 4. In the context of the QGRA model in [1], agents are the crowdsourcing workers.

Definition 3: A *role range vector* [8] L is an n-dimensional row vector of the number of agents required for roles, i.e., $L[j] \in \mathbb{N}^+$.

Definition 4: A *qualification matrix* [19] Q is an $m \times n$ matrix, where $Q[i, j] \in \mathbb{R}^+$ expresses the suitability of an agent i for a role j.

Note: In SSC-related problems, each task within roles is usually simple and requires no specific skills. The most critical criterion for evaluating crowdsourcing workers (agents) is the center distance between an agent and a role. As per Jiang et al. [1], $Q[i, j]$ can be calculated through the haversine formula given that the position information for each role and agent is a sequence of <longitude, latitude> (in radians)

$$Q[i,\ j] = r_{\text{Earth}} \times 2\arcsin\sqrt{\sin^2\frac{y_i - y_j}{2} + \cos(y_i)\times\cos(y_j)\times\sin^2\frac{x_i - x_j}{2}}$$

where r_{Earth} represents the Earth radius, $<x_i, y_i>$ is the location of Agent i and $<x_j, y_j>$ is the location of Role j.

Definition 5: A *role assignment matrix* [10], [20], [21] T is defined as an $m \times n$ control variable matrix, where $T[i, j] \in \{0, 1\}$ indicates whether or not an agent i is assigned to a role j. $T[i, j] = 1$ means yes and 0 no.

Note: The physical significance of the matrix T is that it serves as a control variable in QGRA.

Definition 6: A *capability matrix* [1] $\overline{Q}$ is an $m \times n$ matrix, and $\overline{Q}[i, j] \in \{0, 1\}$ ($i \in A$, $j \in R$). $\overline{Q}[i, j] = 1$ indicates that agent i is willing to play role j and 0 means no.

Note: The matrix $\overline{Q}$ represents the quantification of agents' satisfaction.

Definition 7: A *workable role vector* [1] $\mathcal{T}$ is an n-vector, where $\mathcal{T}[j] \in \{0, 1\}$ ($0 \le j <$ n), and it satisfies the equation that

$$\mathcal{T}[j] = \begin{cases} 0, \sum_{i=0}^{\mathrm{m}-1}\overline{\mathrm{Q}}[\mathrm{i},\ \mathrm{j}] = 0 \\ 1, \sum_{i=0}^{\mathrm{m}-1}\overline{\mathrm{Q}}[\mathrm{i},\ \mathrm{j}] > 0 \end{cases}.$$

Note: $\mathcal{T}[j] = 1$ represents that *role j* will have at least one candidate agent to play it and 0 means no.

Definition 8: A *workable agent vector* [1] $\mathcal{T}^a$ is an m-vector, where $\mathcal{T}^a[i] \in \{0, 1\}$ ($0 \le i < m$). The superscript of $\mathcal{T}^a$ indicates that it is a definition for the agents. Similar to vector $\mathcal{T}$, for each agent i in the PMMP, it fulfills the equation that

$$\mathcal{T}^a[\mathrm{i}] = \begin{cases} 0, \sum_{j=0}^{\mathrm{n}-1}\overline{\mathrm{Q}}[\mathrm{i},\ \mathrm{j}] = 0 \\ 1, \sum_{\mathrm{j}=0}^{\mathrm{n}-1}\overline{\mathrm{Q}}[\mathrm{i},\ \mathrm{j}] > 0 \end{cases}.$$

Note: $\mathcal{T}^a[i] = 1$ indicates that *agent i* is willing to play one of its candidate role(s) and 0 means no.

Definition 9: The *group performance* [22] σ is defined as the sum of the assigned agents' qualifications, i.e.,

$$\sigma = \sum_{i=0}^{m-1}\sum_{j=0}^{n-1} Q[i,\ j] \times T[i,\ j].$$

Definition 10: Given A, R, Q, and L, the QGRA model [1] is to find a matrix T to obtain

$$\min\ \ \sigma^{\text{QGRA}} = \sum_{i=0}^{m-1}\sum_{j=0}^{n-1} Q[i,\ j] \times T[i,\ j]$$

subject to

$$T[i,\ j]\ \in\ \{0,\ 1\}\ (i \in A,\ j \in R) \tag{1}$$

$$\sum_{j=0}^{n-1} T[i,\ j] \le 1\ \ (i \in A) \tag{2}$$

$$\sum_{i=0}^{m-1} T[i,\ j] \le L[j]\ \ \ (i \in A) \tag{3}$$

$$\sum_{i=0}^{m-1}\sum_{j=0}^{n-1} T[i,\ j] = \min\left\{\sum_{j=0}^{n-1}\mathcal{T}[j],\ \ \sum_{i=0}^{m-1}\mathcal{T}^a[i]\right\} \quad (i \in A,\ \ \ j \in R), \tag{4}$$

where σ^{QGRA} signifies the minimum cumulative cost of the crowdsourcing workers to complete their tasks throughout the entire scheduling process obtained by the QGRA model. Expression (1) is a 0-1 constraint; (2) indicates that each agent can play at most one role at the same time; (3) ensures that unworkable roles exist during the assignment process; and (4) represents a lower limit on the number of crowdsourcing workers assigned to carry out crowdsourcing tasks in the QGRA model.

In the following section, we will use a real-world scenario to shed light on the limitations of the QGRA model and, based on this scenario, introduce the proposed QGRAAS model.

III. Real-World Scenario

Company X, dedicated to providing dispatch optimization strategies, has recently won a bid on a PMMP project (Fig. 1) from Company Y in Guangzhou, China, a service provider for electronic maps. The goal of this project is to utilize photographs taken by crowdsourced workers to supplement the information about those new businesses on the map, including aspects such as contact information and operating hours.

Note: In Fig. 1, the blue icons represent the crowdsourcing tasks, with the corresponding numerical values below indicating the profit generated upon completion of each task, denominated in Chinese yuan.

After consulting with the project manager at Company Y, Ann, the chief executive officer (CEO) of Company X, identified two principal requirements for the firm. First, in consideration of the self-service nature of crowdsourcing tasks, it is

Fig. 1. Partial view diagram of the PMMP project.

TABLE I
VITAL PARAMETERS IN PMMP

Parameter	Definition	Value/Range
t_{start}	The start time of the assignment scheme.	10:00 AM
t_{end}	The end time of the assignment scheme.	18:00 PM
$t_{interval}$	The time interval for re-assignment.	0.5 h
w_{max}	The maximum number of tasks a worker can accept at a time.	5
u_{task}	The number of new tasks introduced during the reassignment phases.	[0, 200]
u_{worker}	The number of new workers introduced during the reassignment phases.	[0, 30]
p_{task}	The pricing for crowdsourced tasks, units in Chinese dollars.	(0, 20]

necessary to explore strategies that could enhance the participation levels of crowdsourcing workers, thereby boosting task completion rates, by at least 90%. Second, given the remote geographic locations of some tasks and the periodic influx of new tasks, the mere focus on enhancing completion rates could inadvertently escalate the overall operational costs. Hence, a judicious balance between task completion rates and total expenditures is essential.

To tackle these challenges, Ann asks Bob, the chief technology officer (CTO) of Company X, to accomplish this project. Through on-the-spot investigation and consultation with the operation and maintenance manager of Company Y, Bob lists the vital parameters (Table I) in the PMMP project. Furthermore, Bob procures the initial data pertaining to crowdsourcing tasks and workers from the open dataset furnished by Company Y. He subsequently evaluates the cost associated with all workers completing tasks based on distances as illustrated in Table II.

Meanwhile, Bob observes that the services provided by Company Y primarily cover three cities: Guangzhou, Dongguan, and Shenzhen in China. These cities are adjacent to each other. Based on the longitude and latitude information of these three cities, Bob infers that the locations of newly generated tasks and members fall within the range of 112.96°E to 114.63°E in longitude and 22.44°N to 23.94°N in latitude. Based on the aforementioned location information and the parameters in Table I,

TABLE II
INITIAL EVALUATION OF THE WORKERS ON TASKS

Worker[a]	Task[a]						
	Task 1	*Task* 2	*Task* 3	…	*Task* 851	*Task* 851	*Task* 853
Worker 1	52.40	39.43	50.06	…	21.12	15.97	61.49
Worker 2	1.96	12.35	0.97	…	29.94	57.62	106.65
Worker 3	95.20	82.82	92.71	…	64.68	45.65	25.41
…	…	…	…	…	…	…	…
Worker 1861	110.06	97.23	107.65	…	78.89	56.59	30.47
Worker 1862	14.23	5.56	13.55	…	21.74	47.07	102.36
Worker 1863	98.70	87.30	96.10	…	69.83	54.89	13.29

[a]For display purposes, only the first and last three workers and the first and last three tasks are shown. The evaluation method is depicted in Definition 4, and the unit of values in Table II is kilometers.

Bob collects the influx of new crowdsourcing workers and new tasks into the platform. Starting from time t_{start} to t_{end}, at intervals of $t_{interval}$, 17 new batches were introduced: [19, 14, 28, 9, 18, 12, 10, 0, 12, 3, 14, 6, 10, 5, 21, 22, and 29] (in total, 232 new workers) and [44, 131, 146, 96, 98, 143, 84, 174, 37, 18, 165, 189, 160, 159, 117, 59, and 0] (in total, 1820 new tasks).

Through reviewing the latest research in the related field, Bob discerns that the QGRA model is an emerging and efficacious method to address SSC-related problems. Consequently, he decides to leverage the QGRA model and its AC algorithm (i.e., Algorithm 4 in [1]) for tackling this PMMP project and obtains the following results. The cumulative completion rate of the crowdsourcing tasks is 79.44%, and the total distance traveled by the crowdsourcing workers to fulfill the tasks is 4148.88 km. The computational time required for task assignment is 123.08 s, while the decision-makers incur a total expenditure of 20 211 yuan. Moreover, at each reassignment timestamp, the proportion of crowdsourcing workers satisfied with their assigned task assignments are as follows: [84.72%, 100%, 100%, 93.33%, 96.61%, 98.44%, 94.74%, 98.48%, 92%, 76%, 95.74%, 93.59%, 94.59%, 81.25%, 90.18%, 89.91%, and 97.03%]. Correspondingly, the overall satisfaction values of workers at each reassignment timestamp are [241.23, 18.48, 23.71, 6.23, 19.48, 27.53, 20.60, 12.86, 21.54, −143.39, −13.45, −2.96, −3.71, −64.34, −6.86, −14.58, and 20.24].

Upon evaluation, Bob discerned that the current assignment solution failed to align with Ann's expectations. First, during the reassignment scheduling at the tenth timestamp, the proportion of crowdsourcing workers satisfied with their assigned task assignments is merely 76%, coupled with an overall worker satisfaction value of -143.39. This underscores a critical issue a subset of workers is dissatisfied with their assigned tasks, further diminishing their propensity to engage in crowdsourcing activities. Second, the overall task completion rate achieved by the QGRA model is slightly below 80%, while the decision-maker aspires for the overall task completion rate to reach 90%. The lower overall task completion rate is likely attributed to improper task pricing. Therefore, it might be prudent to consider implementing subsidy policies to ameliorate task completion rates.

The aforementioned intricate issues pose a significant challenge for Bob in devising the scheduling strategy within the stipulated deadline. Fortuitously, these hurdles can be surmounted by augmenting the QGRA model with suitable adaptive properties. The ensuing sections of this article elucidate the specifics of our solutions, poised to facilitate Bob in the timely delivery of his project.

IV. QGRA With Agent Satisfactions Model

As elucidated in Section III, the QGRA model encounters two notable challenges that warrant further refinement for more effectively addressing SSC-related in dynamic crowdsourcing environments. The foremost challenge concerns how to further enhance the satisfaction of crowdsourcing workers ensuring their sustained engagement in executing tasks from the platform. Subsequently, the second challenge involves the introduction of appropriate incentive policies, enabling decision-makers to strike a balance between task completion rates and the expenses associated with crowdsourcing tasks.

In light of these challenges, we propose the QGRAAS method. Initially, to properly ensure the satisfaction of crowdsourcing workers, we introduce the satisfaction filtering algorithm (SFA). This algorithm aims to selectively pair suitable crowdsourcing tasks and workers. While maximizing task completion, the algorithm is designed to improve the profits of workers who accept assigned tasks, thereby boosting their overall satisfaction.

