Vehicular Communications 31 (2021) 100376

Contents lists available at ScienceDirect

Vehicular Communications

www.elsevier.com/locate/vehcom

Task coalition formation for Mobile CrowdSensing based on workers' routes preferences

Rebeca Estrada [a,*], Rabeb Mizouni [b], Hadi Otrok [b], Azzam Mourad [c]

[a] Information Technology Center & Department of Electrical Engineering and Computer Science, Escuela Superior Politécnica del Litoral, Guayaquil, Ecuador
[b] Center of Cyber Physical Systems, Department of Electrical Engineering and Computer Science, Khalifa University, Abu Dhabi, United Arab Emirates
[c] Department of Computer Science and Mathematics, Lebanese America University, Beirut, Lebanon

A R T I C L E I N F O

Article history:
Received 4 March 2021
Received in revised form 3 May 2021
Accepted 19 May 2021
Available online 26 May 2021

Keywords:
Mobile Crowdsensing
Task coalition
Workers route selection
Particle Swarm Optimization (PSO)

A B S T R A C T

In this paper, we address the problem of task allocation in Mobile Crowdsensing (MCS) by means of forming tasks publisher coalition taking into consideration workers' route preferences. In prior research works, only one of the MCS components (either task publishers, contributors or platform) dominates the task allocation process. Currently, other approaches have investigated tasks coalition based on their geographical locations. In this paper, we address the aforementioned problem by proposing a new scheme taking into accounts the interest of all the participating parties. To this end, our approach provides (1) strategies for selecting the best routes for workers with better long-term earnings and (2) task publishers' coalition formation based on worker's routes selection and preferences regardless of the order of individual execution of tasks. We proposed two models for the coalition formation: i) a centralized approach to solve the problem of the coalition formation together with the route selection, and ii) a simplified heuristic version that first determines disjoint tasks' coalitions based on the preferred routes selected by workers, then, MCS platform sorts the coalitions with best quality of information and selects the best routes for each ordered coalition. Simulation results with real data-set show that the coalition of task publishers together with the distributed route selection per worker does guarantee the quality of information satisfaction of the sensing tasks while enhancing the worker payment.

1. Introduction

Thanks to the large number of mobile users and their inherent mobility, large-scale Mobile Crowdsensing (MCS) systems have emerged as an efficient solution for a wide range of applications by gathering contributions from individuals. In this type of applications, participants use their devices to collectively sense, extract and share information related to some phenomenon of interest. Currently, MCS applications are employed in a variety of fields, ranging from environmental monitoring to social or individual entertainment. Some MCS applications include traffic jam alerts, wireless indoor localization, and small cell network monitoring. Gigwalk[1] is a crowd-sourcing application for checking the on-shelf availability of a product in a convenience store. Mobile device owners are "hired" to collect the data regarding the availability of a certain product in a store at a given precise time and location [1]. By adopting crowd-sourcing, companies are reducing the inventory cost while maintaining the proper stock levels at different stores. Currently, several well-known brands and retailers are among the customers of Gigwalk.

Mobile Crowdsensing is defined as a new paradigm that empowers ordinary citizens to contribute data sensed or generated from their mobile devices, aggregates and fuses the data in the cloud for crowd intelligence extraction and human-centric service delivery [2]. Fig. 1 shows a typical architecture of mobile crowdsensing systems. Several sensing tasks can arrive from different task publishers and request information at specific time and locations. The platform plays the role of task manager. It has the responsibility of enabling cost-effective large-scale sensing applications by allocating the appropriate set of mobile users to each sensing task while maximizing the sensing revenue. A MCS system relies on the crowd-sourced information, which means that a task may be answered by one or multiple workers depending of the application domain and the requirement of the task itself. Some existing platforms may require a single user to perform a task like in crowdsourcing delivery [3]. While in others, such as

* Corresponding author.
E-mail addresses: restrada@espol.edu.ec (R. Estrada), rabeb.mizouni@ku.ac.ae (R. Mizouni), hadi.otrok@ku.ac.ae (H. Otrok), azzam.mourad@lau.edu.lb (A. Mourad).

[1] http://www.gigwalk.com/.

https://doi.org/10.1016/j.vehcom.2021.100376

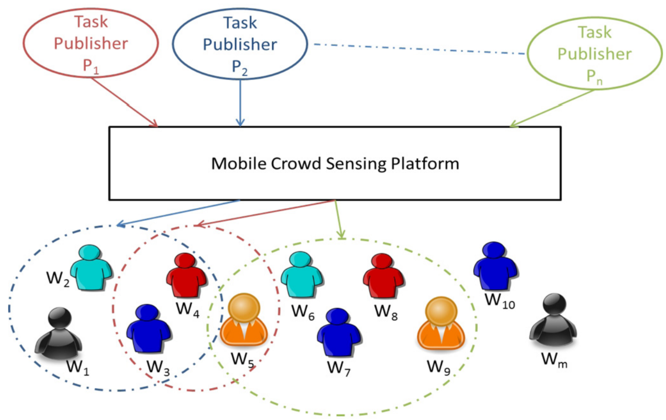


Fig. 1. Mobile Crowdsensing components.

Google Maps, many users are required to answer a task to ensure the reliability of the information.

Task management can be classified, based on the task allocation mechanism as *autonomous task selection* or *coordinated task assignment*. In autonomous task selection mechanism, the workers select their tasks autonomously taking into account different objectives. However, this local selection of the tasks does not lead to a global optimal solution and tends to be inefficient with respect to global utility. Conversely, coordinated task allocation recruits qualified participants and allocates available sensing workers to tasks to meet the goals of various applications such as high task coverage, data quality, or low cost. Centralized approaches are usually efficient but time-consuming. In fact, the selection of workers becomes very complex when the number of tasks increases, especially under hard time constraints. In this paper, we focus on a mixed scheme between autonomous task selection and coordinated task assignment approach.

On the other hand, location and time-sensitive based task allocation faces several challenges such as location dependency, diversity of quality of the sensed data, and budget constraints. The *Quality of Information* (QoI) of the sensed data from mobile users is defined as a value that characterizes how well the sensed data from the worker satisfies the requirement of the task. In order to achieve an adequate level of quality, selecting workers with sufficient QoI is usually a requirement for MCS systems. Many approaches in the literature have tackled these problems by trying to achieve a trade-off between the traveled distances of the workers and the QoI of the sensed data in order to maximize the sensing revenue based on limited budget [4,5]. Others emphasized on the need of motivating users to participate as a key factor for the success of MCS systems.

Few approaches have investigated the coalition formation among the tasks publishers. For instance, the work in [6] proposes an approach that consist of forming coalition of tasks based on tasks' locations using k-mediods algorithm, and assigning a group of workers to each task cluster such as the quality of service (QoS) of the task is maximized while the traveled distance is minimized. Forming coalitions among the task's publishers can have several benefits, among them is the fact that a large budget can be distributed among the workers. This would benefit certain task publishers with reduced budget, whose tasks will obviously not be selected by the workers, which will enable the MCS platform to keep engaged both type of participants (i.e. task publishers and workers). In this paper, we propose a different approach which is based on the idea that active participants are able to select a group tasks that they are willing to perform in a specific order, i.e. a route, that allows them to minimize the traveled distance and to maximize the average payment per task, then, the MCS platform is responsible to form coalitions of tasks based on the preferred selected tasks of the workers maximizing the ratio between the aggregated QoI and the product of the budget and the response time.

In [7], a survey of several incentive mechanisms for participatory sensing is presented. Recently, some mechanisms have been proposed to avoid the misreport of the workers' cost minimizing the total cost while guaranteeing certain quality of experience to tasks publishers such as [8,9]. However, these approaches still suffer from several limitations summarized as follows:

- Centralized approach tends to be time-consuming and very complex to find the optimal solution.
- Task allocation process is dominated by one MCS component (either task initiators or contributor).
- Workers can usually have full control on the allocated tasks to perform. Nevertheless, workers should have the possibility to determine the most effective routes to perform several tasks, a fact that may allow them to earn more in the long term run.
- Coalition of task publishers is based on the location of the sensing tasks which might not guarantee the workers satisfaction regarding their payment.

Unlike our previous work [10], which is a single-task allocation model that enables workers to participate in consecutive tasks allocation as long as they can meet the time constraints, in this paper, we address the problem of multi-task multi-worker allocation by proposing a new pragmatic scheme for task management in MCS systems consisting of a global selection of the task publishers' coalition taking into account workers preferences strategies in terms of routes (i.e. set of tasks to be performed in a consecutive order).

Our motivation is the fact that a worker is more likely to be willing to perform nearby tasks than distant ones. For example, in a crowd sourcing system such as Uber, drivers will prefer to go to pick up customers close to their current locations instead of going to further locations (i.e. more expenses to reach the customer) when the traveled distance to the destination for both customers is the same, which means that the payment will be the same. In addition, it would also be profitable for the driver as well as the customer if the latter is willing to share the ride with other customers (i.e. tasks publishers). Hence, workers will be more engaged to finish their allocated tasks.

Mobile workers should select a set of strategies to perform several tasks as a route that the worker must follow to complete the tasks within the required response time. This means that the workers' strategies are ordered list of routes that they are willing to perform to obtain a higher payment. Dijsktra shortest path algorithm of the graph theory is used to select several routes for a given task. Two different approaches are proposed to form task publishers' coalitions and their corresponding routes selection: i) a centralized approach by means of an exhaustive search of all possible combination of coalitions, and ii) a simplified heuristic version that first determines disjoint tasks' coalitions based on the preferred workers' routes, then, MCS platform sorts the coalitions with best quality of information and selects the best routes for each ordered coalition. For each coalition, a parameter-less scheme of Particle Swarm Optimization (PSO) algorithm is used to select the routes that maximize the aggregated QoI while minimizing the budget and response time.

