IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 25 May 2022, accepted 25 June 2022, date of publication 1 July 2022, date of current version 11 July 2022.
Digital Object Identifier 10.1109/ACCESS.2022.3187692

RESEARCH ARTICLE

Profit and Satisfaction Aware Order Assignment for Online Food Delivery Systems Exploiting Water Wave Optimization

FARHANA HUQ[1], (Student Member, IEEE), NAHAR SULTANA[1], (Student Member, IEEE), SUJAN SARKER[2], (Member, IEEE), MD. ABDUR RAZZAQUE[1], (Senior Member, IEEE), AND MOHAMMAD MEHEDI HASSAN[3], (Senior Member, IEEE)
[1]Green Networking Research Group, Department of Computer Science and Engineering, University of Dhaka, Dhaka 1000, Bangladesh
[2]Department of Robotics and Mechatronics Engineering, University of Dhaka, Dhaka 1000, Bangladesh
[3]Research Chair of Pervasive and Mobile Computing, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia

Corresponding author: Md. Abdur Razzaque (razzaque@du.ac.bd)

This work was supported by the University of Dhaka and a Grant for the Research Fellowship funded by the Information and Communication Technology Division, Ministry of Posts, Telecommunications and Information Technology, Government of Bangladesh.

ABSTRACT Online Food delivery, a specialized application of mobile crowdsourcing, has accelerated its popularity due to rushed urban lifestyle in recent times. The food delivery order assignment to workers that maximizes the service qualities, such as maximization of workers' profit and minimization of order completion time to enhance the customer satisfaction at the same time, is a challenging problem. Existing works in the literature are limited either by focusing solely on minimizing order completion time or reduction of cost incurred in delivery of orders. In this paper, we develop a framework for optimal assignment of food delivery orders to workers as a multi-objective linear programming (MOLP) problem that makes a trade-off in between the worker profit and customer satisfaction. Due to NP-hardness of the above MOLP, a polynomial time solution of the food delivery problem has been developed, namely WWOFooD, exploiting Water Wave Optimization, a meta-heuristic algorithm. The WWOFooD not only explores ways of enhancing service quality of the customers, but also ensures giving additional incentives to the workers providing faster and reliable food delivery services. The results of simulation experiments depict that the WWOFooD offers competitive workers' profit as well as significantly enhances customer satisfaction compared to the state-of-the-art works.

INDEX TERMS Mobile crowdsourcing, food delivery, profit maximization, customer satisfaction, water wave optimization.

I. INTRODUCTION

Spatial Crowdsourcing (SC) has become a new form of highly demanding crowdsourcing system to the customers as well as service providers as the use of smart mobile devices with seamless Internet connectivity increases. Such an SC system involves individuals obtaining Spatio-temporal information from diverse applications which are focused on mobile crowdsensing, including online food delivery, ride-sharing, environmental sensing, intelligent transportation system, etc. [1]. Because of today's fast-paced lifestyle, many customers prefer to have foods delivered at their houses or offices as doorstep service for better time management and easier life. Moreover, during the COVID-19 pandemic, there is an increasing tendency to order foods online instead of physically going to restaurants. Therefore, in recent times, food delivery using crowdsourcing has emerged and is being exploited rapidly in many research works [2], [3].

In an Online Food Delivery (OFD) system, customers place their orders through mobile applications or web portals to the online platform. The platform assigns available workers to pick up the order at the service point (a.k.a. restaurants/food

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

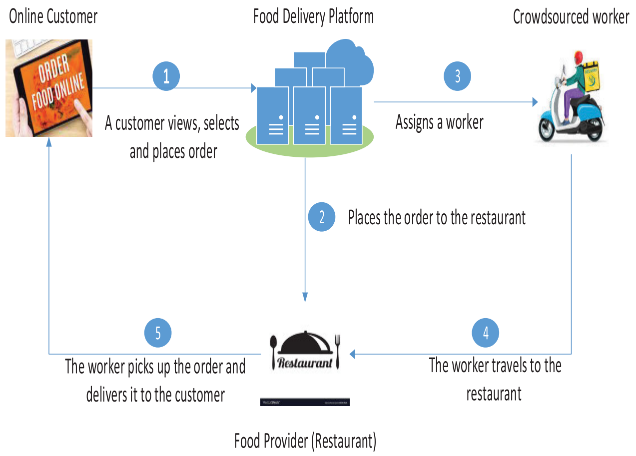


FIGURE 1. A crowdsourced food delivery system.

suppliers) and deliver it to the customer's desired location within an estimated time, as depicted in Fig.1. The number of orders received by the platform varies significantly based on the time of a day. The number of orders received is expected to be much higher in the peak hours (i.e., lunch and dinner time) than that of off-peak hours. An OFD system needs to maintain a sufficient number of workers to ensure timely delivery of the orders as food is a highly perishable product [4]. While faster delivery of orders increases customer satisfaction, a delayed delivery(typically above 1 hour) becomes the reason for customer dissatisfaction, and thus, this is not preferred [5].

The decision-making process of food delivery operations becomes challenging because of the short time span to deliver an order. Furthermore, the quality of end-users' experience, i.e., customer satisfaction, depends on how fast a system can deliver the orders to customers. Thus the fundamental challenge of such a system is to reduce the service time of the food delivery orders to increase customer satisfaction. On the other side, reduction of service time requires assignment of orders to good quality workers, which increases the cost of the whole system. Making a trade-off of these two mutually conflicting objectives under Quality of Service(QoS), budget, and capacity constraints is a challenging research problem.

The issues of food delivery systems are described in [6] which indicated that decisions must be made about assigning appropriate workers (riders) to orders considering worker's associated information. Their centralized system always selected a strong rider, one with high speed and capacity), indicating a lack of proper worker utilization with higher system cost. The authors in [7] generated a set of instances from real-life historical data with a meal delivery routing problem that reduces cost in terms of time and properly utilizes the workers. Even though, the system failed to emphasize workers' income and customer preferences in the order assignment process, which are crucial to the OFD's long-term viability. In [10], the authors, like in previous studies, came up with the idea of reduced, fast, and efficient food delivery by integrating dynamic crowdsourcing with sequential order collection, solution generation and process. However, their approach did not assign orders dynamically to the workers, resulting in longer delivery times and lower customer satisfaction. Similarly, in [11], their system suffered from providing the benefits for workers. The authors only focused on minimizing the delivery cost, which led to selecting unreliable workers, obstructing the system's efficiency. In spite of the fact, it is still unexplored to determine the convenient level of payment that guaranteed the availability of the workers while enhancing customer's satisfaction in terms of promptness of delivery and ensuring the system's profitability. To address the above problem a preliminary version of our work focusing on optimal worker selection was published in [36].

In this work, we develop an order assignment framework that maximizes profit and minimizes the service time of the worker as well as increases the satisfaction of the customers at the same time. We propose a metaheuristic algorithm based on Water Wave Optimization (WWO) to solve the problem. We design an adaptive payment policy based on a relative customer satisfaction index to incentivize a worker to complete an order more rapidly.

The key contributions of this paper are summarized as follows:

- We formulate a multi-objective linear programming (MOLP) objective function with certain necessary constraints that assign the order to a worker that maximizes the profit of workers by allocating multiple orders and enhances customers' satisfaction by minimizing the service time.
- Due to the NP-hardness of the proposed optimal system, we develop a WWO based meta-heuristic assignment algorithm for the online Food Delivery system (WWOFooD) that balances worker's profit and customer satisfaction by selecting appropriate workers to complete the orders.
- We establish a payment policy supporting the interest of the assigned workers to get additional profit for providing faster and reliable food delivery services.
- We perform an extensive simulation of the proposed WWOFooD. The results show significant performance improvements over other state-of-the-art works in terms of average worker profit, customer satisfaction, average service time.

The remainder of this paper is organized as follows. In Section II the works that are related to our problem are presented. The system model is presented in Section III. Section IV formulates the proposed order assignment problem. After that, Section V presents the proposed meta heuristic based order assignment algorithm. Section VI presents the simulation environment and experimental results of the proposed mechanisms with comparative analysis. Finally, the paper is concluded in Section VII.

II. RELATED WORK

A comprehensive research has been carried out for reducing the challenges of assigning workers to a task in various type of mobile crowdsourcing applications [8], [9], [14], [15], [16], [17]. In recent years, the object delivery problem (e.g.,people, goods) with spatial crowdsourcing have emerged, such as service sharing with multiple people and organizing community development services [2], [31], and also rigorous research has been investigated to make these system efficient and sustainable [21], [28], [33], [34]. In this section, we study the existing literature on worker assignment in crowdsourced online food delivery system by broadly categorizing those into two subgroups. The first group includes the works those focus on maximization of customer satisfaction. The second group of works concentrate on maximizing the worker revenue while assigning orders to workers.

In the literature, several works aimed at minimizing workers' traveling time and customers' waiting time to achieve higher customer satisfaction. Lu *et al.* [29] addressed the order assignment and routing problem in the OFD system and employed evolutionary algorithm-based solutions to minimize the total travel distance, resulting in a shorter customer waiting time. However, they did not assign multiple workers to orders from the same restaurant, and the proposed solution methods incur a high computational burden. In [27], the authors proposed a task grouping method considering the order arrival time and assigned workers to those groups for ensuring faster delivery. But, they did not assign workers considering the order pickup and delivery locations and failed to estimate the worker service time and cost based on the real-time traffic conditions. A data-driven framework is proposed in [32] to solve the last-mile problem (LMP) in the food delivery system, and the solution is realized based on worker behavior analysis. However, the pickup of the ordered food was not incorporated, making it difficult to solve LMP in real-life complex scenarios.

To study the meal-delivery routing problem (MDRP), authors in [12] developed a solution approach as introduced in [7]. However, both works minimized the delivery time but overlooked the worker's profit. A dial-a-delivery rider model for handling highly variable customer demand is developed in [20] that aims at minimizing the service time of the riders. However, they only focused on platform profit by reducing the delivery cost while keeping the worker profit and customer satisfaction untreated. FooDNet [13] assigned existing taxi drivers to deliver food items at a lower cost. However, the food delivery was limited only to the taxi routes and proved costly for long-distance delivery. Moreover, it became inconvenient for short-distance delivery, thus resulting in a low customer satisfaction rate. The authors in [7], introduced a dynamic deterministic model of the structure and functioning of meal delivery systems and developed a rolling-horizon repeated-matching algorithm to solve the problem in near real-time. FOODMATCH [11] offered a shorter delivery time and higher order delivery rate compared to [7], while reducing worker waiting time at restaurants. As the system did not consider worker reputation and incentive mechanisms, they suffered from the cancellation of assigned tasks by the workers. This, in turn, negatively affects the system by reducing customers' satisfaction.