Additionally, to improve the overall task completion rate, we implemented a bonus incentive mechanism. Unlike existing bonus incentive schemes [12], [13] in crowdsourcing that primarily focus on task completion rates, considering the unique characteristics of self-service crowdsourcing, our bonus incentives are aimed at aiding decision-makers in striking a balanced tradeoff between cost and task completion rate, thereby achieving a satisfactory decision scheme, see Section V-B.

Utilizing the real-world scenario delineated in Section III as an exemplar, we proceed to explicate the QGRAAS model in detail. Since QGRAAS significantly evolves based on QGRA, the definitions of QGRA presented in Section II remain applicable to QGRAAS. For instance, within the QGRAAS framework, roles are characterized as clustered crowdsourcing tasks, ascertained through the role awareness methodology proposed in Jiang et al. [1], whereas the agents are crowdsourcing workers. Meanwhile, since roles are clustered tasks, the vector L in PMMP is $[1, 1, 1, \ldots, 1, 1, 1]$. The computation method for the Q matrix is consistent with the approach outlined in reference. To clarify the new contributions, all the definitions without citations are proposed in this work for the first time.

Definition 11: A *role task number vector* λ denotes a vector about the number of tasks encompassed within roles, and $\lambda[j] \in \{1, 2, 3, 4, 5\}$ $(j \in R)$.

Note: As per Definition 1 and Table I, roles comprise clustered tasks, with a maximum of five tasks in each role. Consequently, the range of values in vector λ is $\{1, 2, 3, 4, 5\}$. Taking the real-world scenario presented in Section III as an example, the initial vector $\lambda = [1, 1, 1, \ldots, 2, 4, 5]$.

Definition 12: A *task incentive bonus* ν represents the incentive bonus received by the crowdsourcing workers upon completion of each assigned individual task, and $\nu \in [0, 20]$, unit in Chinese yuan, i.e., the extra payment for executing a task.

Note: When crowdsourcing workers complete tasks, they receive additional rewards to enhance their loyalty to the crowdsourcing platform/application. As the pricing of the incentive bonus ν is typically not higher than the task's pricing, and the range of task pricing p_{task} as shown in Table I is $(0, 20]$, consequently, the range of ν is $[0, 20]$. The specific value for ν is determined based on the decision-makers' tradeoff between task costs and task completion rates. The detailed tradeoff method will be elucidated in the QGRAAS algorithm section in Section V.

Definition 13: A *role incentive bonus vector* $\mathcal{V}$ represents the vector of incentive bonuses for roles, and $\mathcal{V}[j] = \lambda[j] \times \nu$.

Note: The values within vector $\mathcal{V}$ represent the aggregate incentive bonuses earned by crowdsourcing workers upon the completion of clustered crowdsourcing tasks. Leveraging the real-world scenario in Section III as an example, the values in its corresponding vector $\mathcal{V}$ are $[1\nu, 1\nu, 1\nu, \ldots, 2\nu, 4\nu, 5\nu]$.

Definition 14: A *role profit vector* $\Diamond$ is an n-vector, and $\Diamond[j]$ ($\Diamond[j] \in \mathbb{R}^+$ and $j \in R$) expresses the total revenue obtained by an agent from completing role j. Additionally, the unit of $\Diamond[j]$ is yuan.

Note: In the PMMP, roles are the clustered tasks. We denote P_d as the predefined price set of clustered tasks in role j. Thereby, the element in the vector $\Diamond$ can be formalized as $\Diamond[j] = \sum_{t=1}^{\lambda[j]} P_d$. In the real-world scenario of Section III, $\Diamond = [3.23, 8.35, 10.10, 7.89, 4.69, \ldots, 14.94, 10.45, 15.52, 23.55, 40.37]$.

Definition 15: An *agent cost matrix* [1] Ξ is an $m \times n$ matrix, where $\Xi[i, j]$ ($\Xi[i, j] \in \mathbb{R}^+$, $i \in A$, and $j \in R$) expresses the overhead for agent i to play role j.

Note: Ξ represents the cost when an agent performs a role. As detailed in Jiang et al. [1], Ξ represented as depicted in the following equation:

$$\Xi[i, j] = \begin{cases} 0, \; Q[i, j] \leq \varepsilon \\ \alpha \times (1 - e^{\varepsilon - Q[i, j]}) \times Q[i, j], \; Q[i, j] > \varepsilon \end{cases} \tag{5}$$

where ε represents a predefined neighborhood radius that signifies an agent's acceptable walking range, while α ($\alpha \in \mathbb{R}^+$) denotes the growth rate. The agent's overhead increases monotonically with the distance, but the increase eventually stabilizes [1]. To capture this relationship, we empirically employ an exponential function. To compare with the QGRA model, we maintain consistent parameter settings with reference, specifically setting $\alpha = 6$ and $\varepsilon = 1.5$ km.

Definition 16: An *agent income level vector* Λ^a is an m-vector, and $\Lambda^a[i]$ ($\Lambda^a[i] \in \mathbb{R}^+$ and $i \in A$) expresses the income level of agent i.

Note: Λ^a represents the average daily income in the cities where the agents are located. Section III mentions that the agents are located in three adjacent cities: Guangzhou, Dongguan, and Shenzhen. The average daily income for residents in Guangzhou, Dongguan, and Shenzhen is 281.9, 214.4, and 376.2 yuan, respectively [1]. In the real-world scenario, $\Lambda^a = [376.2, 214.4, 281.9, 281.9, 281.9, \ldots, 281.9, 281.9, 281.9, 214.4, 281.9]$.

Definition 17: An *agent satisfaction matrix* ψ is an $m \times n$ matrix, and $\psi[i, j]$ ($\psi[i, j] \in \mathbb{R}$, $i \in A$, $j \in R$) expresses the ratio between profit for agent i gained from completing role j and the income level of agent i.

Note: The physical meaning of ψ is to quantitatively assess the satisfaction level of agents in executing tasks. Following the ASE method in Jiang et al. [1], we can quantify ψ using as follows:

$$\psi[i, j] = \frac{(\Diamond[\mathrm{j}] + \mathcal{V}[\mathrm{j}] - \Xi\,[\mathrm{i, j}])}{\Lambda^a[i] \times \frac{t_{\text{interval}}}{t_{\text{work}}}}\ (i \in A,\ j \in R) \tag{6}$$

where t_{work} signifies local working hours, and t_{interval} represents the time agent i is taking to finish role j.

Definition 18: An *agent satisfaction threshold vector* [1] τ^a is an m-dimensional vector, and $\tau^a[i]$ ($\tau^a[i] \in [0, 1]$ and $i \in A$) expresses the acceptable threshold of agent i for the corresponding agent satisfaction value in ψ.

Note: The values in τ^a are precollected from agents and are used to quantify the willingness of agents to carry out the roles currently assigned to them. In the real-world scenario, $\tau^a = [0.1, 0.7, 1.0, \ldots, 0.9, 0.3, 0.1]$.

With the above definitions, the capability matrix $\overline{Q}$ in Definition 6 can be formalized as

$$\overline{Q}[i, j] = \begin{cases} 0, \psi[i, j] < \tau^a[i] \\ 1, \psi[i, j] \geq \tau^a[i] \end{cases}$$

where $\overline{Q}[i, j] = 1$ signifies that agent i exhibits willingness to assume role j, whereas 0 denotes the contrary. The matrix $\overline{Q}$ represents the quantification of agents' satisfaction after considering the incentive bonus for agents.

Building upon Definitions 6–8, once the capability matrix $\overline{Q}$ is determined, QGRA leverages it to quantify vectors *vector* $\mathcal{T}$ and $\mathcal{T}^a$. Thereby, QGRA selects the specific number of workers to play their assigned clustered tasks by the expression $\min\{\sum_{j=0}^{n-1} \mathcal{T}[j],\ \sum_{i=0}^{m-1} \mathcal{T}^a[i]\}$ $(i \in A, j \in R)$. Nevertheless, as new crowdsourcing tasks and workers are phased in periodically into the platform, this expression becomes less capable of accurately representing the number of workers whose satisfaction in task execution exceeds their expected values.

To elucidate this point, Fig. 2 serves as an illustrative example. Suppose there are four clustered crowdsourcing tasks (roles) and four workers (agents). Based on definitions 6–8, the value of $\min\{\sum_{j=0}^{n-1} \mathcal{T}[j],\ \sum_{i=0}^{m-1} \mathcal{T}^a[i]\} = 3$. This implies that three workers are finally assigned to complete their clustered crowdsourcing tasks. Due to Constraints (2) and (3) of the QGRA model, a one-to-one assignment relationship exists between crowdsourcing workers and clustered tasks. It indicates that at most one worker's satisfaction meets the expected value, while two workers' satisfaction falls short of expectation. In some cases, workers may even need to incur a deficit to complete the tasks. Such an assignment scheme adversely impacts the satisfaction levels of crowdsourcing workers, thereby eroding their loyalty and engagement to the platform.

$$\begin{array}{c} \text{Role} \\ \overline{Q}[i, j] = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix} \begin{array}{l} \rightarrow \sum_{i=0}^{m-1} \mathcal{T}^a[i] = 3 \\ \text{Agent} \\ \\ \\ \end{array} \\ \downarrow \\ \sum_{j=0}^{n-1} \mathcal{T}[j] = 3 \end{array}$$

Fig. 2. Simple case to illustrate the bottleneck of QGRA.

To ensure the satisfaction of crowdsourcing workers, we introduce the following new definitions and the SFA.

Definition 19: A *workable capability matrix* $\mathcal{M}$ is an $m \times n$ matrix, and $\mathcal{M}[i, j] \in \{0, 1\}$ $(i \in A, j \in R)$. $\mathcal{M}[i, j] = 1$ indicates that $\overline{Q}[i, j] = 1$ and agent i, selected by the SFA algorithm proposed below, is the most suitable agent for role j.

Note: One prominent feature that distinguishes QGRAAS from QGRA is the enhanced emphasis on the concept of "workability," which filters out suitable roles and agents. This filtering strategy is mainly embodied in the introduction of a workable capability matrix $\mathcal{M}$. To derive this matrix, we design the SFA, with the aim of maximizing the satisfaction of crowdsourcing workers while enhancing the task completion rate. The pseudocode for the SFA is shown in Algorithm 1. To facilitate a comprehensive exposition of Algorithm 1, we introduce certain symbols, newly defined, as follows.

1) $\mathcal{K}_{\text{pair}}$ denotes a tentative set of key-value pairs pertaining to role indices and agent indices, with agents serving as keys and roles as values. We elucidate this set through a rudimentary exemplar; suppose there exists a set $\mathcal{K}_{\text{pair}} = \{<1, 1>, <2, 4>\}$, where 1 and 2 are agent indices, and 1 and 4 are role indices; in this instance, $\mathcal{K}_{\text{pair}}\ [1] = 1$ and $\mathcal{K}_{\text{pair}}\ [2] = 4$.
2) $\mathcal{M}^{\text{t}}$ signifies a tentative $m \times n$ matrix, which is ultimately used to store the values of matrix $\mathcal{M}$.
3) j^t is a tentative variable, which is utilized to record the role index.
4) $\mho_j$ is a tentative variable, where $\mho_j = \sum_{i=0}^{m-1} \mathcal{M}^t[i, j]$ $(j \in R)$. The physical significance of the symbol $\mho_j$ represents the number of agents whose satisfaction in executing the current role j exceeds the expected satisfaction.

Lines 9–19 of Algorithm 1 illustrate the core idea of the SFA algorithm. SFA is not to select an agent with the highest satisfaction value for each role, but to maximize the satisfaction of each agent to the greatest extent possible. Based on the pseudocode of Algorithm 1, the time complexity of our proposed role awareness method is $O(m^2 n)$, where $m = |\mathcal{A}|$ and $n = |\mathcal{R}|$.

Employing SFA, we obtain the workable capability matrix $\mathcal{M}$ essential for the QGRAAS model. More specifically, we can select workable roles and agents (i.e., Definitions 21 and 23), thereby choosing the most suitable executable agents to perform these workable roles.