The main contributions of our simplified model are:

- Maximizing workers' payment by means of allocating several tasks to be performed consecutively taking into account the worker's preferences without increasing the required budget per task.

- Minimizing the budget used per task having sufficient aggregated quality of information while satisfying their time constraints.
- Guaranteeing the workers' payment by reducing the high running times required by centralized approach.

Simulation with real dataset shows that the simplified version of proposed approach outperforms the benchmark modes, namely the consecutive Tasks allocation using Particle Swarm Optimization [10] and the coalition formation based on tasks' location [6], in terms of workers' payment. On the other hand, the proposed solution offers comparable results as our centralized task coalition formation model without increasing the required budget and response time for the sensing tasks.

The remaining of the paper is organized as follows: Section 2 presents an overview of the relevant related work. Section 3 formulates our problem and discuss the challenges of such a centralized formulation by means of an example. Section 3 presents the semi-distributed framework and its components which are based on the Short-Path algorithm, PSO-based task allocation algorithm. Section 4 describes the benchmark model used for comparison purposes and the most commonly used performance metrics for MCS systems. Section 5 presents the simulation scenarios and the numerical results obtained for the proposed model contrasted with the benchmark model. Finally, Section 6 concludes the paper.

2. Related work

Crowdsensing systems can be classified according to the phenomenon measured and the user involvement in the sensing process [11]. Regarding the first criterion, MCS systems can be environmental such as the approach in [12,13], infrastructure (e.g. RoadCrowd [14,15]), and social such as the recommendation to places visited by an individual [16,17]. According to the second criterion, MCS can be either participatory or opportunistic. In participatory sensing, the users participate to send sensed data to a server [7]. In opportunistic sensing, the sensed information is sent automatically with minimal user involvement [18]. In both cases, proper incentives should be offered to the users to encourage their participation in the system (e.g. entertainment, service or financial rewards). In [19], a four-layered architecture for MCS systems is presented, which consists of application, data, communication, and sensing layer.

An analysis presented in [20] reveals some specific features in MCS in comparison with crowdsourcing. The unique characteristic lies in the aspects of mobility and sensing. Moreover, a classification of task allocation algorithm is proposed, which depends on: 1) the number of parameters to optimize: single-objective [21] or multi-objective [22,23], 2) the number of allocated tasks: single-task-oriented [24] or multitask-oriented [25], and 3) how the processing takes place: off-line [26] or online [27].

Prior research works aim at optimizing the process of data sensing by efficient assessment of the available resources (i.e. workers with smart devices) to meet the task requirements. Owing to the fact that several factors can be taken into consideration for the task allocation, the optimization process is hard to achieve. Some approaches aim at optimizing (i.e. maximizing or minimizing) only one of those factors such as sensing costs [28], coverage of area of interest [29,30], quality or credibility of sensed data [4,5], the number of completed tasks [31], and the revenue (i.e. difference between utility and cost) [32] under several constraints such as sensing duration, task capacity, budget and time. Nevertheless, the trade-off between the most commonly used factors needs to be investigated in order to meet the requirements of the sensing task.

In [33] a reverse auction is employed to design an incentive mechanism for mobile users aiming to maximize the participants' expected profits. Luo et al. [34] introduced an all-pay auction to design an incentive mechanism so as to maximize the profit of platform. In [35], the authors adopted the auction approach to design an incentive mechanism for location-aware collaborative sensing. A double auction mechanism is investigated in [36] to study the optimal assignment of mobile users and tasks (with data reuse) systematically, under both information symmetry and asymmetry, taking into account if the user cost and the task valuation are public information. In [8], an incentive mechanism for group recruitment is proposed to solve the problem of workers' greediness, where workers overprice their data to improve their profit. This approach consists of a selection and a payment mechanism taking into account both his contribution to the overall group QoI and the cost of his contribution compared with other workers. The main goal is to avoid the selection of greedy members since their costs will be high compared to the QoI they offer.

Few approaches have investigated the coalition formation between any type of participants. For instance, [37] presents a Bayesian co-clustering for truth discovery approach using a small portion of ground truth data to aggregate user-contributed observations. The groups have fixed size for both task publishers and workers based on a reliability matrix that it is learned from users' correctness on the tasks. A group-based multi-task worker selection that allocates multiple tasks for a group of workers aiming at maximizing the tasks' QoS while minimizing their completion time is presented in [6]. The approach consists of a location based clustering of tasks using k-mediods algorithm, and a genetic algorithm to select a group of workers to a tasks' cluster. However, it is a centralized approach for each task cluster, therefore, it will be time-consuming and might fail to deliver the sensed data for real-time sensing tasks. In [38], a truthful incentive mechanism is proposed to minimize the social cost such that cooperative tasks can be completed by a group of compatible users using real-life relationships from their social networks. Unlike these prior formation coalition approaches, we propose to form coalition of task publishers based on the workers' routes preferences.

3. System model

We consider a Mobile Crowdsensing system consisting of a set of mobile users (workers) $W = \{w_1, w_2, ..., w_M\}$, and a set of sensing tasks $T = \{T_1, T_2, ..., T_N\}$, where M is the number of mobile users and N is the number of sensing tasks or task publishers. In MCS system, MCS platform publishes multiple sensing tasks. In this paper, it is assumed that each task publisher has only one task to be performed and that each mobile user has a smart device equipped with a set of sensors to carry out at least one task and uploads the required for the selected tasks to the platform. Each sensing task T_i is associated with a given budget B^i_{max} representing the monetary incentive to encourage the participation of mobile users.

Our solution proposes to solve this problem in three basic stages as follows:

1) **Preferred routes selection per worker:** Once the sensing tasks are published by the MCS platform, each worker selects a number of preferred routes such as they provide the best payment per task for the worker while minimizing the traveled distance. We define a route as a set of tasks that are going to be carried out in a specific order.
2) **Task Publishers' Coalition:** Once the preferred routes from all workers are received by the MCS platform, MCS platform is responsible of determining the configuration of coalitions for task publishers that maximizes the ratio between the aggre-

Table 1
Model parameters.

Name	Description
Coalition and Route Parameters	
C	number of coalitions
c	coalition identification
S_h	h-th possible configuration of tasks coalition
T_r	set of tasks in a given route r
R	set of preferred routes for all workers
Task Parameters	
T	Set of tasks
B^i_{max}	Maximum budget per task i
q^i	Minimum QoI for task i
l^i	Location of the task i
N	number of tasks
N^i_{max}	Maximum number of workers per task i
t^i	Maximum Response Time required by Task i
Worker Parameters	
W	Set of workers
M	number of workers
ρ_j	reputation of the worker j
R_j	set of routes for worker j
ϕ_j	confidence of the worker j to perform a task
N^{max}_j	Maximum number of consecutive tasks per worker
$q^{i,r}_j$	QoI provided by the worker j to the task i in the route r
l_j	Location of worker j
p_j	Payment that the worker j is willing to receive per traveled km
d^i_j	Distance between task i to worker j
$d^{i,r}_j$	Traveled distance for worker j to reach task i in route r
r^k_j	k-th preferred ordered route for worker j
$t^{i,r}_j$	Time that takes worker j to reach and perform task i in route r
Output Variables	
γ^c_i	Binary variable that associates task i with the coalition c
$\beta^{r,i}_j$	Binary variable that indicates if route r is selected for worker j to perform task i
$\beta^{r,i}_{c,j}$	Binary variable equal to $\gamma^c_i \beta^{r,i}_j$

gated quality of information and the budget taking into account the workers' routes preferences, the budget and the tasks' time constraints. Each coalition represents a cluster of tasks. For convenience, we will use the terms 'cluster' and 'coalition' interchangeably.

3) **Worker Selection for the possible coalition configuration:** The proposed solution solves the coalition combination selection and workers' route association using a meta-heuristic population algorithm such as Particle Swarm optimization. PSO has been proven to obtain a satisfying optimal solution and to speed up the optimization process.

For the sake of clarity, Table 1 summarizes the notation used in this paper.

3.1. Preferred routes selection per worker

This is the first stage of the proposed model. Each worker determines several routes, R_j to perform a number of consecutive tasks, T_r, together with the expected total payment, traveled distance and average payment per task. A route is a set of tasks to be performed in a specific sequential order. Then, the worker selects a fixed number of preferred routes and sends them to the MCS platform. These selected routes should maximize the average payment per task that the worker is willing to receive. For each worker, j, the objective function is given as follows

$$\max_{r \in R_j} \frac{p_j \sum_{i \in T_r} d^{i,r}_j}{|T_r|} \tag{1}$$

where T_r is the set of tasks that are included in route r and $|T_r|$ denotes its cardinality, i.e. number of tasks in the route r. This objective function is subject to the following constraint:

$$\beta^{i,r}_j t^{i,r}_j \leq t^i; \quad i \in N \tag{2}$$

where $t^{i,r}_j$ is the time that takes worker j to reach task i, and it estimated the total traveled distance (i.e. $\sum_{h=1}^{i} d^{h,r}_j$) to reach the task i in route r divided by the speed of the worker j. This is owing to the fact that the worker j will be performing some other tasks before task i, which is determined by the route r. It is worth noticing that if task i is the first task in the route the traveled distance, then, $d^{i,r}_j$ is equal to the difference between the worker location and task location, otherwise, the traveled distance $d^{i,r}_j$ is equal to difference between the last visited task and the task i.