A multi-objective non-linear optimization model was proposed to minimize the overall operational cost and balance the workload among multiple workers in [4]. Their focus was workload management for the system, and they did not consider customer satisfaction and worker profit. Wei Tu et [10] developed a mathematical model of the crowdsourced delivery problem to reduce the total travel cost and delivery delay. The proposed approach assigns riders to tasks and selects delivery routes by integrating dynamic crowdsourcing with sequential order collection and rider assignment. However, as orders and riders join the system dynamically, the sequential approach may end up with sub-optimal assignment resulting in a significant delivery delay. Moreover, the system failed to incentivize the workers in the long term as the workers' profit was remained untreated.

A non-cooperative sequential game considering each worker as a player is proposed in [6] to maximize the worker revenue. The centralized system selected a strong rider (with high speed and capacity) only, thus under utilizing the other available workers and increasing delivery costs. Moreover, they did not consider the delivery deadline of orders, thus may achieve poor customer satisfaction for delayed delivery. In [25], the authors proposed a hybrid evolutionary algorithm that adapted two metaheuristics to maximize revenue and estimate order ready time. However, instead of assigning an order to a worker, they focused on allowing workers to select an order from the available order list.

The research works discussed above did not simultaneously consider worker profit and customer satisfaction. Moreover, they did not assess the reliability of the workers while assigning orders. The key philosophy of this work is to assign an order to a reliable worker so that it can be accomplished in minimal time, yielding a maximum profit for the workers. The proposed system provides workers with additional payment for rendering quick service, which attracts more workers to participate and, at the same time, enhances the customers' satisfaction in the OFD system.

III. SYSTEM MODEL AND ASSUMPTIONS

We consider an online food delivery (OFD) system that consists of four entities: a platform, customers, food providers, and crowdsourced workers. The platform centrally coordinates the interaction among the other three entities connected to it through a registration process. Customers place orders of food items from their preferred restaurants through mobile apps or web portals. Let, J denotes the set of orders received from the customers. Each order $j \in J$ is a five parameter tuple, $< l_j^p, l_j^d, L_j, w_j, s_{max} >$, where, l_j^p is the pickup location, l_j^d is the delivery location, L_j is the list of food items, w_j is the total weight, and s_{max} is maximum delay tolerance of order j. The WWOFooD system facilitates

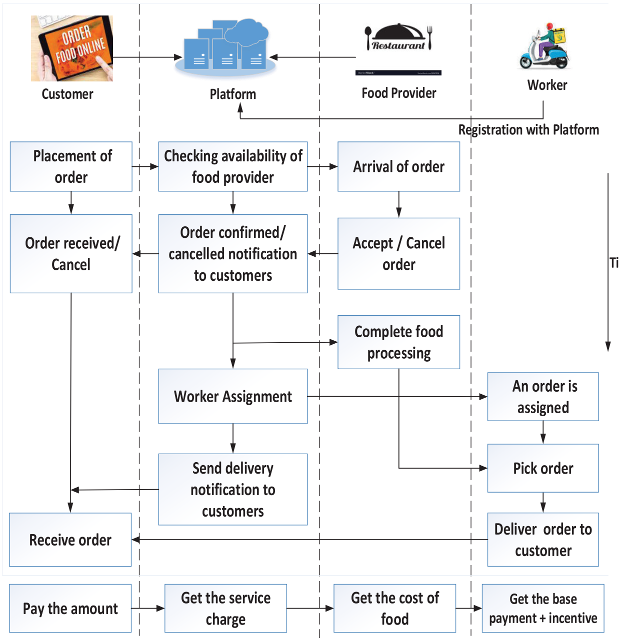


FIGURE 2. Food Delivery Workflow in WWOFooD System.

TABLE 1. List of notations.

Symbol	Meaning
J	The set of orders $\{j_1, j_2, ...j_n\}$
l_j^p	Location of the pick up point for j-th order
l_j^d	Location of the delivery point for j-th order
$t_{j,k}$	Estimated time to deliver j-th order
K	The set of workers $\{K_1, K_2, ...K_m\}$
l_k	Location of the worker at the starting time
q_k	Maximum capacity of the k-th worker
r_k	Reputation of the k-th worker
h_k	Availability rate of k-th worker in peak hour
ρ_k	Orders completion rate of k-th worker
ξ_k	Rejection rate of k-th worker
$c_{j,k}$	Cost of k-th worker to complete an order
$p_{j,k}$	Profit earned by a worker k-th worker
$m_{j,k}$	Payment earned by k-th worker after completing of a order
$s_{j,k}$	Time required by a worker to travel from the service point to customer's location of delivery
s_{max}	Maximum delay tolerance of an order

the customers to input a maximum delay deadline s_{max} for an order j and the platform ensures that it is delivered within this deadline.

The crowdsourced platform relies on the crowdsourced workers for facilitating the delivery of the orders placed by the customers. The platform keeps track of the set of registered workers, K. Each worker $k \in K$ is attributed by a four parameter tuple $< l_k, q_k, \omega_k, r_k >$, where l_k is the current location, q_k is the maximum carrying capacity, ω_k is the current workload, and r_k is the reputation of worker k. In the WWOFooD system, a worker k can pick up orders from multiple food providers and thus are not restricted to any particular provider. The maximum carrying weight of a worker k is limited to q_k so as to prevent delivery delay due to a worker being overloaded with orders.

Fig. 2 depicts the interactions among the entities of WWOFooD system. The platform receives orders directly from the customers. After reception of an order j, the platform sends the request to the food provider to accept or cancel the order. Based on their food processing capacity and/or availability of an ordered item, the food provider accepts or cancels the order; consequently, the platform sends a confirmation/ cancellation notification to customer.

After getting confirmation from the food providers the platform constructs the order set, J and recruits workers from K to deliver the orders. Instead of assigning workers order by order, the WWOFooD runs the worker assignment procedure for a batch or orders collected within a predefined scheduling interval, t_s. t_s is adjusted dynamically according to the order arrival rate in the system. While assigning a worker $k \in K$ to a order j, the service time, $\widetilde{s}_{j,k}$ is estimated from the traveling time from worker's current location to pickup location $\widetilde{t}^p_{j,k}$, waiting time at pickup point, $\widetilde{t}^w_{j,k}$, and travelling time from the pickup point to delivery location, $\widetilde{t}^d_{j,k}$. Note that, worker's traveling varies based on the road traffic condition and can be estimated from the Google Map API [10]. Being assigned by the platform, a worker k picks up orders of total assigned order set, $J_k \subseteq J$ from food providers and delivers those to the customer within the s_{max} of all order $j \in J_k$. The customers provide payment for a delivered order directly to the workers in cash or to the platform through online payment. To make the system sustainable, the appropriate share of the revenue is also distributed among the three entities mentioned above. The proposed system assumes that the choice of restaurants, as well as the decision regarding food quality, is solely based on the customers. These issues are beyond the scope of this work. We provide the major notations used throughout this paper in Table 1.

IV. OPTIMAL ORDER ASSIGNMENT FRAMEWORK

In this section, we first present the computational modules of the order assignment system and unfold the functionality of major modules in details.

A. COMPUTATIONAL MODULES OF ORDER ASSIGNMENT FRAMEWORK

Figure 3 shows the functional blocks of the proposed order assignment framework, unfolded as follows.

1) Worker Registrar module communicates with the worker interface to register the available workers and stores their information such as current location, ongoing orders, and capacity in the worker information database.

2) Order Receiver module collects orders from customers through the customer interface. To facilitate order of food items, the available restaurant list and their food menu are displayed in the customer interface fetched from the restaurant information database. Upon confirmation of an order, the information is also stored in the order information database.

3) Restaurant Manager module registers the restaurants in the system and populate the restaurant information database with updated information. It also provides the estimated food

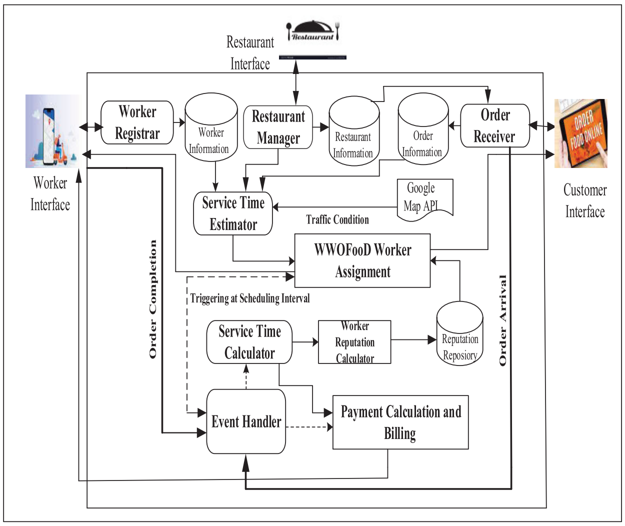


FIGURE 3. **Computational Modules of Order Assignment Framework.**

preparation time when required by the service time estimator module.

4) Service Time Estimator module calculates the time that is estimated to be spent by the workers to deliver the assigned orders. As the estimated time calculation depends on the real traffic scenario, it consults with the Google Maps API to get current traffic conditions. The estimated food preparation time is fed from the respected restaurant through the restaurant manager.

5) WWOFooD Worker Assignment module assigns workers to deliver the food items being triggered by the event handler. To facilitate this, the estimated service time and worker reputation values are fetched from the service time estimator module and worker reputation database, respectively. After completing the assignment procedure, it notifies the workers and customers about the assignment.

6) Service Time Calculator module calculates the actual service time of a worker when he completes an order.

7) Reputation Calculator module updates the reputation value of an order after completion of his assigned orders and store it in the reputation database.

8) Payment Calculation and Billing module calculates the payment of a worker for his completed delivery. This module also determines the satisfaction level of a worker and calculate the additional payment, i.e., profit of the worker.

