Optimizing Multi-Center Collaboration for Task Assignment in Spatial Crowdsourcing

Ximu Zeng[1], Jianxing Lin[1], Liwei Deng[1], Yuchen Fang[1], Yan Zhao[2,✉], Kai Zheng[1,2,✉]
[1]University of Electronic Science and Technology of China, Chengdu, China
[2]Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, Shenzhen, China
{ximuzeng, jianxinglin, deng_liwei, fangyucheng}@std.uestc.edu.cn
zhaoyan@uestc.edu.cn, zhengkai@uestc.edu.cn

***Abstract*—The rapid development of smart devices has fostered the growth of Spatial Crowdsourcing (SC), where workers complete spatial tasks by traveling to specific locations. Task assignment is a key issue in SC due to the inherent complexity of matching workers with these spatial tasks efficiently. Previous studies on task assignment have primarily focused on optimizing worker-task matching within a single, centralized area, often ignoring scenarios that involve multiple independent service centers across an area. To address this gap, we introduce a collaborative multi-center task assignment problem, which focuses on scenarios where an SC platform manages multiple independent service centers within an area, shifting the focus from worker-level cooperation to exploring the solutions specific to multi-center coordination. We target the imbalances between available workers and unassigned tasks among different centers, aiming to maximize the total number of assigned tasks and minimize unfairness in inter-center collaboration. In particular, we propose an Iterative Multi-center Task Assignment and Optimization (IMTAO) framework. IMTAO operates in two phases: (1) center-independent task assignment based on an efficient sequential task assignment algorithm, and (2) inter-center workforce transfer based on a game-theoretic multi-center collaboration algorithm that ensures fair collaboration through bi-directional optimization. Extensive experiments demonstrate the efficiency and effectiveness of IMTAO in enhancing task assignment and improving collaboration fairness compared to baseline methods.**

***Index Terms*—multi-center collaboration, task assignment, spatial crowdsourcing**

I. Introduction

The development of smart devices and the growth of the sharing economy have contributed to the market of Spatial Crowdsourcing (SC) [1]–[6]. As a key component in SC, task assignment involves the process of matching and assigning spatial tasks, which are posted by task requesters on an SC platform, to workers capable of traveling to specific locations to complete them [7]–[9].

In the field of task assignment in SC, most existing studies focus on task assignment in an integrated area, where workers are free to move to various locations to perform accessible tasks [10]–[12]. While this scenario of task assignment is practical for many applications [13]–[16], we identify another scenario where an SC platform operates multiple distribution centers across a city, such as supermarket delivery (e.g., Freshippo and Walmart) and logistics service (e.g., JD Logistics and DoorDash). In this scenario, the platform divides the service area into regions, each managed by a distinct center, and tasks are assigned to centers based on their geographic locations. A center manages a group of workers working on the center's tasks, while tasks of all centers need to be completed before their respective expiration times. However, the dynamic nature of task distribution often leads to imbalances, where some centers experience a surplus of tasks (i.e., these centers have more tasks than available workers) while others may face a shortage (i.e., these centers have more available workers than tasks). To fill the gap between available workers and unassigned tasks, it is necessary for an SC platform to build collaboration among centers by strategically managing the workforce. This involves dispatching available workers from their located centers to those experiencing high task demand, contributing to the overall number of assigned tasks. In addition, ensuring fairness in collaboration is crucial when handling workforce transfer across centers. If some centers are treated unfairly during the multi-center collaboration, they may opt out of future collaboration, which harms the overall task assignment performance of the SC platform.

In this paper, we explore a collaborative task assignment problem across multiple centers within an SC platform, called Collaborative Multi-Center Task Assignment (CMCTA). Specifically, given a set of centers each with workers and tasks, CMCTA aims to optimize the task assignment across centers by maximizing the total number of assigned tasks while minimizing collaboration unfairness among centers through strategic inter-center workforce transfers. JD Logistics is a typical CMCTA application, which manages multi-center task assignments by allocating orders (i.e., tasks) based on inventory, distance, and urgency. When a center is overloaded, collaboration across centers is optimized through worker reallocation and route planning to ensure efficient deliveries.

While there are existing studies on cooperation in SC task assignments, our research diverges significantly in both problem settings and objectives. Previous studies [17], [18]

✉ Corresponding authors: Yan Zhao and Kai Zheng. Kai Zheng is with Yangtze Delta Region Institute (Quzhou), and School of Computer Science and Engineering, University of Electronic Science and Technology of China.

have focused on cooperation between different platforms that share the same service area and provide same services, e.g., ride-sharing platforms. These studies generally address inter-platform cooperation to enhance overall service efficiency. In addition, Cheng et al. [17] and Zhao et al. [19] study the cooperation among workers, particularly in scenarios where tasks require collective effort from multiple workers for completion. These studies concentrate on the dynamics of worker cooperation within the same service area. Recently, Zhao et al. [20] have explored task assignments across centers, where tasks located in one center can be assigned to another. However, they do not consider collaboration among different centers, which may result in poor worker resource utilization. Different from previous studies, we focus on the collaboration among centers through inter-center workforce transfers on an SC platform. The centers responsible for task execution remain fixed, while only workers can be dispatched between centers to facilitate task completion.

We provide an example in Fig. 1 to illustrate the CMCTA problem within a service area divided into three regions (e.g., three universities in Chengdu city), which are managed by their own distribution centers, i.e., c_1, c_2, and c_3. The regions' boundaries are generated by a Voronoi diagram [21], presented by dashed grey lines. A total of six tasks are required for delivery (i.e., $s_1, s_2, \cdots, s_6$). An intuitive and typical method is to assign tasks to suitable workers within each center independently, without considering inter-center collaboration. This method generates a center-independent task assignment, $\{(w_1, s_1), (w_3, s_2), (w_4, s_5)\}$, with worker w_2 from center c_1 unused. We quantify unfairness by the difference of task assignment ratios ρ_c across centers, where the ratio ρ_c is the proportion of tasks assigned by center c to its total tasks. In the center-independent task assignment, center c_1 assigns all tasks (i.e., s_1), achieving a ratio of $\rho_1 = 1.0$. Centers c_2 and c_3 achieve ratios of 0.5 and 0.33, respectively. Consequently, this task assignment completes three of the six tasks (i.e., s_1, s_2, and s_5), resulting in a collaboration unfairness score of 0.45 (computed by Eq. 3). However, if we dispatch surplus workers (e.g., w_2) to assist in other centers, we can potentially enhance the total number of tasks assigned and reduce collaboration unfairness. Specifically, by dispatching worker w_2 to center c_3 and reassigning tasks in c_3, we can get a task assignment $\{(w_1, s_1), (w_2, s_5), (w_3, s_2), (w_4, s_6)\}$. The total number of assigned tasks is increased from three to four, updating the task assignment ratios for centers c_1, c_2, and c_3 to 1.0, 0.5, and 0.67, respectively. This task assignment reduces the collaboration unfairness score from 0.45 to 0.33 and assigns more tasks simultaneously.

Based on the above motivations, we propose a novel framework for collaborative task assignment across multiple centers, namely Iterative Multi-center Task Assignment and Optimization (IMTAO). This framework is structured into two phases: center-independent task assignment and inter-center workforce transfer. In the first phase, we focus on maximizing task assignments based on the distribution of workers and tasks. Specifically, we deploy a sequential task assignment

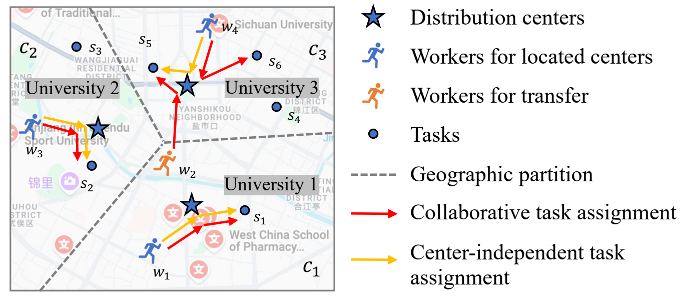


Fig. 1. A toy example for multi-center collaboration

algorithm that prioritizes efficiency over the complexities of an optimal task assignment approach, aiming to maximize the number of assigned tasks independently in each center. In the second phase, we apply principles of game theory to optimize inter-center workforce transfer for collaboration, aiming to improve collaboration fairness, i.e., reduce the difference in task assignment ratios across centers. Specifically, we transfer the CMCTA problem into a multi-player game, treating each center as an individual player. A game-theoretic multi-center collaboration algorithm is proposed to utilize the best-response mechanism and achieve fair collaboration across centers, where a bi-directional optimization strategy is incorporated. The bi-directional optimization strategy enables continuous adjustments to workforce distribution and task assignment. It involves iteratively dispatching available workers to support task assignments at other centers, followed by reapplying the sequential task assignment algorithm in each center.

In this paper, we make the following contributions.

- We identify and study a novel task assignment problem in spatial crowdsourcing, called collaborative multi-center task assignment (CMCTA).
- We propose a sequential task assignment algorithm to enhance efficiency of the center-independent task assignment.
- We propose a game-theoretic multi-center collaboration algorithm for inter-center workforce transfer, incorporating a bi-directional optimization process that simultaneously considers workforce transfer and task assignment to achieve fair and balanced collaboration.
- We conduct extensive experiments that illustrate the effectiveness and efficiency of the proposed framework. The convergence of the game-theoretic algorithm is also analyzed.

The remainder of this paper is organized as follows. The preliminary concepts and problem statement are introduced in Sec. II. We then present the framework overview in Sec. III. The proposed center-independent task assignment and inter-center workforce transfer methods are given in Sec. IV and Sec. V, respectively, followed by the experimental results in Sec. VI. Sec. VII surveys the related work, and Sec. VIII concludes this paper.

II. Problem Statement

In this section, we provide the necessary preliminaries and formally define the problem.

Definition 1 (Distribution Center): A distribution center is represented as $c = (l, S, W)$, consisting of a center location

$c.l$, a set of tasks $c.S$ assigned to center c, and a set of workers $c.W$ managed by center c.

An SC platform has a set of distribution centers, denoted as $C = \{c_1, c_2, \ldots, c_{|C|}\}$. Each distribution center functions as a hub for assigning delivery tasks to workers. When the context is clear, we use "distribution center" and "center" interchangeably. In this work, the delivery region for each center is determined by utilizing a Voronoi graph [21], where the service area (e.g., a city) is divided into non-overlapping regions based on the proximity to each center. This ensures that each center manages an independent delivery region within the overall service area. We will give more details about the Voronoi graph technique in Sec. IV-A.

