IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 29 April 2024, accepted 27 May 2024, date of publication 7 June 2024, date of current version 18 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3411156

RESEARCH ARTICLE

Collaborative Crowdsourced Vehicles for Last-Mile Delivery Application Using Hedonic Cooperative Games

NADA ELSOKKARY[1], SHAKTI SINGH[1,2], (Member, IEEE), RABEB MIZOUNI[1,2], HADI OTROK[1,2], (Senior Member, IEEE), AND HASSAN BARADA[1], (Senior Member, IEEE)

[1]Department of Computer Science, Khalifa University, Abu Dhabi, United Arab Emirates
[2]Center of Cyber-Physical Systems (C2PS), Khalifa University, Abu Dhabi, United Arab Emirates

Corresponding author: Rabeb Mizouni (rabeb.mizouni@ku.ac.ae)

This work was supported by the Khalifa Internal Grant.

ABSTRACT In this paper, the problem of collaboration in crowdsourced last-mile delivery is addressed, where multiple crowdsourced vehicles cooperate to fulfill tasks. Collaborative crowdsourced frameworks allow recruited vehicles, referred to as *workers*, to perform shorter trips while expanding the geographic coverage. Existing solutions in collaborative, crowdsourced last-mile delivery solely maximize the task allocation without considering 1) cost factors such as travel distance and payoff and 2) the self-interest of crowdsourced workers. As a solution, we propose a hedonic cooperative game approach that determines delivery routes and assigns relaying vehicles by maximizing the average payoff per kilometer, where payoffs are based on task contributions. Specifically, the proposed algorithm, hedonic crowd relay assignment (HCRA), uses the Nash equilibria of a series of hedonic games as the basis for the task allocation. To compute the workers' preference lists, HCRA relies on crowd relay breadth-first search (CR-BFS) to find a set of potential routes for task completion, given the constraints of the vehicles. The proposed solution is compared to a benchmark, and the results demonstrate that a more efficient and scalable solution is achieved using HCRA, where both the workers' total payoffs and average payoff per kilometer are increased, even with increasing numbers of vehicles, tasks, and relays.

INDEX TERMS Collaborative algorithm, cooperative hedonic game, crowdsourced vehicles, last-mile delivery, task allocation.

I. INTRODUCTION

The recent rise of the gig economy and advancements in technology have unlocked new possibilities for innovation in last-mile delivery, which is a crucial step in getting a parcel or product to its final destination. In 2020, a report by Statista noted that the global crowdsourced last mile delivery market was valued at $1.3 billion [1]. The report projected the market size to reach $9.31 billion by 2027, leading to an increased demand for last-mile delivery services. Crowdsourcing, the practice of recruiting a number of individuals to perform a task, is emerging as an increasingly popular solution with companies such as Postmates, Amazon, and Uber establishing their own crowdsourced delivery services [2], [3]. These companies can leverage vast networks of individual contractors, each with unique vehicles and capabilities, willing to deliver payment packages.

The main challenge of spatial crowdsourcing applications, whether a task is fulfilled in a single trip or performed collaboratively, is the efficient allocation of workers by accounting for their availability and capabilities [4]. Unlike employed workers using delivery trucks, available crowdsourced workers are mainly characterized by their vehicles or mode of transportation; the location, quality, and energy level of their vehicle dictate their constraints and affect their performance on the task. The vehicles may only be allowed

The associate editor coordinating the review of this manuscript and approving it for publication was Claudio Zunino.

in specific neighborhoods, on set routes, or at certain times. The type of worker, i.e., the vehicle they operate from, is a crucial aspect of the allocation [5], [6], as pedestrians have different capabilities than cyclists and motorists. Crowd-sourcing allocation algorithms must quickly and efficiently navigate the constraints of the workers and find optimal allocations. The allocations are optimized with respect to a goal such as minimizing the total costs [5], [6], [7], [8], [9], [10], maximizing the number of allocated tasks [11], and maximizing the rewards or profits of the workers [12], [13]. Alternatively, the optimization is multi-objective, combining several of the aforementioned goals [14], [15]. The challenge of optimally allocating crowdsourced workers in last-mile delivery has been approached in a variety of ways. Greedy algorithms provide solutions quickly in such a dynamic environment [8], [10], [16], [17], [18], [19]. Additionally, heuristics such as simulated annealing, tabu search, and genetic algorithm have been proposed to reach more optimal solutions within reasonable time [5], [9], [20], [21], [22]. Despite the plethora of solutions in the literature, the challenge of fulfilling tasks over long distances remains; such as when the customer's location is in a suburban area, far from the closest warehouse. If the distance is short, a single trip is sufficient [23]; however, attempting to recruit crowdsourced workers over a longer distance may be time-consuming and costly, especially if the task must be completed quickly [5], [9].

For this reason, recent literature proposes to break down the tasks in crowdsourced last-mile delivery into multiple steps [5], [6], [9], [15], [23]. The delivery task may be performed as a relay between several agents whereby the task is divided into smaller segments; the agents each perform one of the segments of the journey and pass along the package to the next worker at a *relay point*. The first examples of such crowdsourced LMD frameworks include relays between in-house delivery trucks, traveling between or acting as *microhubs*, and crowdsourced vehicles [5], [6], [9], [15]. These systems have been referred to in the literature as *two-echelon* or *truck-crowd* delivery systems. They are often modeled as routing problems to find the optimal relay points and assign workers. Most recently, the literature explores a relay mechanism between two or more crowdsourced workers; henceforth referred to as *crowd-relay* [23], [24]. In the short term, facilitating relay points between the crowd workers, whether through fixed lockers in high-traffic locations or scheduled meeting points, improves the geographic coverage of the workers, thus increasing the task allocation rate. Over time, the approach reduces the time and cost of the allocation by reducing delays in the task allocation caused by waiting for the right worker to be available to complete the task single-handedly. The problem remains that allocating workers in a crowd-relay delivery has added layers of complexity as multiple vehicles need to collaborate to complete a single task while accounting for the aforementioned considerations regarding the task nature and the worker availability and constraints [25].

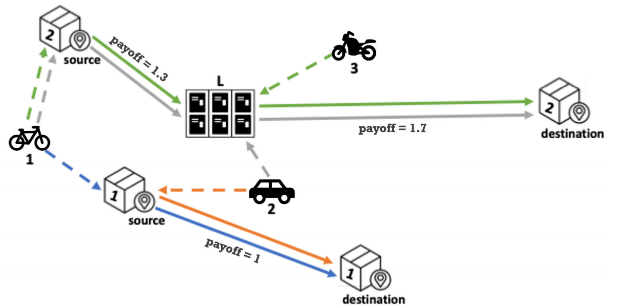

FIGURE 1. **Illustration of a crowd-relay scenario in last-mile delivery. In this example, the source and destination locations of tasks 1 and 2 are represented. Locker *L* represents the location of a parcel locker that can be used as a relay point between collaborating workers. The available workers 1, 2, and 3 are a bicycle, a car, and a motorcycle, respectively. There are four possible delivery paths where the payoff corresponding to each contributing step is indicated. The blue path assigns worker 1 to transport package 1 from its source to its destination. The orange path assigns worker 2 to transport package 1 from its source to its destination. The green path assigns worker 1 to transport package 2 from its source to locker *L*, then worker 3 to transport package 2 from locker *L* to its destination. The gray path assigns worker 1 to transport package 2 from its source to locker *L*, then worker 2 to transport package 2 from locker *L* to its destination.**

A. PROBLEM STATEMENT

Optimizing crowd-relay last mile delivery problems is an open challenge since the current methods aim to maximize the allocation rate while disregarding the self-interested nature of workers when considering the tasks they are capable of participating in or the routes they use to complete them [23]. The algorithm proposed in [23] favors single-vehicle routes over collaborative ones and exhaustively searches all possible routes for all tasks before performing the allocation through random selection. To illustrate the limitations of this strategy, let us consider the scenario shown in Figure 1, with three crowdsourced vehicles, a relay point through locker L, and two tasks to fulfill package 1 and package 2, which would afford the assigned workers a payoff of 1 and 3 upon completion, respectively. Given the constraints of the crowdsourced vehicles, the figure highlights the possible routes for completing the tasks, both single and collaborative, as well as the payoff of the workers for each contributing step.

In this example, the strategy described in [23] would allocate worker 1 to complete task 1 in one step (illustrated by the blue path), leaving task 2 unfulfilled. However, if the payoffs are considered, both worker 1 and worker 2 would prefer to collaboratively complete task 2 (illustrated in the gray path) rather than task 1 alone since their individual payoffs upon completion of task 2 are higher. This would improve the workers' total payoff from 1 to 3 but would still result in a sub-optimal allocation by leaving task 1 unfulfilled. By considering the preference of the workers in terms of payoffs versus the distance traveled, it can be shown that allocating worker 2 to complete task 1 (through the orange path) allows worker 1 and worker 3 to collaborate on task 2 (illustrated by the green path); thus fulfilling all tasks, and raising the total payoffs of the workers to 4. In this paper, the compensation of the workers with respect to the distance they travel is used as the basis for optimal allocation.

B. CONTRIBUTION

This article proposes an algorithm for effectively allocating crowdsourced heterogeneous workers in collaborative delivery scenarios while maximizing the profit of the workers, defined as payoffs per distance traveled. The crowdsourced relay last mile assignment problem (CR-LMAP) is defined and then framed as a hedonic game, where the workers are self-interested players willing to collaborate on task completion, their individual profit defines their utility, and their utility depends only on their collaborating members.

The crowdsourcing platform identifies the available tasks, relay points, and workers, as well as their availability and constraints. The proposed algorithm, HCRA, employs a series of hedonic games whereby the available workers form coalitions following their preferences based on a set of predefined possible paths to perform the available tasks. The paths are computed using a constrained breadth-first search subroutine, CR-BFS, such that the workers on the path can successfully take the intended package from the pickup location (source) to the drop-off location (destination). The paths may consist of one or more collaborating workers, where the workers share the reward of the task based on the proportion of the path they each contribute. At the end of each game, the tasks are allocated based on the lowest distance coalition formed that can successfully complete the task.