9) Event Handler module triggers the functionalities of different module required to render the services of the food delivery system.

B. SERVICE TIME AND COST CALCULATION

Delivery time is a crucial factor in food delivery system as it affects the satisfaction level of the customers as well as the profit of the workers. To address this issue, we define the service time of a worker as the combination of his total travelling time and waiting time for processing the order(food). Figure 4 illustrates different events with the required timing consideration while delivering an order j by a worker k. Let, l_k, l_j^p, and l_j^d denote the current location of worker k, pickup location and delivery location of order j, respectively. Now, the service time of a worker $k \in K$ for serving the order $j \in J$ is calculated as follows,

$$s_{j,k} = t_{j,k}^p + t_{j,k}^w + t_{j,k}^d. \tag{1}$$

Here, $t_{j,k}^p$ is the travelling time from l_k to l_j^p, $t_{j,k}^w$ is the waiting time to process an order j and $t_{j,k}^d$ is the travelling time from l_j^p to l_j^d. The platform estimates the delivery time $\widetilde{s}_{j,k}$ based on the service time, $s_{j,k}$ required by the worker k. Note that, estimating the travelling time, $\widetilde{t}_{j,k}^p$, and $\widetilde{t}_{j,k}^d$ has a significant dependency on the road network distance and can be predicted from Google Maps API. Again, the system can also estimate the food processing time, $\widetilde{t}_{j,k}^d$ from the past historical data [7]. Thus, we calculate the estimated delivery time, $\widetilde{s}_{j,k}$ as follows,

$$\widetilde{s}_{j,k} = \widetilde{t}_{j,k}^p + \widetilde{t}_{j,k}^w + \widetilde{t}_{j,k}^d. \tag{2}$$

Our system relies on crowdsourced workers who usually collects and delivers orders using vehicles such as bicycle and motorcycle to leverage the urban travel time.

The cost, $c_{j,k}$ is determined by a worker k for delivering an order j depends on the travelled distance and time that is spent by the worker, and it is calculated as follows,

$$c_{j,k} = \mu \times (d_{l_j^k, l_j^p} + d_{l_j^p, l_j^d}), \tag{3}$$

where, μ is the per unit travel cost. In real time scenario, travelling distance of a worker highly depends on the traffic condition of road, thus it may vary time to time. Our system estimates the delivery cost, $\widetilde{c}_{j,k}$ by considering $d_{l_j^k, l_j^p}$ and $d_{l_j^p, l_j^d}$ as the distances of shortest routes available between the corresponding locations. Note that if multiple orders at the same restaurant are assigned to a single worker, the cost calculation for the current order takes into account the distance from the prior order's delivery location to the current order's delivery location rather than considering the distance from the restaurant each time. In the case that multiple orders from different restaurants are assigned to a worker, for each subsequent order, the cost calculation considers the distance from the worker's current location to the pick up point and from there to the delivery location.

In this system, a worker k is allowed to deliver multiple orders at a time considering his carrying capacity, and pick up and delivery locations associated to the orders. Worker with a preassigned order set calculates his service time and cost for current orders, considering the previous ones. Algorithm 1 summarizes the steps of service time and cost calculation for a worker j for an assigned set of orders, J_k. First we construct the total workload set, J_k of worker k by merging worker's undelivered order with the current assigned orders in line 3. Next, we extract the locations of interest (LoI), i.e, worker current location and pickup and delivery locations of different orders in line 4. Next, we initialize location flag V_l for all locations $l \in L_k$ to indicated that initially they are not visited

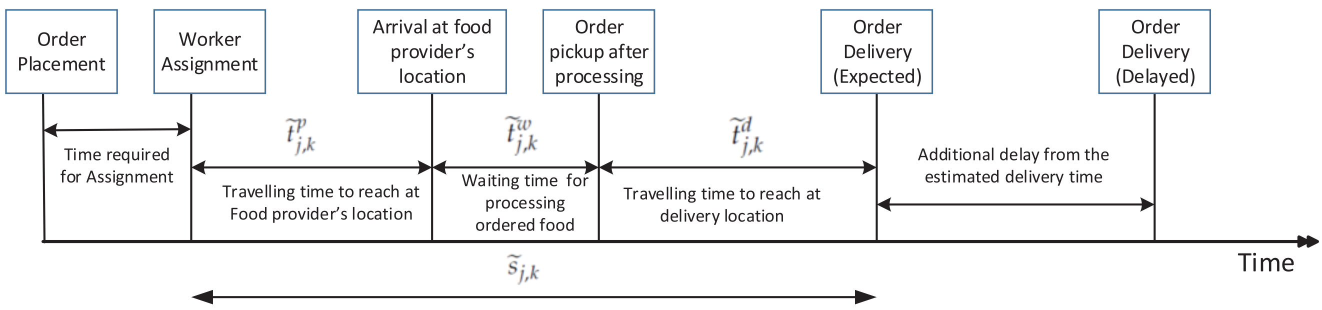


FIGURE 4. **Timing Diagram of Food Delivery.**

in Line 5. In line 6, we estimate the distance between all pair of locations. Line 7 and 8 initialize worker's current location and loop counter. We select the nearest location l' form the current location of worker k in Line 10. If l' is the pick up location of an order j, $\widetilde{s}_{j,k}$ and $\widetilde{c}_{j,k}$ are calculated in Line 13. In case l' is a delivery location (Line 14), we check whether its pick up location is visited or not in Line 15. If the pick up location of l' is visited then we estimate its delivery time and cost in Line 16, otherwise we continue with the next location in Line 18. After that, worker current location, k and location flag, V'_l, and loop counter c are updated in Line 21. Lines 10 to 21 iterate until all the locations of interest are visited. Finally, we calculate the total estimated service time and cost respectively $\widetilde{S}_k$ and $\widetilde{C}_k$ for all the assigned workload of worker k in Line 23.

The complexity of Algorithm 1 is analyzed next. In statement 1, we initialize service time and cost for each order $j \in J_k$, both of which takes $|J_k|$ times. Statement 3 merges two sets that iterates $|J_k|$ times in worst case. Statement 4 requires $|J_k|$ times to extracts locations associated to each order j. Statement 5 also takes $|J_k|$ times to initialize location flags. Statement 6 assigns the distance of each location pair which requires $|J_k|^2$ times. In statement 10, we find the location with minimum distance from worker's current location in $|J_k|$. times. Statements 11 to 21 are enclosed in a loop that iterates $|J_k|$ times. Finally, statement 23 sums up the estimated service time and cost for all the order j and each takes $|J_k|$ times. All the other statements have constant time complexity. Thus the computational complexity of Algorithm 1 is $O(|J_k| + |J_k| + |J_k| + |J_k| + |J_k|^2 + |J_K|^2 + |J_K|) \approx O(|J_k|^2) \approx O(|J \times K|^2)$.

Algorithm 1 Calculation of $\widetilde{S}_k$ and $\widetilde{C}_k$.

Input: J_k: Set of orders of worker k

Output: $\widetilde{S}_k$: Service time of worker k, $\widetilde{C}_k$: Delivery cost of worker k

1: $\widetilde{s}_{j,k}, \widetilde{c}_{j,k} \leftarrow 0, \forall\, j \in J_k$
2: $J'_k \leftarrow$ Undelivered order set of k
3: $J_k \leftarrow J_k \cup J'_k$
4: $L_k \leftarrow ExtractLoc(J_k)$
5: $V_l \leftarrow 0, \forall l \in L_k$
6: $d_{l,l'} \leftarrow$ Distance between l and l', $\forall l, l' \in L_k$
7: $l_k \leftarrow$ Current location of k
8: $c \leftarrow 0$
9: **while** $(c! = |L_k|)$ **do**
10: $\quad l' \leftarrow \underset{\forall l \in L_k, l \neq l_k}{\arg\min}\ (d_{l_k,l})$
11: $\quad j \leftarrow$ Order associated to l'
12: $\quad$ **if** $(l' == l^p_j)$ **then**
13: $\quad\quad \widetilde{s}_{j,k} \leftarrow \widetilde{s}_{j,k} + \widetilde{t}_{l_k,l'} + \widetilde{t}^{w}_{j}, \widetilde{c}_{j,k} \leftarrow \widetilde{c}_{j,k} + x \times d_{l_k,l'}$
14: $\quad$ **else if** $(l' == l^d_j)$ **then**
15: $\quad\quad$ **if** $V_{l^p_j} == 1$ **then**
16: $\quad\quad\quad \widetilde{s}_{j,k} \leftarrow \widetilde{s}_{j,k} + \widetilde{t}_{l_k,l'}, \widetilde{c}_{j,k} \leftarrow \widetilde{c}_{j,k} + x \times d_{l_k,l'}$
17: $\quad\quad$ **else**
18: $\quad\quad\quad$ Go to Line 8
19: $\quad\quad$ **end if**
20: $\quad$ **end if**
21: $\quad l_k \leftarrow l', V'_l \leftarrow 1, c \leftarrow c + 1$
22: **end while**
23: $\widetilde{S}_k = \sum_{\forall j \in J_k} \widetilde{s}_{j,k}, \widetilde{C}_k = \sum_{\forall j \in J_k} \widetilde{c}_{j,k}$

C. PROFIT CALCULATION

The profit $p_{j,k}$ of a worker $k \in K$ for delivering an order $j \in J$ is calculated as the difference between the worker's payment, $m_{j,k}$ received from the platform and worker's service cost, $c_{j,k}$. Again, the actual service cost of a worker k can not be known before completing the delivery of the order, thus the system calculates the estimated profit $\widetilde{p}_{j,k}$ of worker k for serving order j as follows,

$$\widetilde{p}_{j,k} = m_{j,k} - \widetilde{c}_{j,k} \tag{4}$$

Here, $m_{j,k}$ is the portion of platform's income from a order j that is paid to worker k. Note that, $m_{j,k}$ should be greater than $\widetilde{c}_{j,k}$, i.e., $\widetilde{p}_{j,k} > 0$ which incentivizes the workers to render good quality service. The service cost of a worker is directly proportional to his travel distance, thus lowering the cost also reduces service time which in turns increases the satisfaction of the customers. Increasing the quantity of orders completed by workers enhances profit for both parties. As a result, successful completion of multiple orders by a worker within a given delivery time frame yields increase in profit both for the platform and workers.

