Solving Last-Mile Logistics Problem in Spatiotemporal Crowdsourcing via Role Awareness With Adaptive Clustering

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

***Abstract*— Last-mile logistics is a crucial phase of online commodity trades. In last-mile logistics, one of the critical problems is to reasonably assign couriers to distribute the products in time in order to ensure the quality of service, especially for fresh produce. The last-mile assignment problem (LMAP) for fresh produce poses a challenge on traditional logistics since fresh produce is difficult to preserve. This article formalizes the LMAP for fresh produce via the group role assignment framework and proposes a role awareness method by using adaptive clustering in spatiotemporal crowdsourcing based on task granularity. The formalization of LMAP makes it easy to find a solution using the IBM ILOG CPLEX optimization package (CPLEX). The proposed method allows one to take the time and space factor into consideration, helps spatiotemporal crowdsourcing assign couriers for efficient delivering daily orders, and improves the quality of service in last-mile logistics. It is verified by simulation experiments. The experimental results demonstrate the practicability of the proposed solutions in this article.**

***Index Terms*— Adaptive collaboration, group role assignment (GRA), IBM ILOG CPLEX optimization (CPLEX) package, k-means++ method, last-mile logistics, role-based collaboration (RBC), spatiotemporal crowdsourcing.**

Manuscript received July 16, 2020; revised November 15, 2020; accepted December 29, 2020. Date of publication February 15, 2021; date of current version May 28, 2021. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under Grant RGPIN2018-04818; in part by the National Natural Science Foundation of China under Grant 62072120, Grant 61972102, and Grant 61603100; in part by the Research and Development Program of Guangdong Province under Grant 2020B010166006, in part by the Natural Science Foundation of Guangdong Province under Grant 2019A1515011811; in part by the Science and Technology Development Fund (FDCT), Macau SAR, under Grant 0017/2019/A1; and in part by the Innovation for Defense Excellence and Security (IDEaS) Program from the Canadian Department of National Defense (DND) under Grant CFPMN2-051. Any opinions and conclusions in this work are strictly those of the author(s) and do not reflect the views, positions, or policies of—and are not endorsed by—IDEaS, DND, or the Government of Canada. *(Corresponding author: Dongning Liu.)*

Baoying Huang and Yan Qiao are with the Institute of Systems Engineering and Collaborative Laboratory for Intelligent Science and Systems, Macau University of Science and Technology, Cotai 999078, Macau (e-mail: baoyhuang@126.com; yqiao@must.edu.mo).

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

Dongning Liu and Qian Jiang are with the School of Computer Science and Technology, Guangdong University of Technology, Guangzhou 510006, China (e-mail: liudn@gdut.edu.cn; jiangqian@mail2.gdut.edu.cn).

Naiqi Wu is with the Institute of Systems Engineering and Collaborative Laboratory for Intelligent Science and Systems, Macau University of Science and Technology, Taipa, Macau 999078, and also the state key laboratory of precision electronic manufacturing technology and equipment, Guangdong University of Technology, Guangzhou 510006, China (e-mail: nqwu@must.edu.mo).

This article has supplementary downloadable material available at https://doi.org/10.1109/TCSS.2021.3051299, provided by the authors.

Digital Object Identifier 10.1109/TCSS.2021.3051299

I. Introduction

THE last-mile assignment problem (LMAP), i.e., the last link of distribution, is critical in the age of the rapidly developing e-commerce industry. Produce distribution in e-commerce is performed by a crowdsourcing platform that assigns orders of fresh produce to couriers for distribution. With high mix and low volume, how to assign daily orders of fresh produce to couriers is one of the most important problems in last-mile logistics. There are a number of factors that affect the quality of assignments, including the quality of perishable produce, delivery costs, and customer satisfaction.

First, the high time requirement for fresh produce makes the assignment difficult. Many companies resort to a half cold chain that uses cold chain cars between cities, and insulation cans or even foam boxes in last-mile distribution. In these cases, the quality of perishable produce decreases when the delivery time increase in the delivery process. Second, the hit rate, i.e., the number of successful deliveries divided by the total number of deliveries, is an important factor in logistics [13], [30]. A low hit rate not only decreases the quality of the fresh produce but also increases delivery costs. Third, the revenue of couriers is mainly dependent on the number of orders that they deliver. To accomplish more orders for high revenue, many couriers may send items to a collection station that is near the destination without the consent of customers, rather than send items to customers directly. Since customers cannot get products in time, the quality of items in the collection station cannot be guaranteed.

Thus, to ensure the quality of service in last-mile logistics, it is very important to make a reasonable assignment for the courier to accomplish the delivery tasks in time. In this article, we use an *order* to denote a requested delivery and an *agent* to denote a courier. We consider a *delivery task* performed in a trip by an *agent* as a *role*, which often consists of multiple orders. In real-world scenarios, the locations of orders in a delivery task should be near each other. Traditionally, couriers are only responsible for fulfilling deliveries in their assigned region regardless of the daily delivery situation. However, due to the required short time in product delivery, it is hardly feasible to adopt the traditional assignment method.

Therefore, it is necessary to adjust the delivery approach to flexibly assign delivery tasks according to the destinations and the request times of their orders. By doing so, we take the quality of service into account in the assignment and aim to obtain the best performance for the delivery team.

This article focuses on solving the LMAP of fresh produce in crowdsourcing platforms. By using an adaptive role awareness (ARA) method with role evaluation, this article allocates daily orders to couriers to achieve a high quality of services.

The LMAP of fresh produce can be solved using the group role assignment (GRA) [52]–[55]. GRA is an essential aspect of RA that is the kernel of role-based collaboration (RBC) [41], [53]–[55].

RBC is a collaboration methodology that uses roles to facilitate an organizational structure to collaborate and coordinate group activities with or within systems. Its life cycle consists of role negotiation, assignment, and execution [54].

Role negotiation is a basic part of RBC. Role negotiation involves two steps: role awareness and role specification [21]. By role awareness, we mean a process to reach an agreement of what and how roles are specified before role-playing. Role awareness, from the collaboration administrators' viewpoint, abstracts, and categorizes different kinds of tasks into roles that have different responsibilities. In this way, agents can collaborate based on roles. Role awareness can form different organizational structures among roles in collaboration, in which the roles have relationships with each other, i.e., mutual exclusion, prerequisites relationships. The organizational structure of roles significantly influences the efficiency of collaboration [20]. Role awareness can be used to accomplish role negotiation. This article achieves role awareness by an adaptive clustering method in LMAP. By using the adaptive clustering method, we gather orders that are close to each other in time and space to form roles. As a matter of fact, role awareness, a subprocess of role negotiation, is a new step of RBC.

GRA, an essential aspect of RBC, sets up a group by assigning roles to its members (agents) to achieve the highest group performance [55]. In this study, we extend the definition of GRA to allow a number of different roles to be assigned to each participating agent. Then, a new problem is formed, i.e., group multi-RA (GMRA) [52], [53].

Based on GMRA, this article focuses on solving the LMAP problem in crowdsourcing platforms by allocating daily orders to appropriate couriers in order to provide high-quality services. The contributions of this work include.

1) A concise formalization of the LMAP for fresh produce via GMRA is developed.
2) An ARA method by using a clustering method in spatiotemporal crowdsourcing based on task granularity is presented.
3) Practical solutions to the LMAP for fresh produce, i.e., the IBM ILOG CPLEX optimization package (CPLEX) solutions, are provided.

This article is organized as follows. It first describes a real-world scenario related to the proposed problem in Section II. Then, it formally specifies the LMAP problem with the revised environments-classes, agents, roles, groups, and objects (E-CARGO) model in Section III. Section IV presents an ARA method. Using simulation experiments, we verify that the solutions are practical in Section V. Section VI discusses the related work. This article concludes and points out future work in Section VII.

II. Real-World Scenario

In the last-mile logistics, the distribution stations provide delivery service to customers in about a one-mile area. Company X is responsible for delivering the fresh produce that the customers buy online one day in advance. The distribution stations receive fresh food from refrigeration vans in the morning and then deliver food through insulation cans. A crowdsourcing platform is responsible for assigning orders of fresh produce to couriers for distribution.

In general, express delivery for fresh produce is guaranteed within 48 h, but the delivery time is uncertain. Couriers usually deliver fresh produce every hour. However, the hit rate may be very low due to the absence of customers. Multiple deliveries for an order not only are a waste of human resources but also lead to the decline of food quality. Generally, customers usually have free time to receive deliveries at about noon when they take a break or 6:00 PM when they go off work. There may be more customers usually at 6:00 PM. However, in general, the reception time is uncertain.

To avoid a situation where customers are not present to receive deliveries, customers can specify not only the address but also the time that they are available to receive the deliveries. In general, fresh produce with shelf life has a strict time constraint in an insulation can. Therefore, to guarantee food quality, fresh produce should remain in the freezer of the distribution station until the order is near the request time.

Ann, the Chief Executive Officer (CEO), finds that the conventional distribution method in the crowdsourcing platform does not meet the tight time constraint demands for delivering the products. Thus, she asks Bob, the manager, to find a new assignment method that uses the information of distances between orders and their request time to efficiently and effectively allocate the daily delivery orders.

To better illustrate the scenario, Bob asks the department manager Beck to retrieve the data on daily delivery orders, including the destination and the request time for each order (Fig. 1). To simplify the problem, we use a coordinate range of 0 to 1 and use the period from 8:00 AM to 8:00 PM (Table I).