Simulation outcomes demonstrate that the proposed algorithm stably maximizes the profit of the assigned workers, thereby outperforming state-of-the-art methods. This is primarily because the proposed algorithm prioritizes allocating high-paying tasks through paths that require a lower travel distance. In addition to providing more preferable solutions, the proposed algorithm is able to reach the solutions faster than the benchmark.

The remainder of the paper is organized as follows. Section II summarizes the related research in crowdsourced last-mile delivery relay problems as well as that in hedonic games in task assignment problems. Section III formally represents the problem and hedonic game, proves the existence of the hedonic game Nash Equilibrium, and details the proposed algorithm. Section IV presents the environment, benchmark, metrics, datasets, and parameters used to evaluate the performance of the proposed method. Simulation results are presented and discussed in Section V. The article is concluded in Section VI.

II. RELATED WORK

In order to allow for better geographical coverage and shorter trip detours for crowdsourced workers, a variety of multi-step solutions in crowd-enabled last-mile delivery have been proposed. The most common is the use of a truck-crowd delivery model, whereby a fleet of trucks (owned by the shipping provider) travel to lockers or *microhubs*, which then serve as the pickup locations for the crowdsourced workers to complete the last step of the task to deliver to the customer's location [5], [6], [9], [15]. For clarity, the truck-crowd delivery model is distinguished from the concept of combining trucks and crowd shippers where a portion of packages are fully performed by the crowd shippers while the remaining packages are fully performed by a truck [20]; this would not be considered multi-step. Another approach is the use of the crowd-relay model, where a chain of collaborating crowdsourced workers uses relay points for the task to be performed. The scope of this work is within the use of the crowd-relay model for last-mile delivery. The research on this last mile delivery model is reviewed in the following section; Table 1 summarizes the key aspects of the research.

A. CROWD-RELAY MODEL

Zhang et al. were the first to propose collaborative delivery between crowdsourced workers [24]. Their research formulates a general optimization problem for crowdsourced delivery motivated by Packet-Switch networks with three aspects: connectivity, profit, and risk. The study proposed to dynamically optimize the problem using a routing algorithm based on stochastic mobility and social graphs of the workers and aimed to increase and optimize the timely delivery of packages using a multi-criteria function aiming to maximize profit and quality of service and minimize cost.

The recent study conducted by Ghaderi et al. proposed the use of a relay of crowdsourced workers and parcel lockers for last-mile delivery [23]. The authors allow the delivery tasks to be performed by one or two crowd shippers such that they exchange the parcels through a locker infrastructure. The contribution of the work is a novel two-phase algorithm for 1) locating the parcel lockers from a set of potential locations and 2) assigning each delivery job to one or two workers using a random selection strategy. Experiments were conducted to evaluate the performance of the algorithms against an ILP solution. The results showed that the algorithm proposed in Ghaderi et al. [23], JCA, significantly lowered computational expenses. Moreover, it was shown that enabling collaborative delivery improved the delivery rate by up to 5%, with a small number of lockers utilized in essential locations.

Reviewing the literature in crowd-relay LMD concludes the following limitations:

- The methods rely heavily on the predictability of the workers' mobility patterns or habits [23], [24], [26]. This is not a reliable measure of the workers' ability or willingness to perform a task. Many circumstances can cause workers to operate outside of their usual pattern. Assuming that the historical positions of the worker dictate their current location can lead to delayed task completion or job cancellation if the worker is further than expected.
- The travel distance is incorporated as a constraint for delivery; however, the algorithms do not aim to optimize the travel distance [23], [24]. The travel distance required to complete a task represents the cost

TABLE 1. Multi-step last-mile delivery related work.

Solution	T-W	Truck	Relay Point	Task Type	Work Force	Objective Function	Solution
Kafle et al. (2017) [5]	M-1	yes (many)	micro-depot (many)	pick-up drop-off	pedestrians cyclists	min. total cost (operating & time penalty)	Crowdshipper bid generation: Undirected TSPs Truck routing: Tabu Search on Two Sub-problems
Zhang et al. (2017) [24]	1-M	none	locker (many)	pick-up drop-off	homogeneous	max. profit, timely delivery min. cost (hops)	Modelled as a Packet Switch Network Routes based on Historical Mobility
Huang & Ardiansyah (2019) [9]	M-1	yes (many)	micro-depot (many)	pick-up drop-off	homogeneous	min. total cost (operating)	Two Heuristics Construction: Sweep & Nearest Neighbor Improvement: Tabu Search
Ballare & Lin (2020) [6]	M-1	yes (many)	locker (many)	pick-up drop-off	cyclists automobiles	min. total cost (fees, rates, & penalties)	Maximum Split-Benefit with Tabu Search to solve Routing problems for Crowdshippers (M-1) and Truck (M-M)
Elsokkary et al. (2023) [15]	M-1	yes (one)	micro-depot (many)	drop-off	homogeneous	max. QoS	Tabu Search Genetic Algorithm
Ghaderi et al. (2022) [23]	1-M	none	locker (many)	pick-up drop-off	homogeneous	max. delivery rate	Exhaustive Search with Constraints Greedy Random Selection
Our Work	1-M	none	locker (many)	pick-up drop-off	heterogeneous	max. profit (payoff per distance)	Iterative Hedonic Games Routing based on Constrained Breadth-First Search

of delivery; a larger travel distance often translates to more fuel, emissions, and time. Optimizing travel distance is a main goal in last-mile delivery problems.

- The worker's interests or preference over the tasks are not accounted for [23] and [24]. It is naive to assume that the workers are equally likely to do a task regardless of the payment they will receive for completing the task. The workers will have preferences over the tasks they are capable of performing based on the amount of work to be performed and the compensation received.
- The workers' payment cannot be distributed uniformly regardless of the proportion of their work in completing the task, as done by Ghaderi et al [23]. Fair compensation schemes are important factors in promoting collaboration.
- Recent methods do not scale well to tasks with more than two workers [23]. It is our hypothesis that the reason for this limitation is that the proposed algorithm in [23], if extended to include more than two-worker delivery paths, would not be able to produce a feasible solution for larger instances due to the exhaustive nature of its pre-processing step to find potential solutions. Section IV-A provides more details about JCA, which is used as a benchmark to evaluate the proposed work and test the hypothesis above.

B. COALITION BUILDING IN ASSIGNMENT PROBLEMS

The literature on game theory was reviewed to highlight the key work conducted on the use of hedonic games and other coalition-building strategies in general assignment problems. In cooperative game theory, the aim is to predict and analyze which coalitions will form, the joint actions they perform, and the collective payoffs that result when collaboration between players is encouraged or enforced [27]. A cooperative game consisting of a finite set of players N, is formally defined by the following characteristic function:

$$p : 2^N \rightarrow \mathbb{R}$$

It defines a utility value on all possible coalitions of players and satisfies $p(\emptyset) = 0$. The characteristic function defines the collective payoff that a given set of players gains by forming a coalition. The progression or "solution" of a cooperative game is a partition Π on the set of players N, such that each set in the partition is a coalition. Cooperative game theoretic models are offered in the literature on problems such as modeling trust in multi-cloud services communities [28], protocols in urban-VANET [29], [30], [31], [32], and task matching in crowdsourcing applications [33], [34], [35].

Hedonic games are general models for coalition formation in which players have socially "blind" preferences over which group they belong to. Some well-known matching problems are considered hedonic games, such as the stable marriage and stable roommates problems. In hedonic games, the players are greedy and independent; their preferences over the coalitions depend only on the members of the coalition, with no regard for the distribution of the remaining members in the partition. In other words, the players in a hedonic game are not interested in the welfare of others in the game. The players will only join a coalition with other members if it improves their payoff. Hedonic game models are used when forming coalitions of players who are not concerned with the placement of players outside their coalition or the structure of other coalitions. Moreover, hedonic games have non-transferable utility, meaning they are not concerned with determining the payment structure of the members of the coalition. The properties of hedonic games are well-suited to model the preferences of the crowdsourced workers in the task allocation problem described in Section I-A. The main goal of the problem is the find groups of worker that are willing to collaborate over each task in a way that allocates the tasks while accounting for the workers' satisfaction. The delivery workers do not gain rewards based on the allocation of other workers to different tasks; they are only interested in the rewards from the collaboration they participate in. Additionally, since the delivery compensation is known, the workers are not concerned with determining the resulting payment structure within their coalition. In contrast to other cooperative games such as auctions, coalition formation in hedonic games does not choose how to allocate profit among its members.

While hedonic games do not inherently incorporate tasks, hedonic game-based models have been proposed in the research for task assignment problems. Crowdsourced delivery problems, without collaboration, have been modeled as stable matching games and solutions are found using Gale-Shapely [33], [36]. However, using Gale-Shapley in the context of this collaborative last-mile delivery assignment problem would not be suitable due to the algorithm's limitation in navigating highly constrained problems with large state spaces. Stability, using Gale-Shapley, may be reached but may result in an unfavorable solution. Jang et al. (2018) proposed a hedonic game-based framework for decision-making in the problem of task allocation of drone swarms [37]. The authors model the task allocation problem as a hedonic game with self-interested players who are open to forming coalitions such that it improves their utility. The utility of the workers is defined on task-coalition pairs, and at the start of the game, all workers are in singleton coalitions paired with the "void" task, i.e., doing no tasks. The work of Jang et al. [37] informs and inspires the hedonic game proposed in Section III-B; however, some key differences arise from the nature of the problem in which the game is set. The game in [37] is anonymous, meaning that the players are only concerned with the number of participants in the coalition rather than the identity of those members. In last-mile delivery, the identity of the collaborating members is crucial. In drone swarming, each task can have at most one task-coalition pair (if two coalitions are paired with the same task, their union is considered the task-coalition pair; they cannot be considered separate). However, in the setting of last-mile delivery, the same task may be fulfilled by more than one coalition of workers (delivery routes). For this reason, the proposed work does not consider the stability of one hedonic game as the solution; rather, it plays a series of hedonic games and considers the stability of each hedonic game as the basis for the allocation step before starting the next game. Additionally, [37] states that all drones must be allocated to a task; therefore, the utility of being assigned to the void task is zero for all players. In last-mile delivery, this is not a feasible assumption; instead, each worker states a minimum acceptable utility, which serves as their individual utility on the void task. Finally, the algorithm in [37] is designed to play in the game in a distributed asynchronous environment, whereas, the algorithm proposed in this paper plays the game on a centralized platform while simulating the self-interested nature of the players.