D. WORKER'S REPUTATION CALCULATION

We consider the reputation, r_k of a worker, $k \in K$ to ensure his reliability while assigning an order. A worker k

can only be assigned to an order j if r_k attains a minimum reputation threshold, $R_{th} \in [0, 1]$ which is set by the platform. We calculate the reputation r_k of a worker k as the linear combination of three metrics as follows,

$$r_k = \beta \times \rho_k - \gamma \times \xi_k + (1 - \beta - \gamma) \times h_k, \tag{5}$$

where, $\beta, \gamma \in [0, 1]$ and $\beta + \gamma \leq 1$. Note that, platform assigns the value of β and γ to set the relative priority among the metrics. Here, ρ_k denotes the success rate of worker k that is the ratio of successfully completed orders to worker's total assigned orders. ξ_k is the rejection rate that measures the ratio of rejected orders to total assigned orders. h_k denotes worker's availability rate at peak time, that is, the ratio of worker active duration in peak hours to the total peak hour duration. Workers who are serving more orders in peak hours can increase their peak time demand value.

E. OPTIMAL ASSIGNMENT OF ORDERS

This system aims to assign workers with a subset of orders in such a way that increases the profit of the crowdsourced workers as well as decreases the service time of the delivery. Thus, we formulate optimal order assignment problem as follows,

$$Z = \underset{\forall b \in \mathcal{A}}{\arg\max} \sum_{\forall k \in K_b} (\hat{p}_k - \hat{s}_k) \tag{6}$$

Subject to:

$$\widetilde{p}_{j,k} \geq 0, \quad \forall (j, k) \in b \tag{7}$$

$$\widetilde{s}_{j,k} \leq s_{max}, \quad \forall (j, k) \in b \tag{8}$$

$$r_{j,k} \geq R_{th}, \forall (j, k) \in b \tag{9}$$

$$\sum_{\forall j:(j,k) \in b} \omega_{j,k} < q_k, \quad \forall k \in K_b \tag{10}$$

Here, $K_b = \{k : (j, k) \in b\}$ and $\mathcal{A}$ is the power set of $J \times K$. $\hat{p}_k$ and $\hat{s}_k$ denote the normalized profit and service time which are calculated as follows,

$$\hat{s}_k = \frac{1}{|J_k|} \times \sum_{\forall j \in J_k} \frac{\widetilde{s}_{j,k}}{s_{max}} \tag{11}$$

$$\hat{p}_k = \frac{1}{|J_k|} \times \sum_{\forall j \in J_k} \frac{\widetilde{p}_{j,k}}{m_{j,k}} \tag{12}$$

Note that, $0 \leq \hat{s}_k, \hat{p}_k \leq 1$. Here, $s_{max} = \widetilde{s}_{j,k} \times \zeta$ where $1 < \zeta \leq 2$.

Constraint (7) is worker profit constraint that indicates that the profit of a worker for an order must be greater than zero. Constraint (8) is the order deadline constraint that ensures that the estimated service time of a worker k for an order j must not exceed the order's deadline. Constraint (9) is worker reputation constraint that restricts the workers with a reputation value less than the threshold from being assigned to any order. Constraint (10) is worker capacity constraint that guarantees the total assigned orders must not exceed the carrying capacity of the worker.

Theorem: The order assignment problem formulated in Eq. (6) is NP-Hard.

Proof: The order assignment problem in Eq. (6) is a MOLP problem which seeks to find an optimal subset $b \in J \times K$ so as to maximize the total normalized profit and to minimize the normalized service time. In the literature, it is known as an optimal subset selection problem, which is a well known *NP-Hard* problem [35]. Thus, the order assignment problem formulation in Eq. (6) is *NP-Hard*.

V. META HEURISTIC ORDER ASSIGNMENT

The optimal order assignment problem, in the previous section, is proven to be NP-Hard, thus no solution can be found in polynomial time. However, it is critical to develop an acceptable solution within a reasonable time frame in order to meet customers' real-time service expectations. In this section,to solve the order assignment problem in polynomial time, we employ a meta heuristic-based approach called WWOFooD. Meta-heuristics often identify good solutions with less computation than optimization algorithms by employing iterative approaches or simple heuristics since they search a large set of feasible solutions. The WWOFooD system obtains the near optimal solution of the assignment problem by utilizing water wave optimization (WWO) [24] which is well explored for discrete combinatorial optimization in various domains [22], [23], [25]. The cardinal principle of WWO is to assign each wave (i.e, solution) to a wavelength inversely proportional to its fitness and let it to propagate in a range proportional to the wavelength. Therefore, the solution with low fitness explore in large spaces while the high fitness solution exploit in small spaces to attain a satisfactory positive balance between diversification and intensification. In subsequent sections, we elaborate the details of WWO-based order assignment.

A. MAPPING ORDER ASSIGNMENT PROBLEM TO WWO

The WWO starts with a population, $\mathcal{P}$ of waves where each wave is analogous to a solution of the problem. In WWOFooD, we define a solution as a $|J| \times |K|$ vector, X, where each element $X_{j,k}$, $0 \leq j \leq |J|, 0 \leq k \leq |K|$ represents a decision variable. $X_{j,k} = 0$ if order j is assigned to worker k, or 0 otherwise. Now, to assess the quality of a solution X, we design a fitness function that fulfils the requirement of order assignment problem as follows,

$$\Delta f(X) = \begin{cases} \frac{1}{K_x} \times \sum_{\forall k \in K_x} (\hat{p}_k - \hat{s}_k) & \text{if satisfies} \\ & \text{Eq. (7) to (10)} \\ -1 & \text{otherwise,} \end{cases} \tag{13}$$

where, K_x is the set of workers who has at least one assigned order in X, i.e., $\forall k \in K_x$ there are some $j \in J$ such that, $X_{j,k} = 1$. Note that, Eq. (13) assigns higher fitness to a solution that assigns the workers with multiple order while obeying the required constraints, otherwise, it provides a negative fitness (-1). Each wave X is then propagated in the search space and it's propagation range is limited by the wavelength, λ_X calculated as follows,

$$\lambda_X = \lambda_{max} \frac{\sum_{\forall X' \in \mathcal{P}} \Delta f(X') - \Delta f(X)}{\sum_{\forall X' \in \mathcal{P}} \Delta f(X')} \tag{14}$$

Here, λ_{max} is the maximum allowable wavelength whose value is set as the total number of orders. The propagation operator selects a random solution, X from $\mathcal{P}$ and performs λ_X local searches on it. To intensify the searching of optimal solutions, neighborhood search based breaking operator is used.

B. WWO-BASED ORDER ASSIGNMENT ALGORITHM

Algorithm 2 Initial Feasible Solution Generation.

Input: J: Set of orders, K: Set of workers
Output: X: Initial feasible solution

1: $X_{j,k} \leftarrow 0, \forall j \in J, \forall k \in K$
2: $F \leftarrow \phi$
3: **for** each $j \in J$ **do**
4:   **for** each $k \in K$ **do**
5:     Calculate $\widetilde{s}_{j,k}$ and $\widetilde{p}_{j,k}$ using Eq.(2) and (4), respectively
6:     $F \leftarrow F \cup < j, k, \hat{p}_{j,k} - \hat{s}_{j,k} >$
7:   **end for**
8: **end for**
9: Sort F in descending order of $\hat{p}_{j,k} - \hat{s}_{j,k}$
10: $a \leftarrow rndInt\{0, |F|\}$
11: $c \leftarrow 0$
12: **while** $(c \leq |F|)$ **do**
13:   $j \leftarrow a.j, k \leftarrow a.k$
14:   **if** $(\widetilde{s}_{j,k} \leq s_{max} \&\& r_{j,k} \geq R_{th} \&\& \widetilde{c}_{j,k} < m_{j,k})$ **then**
15:     $X_{j,k} = 1$
16:     $c \leftarrow c + 1, a \leftarrow (a + 1)\%|F|$
17:   **end if**
18: **end while**
19: **return** X

The steps of initial feasible solution generation algorithm is summarized in algorithm 2. First we initialize each assignment variable $X_{j,k}$ to zero in line 1. Line 2 initializes a set F for storing the utility of each order-worker pair (j, k). For each order-worker pair (j, k), we estimate the order delivery time, $\widetilde{s_{j,k}}$ and worker profit, $\widetilde{p_{j,k}}$ in line 5. Line 6, calculates the order-worker assignment utility value, $\hat{p}_{j,k} - \hat{s}_{j,k}$ and append it to F. Next we sort F in descending order of the utility values. Then select a random index a of F in line 10. Starting from a, for each pair, (j, k) we check whether this assignment satisfies the required constraints in line 14. If the constraints are obeyed the order j is assigned to worker k, i.e., $X_{j,k} = 1$ in line 15. This process repeats until all indices of F is visited. Finally, a feasible solution is returned in line 19. Note that, starting from the random position ensures that the initial population $\mathcal{P}$ contains diverse solutions which facilitates the better exploration of the search space.

Algorithm 3 Water Wave Optimization Based Worker Assignment.