From Beck's experience, fresh food typically begins to go bad after the courier takes it out of the distribution station. Thus, couriers should send it to the customers as soon as possible. Meanwhile, it is convenient for couriers to complete orders that are close to each other and have a similar request time in one delivery task. A delivery task that often consists of multiple orders is set as a basic unit in the assignment. We should construct the delivery task to satisfy the timeliness requirement as much as possible by gathering similar daily orders into the same delivery task. However, to guarantee the timeliness requirement, this is not enough when there are too many similar orders in a delivery task. Moreover, with a fixed

TABLE I
DAILY ORDER

Orders	0	1	2	3	4	5	6	7	8	9
Destination	(0.2034, 0.8231)	(0.1956, 0.5485)	(0.6362, 0.9645)	(0.8653, 0.7112)	(0.8177, 0.4546)	(0.7155, 0.9224)	(0.2841, 0.5647)	(0.1111, 0.7654)	(0.5231, 0.5654)	(0.3902, 0.2272)
Time (o'clock)	19:34	19:56	17:02	19:40	9:30	13:55	10:41	13:55	14:30	18:00
Orders	10	11	12	13	14	15	16	17	18	19
Destination	(0.4563, 0.1422)	(0.9430, 0.4651)	(0.7681, 0.7808)	(0.8213, 0.3017)	(0.1234, 0.6323)	(0.1189, 0.3518)	(0.5674, 0.7768)	(0.3425, 0.6879)	(0.7683, 0.4654)	(0.8711, 0.2597)
Time (o'clock)	18:30	17:40	17:50	19:00	19:15	20:00	18:30	15:22	17:55	17:30

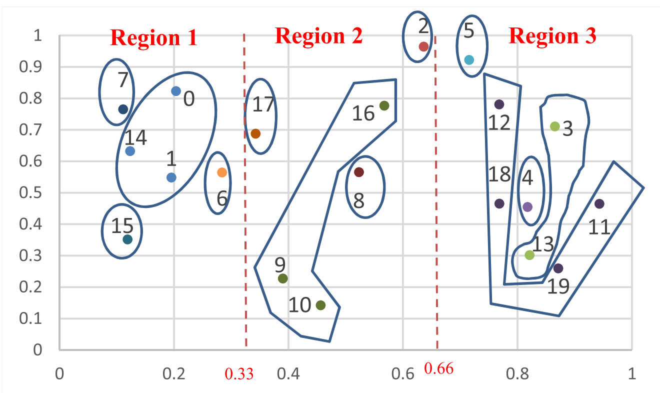


Fig. 1. Daily Orders.

TABLE II
LIMITED NUMBERS OF TASKS

Names	Adam	Brian	Chris
Limited Number of Tasks	8	6	8

capacity for an insulation can, the number of orders that a courier can fulfill in a delivery task is limited.

Ann further puts forward several reasonable requirements for allocating orders to couriers. First, orders that are assigned to a courier in a delivery task should usually be near each other and have a similar request delivery time. For example, Fig. 1 shows a batch of orders grouped based on region and request time. In Fig. 1, couriers deliver fresh produce every hour. The orders in an hour are clustered into delivery tasks based on region. A delivery region is divided into three parts. Each delivery task is shown as a circle in Fig. 1. Second, a courier cannot undertake different tasks that have time conflicts with each other. Third, each carrier (agent) has a different familiarity degree with each region, which has an impact on delivery efficiency. Last but not least, our ambition is to find the assignment that maximizes the performance of the delivery team consisting of several couriers. We aim to assign the appropriate couriers to delivery tasks so as to efficiently and effectively use human resources and improve service quality.

Bob asks Beck to evaluate employees according to their capabilities, experiences, and credits to show the qualification

TABLE III
QUALIFICATION OF CANDIDATES

Names	Adam	Brian	Chris
Personal Qualification	0.97	0.88	0.89

of couriers for each task (Tables III and IV) and determine the number of tasks each courier can fulfill in one day (Table II). Since the orders are made one day in advance, they have about 6–8 hours to make an assignment plan for the daily orders.

From the above-mentioned scenario, Ann and Bob follow the initial steps of GRA. Bob encounters an LMAP problem. The final optimized solution is shown (bolded and underlined) in Table IV, i.e., a tuple set as: $\{\langle$Adam, $\{2, 3, 6, 7, 10, 12\}\rangle$, $\langle$Brian, $\{4, 8, 9, 11\}\rangle$, $\langle$Chris, $\{1, 5\}\rangle\}$. The total sum of the assigned evaluation values is 8.01. It is worth mentioning that Brian's assignment may not be the best use of his talents, which is unlike the traditional task allocation method that assigns a person only based on his/her talents for one role. However, if we consider the overall team's performance, this assignment is optimal.

III. PROBLEM FORMALIZATIONS

To solve the above-mentioned problem, we need to solve the LMAP problem that is a variation of the GRA problem. Therefore, we initially describe the LMAP problem using the E-CARGO model. To be self-contained, we concisely describe the required concepts and definitions of the E-CARGO model. With the E-CARGO model [44], [45], [52]–[55], a system $\sum$ can be described as a nine-tuple $\sum ::= \langle \mathcal{C}, \mathcal{O}, \mathcal{A}, \mathcal{M}, \mathcal{R}, \mathcal{E}, \mathcal{G}, s_0, \mathcal{H}\rangle$, where $\mathcal{C}$ is a set of classes, $\mathcal{O}$ is a set of objects, $\mathcal{A}$ is a set of agents, $\mathcal{M}$ is a set of messages, $\mathcal{R}$ is a set of roles, $\mathcal{E}$ is a set of environments, $\mathcal{G}$ is a set of groups, s_0 is the initial state of the system, and $\mathcal{H}$ is a set of users. In such a system, $\mathcal{A}$ and $\mathcal{H}$, and $\mathcal{E}$ and $\mathcal{G}$ are tightly coupled sets. A human user and his/her agent perform a role together. Every group should work in an environment. An environment regulates a group.

In discussing RA problems [23], [44], [54], [55], environments (e) and groups (g) are simplified into vectors and matrices, respectively. Furthermore, we use nonnegative integers m $(= |\mathcal{A}|$, where $|\mathcal{A}|$ is the cardinality of set $\mathcal{A})$ to express the size of the agent set $\mathcal{A}$; n $(= |\mathcal{R}|)$ to express the size of the role set $\mathcal{R}$; $i, i_1, i_2, \ldots$ to express the indices of agents; and $j, j_1, j_2, \ldots$ to express the indices of roles. The

TABLE IV
CANDIDATES AND TASK EVALUATION

Tasks	1	2	3	4	5	6	7	8	9	10	11	12
Adam	0.75	**1.0**	**0.97**	0.82	0.88	**0.89**	**0.83**	0.51	0.08	**0.01**	0.00	**0.98**
Brian	0.90	0.80	0.76	**0.84**	0.79	0.80	0.82	**0.62**	**0.08**	0.01	**0.00**	0.70
Chris	**0.90**	0.68	0.77	0.78	**0.89**	0.80	0.79	0.61	0.07	0.01	0.00	0.69

Note that the evaluation value is shown with two decimal places. Task 1 is the order set of {4}. Task 2 is the order set of {6}. Task 3 is the order set of {7}. Task 4 is the order set of {5}. Task 5 is the order set of {8}. Task 6 is the order set of {17}. Task 7 is the order set of {2}. Task 8 is the order set of {11, 12, 18, 19}. Task 9 is the order set of {9, 10, 16}. Task 10 is the order set of {0, 1, 14}. Task 11 is the order set of {3, 13}. Task 12 is the order set of {15}

number of daily orders is set as N. According to the LMAP problem, we consider a courier as an agent and a delivery task performed in a trip by an agent as a role. A delivery task consists of a number of orders. From the problem discussed in Section II, we allow an agent to play more than one role, and roles played by an agent must have no time conflict with each other. Although an agent cannot play two roles concurrently, an agent can fulfill one or more different roles alternately within a period, i.e., a day.

Definition 1: A *role range vector* L [44], [52], [55] is a vector of the lower bound of the ranges of roles in the environment e of group g.

Note: L is an important component of the E-CARGO model. It indicates the minimum number of agents required for each role in facilitating a properly functional group. For the scenario in Section II, the cardinal number of L is the number of the delivery task. Orders are divided into delivery tasks that can be performed in one trip by an agent. Thus, a delivery task (role) can be undertaken by a single courier (agent), leading to that the value of each entry of L is equal to one, which indicates that it requires one courier for a task. For example, $L = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]$ in Section II.

Definition 2: An *ability limit vector* [52] L^a is an m-vector, where $L^a[i]$ $(0 \leq i < m)$ indicates how many roles can be assigned to agent i at most.

For example, based on Table II, for the scenario given in Section II, it can be described as $L^a = [8, 6, 8]$.

Definition 3: A *personal qualification vector* Q_p [44], [54], [55] is an m-vector, where $Q_p[i] \in [0, 1]$ expresses the qualification value of agent $i \in \mathcal{N}$ $(0 \leq i < m)$. $Q_p[i] = 0$ indicates the lowest value and 1 the highest.

Note that, personal qualification is affected by many factors such as service quality, experience, and proficiency. Here, we use the historical scores to evaluate it (Table III). It can be described as $Q_p = [0.97, 0.88, 0.89]$ for the case given in Table III. In particular, Q_p is designed to evaluate a *qualification matrix* Q described as follows.

Definition 4: A *qualification matrix* Q [44], [55] is an $m \times n$ matrix, where $Q[i, j] \in [0, 1]$ expresses the qualification value of agent i $(0 \leq i < m)$ for role j $(0 \leq j < n)$.

The *qualification matrix* Q is an important evaluation for assignment. For the case given in Table IV, we have the corresponding Q shown in Fig. 2.