III. PROPOSED METHODOLOGY

This section formally introduces the assignment problem in crowdsourced relay for last-mile delivery and its underlying constraints and assumptions. Table 2 provides the notation used. We propose to solve the CR-LMAP through a hedonic game-based algorithm. The definitions of the game characteristics are detailed in Section III-B and the proposed algorithm can be found in Section III-C.

A. PROBLEM FORMULATION

Suppose there is a set of available crowdsourced workers W, a set of tasks T, and a set of available lockers L acting as relay points, where:

- Each worker $w_i \in W$ is attributed with a current location and maximum travel distance d_i^W, dictated by the type and condition of the vehicle.
- Each task t_j corresponds to a source location, destination location, and reward f_j^T.
- Each locker $l \in L$ is associated with a location.

Given the worker and task constraints, a set of possible delivery paths P_j can be found for each task t_j. Each path $p_{jk} \in P_j$ consists of an ordered list of one or more contributing steps. If the path consists of one step, it is considered as a *single delivery*; if it consists of more than one step, it is referred to as *collaborative delivery*. Each step s in p_{jk} is defined by:

- The pickup location a and dropoff location b; the pickup location of the first step must be the task source, and the dropoff of the last step must be the task destination. Intermediary pickups and dropoffs may be locker locations as long as the dropoff location of one step is the pickup location of the following step.
- The worker w_{jks} who performs the contributing step by traveling from his current location to the pickup location and then proceeding to the dropoff location. w_{jks} belongs to $W_{jk} \subseteq W$ which represents the set of workers collaborating on the path p_{jk}.
- d_{jks}^{wa} the euclidean distance between worker w_{jks} and the pickup location.
- d_{jks}^{wb} the euclidean distance between worker w_{jks} and the dropoff location.
- d_{jks}^{ab} the euclidean distance between the pickup and dropoff locations.
- The distance $d_{jks} = d_{jks}^{wa} + d_{jks}^{ab}$ that worker w_{jks} travels to complete the contributing step. Due to the constraint imposed by workers on their travel distance, a path is only considered valid if:

$$(d_{jks}^{wa} + d_{jks}^{wb}) \leq d_i^W, \quad w_i = w_{jks} \tag{1}$$

- The individual reward, or *payoff*, of r_{jks}, which is a portion of the total task reward f_j^T,

$$r_{jks} = \frac{d_{jks}^{ab}}{\sum_{\forall s \in p_{jk}} d_{jks}^{ab}} \cdot f_j^T \tag{2}$$

The objective of the CR-LMAP is to find the allocation of workers to maximize the overall profit of the workers i.e. the payoff earned per kilometer traveled by the workers. The

TABLE 2. Nomenclature.

Symbol	Definition
L	Set of available lockers
W	Set of available workers
w_i	Worker i
d_i^W	Maximum travel distance of w_i
T	Set of available tasks
t_j	Task j
f_j^T	Total reward i.e., freight value, of t_j
P_j	Set of paths for task t_j
W_{jk}	Set of workers for path $p_{jk} \in P_j$
w_{jks}	Worker performing step s in path p_{jk}
d_{jks}	Distance traveled by w_{jks}
r_{jks}	Individual reward earned by w_{jks} i.e. payoff
t_ϕ	The void task
u^ϕ	The utility of t_ϕ for all workers (parameter)
n^S	Maximum number of relay steps (parameter)
n^P	Maximum number of paths found (parameter)
D_{max}	Maximum Euclidean distance (source-to-destination) of all tasks

problem defined above is then represented as follows:

$$\max_{x_{jk}} \sum_{\forall t_j \in T} \sum_{\forall p_{jk} \in P_j} x_{jk} \cdot \frac{f_j^T}{\sum_{\forall s \in p_{jk}} d_{jks}} \tag{3}$$

$$\text{subject to} \quad \sum_{\forall p_{jk} \in P_j} x_{jk} \leq 1, \quad \forall t_j \in T \tag{4}$$

$$\sum_{\forall t_j \in T} \sum_{\forall p_{jk} \in P_j} x_{jk} \cdot \mathbb{1}(w_i \in W_{jk}) \leq 1, \quad \forall w_i \in W \tag{5}$$

$$x_{jk} \in \{0, 1\}, \quad \forall p_{jk} \in P_j, \forall t_j \in T \tag{6}$$

where x_{jk} is the binary decision variable indicating if task t_j will be completed through path p_{jk} by its corresponding workers W_{jk}. The indicator function $\mathbb{1}(\cdot)$ gives a value of 1 if true and 0 otherwise. Every task can be assigned to at most one path as stated by Equation 4. Equation 5 stipulates that each worker can be assigned in at most one path.

B. GAME DEFINITION

In order to solve the CR-LMAP defined in Section III-A, the following hedonic game event is characterized similarly to that proposed in [37] and summarized in Section II-B. The workers in W are the players, and the individual utility of $w_i \in W$ on the task-coalition pair (t, m) is defined by the maximum obtainable profit using this coalition. The individual utility and profit functions are defined by align 7 and Equation 8. Each worker w_i possesses an individual utility function $u_i : T \times W \rightarrow \mathbb{R}$ attributing each possible path's worker set W_{jk} with its utility to w_i. In other words, it is the maximum individual profit w_i gains for successfully completing a path for delivering task t_j in collaboration with members m. If a subset of W does not correspond to any possible paths, its individual utility for all workers is zero. Furthermore, the void task t_ϕ formally represents the task-coalition pair for unassigned workers. The individual utility for t_ϕ is set to a non-zero minimum acceptable utility, u^ϕ, regardless of the collaborator set.

$$u_i(t_j, m) = \max_{p_{jk} \in P_j} [\text{profit}_i(p_{jk}) \times \mathbb{1}(w_i \in m)] \tag{7}$$

$$\text{profit}_i(p_{jk}) = \begin{cases} \dfrac{r_{jks}}{d_{jks}} & \text{if } \exists s \in p_{jk}, w_i = w_{jks} \\ 0 & \text{if } w_i \notin W_{jk} \end{cases} \tag{8}$$

1) NASH STABILITY

The equations above derive a preference relation for each w_i on the tasks the worker could join; thereby characterizing the hedonic game. To guarantee algorithm convergence, the stability of the game is shown.

Proof: If the number of workers, tasks, and paths for each task is finite, then, using Equation 7, it is possible to compute the utility of each worker on each task i.e. the worker's highest possible ''profit'' from completing this task. Let all workers begin the game in the coalition of the void task: (t_ϕ, W). Let all task coalitions for real tasks be empty member sets: $\forall t_j$ in T, the corresponding coalition is $(t_j, \emptyset)$. In the first round of the game, each worker will decide to move to the coalition of their most preferred task. If more workers join a coalition, the utility of other workers in the coalition does not change. Once each worker has joined their most profitable task, no worker will choose to move to a different coalition unilaterally. □

With this knowledge, the Nash stable partition of this hedonic game can be found easily. However, the Nash equilibrium of this game does not necessarily supply the optimal allocation. The resulting task-coalition pairs contain workers that fall into one of two categories: 1) the worker and their necessary collaborators played the same strategy and can fully deliver the package through one or more paths, or 2) the worker cannot fully deliver the package through any path because some necessary collaborator did not play the same strategy. Post-processing is required to assign a single path of collaborating workers to each task based on the resulting coalition structures.

C. PROPOSED ALGORITHM

The proposed algorithm is designed to assign the available set of crowdsourced workers to the available tasks such that the workers may collaborate by successively completing contributing steps for the same task. The algorithm aims to maximize the overall profit of the workers as defined by Equation 3. Figure 2 illustrates the flowchart of the proposed Algorithm 1. The components of the system are the tasks, the crowdsourced workers, and the lockers. The tasks are defined by their unique source location, unique destination location, and reward (freight value). The available lockers are identified by their locations. The workers are identified by their current location, mode of transport, and maximum travel distance.

The worker assignment process HCRA in Algorithm 1 repeats the following steps until no more tasks can be assigned:

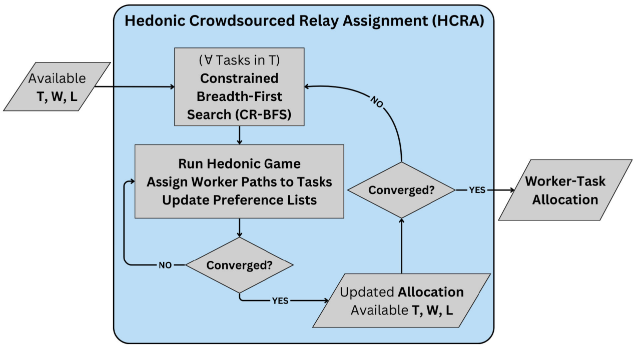

FIGURE 2. Flowchart of the proposed algorithm HCRA and its subroutine CR-BFS.

1) Identifies the set of available tasks, the set of available lockers, and the set of available workers on the crowdsourcing platform,
2) Calls CR-BFS, described in Algorithm 2, to find the sets of possible delivery routes for each of the available tasks; given a maximum number of paths n^P and a maximum number of steps n^S,
3) Plays a series of hedonic games based on the computed paths where for each game it:
   a) Computes the Nash equilibrium of the resulting hedonic game,
   b) Assigns the shortest path of workers to each task based on the Nash equilibrium of the game,
   c) Updates the available tasks, workers, and lockers to reset the preference lists.
   d) Repeats until convergence.
4) Updates the available tasks, workers, and lockers.

1) CROWDSOURCED RELAY BREADTH-FIRST SEARCH

In order to find a set of possible paths that complete a given task, the algorithm represents the problem setting as a graph. The problem graph nodes are the task source and destination locations and the locations of the lockers. Edges on the graph represent a contributing step that can be performed. Each edge stores a pickup node, a drop-off node, and a set of qualified workers. Algorithm 2 creates the worker graph constrained by the limitations of the workers with respect to the task.