Input: X: Feasible solution
Output: X^*: Assignment of orders to workers

1: $\mathcal{P} \leftarrow$ Set of N initial solutions using Algorithm 2.
2: $X^* = \arg\max_{\forall X \in \mathcal{P}} \Delta f(X)$
3: $iter \leftarrow 0$
4: **while** $(iter \leq MAX - IT)$ **do**
5:   **for** each $X \in \mathcal{P}$ **do**
6:     Calculate λ_X using Eq. (14), $\forall X \in \mathcal{P}$
7:     $W \leftarrow rand\_int(1, \lambda_X)$
8:     $X' \leftarrow X$
9:     **for** $w = 1$ to W **do**
10:       $j \leftarrow randInt(0, |J|), k \leftarrow randInt(0, |K|)$
11:       $X'_{j,k} \leftarrow 1 - X_{j,k}$
12:     **end for**
13:     **if** $\Delta f(X') > \Delta f(X)$ **then**
14:       $\mathcal{P} \leftarrow (\mathcal{P} \setminus X) \cup X'$
15:       **if** $\Delta f(X') > \Delta f(X^*)$ **then**
16:         $X^* \leftarrow X'$
17:       **end if**
18:       $n_b \leftarrow randInt(1, N_b^{max})$
19:       **for** $i = 1$ to n_b **do**
20:         $X^n \leftarrow X^*$
21:         $j \leftarrow randInt(0, |J|), k \leftarrow randInt(0, |K|)$
22:         $X^n_{j,k} \leftarrow 1 - X^n_{j,k}$
23:         **if** $\Delta f(X^n) > \Delta f(X^*)$ **then**
24:           $X^* \leftarrow X^n$
25:         **end if**
26:       **end for**
27:     **end if**
28:   **end for**
29: **end while**
30: **return** X^*;

Algorithm 3 summarizes the steps of WWO-based order assignment algorithm. We start with a population, $\mathcal{P}$ of initial solutions using Algorithm 2 in line 1. Line 2 selects the fittest solution $X \in \mathcal{P}$ as the best solution X^*. For each solution, X we calculate its wavelength, λ_X in line 6. In line 7, we randomly determine the number of propagation steps, w and each X is then propagated for W steps to yield a new solution X' in lines 8-12. Note that, in each propagation step w, we randomly select the dimensions (j and k) of X' and reverse its value, $X'_{j,k}$ (lines 10-11). If X' achieves better fitness value than X, X' replaces X in the population in line 14. In line 16, X^* is replaced by X' if the fitness value of X' is higher than X^*. Then we randomly choose a neighborhood size n_b in line 18 and perform n_b steps neighbor search to yield a neighbor solution X^n of X^* in lines 20 to 22. Note that, neighbor search is similar to propagation operation where in each search step i, we selects random dimensions of X^n and its value is reversed (line 22). The best solution X^* is updated if the neighbor solution X^n attains higher fitness value. Algorithm 3 iterates for $MAX - IT$ times and the best solution X^* is returned. Each pair, (j, k) denotes an assignment of order-worker if $X^*_{j,k} = 1$.

C. ADAPTIVE PAYMENT POLICY FOR WORKER

To incentivize a worker to complete the delivery of his assigned orders faster, WWOFooD devises an adaptive payment policy where a worker, k gets an additional payment by completing an order, j prior to its estimated delivery time, $\widetilde{s}_{j,k}$. So, the workers can earn more profit by reducing their service time, $s_{j,k}$. This policy facilitates the workers to get engaged in more orders which also increases their profit. On the other hand, reduced delivery time results in higher customer satisfaction which enhances platform's service reputation. However, the additional payment calculation should be in such a way that ensures the profit of both the worker and the platform while reflecting the customer's satisfaction. To attain this WWOFooD realize a metric, customer satisfaction index, $\theta_{j,k}$ for an order j and worker k, that is measured by how fast an order is delivered compare to estimated delivery time $\widetilde{s}_{j,k}$ and is calculated as follows,

$$\theta_{j,k} = \begin{cases} min[(0.5+\sigma - \frac{1}{1+e^{-\nu_1 \times (s_{j,k}-\widetilde{s}_{j,k})}}), 1], & s_{j,k} \leq \widetilde{s}_{j,k} \\ \sigma \times e^{-\nu_2 \times (s_{j,k}-\widetilde{s}_{j,k})}, & \widetilde{s}_{j,k} < s_{j,k} \leq s_{max} \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

Here, $\theta_{j,k} \in [0, 1]$. Eq. (15) yields a satisfactory level of $\sigma \in [0, 1]$, If the worker delivers the order at the estimated delivery time $\widetilde{s}_{j,k}$. In real-life scenario, customers expect to receive their orders in minimal delivery time. Thus delivery of an order at the estimated time, $\widetilde{s}_{j,k}$ results a accomplished level of satisfaction for the customers. A customer should experience a raised level of satisfaction if the order is delivered earlier than estimated time. However, the satisfactory level falls drastically if the delivery time exceeds the estimated time and drops to zero as it goes beyond a maximum tolerance level, s_{max}. Eq. (15) models the aforementioned behavior of the customer satisfaction. The first case in Eq. (15) causes the value of $\theta_{j,k}$ to increase smoothly from σ to 1 if the order is delivered earlier. Note that, the sigmoid function in case 1 attains the value of 0.5 as $s_{j,k}$ equals to $\widetilde{s}_{j,k}$, thus results in a satisfaction level of σ. Case 1, increases the satisfactory level smoothly, but limits it to 1. On the other hand, as the order delivery time, $s_{j,k}$ approaches to s_{max}, the exponential function in case 2 causes the satisfaction level to decrease sharply. Case 3 ensures that, satisfactory level drops to 0 when $s_{j,k}$ reaches beyond, s_{max}. Here, ν_1 and ν_2 are two scaling factors those control the shape of sigmoid and exponential functions, respectively. The WWOFooD system utilizes the satisfaction level of the customers to provide additional payment to the workers as follows,

$$\tau_{j,k} = min[(m^p_j - m_{j,k}) \times \eta, (m^p_j - m_{j,k}) \times \theta_{j,k}] \tag{16}$$

Here, m^p_j is the platform's income from an order, j which is determined by the agreement between platform and food providers. Eq. 16 limits the additional payment of a worker by platform's earning times a fixed factor, η, where $\eta \in [0, 1]$. Finally, the profit of a worker can be calculated as,

$$p_{j,k} = \begin{cases} m_{j,k} + \tau_{j,k} - c_{j,k}, & s_{j,k} < \widetilde{s}_{j,k} \\ m_{j,k} - c_{j,k}, & \text{otherwise} \end{cases} \tag{17}$$

VI. PERFORMANCE EVALUATION

In this section, the performances of the proposed WWO-based order assignment is evaluated and compared with two state-of-the-art works, **OCD** [10] and **FOOD-MATCH** [11]. The simulation and experimental evaluations are carried out in Python on a real world dataset (`https://github.com/grubhub/mdrplib`) [7]. The dataset contains real world historic experiences of food delivery services and are used in recent works on online food delivery system design [11], [12].

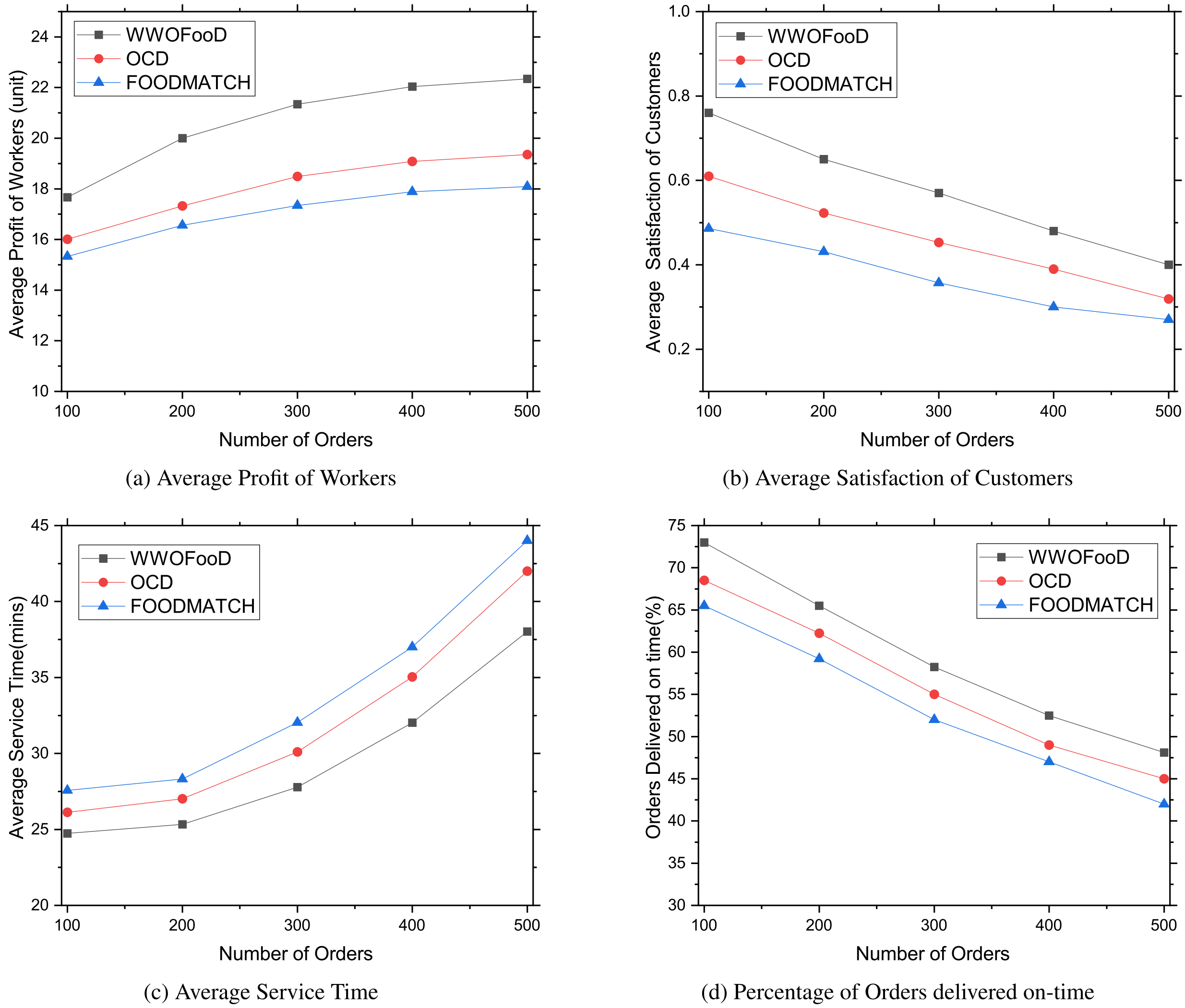
A. SIMULATION SETUP

We consider an area of $5000 \times 5000m^2$ where the location of restaurants, orders and workers are randomly distributed. The orders, workers and restaurants information are extracted from each instance of the dataset. The arrival of orders in the food delivery systems is facilitated using an uniform distribution. The number of restaurants are kept in 50 and the number of orders, workers are varied in 100 $\sim$ 500, and 50 $\sim$ 300, respectively. While measuring the distance between the worker and pickup point, and pickup and delivery point, we consider it to be the euclidean distance. As worker order delivery cost is directly proportional to the distance traveled, per unit distance cost is taken as unit delivery cost. We limit worker's carrying load capacity to 20 kg. The locations of workers, customers and restaurants are represented in $2D$ Cartesian coordinates, (x, y) from a reference point. A summary of the values and ranges of different simulation parameters is provided in Table 4. All simulation experiments were conducted on a PC with an Intel Core i7 Processor and 16 GB memory running Ubuntu 20.04. Each simulation experiment was run for 1000 seconds with different random seed values and the results of 30 simulation runs were averaged to plot each data point in the graphs.