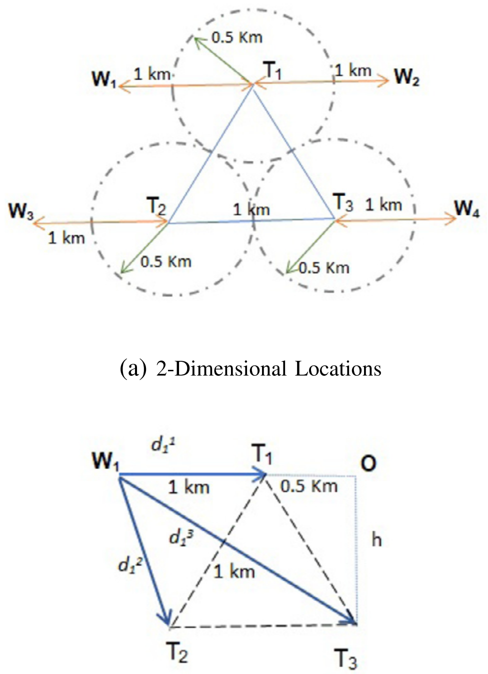
Thus, k preferred routes are selected from each worker using this objective function, which means the first k routes that maximizes the objective function (i.e. $r^1_j \succeq r^2_j \succeq r^3_j \succeq .. \succeq r^k_j$). To do so, we use the concept of the shortest path problem of the graph theory to solve this problem, the algorithm finds a path between two vertices (or nodes) in a graph such that the sum of the weights of its constituent edges is minimized. For the routes selection per worker, the source of the graph is the location of the worker location and the destination node is the task location. We consider that the worker initially selects a set of tasks in the vicinity. In this stage, each worker finds the routes using shortest path algorithm [39] with a length of the route higher than 1. To do this, the algorithm first determines the weight matrices. The weights are either the distance between the worker and the task location or the distance between two tasks locations. This is owing to the fact that the workers can perform several consecutive tasks in order to have more earning in the long term. Algorithm 1 presents the use of Dijsktra algorithm to determine the k-shortest path (i.e. route) to reach a task from a worker initial location.

Algorithm 1: Selection of k preferred routes.

Data: Worker Location l_j,
Tasks Location l^i,
Destination Tasks location
Result: k shortest path routes for worker j
begin
Generate the matrix weight for the worker j, Ω_j, i.e. the distances between worker and tasks;
$[r^k_j, d^{i,r}_j]$ = Dijkstra(Ω_j, source, destination);
Sort routes in descending order according to the average payment $\frac{\sum_{i \in T_r} p_j d^{i,r}_j}{|T_r|}$;
Return the k first elements from r^k_j;
end

In addition to the preferred k routes obtained by this model, single-task routes are added to the list of preferred routes to avoid depriving the completion of certain tasks that are not suitable to be performed in sequence.

More concretely, let's consider a simple MCS system with four available workers (W_1, W_2, W_3, W_4) and three location-based tasks (T_1, T_2, T_3) in a two dimensional (2D) area as illustrated in Fig. 2a. This simplified example is only for illustrative purpose; real



(a) 2-Dimensional Locations

(b) Distances from W_1 to the Tasks

Fig. 2. Scenario with three tasks and four mobile users.

Table 2
k shortest path for worker W_1 to reach task T_1.

Path	Length	Visited tasks	Traveled distance
1	1	T_1	1
2	2	$T_2 - T_1$	2
3	2	$T_3 - T_1$	2.73
4	3	$T_2 - T_3 - T_1$	3
5	3	$T_3 - T_2 - T_1$	3.73

scenarios are more complex and highly scalable involving considerable number of tasks and workers. In the figure, it is shown a dashed gray circle with a radius equal to 0.5 Km around each task. Thus, it can be observed that the tasks are one 1 kilometer away from each other. Workers W_1 and W_2 are located at 1 km from task T_1 while W_3 is located at 1 km from T_2 and W_4 is located at 1 km from T_3. We use Fig. 2b to estimate the distances d_1^2 and d_1^3 using simple trigonometry. For instance, it is clear that the distance d_1^1 is equal to 1 Km and distance between tasks T_1 and T_2 is also 1 km. Owing to the fact that the triangle $\widehat{W_1T_1T_2}$ has two sides that are equal and the inner angles are equal, d_1^2 should be equal to 1 km. To estimate the distance d_1^3, we need first to estimate the height h of the triangle $\widehat{T_1T_2T_3}$, i.e. $h = \sqrt{1-0.5^2}$. Finally, d_1^3 will be equal to the hypotenuse of the triangle $\widehat{W_1OT_3}$, this means $d_1^3 = \sqrt{1.5^2 + 0.5^2} = 1.73$ Km.

Once all distances (i.e. d_j^i, $i = 1, 2, 3$, $j = 1$) are estimated, Algorithm 1 returns the k preferred path between W_1 and task T_1 as shown in Table 2 for the illustrative example given in Fig. 2b.

From Table 2, it can be observed that there are only five paths to reach the task T_1 for worker W_1. For a realistic scenario, it is possible to have more than 5 paths if there are more nearby tasks. Each worker provides several possible routes to perform different tasks that comply with constraint (2) to the MCS platform. The maximum number k of routes per worker depends on the number of sensing tasks, the proximity of the workers to the tasks, and the required time of the tasks to be performed. If k is higher than one, then, the MCS platform have the direct route to a specific task

Table 3
Set of routes for worker W_1.

Route	Payment	Distance	Payment per task	Length
$T_3 - T_2$	4.10	2.7321	2.05	2
$T_3 - T_1$	4.10	2.7321	2.05	2
$T_1 - T_2$	3	2	1.5	2
$T_2 - T_1$	3	2	1.5	2
$T_2 - T_3$	3	2	1.5	2
$T_1 - T_3$	3	2	1.5	2

Table 4
Routes that worker W_1 sends to MCS platform.

Route	Payment	Distance	Payment per task	Length
$T_3 - T_2$	4.10	2.7321	2.05	2
$T_3 - T_1$	4.10	2.7321	2.05	2
T_3	2.60	1.7321	2.60	1
T_2	1.5	1	1.5	1
T_1	1.5	1	1.5	1

Table 5
Received routes at MCS platform.

Route	Worker	Tasks	Payment	Distance	Payment per task	Length
1	w_1	$T_3 - T_2$	4.10	2.73	2.05	2
2	w_1	$T_3 - T_1$	4.10	2.73	2.05	2
3	w_1	T_3	2.60	1.73	2.60	1
4	w_1	T_2	1.50	1.00	1.50	1
5	w_1	T_1	1.50	1.00	1.50	1
6	w_2	$T_2 - T_1$	7.51	2.73	3.76	2
7	w_2	$T_2 - T_3$	7.51	2.73	3.76	2
8	w_2	T_2	4.76	1.73	4.76	1
9	w_2	T_1	2.75	1.00	2.75	1
10	w_2	T_3	2.75	1.00	2.75	1
11	w_3	$T_3 - T_2$	3.00	3.00	2.00	2
12	w_3	$T_3 - T_1$	3.00	3.00	2.00	2
13	w_3	T_3	2.00	2.00	1.00	1
14	w_3	T_1	1.73	1.73	1.00	1
15	w_3	T_2	1.00	1.00	1.00	1
16	w_4	$T_2 - T_1$	7.50	3.00	3.76	2
17	w_4	$T_2 - T_3$	7.50	3.00	3.76	2
18	w_4	T_2	5.00	2.00	4.76	1
19	w_4	T_1	4.33	1.73	2.75	1
20	w_4	T_3	2.50	1.00	2.75	1

and other options to reach the task passing by other tasks. In this example, we use a low value of k. It should be noticed that if k is too big, then, the mobile device will require more time to compute the routes selection and to send k routes to the platform.

Table 3 presents the preferred k routes for worker W_1 to perform each task. Then, worker W_1 selects a given number of routes to send back to the MCS platform but it will be also required to provide the single-task route as it is shown in Table 4. For this example, let's consider that each worker selects 2 best routes plus the three individual routes for T_1, T_2 and T_3, which are highlighted in the Table 3.

As the worker W_1, the same procedure is carried out by the other workers. The other three workers also select 5 routes each and Table 5 presents the selected routes by each worker. At this stage, the MCS platform is responsible for determining the potential coalitions of tasks and their corresponding routes that maximize the ratio between the total aggregated quality of information and the budget.

3.2. Coalition of task publishers

This phase attempts to find coalitions between the task publishers in order to maximize their quality of information and the number of allocated tasks. The main goal is to maximize the ratio between the aggregated quality of information and the product of budget and the maximum response time taking into account the workers' routes preferences subject to the budget and the tasks' time constraints. Each worker has provided an ordered list of routes to perform several sets of consecutive tasks, $\beta_j^{i,r}$. Thus, our objective is to maximize the following:

$$\max_{\gamma,\beta} \frac{\sum_{i\in T}\left(\sum_{c\in C}\sum_{j\in W}\sum_{r\in R}\beta_{c,j}^{i,r} q_j^{i,r} - q^i\right)}{\left(\sum_{c\in C}\sum_{i\in T}\sum_{j\in W}\sum_{r\in R}\beta_{c,j}^{i,r} p_j d_j^{i,r}\right)\times \max_{j,i,c,r}(\beta_{c,j}^{r,i} t_j^i)} \tag{3}$$

where $q_j^{i,r}$ reflects the QoI with which the worker is likely to perform the task i in a route r. Similarly to our prior work in [10], we consider the quality of information of a worker in a route r $c_j^{i,r}$ reflect the accuracy and timeliness of the collected data and is given by

$$q_j^{i,r} = \rho_j \times \phi_j^h \times \delta_j^i \tag{4}$$

ρ_j is the reputation of worker j, h represents the position of the task i in the preferred route r of worker j, ϕ_j^h is the confidence of the worker to perform the task in the route h, and δ_j^i is given as

$$\delta_j^i = 1 - max(0, min(log_{t^i}(t_j^{i,r}), 1)) \tag{5}$$

$t_j^{i,r}$ is the time that takes worker j to reach the task i in the route r and t^i is the maximum response time for task i. Our main focus is to evaluate the task publisher's coalition based on the routes selection given by the workforce. It should be noticed that workers with low reputation are more likely to behave maliciously. In order to avoid this behavior, the current proposal can be modified to consider the preferred routes from workers with reputation higher than a given threshold when forming coalition of tasks.