The algorithm considers a worker qualified to perform a contributing step (edge) if it does not violate the worker's travel threshold constraint as defined in equation 1. BFS first explores paths with the least amount of edges (steps) and, therefore, workers. CR-BFS aims to find paths starting from the task's source node and reaching the destination node while maintaining the constraints listed below. The algorithm continues to search until the maximum number of paths is reached or there are no more nodes in the queue.

1) Each worker may be qualified on multiple edges on the graph; however, each worker may only be assigned to one contributing step on the graph.

Algorithm 1 Hedonic Crowdsourced Relay Assignment (HCRA)

Input: available tasks T, available workers W and lockers L, n^P, n^S
Output: task-worker-path assignment
while T not converged **do**
  for task $t \in T$ **do**
    // compute potential paths for each task
    CR-BFS(t, W, L, n^P, n^S)
  end for
  for worker $w \in W$ **do**
    // compute preference list for all workers
    for task $t \in T$ **do**
      for path $p \in P_t$ **do**
        Store profit$_i(p)$
      end for
    end for
  end for
  while T not converged **do**
    // start new hedonic game
    // compute Nash equilibrium
    for worker $w_i \in W$ **do**
      // according to Equation 7
      Find the (t_j, p_{jk}) pair with max $u_i(t_j, W_{jk})$.
      Join w_i to the coalition of task t_j.
    end for
    // assign tasks based on Nash equilibrium
    for task $t_j \in T$ **do**
      for path $p_{jk} \in P_j$ (in ascending order) **do**
        if $W_{jk} \subseteq$ the coalition of task t_j **then**
          Assign task t_j to W_{jk} using path p_{jk}.
          Update T, W, L.
          Update preference lists ∀ workers.
        end if
      end for
    end for
  end while
end while

2) There is a maximum number of paths n^P parameter to control the complexity of the algorithm.
3) There is a maximum path length (number of relay steps) parameter n^S to control the maximum number of workers relaying to deliver each task. Therefore, the algorithm is depth-limited by the maximum number of workers per path.

D. EVALUATION METRICS

In order to evaluate the efficacy of the proposed model, HCRA + CR-BFS, the following metrics were considered along with the algorithm runtime:

1) NUMBER OF TASKS ALLOCATED

The number of tasks that were assigned by the algorithm according to Equation 9.

$$\sum_{\forall t_j \in T} \sum_{\forall p_{jk} \in P_j} x_{jk} \tag{9}$$

Algorithm 2 Crowdsourced Relay Breadth-first Search (CR-BFS)

Input: task t, available workers W and lockers L, n^P, n^S
Output: possible paths P_t
Let n_P be the maximum number of paths that should be found for task t and n_S be the maximum depth of the paths.

Let P_t be the set of paths found for task t. P_t = []
Compute worker graph G^t_{WL}
Add (task source node, [], []) to the queue.
while queue is not empty **do**
  if $|P_t| = n^P$ **then**
    // path number limit exceeded
    break
  end if
  Dequeue (node a, p_a, W_a).
  // p_a is the path to a
  // W_a is the order of workers on p_a
  if node depth of $a > n^S$ **then**
    // depth limit exceeded
    break
  end if
  if node a is task destination node **then**
    Add the path to P_t.
  else
    Get all paths to the adjacent nodes of node a.
    for (worker w, node b) $\in$ set of adjacent paths **do**
      if constraint (5) is violated **then**
        continue
      end if
      if (a, b) not visited by w & $w \notin W_a$ **then**
        Mark (worker w, node a, node b) visited.
        Enqueue (node b, p_b, W_b).
      end if
    end for
  end if
end while
for path $p \in P_t$ **do**
  Compute time, distance, and payoff for all workers.
end for
Sort paths by distance in ascending order.
// To be used in the assignment step

2) NUMBER OF UNFULFILLABLE TASKS

The number of tasks that the algorithm found no possible paths and thus could not possibly allocate.

$$\sum_{\forall t_j \in T} \mathbb{1}(P_j = \emptyset) \tag{10}$$

3) TOTAL PAYOFFS

The total payoffs earned by the workers according to Equation 11, i.e. the sum of the rewards of the allocated tasks. It gives insight into the algorithm's behavior in maximizing the workers' social welfare.

$$\sum_{\forall t_j \in T} \sum_{\forall p_{jk} \in P_j} x_{jk} \cdot f_j^T \tag{11}$$

4) DISTANCE TRAVELLED PER TASK

The average number of kilometers the workers must travel per assigned task according to Equation 12, i.e. the total distance traveled in the assigned paths divided by the number of allocated tasks. It gives insight into the algorithm behavior in terms of preferring to allocate lower distance tasks.

$$\frac{\sum_{\forall t_j \in T} \sum_{\forall p_{jk} \in P_j} x_{jk} \cdot \sum_{\forall s \in p_{jk}} d_{jks}}{\sum_{\forall t_j \in T} \sum_{\forall p_{jk} \in P_j} x_{jk}} \tag{12}$$

5) 'PROFIT' (PAYOFF PER KM)

The payoff per kilometer traveled to complete the tasks according to Equation 13, i.e. the sum of the rewards of the allocated tasks divided by the total distance traveled in the assigned paths. The payoffs earned versus the investment of travel distance (which implicitly requires resources such as fuel/energy and time) is the main measure of "worth" for a task from the perspective of the crowdsourced worker.

$$\frac{\sum_{\forall t_j \in T} \sum_{\forall p_{jk} \in P_j} x_{jk} \cdot f_j^T}{\sum_{\forall t_j \in T} \sum_{\forall p_{jk} \in P_j} x_{jk} \cdot \sum_{\forall s \in p_{jk}} d_{jks}} \tag{13}$$

6) QUALITY OF ALLOCATION

To assess the overall performance of the algorithm against the benchmark, we define the quality of the allocation (QoA) as the arithmetic mean of five measures:

- Task Allocation Ratio: $\frac{\text{Number of Tasks Allocated}}{|T|}$
- Fulfillable Task Ratio: $\frac{|T| - \text{Number of Unfulfillable Tasks}}{|T|}$
- Reward Allocation Ratio as defined by Equation 14.

$$\frac{\sum_{\forall t_j \in T} \sum_{\forall p_{jk} \in P_j} x_{jk} \cdot f_j^T}{\sum_{\forall t_j \in T} \sum_{\forall p_{jk} \in P_j} f_j^T} \tag{14}$$

- Algorithm Execution Time Performance Ratio as defined by Equation 15, where x is the algorithm runtime in seconds. Sixty seconds are used as the cut-off for the allocation to be computed; beyond 60 seconds, the execution time performance ratio is 0.

$$1 - \max(0, \min(\log_{60} x, 1)) \tag{15}$$

- Distance Performance Ratio as defined by Equation 16, where x is the average distance per assigned task in kilometers according to Equation 12. D_{max} is used as the cut-off for the average distance per task; beyond it the distance performance ratio is 0.

$$1 - \max(0, \min(\log_{D_{max}} x, 1)) \tag{16}$$

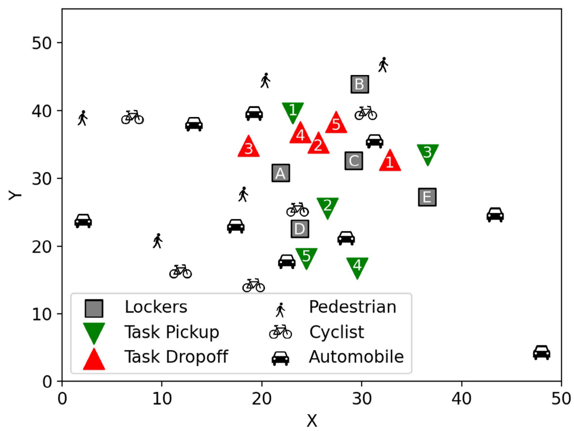

FIGURE 3. Illustration of the workers, lockers, and tasks in the AoI.

IV. EXPERIMENTAL SETUP

A. EXPERIMENTAL TARGET

The work proposed by Ghaderi et al. [23] is the most recent to tackle the relay of crowdsourced workers in last-mile delivery. This work is used as a benchmark to assess the effectiveness of the proposed framework in this paper, HCRA powered by CR-BFS. The authors propose to solve the CR-LMAP by maximizing the task allocation rate using a greedy method, JCA, which finds all possible solutions to the assignment problem, then randomly selects solutions according to an inverse roulette wheel (IRW) probability distribution [23]. IRW gives a higher probability of selection to workers and tasks that have fewer options for assignment. JCA is able to assign tasks with up to two collaborating workers by performing the assignment in two phases: 1) *single delivery* and 2) *collaborative delivery*. Phase 1 finds as many single delivery allocations as possible; then, Phase 2 assigns the remaining tasks through collaborative deliveries between two workers facilitated by a parcel locker as a relay point. For a fair comparison, an extended version of JCA is implemented by adding Phase 3 which takes the unassigned tasks, workers, and lockers following Phase 2 and collaboratively allocates tasks with three steps. Table 3 summarizes Phases 1, 2, and 3. Furthermore, in order to show the effect of CR-BFS on HCRA, it is compared with the performance of HCRA using a constrained random path-finding algorithm defined in Algorithm 3. For clarity, the following naming conventions are used to describe the algorithms' performance results reported in Section V.

- HCRA + CR-BFS: refers to the proposed algorithm HCRA which as defined in Algorithm 1, calls CR-BFS as a subroutine for path-finding.
- HCRA + CR-Rand: refers to a version of HCRA that calls the random tree search, defined in Algorithm 3, as a subroutine for path-finding.
- JCA: refers to the benchmark proposed by Ghaderi et al. [23] and its extension.

Algorithm 3 Crowdsourced Relay Random Tree Search (CR-Rand)

Input: task t, available workers W and lockers L, n^P, n^S
Output: possible paths P_t
Let n_P be the maximum number of paths that should be found for task t and n_S be the maximum depth of the paths.