B. PERFORMANCE METRICS

We consider the following metrics to compare the performance of the studied systems.

- *Average profit of workers*: is the accumulated worker profit divided by their total number. The higher value represents better performances.
- *Average satisfaction of customers*: is the total satisfaction enjoyed by all the workers divided by their number. A higher average satisfaction indicates faster



(a) Average Profit of Workers

(b) Average Satisfaction of Customers

(c) Average Service Time

(d) Percentage of Orders delivered on-time

FIGURE 5. Impact of varying number of orders.

TABLE 2. Simulation parameters.

Parameter	Description
Simulation area	$5000 \times 5000\ m^2$
Number of orders	$100 \sim 500$
Number of workers	$50 \sim 300$
Worker's payment per order ($m_{j,k}$)	$20 \sim 25$
Max.carrying capacity of worker (q_k)	20 kg
Min. accepted reputation value (R_{th})	0.25
Delivery deadline, ($t_{j,k}$)	≤ 40 min
Maximum estimated delivery time, (T_{max})	80 min

delivery of the orders thus results in better system performances.

- *Average service time*: is the average duration from reception of an order to its delivery at the customer's location by the workers. The lower value represents better performances.
- *Orders delivered on-time*: is the percentage of orders that is delivered within the estimated delivery time. The higher value indicates better performance.

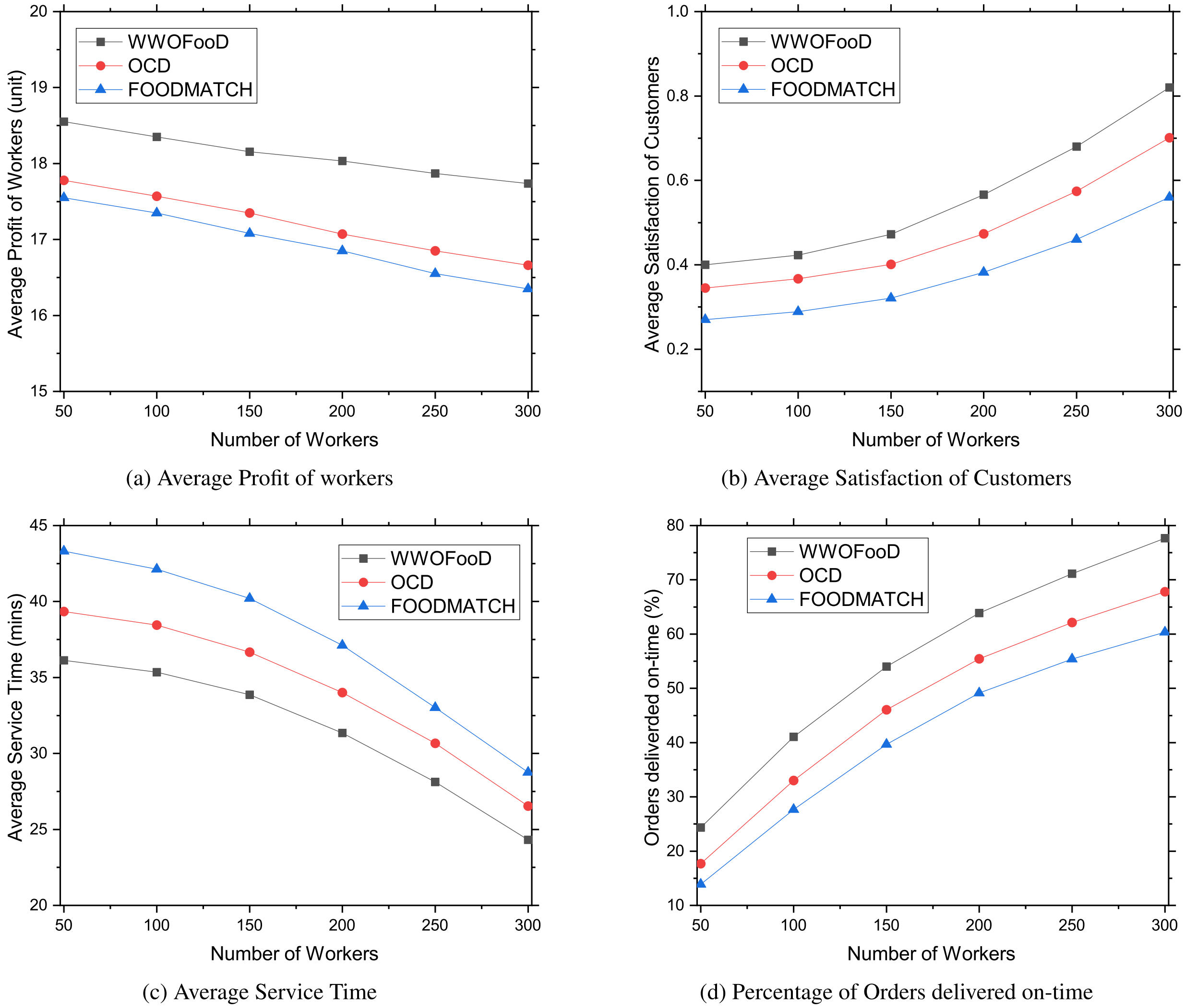
C. SIMULATION RESULTS

In this section, we present the results of our performance evaluation for varying number of orders, workers, and customer's preferred time.

1) IMPACT OF VARYING NUMBER OF ORDERS

Figure 5 depicts the impact of varying number of orders while keeping the number of workers and restaurants fixed at 100 and 50, respectively.

The graphs of this Fig. 5(a) indicates that as the number of orders increases, the average profit of workers also increases for all studied systems as long as it supports the carrying capacity of each of the workers. The reason behind this can be articulated by the fact that, the increased order facilitates the workers to deliver more orders within its travel vicinity which decreases the per unit travelling cost, resulting in increased profit. However, in WWOFooD system workers experience a boost in their profit as its adaptive payment policy disbursed additional profits to the workers. This additional



(a) Average Profit of workers

(b) Average Satisfaction of Customers

(c) Average Service Time

(d) Percentage of Orders delivered on-time

FIGURE 6. **Impact of varying number of workers.**

profit encourages the workers to complete an order with minimal delivery time. On the other hand, the OCD and FOODMATCH fail to provide any additional benefit to the workers as their main design principle is to minimize the delivery cost by reducing the traveling distance of the workers.

The average customer satisfaction for increasing number of orders is shown in Fig. 5(b). The graphs depict that the WWOFooD achieves the higher level of satisfaction comparing to OCD and FOODMATCH. This is because, the WWOFooD system provides the customers with the option to choose their maximum delay tolerance level while placing an order. The WWOFooD's adaptive payment policy intencivizes the workers to deliver orders in minimum possible time with the maximum tolerance time and thus it results in higher satisfaction level. While minimizing service time of the orders, the OCD accounted customer preferred time but their specified sequential approach of delivery did not satisfy customers' preference in real time and the FOODMATCH did not consider any customer preferences, thus resulting poor customer satisfaction.

As depicted in Fig. 5(c), average service time increases with increasing orders for all the studied mechanisms as the workers require to deliver more orders. However, the proposed WWOFooD system requires the least average service time compared to OCD and FOODMATCH. The reason is that as the average order per restaurant increases, it results in an elevating number of co-located orders. This aids the WWOFooD system to assign a worker with co-located multiple orders so as to minimize his travelling time to the pickup points. Similarly, carrying multiple orders to adjacent customer locations by a worker further reduces travel time to the delivery points. The OCD system excludes the waiting time of the workers at restaurants which is a crucial time consideration for food delivery system. Though FOODMATCH considered this waiting time, their main motivation was to optimize the travelling distance which may not always result in optimal service time.

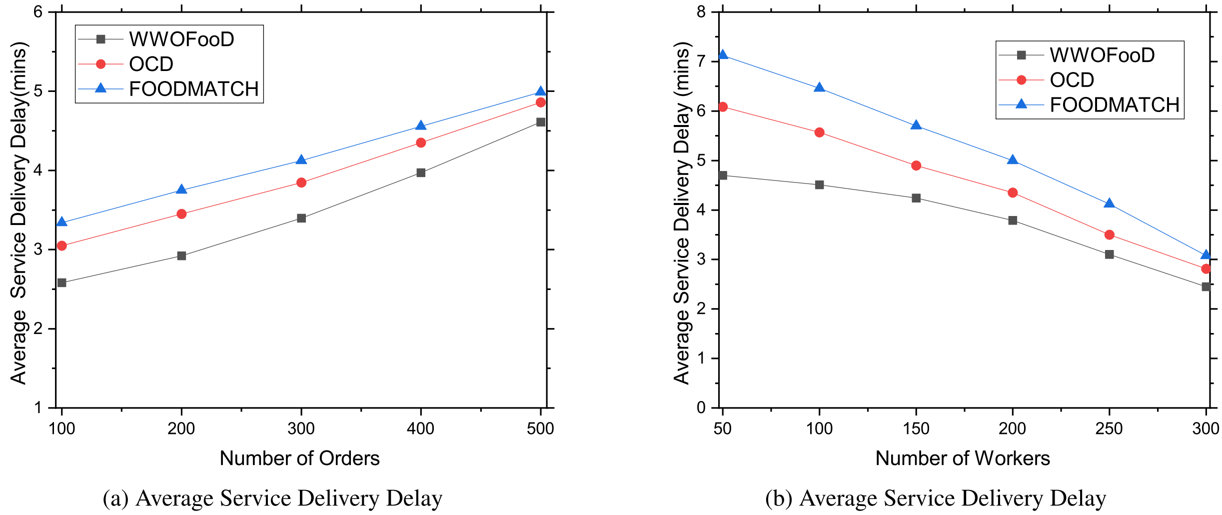


FIGURE 7. Average service delivery delay for varying number of orders and workers.