Definition 20: A *workable role* r_w is defined as a role that fulfills the conditions $r_w \in \mathcal{R}$ and $\sum_{i=0}^{m-1} \mathcal{M}[i, j] \geq 1$.

Definition 21: A *workable role set* $\mathcal{R}_w$ is defined as the set pertaining to r_w, and R_w denotes the cardinality of set $\mathcal{R}_w$.

Algorithm 1: Satisfaction Filtering Algorithm

Input:
- Roles capability matrix $\overline{Q}$
- Cardinality of agents m
- Cardinality of roles n
- Agent satisfaction matrix ψ

Output:
- Workable capability matrix $\mathcal{M}$

```
begin
  Initializing M as a zero matrix with dimensions m × n;
  M^t ← Q̄; /* Initialization */
  K_pair ← { };
  for i ← 0 to m−1 do
    for j ← 0 to n−1 do
      if Q̄[i, j] > 0 do
        if i in K_pair then
          j' ← K_pair[i]; /* Retrieving the index of the
            optimal role recorded for the current K_pair */
          if ψ[i, j'] < ψ[i, j] then
            M^t[i, j'] ← 0, K_pair[i] ← j;
          else if ψ[i, j'] > ψ[i, j] then
            M^t[i, j] ← 0;
          else
            Computing Ʊ_j' and Ʊ_j based on the defini-
            tion of the symbol Ʊ;
            if Ʊ_j' > Ʊ_j then
              M^t[i, j'] ← 0, K_pair[i] ← j;
            else
              M^t[i, j] ← 0;
        K_pair[i] ← j;
  M ← M^t;
  return M;
end
```

Definition 22: A *workable agent* a_w is defined as an agent that satisfies the conditions $a_w \in \mathcal{A}$ and $\sum_{j=0}^{n-1}\mathcal{M}[i,\ j] \ \geq \ 1$.

Definition 23: A *workable agent set* $\mathcal{A}_{\mathrm{w}}$ is defined as the set pertaining to a_w, and A_w denotes the cardinality of set $\mathcal{A}_{\mathrm{w}}$.

Definition 24: A *workable qualification matrix* Q_w expresses the suitability of a workable agent i $(0 \leq i < A_{\mathrm{w}})$ for a workable role j $(0 \leq j < R_{\mathrm{w}})$.

Note: The values of matrix Q_w under the initial data (before the crowdsourcing tasks are assigned) in the scenario is listed as follows:

$$Q_w = \begin{pmatrix} 63.98 & 62.82 & 73.98 & \dots & 17.64 & 59.67 & 59.65 \\ 31.63 & 24.56 & 40.82 & \dots & 38.56 & 104.73 & 99.08 \\ 106.88 & 106.29 & 116.52 & \dots & 59.93 & 24.35 & 35.68 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 119.37 & 119.53 & 128.60 & \dots & 72.85 & 30.45 & 45.65 \\ 25.68 & 21.46 & 36.04 & \dots & 28.07 & 100.45 & 96.79 \\ 113.54 & 112.12 & 123.52 & \dots & 66.90 & 11.93 & 23.41 \end{pmatrix}.$$

Fig. 3 serves as a simplified illustrative example to elucidate the relationship matrix Q_w in QGRAAS and matrix Q in QGRA. Let us consider a scenario comprising eight agents and four roles. Fig. 3 showcases the corresponding workable capability matrix $\mathcal{M}$, and qualification matrix Q. In accordance with Definitions 19–24, we can obtain the corresponding workable role set $\mathcal{R}_w$ and workable agent set $\mathcal{A}_{\mathrm{w}}$, which are respectively {Role 1, Role 3, Role 5} and {Agent 1, Agent 3, Agent 5, Agent 6, Agent 8}. Further, based on $\mathcal{R}_w$ and $\mathcal{A}_{\mathrm{w}}$, the qualification matrix Q is compressed to yield the workable one Q_w, as depicted in Fig. 3.

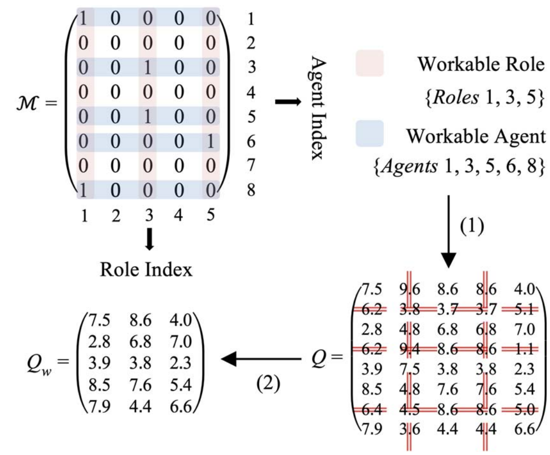


Fig. 3. Simple case to illustrate the procedure of generating the matrix Q_w.

Definition 25: A *workable role range vector* L_w is an R_w-dimensional row vector representing the number of workable agents required for each workable role, and $L_w[j] \in \mathbb{N}^+$ $(0 \leq j < R_w)$.

Note: In the QGRAAS model, the process of deriving the vector L_w leveraging the vector L from the QGRA model is nearly identical to the process of obtaining matrix Q_w from matrix Q. Refer to Fig. 3 for the detailed derivation process.

Definition 26: A *workable role assignment matrix* T_w is an $A_w \times R_w$ control variable matrix, where $T_w[i, j] \in \{0, 1\}$, signifying whether the workable agent i is assigned to the workable role j. Specifically, $T_w[i, j] = 1$ indicates an assignment, while $T_w[i, j] = 0$ denotes the absence of such.

Definition 27: Given R_w, A_w, Q_w, and L_w, the QGRAAS model is to find a T_w to obtain

$$\min \ \sigma^{\mathrm{QGRAAS}} = \sum_{i=0}^{A_w-1}\sum_{j=0}^{R_w-1} Q_w[i,\ j] \times T_w[i,\ j].$$

Subject to

$$T_w[i,\ j]\in\{0,\ 1\}\ (0 \leq i < A_w,\ \ 0 \leq j < R_w) \tag{7}$$

$$\sum_{i=0}^{A_w-1} T_w[i,\ j] = L_w[j] (0 \leq j < R_w) \tag{8}$$

$$\sum_{j=0}^{R_w-1} T_w[i,\ j] \leq 1\ (0 \leq i < A_w). \tag{9}$$

With the novel SFA method, we finally transform the PMMP into a revised GRA problem [17], [18], which is essentially an

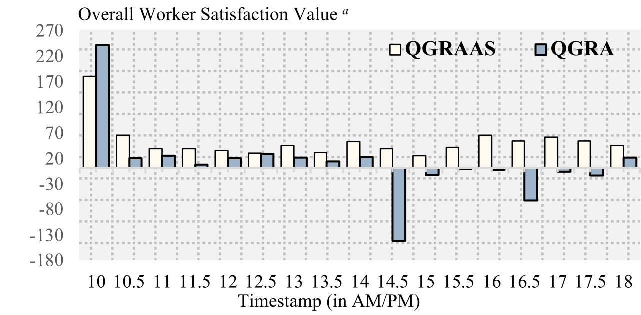


Fig. 4. Comparison of solution approaches in terms of overall worker satisfaction value. [a] The overall worker satisfaction value is equal to $\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\psi[i, j] \times T[i, j]$.

optimal matching problem of a weighted bipartite graph. From the modeling process described above, we confirm that transformation is nontrivial. Instead of utilizing the IBM ILOG CPLEX optimization package (CPLEX) [14], we can employ the Kuhn–Munkres (KM) Algorithm [8] to solve it, thereby further accelerating the solving process. Given that the time complexity of the KM algorithm is $O(m^3)$ ($m = |\mathcal{A}|$), the time complexity for solving the QGRAAS model is also $O(m^3)$.

V. Algorithms for QGRAAS

Similar to the QGRA model, the above QGRAAS model can only accomplish a single-time assignment. To accomplish the continuous and dynamic work as described in Section III, we propose to introduce the AC methodology [5], [6], [7] and realize adaptive scheduling of crowdsourcing tasks, i.e., Algorithm 2 in Section V-A. Simultaneously, to balance the relationship between task completion rate and total expenditure, we need to assist decision-makers in determining the value of the incentive bonus ν.

This section elucidates the two specific solution algorithms for the above-mentioned two challenges encountered during the modeling process of QGRAAS. First, akin to the QGRA model in Jiang et al. [1], we propose the AC algorithm corresponding to QGRAAS to achieve adaptive scheduling in the PMMP project. Second, we formulate the QGRAAS algorithm which employs an incentive bonus strategy to aid decision-makers in balancing task completion rates and overall expenses.

The noteworthy improvements in this work compared with the AC algorithms in [1] include 1) the phased introduction of new crowdsourcing tasks and workers; 2) the incorporation of incentive bonuses; and 3) the utilization of the SFA and QGRAAS model to enhance the overall satisfaction of crowdsourcing workers under the assigned scheme (Fig. 4).

A. Adaptive Collaboration

Owing to the inherent self-service attributes of SSC-related issues, every single assignment of crowdsourcing tasks may potentially leave some tasks uncompleted. Consequently, it is imperative to dynamically reassign new batches of crowdsourcing tasks to the crowdsourcing workers at each timestamp.

In contrast with the AC algorithm of the QGRA model, which is discussed in Jiang et al. [1], the crowdsourcing environment encompassed by the QGRAAS model is not static and

Algorithm 2: Adaptive Collaboration

Input:
- Set of crowdsourcing tasks $\mathcal{D}$
- Agent set $\mathcal{A}$
- Neighborhood radius ε
- Start time of the assignment scheme t_{start}
- End time of the assignment scheme t_{end}
- Time interval for re-assignment t_{interval}
- Task incentive bonus ν
- Agent income level Λ^a

Output:
- Overall group performance of QGRAAS $\overline{\sigma}^{\text{QGRAAS}}$
- Set of completed tasks $\mathcal{D}'^{\text{QGRAAS}}$
- Overall task set $\overline{\mathcal{D}}$
- Overall expenditure c^{QGRAAS}

begin
Initializing Γ as a zero vector;
$t_{\text{current}} \leftarrow t_{\text{start}}$, $\overline{\sigma}^{\text{QGRAAS}} \leftarrow 0$, $\mathcal{D}'^{\text{QGRAAS}} \leftarrow \emptyset$, $\overline{\mathcal{D}} \leftarrow \mathcal{D}$;
$\nu_{\text{current}} \leftarrow 0$; /* Initialization */
while $t_{\text{current}} \leq t_{\text{end}}$ **do**
  $\mathcal{R} \leftarrow$ roleAwareness($\mathcal{D}$, ε, Γ, $w_{\max}$);
  Evaluating Q, L, Ξ , $\lozenge$ with $\mathcal{R}$ and $\mathcal{A}$;
  if $t_{\text{current}} \neq t_{\text{start}}$ **then**
    $\nu_{\text{current}} \leftarrow \nu$;
  end if
  Calculating $\overline{Q}$ and ψ with ν_{current}, Λ^a, Ξ , $\lozenge$, $\mathcal{R}$ and $\mathcal{A}$;
  Reckoning $\mathcal{M}$ by Algorithm 1, $\overline{Q}$, and ψ;
  Determining R_w, A_w, Q_w, L_w by $\mathcal{M}$, $\mathcal{R}$, $\mathcal{A}$, Q and L;
  σ^{QGRAAS}, $T_w \leftarrow$ QGRAAS(R_w, A_w, Q_w, L_w);
  Deducing T^{QGRAAS} in reverse utilizing T_w based on the transformation relationship between Q and Q_w;
  Adding those completed tasks to $\mathcal{D}'^{\text{QGRAAS}}$ by T^{QGRAAS};
  $\mathcal{D} \leftarrow \mathcal{D} - \mathcal{D}'^{\text{QGRAAS}}$;
  $\overline{\sigma}^{\text{QGRAAS}} \leftarrow \overline{\sigma}^{\text{QGRAAS}} + \sigma^{\text{QGRAAS}}$;
  Updating the locations of the agents in $\mathcal{A}$;
  $t_{\text{current}} \leftarrow t_{\text{current}} + t_{\text{interval}}$;
  if $t_{\text{current}} \leq t_{\text{end}}$ **then**
    Generating new crowdsourcing tasks and crowdsourcing workers based on the parameter ranges in Table I, and updating the corresponding sets $\overline{\mathcal{D}}$, $\mathcal{D}$, and $\mathcal{A}$;
  end if
end while
compute c^{QGRAAS} through those finished tasks in $\mathcal{D}'^{\text{QGRAAS}}$;
return $\overline{\sigma}^{\text{QGRAAS}}$, $\mathcal{D}'^{\text{QGRAAS}}$, $\overline{\mathcal{D}}$, c^{QGRAAS};
end

accommodates the incorporation of new crowdsourcing tasks and workers. Furthermore, since the QGRAAS model incorporates a novel filtering process for roles (i.e., clustered crowdsourcing tasks) and agents (i.e., workers), it necessitates the establishment of index mapping between role r and workable role r_w during the elimination of completed roles as illustrated in Fig. 3. To these concerns, we design Algorithm 2 for the

QGRA model to achieve adaptive collaboration for the workers. To provide a comprehensive exposition of Algorithm 2, we introduce the following newly defined symbols.