However, the evaluation of Q obtained just based on historical data is not enough. Q can be influenced by the qualification of roles, personal qualification Q_p, and familiarity of agents on roles. The personal qualification Q_p can be obtained from the human resource department. The qualification of roles is mainly affected by role structure, which specifies the relationships among roles in a group. Different daily orders can cluster into different delivery tasks (roles) using role awareness. Different role structures have different qualifications of roles and different familiarity degrees of agents on roles in last-mile logistics. Thus, to evaluate Q in different role structures, we further consider the formation of Q in Section IV.

$$\begin{bmatrix} 0.75 & 1.00 & 0.97 & 0.82 & 0.88 & 0.89 & 0.83 & 0.51 & 0.08 & 0.01 & 0.00 & 0.98 \\ 0.90 & 0.80 & 0.76 & 0.84 & 0.79 & 0.80 & 0.82 & 0.62 & 0.08 & 0.01 & 0.00 & 0.70 \\ 0.90 & 0.68 & 0.77 & 0.78 & 0.89 & 0.80 & 0.79 & 0.61 & 0.07 & 0.01 & 0.00 & 0.69 \end{bmatrix}$$

Fig. 2. Qualification matrix Q.

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

Fig. 3. R^c matrix.

Definition 5: A *conflicting role matrix* is defined as an $n \times n$ matrix R^c [44], [45], [54] $(R^c[j_1, j_2] \in \{0, 1\}, 0 \leq j_1, j_2 < n)$, where $R^c[j_1, j_2] = 1$ expresses that role j_1 is in conflict with role j_2 and 0, otherwise.

Note: R^c is a symmetric matrix along the diagonal from $R^c[0, 0]$ to $R^c[n-1, n-1]$, i.e., $R^c[j_1, j_2] = R^c[j_2, j_1]$ $(0 \leq j_1, j_2 < n, j_1 \neq j_2)$.

In LMAP, conflicts are mainly reflected in time requests between roles. This is constructed following role awareness. For example, for the case given in Fig. 1, task 10 (role 10) includes the orders of 0, 1, and 14, and *task* 11 (role 11) is composed of the orders of 3 and 13. Since the delivery time for *task* 10 is from 19:15 to 19:56, and for *task* 11 is from 19:00 to 19:40 (Table I), the duration of *task* 10 overlaps with *task* 11. Thus, *task* 10 has a conflict with *task* 11. For the tasks in Fig. 1, we have the corresponding R^c shown in Fig. 3.

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

Fig. 4. T matrix.

Definition 6: An *RA matrix* T [44], [45], [54], [55] is defined as an $m \times n$ matrix, where $T[i, j] \in \{0, 1\}$ $(0 \leq i < m, 0 \leq j < n)$ indicates whether agent i is assigned to role j or not. $T[i, j] = 1$ means yes and 0 means no.

For example, Fig. 4 shows a T for the scenario given in Section II.

Definition 7: Role j is *workable* [44], [55] in group g if it has been assigned enough agents, that is

$$\sum_{i=0}^{m-1} T[i, j] \geq L[j].$$

Definition 8: T is *workable* [44], [55] if each role j is workable, i.e., $\sum_{i=0}^{m-1} T[i, j] \geq L[j] (0 \leq j < n)$. Group g is *workable* if T is workable.

Definition 9: The group performance σ [44], [55] *of group* g is defined as the sum of the assigned agents' qualifications, that is,

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

From the above-mentioned definitions, group g can be expressed by Q, L, L^a, R^c, and T. Thus, the proposed LMAP problem can be formalized in a concise way as presented in Definition 10.

Definition 10: The LMAP for fresh produce via GRA aims to find a T such that

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

$$\text{s.t. } T[i, j] \in \{0, 1\} \quad (0 \leq i < m, 0 \leq j < n) \tag{1}$$

$$\sum_{i=0}^{m-1} T[i, j] = L[j] \quad (0 \leq j < n) \tag{2}$$

$$\sum_{j=0}^{n-1} T[i, j] \leq L^a[i] \quad (0 \leq i < m) \tag{3}$$

$$R^c[j_1, j_2] \times (T[i, j_1] + T[i, j_2]) \leq 1 \quad (0 \leq i < m, 0 \leq j_1, j_2 < n) \tag{4}$$

where expression (1) is a 0-1 constraint; (2) guarantees that each role should be assigned by following the roles' cardinality requirements; (3) assigns an agent to a limited number of roles; and (4) guarantees that roles undertaken by one agent cannot have conflicts with each other.

Therefore, from this formalization, the LMAP problem becomes a typical extended-integer linear programming problem, i.e., zero-one linear programming [31], [38], [44], [55]. Then, it can be solved using an optimization tool widely accepted in the industry, such as IBM ILOG CPLEX optimization studio (CPLEX). The performance experiments are based on the platform shown in Table V.

TABLE V
CONFIGURATION OF THE EXPERIMENTAL PLATFORM

Hardware	
CPU	Intel Core i7-7700K CPU @4.20GHz 4.20 GHz
MM	8GB
Software	
OS	Windows 10 Enterprise
Eclipse	4.2.0
JDK	Java 8 Update

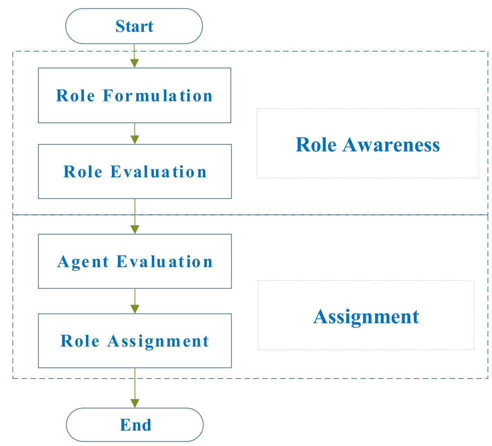


Fig. 5. Flowchart of the solution.

Up to now, we have an optimal assignment solution for the LMAP problem for the case given in Fig. 1. The total sum of the assigned evaluation values is 8.01. Fig. 1 provides an example of role structures created by task division by clustering orders into tasks based on region and request time. However, daily delivery orders are different every day. Different task divisions form different role structures that affect the performance of the delivery. Therefore, it is critical to partition the daily orders adaptively. Following the task division, the qualification matrix Q and the conflicting role matrix R^c could be determined accordingly. Finally, an optimal solution for LMAP can be obtained based on *Definition* 10.

Therefore, to achieve a high team performance, this article proposes an ARA method to create a reasonable role structure via adaptive task division and evaluation. Then, we present a method that reasonably assigns couriers to deliver daily orders in order to effectively allocate resources to improve service quality. (Fig. 5) The assignment part in Fig. 5 is based on *Definition 10*, where the agent evaluation is reflected in the personal qualification vector Q_p.

IV. ADAPTIVE ROLE AWARENESS

Role negotiation and its role awareness are basic parts of RBC. In LMAP, role awareness aims to make a reasonable task granularity. Task granularity has two meanings.

The first one refers to the level of detail planning for a task. Task allocation usually adopts two algorithmic models: 1) a coarse-grained bipartite matching model that assigns a crowdsourcing task for each participant in a period and 2) a fine-grained-task-planning-based model that allocates each participant to undertake multiple crowdsourcing tasks with complete execution order and the path to perform these tasks in a period [36]. In LMAP, each participant would undertake crowdsourcing tasks related to each other, which looks like a fine-grained-task-planning-based model. However, because of the short distance in a delivery process, task granularity could increase in LMAP. In this article, we assign related tasks to couriers without route planning, which allows couriers to flexibly complete tasks according to the circumstance of short distances. The assignment aims to assign tasks to couriers in order to achieve optimal performance and reduce the difficulty of execution.

The second meaning of task granularity is the scope of a task in terms of time and space. Since a delivery task is performed at one time by an agent, the task granularity in LMAP means the scope of time and space of the task for each courier in a delivery mission. The time of a task usually means the delivery time in distribution. The space of a task means the distance between orders in the task. A task is difficult for a courier to accomplish when the task granularity becomes too large. If the range of time and space of a role (or a delivery task after clustering) is large, the agents playing that role would be overloaded. Meanwhile, the complexity of task allocation may increase when granularity becomes too small. The orders with similar destinations and time may not cluster into the same delivery task, which causes low resource utilization. An adaptive method [34], [50] is used in this article to implement role awareness in order to determine appropriate task granularity.

Role awareness mainly includes role formulation and role evaluation, where role formulation determines role structure [20] in collaboration, and role evaluation assigns a value to each role in the role formulation. It constructs the roles that have a relationship with each other, i.e., mutual exclusion, prerequisites relationships. In collaboration, role structure has a great impact on the effectiveness of assignments. Role awareness is closely related to the situation of daily orders in LMAP. Since different role structures for the same daily order have different impacts on assignment, ARA is required for LMAP. To evaluate different effects of role structures in the adaptive process, role evaluation is used to evaluate the qualification of roles based on factors that impact performance after role formulation. In LMAP, role formulation is accomplished by task division. Using task division, similar orders can be clustered into a delivery task (role).

In order to determine what kind of role is the goal of ARA in LMAP, we first propose a method to evaluate each role following role formulation.

A. Role Evaluation and Qualification Matrix Q

Role evaluation is to assign each role a value in [0, 1] after role formulation. In the ARA, the evaluation method is used to assess the performance of the service based on different role structures. Role evaluation needs to consider many factors to guarantee that the values assigned to roles reflect the real-world value.

According to the problem in Section II, the following are some of the factors that can impact the quality of role formulation: 1) the capacity of insulation can. It influences the number of orders in a delivery task (role). It is better to fully fill the insulation can in a delivery task. However, too many orders may decline the quality of fresh produce when the insulation can is overload. 2) The shelf life of fresh products in last-mile logistics. It means that the delivery time should be as short as possible for a courier in a delivery task (role). 3) The convenience for couriers. It means that the distance between orders should be as short as possible for a courier in a delivery task (role).