Fig. 5(d) demonstrated that the increasing number of orders results in degradation of order delivery on time. This is because, workers with increased number of assigned orders cause violation of the delivery deadline. However, in WWOFooD, the distribution of order and delivery locations are considered while assigning multiple orders to the workers. Moreover, the WWOFooD selects workers whose service time is not more than the delivery deadline. As a result, the WWOFooD achieves higher percentage of orders delivered on-time. The OCD and FOODMATCH experienced poor performances as they lack these strategies.

2) IMPACT OF VARYING NUMBER OF WORKERS

In general, increasing the number of workers makes it easier for the food delivery system to choose the suitable workers, resulting in higher worker profit, more customer satisfaction, and reduced service latency. Figure 6 presents the system performance for varying number of workers while keeping the number of orders and restaurants fixed at 500, and 50, respectively.

Fig. 6(a) illustrates that as the number of workers increases, average worker profit decreases and becomes stable after 150 workers. This is because as the number of workers goes high, orders per worker decreases which results in a lower average worker profit. However, WWOFooD system attains higher average profit than the other system due to its adaptive payment strategy that gives workers an opportunity to avail additional profits by delivering orders earlier than the estimated delivery time. OCD and FOODMATCH system focused on minimization of traveling cost, the profit of the workers was being neglected.

Fig. 6(b) shows that, the WWOFooD system achieves higher customer satisfaction than the others with increasing number of workers. This is because it assigns an order to a worker with the shortest possible service time considering the delay tolerance of the customers. Both the OCD and FOODMATCH systems did not take employ any strategy for customer satisfaction correlated with service time minimization, thus result in lower customer satisfaction level.

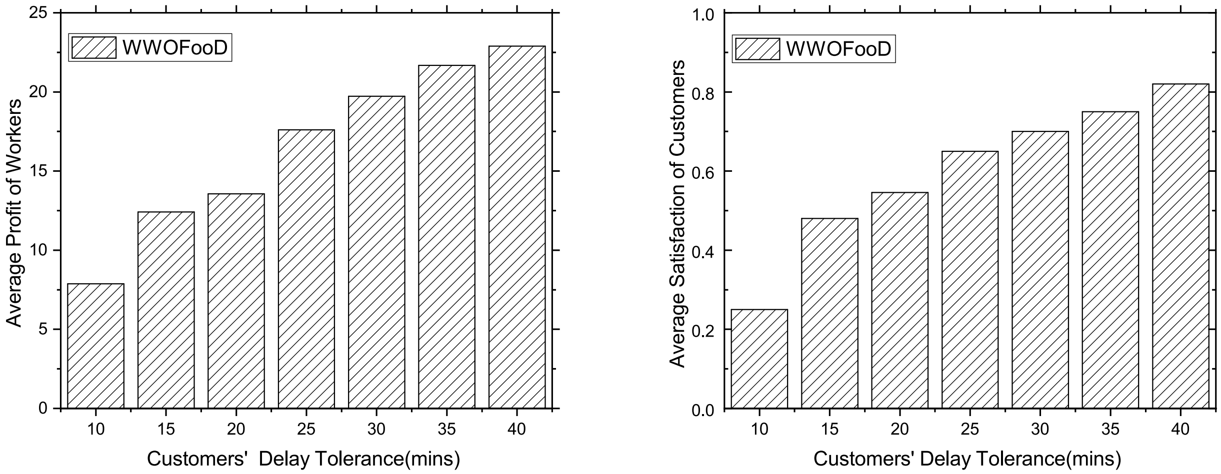
Fig. 6(c) shows that the average service time decreases with respect to the increasing number of workers for all the studied systems. However, WWOFooD experiences lower service time compared to the other system. The reason behind this can be articulated by the fact that WWOFooD assigns multiple orders to a worker not only based on his/her carrying capacity but also considering the location of orders which minimizes the total service time of a worker. On the other hand, both OCD and FOODMATCH did not consider worker's total travelling time to complete an order while assigned multiple orders, thus experience higher average service time.

Fig. 6(d) indicates that the percentage of orders delivered on-time increases with the increasing number of workers. The WWOFooD outperforms other system even for small number of workers as it checks the maximum allowable delay tolerance constraint while assigning multiple orders.

3) AVERAGE SERVICE DELIVERY DELAY

Fig. 7(a) depicts that the average delay increases with the increasing number of orders. From the graphs it is clear that the WWOFooD system achieves significantly less delay compared to OCD and FOODMATCH. This is because while minimizing the estimated service delivery delay, WWOFooD also obeys the maximum delay tolerance constraint. However, none of the OCD and FOODMATCH systems consider the maximum delay deadline thus ends up with increased average service delivery delay.

Fig. 7(b) illustrates that as the number of workers increases, average service delay reduces for all the system. However, overall average service delay consumed by the WWOFooD system is lower than the OCD and FOODMATCH systems. This is because, the WWOFooD considers workers' current location and travel time as well as customers' maximum



(a) Average Profit of Workers

(b) Average Satisfaction of Customers

FIGURE 8. **Impact of customer's delay tolerance on profit and customer's satisfaction.**

tolerance constraint, thus results in lower average service delay than OCD and FOODMATCH. The FOODMATCH system experience highest average service delay compared to WWOFooD and OCD as it does not directly consider worker service time minimization.

4) IMPACT OF CUSTOMER'S DELAY TOLERANCE

Fig. 8(a) depicts the relationship between customers' delay tolerance and workers' average profit. From the graphs, it is clear that, with a lower customer tolerance level, workers receive lower profit from orders. As the tolerance level increases, the percentage of profit is also increased. This is because the possibility of assigning multiple orders within carry capacity is increased with the longer customers' tolerance time. As a result, the average order per restaurant increases resulting in an increased number of co-located orders from the same restaurant which in turn increases the worker profit. Similarly, as shown in Fig. 8(b), customers' satisfaction also increases with the longer tolerance time. Because with the longer delay tolerance time, it becomes convenient to deliver more orders on time, resulting in increased customer satisfaction.

VII. CONCLUSION

In this work, we developed an order assignment framework for food delivery system that enhances worker's profit as well as the satisfaction of customers. Furthermore, taking into account a worker's service time, reputation, and the cost of a worker to execute an order helped our assignment algorithm achieve maximum profits and lowest service time, resulting in increased customer satisfaction. The simulation results indicated that the proposed system can maximize worker's profit by as much as 20% along with up to 40% customer satisfaction while minimizing service time by up to 15%. The results also revealed that increasing the number of orders in the system can improve workers' profit while adversely can affect customer satisfaction, service time and percentage of orders delivered on time. Whereas large number of workers can steadily improve customer satisfaction, service time and percentage of orders delivered on time but can affect worker's profit.

In this work, we have considered a service time of the workers for execution of orders based on worker's traveling time by calculating the distance of the geographical locations. Developing an assignment algorithm for online food delivery tasks that can improve overall system performance by taking into account the road network and projecting time using a learning algorithm could be an intriguing topic to solve in the future.

REFERENCES

[1] J. Phuttharak and S. W. Loke, "A review of mobile crowdsourcing architectures and challenges: Toward crowd-empowered Internet-of-Things," *IEEE Access*, vol. 7, pp. 304–324, 2019.

[2] B. Cohen and J. Kietzmann, "Ride on! Mobility business models for the sharing economy," *Org. Environ.*, vol. 27, no. 3, pp. 279–296, Sep. 2014.

[3] B. Guo, Y. Liu, L. Wang, V. O. K. Li, J. C. K. Lam, and Z. Yu, "Task allocation in spatial crowdsourcing: Current state and future directions," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1749–1764, Jun. 2018.

[4] Z. Lou, W. Jie, and S. Zhang, "Multi-objective optimization for order assignment in food delivery industry with human factor considerations," *Sustainability*, vol. 12, no. 19, p. 7955, Sep. 2020.

[5] *Foodpanda*. Accessed: Mar. 28, 2022. [Online]. Available: https://www.foodpanda.com/

[6] M. Li, J. Zhang, and W. Wang, "Task selection and scheduling for food delivery: A game-theoretic approach," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.

[7] D. Reyes, A. Erera, M. Savelsbergh, S. Sahasrabudhe, and R. O'Neil, "The meal delivery routing problem," *Optim. Online*, p. 6571, Mar. 2018.

[8] M. Abououf, S. Singh, H. Otrok, R. Mizouni, and A. Ouali, "Gale-shapley matching game Selection—A framework for user satisfaction," *IEEE Access*, vol. 7, pp. 3694–3703, 2019.

[9] M. Abououf, R. Mizouni, S. Singh, H. Otrok, and A. Ouali, "Multi-worker multi-task selection framework in mobile crowd sourcing," *J. Netw. Comput. Appl.*, vol. 130, pp. 52–62, Mar. 2019.

[10] W. Tu, T. Zhao, B. Zhou, J. Jiang, J. Xia, and Q. Li, "OCD: Online crowdsourced delivery for on-demand food," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 6842–6854, Aug. 2020.

[11] M. Joshi, A. Singh, S. Ranu, A. Bagchi, P. Karia, and P. Kala, "Batching and matching for food delivery in dynamic road networks," in *Proc. IEEE 37th Int. Conf. Data Eng. (ICDE)*, Apr. 2021, pp. 2099–2104.

[12] B. Yildiz and M. Savelsbergh, "Provably high-quality solutions for the meal delivery routing problem," *Transp. Sci.*, vol. 53, no. 5, pp. 1372–1388, Sep. 2019.

[13] Y. Liu, B. Guo, C. Chen, H. Du, Z. Yu, D. Zhang, and H. Ma, "FooDNet: Toward an optimized food delivery network based on spatial crowdsourcing," *IEEE Trans. Mobile Comput.*, vol. 18, no. 6, pp. 1288–1301, Jun. 2019.

[14] J. Wang, Y. Wang, D. Zhang, F. Wang, H. Xiong, C. Chen, Q. Lv, and Z. Qiu, "Multi-task allocation in mobile crowd sensing with individual task quality assurance," *IEEE Trans. Mobile Comput.*, vol. 17, no. 9, pp. 2101–2113, Sep. 2018.