Let P_t be the set of paths found for task t. P_t = []
Compute worker graph G^t_{WL}
Add (task source node, [], []) to the queue.
while queue is not empty **do**
  if $|P_t| = n^P$ **then**
    // path number limit exceeded
    break
  end if
  Dequeue a randomly selected (node a, p_a, W_a).
  // using a uniform distribution
  // p_a *is the path to* a
  // W_a *is the order of workers on* p_a
  if node depth of $a > n^S$ **then**
    // depth limit exceeded
    continue
  end if
  if node a is task destination node **then**
    Add the path to P_t.
  else
    Get all paths to the adjacent nodes of node a.
    for (worker w, node b) $\in$ set of adjacent paths **do**
      if constraint (5) is violated **then**
        continue
      else
        // x *is the new potential queue member*
        x = (node b, $[p_a, a]$, $W_a \cup w$)
        if $w \notin W_a$ & $x \notin$ visited **then**
          Mark x visited.
          Enqueue x.
        end if
      end if
    end for
  end if
end while
for path $p \in P_t$ **do**
  Compute time, distance, and payoff for all workers.
end for
Sort paths by distance in ascending order.
// To be used in the assignment step

B. DATA PREPARATION

The simulations utilize three combined real-life datasets collected in the state of Sao Paulo in Brazil. These datasets are used to extract the locations of the workers, lockers, and tasks, as detailed next. The resulting dataset represents an AoI grid of 50 km × 55 km. Figure 3 depicts a subset of the extracted workers, lockers, and tasks in the AoI.

TABLE 3. JCA Phase 1, Phase 2, and extended Phase 3 for fair comparison with HCRA.

JCA (Phase 1)	JCA (Phase 2)	JCA (Phase 3)
Input: available tasks T & workers W	**Input:** available tasks T, workers W, & lockers L	**Input:** available tasks T, workers W, & lockers L
Output: updated tasks T & workers task-worker assignment	**Output:** updated tasks T workers W, & lockers L updated task-worker assignment	**Output:** updated tasks T workers W, & lockers L updated task-worker assignment
1: Find all potential crowdshippers $\forall t \in T$	1: Find all potential crowdshipping pairs $\forall t \in T$	1: Find all potential crowdshipping tuples $\forall t \in T$
2: **While** (there are potential task assignments):	2: **While** (there are potential task assignments):	2: **While** (there are potential task assignments):
3: Choose a task t (*rand. IRW*)	3: Choose a task t (*rand. IRW*)	3: Choose a task t (*rand. IRW*)
4: From potential crowdshippers of t:	4: From potential crowdshippers of t:	4: From potential crowdshippers of t:
5: choose w (*rand. IRW*)	5: choose (w_1, w_2) and l (*rand. IRW*)	5: choose (w_1, w_2, w_3) and (l_1, l_2) (*rand. IRW*)
6: Assign t to w	6: Assign t to (w_1, w_2) transferring at l	6: Assign t to (w_1, w_2, w_3) transferring at l_1 then l_2
7: Update T and W	7: Update T, W, and L	7: Update T, W, and L
8: **end while**	8: **end while**	8: **end while**

TABLE 4. Worker capacity generation.

Mode of Transport	Travel Threshold	# of Samples
Pedestrian	2.5 km $\pm$ 0.2 km	15
Bicycle	6.5 km $\pm$ 0.5 km	35
Automobile	9.5 km $\pm$ 1.0 km	70

1) TASKS

The details of the tasks are obtained from the Brazilian E-Commerce Public Dataset by the Olist Store published on Kaggle [38]. Olist, a Brazilian e-commerce platform provides the details for 120 delivery packages, including the seller locations, customer locations, and freight value.

2) LOCKERS

The locations of the large chain of Extra supermarkets, owned by Grupo Pão de Açúcar, were collected by the authors using Google Earth [39]. The extracted locations of 25 stores are used to represent the available locker locations.

3) WORKERS

Check-in locations of users of Gowalla, an international social networking application, collected through their public API by the SNAP group in Stanford University [40]. This dataset provides 120 unique user check-in locations used as the current locations of the delivery workers. Additionally, the workers' vehicles or mode of transport and travel distance threshold are randomly generated. The workers were randomly allocated a mode of transport where 10% of the workers were pedestrians, 30% were bicycle riders, 60% used a motorcycle or a car. The resulting number of samples of each mode is reported in Table 4. Within each mode of transport, the workers' travel distance thresholds were generated using a normal distribution with the corresponding mean and standard deviation. Table 4 defines the parameters of the random distributions used.

C. EXPERIMENT DESCRIPTIONS

This section defines the experiments conducted to evaluate the performance of HCRA + CR-BFS and HCRA + CR-Rand against JCA. Three experiments were conducted:

- Validation of the efficacy of the proposed work in single delivery scenarios ($n^S = 1, |L| = \emptyset$); for example, when no lockers are available.
- Comparative performance of algorithms allowing collaborative delivery with two workers ($n^S = 2, |L| = 25$).
- Comparative performance of algorithms allowing collaborative delivery with three workers ($n^S = 3, |L| = 25$).

In each experiment, the algorithms were compared as the task load on the platform increased and as the density of the worker network varied. In the first set of simulations, the task load on the platform is increased from 10 to 100 tasks while the number of available workers is set to 100. In the second set, the algorithms' behavior is observed as the number of workers on the platform increases from 10 to 100, while the number of available tasks is set to 30.

For all executions of HCRA, the maximum number of paths parameter is $n^P = 5$, and the utility of t_ϕ is $u_\phi = 0.001$. D_{max} is computed to be 24 kilometers. The simulation results are presented in Section V, where each result is the average of 15 random selection simulations. Furthermore, all JCA and HCRA + CR-Rand simulations are repeated 5 times due to the probabilistic nature of their solutions.

D. EVALUATION ENVIRONMENT

The implementation and experiments detailed in the following sections were implemented using Python 3.7.13 on a PC with a 64-bit Windows 10 operating system, 8 GB of RAM, and a 2.60-GHz-Core(TM) i7-based processor.

V. RESULTS

A. VALIDATION ON SINGLE DELIVERY SCENARIOS

The results of the simulations, while the number of tasks and workers varied, are reported in Table 5. In both simulations, HCRA + CR-BFS and HCRA + CR-Rand reach more optimal solutions than JCA within up to 1.43 and 2.04 seconds additional time, respectively. HCRA + CR-BFS improves the overall QoA by an average of 0.61 to 1.43% more than JCA. HCRA + Rand improves the overall QoA by an average of 0.1 to 1.54% more than JCA. The following discussions demonstrate that deploying the proposed algorithm upholds a performance advantage even

TABLE 5. Summary of results in single delivery scenarios. The mean values of each metric are reported. The better-performing algorithm result is highlighted for each test.

Single Delivery						
		Number of Tasks				$\lvert W\rvert = 100$
		20	50	70	100	
Metric	**Algorithm**					**AVG.**
Distance per task (km)	*HCRA+CR-BFS*	4.61	5.23	5.04	**5.06**	4.97
	HCRA+CR-Rand	**4.01**	**4.82**	**4.98**	5.21	**4.7**
	JCA	5.44	5.8	5.87	5.94	5.76
Profit ($/km)	*HCRA+CR-BFS*	3.34	3.08	3.16	**3.05**	3.18
	HCRA+CR-Rand	**3.82**	**3.33**	**3.19**	2.97	**3.39**
	JCA	2.73	2.76	2.71	2.6	2.74
QoA	*HCRA+CR-BFS*	47.46	45.86	**45.89**	44.31	**46.09**
	HCRA+CR-Rand	**48.33**	**46.09**	44.42	42.3	45.58
	JCA	46.32	45.2	45.05	**45.05**	45.48
		Number of Workers				$\lvert T\rvert = 30$
		20	50	70	100	
Metric	**Algorithm**					**AVG.**
Number of Allocated Tasks	*HCRA+CR-BFS*	6.47	8	8	8	7.39
	HCRA+CR-Rand	6.47	8	8	8	7.4
	JCA	**7.04**	8	8	8	**7.53**
Tot. Payoffs ($)	*HCRA+CR-BFS*	108.1	133.78	133.78	133.78	123.68
	HCRA+CR-Rand	108.1	133.78	133.78	133.78	123.71
	JCA	**117.94**	133.78	133.78	133.78	**125.8**
Distance per task (km)	*HCRA+CR-BFS*	**4.29**	4.4	4.46	4.18	4.29
	HCRA+CR-Rand	4.29	**4.19**	**4.13**	**4.08**	**4.21**
	JCA	5.49	5.54	5.56	5.53	5.53
Profit ($/km)	*HCRA+CR-BFS*	**3.95**	3.82	3.79	4.01	3.94
	HCRA+CR-Rand	3.95	**4.03**	**4.08**	**4.12**	**4.02**
	JCA	3.07	3.05	3.04	3.04	3.05
QoA	*HCRA+CR-BFS*	**44.55**	46.32	46.25	46.63	45.63
	HCRA+CR-Rand	44.54	**46.65**	**46.73**	**46.7**	**45.74**
	JCA	43.71	44.88	44.87	44.89	44.2

when relays are not possible with a negligible computational cost.

1) VARYING THE TASK LOAD

With an abundant set of workers, HCRA + CR-BFS, HCRA + CR-Rand, and JCA allocate all the tasks that could be assigned through single-worker delivery. The workers assigned by HCRA + CR-BFS travel about 790 meters per task less than those selected by JCA. The workers assigned by HCRA +CR-Rand travel about 1.06 km per task less than those selected by JCA. Since the same tasks are allocated by all algorithms, the workers allocated by both HCRA algorithms achieve the same total payoffs using a shorter travel distance. Thus, HCRA improves the profit of the workers in single-delivery by 16.2 to 23.7% over JCA, on average. This also validates that using HCRA indeed assigns tasks using shorter paths. The results further indicate that, in single delivery cases with an abundant worker network, using HCRA + CR-Rand can result in more optimal allocations than HCRA + CR-BFS with a negligible additional runtime.

2) VARYING THE WORKER DENSITY

With a relatively low number of tasks, the algorithms' behavior can be classified into two scenarios: sparse worker set and abundant worker set. When the worker set is sparse (with 10 to 40 available workers), not all tasks can be allocated. Due to the greedy nature of the workers and the non-cooperative scenario, the HCRA prioritizes the allocation of higher-paying tasks and lower-distance tasks, which results in a slightly lower task allocation rate. In other words, the workers' self-interested nature is exaggerated,

TABLE 6. Summary of results in collaborative delivery with up to two workers. The mean values of each metric are reported. The better-performing algorithm result is highlighted for each test.