Definition 2 (Worker): A worker is denoted as $w = (c, l, maxT)$, where $w.c$ indicates a distribution center that worker w primarily works for, $w.l$ is the current location of w, and $w.maxT$ denotes the maximum number of tasks that w can be assigned. In our work, workers can also be dispatched to other centers to assist when needed.

Definition 3 (Spatial Task): A spatial task is denoted as $s = (c, l, e, r)$, including a distribution center $s.c$ that s is associated with, a delivery location $s.l$, a task expiration deadline $s.e$, and a reward $s.r$ offered by the task requester.

For example, a spatial task s is requested by a task requester from a distribution center and needs to be delivered to a specific location (e.g., the requester's home) before its expiration time (e.g., 11 a.m.). A worker who completes the task will receive a reward offered by the task requester. We use the single-task assignment model [22], where each task is assigned to only one worker. Without loss of generality, we assume that the processing time of tasks (e.g., food delivery and goods delivery) is zero. Assuming that a worker can head to the next task's location upon finishing the current task, we focus on the spatial task assignment aspects of the problem. When the task process time is considered, the proposed framework can easily adapt to the scenario by incorporating the process time into workers' travel time.

Definition 4 (Task Delivery Sequence): Given a worker w and a set of tasks S_w assigned to worker w in a center c (either the located center or another center where the worker assists), we define the delivery sequence on S_w as $R(S_w)$, representing the order in which w completes tasks in S_w. The time taken by w to finish a task $s_i \in S_w$ can be calculated as follows:

$$t_{w,c,R}(s_i.l) = \begin{cases} tt(w.l, c.l) + tt(c.l, s_i.l) & \text{if } i = 1, \\ t_{w,c,R}(s_{i-1}.l) + tt(s_{i-1}.l, s_i.l) & \text{if } i > 1, \end{cases} \quad (1)$$

where $tt(l_a, l_b)$ is the travel time from location l_a to location l_b, and $t_{w,c,R}(s_i.l)$ is the total time required to complete task s_i after worker w picks up deliveries at center c and proceeds according to the delivery sequence $R(S_w)$.

Definition 5 (Valid Task Delivery Set): For worker w, a set of tasks S_w is considered a valid task delivery set (VTDS), denoted as $VTDS(w)$, if each task $s \in S_w$ can be completed before its expiration time, i.e., $\forall s \in S_w, t(s.l) \leq s.e$). When more than one delivery sequences exist for a given $VTDS(w)$, we choose the one with the minimal travel time.

Given the random emergence of tasks in each center, it is necessary to dispatch surplus workers from their located centers to nearby, busier centers for assistance while retaining workers in their located centers whenever possible. Hence, worker dispatching is managed at the center level, called *inter-center workforce transfer*, where *source centers* send out surplus workers and *recipient centers* receive additional workers from other centers.

Definition 6 (Inter-center Workforce Transfer): Inter-center workforce transfer refers to the reallocation of workers from one center, known as the source center c_i^s, to another center, known as the recipient center c_j^r, to assist with task demand. We denote this dispatch as a tuple (c_i^s, c_j^r, w), where w denotes the worker being sent from c_i^s to c_j^r.

As the example shown in Fig. 1, center c_1 is a source center, while center c_2 and c_3 are two recipient centers. The workforce transfer of w_2 from center c_1 to c_3 can be referred to as (c_1, c_3, w_2).

Definition 7 (Borrowing Worker Set): For recipient centers, we denote the set of workers that a center c_j^r borrows from other centers as $BWS(c_j^r)$, which consists of workforce transfer tuples.

A recipient center may have a few candidate BWS, which serve as the strategies of a center in the game-theoretic multi-center collaboration in Sec. V. For example, there are two different BWS for center c_3 in Fig. 1, $BWS(c_3)_1 = \{(c_1, c_3, w_2)\}$, and $BWS(c_3)_2 = \emptyset$.

Definition 8 (Center-specific Spatial Task Assignment): Given a distribution center c with a set of workers and tasks to be assigned, a spatial task assignment in center c is denoted as $A(c)$, which consists of a set of $(w, VTDS)$ pairs. Due to the inter-center workforce transfer, a source center c_i^s uses some of its workers $c_i^s.W$, while a recipient center c_j^r uses its workers $c_j^r.W$ and workers in the $BWS(c_j^r)$.

Definition 9 (Task Assignment Ratio): To evaluate the level of task assignment in each center c_i, we define the task assignment ratio ρ_i in center c_i as follows.

$$\rho_i = \frac{|A(c_i).S|}{|c_i.S|} \quad (2)$$

where $|A(c_i).S|$ and $|c_i.S|$ are the total number of assigned tasks and the total number of tasks in center c_i, respectively.

Definition 10 (Collaboration Unfairness): To capture the difference in task assignment ratios across centers and reflect the collaboration unfairness among them, we define the collaboration unfairness metric U_ρ, calculated in Eq. 3.

$$U_\rho = \frac{\sum_{1 \leq i \leq |C|, 1 \leq j \leq |C|, i \neq j} |\rho_i - \rho_j|}{|C|(|C| - 1)} \quad (3)$$

where ρ_i is the task assignment ratio in center c_i, $|C|$ is the total number of centers and higher values of U_ρ indicate greater unfairness in the distribution of tasks.

Collaborative Multi-Center Task Assignment (CMCTA) Problem Statement. Given a set of centers C, each with a set of tasks to be assigned, and a set of workers on an SC platform, the CMCTA problem aims to find a task

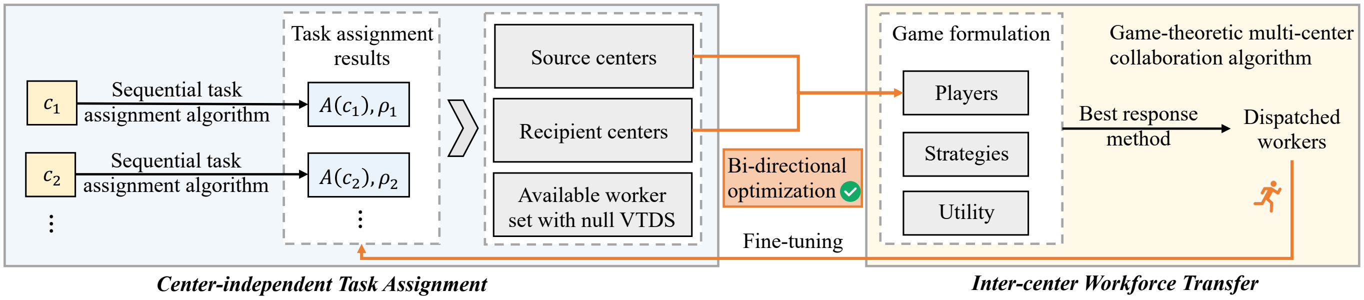


Fig. 2. Framework overview of Iterative Multi-center Task Assignment and Optimization (IMTAO)

assignment $\mathcal{A}_{opt} = \{A(c)\}_{c \in C}$ for all the centers by inter-center workforce transfer, which satisfies the following goals:

1) primary optimization goal: maximize the total number of assigned tasks, i.e., $\forall \mathcal{A}_i \in \mathbb{A}$ $(|\mathcal{A}_i.S| \leq |\mathcal{A}_{opt}.S|)$, where $\mathbb{A}$ denotes all possible assignments and $|\mathcal{A}_i.S|$ denotes the number of assigned tasks in task assignment $\mathcal{A}_i$; and

2) secondary optimization goal: minimize the collaboration unfairness by reducing U_ρ.

III. Framework Overview

The Iterative Multi-center Task Assignment and Optimization (IMTAO) framework contains two phases, center-independent task assignment and inter-center workforce transfer, as shown in Fig.2.

Center-independent Task Assignment. In this phase, we handle each center's task assignment independently and focus on the primary optimization goal (i.e., maximizing the total number of assigned tasks). The task assignment process is inherently complex due to both spatial and temporal constraints, as well as the limited availability of workers. To address the computational complexity of optimal task assignments, we introduce a sequential task assignment algorithm that prioritizes efficiency over an exhaustive search for the optimal solution. Specifically, the algorithm iteratively assigns tasks to workers $c.W$ at each center c, and generates task assignment $A(c)$ without the need for intensive global optimization.

However, since centers operate independently, task assignments ignore the worker availability across centers, which potentially leads to underutilization or overloading of workers at certain centers. As a result, some centers act as source centers with unused workers, while others act as recipient centers facing a worker shortage. To address this challenge, we define a task assignment ratio ρ_c for each center, and identify an available worker set containing workers with a null VTDS (i.e., unused workers with no tasks assigned) which can be dispatched in the next phase.

Inter-center Workforce Transfer. In the following phase, we aim to narrow the demand-supply gap (i.e., tasks and workers) across centers by establishing workforce transfer among centers. The main objective is to minimize collaboration unfairness, which arises when recipient centers are not allocated a reasonable share of workers. The challenge here is twofold: conflicts among recipient centers for available workers and the need for fair workforce transfer. Since there may be more than one recipient center, conflicts can arise as they compete for workers in the available worker set. To achieve the fairness-oriented goal, we propose a game-theoretic multi-center collaboration algorithm, modeling each center as a player. The borrowing worker set $BWS(c)$ is the strategy set of center c, and the utility of center c is based on punishment for unfairness. Due to the challenges in balancing fairness and task completion, we apply a best-response mechanism to carefully dispatch workers among centers. Workers are transferred iteratively to recipient centers to reduce the imbalance in task assignment ratios ρ.

Instead of decoupling the two phases of IMTAO, we use bi-directional optimization and perform the two phases iteratively to further enhance the task assignment performance. Specifically, when a recipient center c receives a dispatched worker, we fine-tune the task assignment $A(c)$ for assigning a VTDS to the dispatched worker. Then we update the source and recipient centers generated in the first phase, which are fed into the second phase as players. The task reassignment helps improve the task assignment ratio and narrow the demand-supply gap, thereby contributing to the optimization of both the primary and secondary goals.