1) $\mathcal{D}$ represents a set of crowdsourcing tasks.
2) $\mathcal{D}'^{[-]}$ represents those completed tasks of model [-] evaluated by Algorithm 2. For example, $\mathcal{D}'^{\text{QGRAAS}}$ is that of the QGRAAS model.
3) $\overline{\mathcal{D}}$ records all the crowdsourcing tasks, including the initial crowdsourcing tasks as well as the newly generated ones.
4) t_{current} is a tentative variable used to represent the current timestamp.
5) ν_{current} is a tentative variable utilized to represent the value of the current task incentive bonus.
6) Γ serves as an identifier vector, where $\Gamma[d] \in \{-1\} \cup \mathbb{N}^*$ $(0 \leq d < |\mathcal{D}|)$ signifies the corresponding cluster-ID associated with task d. $\Gamma[d] = -1$ is indicative of an isolated task. Initially, $\Gamma[d]$ is set to 0, denoting that task d is yet to be clustered.
7) roleAwareness($\mathcal{D}$, ε, Γ, and $w_{\max}$) represents the role awareness method mentioned in Jiang et al. [1], which is employed for extracting the set of roles $\mathcal{R}$ within the PMMP project.
8) QGRAAS(R_w, A_w, Q_w, and L_w) denotes the function of the QGRAAS model (i.e., Definition 27). This model is to be solved by the KM algorithm mentioned earlier.
9) $\overline{\sigma}^{[-]}$ stores the overall team performance of model [-] throughout the adaptive collaboration process, e.g., $\overline{\sigma}^{\text{QGRAAS}}$.
10) $c^{[-]}$ stores the overall expenditure associated with the tasks completed in $\mathcal{D}'^{[-]}$, as determined by solving the model [-].

Algorithm 2 delineates the pseudo-code devised for attaining adaptive collaboration through the QGRAAS model. The AC algorithm of the QGRAAS model differs from that of the QGRA model in three respects.

First, the QGRAAS model employs the SFA to filter agents and roles, resulting in the T_w matrix and T^{QGRAAS} matrix having different dimensions. Therefore, an inverse transformation is needed for the T_w matrix, by referencing the transformation between Q and Q_w as depicted in Fig. 3, to obtain the T^{QGRAAS} matrix. The T^{QGRAAS} matrix is subsequently employed to identify those completed tasks in correspondence with the assigned roles, i.e., lines 11–15 of Algorithm 2.

Second, we introduce the concept of an incentive bonus in the reassignment of tasks. We increase the profit for completing tasks that were not completed in the initial assignment and the newly introduced tasks, thereby enhancing the task completion rate, as described in lines 7–10 of Algorithm 2.

Third, it is vital to note that QGRAAS operates within a dynamic crowdsourcing environment, where new crowdsourcing tasks and workers are periodically introduced, as evidenced by lines 20–22 in Algorithm 2.

Given that the complexity of solving the QGRAAS model is $O(m^3)$, where m denotes the cardinality of set $\mathcal{A}$ (i.e., $|\mathcal{A}|$), it follows that the complexity of Algorithm 2 is also $O(m^3)$.

To evaluate the performance of the QGRAAS model relative to the QGRA model in addressing the real-world PMMP project

TABLE III
COMPARISON OF SOLUTIONS IN THE REAL-WORLD SCENARIO

Evaluation Metrics	Solution Approach	
	QGRA	QGRAAS
Task completion rate	**79.44%**	77.74%
Total expenditure for tasks	20 211 yuan	**19 628 yuan**
Total distance traveled by workers	4148.88 km	**1954.82 km**
Overall solution time	123.08 s	**10.95 s**

Note: The bold entries represent the superior results among the comparison approaches.

discussed in Section III, we make an assumption that the task incentive bonus ν is 0. At this point, the assignment results obtained by the QGRAAS model and its adaptive algorithm are illustrated in Table III and Fig. 4. Moreover, the satisfaction rates of task assignments to crowdsourcing workers in each reassignment timestamp obtained by QGRAAS are [98.82%, 100%, 100%, 100%, 100%, 100%, 98.75%, 100%, 100%, 100%, 100%, 98.59%, 100%, 100%, 100%, 100%, and 100%], respectively. Drawing upon the data presented in Section III, we ascertain that within the scope of scheduling tasks throughout the entire PMMP project, QGRAAS yields an overall worker satisfaction value of 884.90, whereas QGRA registers a corresponding value of 162.59.

Based on Table III and Fig. 4, it is obvious that, in the real-world scenario presented in Section II, QGRAAS is capable of significantly enhancing the satisfaction of crowdsourcing workers while sacrificing less than 2% of the task completion rate. This leads to a reduction in both the cost incurred by workers to complete tasks and the expenses of the decision-makers. Meanwhile, the solution time of QGRAAS is substantially reduced compared with that of QGRA. To further bolster the task completion rate, we will elaborate on how to employ an incentive bonus strategy in Section V-B, enabling decision-makers to strike a balance between task completion rate and overall expenses.

B. QGRAAS Algorithm

Table III demonstrates that, when the incentive bonus is set to zero, although the QGRAAS model inherently improves the satisfaction of crowdsourcing workers, the task completion rate still fails to reach a relatively high value. In response to this, we aim to stimulate the enthusiasm of crowdsourcing workers to complete tasks by increasing the incentive bonus, which in turn enhances the task completion rate. Concurrently, we provide corresponding results to aid decision-makers in striking a balance between expenses and task completion rates. Specifically, we introduce the QGRAAS algorithm to assist decision-makers in determining the value of the task incentive bonus ν. For a more comprehensive description of the QGRAAS algorithm, we introduce certain symbols that are newly defined, as follows.

1) η denotes the task completion rate for crowdsourcing tasks obtained through Algorithm 2, that is $\mathcal{D}'^{\text{QGRAAS}}/\overline{\mathcal{D}}$, where η ranges from 0 to 1.
2) ρ represents the task completion rate expected by the decision-makers. ρ ranges from 0 to 1, and this value is usually predetermined by the decision-makers. For instance, in the real-world scenario mentioned in Section III, the

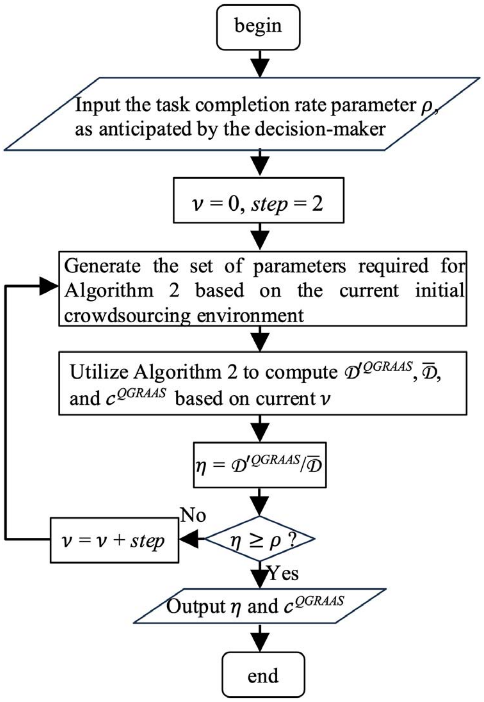


Fig. 5. Flowchart of the QGRAAS algorithm.

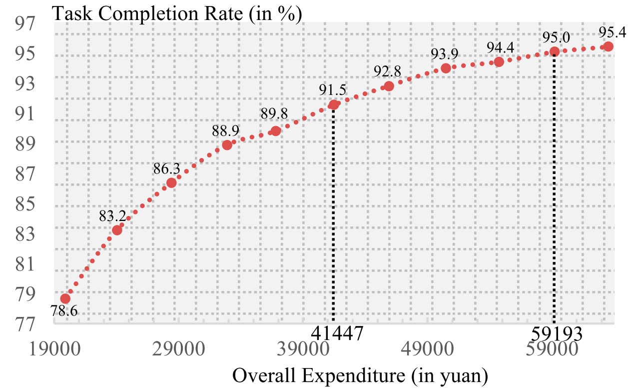


Fig. 6. Relationship between task completion rate and total expenditure as incentive bonus ν increases.

decision-maker expects a task completion rate of 90%, i.e., $\rho = 0.9$.

3) *step* is a value representing the step size in searching for the optimal value of ν, with $step \in (0, \nu_{\max}]$, where $\nu_{\max}$ represents the maximum value that ν can take. In the real-world scenario, $\nu_{\max} = 20$. Furthermore, we empirically set it to 2. Without loss of generality, we will conduct extensive randomized experiments in Section VI to validate the impact of different step sizes on the search results.

Fig. 5 elucidates the flowchart of the QGRAAS algorithm. Here, we primarily utilize a specific step size (i.e., $step = 2$) to search for an appropriate incentive bonus value ν that facilitates the overall completion rate of the crowdsourcing tasks in meeting the decision-makers' requirements. When $step = 2$ and $\rho = 0.9$, the QGRAAS algorithm yields a bonus value ν of 10 yuan. At this juncture, the overall task completion rate reaches 91.5%, the total expenditure amounts to 41 447 yuan, the total solution time is 14.73 s, the overall traveling distance of the workers to complete the tasks is 2832.48 km, and the overall satisfaction value of the workers measures 1710.83.

Simultaneously, Fig. 6 delineates the relationship between the overall cost and the task completion rate as the incentive bonus v increases. When the overall cost reaches 59 193 yuan, the overall task completion rate exceeds 95%. Therefore, decision-makers can use the numerical relationship between the task completion rate and overall cost to strike a balance and set an appropriate value for the incentive bonus.

C. *Computational Time Complexity Analysis of QGRAAS*

In this section, we present a detailed analysis of the computational complexity of the proposed QGRAAS algorithm, focusing on its time complexity. Our objective is to elucidate the scalability and efficiency of QGRAAS in handling large-scale SSC systems.

The computational time complexity of the QGRAAS algorithm is delineated through various fundamental segments, showcasing the procedural progression of the algorithm. At the outset, as delineated in Fig. 5, the algorithm incorporates an overarching loop predicated on the rate of task completion. Given the implementation of an incentive framework, assuming the incentive reward is sufficiently substantial to ensure compliance with the task completion benchmarks, this iterative process is inherently finite. Let k denote the iteration count within the QGRAAS algorithm's overarching loop, attributing a time complexity of $O(k)$ to this segment.

Proceeding to the inner workings, the quintessence of the QGRAAS algorithm lies in a dynamic collaborative mechanism (identified as Algorithm 2). As explicated in Section V-A, the computational effort required to resolve the dynamic collaborative mechanism mirrors that of the QGRAAS framework, both quantified as $O(m^3)$, where m signifies the size of the agent set (i.e., $|\mathcal{A}|$). Thus, encapsulating the analysis, the QGRAAS algorithm's computational time complexity is succinctly encapsulated as $O(km^3)$.

The computational time complexity analysis of QGRAAS has demonstrated its potential for efficient and real-time matching capabilities within large-scale systems. Furthermore, to assess the method's performance across various system sizes, we will conduct large-scale simulation experiments based on two real-world PMMP datasets, complemented by a data generation strategy that introduces random perturbations.