Collaborative Delivery with up to 2 Workers						
		Number of Tasks				$\lvert W\rvert = 100$
		20	50	70	100	
Metric	**Algorithm**					**AVG.**
Number of Allocated Tasks	*HCRA+CR-BFS*	10.53	**21.05**	24.53	29.68	20.25
	HCRA+CR-Rand	10.31	20.27	25.91	32.33	20.85
	JCA	**10.65**	20.77	**26.34**	**33.52**	**21.47**
Number of Unfulfillable Tasks	*HCRA+CR-BFS*	**7.37**	**18.16**	**25.16**	**37.16**	**20.18**
	HCRA+CR-Rand	**7.37**	**18.16**	**25.16**	**37.16**	**20.18**
	JCA	7.71	20.88	31.61	51.03	25.02
Tot. Payoffs ($)	*HCRA+CR-BFS*	**189.07**	430.84	507.58	594.41	407.29
	HCRA+CR-Rand	187.39	**434.08**	**558.86**	**666.91**	**432.89**
	JCA	181.63	375.68	471.54	565.53	375.68
Distance per task (km)	*HCRA+CR-BFS*	12.53	10.08	**8.4**	**7.29**	**9.86**
	HCRA+CR-Rand	**12.39**	**9.89**	9.03	7.72	10.08
	JCA	13.39	11.02	9.71	8.07	10.99
QoA	*HCRA+CR-BFS*	**57.97**	**53.9**	**51.13**	**47.27**	**53.43**
	HCRA+CR-Rand	57.59	51.1	48.52	45.27	51.6
	JCA	56.27	48.37	45.92	43.75	49.64
		Number of Workers				$\lvert T\rvert = 30$
		20	50	70	100	
Metric	**Algorithm**					**AVG.**
Number of Allocated Tasks	*HCRA+CR-BFS*	7.05	11.68	13.37	14.89	11.16
	HCRA+CR-Rand	7.08	11.34	13.01	14.72	11
	JCA	**7.77**	**11.82**	**13.66**	**15.43**	**11.57**
Number of Unfulfillable Tasks	*HCRA+CR-BFS*	**17.47**	**13.95**	**12.89**	**12.42**	**14.56**
	HCRA+CR-Rand	**17.47**	**13.95**	**12.89**	**12.42**	**14.56**
	JCA	20.66	15.28	13.52	12.55	16.06
Tot. Payoffs ($)	*HCRA+CR-BFS*	132.64	**226.18**	**271.6**	**295.85**	**219.41**
	HCRA+CR-Rand	135.42	223.14	263.9	295.11	217.82
	JCA	**136.28**	210.97	255.14	295.03	211.88
Distance per task (km)	*HCRA+CR-BFS*	**5.57**	9.78	**11.21**	**12.67**	**9.34**
	HCRA+CR-Rand	6.14	**9.65**	11.45	12.58	9.6
	JCA	6.78	11.09	12.85	14.07	10.84
Runtime (seconds)	*HCRA+CR-BFS*	0.26	**0.36**	**0.44**	**0.54**	**0.39**
	HCRA+CR-Rand	0.4	0.59	0.75	0.98	0.64
	JCA	**0.03**	0.36	0.88	1.98	0.69
QoA	*HCRA+CR-BFS*	**47.44**	**52.59**	**55.21**	**56.66**	**52.25**
	HCRA+CR-Rand	46.97	52.33	54.54	56.32	51.81
	JCA	44.54	50.45	53.39	52.93	49.56

resulting in a slightly lower total payoff, but individually, the allocated workers maintain a higher profit. As the worker set becomes more abundant (50 or more available workers), HCRA + CR-BFS, HCRA + CR-Rand, and JCA allocate all the tasks that were assignable through single-worker delivery. Similarly to the performance of the previous simulation in Section V-A1, when the worker set became sufficiently large in the AoI, the three algorithms assign the same tasks, and HCRA + CR-BFS selects paths that require about 1.23 km per task less than those assigned by JCA. HCRA + CR-Rand selects paths that require about 1.32 km per task less than those assigned by JCA. This results in a 29.2 to 31.8% increase in profit for the workers assigned by HCRA, on average. These results confirm that, in single delivery cases with an abundant worker network, using HCRA + CR-Rand can result in more optimal allocations than HCRA + CR-BFS with a negligible additional runtime.

B. COLLABORATIVE DELIVERY WITH TWO WORKERS

The results of the simulations, while the number of tasks and workers varied, are reported in Table 6. In both simulations, all algorithms allocated a similar number of tasks in the dataset; however, the HCRA algorithms achieved higher total payoffs to the allocated workers and lower travel distance per task. Moreover, using HCRA lowered the number of unfulfillable tasks as it found potential paths for up to 14 more tasks than JCA.

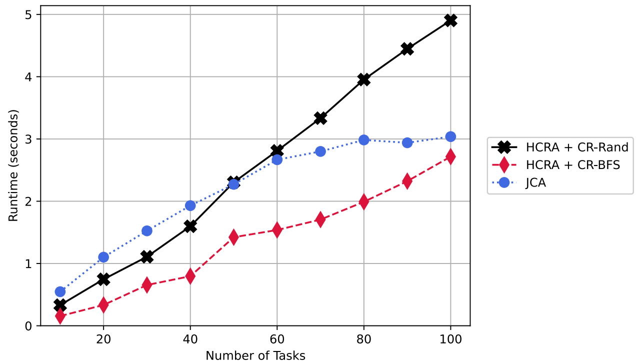

FIGURE 4. The average runtime using collaborative delivery with up to two workers as the number of tasks varied.

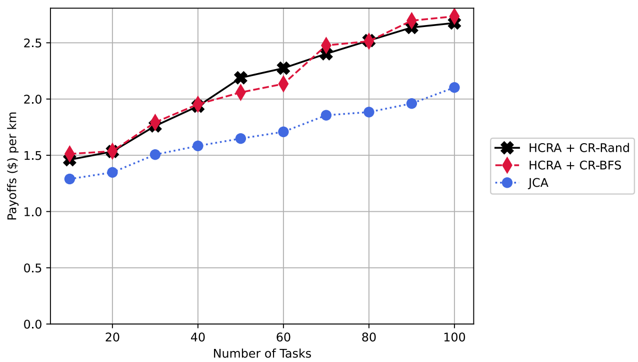

FIGURE 5. The average payoffs per kilometer using collaborative delivery with up to two workers as the number of tasks varied.

1) VARYING THE TASK LOAD

As the number of tasks increased, HCRA + CR-BFS solutions increased the total payoffs of the workers by an average of 8.4% where the mean reward of the assigned tasks was 12.6% more than those of JCA. HCRA + CR-Rand solutions also increased the total payoffs of the workers by an average of 15.2%, where the mean reward of the assigned tasks was 15.8% more than those of JCA. The paths assigned by HCRA + CR-BFS required the workers to travel 1.1 km per task less than those found by JCA. The paths assigned by HCRA + CR-Rand required the workers to travel 900 meters per task less than those found by JCA. Finally, HCRA + CR-BFS achieved its performance advantage over JCA within about 1 second less execution time than of JCA, as shown in Figure 4. Whereas, the execution time of HCRA + CR-Rand exceeds that of JCA as the number of tasks is above 40. Thus, both HCRA algorithms improved the profit of the workers by an average of 26.6% to 26.8% over JCA, as depicted in Figure 5. The advantage of the HCRA + CR-BFS and HCRA + CR-Rand over JCA is also captured by the QoA, which increased by an average of 4.0% and 2.0%, respectively. It is notable that HCRA + CR-Rand achieves a higher total payoff than HCRA + CR-BFS while maintaining a similar profit for the workers as the number of available tasks increases.

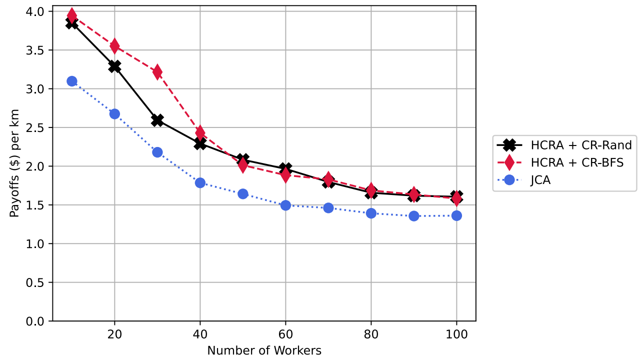

FIGURE 6. The average payoffs per kilometer using collaborative delivery with up to two workers as the number of workers varied.

2) VARYING THE WORKER DENSITY

As the number of available workers in the platform increases, HCRA maintained an advantage over JCA by allocating a similar number of tasks to those allocated by JCA while increasing the total payoffs of the workers and decreasing the average travel distance per task. HCRA + CR-BFS solutions increased the total payoffs of the workers by an average of 3.6% where the mean reward of the assigned tasks was 7.4% more than those of JCA. HCRA + CR-Rand solutions also increased the total payoffs of the workers by an average of 2.8% where the mean reward of the assigned tasks was 8.2% more than those of JCA. The paths assigned by HCRA + CR-BFS required the workers to travel 1.5 km per task less than those found by JCA. The paths assigned by HCRA + CR-Rand required the workers to travel 1.2 km per task less than those found by JCA. The execution time of HCRA + CR-BFS was superior to that of JCA for tests with 60 or more workers and always reached a solution faster than HCRA + CR-Rand. Figure 6 illustrates the results of the algorithms in terms of the payoff per km traveled, demonstrating an average increase of 28.9% and 23.4% using HCRA + CR-BFS and HCRA + CR-Rand, respectively. Therefore, HCRA improved the overall QoA by between 2.3% and 2.7% over JCA.