Thus, the number of orders (x_j), time of orders (y_j), and distance between orders (z_j) for a delivery task (role) j should be taken into consideration to evaluate the quality of a delivery task (role). Based on the above requirements, the evaluation of a role can be described as follows.

Definition 11: A *role evaluation vector* E is an n-vector, where $E[j]$ indicates the qualification of role j with the factors of capacity (x_j), time (y_j), and distance (z_j)

$$\begin{aligned} E[j] &= f(x_j, y_j, z_j) \\ &= A \exp\left(-\frac{1}{2}\left(\frac{(x_j-\mu_x)^2}{\delta_x^2}+\frac{(y_j-\mu_y)^2}{\delta_y^2}+\frac{(z_j-\mu_z)^2}{\delta_z^2}\right)\right) \end{aligned}$$

where $A = 1/((2\pi)^{3/2}\delta_x\delta_y\delta_z)$, $x_j \in [1, N_o]$, $y_j \in [0, 720]$, and $z_j \in [0, \sqrt{2}]$. Note that N_0 is the capacity of the insulation can. A range of 0 to 720 minutes is used to represent the period from 8:00 AM to 8:00 PM. The longest distance is set as $\sqrt{2}$ km. The evaluation function is very close to but different from the three-dimension normal distribution. The definition mainly expresses that the farther from the expectation value μ_x, μ_y, and μ_z are, the lower the value of the task j is. However, we can use the characteristics of the normal distribution to understand the evaluation function and determine its parameters.

Parameters μ_x, μ_y, μ_z, δ_x^2, δ_y^2, and δ_z^2 can be set based on the characteristics of three-dimension normal distribution. Assume that $\mathcal{N}(\mu_x, \delta_x^2)$, $\mathcal{N}(\mu_y, \delta_y^2)$, and $\mathcal{N}(\mu_z, \delta_z^2)$ obey normal distribution. Thus, $X = (x_j\text{-}\mu_x)/\delta_x \sim \mathcal{N}(0, 1)$, $Y = (y_j - \mu_y)/\delta_y \sim \mathcal{N}(0, 1)$, and $Z = (z_j\text{-}\mu_z)/\delta_z \sim \mathcal{N}(0, 1)$. Parameters μ_x, μ_y, and μ_z represent mathematical expectation. Here, parameter μ_x is set as an expected value of the number of orders in a task for a courier. The parameter μ_y is set as an expected value of time range and μ_z is set as an expected value of distance in a task for a courier to ensure the quality of fresh products. Similarly, parameters δ_x^2, δ_y^2, and δ_z^2 represent the respective variance. Based on the characteristics of the normal distribution, the value of δ is

$$\delta = (x - \mu)/\varphi^{-1}(B) \tag{5}$$

where

$$\varphi((x - \mu)/\delta) = P(X \leq (x - \mu)/\delta) = B.$$

B is a constant that represents the probability. $\varphi^{-1}(B)$ is the inverse distribution function of $\varphi(B)$ and $B \in (0, 1)$.

In evaluation, the value of $f(x_j, y_j, z_j)$ no longer represents the probability, but an evaluation value in [0, 1]. It mainly expresses that the farther from the expectation value μ_x, μ_y, and μ_z, the lower the value of the task j. To design the evaluation function, we have to further determine the parameters δ_x^2, δ_y^2, and δ_z^2 based on the expression (5). We suppose that an agent may be made overload when the number of orders (x_j) in role j is bigger than a value x. For example, in $\mathcal{N}(\mu_x, \delta_x^2)$, we assume that the probability of variable x_j, i.e., $P((x_j - \mu_x)/\delta_x > (x - \mu_x)/\delta_x)$, should be small. Since the probability reflects the evaluation value, the assumption reflects that too many orders could decline the quality of fresh produce. It is known that there is a low probability for x_j when $P((x_j - \mu_x)/\delta_x > (x - \mu_x)/\delta_x) = 0.01$, i.e., $P((x_j - \mu_x)/\delta_x \leq (x - \mu_x)/\delta_x) = 1 - P((x_j - \mu_x)/\delta_x > (x - \mu_x)/\delta_x) = 0.99$. Thus, parameters δ_x^2, δ_y^2, and δ_z^2 can be determined based on the expression (5) when B is equal to 0.99.

For example, we suppose that the expected value of the number of orders for a delivery task is ten. There is a high damage risk of overload when the number of orders is over 20, where customer satisfaction is very low, and the evaluation value is very close to 0. Thus, for normal distribution, the value at δ_x is $\delta_x = (x - \mu_x)/\varphi^{-1}(0.99) = (20 - 10)/2.36 \approx 4.237$.

Through the lookup table, we know $\varphi^{-1}(0.99) = 2.36$. Similarly, δ_y and δ_z can also be found in this way. We suppose that the expected value of the time for a delivery task is one minute. The damage risk is high when the time for a delivery task is over 30 min. Parameter δ_y can be set as 12.29. Parameter δ_z is 0.42 when the expected distance in a delivery task is zero and the excessive distance in a delivery task is over one mile.

From the process of RBC, our ARA includes a series of role negotiations, agent evaluations, and RAs and finds the best role structure for the final assignment. The qualification matrix Q is influenced by the role evaluation and agent evaluation significantly. Based on *Definition 4*, Q can be influenced by the evaluation values of roles, personal qualification Q_p, and familiarity of agents on roles. The familiarity of agents on roles can be described as follows.

Definition 12: A *familiarity matrix F* is an $m \times n$ matrix, where $F[i, j] \in [0, 1]$ expresses the familiarity degree of agent i ($0 \leq i < m$) for a task j ($0 \leq j < n$).

For example, to evaluate the familiarity, we record the top three familiar positions of agent i, i.e., $P_i(p_{1i}, p_{2i}, p_{3i})$, where p_{1i}, p_{2i}, and p_{3i} are tuples of the abscissa and ordinate axes for the position. The familiarity degree of position p_{1i}, p_{2i}, and p_{3i} for agent i is $W_i(\omega_{1i}, \omega_{2i}, \omega_{3i})$, where ω_{1i}, ω_{2i}, and $\omega_{3i} \in [0, 1]$, and c_j is the center position of task j in the abscissa and ordinate axes of the map:

$$d_{ij} = \min(c_j - p_{1i}, c_j - p_{2i}, c_j - p_{3i})$$
$$F[i, j] = (1/\exp(-d_{ij}^2/2) \times W_i^T \times V_{ij}$$

where d_{ij} is the distance from the center position of task j to the nearest position in P_i of agent i. V_{ij} is a three-vector and $v_{1ij}, v_{2ij}, v_{3ij} \in \{0, 1\}$, where one of the elements in V_{ij} must be equal to 1 and the others must be 0. It expresses the nearest position to c_j in P_i when the element in V_{ij} is equal to 1. $F[i, j]$ expresses that the farther from the familiar position of agent i is, the lower the familiarity degree with the task j is.

With the role evaluation vector E, personal qualification Q_p, and familiarity matrix F, the qualification matrix Q can be calculated as follows. For each agent i and role j

$$Q[i, j] = \frac{n_j \times Q_p[i] \times F[i, j] \times E[j]}{N} \quad (0 \leq i < m, 0 \leq j < n)$$

where n_j represents the number of orders in a role j, and N represents the number of daily orders. The qualification matrix Q can be normalized before the assignment, where $\min\{Q\}$ is the minimum element in the matrix Q, and $\max\{Q\}$ is the maximum element

$$Q[i, j] = \frac{Q[i, j] - \min\{Q\}}{\max\{Q\} - \min\{Q\}} \quad (0 \leq i < m, 0 \leq j < n).$$

B. Role Formulation

Since a delivery task performed in one trip by an agent is considered as a role, role formulation is to generate roles by clustering the next day's orders from 8:00 AM to 8:00 PM into delivery tasks (roles). As described in role evaluation, the range of time and space in a role should be as small as possible. Hence, a clustering method that takes time and distance into consideration should be used. Here, we partition orders by a traditional clustering algorithm called k-means++, which is both simple and efficient [4].

In this scenario, the range of distance (z_j) is $[0, \sqrt{2}]$ and the range of time (y_j) is [0, 720]. In *Definition 11*, the scale of distance and the scale of time differ by orders of magnitude. Thus, this clustering method tends to be too strict with time and be too loose with distance. For example, the tasks divided by this clustering method are likely to require the courier to complete a mile-long delivery in one minute, which may not be feasible. To integrate time and space more reasonably, we use the generalized Euclidean distance to normalize and define the distance of orders between each other in the clustering method.

The generalized Euclidean distance is shown as follows:

$$D = \sqrt{(z - z_l)^2 + a^2(y - y_l)^2} \quad (a \in (0, 1], a \in \mathbb{R})$$

where (y_l, z_l) denotes a center position of a cluster in time and distance. Parameter a is a coefficient of the factor of time, which is used to integrate time and space.

However, k-means++ brings out a number of problems.

Problem (1): it is difficult to determine the value of parameter a since time (y_j) and distance (z_j) differ by orders of magnitude. Problem (2): the number of orders (x_j) in a role j cannot be considered in k-means++. Problem (3): the number of clusters k needed by k-means++ is difficult to determine at the beginning.

To solve Problem (1), parameter a can be set according to our expectations. For example, we assume that one mile is the longest travel distance for any order in a delivery task (*role*). Similarly, we suppose that the quality of fresh produce is unacceptable after u minutes. The longest time available for each task is u. Parameter u is an integer. To ensure that

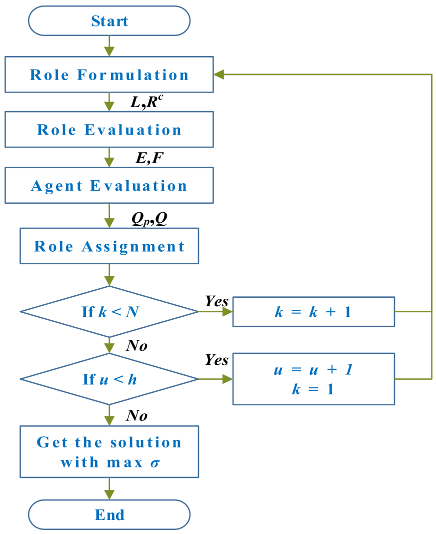


Fig. 6. Flowchart of the solution with the ARA method.

tasks with the longest travel distance can be completed in u minutes, parameter a can be set as $a = 1/u$.