VI. Experimental Results

A. *Experimental Setup*

To showcase the viability and efficiency of our proposed method, we conduct large-scale random simulation experiments on a MacOS-13.4.1 laptop equipped with Apple M2 and 8GB LPDDR5, utilizing Python-3.8.16 programming on the Visual Studio Code platform.

Apart from the experimental parameters in Jiang et al. [1], the scale of the current experiment mainly depends on several

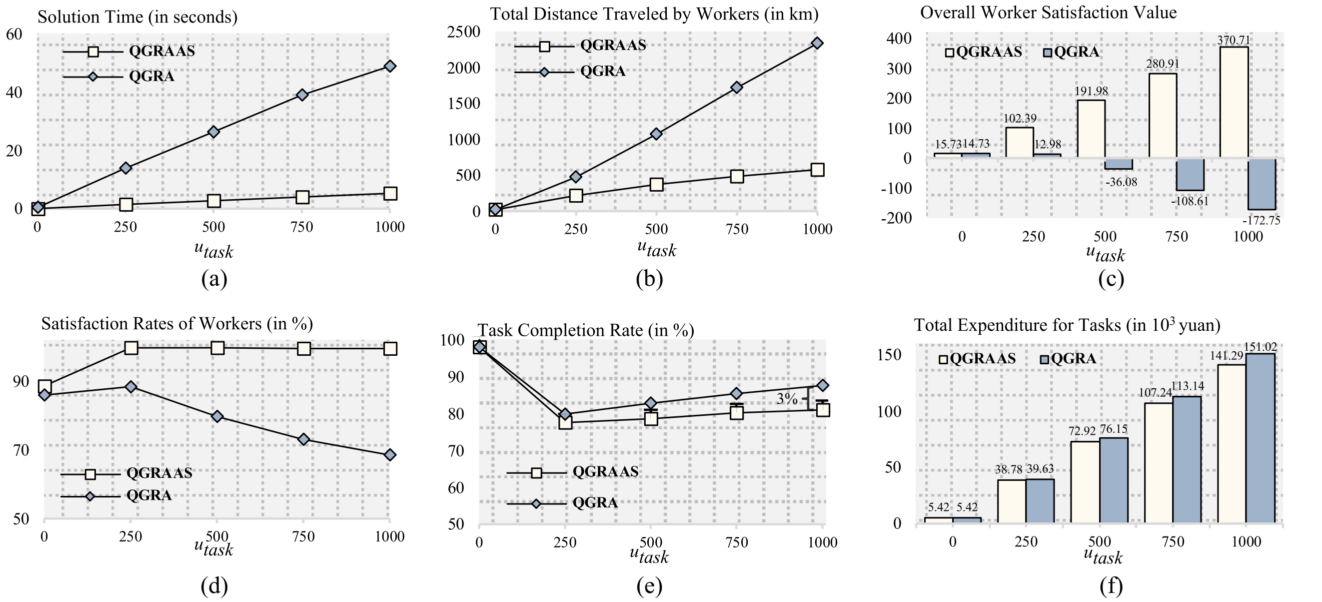


Fig. 7. Comparison of solution approaches under scenarios without an incentive bonus. Note: the physical meaning of u_{task} in the y-axis represents the number of new tasks introduced during the reassignment phases. In (d), the y-axis represents the proportion of workers assigned tasks whose profits exceed their own expectations (i.e., $i \in A$ and $\overline{Q}[i, j] = 1$) to the total number of workers. (a) Solution time (s). (b) Total distance traveled by workers (km). (c) Overall worker satisfaction value. (d) Satisfaction rates of workers (%). (e) Task completion rate (%). (f) Total expenditure for tasks (10^3 yuan).

parameters as follows. Referring to the related PMMP applications [3], [4], we set the ranges of u_{task} and u_{worker} as $\{0, 1, \ldots, 1000\}$ and $\{0, 1, \ldots, 50\}$, respectively. Additionally, as pointed out in Jiang et al. [1], when the task pricing reaches more than 20 yuan, the task completion rate can reach 100% in a stable crowdsourcing environment. As a reference, we set the maximum task price to 20 yuan, i.e., $p_{\text{task}} \in (0, 20]$. Furthermore, as previously defined, the range of the incentive bonus ν should not exceed the maximum value of product pricing, and the range of step should be less than ν. Therefore, we set the range of ν to $[0, 20]$ and the range of *step* to $(0, \nu]$.

Next, we will leverage the real-world scenario described in Section III as a foundation, and generate various scales of scenarios randomly based on the parameter ranges defined above. Subsequently, we will conduct hundreds of random experiments using the proposed methods to validate their reliability and practicality.

B. Performance Analysis of Comparative Models Under Scenarios Without an Incentive Bonus

First, we compare the performance of the QGRAAS model to the QGRA model under scenarios without an incentive bonus. Consequently, we assign the value of ν as 0. Considering the more pronounced influence of u_{task} relative to u_{worker}, we implement a strategy of variable control. Here, we fix the increased number of workers as a constant, taking an average value (i.e., $u_{\text{worker}} = 25$), while continually varying the number of additional crowdsourcing tasks. Under this setup, Fig. 7 compares the performance of the QGRAAS model and the QGRA model in solving the PMMP under various evaluation standards.

Fig. 7(a) presents the solution time for the PMMP using the QGRAAS and QGRA methods. It can be observed that, with the optimization of Algorithm 1 and the solution via the KM algorithm, the solution time of the QGRAAS model is lower than that of the QGRA model solved utilizing CPLEX. Moreover, the difference in solution speed between the two models gradually widens with the increase in scale.

Fig. 7(b) illustrates the total traveled distance of workers obtained by solving the PMMP using the QGRAAS and QGRA methods. It is obvious that the total distance resulting from the QGRAAS model is superior compared with the one obtained from the QGRA model, and as the number of tasks increases, the gap between the two widens.

Fig. 7(c) shows the overall worker satisfaction values obtained by different methods when the scale of new tasks increases. It becomes evident that the QGRA model is more suitable for a stable crowdsourcing environment. Once new tasks are introduced and their numbers increase, the resulting worker satisfaction values decrease continuously and may even present negative values. In contrast, the QGRAAS model adeptly adapts to dynamic crowdsourcing environments.

Fig. 7(d) and 7(e), respectively display the overall task completion rates of the crowdsourcing tasks and the corresponding total expenditures for decision-makers in the assignment solutions obtained by the QGRAAS and QGRA models.

Fig. 7(d) displays the corresponding proportion of worker satisfaction. It can be seen that the scheduling solution obtained by the QGRAAS model can reach approximately 100% as the number of newly generated tasks increases, while the scheduling solution obtained by the QGRA model decreases to below 69% with an increase in crowdsourcing tasks. Fig. 7(e) indicates that both the QGRA and QGRAAS models can achieve nearly a 100% task completion rate when no new crowdsourcing tasks are introduced. However, with the introduction of new tasks, the overall task completion rates obtained by these two models all

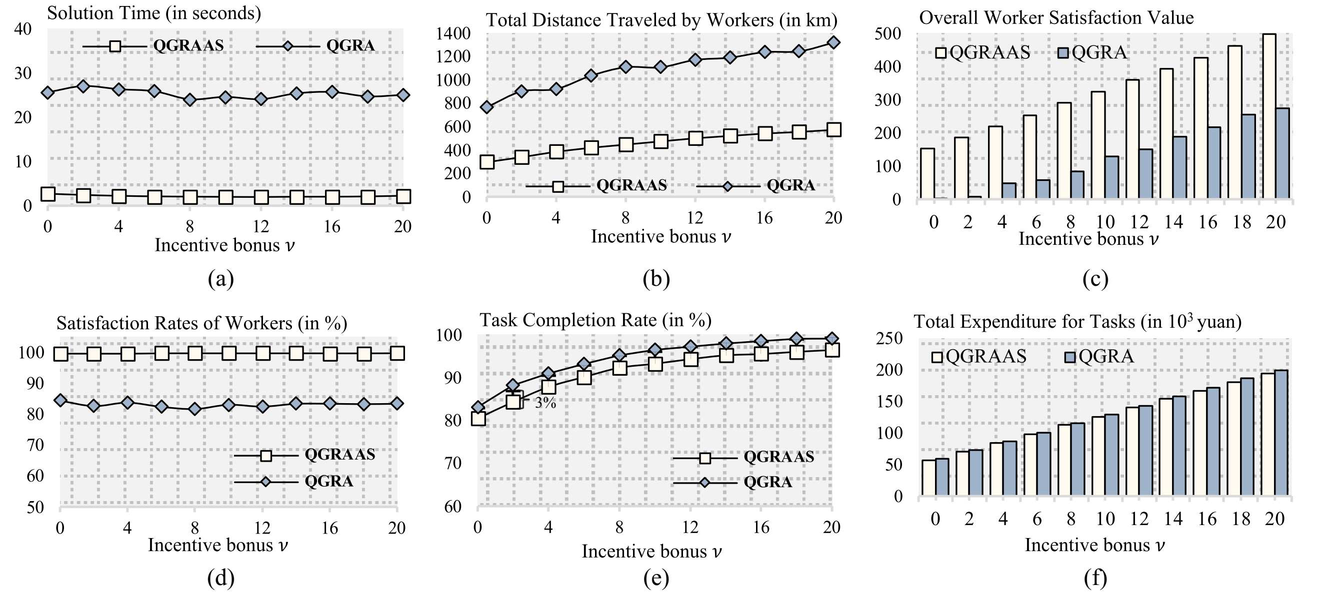


Fig. 8. Comparison of solution approaches under scenarios with an incentive bonus. (a) Solution time (s). (b) Total distance traveled by workers (km). (c) Overall worker satisfaction value. (d) Satisfaction rates of workers (%). (e) Task completion rate (%). (f) Total expenditure for tasks (10^3 yuan).

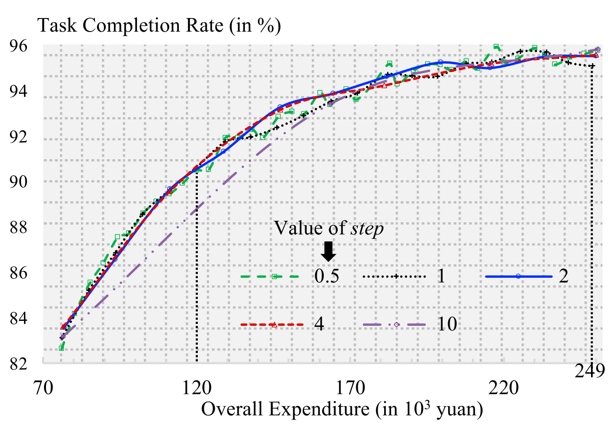


Fig. 9. Relationship between task completion rate and overall expenditure as incentive bonus ν increases under different *step* values.

decline. More pertinently, the QGRAAS model, To maintain a high level of worker satisfaction, sacrifices a task completion rate of merely 6% compared with the QGRA model. Simultaneously, the total expenditure in the solutions obtained by the QGRAAS model is lower than that of the QGRA model, and as the scale of new tasks increases, the gap in expenditure gradually widens.

C. Performance Analysis of Comparative Models Under Scenarios With an Incentive Bonus

We subsequently analyze the performance of the QGRAAS and QGRA methods following the incorporation of incentive bonuses. In this context, we randomly generate u_{task} and u_{worker} based on the parameter range defined in Section VI-A. To circumvent the impact of the final batch of new tasks on the overall task completion rate, we restrict the range of tasks generated in the final round to $\{0, 1, 2, \ldots, 250\}$. Also, for ease of comparison, we strategically fix the value of *step* at 2.

With these settings, Fig. 8 delineates the variation of different metrics with the increase in incentive bonuses for the compared methods. Combining Figs. 7 and 8, we can find that under both scenarios, with and without the introduction of the incentive bonus, the overall performances of the QGRAAS and QGRA models remain consistent.

With the introduction of incentive bonuses ν, the solution derived from the QGRA method continues to sacrifice the satisfaction of workers and increase the travel distance of crowdsourcing workers for a higher task completion rate [Fig. 8(b)–8(e)]. However, the QGRAAS method, at the cost of a marginal decrease (approximately 3%) in the task completion rate [Fig. 8(e)], yields an assignment scheme that almost completely satisfies worker expectations, all while maintaining a lower overall cost and faster solution speed [Fig. 8(a) and 8(f)].