C. COLLABORATIVE DELIVERY WITH THREE WORKERS

1) VARYING THE TASK LOAD

Due to space and time limitations, the task load simulations with up to three collaborating workers could not be performed using the benchmark algorithm, JCA. This is due to the exhaustive nature of the path-finding strategy of Ghaderi et al. [23]. It cannot find paths with more than two steps within reasonable computational time. However, HCRA + CR-BFS found solutions within 2.3 seconds on average and up to 4.4 seconds when the number of tasks is 100. HCRA + CR-Rand found solutions within 4.2 seconds on average and up to 8.8 seconds when the number of tasks is 100. HCRA + CR-Rand achieves 9.4 % higher total payoffs for the workers than HCRA + CR-BFS.

TABLE 7. Summary of results in collaborative delivery with up to three workers. The mean values of each metric are reported. The better-performing algorithm result is highlighted for each test.

Collaborative Delivery with up to 3 Workers						
		Number of Workers				$\|T\| = 30$
		20	50	70	100	
Metric	**Algorithm**					**AVG.**
Number of Allocated Tasks	*HCRA+CR-BFS*	7.37	**12.58**	**15.21**	16.95	12.13
	HCRA+CR-Rand	7.16	12.47	14.64	**17.47**	12.09
	JCA	**7.58**	12.09	13.67	15.46	11.59
Number of Unfulfillable Tasks	*HCRA+CR-BFS*	**12.26**	**3.32**	**0.95**	**0.63**	**5.31**
	HCRA+CR-Rand	**12.26**	**3.32**	**0.95**	**0.63**	**5.31**
	JCA	20.51	9.4	5.11	3.18	10.8
Distance per task (km)	*HCRA+CR-BFS*	6.57	11.62	13.78	14.88	10.88
	HCRA+CR-Rand	**6.31**	11.81	13.93	16.04	11.35
	JCA	6.43	**11.38**	**12.9**	**13.92**	**10.82**
Profit ($/km)	*HCRA+CR-BFS*	3.05	**1.78**	1.56	**1.46**	**2.15**
	HCRA+CR-Rand	**3.21**	1.77	**1.57**	1.31	2.09
	JCA	2.82	1.6	1.48	1.36	1.86
QoA	*HCRA+CR-BFS*	50.36	**60.39**	**65.06**	**66.67**	**59.22**
	HCRA+CR-Rand	**50.36**	59.69	63.24	65.4	58.24
	JCA	42.87	35.81	39.22	42.47	39.14

2) VARYING THE WORKER DENSITY

The results of the simulations, while the number of workers varied, are reported in Table 7.

When allowing up to three workers to collaborate on a single task, HCRA + CR-BFS and HCRA + CR-Rand enabled the allocation of about 1 additional task and achieved a total payoff that was 17.5% higher than that of JCA, as illustrated in Figure 7. Despite the average additional 60 to 540 meters traveled per task, HCRA + CR-BFS improved the profit over JCA by 15.5% and HCRA + CR-Rand improved it by 11.8%. The additional travel distance, in this case, can be attributed to the fact that higher-paying jobs in the dataset generally required higher travel distances. Moreover, HCRA lowered the number of unfulfillable tasks as it found potential paths for up to 8 more tasks than JCA. Finally, in this scenario, HCRA + CR-BFS and HCRA + CR-Rand provide solutions within a maximum of about 1.1 and 1.6 seconds, respectively. In contrast, JCA requires minutes when the number of workers exceeds 40 and requires about 16.9 minutes to reach a solution when the number of workers is 100. Figure 8 demonstrates the exponential growth of the computational time of JCA when optimizing the allocation with more than two steps. As a result, using HCRA + CR-BFS and HCRA + CR-Rand when allowing up to three workers to collaborate improved the overall QoA by 19.1% and 20.1%, respectively.

D. DISCUSSION

The results validate that when the number of tasks is relatively low and the availability of workers is limited, it is best to use a single delivery solution where JCA would provide fast allocation aiming to maximize the number of allocated tasks. However, with a negligible additional computation time, HCRA provides a multi-objective solution that can easily scale to collaborative scenarios without compromising runtime. As the availability of workers increases, it becomes advantageous to use collaborative delivery. When 25 lockers are available in the AoI, collaborative delivery algorithms improved the task allocation rate over the single delivery algorithms by 14.0% to 21.7%, on average, as the task load varied. In all simulations, both HCRA algorithms indeed yield a better allocation than JCA by assigning the tasks using shorter paths and prioritizing the allocation of higher reward tasks. Another advantage of HCRA can be attributed to its path-finding strategy: finding a few potential paths for as many tasks as possible before assigning any workers. This minimizes the number of unfulfillable tasks and further demonstrates the flexibility of HCRA in handling larger instances of the allocation problem with computational ease. In contrast, JCA utilizes a strategy that first finds all potential paths of single deliveries, allocates workers, and then finds all potential paths of collaborative deliveries with the remaining tasks and workers. The iterative and exhaustive nature of JCA prevents some tasks from being considered for collaborative delivery and requires a longer computation time. The results show that replacing CR-BFS in the proposed algorithm with a random path-finding sub-routine CR-Rand always requires more computation time but only sometimes finds more favorable solutions. This can be attributed to the probabilistic nature of CR-Rand, which may explore more collaborative paths than CR-BFS. For example, if there are many possible single-delivery paths, HCRA + CR-BFS would likely assign the task only through single delivery, whereas HCRA + CR-Rand is not limited in the same way. HCRA + CR-BFS consistently outperforms JCA, faster than HCRA + CR-Rand.

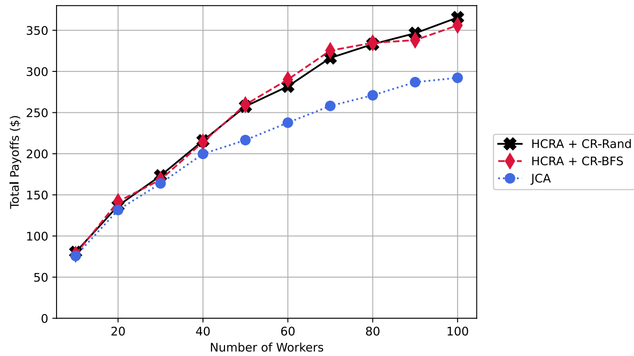

FIGURE 7. The total payoffs using collaborative delivery with up to three workers as the number of workers varied.

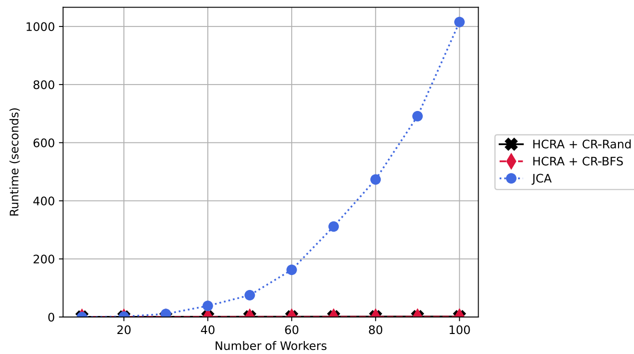

FIGURE 8. The computational time using collaborative delivery with up to three workers as the number of workers varied.

VI. CONCLUSION

In this article, we propose a hedonic game-based model for collaboratively allocating a heterogeneous pool of crowdsourced vehicles. The proposed model is applied within a crowd-relay model for crowdsourced last-mile delivery where the recruited vehicles transfer the task to the next worker at a relay point. The paper defines the crowdsourced relay last mile assignment problem (CR-LMAP) with the aim of maximizing the profit of workers i.e. their returns relative to the cost they incurred. To achieve this goal, a game-theoretic approach is proposed that maximizes the average worker payoff per kilometer. The proposed algorithm, HCRA, utilizes a contained breadth-first search subroutine, CR-BFS, to find potential delivery paths (both single and collaborative) as a basis for the hedonic games used to find optimal task assignments. The notable contribution of this work is the demonstration that the use of the proposed hedonic-based worker assignment method, HCRA, provides 15.5 to 29.2% more profitable solutions for the crowdsourced vehicles in relayed last mile delivery in a reduced computation time than state-of-the-art methods.

A. FUTURE WORK

While the proposed work provides a multi-objective method for optimizing the allocation of collaborating workers, considering economic and worker satisfaction metrics, and promoting fair compensation, several avenues for future research and development remain open. The proposed model and simulations should be enhanced to be adopted in real-life scenarios. In order to maintain a fair comparison with the literature, the proposed algorithm was designed to consider travel distance and payment as the only factors in the allocation. Future studies will include the removal of naive assumptions, such as assuming that the workers' willingness to complete a task is simply in terms of travel distance. In reality, the workers consider their working time windows, carrying capacity, and traffic or weather events. The behavior of the workers is inherently probabilistic. Including machine learning models to predict workers' likelihood to deliver the tasks will enhance the efficacy of the proposed work [14], [41], [42]. Another interesting direction could be the adoption of dynamic task allocation strategies. Investigating the behavior of the proposed algorithm in a dynamic task allocation setting could give insight into the real-time applicability of the algorithm. Exploring real-time adaptation mechanisms such as auctions while dynamically adjusting the allocation process based on the ongoing performance and availability of the crowd could improve the effectiveness of the proposed algorithm.

REFERENCES

[1] Statista. (2021). *Last Mile Logistics Worldwide*. Accessed: Mar. 21, 2023. [Online]. Available: https://www.statista.com/study/49682/last-mile-delivery/

[2] J. Howe, "The rise of crowdsourcing," *Wired Mag.*, vol. 14, no. 6, pp. 1–4, 2006.

[3] A. Ghezzi, D. Gabelloni, A. Martini, and A. Natalicchio, "Crowdsourcing: A review and suggestions for future research," *Int. J. Manage. Rev.*, vol. 20, no. 2, pp. 343–363, Apr. 2018.

[4] O. Urra and S. Ilarri, "Spatial crowdsourcing with mobile agents in vehicular networks," *Veh. Commun.*, vol. 17, pp. 10–34, Jun. 2019.

[5] N. Kafle, B. Zou, and J. Lin, "Design and modeling of a crowdsource-enabled system for urban parcel relay and delivery," *Transp. Res. B, Methodol.*, vol. 99, pp. 62–82, May 2017.