IV. Center-Independent Task Assignment

The center-independent task assignment phase includes two modules, service area partition and spatial task assignment in centers independently. Specifically, we utilize Voronoi diagrams to partition the whole service area into $|C|$ delivery regions, each with a distribution center. Therefore, the center-independent task assignment is decomposed into $|C|$ individual spatial task assignments. Each center strives to maximize its total number of assigned tasks, which is independent of other centers in this phase. We illustrate the complexity of spatial task assignment problem in a center and then introduce a sequential task assignment algorithm to solve it efficiently.

A. Service Area Partition

As described in Sec. II, a distribution center c consists of a set of tasks $c.S$ and a set of workers $c.W$. Considering practical applications, such as logistics or food delivery services, it is common practice to assign workers and tasks to the nearest distribution center. This strategy can significantly enhance delivery efficiency by reducing travel distances and times, thereby improving service responsiveness and minimizing travel costs.

Algorithm 1 Voronoi-based Service Area Partition

1: **Input:** C, S, W
2: **Output:** $C.S, C.W$
3: $C.S \leftarrow \emptyset$
4: $C.W \leftarrow \emptyset$
5: **for** each center $c_i \in C$ **do**
6: $\quad c_i.S \leftarrow \{s \in S : c_i$ is the nearest center to $s\}$
7: $\quad c_i.W \leftarrow \{w \in W : c_i$ is the nearest center to $w\}$
8: $\quad S \leftarrow S \setminus c_i.S$
9: $\quad W \leftarrow W \setminus c_i.W$
10: **end for**
11: $C.S \leftarrow \bigcup_{c_i \in C} c_i.S$
12: $C.W \leftarrow \bigcup_{c_i \in C} c_i.W$
13: **return** $C.S, C.W$

To implement this strategy effectively, we employ Voronoi diagrams [21], [23] to partition the service area into distinct regions. A Voronoi diagram partitions a plane into regions based on the distance to a set of predefined sites, which creates regions termed Voronoi cells. Each cell encompasses all points that are closer to its predefined site than to any other site. Applying Voronoi diagrams in the service area partition, we regard distribution centers as sites, while taking workers and tasks as points. As illustrated in Algo. 1, we effectively divide a service area into $|C|$ regions and assign tasks and workers to their respective region centers (lines 5–10). Algo. 1 receives centers, tasks, and workers as input, and generates the tasks and workers of each center.

B. Center-independent Spatial Task Assignment

In an SC platform that operates multiple centers within a service area, a straightforward approach to task assignment is to treat each center independently [11], [24]. After the service area partition, each center has its own tasks and workers to be assigned. In this section, we give the hardness analysis of the spatial task assignment problem for a specific center, defined in Definition 8, and then introduce a sequential task assignment algorithm, which is designed to efficiently assign tasks within an individual center.

Hardness Analysis. The task assignment problem in SC is inherently classified as NP-hard, as highlighted in studies by Cheng et al. [17] and Zhao et al. [25]. This complexity also extends to task assignments with a distribution center, which reduces from the problem of Capacitated Vehicle Routing Problem with Time Windows (CVRPTW), a well-known NP-hard problem. In the CVRPTW problem, the objective is to optimize vehicle routes to serve a group of customers while satisfying the capacity constraints of the vehicles and the time constraints of customers. Now we reduce an instance of the CVRPTW problem to the spatial task assignment problem with a center. We can regard the departure point for vehicles as the distribution center c, each customer destination as a spatial task s, and each vehicle as a worker w. The vehicle capacity limitation and the time window for customers in CVRPTW can move to the limited number of tasks of workers $w.maxT$ and the expiration times of tasks $s.e$, respectively. If we can solve the task assignment problem with a distribution center in polynomial time, we can solve any instance of CVRPTW by transforming it into a corresponding task assignment instance and then solve the instance of CVRPTW in polynomial time as well. This contradicts the fact that the CVRPTW problem is NP-hard [26], which means that the task assignment problem instance cannot be solved in polynomial time. Hence, we can affirm the NP-hard nature of spatial task assignment with a distribution center.

Algorithm 2 Sequential Task Assignment

Input: $c = (l, S, W)$
Output: $A(c)$, $c.W_{left}$, $c.S_{left}$
Initialize $A(c) \leftarrow \emptyset$, $c.W_{left} \leftarrow \emptyset$, $c.S_{left} \leftarrow c.S$
Sort $c.W$ in the descending order of $tt(w.l, c.l)$
for each worker $w \in c.W$ **do**
$\quad S_w \leftarrow \emptyset$
$\quad t_{w,c,R}S_w = tt(w.l, c.l)$
$\quad w.l \leftarrow c.l$
$\quad$ **while** $t_{w,c,R}S_w < e$ and $|S_w| < w.maxT$ **do**
$\quad\quad$ Get the nearest unassigned task $s_i \in c.S_{left}$ to w
$\quad\quad$ **if** $t_{w,c,R}S_w + tt(w.l, s_i.l) < e$ **then**
$\quad\quad\quad c.S_{left} \leftarrow c.S_{left} \setminus \{s_i\}$
$\quad\quad\quad S_w \leftarrow S_w \cup \{s_i\}$
$\quad\quad\quad w.l \leftarrow s_i.l$
$\quad\quad$ **end if**
$\quad$ **end while**
$\quad A(c) \leftarrow A(c) \cup (w, VTDS(w))$
$\quad$ **if** S_w is $\emptyset$ **then**
$\quad\quad c.W_{left} \leftarrow c.W_{left} \cup \{w\}$
$\quad$ **end if**
end for
return $A(c), c.W_{left}, c.S_{left}$

Sequential Task Assignment Algorithm. To enhance the efficiency of task assignment within a center, we propose a sequential task assignment algorithm that considers both spatial and temporal constraints, significantly speeding up the process of task assignment to workers. Previous task assignment studies [17], [25] generate all candidate tasks set for each worker and subsequently resolve assignment conflicts where tasks may overlap among workers, where the complexity reaching $O(2^{|c.S|} \cdot |c.S|^3)$ and $|c.S|$ denotes the number of tasks in center c. However, we recognize that the primary difference among workers lies in their arrival times at the center for picking up deliveries. This motivates us to achieve the optimization goal (i.e., maximize the number of assigned tasks) in a center by adopting a more streamlined approach. Specifically, by assigning tasks sequentially, we generate the task delivery sequence $R(S_w)$ for each worker w individually.

Algo. 2 illustrates the sequential task assignment algorithm, which takes the location, tasks, and workers of a center $c = (l, S, W)$ as input and task assignment $A(c)$, unused workers $c.W_{left}$, and unassigned tasks $c.S_{left}$ as output. It first initializes task assignment $A(c)$ and unused workers $c.W_{left}$ as an empty set, and unassigned tasks $c.S_{left}$ as the whole tasks $c.S$ (line 3). Algo. 2 consists of two primary sequences: the worker sequence and the task delivery sequence assigned to each worker. We observe that marginal workers (i.e., workers located far from the center) are inferior in task assignment due to having less time for delivery (i.e., $e - tt(w.l, c.l)$, where e denotes the expiration time of tasks and $tt(w.l, c.l)$ denotes the travel time from the worker's location to the center's location. If no tasks are achievable within their constraints, these marginal workers are left unutilized, reducing the overall number of assigned tasks and impacting the optimization goal negatively. To address this issue, we strategically prioritize marginal workers in the task assignment process. Workers are sorted by their distances from the center, with those farthest away being given precedence (line 4).

Once a worker is selected, we then generate a task sequence for the worker (lines 6–16). Given that the task execution time is assumed to be negligible, we focus on minimizing the travel durations between consecutive tasks in the task sequence. Task assignments start by calculating the travel time from the worker's initial location to the center (lines 7–8). Following the arrival at the center, the nearest unassigned tasks are assigned to the worker sequentially according to the worker's current location (lines 9–16). The process continues until the worker reaches her task capacity or the expiration time for task completion, which means that the task set of the generated task sequence $R(S_w)$ is a $VTDS(w)$. The pair of $(w, VTDS(w))$ is then add in task assignment $A(c)$ (line 17). After task assignment, we can get some unused workers $c.W_{left}$ and unassigned tasks $c.S_{left}$ in center c (lines 18–20). Overall, this sequential task assignment algorithm ensures optimal use of each worker's time and capabilities by minimizing unnecessary travel, thereby maximizing the number of assigned tasks within the time constraints.

C. Complexity Analysis

We analyze the complexity of the sequential task assignment algorithm. The complexity of sorting workers according to the descending sort is $O(|c.W| \cdot log|c.W|)$, where $|c.W|$ denotes the number of workers in center c. The complexity of getting the task sequence $R(S_w)$ for a worker w is $O(w.maxT \cdot |c.S|)$ since a worker may need to check all task points to find the nearest unassigned task, where $w.maxT$ denotes the maximum number of deliveries for worker w. Hence, the total computational complexity is $O(|c.W| \cdot log|c.W| + |c.W| \cdot w.maxT \cdot |c.S|)$.

V. Inter-center Workforce Transfer

In this section, we design a game-theoretic approach to solve the inter-center workforce transfer problem, while taking the secondary optimization goal (i.e., the fairness of collaboration) into consideration. We introduce the approach in terms of motivation, unfairness punishment, game formulation, and game-theoretic multi-center collaboration.

A. Motivation

Following the center-independent task assignment phase, there remain some inefficiencies, such as unused workers $c.W_{left}$ in source centers and unassigned tasks $c.S_{left}$ in recipient centers. This demand-supply gap lays the foundation of multi-center collaboration, where unused workers can be dispatched to recipient centers to deliver unassigned tasks.

Fairness Objective in Multi-center Collaboration. While multi-center collaboration can help increase the total number of assigned tasks, it introduces the risk of collaboration unfairness. In the center-independent phase, each center c equips an initial task assignment $A(c)$ and achieves a task assignment ratio ρ_c, which is the proportion of tasks assigned relative to the total tasks in center c. However, the objective of multi-center collaboration is not merely to increase task completion but also to preserve or reduce unfairness across centers. To achieve this balance, it is crucial to manage the multi-center collaboration by reasonably dispatching available workers to recipient centers in a fair and effective manner.

Core Issues in Collaboration. The fundamental essence of multi-center collaboration is that a recipient center needs to choose a set of available workers to deliver tasks while interacting with other centers for the division of available workers. Two issues occur in the multi-center collaboration.

(1) *Center Selection.* Among multiple recipient centers needing additional workers, which center should be prioritized to reduce the unfairness of collaboration?