D. Striking a Balance Between Overall Costs and Task Completion Rates

Ultimately, we examine the impact of different values of *step* on the QGRAAS method. As depicted in Section VI-A, we consider the range of *step* from 0 to 20. On this basis, we empirically select a few different *step* values that are divisible by 20, specifically *step* $\in \{0.5, 1, 2, 4, 10\}$, and scrutinize the performance of the QGRAAS method under these varying *step* values.

Fig. 9 illustrates the variations between the task completion rate and overall expenditure with respect to increasing incentive bonus ν under diverse *step* values. Fig. 9 reveals that varying values of *step* do not impact the solutions derived by the QGRAAS method. Conversely, with the increment in incentive bonus, the trend of the QGRAAS solutions exhibits consistency across different step values. Note that the fluctuation in the curves presented in Fig. 9 is due to the stochastic generation of new tasks and workers in each phase. Additionally, Fig. 9 demonstrates that, due to the remoteness of some task locations, increasing the crowdsourcing task completion rate from 90% to 95% necessitates an expenditure increase of more than double (from 120 000 to 249 000 yuan). Therefore, decision-makers need to select an appropriate level of expenditure, in accordance with their budget, to ensure a specified level of task completion rate.

E. Case Study for PMMP

Since [1] offers two different densities of the real-world crowdsourcing task datasets and a worker dataset for PMMP, we use them as the original datasets in the case study, i.e., ($|\mathcal{A}| = m = 1863$, $|\mathcal{D}| = 835$), and ($|\mathcal{A}| = m = 1863$, $|\mathcal{D}| = 2066$), where $|\mathcal{D}|$ represents the cardinality of the tasks. Meanwhile, we randomly generate new crowdsourcing tasks and workers to simulate datasets of varying scales and distributions. Both original datasets can be found on the website of the GitHub project provided in the appendix section.

In addition to comparing the QGRA model, we also compared the group multirole assignment (GMRA) and quasi-GMRA (QGMRA) models for M-M assignments mentioned in [1]. To better describe these models, we introduced the following definitions.

Definition 28: An *ability limit vector* [44] L^a is an m-vector, where $L^a[i]$ $(0 \le i < m)$ indicates the number of roles that can be assigned to agent i. The superscript of L^a indicates that L^a is a definition for the agents.

Note: When formalizing the PMMP via the GMRA model, for each agent i, it satisfies $L^a[i] = w_{\max}$, where $w_{\max}$ is the maximum number of tasks a worker can accept at a time.

Definition 29: Given $\mathcal{A}$ ($|\mathcal{A}|$=m), $\mathcal{R}$ ($|\mathcal{R}|$=n), Q, L, and L^a, the GMRA [1] model is to find a workable T to obtain

$$\min \ \sigma^{\text{GMRA}} = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} Q[i, j] \times T[i, j]. \tag{10}$$

Subject to (1), and

$$\sum_{i=0}^{m-1} T[i, \ j] = L[j] (0 \le j < \text{n}) \tag{11}$$

$$\sum_{j=0}^{n-1} T[i, \ j] \le L^a[i] \ (0 \le i < \text{m}) \tag{12}$$

where (11) ensures that each role has enough agents to be played, and (12) ensures that each agent can accept a limited number of roles at a time.

Definition 30: Given $\mathcal{A}$ ($|\mathcal{A}|$=m), $\mathcal{R}$ ($|\mathcal{R}|$=n), Q, L, L^a, $\mathcal{T}$, $\mathcal{T}^a$, the QGMRA [1] model is to find a workable T to obtain

$$\min \ \sigma^{\text{QGMRA}} = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} Q[i, j] \times T[i, j]. \tag{13}$$

Subject to (1), (3), (4), and (12).

Note: The GMRA and QGMRA models assume that roles are divided based on tasks as the smallest unit, rather than clustering nearby tasks into roles based on the role awareness method.

Figs. 10 and 11 showcase the performance of the compared algorithms on dense and sparse real-world datasets, respectively. It is evident that the QGRAAS algorithm's performance aligns with the results of previous experiments. As the scale of tasks increases, QGRAAS exhibits the fastest solution speed with the lowest growth rate. Additionally, it significantly reduces the total mileage for crowdsourcing workers to complete tasks. While maintaining a task completion rate of over 93%, it also ensures the satisfaction of each crowdsourcing worker as much as possible. This further corroborates the bottleneck of the QGRA depicted in Fig. 2, as well as the effectiveness of the SFA algorithm and incentive mechanisms we proposed. Still, Figs. 10(e) and 11(e) show that as the task scale increases, the task completion rate of the QGRAAS model tends to decrease to maintain high-level agent satisfaction. When decision-makers need to sacrifice a certain level of agent satisfaction to improve task completion rates, it may be necessary to balance these two aspects. This is also a potential direction for our future research.

From the experiment above, we can conclude that

1) Compared with QGRA, QGRAAS can rapidly solve for a proper assignment solution, which sacrifices a small degree of task completion rate but attempts to maximize the satisfaction of crowdsourcing workers.
2) The introduction of incentive bonuses can enhance the completion rate of crowdsourcing tasks to a certain extent, there are instances where the location of some tasks is relatively remote. As such, ensuring a high completion rate entails an extremely high cost. Consequently, decision-makers need to establish a reasonable expected task completion rate in line with their budget.
3) In all, the QGRAAS method represents a significant extension of the QGRA method to better solve PMMP-related SSC problems.

VII. Related Work

Spatiotemporal crowdsourcing (SC) [23], [24], [25] is an emerging computational paradigm that has garnered significant attention across various disciplines, including mobile crowdsourcing [26], [27], [28], event reporting [29], privacy preservation [30], and task allocation [31], [32], [33], among others [34], [35], [36], [37]. The focal point of this article is the application of SC within the sphere of task allocation.

There is a wealth of research in this area [29], [38], [39], [40], [41]. For instance, Hamrouni et al. [29] developed a generic spatial mobile crowdsourcing (SMCS) framework for event reporting that helps requesters recruit ideal reporters select highly relevant data from an evolving picture stream, and receive accurate responses. Xiao et al. [38] develop an incentive mechanism to ensure truthful reporting of answers and profiles by workers in mobile crowdsourcing, combining theoretical analysis with empirical validation to enhance data reliability and task assignment efficiency. Hamrouni et al. [39] introduce a novel framework for optimizing the recruitment and scheduling of workers in spatial mobile crowdsourcing, proven to improve the quality of the returned results by assigning suitable workers to tasks. Xu et al. [40] present a two-tiered social crowdsourcing architecture

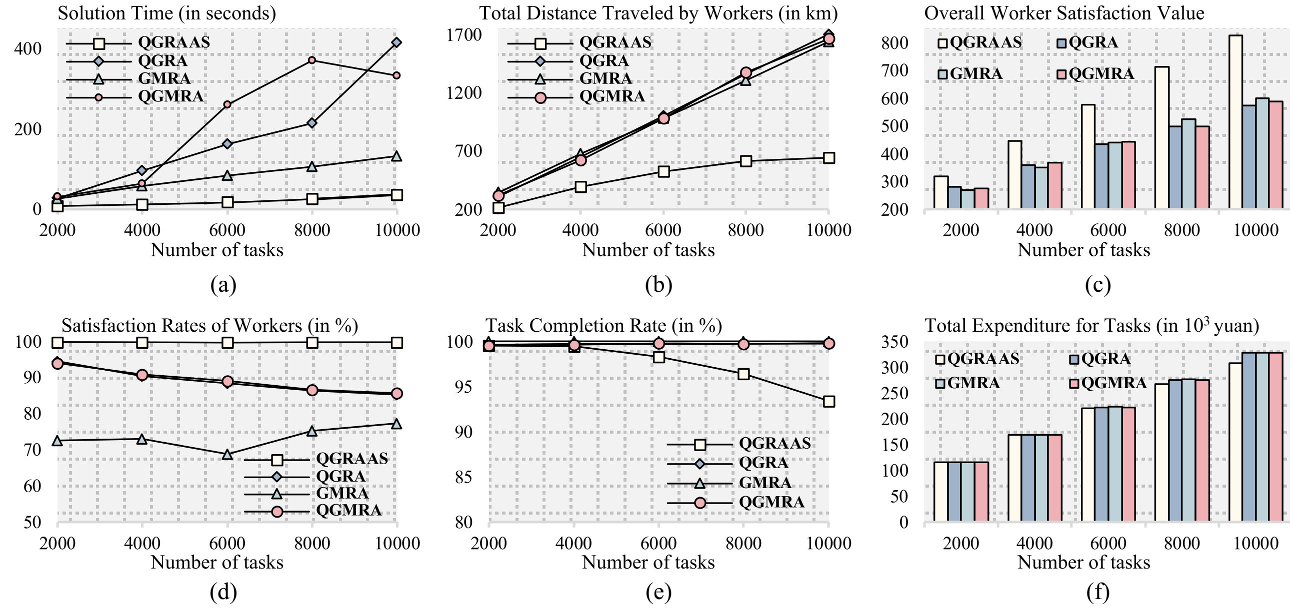


Fig. 10. Comparison of solution approaches under the dense real-world dataset with an incentive bonus. (a) Solution time (s). (b) Total distance traveled by workers (km). (c) Overall worker satisfaction value. (d) Satisfaction rates of workers (%). (e) Task completion rate (%). (f) Total expenditure for tasks (10^3 yuan).

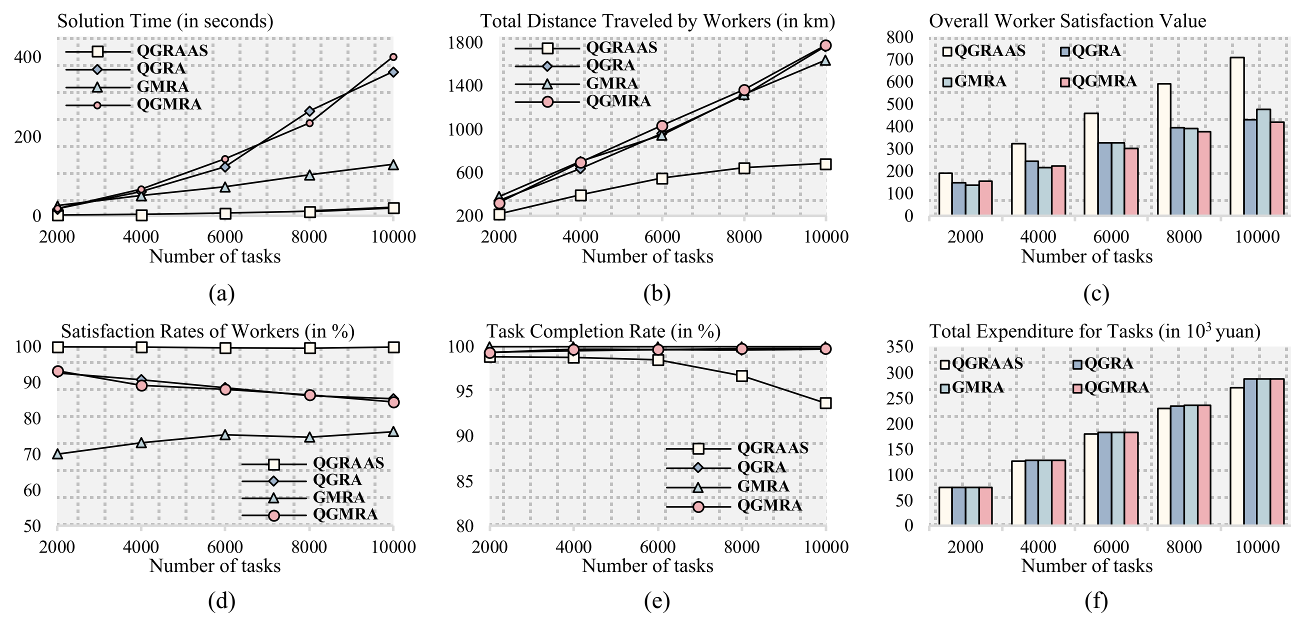


Fig. 11. Comparison of solution approaches under the sparse real-world dataset with an incentive bonus. (a) Solution time (s). (b) Total distance traveled by workers (km). (c) Overall worker satisfaction value. (d) Satisfaction rates of workers (%). (e) Task completion rate (%). (f) Total expenditure for tasks (10^3 yuan).

to solve the insufficient participation problem in online mobile crowdsourcing systems by enabling the selected registered users to recruit more users from their social circles. Jiang et al. [41] an innovative incentive mechanism for truth discovery in crowdsourcing environments with copiers, demonstrating its effectiveness through theoretical analysis and simulation validation.