However, parameter a may fluctuate within a certain range, we should find a better and more accurate method to set parameter a. To simplify, we use parameter u to further determine parameter a that discusses the coefficient of the factor of time, since a is influenced by u.

Problem (2) can be indirectly solved by adjusting the number of clusters. The greater the number of clusters is, probably the fewer the orders in a cluster (role) is. Thus, Problem (3) is much more complicated than Problem (2). In fact, Problem (2) can be viewed as a subproblem of the Problem (3).

For problem (3), the number of clusters k is difficult to determine, as it not only affects the capacity of a role but also has an impact on the time and distance of a role.

In this article, we use the clustering method k-means++ to gather orders related to time and space. However, because the time and space factors differ by orders of magnitude, it is difficult to compare them. The normalization method is applied to overcome this difficulty. Moreover, general clustering methods usually need to set the number of clusters, which are difficult to determine. Therefore, to find a reasonable task granularity, this article sets up a role evaluation system and uses an adaptive clustering method to find the ratios of time and space and the number of clusters. To find the parameters for best performance, we adopt an adaptive clustering method that adjusts parameters u, within a certain range (d, h), and adjusts k within a range from 1 to N, where d and h are integers $(d < h)$. After the task division is done with fixed u and k, the qualification matrix Q can be calculated. Meanwhile, conflicting role matrix R^c can also be calculated based on *Definition 5*. The qualification of the role formulation can be evaluated by the performance of the assignment after role evaluation. In this way, this article composes a solution to the LMAP problem. The flowchart is shown in Fig 6.

To present the assignment algorithm with an ARA method in detail, a pseudocode is described as follows. To illustrate the assignment algorithm, we first introduce some special symbols.

1) *u:* a parameter in the generalized Euclidean distance within a range (*u.start*, *u.end*) with a step of *u.update*.
2) *k:* a parameter of the number of clusters.
3) *Order:* a list of daily orders with information of destination and time.
4) *r:* the delivery task after partition. It contains information on the time, distance, and capacity.
5) $|V|$: the cardinality of V, where V is a vector.
6) *InitializeMatrix (m, k, 0):* a function to initialize an $m \times k$ matrix, where the values are 0.
7) *k-means++ (Order, u, k):* a function to partition orders into k parts by a traditional clustering algorithm with parameters u for normalization.
8) *InitializeVector (k, 1):* a function to initialize a k-vector with values of 1.
9) *getConflictMatrix (r):* a function to build the conflicting role matrix based on *Definition 5*.
10) *f (r.time, r.distance, r.capacity):* a function to calculate the role evaluation vector based on *Definition 11*.
11) $g\ (E,\ Q_p,\ F)$: a function to calculate the qualification matrix Q with a role evaluation vector E, personal qualification Q_p, and familiarity degree F.
12) *doAssignment* (L, L^a, Q, R^c): a function that performs assignments based on *Definition 10*.
13) *getMaxPerformance (TA, T):* a function that records the solution T with the maximal performance σ.

Algorithm 1 ARA Algorithm: RA Algorithm

Input:

- An m-vector L^a with the above preliminaries;
- An m-vector Q_p with the above preliminaries;
- A list of orders *Order* with information of destination and time with the above preliminaries;
- The parameter of u;
- The parameter of k;

Output:

- A *role assignment matrix* T.

ARA (L^a, Q_p, *Order*, u, k){
T = InitializeMatrix (m, k, 0);
for (u = *u.start*; $u \le$ *u.end*; $u = u+$ *u.update*){
    for (k = *1*; $k \le N$; $k = k+$ 1){
        r = *k-means++*(*Order*, u, k);
        L = *InitializeVector* (k, 1);
        R^c = *getConflictMatrix* (r);
        $E = f$ (*r.time*, *r.position*, *r.capacity*);
        $Q = g\ (E,\ Q_p,\ F)$;
        TA = *doAssignment* (L, L^a, Q, R^c);
        T = *getMaxPerformance* (*TA*, T);}}
return T;}

Note that m is the cardinality of L^a and N represents the number of daily orders. The outside loop of ARA runs $\lceil (u.end - u.start)/u.update \rceil$ times. As it is a constant value, we set that a constant variable p is equal to

TABLE VI
TOP THREE FAMILIAR POSITIONS OF COURIERS

Position	P1	P2	P3
Adam	(0.23,0.54)	(0.87, 0.57)	(0.66,0.37)
Brian	(0.42,0.65)	(0.34,0.54)	(0.33,0.26)
Chris	(0.57,0.44)	(0.45,0.84)	(0.21,0.42)

Note that the familiarity degrees of P_1, P_2, P_3 are 0.9, 0.8, and 0.7, respectively.

$\lceil (u.end - u.start)/u.update \rceil$. Hence, the complexity of the two-outside loop (for loop) is $O(p \times N)$. The complexity of the function *k-means++* (*Order*, u, k) is $O(k \times N \times t)$, where k is the number of clusters and t is the number of iterations of the clustering method. The function of *InitializeVector* (k, 1) is $O(k)$. The function of *getConflictMatrix* (r) is $O(k^2)$. The function f (*r.time*, *r.position*, *r.capacity*) and g (E, Q_p, F) are $O(k)$ and $O(k \times m)$. The function of *getMaxPerformance* (*TA*, T) is $O(m \times n)$. The function of *doAssignment* (L, L^a, Q, R^c) can use CPLEX to calculate. CPLEX is a commercial optimization software package supported by IBM. We do not know the actual complexity of CPLEX. However, it is known that the method to solve mixed integer programming may be the branch and cut method in CPLEX. Thus, we suppose that the worst complexity of *doAssignment* (L, L^a, Q, R^c) is $O(2^n)$. Therefore, the algorithm of ARA has the computational complexity of $O(p \times N \times (k \times N \times t + k + k^2 + k + k \times m + m \times n + 2^n))$, which can be simplified as $O(2^n)$.

This article uses role awareness to create reasonable role structures through iterative role formulation and evaluation. In role awareness, the evaluation method is used to evaluate the qualification of roles based on different role structures after each role formulation. We then assign appropriate couriers to the relevant delivery task to effectively use human resources and improve the quality of service.

Furthermore, to check the applicability of the solution, we conduct experiments for the scenario given in Section II, where parameter u is from 10 to 290, i.e., $d = 10$, $h = 290$, with an increment of 10, the number of clusters k is from 1 to N with an increment of 1. Parameters μ_x, μ_y, μ_z, δ_x^2, δ_x^2, and δ_x^2 are set as 15, 1, 0, 6.36, 12.29, and 0.42. The top three familiar positions are shown in Table VI. The performance experiments are based on the platform shown in Table V.

The experimental result is shown in Fig. 7. We get the best performance when parameter u is in the range of 190 to 290 and k is equal to 19. The total sum of the assigned evaluation values is 16.19, which is more than twice the previous performance value in the scenario. The matrixes Q, R^c, and T are shown in Fig. 8(a)–(c).

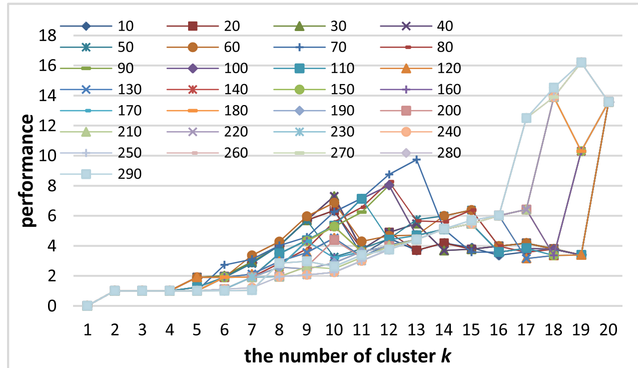


Fig. 7. Performance of the ARA solution for the scenario in Section II ($m = 3$, and the labels in the legend show parameter u from 10 to 290 with an increment of 10).

$$\begin{bmatrix} 0.84 & 0.72 & 0.87 & 0.70 & 0.81 & 0.68 & 0.99 & 0.96 & 0.01 & 0.97 & 0.86 & 0.80 & 0.71 & 0.87 & 0.81 & 0.73 & 0.99 & 1.0 & 0.69 \\ 0.74 & 0.89 & 0.77 & 0.85 & 0.79 & 0.89 & 0.78 & 0.74 & 0.00 & 0.66 & 0.73 & 0.82 & 0.86 & 0.77 & 0.87 & 0.89 & 0.65 & 0.66 & 0.89 \\ 0.77 & 0.89 & 0.88 & 0.88 & 0.77 & 0.83 & 0.64 & 0.74 & 0.01 & 0.66 & 0.78 & 0.76 & 0.89 & 0.78 & 0.75 & 0.90 & 0.73 & 0.64 & 0.86 \end{bmatrix}$$

(a)

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

(b)

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

(c)

Fig. 8. Q, R^c, and T matrix.