(2) *Worker Dispatching.* Once a recipient center is selected, which workers from the pool of available workforce should be dispatched to maximize efficiency and fairness?

No existing method can effectively solve these collaboration issues for two main reasons. First, the number of workers required at each center is unknown since task assignment depends heavily on the positions of workers and tasks. Hence, the CMCTA problem cannot be reduced to a maximum flow problem, which requires prior knowledge of surplus and shortage of workers. Second, the CMCTA problem is more complex than typical spatial task dispatching problems, e.g., taxi dispatching, where a taxi serves a single order, while a worker can deliver multiple tasks from a center. To address these issues, we design a game-theoretic method. By modeling the interaction between centers as a multi-player game, each center acts as an independent player with a workforce transfer strategy (i.e., the borrowing worker set $BWS(c)$). This allows us to optimize the fairness of task assignment ratios by deciding how to dispatch workers to recipient centers. In the rest of the paper, we use "player" and "center" interchangeably.

B. Unfairness Punishment

Focusing on fairness across collaboration, we adopt the concept of unfairness punishment [27]–[29] to quantify imbalances in task assignment ratios among centers. For example, if a collaboration strategy significantly increases the total number of tasks assigned but fails to support centers with low task assignment ratios, it may potentially harm future cooperation in the SC platform.

In the game of multi-center collaboration, we define the Utility of Unfair Punishment (UUP) as follows.

$$UUP(c_i, BWS(c_i)) = \rho_i - \frac{\sum_{1\le j\le |C|, j\ne i} \rho_j}{|C|-1} \quad (4)$$

where $UUP(c_i, BWS(c_i))$ represents the utility for center c_i with a workforce transfer strategy $BWS(c_i)$. The utility calculation contains two parts: the first part ρ_i is the task assignment ratio of c_i assuming it adopts $BWS(c_i)$, and the second part quantifies the average task assignment ratio of all other centers, excluding center c_i. This utility helps assess how center c_i's ratio ρ_i compares to the average of the other centers' ratios, thereby representing the unfairness punishment, i.e., the degree of unfairness.

C. Game Formulation

In this section, we introduce the game formulation and theoretical analysis of Nash Equilibrium to illustrate the stability of multi-center collaboration through workforce transfer.

Game Components. The game of multi-center collaboration can be formulated as an $|C|$-player strategic game, denoted as $G = (C, \mathbb{ST}, \mathbb{U})$, consisting of the following components:

(1) *Players.* $C = \{c_1, c_2, \ldots, c_{|C|}\}$ represents a finite set of centers involved in the game, each as an independent player.

(2) *Strategy Spaces.* The strategy space of the game, $\mathbb{ST} = \cup_{1\le i\le|C|} ST_i$, is the union of all strategy sets across players (i.e., all possible workforce transfer decisions across centers). Next, $ST_i = \{\mathbb{BWS}(c_i)\}$ is a finite set of all candidate worker transfer strategies for center c_i, which contains c_i's all possible BWS sets, such as borrowing a specific set of workers or choosing not to borrow any workers at all. A joint strategy for the game is defined as $\vec{st} = (st_1, st_2, \cdots, st_|C|) \in \mathbb{ST}$, where $st_i \in ST_i$ is the strategy selected by player c_i.

(3) *Utility Functions.* The utility functions are defined by $\mathbb{U} = \cup_{1\le i\le|C|} U_i$, where U_i is the utility function of player c_i. The utility for each center based on the joint strategy $\vec{st}$ is calculated as follows.

$$U_i(\vec{st}) = UUP(c_i, BWS(c_i)) \quad (5)$$

where $U_i(\vec{st})$ measures the utility of center c_i when the workforce transfer strategy $BWS(c_i)$ is applied as part of the joint strategy $\vec{st}$.

Establishment of Exact Potential Game. We prove that the multi-collaboration game conforms to an exact potential game (EPG), which has at least one pure Nash Equilibrium (NE). EPG is a specific type of game in game theory characterized by a direct relationship between a player's utility changes and a global potential function, which is defined as follows.

Definition 11 (Exact Potential Game): A strategic game $G = (C, \mathbb{ST}, \mathbb{U})$ is an EPG if there exists an exact potential function Φ, such that any change in a player's strategy results in an identical change in both the player's utility and the potential function, as shown below:

$$U_i(st'_i, \vec{st}_{-i}) - U_i(st_i, \vec{st}_{-i}) = \Phi(st'_i, \vec{st}_{-i}) - \Phi(st_i, \vec{st}_{-i}) \quad (6)$$

where st_i and st'_i are the strategies chosen by player c_i before and after a workforce transfer strategy $BWS(c_i)$ change, respectively, $\vec{st}_{-i}$ is the joint strategy of other players except for player c_i, and Φ is the exact potential function to quantify the performance of the entire game.

Lemma 1: The multi-center collaboration game is an exact potential game which has at least one pure Nash Equilibrium.

Proof 1: We define the exact potential function in the multi-center collaboration game as follows.

$$\Phi(\vec{st}) = \sum_{1\le i\le|C|} U_i(\vec{st}) = \sum_{1\le i\le|C|} UUP(c_i, BWS(c_i)) \quad (7)$$

where the exact potential function $\Phi(\vec{st})$ denotes the sum of utilities of all centers in C. Now we can obtain:

$$\begin{aligned}
&\Phi(st'_i, \vec{st}_{-i}) - \Phi(st_i, \vec{st}_{-i}) \\
&= (UUP(c_i, BWS'(c_i)) + \sum_{1\le j\le|C|, i\ne j} UUP(c_j, BWS(c_j))) \\
&\quad - (UUP(c_i, BWS(c_i)) + \sum_{1\le j\le|C|, i\ne j} UUP(c_j, BWS(c_j))) \\
&= UUP(c_i, BWS'(c_i)) - UUP(c_i, BWS(c_i)) \\
&= U_i(st'_i, \vec{st}_{-i}) - U_i(st_i, \vec{st}_{-i})
\end{aligned} \quad (8)$$

where the BWS selected by player c_i before and after the strategy change are $BWS(c_i)$ and $BWS'(c_i)$, respectively. Since the multi-center collaboration game meets the requirements of EPG in Definition 11, it is an EPG with at least one pure Nash Equilibrium.

D. Game-Theoretic Multi-Center Collaboration

In this section, we utilize the best-response mechanism [30], [31] to guide the multi-center collaboration game towards a pure Nash Equilibrium (NE). The best-response mechanism involves players continuously revising their strategies in reaction to the most recent strategies of other players. Each player selects the strategy that maximizes their personal utility given the current strategic environment. Importantly, if the game is an exact potential game, the iterative strategy updates facilitated by the best-response mechanism not only converge to a local maximum of the potential function, but also inherently lead to a Nash Equilibrium for the game. We propose a multi-center collaboration algorithm that leverages the best-response mechanism to solve the two core issues mentioned in Sec. V-A. The algorithm includes three main steps, worker transfer strategy updating, task assignment updating, and recipient center selection.

Worker Transfer Strategy Updating. The input of Algo. 3 is tasks $C.S$, workers $C.W$, unused workers $C.W_{left}$, and task assignment $\mathcal{A} = \{A(c)_{c\in C}\}$ in the first phase for all centers; while the output is task assignment $\mathcal{A}$ of all centers after the inter-center workforce transfer. We first identify recipient centers as C' and their worker transfer strategies $\mathbb{BWS}(c)$ based on the available workers (lines 3–10). The strategy for each recipient center $BWS(c)$ is initialized as no worker transferring in it (line 6). We use η to denote the iteration of the game, which is set as $\eta = 1$ initially (line 11).

Algorithm 3 Game-Theoretic Multi-Center Collaboration

1: **Input:** $C.S$, $C.W$, $C.W_{left}$, $\mathcal{A}$
2: **Output:** $\mathcal{A}$
3: $C' \leftarrow C$
4: **for** each center c_i **do**
5:   **if** $\rho_i < 1$ **then**
6:     Obtain $\mathbb{BWS}(c_i)$ and $BWS(c_i) \leftarrow \emptyset$
7:   **else**
8:     $C' \leftarrow C' - c_i$
9:   **end if**
10: **end for**
11: $\eta \leftarrow 1$
12: **repeat**
13:   Pick a center $c_i \in C'$ with $\arg\min \rho_i$
14:   Get the best-response $BWS(c_i)$ from $\mathbb{BWS}(c_i)$ with one more dispatched worker w_{move} in $C.W_{left}$
15:   Check the ratio ρ_i^{η} of task assignment $A(c_i)$ with workers $c_i.W \cup \{w \in BWS(c_i)\}$ in Algo. 2
16:   **if** $\rho_i^{\eta} > \rho_i^{\eta-1}$ **then**
17:     $st_i \leftarrow BWS(c_i)$
18:     Update the task assignment $A(c_i)$ in $\mathcal{A}$
19:     $C.W_{left} \leftarrow C.W_{left} \setminus \{w_{move}\}$
20:   **else if** $\rho_i^{\eta} \le \rho_i^{\eta-1}$ **then**
21:     $C' \leftarrow C' \setminus \{c_i\}$
22:   **end if**
23:   $\eta \leftarrow \eta + 1$
24: **until** $\vec{st}^{\eta} = \vec{st}^{\eta-1}$
25: **return** $\mathcal{A}$

Algo. 3 iteratively updates a selected center's strategy st_i by incorporating one newly dispatched worker w_{move} from the available worker set $C.W_{left}$ (lines 12–24). The strategy updating is based on the current joint strategies of other centers $\vec{st}_{-i}$ and maximizes the selected center's utility, which follows the best-response mechanism (line 14). The process of Algo. 3 is iterated until a Nash Equilibrium (NE) is achieved (line 24), defined as the state where no center can increase its utility by updating its own strategy independently.

Task Assignment Updating. Once the worker transfer strategy of a center $BWS(c)$ is updated, the task assignment of this center $A(c)$ also needs to be updated to incorporate the newly dispatched worker w_{move}. A straightforward approach is to simply assign unassigned tasks $c.S_{left}$ to worker w_{move}. We refer to this approach as the ***decomposed collaboration*** (DC), which leaves the initial task assignment from the first phase unchanged. However, this decomposed collaboration might not fully leverage the combined capabilities of both existing workers at the center (i.e., the original worker $c.W$ and previously dispatched workers in center c) and the newly dispatched worker w_{move}. Taking Fig. 1 as an example, if we use DC, only the three unassigned tasks (i.e., s_3, s_4, s_6) are considered for available worker w_2. However, due to the limitation of expiration time compared to the travel time, none of these unassigned tasks is reachable for worker w_2.