[15] L. Kazemi and C. Shahabi, "GeoCrowd: Enabling query answering with spatial crowdsourcing," in *Proc. 20th Int. Conf. Adv. Geograph. Inf. Syst. (SIGSPATIAL)*. New York, NY, USA: ACM Press, 2012, p. 189.

[16] S. Sarker, M. A. Razzaque, M. M. Hassan, A. Almogren, G. Fortino, and M. Zhou, "Optimal selection of crowdsourcing workers balancing their utilities and platform profit," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8602–8614, Oct. 2019.

[17] S. Ma, Y. Zheng, and O. Wolfson, "T-share: A large-scale dynamic taxi ridesharing service," in *Proc. IEEE 29th Int. Conf. Data Eng. (ICDE)*, Apr. 2013, pp. 410–421.

[18] A. Bozanta, M. Cevik, C. Kavaklioglu, E. M. Kavuk, A. Tosun, S. B. Sonuc, A. Duranel, and A. Basar, "Courier routing and assignment for food delivery service using reinforcement learning," *Comput. Ind. Eng.*, vol. 164, Feb. 2022, Art. no. 107871.

[19] C. Chen, D. Zhang, X. Ma, B. Guo, L. Wang, Y. Wang, and E. Sha, "Crowddeliver: Planning city-wide package delivery paths leveraging the crowd of taxis," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 6, pp. 1478–1496, Jun. 2017.

[20] G. Xue, Z. Wang, and G. Wang, "Optimization of rider scheduling for a food delivery service in O2O business," *J. Adv. Transp.*, vol. 2021, pp. 1–15, May 2021.

[21] C. Li, M. Mirosa, and P. Bremer, "Review of online food delivery platforms and their impacts on sustainability," *Sustainability*, vol. 12, no. 14, p. 5528, Jul. 2020.

[22] J.-Y. Wu, M.-X. Zhang, X. Wu, and Y.-J. Zheng, "A water wave optimization algorithm for order selection and delivery path optimization for takeaway deliverymen," in *Proc. 11th Int. Conf. Inf. Sci. Technol. (ICIST)*, May 2021, pp. 620–628.

[23] F. Zhao, H. Liu, Y. Zhang, W. Ma, and C. Zhang, "A discrete water wave optimization algorithm for no-wait flow shop scheduling problem," *Expert Syst. Appl.*, vol. 91, pp. 347–363, Jan. 2018.

[24] X.-B. Wu, J. Liao, and Z.-C. Wang, "Water wave optimization for the traveling salesman problem," in *Proc. Int. Conf. Intell. Comput.*, 2015, pp. 137–146.

[25] M.-X. Zhang, J.-Y. Wu, X. Wu, and Y.-J. Zheng, "Hybrid evolutionary optimization for takeaway order selection and delivery path planning utilizing habit data," *Complex Intell. Syst.*, pp. 1–16, Jun. 2021.

[26] Y.-J. Zheng, X.-Q. Lu, Y.-C. Du, Y. Xue, and W.-G. Sheng, "Water wave optimization for combinatorial optimization: Design strategies and applications," *Appl. Soft Comput.*, vol. 83, Oct. 2019, Art. no. 105611.

[27] S. Ji, Y. Zheng, Z. Wang, and T. Li, "Alleviating users' pain of waiting: Effective task grouping for online-to-offline food delivery services," in *Proc. World Wide Web Conf. (WWW)*. New York, NY, USA: ACM Press, 2019, pp. 773–783.

[28] F. Samouh, V. Gluza, S. Djavadian, S. Meshkani, and B. Farooq, "Multimodal autonomous last-mile delivery system design and application," in *Proc. IEEE Int. Smart Cities Conf. (ISC2)*, Sep. 2020, pp. 1–7.

[29] Y. Lu, Y. Wu, and Y. Zhou, "Order assignment and routing for online food delivery: Two meta-heuristic methods," in *Proc. Int. Conf. Intell. Syst., Metaheuristics Swarm Intell. (ISMSI)*. New York, NY, USA: ACM Press, 2017, pp. 125–129.

[30] Y. Yu and Z. Sun, "Delay to group in food delivery system: A prediction approach," in *Proc. 17th Int. Conf. Intell. Comput.* Springer, 2021, pp. 540–551.

[31] A. Bozanta, M. Cevik, C. Kavaklioglu, E. M. Kavuk, A. Tosun, S. B. Sonuc, A. Duranel, and A. Basar, "Courier routing and assignment for food delivery service using reinforcement learning," *Comput. Ind. Eng.*, vol. 164, Feb. 2022, Art. no. 107871.

[32] H. Chu, W. Zhang, P. Bai, and Y. Chen, "Data-driven optimization for last-mile delivery," *Complex Intell. Syst.*, pp. 1–14, Feb. 2021.

[33] S. Saha, M. A. Habib, T. Adhikary, M. A. Razzaque, M. M. Rahman, M. Altaf, and M. M. Hassan, "Quality-of-Experience-Aware incentive mechanism for workers in mobile device cloud," *IEEE Access*, vol. 9, pp. 95162–95179, 2021.

[34] M. T. Arafat, M. H. Emon, S. Sarker, M. A. Razzaque, and M. M. Rahman, "Balancing worker utility and recruitment cost in spatial crowdsensing: A Nash game approach," in *Proc. 8th Int. Conf. Netw., Syst. Secur.*, Dec. 2021, pp. 50–59.

[35] M. Binshtok and C. Boutilier. (2007). Computing optimal subsets. Association for the Advancement of Artificial Intelligence. [Online]. Available: https://www.aaai.org

[36] F. Huq, N. Sultana, S. Sarkar, M. A. Razzaque, and M. H. K. Tushar, "Optimal worker selection for maximizing quality-of-service of online food delivery system," in *Proc. Int. Conf. Sustain. Technol. for Ind. 4.0 (STI)*, Dec. 2019, pp. 1–6.

FARHANA HUQ (Student Member, IEEE) received the B.Sc. degree in computer engineering from American International University, Bangladesh, in 2006, and the M.Sc. degree from the Department of Electronics and Radio Engineering, Kyung Hee University, South Korea, in 2009. She is currently pursuing the Ph.D. degree with the Green Networking Research Laboratory, Department of Computer Science and Engineering, University of Dhaka. She is also a Lecturer (on study leave) with the Department of Computer Science and Engineering, Northern University Bangladesh. She worked in the field of rate adaptation in WLAN. She has published a few papers in reputed journals and conferences. Her current research interests include optimized task assignment in mobile crowd sourcing and NB-IoT for health care.

NAHAR SULTANA (Student Member, IEEE) received the B.Sc. degree in computer engineering from American International University, Bangladesh, in 2005, and the M.Sc. degree from the Department of Computer Science and Engineering, Kyung Hee University, South Korea, in 2008. She is currently pursuing the Ph.D. degree with the Green Networking Research Laboratory, Department of Computer Science and Engineering, University of Dhaka. She is also an Assistant Professor (on leave) with American International University Bangladesh (AIUB). She has published some papers in reputed journals and conferences. Her current research interests include NB-IoT for health care, optimized task assignment in mobile crowd sourcing, 5G networks, and high performance networks.

SUJAN SARKER (Member, IEEE) received the B.S. and M.S. degrees from the Department of Computer Science and Engineering, University of Dhaka, Dhaka, Bangladesh, in 2013 and 2017, respectively. He is currently a Lecturer with the Department of Robotics and Mechatronics Engineering, University of Dhaka, where he is a member of the Green Networking Research Group. His current research interests include sensor networks, participatory sensing, mobile crowdsourcing, human–robot interaction, and swarm robotics. He is a member of the IEEE Computer Society and the IEEE Robotics and Automation Society.

MD. ABDUR RAZZAQUE (Senior Member, IEEE) received the B.S. degree in applied physics and electronics and the M.S. degree in computer science from the University of Dhaka, Bangladesh, in 1997 and 1999, respectively, and the Ph.D. degree in computer engineering from Kyung Hee University, South Korea, in August 2009. He was a Research Professor with the College of Electronics and Information, Kyung Hee University, from 2010 to 2011. He worked as a Visiting Professor at Stratford University, VA, USA, in 2017. He is currently a Professor (on leave) with the Department of Computer Science and Engineering, University of Dhaka. He is also the Director of the Green Networking Research Group, Department of Computer Science and Engineering (http://gnr.cse.univdhaka.edu). He is also working with the Green University of Bangladesh as the Pro Vice-Chancellor and the Dean of the Faculty of Science and Engineering. He has been teaching a good number of courses to graduate and undergraduate students of reputed universities. He was the Principal Investigators of some national and international research projects funded by Government of Bangladesh and Information Society Innovation Fund (ISIF) Asia. He has published more than 140 research papers in international conferences and journals. His research interests include the area of modeling, analysis and optimization of wireless networking protocols and architectures, wireless sensor and body area networks, sensor data clouds, the Internet of Things, and cognitive radio networks. He is an Associate Editor of IEEE Access, an Editorial Board Member of *Journal of Networks and Applications* (JNCA) (Elsevier), *International Journal of Distributed Sensor Networks*, the General Chair of STI 2019, the TPC Chair of ICIET 2019, 2018, a TPC Member of IEEE HPCC, ICOIN, SCALCOM, SKIMA, ICIEV, ADM, NSysS, and ICACCI. He is a member of IEEE Communications Society, IEEE Computer Society, Internet Society (ISOC), Pacific Telecommunications Council (PTC), and KIPS.

MOHAMMAD MEHEDI HASSAN (Senior Member, IEEE) received the Ph.D. degree in computer engineering from Kyung Hee University, Seoul, South Korea, in February 2011. He is currently an Associate Professor with the Information Systems Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. He has authored and coauthored around more than 180 publications, including refereed IEEE/ACM/Springer/Elsevier journals, conference papers, books, and book chapters. His research interests include edge/cloud computing, the Internet of Things, cyber security, deep learning, artificial intelligence, body sensor networks, 5G networks, and social networks. He was a recipient of a number of awards, including the Best Journal Paper Award from the IEEE Systems Journal, in 2018, the Best Paper Award from CloudComp, in 2014, Conference, and the Excellence in Research Award from King Saud University (two times in a row, 2015 and 2016).