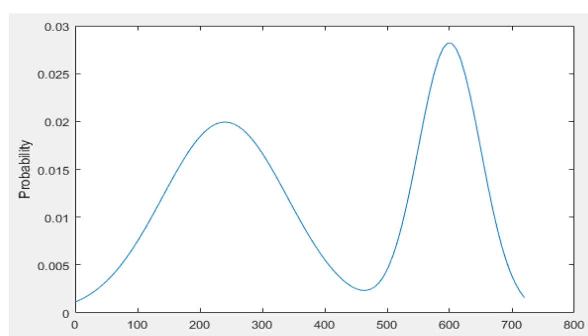


Fig. 9. Probability of the time for daily order. (The range from 0 to 720 min is used to represent the period from 8 AM to 8 PM.).

V. EXPERIMENTS

We conduct experiments based on Fig. 6. Here, we make experiments with random groups of different scales, where the number of orders N is from 100 to 1000 with an increment of 100. Parameter u is from 10 to 150 with an increment of 10. The number of clusters k is from 1 to N with an increment of 1. The number of agents is 5. The value of the top three familiar positions and the personal qualification vector Q_p are randomly generated. The platform is shown in Table V. We simulate the order's destination data using a uniform distribution and simulate the order's request time data using a distribution shown in Fig. 9.

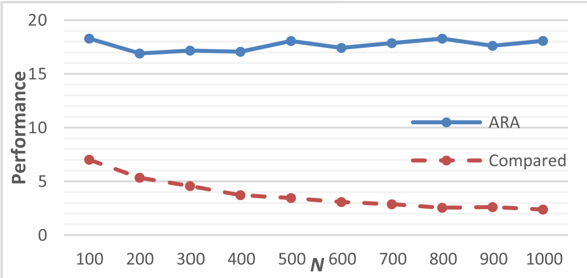


Fig. 10. Comparison of the performance.

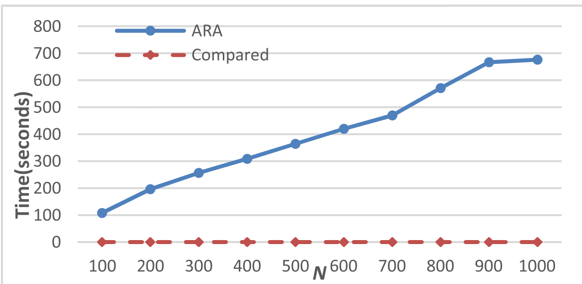


Fig. 11. Comparison of the time.

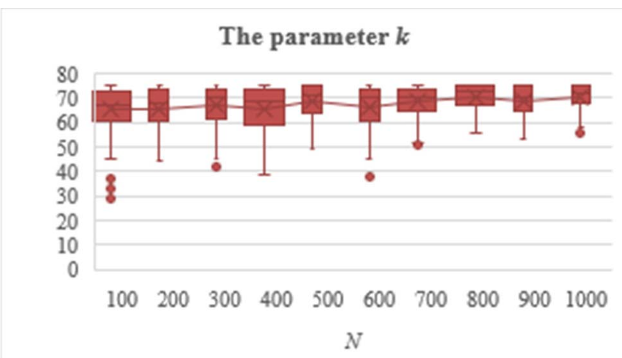


Fig. 12. Number of clusters.

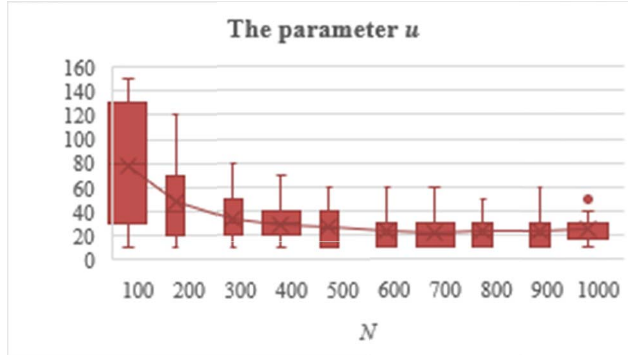


Fig. 13. Coefficient of the factor of time.

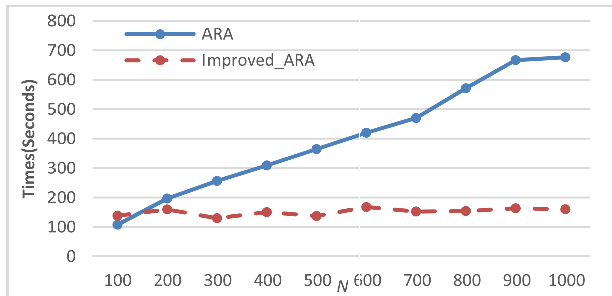


Fig. 14. Comparison of the time.

TABLE VII
SEARCH RANGES OF THE PARAMETERS

The search ranges of the parameter u										
N	100	200	300	400	500	600	700	800	900	1000
Lowest	10	10	10	10	10	10	10	10	10	10
Largest	160	120	80	70	60	60	60	50	60	50
The search ranges of the parameter k										
N	100	200	300	400	500	600	700	800	900	1000
Lowest	25	40	40	35	45	35	50	45	50	50
Largest	80	80	80	80	80	80	80	80	80	80

Here is a comparable method that clusters orders of the same hour into delivery tasks based on the region shown in the scenario. We collect the average group performance σ and processing time to process the same 100 random problems at each scale (N) (Fig. 10). Both methods use Definition 10 to assign and evaluate roles in Section IV.

Compared with the cluster method in comparison, the solution of the ARA approach has a higher team performance. However, the computational complexity of the ARA approach is relatively high, resulting in a long calculation time (Fig. 11). The ARA approach consumes a lot of time to find the optimal parameters for role formulation through adaptive searching. This is contrasted with the traditional method that uses brute force to determine the roles. From Figs. 12 and 13, we find that there are some correlations between the parameters and the number of daily orders N. For example, the numbers of clusters (parameter k) are bigger than 40 when the number of daily orders N is larger than 700. In contrast, parameter u is less than 60 when the number of daily orders (N) is large than 500. Parameter u decreases when the number of daily order N increases.

Therefore, we can reduce the search ranges of the parameters in adaptive optimization in order to reduce the calculation time in experiments. The search ranges of the parameters are shown in Table VII. In Fig. 14, the time of the ARA approach used by CPLEX is obviously reduced by narrowing the search ranges of the parameters. By learning the varying trend of the parameters, experiments indicate that the improved ARA solution can use a shorter time to process the same number of LMAP problems. The result demonstrates that solving the LMAP problem via GRA is practical.

VI. Related Work

A. Last-Mile Logistics

Many researchers study problems in last-mile logistics [27]. With the advent of drones in last-mile logistics, routing problem has become rather popular [2], [6], [7], [11], [12],

[26], [32], [51]. However, in developing countries, delivery via couriers is still the mainstream. There are many flexible delivery paths for couriers. Therefore, the influences of routing and road conditions are not significant in last-mile logistics.

With the advent of crowdsourcing, task assignment is a critical part of the delivery in last-mile logistics [9], [40], [49]. In normal last-mile logistics, there is rarely any need to consider the timeliness factor in deliveries. Thus, researchers usually focus on the distance factor in task allocation [40], [49]. However, there is a high demand for timeliness to avoid food spoilage during delivery for fresh products in the last-mile logistics. For many investigators, distance is a major consideration in task allocation. They usually consider timeliness in vehicle routing after task allocation [6], [11], [17], [28], [51]. In their allocation, the hit rate is usually ignored. However, the hit rate is the main factor affecting the quality of fresh products. Some researchers study the hit rate in logistics by estimating the probability of a successful delivery based on historical data [13], [30]. They aim to design a set of routes that maximizes the expected number of successful hits in a given maximum route duration. However, the timeliness factor is not considered in these studies.

This article focuses on the last-mile logistics problem of fresh food in crowdsourcing. The crowdsourcing platform assigns tasks to couriers. In task allocation, we consider the request time specified by customers to improve the hit rate. Meanwhile, we also take timeliness into consideration since fresh food easily spoils in last-mile logistics.

B. Spatiotemporal Crowdsourcing

Current researches on crowdsourcing mainly focus on task allocation, quality control, and so on [1], [8], [9], [16], [24], [30], [35], [46], [48]. This article focuses on a unified crowdsourcing platform that allocates jobs to participants to accomplish a collaborative goal.

Time and space are the most common and important factors in service. Spatiotemporal crowdsourcing takes time and space factors into consideration by assigning tasks based on these factors to achieve high performance [3], [40]. Compared with traditional crowdsourcing, spatiotemporal crowdsourcing has stronger and more complex space–time constraints. The following are some key issues in task allocation and quality control [1], [8], [16], [24], [30], [36].

1) *Real-Time Scheduling:* Dynamic scheduling has become the mainstream model of task allocation [35], [49] in crowdsourcing platforms. The assignment mainly considers the distance between orders and customers in order to respond quickly. Path planning is an important consideration of the assignment [14], [35], [49]. However, such crowdsourcing platforms often need a lot of staff. Moreover, the crowdsourcing platform does not take the overall quality of service into account because of the timeliness requirements.

   This article studies the last mile logistics of daily fresh produce orders reserved one day in advance. In this article, we consider the timeliness requirement as a soft constraint in the assignment. We construct the delivery task to meet the timeliness requirement as much as possible.
2) *Task Granularity:* Many investigations focus on granularity in different applications [25], [29]. The study order consolidation in last-mile delivery, in which orders with the same customer and destination are consolidated into a task [11], [13], [15], [18], [28]. Similar orders could be integrated into one task for delivery to improve efficiency and service.

   The task granularity considers the association between orders. This article considers task granularity as the scope of a task in terms of time and space. We aim to find a reasonable task granularity to improve the assignment performance by using the clustering method.
3) *Assignment With the Best Performance:* Task allocation and quality control aim to allocate tasks to participants to produce high-quality performances, such as allocation for the shortest distance, cost minimization, and so on [9], [35], [36], [40], [42], [47].