To more effectively utilize the workforce, we reapply the Algo. 2 to reassign tasks among all workers in a recipient center (i.e., $c_i.W \cup \{w \in BWS(c_i)\}$) (line 15). We term this approach as ***bi-directional collaboration***, where the workforce transfer in the phase of collaboration fine-tunes the results of the task assignment phases. This ensures a comprehensive redistribution that potentially maximizes efficiency and task completion rates. After reassigning tasks, we evaluate the task assignment ratio, ρ_c, in this center (lines 16–22). If ratio ρ_c is enhanced, we remove worker w_{move} from the pool of available workers and update the workforce transfer strategy $BWS(c)$ accordingly. If ratio ρ_c remains unchanged, we put worker w_{move} back to the available worker set and exclude center c from recipient centers, since no workforce transfer can enhance task assignment of center c (lines 20–21).

Recipient Center Selection. We observe that the selection of the recipient center in each iteration affects the collaboration results. For example, in Fig.1, if we use a random bi-directional collaboration method, the center c_2 might be selected first, with worker w_2 as the dispatched worker to center c_2. Consequently, the collaboration results are 4 assigned tasks, task assignment ratio in three centers as $[1.0, 1.0, 0.33]$, and collaboration unfairness as 0.45. While the total number of assigned tasks increase, collaboration unfairness remains unchanged compared to center-independent task assignments.

To enhance the efficiency and effectiveness of the game-theoretic multi-center collaboration, we strategically select recipient centers based on their current needs for assistance, rather than a random selection. For example, in Fig.1, the center c_3 is selected. Specifically, we select the recipient center with the lowest task assignment ratio at the current time (line 13), which is supported by the following two reasons. First, in the phase of multi-center collaboration, we want to maximize the total number of assigned tasks while also minimizing the collaboration unfairness. This strategy ensures that resources are allocated where they are most needed, thereby directly reducing imbalances in task assignment ratios. Second, from the view of the potential function and utility, this strategy contributes positively to the overall potential function of the system. In essence, by elevating the center with the lowest ratio, this strategy effectively raises the minimum value of UUP across centers.

E. Discussion

One key limitation is our assumption of task predictability [32]–[36], where tasks are typically pre-scheduled and known in advance, as seen in supermarket deliveries and logistics services. However, tasks may arrive dynamically or with uncertainty in real-world settings. Relaxing this assumption would introduce additional challenges but also enhance the model's adaptability. Another important assumption is worker compliance [37], [38], where we assume that tasks are relatively simple and that workers are fully available for assignments. In practice, workers may have varying participation levels or possess different skill sets, with certain tasks requiring specific expertise. Future work could explore multi-skilled worker assignments and integrate skill-based task matching to improve applicability in more diverse settings.

VI. Experimental Evaluation

A. Experimental Setup

Datasets. The experiments are conducted using both real and synthetic datasets to validate the robustness and scalability of our approach. (1) The real dataset utilized is the gMission dataset (referred to as GM), an open-source SC dataset [39]. In GM, each task is treated as a delivery point with a specific location, and similarly, each worker is associated with a geographical location. Since the GM dataset does not include predefined distribution centers, we simulate $|C|$ distribution centers by randomly generating their locations. (2) For the synthetic dataset (referred to as SYN), we build scenarios that mirror real-world settings but allow for controlled variability in key parameters. The locations of distribution centers, workers, and delivery points were uniformly generated within a 2D space bounded by $[0, 2000]^2$. The default setting is underlined in Table I.

Baselines. In the few related studies of task assignments with multi-centers, no study sets the same problem definition or proposes a method that can solve the CMCTA problem, with no suitable sota baseline. Instead, we compare several approaches as discussed in this paper.

In the phase of center-independent task assignment, we study the following algorithms.

- Seq. The sequential task assignment algorithm in Algo. 2 for center-specific task assignment.
- Opt. The optimal task assignment method achieves the highest number of assigned tasks in each center, which first generates all possible VTDS for each worker and then handles the conflicts in task assignment.

In the phase of multi-center collaboration, we study the following algorithms.

- BDC. BDC is the proposed bi-directional collaboration approach that fine-tunes the task assignment in the center-independent task assignment phase, with consideration of all workers in a recipient center.
- RBDC. RBDC is a bi-directional collaboration approach that randomly selects a recipient center in each iteration.
- DC. DC is an approach that merely considers the unassigned tasks for unused workers, which leaves the initial task assignment from the first phase unchanged.
- w/o-C. The approach excludes collaboration, activating only the center-independent task assignment phase.

Hence, combining the methods of these two phases, we compare the proposed methods Seq-BDC with other baselines (i.e., Seq-RBDC, Seq-DC, Seq-w/o-C, Opt-BDC, Opt-RBDC, Opt-DC and Opt-w/o-C).

Settings. All tasks share the same reward (i.e., $s.r$ = 1) and the same expiration time e. For simplicity, we assume that all workers share the same delivery speed and the capability of delivery task $w.maxT = 4$, which means a worker can deliver at most four tasks. We implement our methods and baselines in Python 3.8. All the experiments are conducted on Intel(R) Xeon(R) Silver 4214 CPU @ 2.20GHz, 256GB memory, and 4 NVIDIA GeForce RTX 3080.

TABLE I
Experiment Parameters

Parameter	Value
Number of tasks $\|S\|$ in GM	400, 500, 600, 700, 800
Number of tasks $\|S\|$ in SYN	400, 500, 600, 700, 800
Number of workers $\|W\|$ in GM	80, 90, 100, 110, 120
Number of workers $\|W\|$ in SYN	100, 125, 150, 175, 200
Number of centers $\|C\|$ in GM	20, 30, 40, 50, 60
Number of centers $\|C\|$ in SYN	20, 30, 40, 50, 60
Expiration time of tasks e (h) in GM	1.00, 1.25, 1.50, 1.75, 2.00
Expiration time of tasks e (h) in SYN	1.00, 1.25, 1.50, 1.75, 2.00

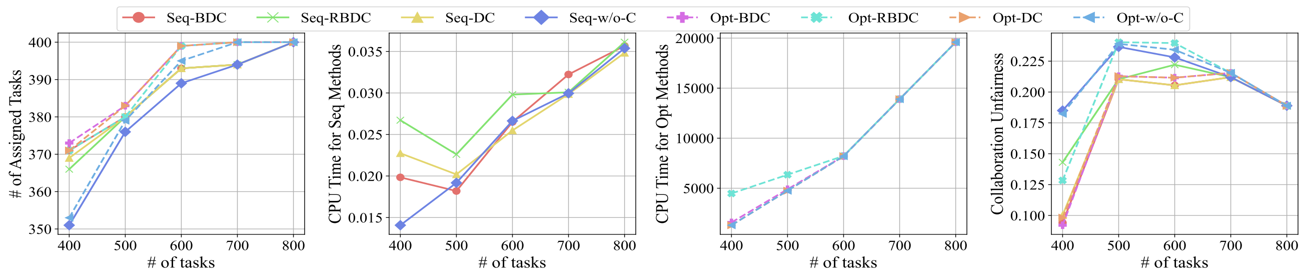
Evaluation Metrics. Three main metrics are used for the above methods. We use *the number of total assigned tasks* and *collaboration unfairness*, U_ρ, to reflect the primary and secondary optimization goals, respectively. We use *CPU time* to imply the efficiency of finding the task assignments in the multi-center collaboration scenario.

B. Experimental Results

The experiments explored the effect of varying key parameters on the effectiveness and efficiency of task assignment and collaboration fairness, including the number of tasks $|S|$, the number of workers $|W|$, the number of centers $|C|$, and the task expiration time e.

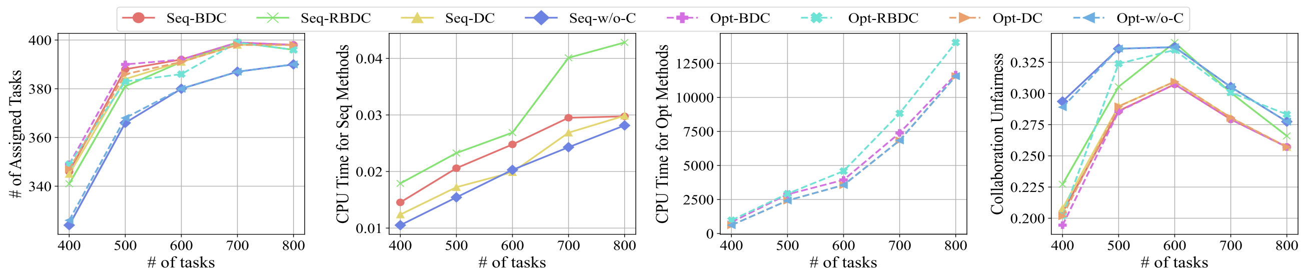
Effect of $|S|$. We study the effect of $|S|$. From Figs. 3 and 4, we can see that with the increase of $|S|$, the number of assigned tasks and CPU time of all methods exhibit a similar upward trend. The reason for this phenomenon is that a larger $|S|$ means that more tasks can be chosen during the task assignment process. Note that to clearly show the curve trend for CPU time in the figures, we show the CPU time values in two groups due to the significant difference in the magnitudes. From Figs. 3(c) and 4(c), methods with optimal task assignment (i.e., Opt-BDC, Opt-RBDC, Opt-DC, and Opt-w/o-C) cost thousands of seconds in calculation time, while other methods with sequential task assignment merely occupy tens of milliseconds. This illustrates that the sequential task assignment algorithm can efficiently process center-independent task assignments, and achieve a comparable number of assigned tasks to an optimal task assignment. As for the detailed performance of task assignment, the methods with multi-center collaboration (i.e., Seq-BDC, Seq-RBDC, and Seq-DC) enhance at most 5.6% and 6.7% task assignment than Seq-w/o-C on GM and SYN, respectively, supporting the effectiveness of enhancing the overall task assignment by workforce transfer. In particular, the proposed Seq-BDC continuously outperforms other baselines with sequential task assignment (i.e., Seq-RBDC, Seq-DC, Seq-w/o-C), in terms of the number of assigned tasks and collaboration unfairness.