Our objective function (3) is subject to the following constraints:

- Selection of one route per worker to perform a task

$$\sum_{c\in C}\sum_{r\in R}\beta_{c,j}^{i,r} \le 1; \quad (j,i) i \in T, j \in W \tag{6}$$

- Maximum number of workers per task

$$\sum_{c\in C}\sum_{j\in W}\sum_{r\in R}\beta_{c,j}^{i,r} \le N_{max}^i; \quad i \in T \tag{7}$$

- Budget Constraint

$$\sum_{c\in C}\sum_{j\in W}\sum_{r\in R}\beta_{c,j}^{i,r} p_j d_j^{i,r} \le B^i; \quad i \in T \tag{8}$$

- Quality of information satisfaction

$$\sum_{c\in C}\sum_{r\in R}\sum_{j\in W}\beta_{c,j}^{i,r} q_j^{i,r} \ge q^i; \quad i \in T \tag{9}$$

- Time constraint

$$\max_{i,j} \beta_{c,j}^{i,r} t_j^i \le t^i; \quad i \in T \tag{10}$$

- One task belongs to only one coalition

$$\sum_{c\in C}\gamma_i^c \le 1; \quad i \in T \tag{11}$$

It should be noticed that $\beta_{c,j}^{i,r}$ is equal to $\gamma_i^c \beta_j^{i,r}$, we used this notation to reduce the length of the equations.

As mentioned previously, each worker selects the routes that minimize the traveled distance while maximizing the total payment, the average payment per task and the number of tasks in the preferred route. Table 3 presents the routes that meet the requirements for tasks ordered by the worker preferences. Then, the MCS Platform determines the coalitions that can be formed based on the workers' preferred routes using the Algorithm 2.

Algorithm 2: Task coalition formation algorithm.

Data: Worker Routes R_j
Result: Tasks Coalition Set C,
  Association tasks and coalition binary variable γ_i^k
begin
  for *each route* $r \in R_j$ **do**
    Define the set of Tasks T_r for route r;
    if $T_r \notin C$ **then**
      Add the new coalition $C_k = T_r$;
      $C \leftarrow C \cup C_k$;
    else
      Find the Coalition $C_k \in C$ that is equal to set T_r;
    end
    Add r to the set of routes for Coalition C_k;
  end
  Return γ_i^c, C;
end

In a centralized approach, the MCS platform should find all possible coalition combinations among the set of coalitions. To do this, an exhaustive search Algorithm 3 determines all the coalition combinations and finds the one that maximizes the ratio among the aggregated quality of information and the product of budget and maximum response time. The complexity of an exhaustive search problem increases as the number of tasks increases because the number of possible coalitions configurations is exponentially increased as the number of tasks increases.

Algorithm 3: Exhaustive search algorithm.

Data: Tasks Coalitions Set C
Result: Coalition combination with highest QoI S_h^*
begin
  for *each* $C_k \in C$ **do**
    $h \leftarrow 0$;
    $S_h \leftarrow C_k$;
    $\beta_{h,j}^{r,i} \leftarrow \beta_{k,j}^{r,i}$;
    for *each* $C_l \in C$ **do**
      if $C_l \cap C_k =$ **then**
        Add the coalition C_l to feasible solution ;
        $S_h \leftarrow S_h = S_h \cup C_l$;
      end
      $h \leftarrow h + 1$;
    end
  end
  for *each feasible solution* $S_h \in S$ **do**
    Run Algorithm 4 for S_h;
    Evaluate Eq. (3) for S_h ;
  end
  Return S_h^* with highest evaluation of Eq. (3) and its association variables $\beta_{h,j}^{r,i}$;
end

Since the multi-task multi-worker allocation problem has been proven NP-hard in our prior work [10] or in [40], we propose to solve the formulated problem given in Eqs. (3) - (11) using PSO for each possible tasks' coalition combination. This is owing to the fact that PSO has been proven to obtain a satisfying optimal solution and to speed up the optimization process in comparison to other evolutionary-based algorithms. PSO requires information exchange among the population members to enhance the search process using a combination of deterministic and probabilistic rules.

3.3. Workers selection

Hence, the proposed solution solves the PSO-based coalition combination selection and workers' route association. In the PSO algorithm, one vector (X) is used to represent the location of each particle n in the search space and one vector for the speed of particle movement, (v_x). For our model, we map the both binary variables γ_i^c and $\beta_j^{r,i}$ to the vector (X). Therefore, the dimension of the particle's location vector (X) is equal to $C \times M + R \times M \times N$. We use a parameter-less scheme [41] of PSO. This scheme defines penalties based on the average of the objective function and the level of violation of each constraint during each iteration. Thus, the penalty coefficients p_l are determined by

$$p_l = |\overline{f}(x)| \frac{\overline{g_l}(x)}{\sum_{j=1}^{P} [\overline{g}(x)]^2}, \tag{12}$$

where $\overline{f}(x)$ is the average objective function, $\overline{g}(x)$ is the average level of l_{th} constraint violation over the current population and P is the number of constraints. The constraints (6) - (11) are included in $\sum_{l=1}^{P} k_l \widehat{g}(\mathbf{X})$ to penalize unfeasible solutions.

PSO is defined to solve a minimization problem, therefore, our fitness function is derived from (1) as follows:

$$f'(x) = \begin{cases} G - \sum\limits_{c \in C} \sum\limits_{j \in W} \sum\limits_{i \in T} \gamma_i^c X_j^{i,r} q_j^{i,r}, & \text{for feasible solutions} \\ G - \sum\limits_{c \in C} \sum\limits_{j \in W} \sum\limits_{i \in T} \gamma_i^c X_j^{i,r} q_j^{i,r} + \sum_{l=1}^{C} k_l \widehat{g}(\mathbf{X}), & \text{otherwise} \end{cases}, \tag{13}$$

where G is a large number (e.g. 10000) in order to convert our maximization problem into a minimization problem and $\widehat{g}(x_n^k)$ is determined as follows:

$$\widehat{g}(x_n^k) = \max\left(0, [g_j(x_n^k)]\right). \tag{14}$$

The average of the fitness function for any population is approximately equal to $\overline{f}(x) + |\overline{f}(x)|$. Algorithm 4 presents the PSO-based multi-task allocation algorithm.

Algorithm 4: PSO based joint coalition combination selection and route association algorithm.

Data: MS Worker Routes (r_j),
Maximum Budget, B^j,
Required Time, t_{max}^j,
QoI, q^i, for all the tasks.
Result: coalition of tasks and selected routes $(\gamma_i^c, \beta_j^{r,i})$
begin
  Generate initial swarm with the particle positions and velocities as follows;
  $\mathbf{X_j} \leftarrow \mathbf{randi(0, 1)}$;
  $\mathbf{v_X} \leftarrow \mathbf{randi(0, 1)}$;
  Evaluate Fitness Function (13);
  Determine first global best of the swarm;
  while $k \leq MaxIteration$ **do**
    Update Position;
    Evaluate Fitness Function;
    Determine best local for each particle;
    Determine best global in the swarm and update the best global;
    Update velocity;
  end
end

Table 6
Potential coalitions for the example in Fig. 2.

Coalition	Route	Worker	Tasks	Budget	Time	Min QoI
1	6	w_2	T_1, T_2	7.51	3.42	0.855
1	16	w_4	T_1, T_2	7.50	4	0.81
2	2	w_1	T_1, T_3	4.10	5.46	0.63
2	12	w_3	T_1, T_3	3.00	5.71	0.675
3	7	w_2	T_2, T_3	7.51	3.42	0.855
3	17	w_4	T_2, T_3	7.50	4	0.81
3	1	w_1	T_2, T_3	4.10	5.46	0.63
3	11	w_3	T_2, T_3	3.00	8.57	0.675
4	9	w_2	T_1	2.75	1.25	0.95
4	5	w_1	T_1	1.50	2	0.7
4	19	w_4	T_1	4.33	2.31	0.9
4	14	w_3	T_1	1.73	4.95	0.75
5	4	w_1	T_2	1.50	2	0.7
5	8	w_2	T_2	4.76	2.17	0.95
5	18	w_4	T_2	5.00	2.67	0.9
5	15	w_3	T_2	1.00	2.86	0.75
6	10	w_2	T_3	2.75	1.25	0.95
6	20	w_4	T_3	2.50	1.33	0.9
6	13	w_3	T_3	2.00	2.85	0.75
6	3	w_1	T_3	2.60	3.46	0.7

3.4. Simplified task coalition based on preferred route selection of workers

We modified the centralized approach to a simplified version that first determines disjoint coalitions of tasks that can be performed from the set of preferred routes given by the workers. In this approach, workers choose first the routes to perform several consecutive tasks and send this information to the MCS platform. MCS platform finds first all the possible tasks' coalitions that can be formed based on workers' route preferences using Algorithm 2. Each coalition contains the routes with a subset of tasks regardless the order of how these tasks will be carried out by the workers. For each coalition, the MCS platform determines the optimal solution for route selection using the Algorithm 4. After the route selection per coalition, an exhaustive search algorithm combines the coalitions with different subset of tasks to determine all possible coalition combination, and it also determines the routes to be considered within each coalitions' combination owing to the fact that one worker can just perform one route within a coalition combination. This algorithm is similar to the one presented in Algorithm 3 without the line that runs the PSO algorithm because the route selection was already performed for each coalition. The coalition combination with the highest value of ratio between aggregated QoI and the product of the budget and maximum response time given by Eq. (3) is used for workers selection by MCS platform. Finally, a fair allocation of the routes per coalition is used, which means that one route per coalition in the solution is allocated at a time subject to the budget of the involved tasks.