In LMAP, this article solves the assignment problem to obtain the optimal team performance based on role and agent evaluations. The evaluation considers the qualification of roles, personal qualification Q_p, and so on. After finding a reasonable task granularity, we use GRA$^+$ to maximize group performance in last-mile logistics.

C. Role Assignment

Last-mile logistics is one of the collaboration problems that aim to coordinate couriers with suitable delivery tasks. Collaboration research is a challenging topic in a variety of domains [2], [54]. In collaboration research, roles (tasks) and agents (workers) are used to describe collaboration activities [20], [25], [37], [41], [43]. A methodology called RBC that uses roles as the primary underlying mechanism to facilitate collaboration activities is proposed [41], [45], [52]–[55]. RBC is the fundamental theory and methodology of this article. One of the kernels of RBC is GRA [50], [53]. In GRA, the process of task assignment aims to propose an assignment with optimal team performance based on agent evaluations.

Many investigations focus on assignment problems with various constraints tailored to different applications [10], [20], [23], [41], [45]. Roles and agents have many constraints, including mutual exclusion, prerequisites, preferences, and so on, which make assignment problem complex. Many researchers study the assignment problem by using heuristic methods to find a global near-optimal solution [22], [33]. Meanwhile, many researchers use linear programming to solve assignment problems. IBM developed a linear programming platform CPLEX to address the complexity of assignment algorithms [19]. CPLEX can easily be used to solve a series of assignment problems [20].

However, it is worth mentioning that existing assignment methods [5] to solve RA problems cannot be directly used to solve the addressed LMAP problem [54].

The aforementioned research indicates a strong need to fundamentally investigate RAs to assist the crowdsourcing platform in developing an effective task distribution process.

VII. Conclusion

This article deals with the LMAP for fresh produce, one of the fundamental and important problems in e-commerce, via GRA with constraints (GRA$^+$).

We formalize the LMAP problem via GRA at first. Next, we implement a CPLEX solution and verify its performance. The evaluation of qualification value is based on assumed historical data. The orders in the same hour are clustered into tasks based on fixed regions. However, the daily delivery orders are different every day. Thus, different role formulations affect the quality of service. We provide then a role awareness method by using the adaptive clustering method based on task granularity. We obtain a reasonable role structure by adaptive role formulation and evaluation. The proposed method not only assigns couriers to deliver daily orders efficiently but also improves the quality of service. Lastly, experiments indicate that the ARA solution for the LMAP problem via GRA$^+$ is practical.

From this article, it is clear that further investigations may be required in the following directions: 1) although the proposed solutions perform well on a practical level, it is still possible to find more efficient implementation methods, such as parallel processing, matrix processing, and so on. 2) The proposed method could be applied in a real-world environment to verify their practicability for further investigation. 3) Last but not least, there are more general relations and additional constraints among roles, which can be investigated in the future.

Acknowledgment

The authors would like to thank Zikai Wang of Nipissing University, North Bay, ON, Canada, for his assistance in proofreading this article.

References

[1] 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.

[2] E. Aktas, M. Bourlakis, and D. Zissis, "Collaboration in the last mile: Evidence from grocery deliveries," *Int. J. Logistics Res. Appl.*, pp. 1–15, 2020.

[3] H. Aly, J. Krumm, G. Ranade, and E. Horvitz, "To buy or not to buy: Computing value of spatiotemporal information," *ACM Trans. Spatial Algorithms Syst.*, vol. 5, no. 4, pp. 1–25, Dec. 2019.

[4] D. Arthur and S. Vassilvitskii, "K-means plus plus: The advantages of careful seeding," in *Proc. 18th Annu. ACM-SIAM Symp. Discrete Algorithms (ACM SIGACT)*, Jan. 2007, pp. 1027–1035.

[5] R. Burkard, M. DellAmico, and S. Martello, *Assignment Problems*. Philadelphia, PA, USA: SIAM, 2012.

[6] M. Bortolini, M. Faccio, E. Ferrari, M. Gamberi, and F. Pilati, "Fresh food sustainable distribution: Cost, delivery time and carbon footprint three-objective optimization," *J. Food Eng.*, vol. 174, pp. 56–67, Apr. 2016.

[7] F. M. Bergmann, S. M. Wagner, and M. Winkenbach, "Integrating first-mile pickup and last-mile delivery on shared vehicle routes for efficient urban e-commerce distribution," *Transp. Res. B, Methodol.*, vol. 131, pp. 26–62, Jan. 2020.

[8] A. I. Chittilappilly, L. Chen, and S. Amer-Yahia, "A survey of general-purpose crowdsourcing techniques," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 9, pp. 2246–2266, Sep. 2016.

[9] P. Cheng, X. Lian, L. Chen, J. Han, and J. Zhao, "Task assignment on multi-skill oriented spatial crowdsourcing," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 8, pp. 2201–2215, Aug. 2016.

[10] S. Chopra, G. Notarstefano, M. Rice, and M. Egerstedt, "A distributed version of the hungarian method for multirobot assignment," *IEEE Trans. Robot.*, vol. 33, no. 4, pp. 932–947, Aug. 2017.

[11] C. Chao, T. Zhihui, and Y. Baozhen, "Optimization of two-stage location–routing–inventory problem with time-windows in food distribution network," *Ann. Oper. Res.*, vol. 273, nos. 1–2, pp. 111–134, Feb. 2019.

[12] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, "Vehicle routing problems for drone delivery," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 1, pp. 70–85, Jan. 2017.

[13] A. M. Florio, D. Feillet, and R. F. Hartl, "The delivery problem: Optimizing hit rates in e-commerce deliveries," *Transp. Res. B, Methodol.*, vol. 117, pp. 455–472, Nov. 2018.

[14] C. Fikar, P. Hirsch, and M. Gronalt, "A decision support system to investigate dynamic last-mile distribution facilitating cargo-bikes," *Int. J. Logistics Res. Appl.*, vol. 21, no. 3, pp. 300–317, May 2018.

[15] G. Fancello, D. Paddeu, and P. Fadda, "Investigating last food mile deliveries: A case study approach to identify needs of food delivery demand," *Res. Transp. Econ.*, vol. 65, pp. 56–66, Oct. 2017.

[16] H. Garcia-Molina, M. Joglekar, A. Marcus, A. Parameswaran, and V. Verroios, "Challenges in data crowdsourcing," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 4, pp. 901–911, Apr. 2016.

[17] Y.-H. Hsiao, M.-C. Chen, K.-Y. Lu, and C.-L. Chin, "Last-mile distribution planning for fruit-and-vegetable cold chains," *Int. J. Logistics Manage.*, vol. 29, no. 3, pp. 862–886, Aug. 2018.

[18] Y. Huang, M. Savelsbergh, and L. Zhao, "Designing logistics systems for home delivery in densely populated urban areas," *Transp. Res. B, Methodol.*, vol. 115, pp. 95–125, Sep. 2018.

[19] IBM. (2018). *ILOG CPLEX Optimization Studio*. [Online]. Available: http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/

[20] 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.

[21] D. Liu, Q. Jiang, H. Zhu, and B. Huang, "Distributing uavs as wireless repeaters in disaster relief via group role assignment," *Int. J. Cooperat. Inf. Syst.*, vol. 29, nos. 1–2, pp. 2040002-1–2040002-22, 2020.

[22] D. Li, M. Li, X. Meng, and Y. Tian, "A hyperheuristic approach for intercell scheduling with single processing machines and batch processing machines," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 2, pp. 315–325, Feb. 2015.

[23] 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.

[24] G. Li, J. Wang, Y. Zheng, and M. J. Franklin, "Crowdsourced data management: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 9, pp. 2296–2319, Sep. 2016.

[25] M. Motamedi, D. Fong, and S. Ghiasi, "Machine intelligence on resource-constrained IoT devices: The case of thread granularity optimization for CNN inference," *ACM Trans. Embedded Comput. Syst.*, vol. 16, no. 5, pp. 1–19, Oct. 2017.

[26] M. Moshref-Javadi, A. Hemmati, and M. Winkenbach, "A truck and drones model for last-mile delivery: A mathematical model and heuristic approach," *Appl. Math. Model.*, vol. 80, pp. 290–318, Apr. 2020.

[27] J. Olsson, D. Hellström, and H. Pålsson, "Framework of last mile logistics research: A systematic review of the literature," *Sustainability*, vol. 11, no. 24, p. 7131, Dec. 2019.

[28] J. A. Orjuela-Castro, J. P. Orejuela-Cabrera, and W. Adarme-Jaimes, "Last mile logistics in mega-cities for perishable fruits," *J. Ind. Eng. Manage.*, vol. 12, no. 2, pp. 318–327, 2019.

[29] W. Pedrycz, "Allocation of information granularity in optimization and decision-making models: Towards building the foundations of granular computing," *Eur. J. Oper. Res.*, vol. 232, no. 1, pp. 137–145, Jan. 2014.

[30] L. Pournajaf, D. A. Garcia-Ulloa, L. Xiong, and V. Sunderam, "Participant privacy in mobile crowd sensing task management: A survey of methods and challenges," *ACM SIGMOD Rec.*, vol. 44, no. 4, pp. 23–34, May 2016.

[31] A. Schrijver, *Theory of Linear and Integer Programming*. Chichester, U.K.: Wiley, 2000.

[32] A. Stenger, D. Vigo, S. Enz, and M. Schwind, "An adaptive variable neighborhood search algorithm for a vehicle routing problem arising in small package shipping," *Transp. Sci.*, vol. 47, no. 1, pp. 64–80, Feb. 2013.

[33] J. Schwarzrock, I. Zacarias, A. L. C. Bazzan, R. Q. de Araujo Fernandes, L. H. Moreira, and E. P. de Freitas, "Solving task allocation problem in multi unmanned aerial vehicles systems using swarm intelligence," *Eng. Appl. Artif. Intell.*, vol. 72, pp. 10–20, Jun. 2018.