[6] S. Ballare and J. Lin, "Investigating the use of microhubs and crowd-shipping for last mile delivery," *Transp. Res. Proc.*, vol. 46, pp. 277–284, Jan. 2020.

[7] A. M. Arslan, N. Agatz, L. Kroon, and R. Zuidwijk, "Crowdsourced delivery—A dynamic pickup and delivery problem with ad hoc drivers," *Transp. Sci.*, vol. 53, no. 1, pp. 222–235, Feb. 2019.

[8] 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.

[9] K. Huang and M. N. Ardiansyah, "A decision model for last-mile delivery planning with crowdsourcing integration," *Comput. Ind. Eng.*, vol. 135, pp. 898–912, Sep. 2019.

[10] B. Huang, H. Zhu, D. Liu, N. Wu, Y. Qiao, and Q. Jiang, "Solving last-mile logistics problem in spatiotemporal crowdsourcing via role awareness with adaptive clustering," *IEEE Trans. Computat. Social Syst.*, vol. 8, no. 3, pp. 668–681, Jun. 2021.

[11] F. Basik, B. Gedik, H. Ferhatosmanoglu, and K.-L. Wu, "Fair task allocation in crowdsourced delivery," *IEEE Trans. Services Comput.*, vol. 14, no. 4, pp. 1040–1053, Jul. 2021.

[12] F. Wang, Y. Zhu, F. Wang, J. Liu, X. Ma, and X. Fan, "Car4Pac: Last mile parcel delivery through intelligent car trip sharing," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 10, pp. 4410–4424, Oct. 2020.

[13] 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.

[14] M. Kadadha, H. Otrok, R. Mizouni, S. Singh, and A. Ouali, "On-chain behavior prediction machine learning model for blockchain-based crowdsourcing," *Future Gener. Comput. Syst.*, vol. 136, pp. 170–181, Nov. 2022.

[15] N. Elsokkary, H. Otrok, S. Singh, R. Mizouni, H. Barada, and M. Omar, "Crowdsourced last mile delivery: Collaborative workforce assignment," *Internet Things*, vol. 22, Jul. 2023, Art. no. 100692.

[16] Y. Li, M. L. Yiu, and W. Xu, "Oriented online route recommendation for spatial crowdsourcing task workers," in *Proc. Int. Symp. Spatial Temporal Databases*. Hong Kong: Springer, 2015, pp. 137–156.

[17] 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.

[18] 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, May 2018.

[19] 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.

[20] V. F. Yu, G. Aloina, P. Jodiawan, A. Gunawan, and T.-C. Huang, "The vehicle routing problem with simultaneous pickup and delivery and occasional drivers," *Expert Syst. Appl.*, vol. 214, Mar. 2023, Art. no. 119118.

[21] 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.

[22] 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.

[23] H. Ghaderi, L. Zhang, P.-W. Tsai, and J. Woo, "Crowdsourced last-mile delivery with parcel lockers," *Int. J. Prod. Econ.*, vol. 251, Sep. 2022, Art. no. 108549.

[24] C. Zhang, Z. Du, M. D. Parmar, and Y. Bai, "Pocket-switch-network based services optimization in crowdsourced delivery systems," *Comput. Electr. Eng.*, vol. 62, pp. 53–63, Aug. 2017.

[25] S. Castano, A. Ferrara, L. Genta, and S. Montanelli, "Combining crowd consensus and user trustworthiness for managing collective tasks," *Future Gener. Comput. Syst.*, vol. 54, pp. 378–388, Jan. 2016.

[26] Z. Zhao, M. Karimzadeh, F. Gerber, and T. Braun, "Mobile crowd location prediction with hybrid features using ensemble learning," *Future Gener. Comput. Syst.*, vol. 110, pp. 556–571, Sep. 2020.
[27] R. Branzei, D. Dimitrov, and S. Tijs, *Models in Cooperative Game Theory*, vol. 556. Berlin, Germany: Springer, 2008.
[28] O. A. Wahab, J. Bentahar, H. Otrok, and A. Mourad, "Towards trustworthy multi-cloud services communities: A trust-based hedonic coalitional game," *IEEE Trans. Services Comput.*, vol. 11, no. 1, pp. 184–201, Jan. 2018.
[29] H. Abualola and H. Otrok, "Stable coalitions for urban-VANET: A hedonic game approach," *Veh. Commun.*, vol. 30, Aug. 2021, Art. no. 100355.
[30] M. Kadadha and H. Otrok, "A blockchain-enabled relay selection for QoS-OLSR in urban VANET: A Stackelberg game model," *Ad Hoc Netw.*, vol. 117, Jun. 2021, Art. no. 102502.
[31] M. Kadadha, H. Otrok, H. Barada, M. Al-Qutayri, and Y. Al-Hammadi, "A Stackelberg game for street-centric QoS-OLSR protocol in urban vehicular ad hoc networks," *Veh. Commun.*, vol. 13, pp. 64–77, Jul. 2018.
[32] A. Anwar, T. Halabi, and M. Zulkernine, "A coalitional security game against data integrity attacks in autonomous vehicle networks," *Veh. Commun.*, vol. 37, Oct. 2022, Art. no. 100517.
[33] H. Abualola, R. Mizouni, H. Otrok, S. Singh, and H. Barada, "A matching game-based crowdsourcing framework for last-mile delivery: Ground-vehicles and unmanned-aerial vehicles," *J. Netw. Comput. Appl.*, vol. 213, Apr. 2023, Art. no. 103601.
[34] M. Kadadha, H. Otrok, S. Singh, R. Mizouni, and A. Ouali, "Two-sided preferences task matching mechanisms for blockchain-based crowdsourcing," *J. Netw. Comput. Appl.*, vol. 191, Oct. 2021, Art. no. 103155.
[35] J. Shu, X. Liu, Y. Zhang, X. Jia, and R. H. Deng, "Dual-side privacy-preserving task matching for spatial crowdsourcing," *J. Netw. Comput. Appl.*, vol. 123, pp. 101–111, Dec. 2018.
[36] M. Kadadha, R. Mizouni, S. Singh, H. Otrok, and A. Mourad, "Crowdsourced vehicles and UAVs for last-mile delivery application using blockchain-hosted matching mechanism," *Veh. Commun.*, vol. 47, Jun. 2024, Art. no. 100761.
[37] I. Jang, H.-S. Shin, and A. Tsourdos, "Anonymous hedonic game for task allocation in a large-scale multiple agent system," *IEEE Trans. Robot.*, vol. 34, no. 6, pp. 1534–1548, Dec. 2018.
[38] A. Sionek. (2018). *Brazilian e-Commerce Public Dataset by OLIST*. [Online]. Available: https://www.kaggle.com/dsv/195341
[39] (2005). *Google Earth*. Version 9.186.0.0. [Online]. Available: https://www.google.com/earth/
[40] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: User movement in location-based social networks," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2011, pp. 1082–1090.
[41] M. Abououf, S. Singh, H. Otrok, R. Mizouni, and E. Damiani, "Machine learning in mobile crowd sourcing: A behavior-based recruitment model," *ACM Trans. Internet Technol.*, vol. 22, no. 1, pp. 1–28, Feb. 2022.
[42] C. Qiu, A. C. Squicciarini, B. Carminati, J. Caverlee, and D. R. Khare, "CrowdSelect: Increasing accuracy of crowdsourcing tasks through behavior prediction and user selection," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2016, pp. 539–548.

NADA ELSOKKARY received the B.Sc. degree in applied mathematics and statistics and the M.Sc. degree in computer science degree from Khalifa University, Abu Dhabi, United Arab Emirates. Since 2021, she has been a Research Associate with the Department of Computer Science, Khalifa University. She is the author in the fields of crowdsourced last-mile delivery, cryptocurrency, and quantum computing. Her research interests include game theory, artificial intelligence, blockchain, and adiabatic quantum computing.

SHAKTI SINGH (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in electrical and computer engineering from Purdue University, West Lafayette, IN, USA. He is currently an Assistant Professor with the Department of Computer Science, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates. His research interests include semiconductor devices and integrated circuits, sensors, sensing technologies, crowd sourcing, crowd sensing, and development of the IoT and wireless sensor networks.

RABEB MIZOUNI received the M.Sc. and Ph.D. degrees in electrical and computer engineering from Concordia University, Montreal, Canada, in 2002 and 2007, respectively. She is currently an Associate Professor of computer Science with Khalifa University. Her current research interests include blockchain, machine learning, the Internet of Things (IoT), crowdsensing/crowdsourcing, and cloud computing.

HADI OTROK (Senior Member, IEEE) received the Ph.D. degree in ECE from Concordia University. He holds a Professor Position with the Department of Computer Science, Khalifa University of Science and Technology. He is an Affiliate Associate Professor with the Concordia Institute for Information Systems Engineering, Concordia University, Montreal, Canada; and an Affiliate Associate Professor with the Department of Electrical, Ecole de Technologie Superieure (ETS), Montreal. His research interests include the domain of blockchain, federated learning, reinforcement learning, crowd sensing and sourcing, ad hoc networks, game-theoretic models, and cloud security. He is an Editor of IEEE Transactions on Services Computing (TSC) and an Associate Editor of IEEE Transactions on Network and Service Management (TNSM) and *Ad-Hoc networks* (Elsevier). He is a Co-Chaired several committees at various IEEE Conferences.

HASSAN BARADA (Senior Member, IEEE) received the bachelor's, master's, and Ph.D. degrees in electrical engineering from Louisiana State University, USA, in 1984, 1986, and 1989, respectively. Prior to 2008, he held academic posts at Lehigh University, USA, from 1989 to 1995, the King Fahd University of Petroleum and Minerals, Saudi Arabia, from 1995 to 1999, and the Etisalat University College, United Arab Emirates, from 1999 to 2008. He was the Associate Dean for Undergraduate Studies, College of Engineering, from 2010 to 2021. He is currently an Associate Provost with the Faculty Affairs with KU. He has also served as the Acting Dean of the College of Engineering from 2021 to 2022. His research interests include distributed computing with a focus on algorithm design for distributed services and applications. He is a member of the IEEE Computer Society and the IEEE Education Society. He was the Chairperson of IEEE Computer Society, Lehigh Valley Section, USA, from 1992 to 1995.