From the example given in Section 3, we demonstrated in the following how our modified algorithm works. Table 6 identifies all coalitions that can be formed based on the preferred routes sent by the workers.

Table 6 indicates that 6 coalitions can be identified based on the workers' preferred routes. However, each solution should select only disjoint coalitions, which means that they don't share tasks in common. Thus, Table 7 summarizes the possible solutions.

Table 7 shows all the possible routes. However, it should be noticed that only one route per worker is selected for one coalition in each solution. Further analysis for each combination of coalitions is more complex and it is not included as illustrative example. Our approach takes into account all possible solutions and selects the one that provides the maximum ratio between the expected aggregated QoI and the product of the average budget and maximum response time.

Table 7
Coalition combinations.

Solution	Coalition	Tasks	Routes
1	(1,6)	$(T_1, T_2), T_3$	[6,16] [10,20,13,10]
2	(2,5)	$(T_1, T_3), T_2$	[2,12] [4,8,18,15]
3	(3,4)	$(T_2, T_3), T_1$	[7,17,1,11] [9,5,19,14]
3	(4,5,6)	T_1, T_2, T_3	[9,5,19,14] [4,8,18,15] [10,20,13,3]

In summary, the proposed solution selects the routes and the group of tasks that allows the MCS platform to maximize the total aggregated quality of information while minimizing the total budget and response time based on the workers' ability of establishing their preference to perform routes of consecutive tasks.

4. Benchmark model and performance metrics

Here, we briefly describe two benchmark models that we use for comparison purposes as well as the definition of the selected performance metrics to carry out this comparison.

4.1. Consecutive tasks allocation using PSO (PSO-MOA)

In our prior work [10], a service computing framework for task management in MCS system was proposed. This approach is centralized solution for the allocation of consecutive tasks to be performed by the available workers in the MCS system taking into account the convenience of those workers based on their reputation, proximity and their willingness to participate. This model provides a solution considering several components such as a multi-objective aware PSO based single task allocation algorithm, reputation evolution and delegation mechanisms.

PSO is used to solve the workers selection for each arriving task. Therefore, the approach is a single task allocation that maximizes the aggregated quality of information while minimizing the budget and the response time for a single task considering all workers willing to participate in the sensing task even those that are already performing another task. Moreover, PSO algorithm also determines the optimal payment for the workers which depends on the worker's reputation.

Finally, our previous work has other two components, i.e. the mechanism to evaluate the worker's reputation and the delegation mechanism, which are out of the scope of the current proposal.

To perform a fair comparison with our new proposal, the objective function is modified taking into account that the payment is determined by each worker and is given as follows:

$$\max_{\mathbf{x}} \frac{\left(\sum_{j\in W} q_j^i \beta_j^i\right) - q^i}{\left(\sum_{j\in W} d_j^i p_j\right) \times \max_{j\in W}(t_j^i)} \tag{15}$$

where q_j^i is the quality of information of worker j and q^i is the minimum quality of information required by the task i. The numerator in the objective function indicates the aggregated quality of information. The denominator in (15) corresponds to the product of task budget and the response time to gather the information from the allocated workers to the task (i.e. the maximum time of the allocated workers to perform a task). The budget is estimated as the payment paid per traveled kilometer to the worker, P_j, multiplied by the traveled distance per worker, d_j^i. The second term in the denominator is introduced to reduce the total required time to collect the sensed information. The Algorithm 5 describes how this approach works.

Algorithm 5: PSO-MOA worker selection algorithm.

```
Data: Worker Locations (l_j),
      Worker Payment Demand (p_j),
      Tasks Coordinates (l^i),
      Maximum Budget per Tasks B^i_max,
      Coverage radius d^i
      Required Time t^i
Result: Set of worker allocated to the tasks per worker (β^i_j).
begin
   Generate initial swarm with the particle positions Y^i_j = (β^i_j) and
    velocities randomly v^i_j;
   Evaluate Fitness Function;
   Determine first global best of the swarm;
   while k ≤ MaxIteration do
      Update Position;
      Evaluate Fitness Function;
      Determine best local for each particle;
      Determine best global in the swarm and update the best global;
      Update the inertia parameter w;
      Update velocity ;
   end
end
```

4.2. Tasks clustering based on geographical locations (Cluster-Geo)

The work in [6] presents a heuristic approach where task clustering and workers grouping is proposed to reduce the complexity of a multi-task multi-worker allocation problem. In their work, the clustering of tasks is carried using their geographic locations regardless the tasks distribution or number of tasks available while the workers selection for each task cluster is performed using Genetic Algorithm aiming at maximizing the QoS and minimizing the traveled distance. To perform a fair comparison with our proposal, we take the idea of clustering tasks and for the worker selection within each cluster of tasks and use a PSO algorithm given in 5. This is owing to the fact that our objective function is different from their objective function. The algorithm for task clustering is shown in 6.

Algorithm 6: Location based task clustering algorithm.

```
Data: Tasks coordinates l^i,
      Tasks (q^i, B^i_max, t^i )
Result: Clusters of tasks C,
        Tasks belonging to each cluster γ^c_i,
        Workers Selection for each cluster β^i_j
begin
   eval ← Silhoutte Evaluation;
   k ←eval.OptimalK;
   [idx_c, C] =kmedoids(l^i, k);
   for i ← 1 to k do
      T^c ← Tasks belonging to the cluster C(i);
      q^C(i) ← max(q^i(idx_c == i));
      B^C(i) ← ∑(B^i_max(idx_c == i));
      t^C(i) ← max(t^i(idx_c == i)) ;
      Evaluate Equation (15);
   end
   Sort clusters in descending order of its value of Equation (15);
   for each cluster c ∈ C do
      Run Algorithm 5;
   end
end
```

The input of Algorithm 6 is the dataset of tasks, which includes the locations, l^i, the minimum QoI, q^i, the maximum budget allowed B^i_{max} and required response time t^i. In [6], Silhouette evaluation criterion was adopted to evaluate the number of clusters needed based on the Euclidean distance of the tasks. Let k be the number of clusters needed, idx_c be the index of the cluster c to which the task belongs to, and C is the set of clusters. K-mediods partitioning algorithm uses the number of clusters that achieves the highest separation score to cluster the tasks. Each cluster is represented by the included tasks, their locations, maximum QoI, $q^C(i)$, budget, $B^C(i)$ and maximum response time, $t^C(i)$.

For Cluster-Geo model, the setup of the cluster size plays an important role and can have negative effect on the PSO convergence leading to a premature convergence. In our case, the models will determine the cluster size based on the preferred workers routes. This means that the only way that we can have a higher cluster size is to increase the required response time for the tasks such as the worker can select more tasks in one route to have more payment.

4.3. Performance metrics

- **Task allocation rate:** This metric represents the percentage of tasks being effectively allocated to workers.

$$\overline{\phi_T} = \frac{T_{assigned}}{|T|} \tag{16}$$

where $T_{assigned}$ is the number of tasks that are effectively allocated and performed within their respective response time and it can be determined as:

$$T_{assigned} = \sum_{c\in C}\sum_{i\in T}\gamma^c_j \tag{17}$$

- **Average QoI Satisfaction per Task:** This metric measures the average satisfaction of the quality of information over the set of tasks in a given instant.

$$\overline{S_{QoI}} = \frac{\sum\limits_{c\in C}\sum\limits_{i\in T}\sum\limits_{r\in R}\sum\limits_{j\in W} max\left(1, \beta^{r,i}_{c,j}q^{i,r}_j - q^i\right)}{T_{assigned}} \tag{18}$$

- **Effective Crowd Size**: This metric measures the number of participating workers in the MCS system.

$$\overline{SIZE} = \sum_{c\in C}\sum_{i\in T}\sum_{j\in W}\sum_{r\in R}\beta^{r,i}_{c,j} \tag{19}$$

- **Average Budget per Task**: This metric measures the average budget used per task and is given by

$$\overline{B_T} = \frac{\sum\limits_{c\in C}\sum\limits_{r\in R}\sum\limits_{i\in T}\sum\limits_{j\in W}\beta^{r,i}_{c,j}p_j d^{i,r}_j}{T_{assigned}} \tag{20}$$

- **Average Response time per task:** It indicates the average time to perform a location-based task.

$$\overline{t_T} = \frac{\sum\limits_{c\in C}\sum\limits_{r\in R}\sum\limits_{i\in T}\sum\limits_{j\in W}\beta^{r,i}_{c,j}t^i_j}{T_{assigned}} \tag{21}$$

- **Average Payment per Worker**: It indicates the average payment received by the worker per traveled kilometer and it can be expressed as follows:

$$\overline{P_W} = \frac{\sum\limits_{c\in C}\sum\limits_{r\in R}\sum\limits_{i\in T}\sum\limits_{j\in W}\beta^{r,i}_{c,j}d^{i,r}_j p_j}{\overline{SIZE}} \tag{22}$$

Table 8
Simulation parameters.