[34] 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.

[35] L. Tran, H. To, L. Fan, and C. Shahabi, "A real-time framework for task assignment in hyperlocal spatial crowdsourcing," *ACM Trans. Intell. Syst. Technol.*, vol. 9, no. 3, pp. 1–26, Feb. 2018.

[36] Y. Tong, Z. Zhou, Y. Zeng, L. Chen, and C. Shahabi, "Spatial crowdsourcing: A survey," *VLDB J.*, vol. 29, no. 1, pp. 217–250, Jan. 2020.

[37] W. Wang, Y. Jiang, and W. Wu, "Multiagent-based resource allocation for energy minimization in cloud computing systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 2, pp. 205–220, Feb. 2017.

[38] L. A. Wolsey and G. L. Nemhauser, *Integer and Combinatorial Optimization*. New York, NY, USA: Wiley, 1999.

[39] Z. Wang and C. Shen, "Small cell transmit power assignment based on correlated bandit learning," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1030–1045, May 2017.

[40] 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.

[41] Y. Wang, D. Zhang, Q. Liu, F. Shen, and L. H. Lee, "Towards enhancing the last-mile delivery: An effective crowd-tasking model with scalable solutions," *Transp. Res. E, Logistics Transp. Rev.*, vol. 93, pp. 279–293, Sep. 2016.

[42] W. Xiong, C. Pan, Y. Qiao, N. Wu, M. Chen, and P. Hsieh, "Reducing wafer delay time by robot idle time regulation for single-arm cluster tools," *IEEE Trans. Autom. Sci. Eng.*, early access, Aug. 17, 2020, doi: 10.1109/TASE.2020.3014078.

[43] H. Zhu, "A role agent model for collaborative systems," in *Proc. Int. Conf. Inf. Knowl. Eng. (IKE)*, Las Vegas, NV, USA, 2003, pp. 438–444.

[44] H. Zhu, "Avoiding conflicts by group role assignment," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 4, pp. 535–547, Apr. 2016.

[45] H. Zhu, "Group multi-role assignment with conflicting roles and agents," *IEEE/CAA J. Automatica Sinica*, vol. 7, no. 6, pp. 1498–1510, Nov. 2020.

[46] K. Zaamout, T. Arjannikov, and K. Barker, "A social crowdsourcing community case study: Interaction patterns, evolution, and factors that affect them," *IEEE Trans. Comput. Social Syst.*, vol. 7, no. 3, pp. 659–671, Jun. 2020.

[47] L. Zhou, R. Baldacci, D. Vigo, and X. Wang, "A multi-depot two-echelon vehicle routing problem with delivery options arising in the last mile distribution," *Eur. J. Oper. Res.*, vol. 265, no. 2, pp. 765–778, Mar. 2018.

[48] L. Zhang, Y. Ding, X. Wang, and L. Guo, "Conflict-aware participant recruitment for mobile crowdsensing," *IEEE Trans. Comput. Social Syst.*, vol. 7, no. 1, pp. 192–204, Feb. 2020.

[49] L. Zhang *et al.*, "A taxi order dispatch model based on combinatorial optimization," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 2151–2159.

[50] H. Zhu, M. Hou, and M. Zhou, "Adaptive collaboration based on the E-CARGO model," *Int. J. Agent Technol. Syst.*, vol. 4, no. 1, pp. 59–76, Jan. 2012.

[51] F. E. Zulvia, R. J. Kuo, and D. Y. Nugroho, "A many-objective gradient evolution algorithm for solving a green vehicle routing problem with time windows and time dependency for perishable products," *J. Cleaner Prod.*, vol. 242, Jan. 2020, Art. no. 118428.

[52] H. Zhu, D. Liu, S. Zhang, S. Teng, and Y. Zhu, "Solving the group multirole assignment problem by improving the ILOG approach," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 12, pp. 3418–3424, Dec. 2017.

[53] 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.

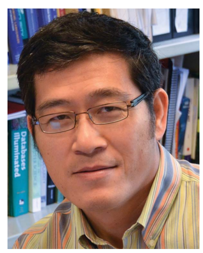
[54] H. Zhu and M. Zhou, "Role-based collaboration and its kernel mechanisms," *IEEE Trans. Syst., Man Cybern. C, Appl. Rev.*, vol. 36, no. 4, pp. 578–589, Jul. 2006.

[55] H. Zhu, M. Zhou, and R. Alkins, "Group role assignment via a Kuhn–Munkres algorithm-based solution," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 42, no. 3, pp. 739–750, May 2012.

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. She is currently pursuing the Ph.D. degree with the Institute of Systems Engineering, Macau University of Science and Technology, Macau.

She has authored or coauthored a regular article in the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING.

Ms. Huang is a reviewer for a number of journals and conferences.



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

He was 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 is currently a Full Professor and the Chair of the Department of Computer Science and Mathematics, Nipissing University, North Bay, ON, Canada, where he is also the Founding Director of the Collaborative Systems Laboratory. He has authored or coauthored (and been accepted) over 200 research works, including over 20 IEEE TRANSACTIONS articles, 5 books, 5 book chapters, 3 journal issues, and 3 conference proceedings. He is the founding researcher of role-based collaboration and adaptive collaboration. He has offered over 70 invited talks on collaboration topics internationally. His research has been being sponsored by the Natural Sciences and Engineering Research Council of Canada (NSERC), International Business Machines Corporation (IBM), Defence Research and Development Canada (DRDC), and Ontario Partnership for Innovation and Commercialization (OPIC). His research interests include collaboration theory, technologies, systems, and applications, human-machine systems, computer-supported cooperative work, multi-agent systems, software engineering, and distributed intelligent systems.

Dr. Zhu is a member of the SSE Technical Activity Committee, the Conferences and Meetings Committee, and the Electronic Communications Subcommittee of the IEEE SMC Society. He was a receipt of the Meritorious Service Award from the IEEE Systems, Man and Cybernetics (SMC) Society in 2018, the Chancellor's Award for excellence in research in 2011, two Research Achievement Awards from Nipissing University in 2006 and 2012, respectively, the IBM Eclipse Innovation Grant Awards in 2004 and 2005, the Best Paper Award from the 11th ISPE International Conference on Concurrent Engineering (ISPE/CE2004), the Educator's Fellowship of OOPSLA'03, the 2nd class National Award for Education Achievement in 1997, and three 1st Class Ministerial Research Achievement Awards from China in 1997, 1994, and 1991. He is serving as the Co-Chair for the Technical Committee of the Distributed Intelligent Systems of the IEEE SMC Society. He has been an Active Organizer for many IEEE conferences, including the PC (Co-) Chair, the Special Session Chair, the Tutorial Chair, the Area Co-Chair, the Social Media Co-Chair, the Web Co-Chair, and the Poster Co-Chair. He also served as a PC member for more than 80 academic conferences. He is serving as an Associate Editor of the IEEE TRANSACTIONS ON SMC: SYSTEMS, the IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, *IEEE SMC Magazine*, and the IEEE CANADA REVIEW.

Dongning Liu (Member, IEEE) was born in 1979. He received the Ph.D. degree in logic from Sun Yat-sen University, Guangzhou, China, in 2007.

He was a Post-Doctoral Fellow in math with Sun Yat-sen University from 2007 to 2009. He has been a Full Professor with the Guangdong University of Technology, Guangzhou, since 2009, where he is responsible for teaching artificial intelligent logic and discrete math at the School of Computer Science and Technology. He is engaged in education and technology transfer on collaborative computing. He is also the Vice Dean of the School of Computer Science and Technology, Guangzhou. He was a Visiting Professor with Nipissing University, North Bay, ON, Canada, from 2015 to 2016. He has authored or coauthored more than 50 articles on computer magazines and international conferences.

Dr. Liu is a member of the IEEE Systems, Man and Cybernetics (SMC) Society. He is serving as a TC member of the technical committee of the Distributed Intelligent Systems and a Senior Member of the China Computer Federation (CCF) Society. He is serving as the Vice Secretary-General and a Standing Committee Member of the Technical Committee on Cooperative Computing of the China Computer Federation. He is a reviewer of the IEEE TRANSACTIONS ON SMC: SYSTEMS, the IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, and the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY.

Naiqi Wu (Fellow, IEEE) received the B.S. degree in electrical engineering from the Huainan Institute of Technology, Huainan, China, in 1982, 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 Guangdong University of Technology, Guangzhou, China, in 1998. He joined the Macau University of Science and Technology, Macau, in 2013, where he is currently the Chair Professor with the Institute of Systems Engineering. He has authored or coauthored one book, five book chapters, and more than 180 peer-reviewed journal articles. 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.

Dr. Wu was an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, PART C, the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, and the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, and the Editor-in-Chief of the *Industrial Engineering Journal*. He is an Associate Editor of *Information Sciences*.

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

From 2014 to 2015, he was a Visiting Student with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, USA. From 2016 to 2017, he was a Post-Doctoral Research Associate with the Institute of Systems Engineering, Macau University of Science and Technology, Macau, where he has been an Assistant Professor since 2018. He has over 70 publications, including 1 book chapter and more than 30 regular articles in the IEEE TRANSACTIONS. His research interests include discrete-event systems, production planning, Petri nets, scheduling, and control.

Qian Jiang (Graduate Student Member, IEEE) was born in 1997. He received the B.S. degree in network engineering from the School of Computer Science and Technology, Guangdong University of Technology, Guangzhou, China, in 2018, where he is currently pursuing the master's degree majoring in collaborative computing.

He has reported a conference article as the first author in the Chinese Conference on Computer Supported Cooperative Work and Social Computing (CSCW) 2019. He has also authored or coauthored an article in the *International Journal of Cooperative Information Systems* in 2020.