In addition, multi-center collaboration can also reduce the unfairness of task assignment ratio across centers, as we can see Seq-BDC reduces at most 46% unfairness and reduces 30.5%-7.3% compared to Seq-w/o-C on GM and SYN, respectively. Notably, the Seq-RBDC sometimes performs worse than the Seq-w/o-C in collaboration unfairness, since it randomly picks a recipient center to receive assistance in each iteration, and potentially harms the collaboration unfairness.



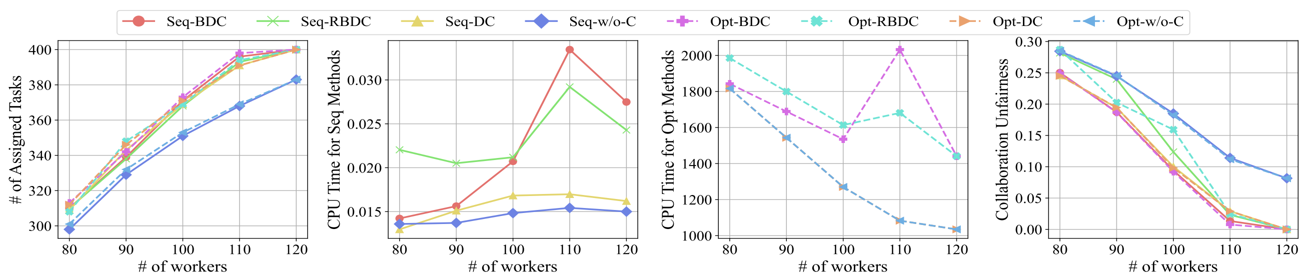
(a) Assigned Tasks (b) CPU Time for Seq Methods (c) CPU Time for Opt Methods (d) Collaboration Unfairness

Fig. 3. Effect of $|S|$ on GM



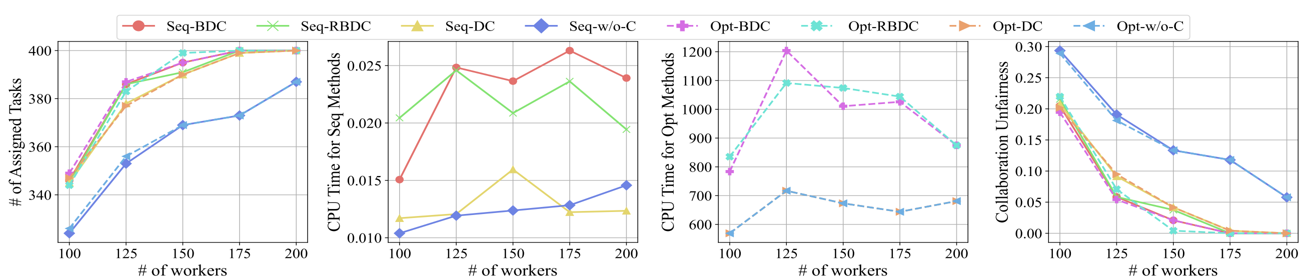
(a) Assigned Tasks (b) CPU Time for Seq Methods (c) CPU Time for Opt Methods (d) Collaboration Unfairness

Fig. 4. Effect of $|S|$ on SYM



(a) Assigned Tasks (b) CPU Time for Seq Methods (c) CPU Time for Opt Methods (d) Collaboration Unfairness

Fig. 5. Effect of $|W|$ on GM

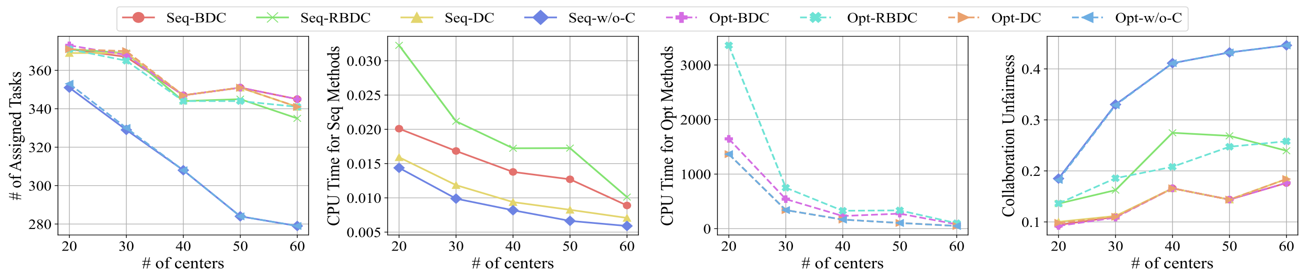


(a) Assigned Tasks (b) CPU Time for Seq Methods (c) CPU Time for Opt Methods (d) Collaboration Unfairness

Fig. 6. Effect of $|W|$ on SYM

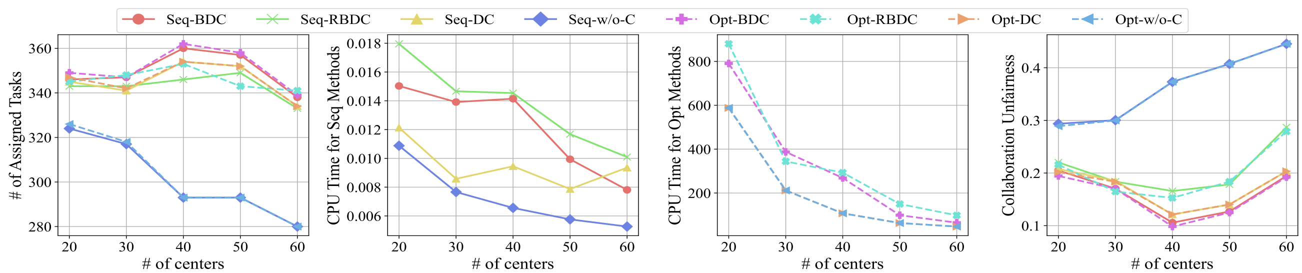
Effect of $|W|$. We study the effect of $|W|$ as shown in Figs. 5 and 6. With the increase in the number of workers, both task assignment and collaboration fairness perform better, where ultimately all tasks are assigned and collaboration unfairness reduces to zero with multi-center collaboration. The reason behind this phenomenon is that when the tasks remain the same, the large number of workers contributes to the center-independent task assignment and the inter-center workforce transfer. In this experiment of $|W|$, the Seq-DC is generally inferior to the Seq-BDC, which illustrates the importance of bi-directional optimization. Specifically, with bi-directional optimization between task assignment and workforce transfer, the center-specific task assignment achieves better performance since all workers are reassigned with a VTDS. On the contrary, the improvement of task assignment with decoupled collaboration is limited by the travel cost of available workers to unassigned tasks.

Effect of $|C|$. We study the effect of $|C|$, which is presented in Figs. 7 and 8. The gap between no collaboration baselines (i.e., Seq-w/o-C and Opt-w/o-C) and other collaboration baselines expands compared to tuning $|S|$ and $|W|$. Specifically, as the number of centers $|C|$ increases from 20 to 60, the number of assigned tasks with Seq-w/o-C reduces from 351 to 279, and from 324 to 280 on GM and SYN, respectively. Simul-



(a) Assigned Tasks (b) CPU Time for Seq Methods (c) CPU Time for Opt Methods (d) Collaboration Unfairness

Fig. 7. Effect of $|C|$ on GM



(a) Assigned Tasks (b) CPU Time for Seq Methods (c) CPU Time for Opt Methods (d) Collaboration Unfairness

Fig. 8. Effect of $|C|$ on SYM

taneously, the score of collaboration unfairness deteriorates from 0.18 to 0.44 and from 0.29 to 0.44 on GM and SYN, respectively. The Opt-w/o-C performs a similar degeneration as Seq-w/o-C, although it applies better center-independent task assignments than Seq-w/o-C. This degeneration in Seq-w/o-C and Opt-w/o-C is because as the $|C|$ increases, the demand-supply gap (i.e., the distribution tasks and workers) across centers also enlarges, where more available workers and unassigned tasks are left after the phase of center-independent task assignment. In addition, with more centers, the Seq-RBDC tends to cause approximately double collaboration unfairness compared to the Seq-BDC, since the random selection of recipient centers has potential risks the multi-center collaboration, especially with more centers.

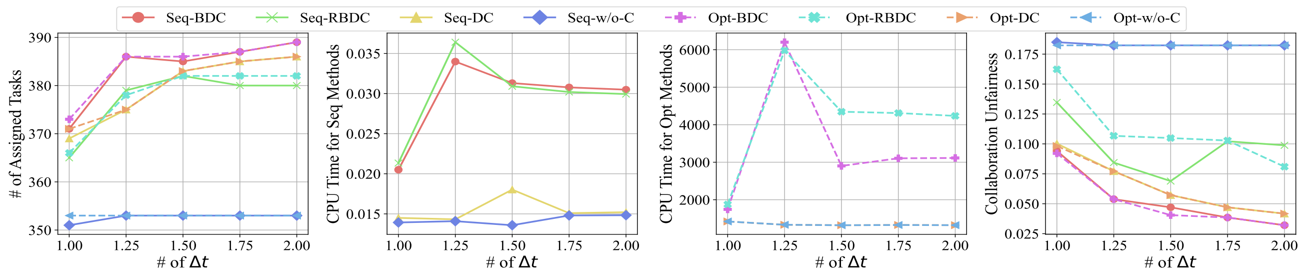
Effect of e. We study the effect of e, which is the expiration time for tasks. Not surprisingly, as shown in Figs. 9 and 10, the number of assigned tasks in no collaboration baselines (i.e., Seq-w/o-C and Opt-w/o-C) increases at first and then remains stable; while other baselines with multi-center collaboration exhibit an increasing trend. On the contrary, the performance of baselines in collaboration unfairness shows oppositely. This is because each worker tends to have fewer reachable tasks with $e = 1.0$ initially, while a worker can expand her delivery range along with the expiration time e increase, which benefits both task assignment and multi-center collaboration. However, this enhancement of the delivery range is limited by the break of center-independent task assignments in no collaboration baselines, where a worker cannot deliver beyond her located center. In addition, the fluctuation in collaboration unfairness with Seq-RBDC and Opt-RBDC is due to the random selection of recipient centers.