Name	Description	Value
N^i_{max}	Maximum number of workers per task i	5
B^i_{max}	Maximum Budget per Task	50
N^{max}_j	Maximum number of task per worker j	5
p_j	Requested minimum payment per traveled km for worker j	1.25 - 3
q^i	Quality of information per task i	0.75 - 1
ρ_j	Worker Reputation	0.7 - 1
ϕ_j	Confidence of worker j	0.95 -1
s_j	Worker Speed	10 - 50 km/h

- **Average Traveled Distance per Worker**: It indicates the traveled distance by the worker and it is given as follows:

$$\overline{D_W} = \frac{\sum\limits_{c\in C}\sum\limits_{r\in R}\sum\limits_{i\in T}\sum\limits_{j\in W}\beta^{r,i}_{c,j}d^{i,r}_j}{\overline{SIZE}} \tag{23}$$

- **Average Payment per km**: It indicates the traveled distance by the worker and it is given as follows:

$$\overline{P_{km}} = \frac{\sum\limits_{j\in W}\frac{\sum\limits_{c\in C}\sum\limits_{r\in R}\sum\limits_{i\in T}\beta^{r,i}_{c,j}p_j d^{i,r}_j}{\sum\limits_{c\in C}\sum\limits_{r\in R}\sum\limits_{i\in T}\beta^{r,i}_{c,j}d^{i,r}_j}}{\overline{SIZE}} \tag{24}$$

5. Simulation results

For our simulations, we used the Foursquare dataset[2] as in [10]. Specifically, two files from this dataset are used: 1) the venues' file that represents the task locations and 2) the users' file that corresponds to the workers' locations. From venues files, we select 500 venues that are close to each other taking into account their location, then, we select workers from the users file that are within the location of the 500 selected venues. In particular, we extracted 300 venues and 1412 users to represent the tasks and workers in our model. These subsets allow us to construct realistic spatial crowdsensing scenario.[3]

We run the simulations under two incremental scenarios:

- **Incremental scenario of tasks:** The number of tasks varies from 20 tasks up to 300 tasks with incremental steps of 40 tasks keeping a fixed number of 1000 workers.
- **Incremental scenario of workers:** The number of workers varies from 200 workers up to 1000 workers with incremental steps of 200 workers having a fixed number of 100 tasks.

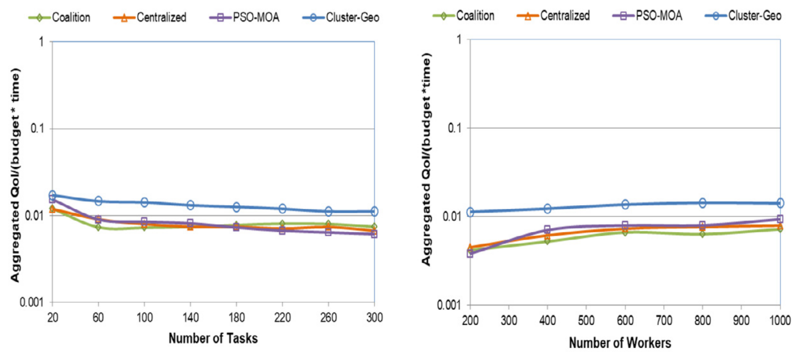
Table 8 summarize the setup used in our simulation, which is similar to the one used in our prior work in [10]. It is important to notice that where the table shows a range of values, this means that they are randomly generated between these two values.

5.1. Performance metrics for MCS platform

In this section, we present the performance metrics that the MCS platform should evaluate such as the value of the objective function of the proposed model given by Eq (1), task allocation rate, QoI satisfaction and the crowd size.

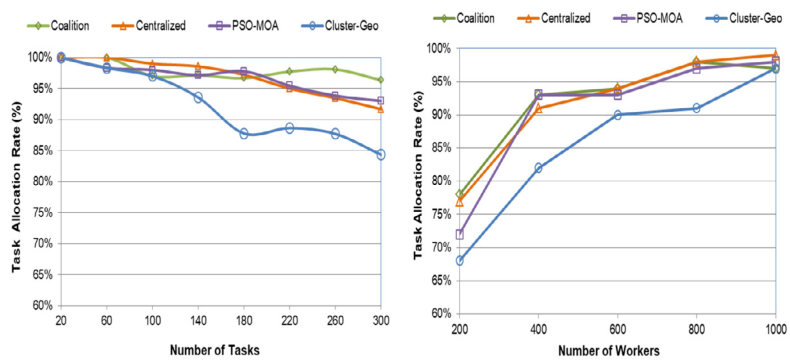
Fig. 3 shows the objective function values obtained for all models. It is worth noticing that the maximization problem is solved for each cluster in the case of Cluster-Geo model, this means that

[2] https://archive.org/details/201309_foursquare_dataset_umn.
[3] https://1drv.ms/u/s!AqOE4FreqsBrgaZUE6RXga_8ZKzi-w?e=oRMDFC.



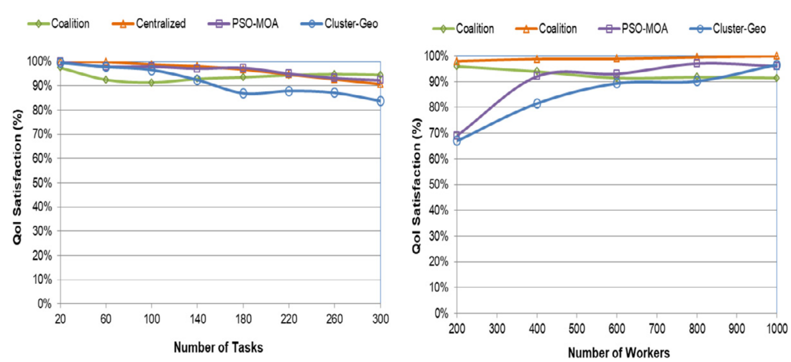
(a) Incremental scenario of tasks (b) Incremental scenario of workers

Fig. 3. Aggregated QoI divided by the product of budget and the response time.



(a) Incremental scenario of Tasks (b) Incremental scenario of Workers

Fig. 4. Task allocation rate.



(a) Incremental Scenario of Tasks (b) Incremental Scenario of Workers

Fig. 5. QoI satisfaction.

objective function is maximized within each cluster for a reduced set of tasks. On the other hand, PSO-MOA model performs the selection of the workers maximizing the objective function for each task. To get an estimated value of our objective function for the benchmark models, we sum the aggregated QoI for all tasks and divided by the product of the total budget and maximum response time. Doing this, it can be observed that Cluster-Geo model presents the best results for the objective function values while the centralized, coalition and PSO-model have similar results in both scenarios.

Fig. 4 presents the task allocation rate for the models. Fig. 4a indicates that as the number of tasks increases the models reduce their capability to allocate the tasks to the available workforce. In particular, Cluster-Geo model reduces the task allocation rate to 85% for the case of 300 tasks, while the other three models keep the allocation rate over 90%. From Fig. 4b, it can be observed that as the number of workers in the platform increases, the performance of the models increases. In general, we can notice that Cluster-Geo model presents the worst behavior for both scenarios, which can be attributed to the way that this approach performs the tasks' grouping based on locations without taking into account the proximity of workers to perform these tasks.

The average quality of information satisfaction per tasks given in (18) is presented in Fig. 5. Fig. 5a shows that the centralized model obtains QoI satisfaction values between 90% and 100% as well as the coalition and PSO-MOA models for the incremental scenario of tasks. However, Cluster-Geo model presents QoI satisfaction values lower than 90% for more than 140 tasks. Fig. 5b presents the quality satisfaction for the incremental scenario of workers. It can be observed that Cluster-Geo model starts with a

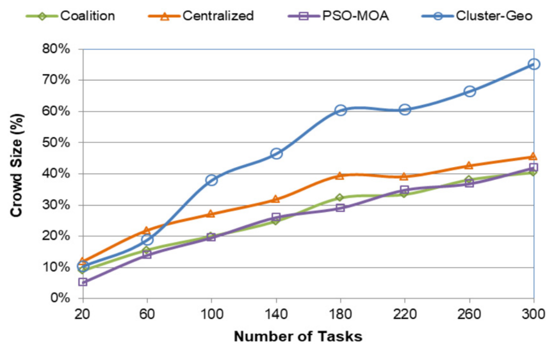


Fig. 6. Crowd size percentage for incremental scenario of tasks.

Table 9
Task publisher metrics: incremental scenario of tasks.

No.	Average QoI				Average budget			
	Central	Coal.	Cluster-Geo	PSO-MOA	Central	Coal.	Cluster Geo	PSO-MOA
20	8.60	5.11	4.61	2.14	28.37	15.44	9.13	4.02
60	6.97	3.39	2.85	2.11	27.05	13.84	5.66	5.54
100	6.31	3.27	3.49	2.06	28.02	13.43	7.76	5.80
140	6.00	3.45	3.17	2.21	27.65	13.94	7.26	6.80
180	6.01	3.81	3.51	2.08	27.80	15.18	8.41	6.72
220	5.76	4.02	2.85	2.08	27.71	15.70	6.84	7.38
260	5.61	3.83	2.69	2.05	27.16	14.94	6.69	7.56
300	5.40	3.98	2.80	2.08	27.44	17.23	7.11	8.01

67% of QoI satisfaction for 200 workers and reaches 97% for 1000 workers. In the case of PSO-MOA model, QoI satisfaction increases from 70% to 97%. This means that PSO and Cluster-Geo model are severely affected if the workforce is reduced while the centralized and coalition models guarantee the QoI satisfaction over 90%.

Fig. 6 shows the percentage of the workforce that is involved in the execution of the sensing tasks. As expected, as the number of tasks increases, the number of workers required to perform more tasks increases for the four models. In the particular case of Cluster-Geo model, 30% more of the workforce is required to perform 300 tasks in comparison with the other three models.

Finally, it is important to remark that based on the workers' route preferences the average coalition size for both models (i.e. centralized and coalition) varies from 1.5 to 2 with a standard deviation of 0.7. In the case of Cluster-Geo model, the average coalition size varies from 2 to 2.11 with a standard deviation of 1, which depends mostly on the geographical location of the tasks.