The studies previously mentioned have yielded impressive outcomes within their relevant fields of research. However, they diverge from the SSC issue addressed in this work, specifically the PMMP. A notable difference is the indeterminate task granularity in PMMP, where the focus is not on singular tasks but rather on sets of tasks, or roles. To facilitate more effective role division, we leverage a role awareness method from QGRA, as mentioned in Definition 1. In the experimental section, by comparing QGRAAS with the M-M algorithms (namely, the GMRA and QGMRA algorithms), we validated the efficiency of our role awareness method.

Furthermore, in PMMP, agents may decline to perform roles, necessitating the consideration of agent satisfaction as a vital criterion for role execution. This involves a selection process to identify suitable agents for task performance. Nonetheless, the aforementioned studies provide invaluable guidance for revising

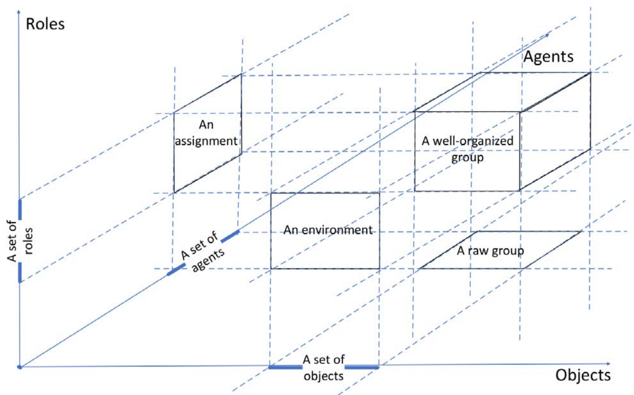


Fig. 12. 3-D view of the metamodel E-CARGO for QGRAAS [17].

this article and our future in-depth research, which is discussed in Section VIII.

Existing literature on SSC, notably PMMP, predominantly concentrates on task pricing [42] rather than task allocation. Considering the low task completion rate coupled with the prevailing competition dilemma [25] among workers, there is a critical need for a centrally orchestrated approach incorporating distributed assignment.

Recently, some scholars have tended to solve SC-related problems by role-based collaboration (RBC) methodology [1], [8], [21], [32], [43], [44], its E-CARGO [17] metamodel (see Fig. 12), and its submodels GRA/GMRA, which are notable for their centralized modeling and distributed execution. RBC and its derived models are designed to utilize roles as the foundational mechanism for optimizing collaborative activities, assigning the right roles to the right agents in a cooperative system. Due to its inherent interpretability, it has developed into a universal problem-solving paradigm to solve industry or engineering problems, e.g., team assignment problems [8], [21], [45], [46], deployment of multi-SUAV systems [22], [47], cloud computing problems [48], SSC problems [1], [32], [44], and others [49], [50], [51].

As an example, Liang et al. [44] expand the GMRA model to address the team assignment problem (TAP) in SC under high-order cardinality (HC) constraints. However, these proposed models are only equipped to handle one-time assignment problems, while tasks in SSC necessitate adaptive collaboration (AC) to reassign incomplete tasks. Furthermore, these methods operate under the assumption that the agent dutifully fulfills the role without considering the agent's satisfaction. More relevantly, Jiang et al. [1] propose the QGRA model to address SSC-related concerns. They first introduce the agent satisfaction evaluation (ASE) method to quantify workers' satisfaction. Concurrently, they establish the AC algorithm to assist QGRA in swiftly reassigning incomplete tasks. Relevant experimental evaluations confirm that their solution is capable of handling SSC-related issues.

However, the QGRA model faces two bottlenecks. First, it has not ensured the overall satisfaction of workers. Second, the QGRA model remains in low task completion rates in a dynamic crowdsourcing environment. These challenges led to the birth of the proposed QGRAAS model.

VIII. Conclusion and Future Work

In this article, we propose the QGRAAS model to address key limitations of the QGRA model for practically solving SSC-related problems: sacrificing a certain degree of worker satisfaction for a higher task completion rate and lack of performance for solving SSC-related problems under dynamic crowdsourcing environments. More pertinently, we first propose the SFA, a novel algorithm developed with the intent of maximizing worker satisfaction in crowdsourcing environments, while also improving the task completion rate. Thereafter, to the dynamic shifts within the crowdsourcing environment, we propose the principle of bonus incentives to enhance the task completion rate, while assisting decision-makers in fine-tuning the subtle balance between the costs and completion rates of crowdsourcing tasks.

In essence, our proposed solution constitutes a notable stride toward actualizing practical applications of the QGRA model to address SSC-related issues. In other words, the QGRAAS solution surpasses QGRA in multiple aspects, including improving worker engagement and satisfaction, a more rationalized bonus incentive strategy, and the facilitation of a mutually beneficial outcome for both decision-makers and crowdsourcing workers. These bring in much more benefit than QGRA did in the related industry.

Although the QGRAAS model demonstrates strong performance in self-serve crowdsourcing contexts like PMMP, the model and its corresponding algorithms require iterative adjustments and optimizations in response to evolving scenarios, diverse tasks, and decision-makers' demands within future industrial applications. Drawing from existing cutting-edge research, we identify several areas for extending our work. Here are the directions for future enhancements.

1) Introducing a review mechanism for crowdsourcing tasks on the platform side to increase settlement efficiency [29], [38].
2) Considering the execution sequence and trajectory length of task clusters as the clustering range increases, to determine the optimal task execution sequence [39].
3) Taking into account the social network relationships among crowdsourcing workers and considering the recommendation effect as part of the incentive mechanism [40].
4) Identifying copiers, and optimizing the incentive mechanism accordingly [41].
5) Considering the granularity of the incentive bonus, that is, assigning different incentive amounts according to the importance of the crowdsourcing tasks' locations to further reduce the decision-makers' costs.
6) Validating and optimizing the QGRAAS model under various dataset sizes, diverse tasks, and more generalized metrics for performance evaluation and satisfaction, aiming to enhance the universality of our model and its algorithms.
7) When decision-makers need to compromise agent satisfaction to enhance task completion rates, it becomes essential to balance these two aspects within the QGRAAS model.

Acknowledgment

The authors would like to thank the anonymous reviewers and the associate editor for their insightful comments that improved the presentation of this article.

Appendix

A. Relevant Utilized Datasets

The datasets utilized in case study are accessible in the public Github project of the authors: https://github.com/jiangqian1997/E-CARGO-Codes/tree/main/QGRA_datasets. The descriptions of these datasets are in a file named "README.txt".

References

[1] Q. Jiang, D. Liu, H. Zhu, Y. Qiao, and B. Huang, "Quasi group role assignment with role awareness in self-service spatiotemporal crowdsourcing," *IEEE Trans. Comput. Social Syst*, vol. 9, no. 5, pp. 1456–1468, Oct. 2022.
[2] Foap, "Foap." Accessed: May 10, 2023. [Online]. Available: https://www.foap.com/
[3] Meituan, "Meituan Paidian." Accessed: May 10, 2023. [Online]. Available: https://paidian.meituan.com/
[4] Antzb, "Antzb." Accessed: May 10, 2023. [Online]. Available: https://www.antzb.com/
[5] Y. Sheng, H. Zhu, X. Zhou, and W. Hu, "Effective approaches to adaptive collaboration via dynamic role assignment," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 46, no. 1, pp. 76–92, Jan. 2016.
[6] H. Zhu, "Adaptive collaboration systems," *IEEE Syst. Man, Cybern. Mag.*, vol. 1, no. 4, pp. 8–15, Oct. 2015.
[7] H. Zhu, M. C. Zhou, and M. Hou, "Adaptive collaboration based on the E-CARGO model," *Int. J. Agent Technol. Syst.*, vol. 4, no. 1, pp. 59–76, 2012.
[8] H. Zhu, D. Liu, S. Zhang, Y. Zhu, L. Teng, and S. Teng, "Solving the many to many assignment problem by improving the Kuhn-Munkres algorithm with backtracking," *Theor. Comput. Sci.*, vol. 618, pp. 30–41, Mar. 2016.
[9] H. Zhu, "Avoiding conflicts by group role assignment," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 46, no. 4, pp. 535–547, Apr. 2016.
[10] H. Zhu, "Avoiding critical members in a team by redundant assignment," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 50, no. 7, pp. 2729–2740, Jul. 2020.
[11] X. Zhu, H. Zhu, D. Liu, and X. Zhou, "Criteria making in role negotiation," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 50, no. 10, pp. 3731–3740, Oct. 2020.
[12] S. Shin, H. Choi, Y. Yi, and J. Ok, "Power of bonus in pricing for crowdsourcing," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 50, no. 1, pp. 43–44, Dec. 2022.
[13] M. Yin and Y. Chen, "Bonus or not? Learn to reward in crowdsourcing," in *Proc. Int. Joint Conf. Artif. Intell.*, Buenos Aires, Argentina, 2015, pp. 201–207.
[14] "IBM ILOG CPLEX optimization package." IBM. Accessed: May 30, 2023. [Online]. Available: http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/
[15] R. Wachowiak-Smolíková and H. Zhu, "Data analytics and visualization of adaptive collaboration simulations," *IEEE Trans. Comput. Social. Syst.*, vol. 10, no. 1, pp. 84–93, Feb. 2023.
[16] D. Liu, Y. Yuan, H. Zhu, S. Teng, and C. Huang, "Balance preferences with performance in group role assignment," *IEEE Trans. Cybern.*, vol. 48, no. 6, pp. 1800–1813, Jun. 2018.
[17] H. Zhu, *E-CARGO and Role-Based Collaboration: Modeling and Solving Problems in the Complex World.* Hoboken, NJ, USA: Wiley-IEEE Press, 2022.
[18] H. Zhu, "Pareto improvement: A GRA perspective," *IEEE Trans. Comput. Social Syst.*, vol. 10, no. 3, pp. 1241–1251, Jun. 2023.
[19] Q. Jiang, H. Zhu, Y. Qiao, D. Liu, and B. Huang, "Refugee resettlement by extending group multirole assignment," *IEEE Trans. Comput. Social Syst.*, vol. 10, no. 1, pp. 36–47, Feb. 2023.
[20] H. Zhu, "Maximizing group performance while minimizing budget," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 50, no. 2, pp. 633–645, Feb. 2020.
[21] D. Liu, B. Huang, and H. Zhu, "Solving the tree-structured task allocation problem via group multirole assignment," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 1, pp. 41–55, Jan. 2020.
[22] D. Liu, Q. Jiang, H. Zhu, and B. Huang, "Distributing UAVS as wireless repeaters in disaster relief via group role assignment," *Int. J. Coop. Inf. Syst.*, vol. 29, nos. 1–2, pp. 2040002–1–22, 2020.
[23] P. Michelucci and J. L. Dickinson, "The power of crowds," *Science*, vol. 351, no. 6268, pp. 32–33, 2016.
[24] E. Briscoe, S. Appling, and J. Schlosser, "Technology futures from passive crowdsourcing," *IEEE Trans. Comput. Social Syst.*, vol. 3, no. 1, pp. 23–31, Mar. 2016.
[25] J. Lu, Y. Xin, Z. Zhang, S. Tang, C. Tang, and S. Wan, "Extortion and cooperation in rating protocol design for competitive crowdsourcing," *IEEE Trans. Comput. Social Syst.*, vol. 8, no. 1, pp. 246–259, Feb. 2021.
[26] Y. Wang, Z. Cai, Z.-H. Zhan, Y.-J. Gong, and X. Tong, "An optimization and auction-based incentive mechanism to maximize social welfare for mobile crowdsourcing," *IEEE Trans. Comput. Social Syst.*, vol. 6, no. 3, pp. 414–429, Jun. 2019.
[27] Y. Wang, Z. Cai, Z.-H. Zhan, B. Zhao, X. Tong, and L. Qi, "Walrasian equilibrium-based multiobjective optimization for task allocation in mobile crowdsourcing," *IEEE Trans. Comput. Social Syst.*, vol. 7, no. 4, pp. 1033–1046, Aug. 2020.
[28] L. Wang, Z. Yu, D. Zhang, B. Guo, and C. H. Liu, "Heterogeneous multi-task assignment in mobile crowdsensing using spatiotemporal correlation," *IEEE Trans. Mobile Comput.*, vol. 18, no. 1, pp. 84–97, Jan. 2019.
[29] A. Hamrouni, H. Ghazzai, M. Frikha, and Y. Massoud, "A spatial mobile crowdsourcing framework for event reporting," *IEEE Trans. Comput. Soc. Syst.*, vol. 7, no. 2, pp. 477–491, Apr. 2020.
[30] X. Xu, Q. Liu, X. Zhang, J. Zhang, L. Qi, and W. Dou, "A blockchain-powered crowdsourcing method with privacy preservation in mobile environment," *IEEE Trans. Comput. Social Syst.*, vol. 6, no. 6, pp. 1407–1419, Dec. 2019.
[31] A. Tarable, A. Nordio, E. Leonardi, and M. A. Marsan, "The importance of worker reputation information in microtask-based crowd work systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 2, pp. 558–571, Feb. 2017.
[32] B. Huang, H. Zhu, D. Liu, N. Wu, Y. Qiao, and Q. Jiang, "Solving last-mile logistics problem in spatiotemporal crowdsourcing via role awareness with adaptive clustering," *IEEE Trans. Comput. Social Syst.*, vol. 8, no. 3, pp. 668–681, Jun. 2021.
[33] Y. Zheng, J. Wang, G. Li, R. Cheng, and J. Feng, "QASCA: A quality-aware task assignment system for crowdsourcing applications," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, New York, NY, USA, May 2015, pp. 1031–1046.
[34] C. Shan, N. Mamoulis, G. Li, R. Cheng, Z. Huang, and Y. Zheng, "A crowdsourcing framework for collecting tabular data," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 11, pp. 2060–2074, Nov. 2020.
[35] S. K. Mridha and M. Bhattacharyya, "Introducing collaboration in competitive crowdsourcing markets," *IEEE Intell. Syst.*, vol. 34, no. 1, pp. 23–31, Jan./Feb. 2019.
[36] L. Tran-Thanh, T. D. Huynh, A. Rosenfeld, S. D. Ramchurn, and N. R. Jennings, "Crowdsourcing complex workflows under budget constraints," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 1298–1304.
[37] L. Wang, Z. Yu, Q. Han, B. Guo, and H. Xiong, "Multi-objective optimization based allocation of heterogeneous spatial crowdsourcing tasks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 7, pp. 1637–1650, Jul. 2018.
[38] M. Xiao, W. Jin, C. Li, and M. Li, "Eliciting joint truthful answers and profiles from strategic workers in mobile crowdsourcing systems," *IEEE Trans. Mobile Comput.*, vol. 22, no. 8, pp. 4620–4633, Aug. 2023.
[39] A. Hamrouni, H. Ghazzai, and Y. Massoud, "Many-to-many recruitment and scheduling in spatial mobile crowdsourcing," *IEEE Access*, vol. 8, pp. 48707–48719, 2020.
[40] J. Xu, Z. Luo, C. Guan, D. Yang, L. Liu, and Y. Zhang, "Hiring a team from social network: Incentive mechanism design for two-tiered social mobile crowdsourcing," *IEEE Trans. Mobile Comput.*, vol. 22, no. 8, pp. 4664–4681, Aug. 2023.
[41] L. Jiang, X. Niu, J. Xu, D. Yang, and L. Xu, "Incentive mechanism design for truth discovery in crowdsourcing with copiers," *IEEE Trans. Services Comput.*, vol. 15, no. 5, pp. 2838–2853, Sep./Oct. 2022.
[42] Z. Li, Y. Li, W. Lu, and J. Huang, "Crowdsourcing logistics pricing optimization model based on DBSCAN clustering algorithm," *IEEE Access*, vol. 8, pp. 92615–92626, 2020.