Convergence of Game-Theoretic Collaboration. We also illustrate the convergence of the game-theoretic collaboration to support that utilizing the best-response mechanism can achieve an NE. To provide a fine-grained optimization process with more iterations, we conduct experiments with the proposed Seq-BDC methods with $|C| = 50$ instead of the default $|C| = 20$ on datasets GM and SYN, while other parameters follow the default setting. As illustrated in Fig. 11, we present the detailed number of assigned tasks (i.e., the blue dotted line) and collaboration unfairness (i.e., the red cross line) in each iteration of the game. We can observe that during the iterations, both these two metrics are optimized consistently. The consistent optimization shows that in every iteration of the multi-center collaboration game, a newly dispatched worker from a source center to a recipient center not only contributes to the task assignment in the recipient center, but also reduces the overall collaboration unfairness across all centers.

VII. Related Work

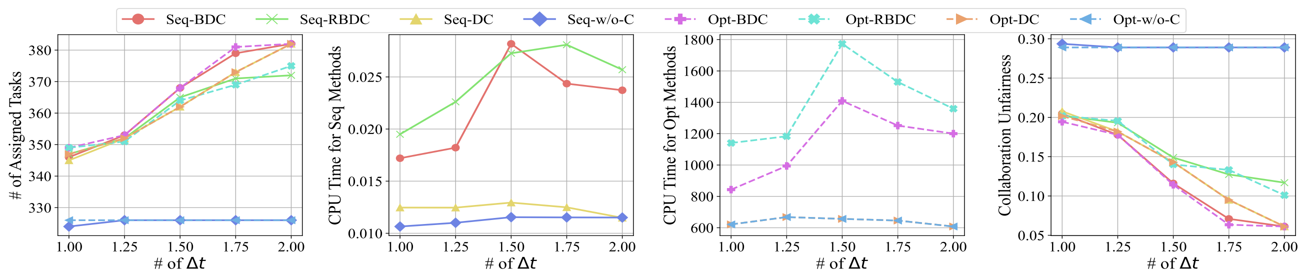
Spatial Crowdsourcing (SC) has attracted attention in the field of spatial-temporal data [40]–[46] and database [47]–[49] in recent years. Research proposes different application scenarios in in task assignments of SC, such as taxi dispatching [7], [24], [50], [51] and delivery routing [52]–[54]. In this section, we mainly introduce some works solving cooperation [17]–[19], [55]–[58] and fairness [25], [59]–[62] in task assignments.

Cooperation in Task Assignment. Chen et al. [17] study a problem of cooperation-aware spatial crowdsourcing (CA-SC), where spatial tasks are time-constrained and require more than one worker. With the consideration of the different cooperation relationships among workers, this work aims to maximize the overall cooperation quality revenue in the CA-SC problem. Zhao et al. [19], [58] propose another study that focuses on the coalition between workers. It proposes a novel SC problem, namely coalition-based task assignment, to maximize the overall rewards of workers. Li et al. [55] define the



(a) Assigned Tasks (b) CPU Time for Seq Methods (c) CPU Time for Opt Methods (d) Collaboration Unfairness

Fig. 9. Effect of e on GM



(a) Assigned Tasks (b) CPU Time for Seq Methods (c) CPU Time for Opt Methods (d) Collaboration Unfairness

Fig. 10. Effect of e on SYM

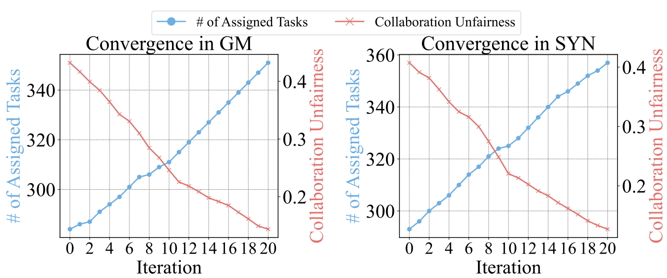


Fig. 11. Convergence of game-theoretic collaboration in Seq-BDC

autonomy and coordination task assignment problem, where orders are shared with multiple similar platforms. It aims to solve the imbalance between supply and demand through cooperation and to maximize global revenue and fairness and proposes a framework consisting of public order sending, local matching, global conflict adjustment, and result notification.

Fairness in Task Assignment. Chen et al. [59] define worker fairness by extending a common concept, Fagin-Williams share. It aims to assign tasks, considered as a resource in short supply, to individual spatial workers in a fair manner and formally defines an online bi-objective matching problem, namely the fair and effective task assignment problem. Zhao et al. [25] study the problem of fairness-aware task assignment in SC, where tasks are assigned to achieve fairness across workers. In particular, it aims to minimize the payoff difference among workers while maximizing the average worker payoff. Shi et al. [60] propose learning to assign with fairness, an effective and efficient task assignment scheme that optimizes both utility (i.e., the expected accumulated earnings across all drivers till batch time) and fairness (i.e., temporal earnings fairness among workers). It adopts reinforcement learning to make assignments holistically and proposes a set of acceleration techniques to enable fast and fair assignments. Shi et al. [60] propose a fair task assignment method in spatial crowdsourcing, but it differs from our method in terms of fairness definition and multi-center collaboration scenario. More recently, Cheng et al. [17] developed an advanced game-theoretic model for solving cooperation between workers. However, this model is not directly applicable to our problem due to the different cooperation targets.

VIII. Conclusion

In this paper, we introduce a Collaborative Multi-Center Task Assignment (CMCTA) problem where multiple distribution centers are managed by a spatial crowdsourcing platform. Specifically, CMCTA tackles the imbalance of workers and tasks across centers, which aims to maximize task assignment while ensuring fair collaboration across multiple centers. To address CMCTA, we propose the Iterative Multi-center Task Assignment and Optimization (IMTAO) framework, which is designed to leverage workforce transfer among centers to alleviate the gaps in worker and task distribution among centers. IMTAO integrates a two-phase optimization: center-independent task assignment with an efficient sequential task assignment algorithm, and inter-center workforce transfer with a multi-center collaboration game that both enhances task assignment and reduces collaboration unfairness through bi-directional optimization. Experiments validate the effectiveness and efficiency of IMTAO, showing significant optimization in task assignment efficiency and collaboration unfairness compared to baseline methods.

Acknowledgment

This work is partially supported by NSFC (No. 62472068) and Municipal Government of Quzhou under Grant (No. 2023D044, 2024D036). It is also partially supported by the Inge Lehmann Project (No. 4303-00014B).

References

[1] Y. Zhao, K. Zheng, Y. Li, J. Xia, B. Yang, T. B. Pedersen, R. Mao, C. S. Jensen, and X. Zhou, "Profit optimization in spatial crowdsourcing: Effectiveness and efficiency," *TKDE*, 2022.

[2] X. Li, Y. Zhao, X. Zhou, and K. Zheng, "Consensus-based group task assignment with social impact in spatial crowdsourcing," *Data Science and Engineering*, vol. 5, no. 4, pp. 375–390, 2020.

[3] Y. Zhao, Y. Li, Y. Wang, H. Su, and K. Zheng, "Destination-aware task assignment in spatial crowdsourcing," in *CIKM*, 2017, pp. 297–306.

[4] J. Xia, Y. Zhao, G. Liu, J. Xu, M. Zhang, and K. Zheng, "Profit-driven task assignment in spatial crowdsourcing." in *IJCAI*, 2019, pp. 1914–1920.

[5] Y. Zhao, K. Zheng, Y. Li, H. Su, J. Liu, and X. Zhou, "Destination-aware task assignment in spatial crowdsourcing: A worker decomposition approach," *TKDE*, pp. 2336–2350, 2019.

[6] Y. Cui, L. Deng, Y. Zhao, B. Yao, V. W. Zheng, and K. Zheng, "Hidden poi ranking with spatial crowdsourcing," in *KDD*, 2019, pp. 814–824.

[7] B. Zhao, P. Xu, Y. Shi, Y. Tong, Z. Zhou, and Y. Zeng, "Preference-aware task assignment in on-demand taxi dispatching: An online stable matching approach," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 2245–2252.

[8] Y. Tong, L. Chen, and C. Shahabi, "Spatial crowdsourcing: Challenges, techniques, and applications," *Proceedings of the VLDB Endowment*, vol. 10, no. 12, pp. 1988–1991, 2017.

[9] Y. Tong, Z. Zhou, Y. Zeng, L. Chen, and C. Shahabi, "Spatial crowdsourcing: a survey," *The VLDB Journal*, vol. 29, pp. 217–250, 2020.

[10] R. C.-W. Wong, Y. Tao, A. W.-C. Fu, and X. Xiao, "On efficient spatial matching," in *Proceedings of the 33rd international conference on Very large data bases*, 2007, pp. 579–590.

[11] Z. Chen, P. Cheng, Y. Zeng, and L. Chen, "Minimizing maximum delay of task assignment in spatial crowdsourcing," in *2019 IEEE 35th international conference on data engineering (ICDE)*. IEEE, 2019, pp. 1454–1465.

[12] B. Li, Y. Cheng, Y. Yuan, C. Li, Q. Jin, and G. Wang, "Competition and cooperation: Global task assignment in spatial crowdsourcing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 10, pp. 9998–10 010, 2023.

[13] Y. Tong, Y. Zeng, B. Ding, L. Wang, and L. Chen, "Two-sided online micro-task assignment in spatial crowdsourcing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 5, pp. 2295–2309, 2019.

[14] X. Mao, H. Wen, H. Zhang, H. Wan, L. Wu, J. Zheng, H. Hu, and Y. Lin, "Drl4route: A deep reinforcement learning framework for pick-up and delivery route prediction," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 4628–4637.

[15] C. Gao, F. Zhang, G. Wu, Q. Hu, Q. Ru, J. Hao, R. He, and Z. Sun, "A deep learning method for route and time prediction in food delivery service," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 2879–2889.

[16] H. Wen, Y. Lin, X. Mao, F. Wu, Y. Zhao, H. Wang, J. Zheng, L. Wu, H. Hu, and H. Wan, "Graph2route: A dynamic spatial-temporal graph neural network for pick-up and delivery route prediction," in *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 2022, pp. 4143–4152.

[17] P. Cheng, L. Chen, and J. Ye, "Cooperation-aware task assignment in spatial crowdsourcing," in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 2019, pp. 1442–1453.

[18] Y. Yang, Y. Cheng, Y. Yang, Y. Yuan, and G. Wang, "Batch-based cooperative task assignment in spatial crowdsourcing," in *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 2023, pp. 1180–1192.