5.2. Task publisher's metrics

Here, we evaluate the performance metrics from the perspective of a task publisher (i.e. average budget and average quality of information per task). Tables 9 and 10 summarizes the average quality of information and the average budget after running the algorithms the for the models considering both incremental scenarios. From the point of view of a task publisher, Tables 9 and 10 show the centralized model requires double amount of budget for a higher quality of information of the sensing task than the coalition model.

Let's analyze the highlighted line with 220 tasks in Table 9, it is shown that the centralized model achieves a QoI of 5.76 while the coalition model reaches only 4.02. However, the centralized model is using double amount of budget for just 40% extra of the QoI reached by the coalition model. On the other hand, Cluster-Geo and PSO-MOA models provide better results from the task publisher perspective by reducing the budget but these models also reduce aggregated quality of information. However, this QoI reduction could be detrimental if some workers are not able finish the allocated task.

Table 10
Task publisher metrics: incremental scenario of workers).

C.S.	Average QoI				Average budget			
	Central	Coal.	Cluster-Geo	PSO-MOA	Central	Coal.	Cluster Geo	PSO-MOA
200	3.71	2.58	2.14	2.07	25.17	16.38	4.95	12.97
400	5.04	2.79	2.41	2.31	27.08	14.67	4.99	8.38
600	5.79	3.20	2.53	2.18	26.90	13.99	4.83	6.74
800	6.11	2.98	3.50	2.20	27.25	13.42	7.60	6.84
1000	6.31	3.27	3.49	2.13	28.02	13.43	7.76	5.41

Table 11
Worker metrics: incremental scenario of tasks.

No.	Average distance (km)				Average payment per km ($)			
	Central	Coal.	Cluster-Geo	PSO-MOA	Central	Coal.	Cluster Geo	PSO-MOA
20	6.68	5.55	6.00	4.17	0.89	0.69	0.82	0.70
60	7.42	6.74	5.30	4.17	1.28	1.03	1.09	0.89
100	8.18	7.01	4.44	4.06	1.53	1.22	1.04	1.08
140	9.18	9.17	4.78	4.46	1.59	1.48	0.77	1.14
180	8.84	8.54	4.95	4.54	1.82	1.61	0.89	1.24
220	10.19	10.53	4.77	4.49	1.80	1.41	1.11	1.38
260	10.51	10.22	4.87	4.71	1.86	1.54	1.13	1.41
300	10.77	10.90	4.60	4.71	1.93	1.62	1.26	1.54

Table 12
Worker metrics: incremental scenario of workers.

C.S.	Average distance (km)				Average payment per km ($)			
	Central	Coal.	Cluster-Geo	PSO-MOA	Central	Coal.	Cluster Geo	PSO-MOA
200	11.62	13.97	4.76	5.48	1.96	1.46	2.47	1.90
400	9.38	10.47	4.20	5.13	1.81	1.53	1.04	1.58
600	9.02	9.14	4.81	4.71	1.56	1.26	0.84	1.24
800	9.04	7.69	5.20	4.50	1.50	1.31	0.87	1.41
1000	8.18	7.01	4.44	4.27	1.53	1.22	1.04	1.13

From Table 10, it can be observed that as the number of workers increases, the budget of the coalition and PSO-MOA models is reduced. This means that both models are selecting workers that allows them to reduce the average budget per task. In the particular case of the Cluster-Geo model, the budget is increased as the number of workers increases, which can be attributed to the fact that more workers will be in the proximity of the cluster to perform the tasks belonging to the cluster. For PSO-MOA model, the budget is highly reduced as the number of workers increases. This is owing to the fact that the workers' selection is different as more workers are willing to perform the same task and these new workers are asking for less payment than the others while for the coalition formation models, neither the quality of information nor budget are highly affected by the increase of the crowd size.

5.3. Worker's performance metrics

In this section, the performance metrics from the workers' point of view (i.e. average traveled distance and average payment per worker) are presented in Tables 11 and 12 for the first scenario (i.e. incremental number of tasks) and second scenario (i.e. incremental number of workers) respectively.

From Table 11, one can observe that for the case with 140 tasks (highlighted line), coalition and centralized models present similar values of traveled distance and payment per kilometer while Cluster-Geo and PSO-model have around 50% of the average traveled distance per worker and less payment per km. In general, it

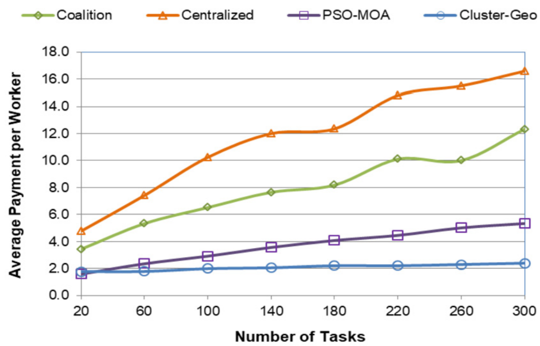


Fig. 7. Average payment per worker in an incremental scenario of tasks.

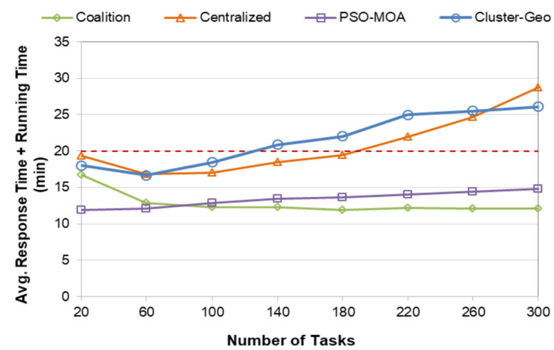


Fig. 8. Average response time plus running time for the incremental scenario of tasks.

can be noticed that the traveled distance and the average payment per worker increases as the number of sensing tasks increases. This is owing to the fact that there will be more sensing tasks that the workers will prefer to perform because of their proximity to these new tasks. In addition, it can be observed that the models forming coalition based on route preferences present promising benefits for the workers unlike the Cluster-Geo and PSO-MOA models. Finally, it seems that the centralized model would provide better payment for the worker in the long term, however, the main limitation of this model is the high running time, which is analyzed later in section 5.4. High running time can deprive the workers to obtain their payment due to the fact the MCS platform will give the sensed information to the task publisher after the requested time.

Fig. 7 shows the average payment per worker for the four models in the incremental scenario of tasks. It can be observed that the centralized model can provide better average payment per worker followed by coalition model. It should be noted that the centralized model requires high execution time due to its complexity, implying a higher response time, which can deprive the workers of receiving the payment from the tasks' publishers. Cluster-Geo model presents the lowest values of payment per worker, this can be attributed to the fact that cluster of task is based on the proximity of tasks disregarding workers' proximity. On the other hand, PSO-MOA model sorts the tasks in descending order of the objective function values, then, the worker selection is performed for one task at a time in such a way the first task to be allocated is the one that maximizes the objective function. Thus, the first tasks could select the best workers leaving a reduced workforce with lower QoI for the rest of the tasks.

From Table 12, it can be noticed that as the number of workers increases the traveled distance and payment is reduced, which is owing to the fact that more workers are willing to carry out the same number of sensing tasks and there are more available routes that can reduce the budget that a task publisher has to spend to have the sensed data.

5.4. Average response time and running time

We also analyze the impact of variation of the number of sensing tasks on the response time taking into account that there is no need to pay to the workers that deliver the sensed data after the required time because this information will be useless. Fig. 8 shows the sum of the average response time and the running time of each model to determine the solution. In this figure, the dashed line represents the average required time per tasks. The response time depends on the order of the workers' route preferences to perform a set of tasks.

For the centralized model, the running time depends on the complexity of the coalition formation and the selection of the coalition combination that provides the higher objective function using PSO algorithm for all coalition combinations. In the case of the coalition and Cluster-Geo model, the running time corresponds to the complexity of the coalition formation plus the running time of PSO algorithm multiplied by the number of coalitions. PSO-MOA model does not perform any type of coalition formation, therefore, the running time depends on the convergence time of the PSO algorithm for a single task allocation multiplied by the number of tasks.

It can be observed that the centralized and Cluster-Geo models will fail to pay the selected workers in the case of having more than 180 tasks (120 tasks for Cluster-Geo model) because the average response time plus the running time of the algorithm is higher than the average required response time of the set of tasks. This means that the workers will be discouraged to participate using any of these types of approach.

In summary, the coalition model can guarantee the payment to the workers because of the lower running time while satisfying the constraint of budget and response time with sufficient aggregated quality of information for task publishers. This enables the MCS platform to have always engaged both types of participants (i.e. task publishers and workers) to the MCS system.

6. Conclusion

In this paper, a task coalition framework for mobile crowdsensing that takes into account the interest of all participating parties (i.e. task initiators, platform and contributors) is proposed. Unlike prior works, this framework is based on two main components: 1) the strategies selection (e.g. best routes) for the workers, and 2) the coalition formation based on the selection of set of tasks based on workers' best routes that maximizes the ratio between the aggregated quality of information and the budget. We run extensive simulations using a real dataset for two scenarios: 1) a fixed number of workers varying number of tasks, and 2) a fixed number of tasks varying the number of workers. In the case of task incremental scenario with 1000 workers, it was shown that the coalition and centralized models allocate between 90% and 100% of the sensing tasks having values of QoI satisfaction between 90% and 100%. In the case of workers incremental scenario with 100 tasks, centralized and coalition models allocates between 78% to 100% of the tasks with QoI Satisfaction between 85% and 100%. Moreover, it was shown that the task coalition framework based on individual worker's route selection can guarantee QoI satisfaction of the sensing tasks while increasing the workers' payment without having to increase the budget per task as the centralized model. Finally, owing to the fact that the proposed algorithm has lower running time than the centralized model, the MCS platform can guarantee the workers' payment such as the platform can keep engaged both types of participants (i.e. task publishers and workers).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] M.H. Cheung, R. Southwell, F. Hou, J. Huang, Distributed time-sensitive task selection in mobile crowdsensing, in: MobiHoc, ACM, 2015, pp. 157–166.