[43] H. Zhu, "Why did Mr. Trump oppose globalization? An E-CARGO approach," *IEEE Trans. Comput. Social Syst.*, vol. 8, no. 6, pp. 1333–1343, Dec. 2021.

[44] L. Liang, J. Fu, H. Zhu, and D. Liu, "Solving the team allocation problem in crowdsourcing via group multirole assignment," *IEEE Trans. Comput. Social Syst.*, vol. 10, no. 3, pp. 843–854, Jun. 2023.

[45] H. Zhu, "Group role assignment with constraints (GRA+): A new category of assignment problems," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 53, no. 3, pp. 1670–1683, Mar. 2023.

[46] H. Zhu, "Agent categorization with group role assignment with constraints and simulated annealing," *IEEE Trans. Comput. Social Syst.*, vol. 7, no. 5, pp. 1234–1245, Oct. 2020.

[47] Q. Jiang, H. Zhu, Y. Qiao, Z. He, D. Liu, and B. Huang, "Agent evaluation in deployment of multi-SUAVs for communication recovery," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 52, no. 11, pp. 6968–6982, Nov. 2022.

[48] H. Ma, H. Zhu, K. Li, and W. Tang, "Collaborative optimization of service composition for data-intensive applications in a hybrid cloud," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 5, pp. 1022–1035, May 2019.

[49] Q. Jiang et al., "Iterative role negotiation via the bi-level GRA++ with decision tolerance," *IEEE Trans. Comput. Social Syst.*, Jun. 2024, early access, doi: 10.1109/TCSS.2024.3409893.

[50] Q. Jiang, H. Zhu, Y. Qiao, D. Liu, and B. Huang, "Extending group role assignment with cooperation and conflict factors via KD45 logic," *IEEE Trans. Comput. Social Syst.*, vol. 10, no. 1, pp. 178–191, 2023.

[51] Q. Jiang, D. Liu, H. Zhu, Y. Qiao, and B. Huang, "Equilibrium means equity? An E-CARGO perspective on the golden mean principle," *IEEE Trans. Comput. Social Syst.*, vol. 10, no. 4, pp. 1443–1454, Aug. 2023.

Qian Jiang is currently working toward the Ph.D. degree in intelligent science and system with Macao Institute of Systems Engineering, Macau University of Science and Technology, Macao, China.

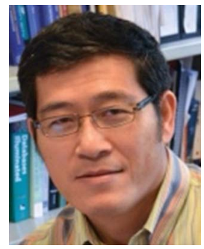
His research interests include human–machine systems, computational social simulation, and scheduling and optimization.

Dr. Jiang recently has published eight regular papers on IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, and *International Journal of Cooperative Information Systems*, respectively. He has served as a reviewer for IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING and IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS.

Dongning Liu (Senior Member, IEEE) received the Ph.D. degree in logic from Sun Yat-Sen University, Guangzhou, China, in 2007.

Since 2009, he has been a Full Professor with Guangdong University of Technology, Guangzhou, China. He is responsible for teaching artificial intelligent logic and discrete math with the School of Computer Science and Technology, Guangdong University of Technology. He is engaged in education and technology transfer on collaborative computing. He is a Vice Dean with the School of Computer Science and Technology. He has published more than 50 papers on computer magazines and international conferences.

Dr. Liu is a reviewer for IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, and IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He is a Member of IEEE SMC Society and serving as a TC Member of the technical committee of distributed intelligent systems, a Senior Member of CCF Society, and serving as a Vice Secretary-General and a Standing Committee Member of Technical Committee on Cooperative Computing of China Computer Federation.



Haibin Zhu (Senior Member, IEEE) received the B.Sc. degree in computer engineering from the Institute of Engineering and Technology, Changsha, China, in 1983, and the M.Sc. and Ph.D. degrees in computer science from the National University of Defense Technology (NUDT), Changsha, China, in 1988 and 1997, respectively.

He is a Full Professor and the Coordinator of the Computer Science Program, Montreal, Canada, the Founding Director of the Collaborative Systems Laboratory, Nipissing University, North Bay, ON, Canada, a Member of the University Budget Plan committee, Arts and Science Executive Committee, Nipissing University, Canada. He is also an Affiliate Professor with Concordia University, Montreal, Canada, and an Adjunct Professor of Laurentian University, Sudbury, Canada. He was the Chair with the Department of Computer Science and Mathematics, Nipissing University, from 2019 to 2021, a Visiting Professor and a Special Lecturer with the College of Computing Sciences, New Jersey Institute of Technology, Newark, NJ, USA, from 1999 to 2002, and a Lecturer, an Associate Professor, and a Full Professor with NUDT, from 1988 to 2000. He has accomplished (published or in press) over 230 research works including 30+ IEEE TRANSACTIONS articles, six books, five book chapters, four journal issues, and four conference proceedings.

Dr. Zhu is serving as a Member-at-Large of the Board of Governors and a Co-Chair of the technical committee of Distributed Intelligent Systems of IEEE Systems, Man and Cybernetics (SMC) Society (SMCS), the Editor-in-Chief of IEEE SMC Magazine, an Associate Editor (AE) of IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, *Frontiers of Computer Science*, and *IEEE Canada Review*. He was an Associate Vice President (AVP), Systems Science and Engineering (SSE) (2021), IEEE SMCS, a Program (Co-)Chair for many international conferences, and a PC Member for 130+ academic conferences. He is the Founding Researcher of Role-Based Collaboration and Adaptive Collaboration and the creator of the E-CARGO model. His research monograph E-CARGO and Role-Based Collaboration can be found at https://www.amazon.com/CARGO-Role-Based-Collaboration-Modeling-Problems/dp/1119693063.

Baoying Huang received the B.S. degree in network engineering and the M.S. degree in computer science and technology from the School of Computer Science and Technology, Guangdong University of Technology, Guangzhou, China, in 2016 and 2019, respectively. Currently, she is working toward the Ph.D. degree in intelligent science and system with the Institute of Systems Engineering, Macau University of Science and Technology, Macao, China.

Her research interests include scheduling and optimization, human–machine systems, computational social simulation, and semiconductor manufacturing systems.

Dr. Huang has published seven regular papers in IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, and IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, respectively. She has served as a reviewer for a number of journals and conferences.

Naiqi Wu (Fellow, IEEE) received the B.S. degree in electrical engineering from Anhui University of Technology, Huainan, China, in 1982, and the M.S. and Ph.D. degrees in systems engineering from Xi'an Jiaotong University, Xi'an, China, in 1985 and 1988, respectively.

From 1988 to 1995, he was with the Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China. From 1995 to 1998, he was with Shantou University, Shantou, China. He moved to the Guangdong University of Technology, Guangzhou, China, in 1998. He joined the Macau University of Science and Technology, Macau, China, in 2013. He is currently a Professor with the Institute of Systems Engineering, Macau University of Science and Technology. His research interests include production planning and scheduling, manufacturing system modeling and control, discrete event systems, Petri net theory and applications, intelligent transportation systems, and energy systems.

Yan Qiao (Senior Member, IEEE) received the B.S. and Ph.D. degrees in industrial engineering and mechanical engineering from Guangdong University of Technology, Guangzhou, China, in 2009 and 2015, respectively.

From September 2014 to September 2015, he was a Visiting Student with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, USA. From January 2016 to December 2017, he was a Postdoctoral Research Associate with the Institute of Systems Engineering, Macau University of Science and Technology, Macao, China. He is currently an Associate Professor with the Institute of Systems Engineering, Macau University of Science and Technology.

Dr. Qiao has over 100 publications, including one book chapter and 50+ regular papers in IEEE TRANSACTIONS. Besides, he was a recipient of QSI Best Application Paper Award Finalist of 2011 IEEE International Conference on Automation Science and Engineering, Best Student Paper Award of 2012 IEEE International Conference on Networking, Sensing and Control, Best Conference Paper Award Finalist of 2016 IEEE International Conference on Automation Science and Engineering, Best Student Paper Award Finalist of 2020 IEEE International Conference on Automation Science and Engineering, and 2021 Hsue-shen Tsien Paper Award of IEEE/CAA Journal of Automatica Sinica.