[19] Y. Zhao, K. Zheng, Z. Wang, L. Deng, B. Yang, T. B. Pedersen, C. S. Jensen, and X. Zhou, "Coalition-based task assignment with priority-aware fairness in spatial crowdsourcing," *The VLDB Journal*, vol. 33, no. 1, pp. 163–184, 2024.

[20] Y. Zhao, X. Chen, G. Ye, F. Guo, K. Zheng, and X. Zhou, "Task allocation in spatial crowdsourcing: An efficient geographic partition framework," *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[21] F. Aurenhammer, "Voronoi diagrams—a survey of a fundamental geometric data structure," *ACM Computing Surveys (CSUR)*, vol. 23, no. 3, pp. 345–405, 1991.

[22] L. Kazemi and C. Shahabi, "Geocrowd: enabling query answering with spatial crowdsourcing," in *Proceedings of the 20th international conference on advances in geographic information systems*, 2012, pp. 189–198.

[23] X. Zeng, L. Deng, P. Chen, X. Chen, H. Su, and K. Zheng, "Lira: A learning-based query-aware partition framework for large-scale ann search," *arXiv preprint arXiv:2503.23409*, 2025.

[24] Y. Tong, D. Shi, Y. Xu, W. Lv, Z. Qin, and X. Tang, "Combinatorial optimization meets reinforcement learning: Effective taxi order dispatching at large-scale," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 10, pp. 9812–9823, 2021.

[25] Y. Zhao, K. Zheng, J. Guo, B. Yang, T. B. Pedersen, and C. S. Jensen, "Fairness-aware task assignment in spatial crowdsourcing: Game-theoretic approaches," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 2021, pp. 265–276.

[26] K. Braekers, K. Ramaekers, and I. Van Nieuwenhuyse, "The vehicle routing problem: State of the art classification and review," *Computers & industrial engineering*, vol. 99, pp. 300–313, 2016.

[27] M. Gummerum, B. López-Pérez, E. Van Dijk, and L. F. Van Dillen, "When punishment is emotion-driven: Children's, adolescents', and adults' costly punishment of unfair allocations," *Social Development*, vol. 29, no. 1, pp. 126–142, 2020.

[28] P. Brañas-Garza, A. M. Espín, F. Exadaktylos, and B. Herrmann, "Fair and unfair punishers coexist in the ultimatum game," *Scientific reports*, vol. 4, no. 1, p. 6025, 2014.

[29] G. E. Bolton and R. Zwick, "Anonymity versus punishment in ultimatum bargaining," *Games and Economic behavior*, vol. 10, no. 1, pp. 95–121, 1995.

[30] D. S. Leslie, S. Perkins, and Z. Xu, "Best-response dynamics in zero-sum stochastic games," *Journal of Economic Theory*, vol. 189, p. 105095, 2020.

[31] M. Voorneveld, "Best-response potential games," *Economics letters*, vol. 66, no. 3, pp. 289–295, 2000.

[32] H. Miao, Y. Zhao, C. Guo, B. Yang, K. Zheng, and C. S. Jensen, "Spatio-temporal prediction on streaming data: A unified federated continuous learning framework," *TKDE*, 2025.

[33] Y. Fang, Y. Liang, B. Hui, Z. Shao, L. Deng, X. Liu, X. Jiang, and K. Zheng, "Efficient large-scale traffic forecasting with transformers: A spatial data management perspective," *arXiv preprint arXiv:2412.09972*, 2024.

[34] X. Zeng, Q. Yu, S. Liu, Y. Xia, H. Su, and K. Zheng, "Target-oriented maneuver decision for autonomous vehicle: A rule-aided reinforcement learning framework," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023, pp. 3124–3133.

[35] Y. Zhao, K. Zheng, H. Yin, G. Liu, J. Fang, and X. Zhou, "Preference-aware task assignment in spatial crowdsourcing: from individuals to groups," *TKDE*, vol. 34, no. 7, pp. 3461–3477, 2022.

[36] Y. Zhao, K. Zheng, Y. Cui, H. Su, F. Zhu, and X. Zhou, "Predictive task assignment in spatial crowdsourcing: a data-driven approach," in *ICDE*, 2020, pp. 13–24.

[37] H. Miao, X. Zhong, J. Liu, Y. Zhao, X. Zhao, W. Qian, K. Zheng, and C. S. Jensen, "Task assignment with efficient federated preference learning in spatial crowdsourcing," *TKDE*, vol. 36, no. 4, pp. 1800–1814, 2023.

[38] J. Liu, L. Deng, H. Miao, Y. Zhao, and K. Zheng, "Task assignment with federated preference learning in spatial crowdsourcing," in *CIKM*, 2022, pp. 1279–1288.

[39] Z. Chen, R. Fu, Z. Zhao, Z. Liu, L. Xia, L. Chen, P. Cheng, C. C. Cao, Y. Tong, and C. J. Zhang, "gmission: A general spatial crowdsourcing platform," *Proceedings of the VLDB Endowment*, vol. 7, no. 13, pp. 1629–1632, 2014.

[40] L. Chen, Q. Zhong, X. Xiao, Y. Gao, P. Jin, and C. S. Jensen, "Price-and-time-aware dynamic ridesharing," in *2018 IEEE 34th international conference on data engineering (ICDE)*. IEEE, 2018, pp. 1061–1072.

[41] L. Deng, Y. Zhao, Y. Cui, Y. Xia, J. Chen, and K. Zheng, "Task recommendation in spatial crowdsourcing: A trade-off between diversity and coverage," in *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 2024, pp. 276–288.

[42] Y. Zhao, L. Deng, and K. Zheng, "Adataskrec: An adaptive task recommendation framework in spatial crowdsourcing," *ACM Transactions on Information Systems*, vol. 41, no. 4, pp. 1–32, 2023.

[43] G. Ye, Y. Zhao, X. Chen, and K. Zheng, "Task allocation with geographic partition in spatial crowdsourcing," in *CIKM*, 2021, pp. 2404–2413.

[44] Y. Zhao, K. Zheng, Z. Wang, L. Deng, B. Yang, T. B. Pedersen, C. S. Jensen, and X. Zhou, "Coalition-based task assignment with priority-aware fairness in spatial crowdsourcing," *VLDBJ*, 2023.

[45] Y. Zhao, J. Liu, Y. Li, D. Zhang, C. S. Jensen, and K. Zheng, "Preference-aware group task assignment in spatial crowdsourcing: Effectiveness and efficiency," *TKDE*, 2023.

[46] Y. Zhao, T. Lai, Z. Wang, K. Chen, H. Li, and K. Zheng, "Worker-churn-based task assignment with context-lstm in spatial crowdsourcing," *TKDE*, 2023.

[47] L. Chen, Y. Gao, X. Li, C. S. Jensen, and G. Chen, "Efficient metric indexing for similarity search and similarity joins," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 3, pp. 556–571, 2015.

[48] X. Ding, L. Chen, Y. Gao, C. S. Jensen, and H. Bao, "Ultraman: A unified platform for big trajectory data management and analytics," *Proceedings of the VLDB Endowment*, vol. 11, no. 7, pp. 787–799, 2018.

[49] L. Chen, Y. Gao, Z. Fang, X. Miao, C. S. Jensen, and C. Guo, "Real-time distributed co-movement pattern detection on streaming trajectories," *Proceedings of the VLDB Endowment*, vol. 12, no. 10, pp. 1208–1220, 2019.

[50] T. M. Rajeh, Z. Luo, M. H. Javed, F. Alhaek, and T. Li, "A clustering-based multi-agent reinforcement learning framework for finer-grained taxi dispatching," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[51] X. Tang, F. Zhang, Z. Qin, Y. Wang, D. Shi, B. Song, Y. Tong, H. Zhu, and J. Ye, "Value function is all you need: A unified learning framework for ride hailing platforms," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 3605–3615.

[52] Z. Zong, H. Wang, J. Wang, M. Zheng, and Y. Li, "Rbg: Hierarchically solving large-scale routing problems in logistic systems via reinforcement learning," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 4648–4658.

[53] Y. Ma, X. Hao, J. Hao, J. Lu, X. Liu, T. Xialiang, M. Yuan, Z. Li, J. Tang, and Z. Meng, "A hierarchical reinforcement learning based optimization framework for large-scale dynamic pickup and delivery problems," *Advances in neural information processing systems*, vol. 34, pp. 23 609–23 620, 2021.

[54] Y. Li, Y. Zheng, and Q. Yang, "Efficient and effective express via contextual cooperative reinforcement learning," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 510–519.

[55] B. Li, Y. Cheng, Y. Yuan, Y. Yang, Q. Jin, and G. Wang, "Acta: Autonomy and coordination task assignment in spatial crowdsourcing platforms," *Proceedings of the VLDB Endowment*, vol. 16, no. 5, pp. 1073–1085, 2023.

[56] Y. Cheng, Z. Liao, X. Huang, Y. Yang, X. Zhou, Y. Yuan, and G. Wang, "Cross online ride-sharing for multiple-platform cooperations in spatial crowdsourcing," in *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 2024, pp. 4140–4152.

[57] Y. Ye, C. H. Liu, Z. Dai, J. Zhao, Y. Yuan, G. Wang, and J. Tang, "Exploring both individuality and cooperation for air-ground spatial crowdsourcing by multi-agent deep reinforcement learning," in *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 2023, pp. 205–217.

[58] Y. Zhao, J. Guo, X. Chen, J. Hao, X. Zhou, and K. Zheng, "Coalition-based task assignment in spatial crowdsourcing," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 2021, pp. 241–252.

[59] Z. Chen, P. Cheng, L. Chen, X. Lin, and C. Shahabi, "Fair task assignment in spatial crowdsourcing," *Proceedings of the VLDB Endowment*, vol. 13, no. 12, 2020.

[60] D. Shi, Y. Tong, Z. Zhou, B. Song, W. Lv, and Q. Yang, "Learning to assign: Towards fair task assignment in large-scale ride hailing," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 3549–3557.

[61] A. Gupta, R. Yadav, A. Nair, A. Chakraborty, S. Ranu, and A. Bagchi, "Fairfoody: Bringing in fairness in food delivery," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 11, 2022, pp. 11 900–11 907.

[62] J. Sun, H. Jin, Z. Yang, L. Su, and X. Wang, "Optimizing long-term efficiency and fairness in ride-hailing via joint order dispatching and driver repositioning," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 3950–3960.