[2] B. Guo, Z. Wang, Z. Yu, Y. Wang, N.Y. Yen, R. Huang, X. Zhou, Mobile crowd sensing and computing: the review of an emerging human-powered sensing paradigm, ACM Comput. Surv. 48 (1) (2015) 7:1–7:31.

[3] A.M. Arslan, N. Agatz, L. Kroon, R. Zuidwijk, Crowdsourced delivery—a dynamic pickup and delivery problem with ad hoc drivers, Transp. Sci. 53 (1) (2019) 222–235.

[4] C. Zhou, C.-K. Tham, M. Motani, QOATA: Qoi-aware task allocation scheme for mobile crowdsensing under limited budget, in: ISSNIP, April 2015, pp. 1–6.

[5] H. Yu, C. Miao, Z. Shen, C. Leung, Quality and budget aware task allocation for spatial crowdsourcing, in: AAMAS, 2015, pp. 1689–1690.

[6] M. Abououf, R. Mizouni, S. Singh, H. Otrok, A. Ouali, Multi-worker multi-task selection framework in mobile crowd sourcing, J. Netw. Comput. Appl. 130 (2019) 52–62.

[7] F. Restuccia, S.K. Das, J. Payton, Incentive mechanisms for participatory sensing: survey and research challenges, ACM Trans. Sens. Netw. 12 (2) (Apr. 2016).

[8] A. Suliman, H. Otrok, R. Mizouni, S. Singh, A. Ouali, A greedy-proof incentive-compatible mechanism for group recruitment in mobile crowd sensing, Future Gener. Comput. Syst. 101 (2019) 1158–1167.

[9] M. Abououf, H. Otrok, S. Singh, R. Mizouni, A. Ouali, A misbehaving-proof game theoretical selection approach for mobile crowd sourcing, IEEE Access 8 (2020) 730–58 741.

[10] R. Estrada, R. Mizouni, H. Otrok, A. Ouali, J. Bentahar, A crowd-sensing framework for allocation of time-constrained and location-based tasks, IEEE Trans. Serv. Comput. 13 (5) (2020) 769–785.

[11] D.E. Boubiche, M. Imran, A. Maqsood, M. Shoaib, Mobile crowd sensing – taxonomy, applications, challenges, and solutions, Comput. Hum. Behav. 101 (2019) 352–370.

[12] J. Zuo, H. Xia, S. Liu, Y. Qiao, Mapping urban environmental noise using smartphones, Sensors 16 (10) (Oct 2016) 1692.

[13] A. Coletta, N. Bartolini, G. Maselli, A. Kehs, P. McCloskey, D.P. Hughes, Optimal deployment in crowd sensing for plant disease diagnosis in developing countries, IEEE Int. Things J. (2020) 1–1.

[14] S.Z. Khan, W.M. Abdul Rahuman, S. Dey, T. Anwar, A.S.M. Kayes, Roadcrowd: an approach to road traffic forecasting at junctions using crowd-sourcing and Bayesian model, in: 2017 International Conference on Research and Innovation in Information Systems (ICRIIS), 2017, pp. 1–6.

[15] X. Kong, H. Gao, O. Alfarraj, Q. Ni, C. Zheng, G. Shen, Huad: hierarchical urban anomaly detection based on spatio-temporal data, IEEE Access 8 (2020) 573–26 582.

[16] M. Elhamshary, A. Basalmah, M. Youssef, A fine-grained indoor location-based social network, IEEE Trans. Mob. Comput. 16 (5) (2017) 1203–1217.

[17] Z. Yu, Y. Feng, H. Xu, X. Zhou, Recommending travel packages based on mobile crowdsourced data, IEEE Commun. Mag. 52 (8) (2014) 56–62.

[18] M. Grossi, A sensor-centric survey on the development of smartphone measurement and sensing systems, Measurement 135 (2019) 572–592.

[19] A. Capponi, C. Fiandrino, B. Kantarci, L. Foschini, D. Kliazovich, P. Bouvry, A survey on mobile crowdsensing systems: challenges, solutions, and opportunities, IEEE Commun. Surv. Tutor. 21 (3) (2019) 2419–2465.

[20] J. Wang, L. Wang, Y. Wang, D. Zhang, L. Kong, Task allocation in mobile crowd sensing: state-of-the-art and future opportunities, IEEE Int. Things J. 5 (5) (2018) 3747–3757.

[21] H. Xiong, D. Zhang, G. Chen, L. Wang, V. Gauthier, L.E. Barnes, iCrowd: near-optimal task allocation for piggyback crowdsensing, IEEE Trans. Mob. Comput. 15 (8) (2016) 2010–2022.

[22] C.H. Liu, B. Zhang, X. Su, J. Ma, W. Wang, K.K. Leung, Energy-aware participant selection for smartphone-enabled mobile crowd sensing, IEEE Syst. J. 11 (3) (2017) 1435–1446.

[23] Y. Wang, Z. Cai, Z. Zhan, B. Zhao, X. Tong, L. Qi, Walrasian equilibrium-based multiobjective optimization for task allocation in mobile crowdsourcing, IEEE Trans. Comput. Soc. Syst. (2020) 1–14.

[24] M. Karaliopoulos, O. Telelis, I. Koutsopoulos, User recruitment for mobile crowdsensing over opportunistic networks, in: 2015 IEEE Conference on Computer Communications (INFOCOM), 2015, pp. 2254–2262.

[25] W. Wang, H. Gao, C.H. Liu, K.K. Leung, Credible and energy-aware participant selection with limited task budget for mobile crowd sensing, Ad Hoc Netw. 43 (2016) 56–70, smart Wireless Access Networks and Systems for Smart Cities.

[26] J. Wang, Y. Wang, D. Zhang, F. Wang, Y. He, L. Ma, PSAllocator: multi-task allocation for participatory sensing with sensing capability constraints, in: Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing, 2017, pp. 1139–1151.

[27] I.J. Vergara-Laurens, D. Mendez, M.A. Labrador, Privacy, quality of information, and energy consumption in participatory sensing systems, in: 2014 IEEE International Conference on Pervasive Computing and Communications (PerCom), 2014, pp. 199–207.

[28] L. Wang, D. Zhang, H. Xiong, J.P. Gibson, C. Chen, B. Xie, ecoSense: minimize participants' total 3g data cost in mobile crowdsensing using opportunistic relays, IEEE Trans. Syst. Man Cybern. Syst. 47 (6) (2017) 965–978.

[29] C. Wang, C. Li, C. Qin, W. Wang, X. Li, Maximizing spatial–temporal coverage in mobile crowd-sensing based on public transports with predictable trajectory, Int. J. Distrib. Sens. Netw. 14 (8) (2018) 1550147718795351.

[30] R. Azzam, R. Mizouni, H. Otrok, A. Ouali, S. Singh, Grs: a group-based recruitment system for mobile crowd sensing, J. Netw. Comput. Appl. 72 (2016) 38–50.

[31] C. Lai, X. Zhang, Duration-sensitive task allocation for mobile crowd sensing, IEEE Syst. J. (2020) 1–12.

[32] H. Shah-Mansouri, V. Wong, Profit maximization in mobile crowdsourcing: a truthful auction mechanism, in: ICC, June 2015, pp. 3216–3221.

[33] Y. Liu, H. Li, G. Zhao, J. Duan, Reverse auction based incentive mechanism for location-aware sensing in mobile crowd sensing, in: 2018 IEEE International Conference on Communications (ICC), 2018, pp. 1–6.

[34] T. Luo, S.K. Das, H.P. Tan, L. Xia, Incentive mechanism design for crowdsourcing: an all-pay auction approach, ACM Trans. Intell. Syst. Technol. 7 (3) (Feb. 2016) 35:1–35:26.

[35] Z. Feng, Y. Zhu, Q. Zhang, L.M. Ni, A.V. Vasilakos, Trac: truthful auction for location-aware collaborative sensing in mobile crowdsourcing, in: IEEE INFOCOM 2014 - IEEE Conference on Computer Communications, April 2014, pp. 1231–1239.

[36] X. Zhang, L. Gao, B. Cao, Z. Li, M. Wang, A double auction mechanism for mobile crowd sensing with data reuse, in: GLOBECOM 2017 - 2017 IEEE Global Communications Conference, 2017, pp. 1–6.

[37] Y. Du, Y. Sun, H. Huang, L. Huang, H. Xu, Y. Bao, H. Guo, Bayesian co-clustering truth discovery for mobile crowd sensing systems, IEEE Trans. Ind. Inform. 16 (2) (2020) 1045–1057.

[38] J. Xu, Z. Rao, L. Xu, D. Yang, T. Li, Incentive mechanism for multiple cooperative tasks with compatible users in mobile crowd sensing via online communities, IEEE Trans. Mob. Comput. 19 (7) (2020) 1618–1633.

[39] D. Jungnickel, Graphs, Networks and Algorithms, 4th ed., Springer Berlin, Berlin, Germany, 2012.

[40] T. Hu, M. Xiao, C. Hu, G. Gao, B. Wang, A qos-sensitive task assignment algorithm for mobile crowdsensing, Pervasive Mob. Comput. 41 (2017) 333–342 [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574119217300317.

[41] D. Wang, D. Tan, L. Liu, Particle swarm optimization algorithm: an overview, Soft Comput. 22 (2) (Jan. 2018) 387–408.