IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received October 8, 2018, accepted December 3, 2018, date of publication December 19, 2018,
date of current version January 11, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2888696

Gale-Shapley Matching Game Selection–A Framework for User Satisfaction

MENATALLA ABOUOUF[1], (Member, IEEE), SHAKTI SINGH[1], (Member, IEEE), HADI OTROK[1], (Senior Member, IEEE), RABEB MIZOUNI[1], AND ANIS OUALI[2]
[1]Center on Cyber-Physical Systems, Electrical and Computer Engineering Department, Khalifa University, Abu Dhabi, United Arab Emirates
[2]Emirates ICT Innovation Center, Abu Dhabi, United Arab Emirates

Corresponding author: Menatalla Abououf (menatalla.abououf@ku.ac.ae)

This work was supported by the Khalifa University Internal Fund (KUIRF level 2) under Project 8474000012.

ABSTRACT In large-scale mobile crowd sensing systems, multi-task-oriented worker selection has shown an increased efficiency in workers' allocation. However, existing solutions for multi-task selection mainly focus on meeting the requirements of the available tasks and fall short in considering workers' preferences. Assigning workers to their preferred tasks should substantially improve the possibility that they will perform the tasks assigned to them, which will improve the quality of the sensing outcome. In this paper, we propose to use the Gale–Shapley matching game selection to allocate multiple workers to multiple tasks based on the preferences of both the tasks and workers. It aims at maximizing the level of satisfaction for the workers, by assigning them to their most preferred tasks, as well as maximizing the Quality of Service (QoS) and the completion confidence of the tasks. Simulations based on the real-life dataset show that the proposed approach outperforms other multi-task allocation benchmark in terms of the completion confidence of the tasks, the QoS of the sensing outcome, and the workers' satisfaction level without compromising on their traveling distance.

INDEX TERMS Multi-task allocation, mobile crowd sourcing, Gale-Shapley, workers' satisfaction.

I. INTRODUCTION

The on-demand services provided by mobile crowd sourcing (MCS) have created work opportunities for mobile users [1], [2], and have also been beneficial for businesses. MCS systems have great potential in outsourcing tasks that require large-scale data collection [3], [4], where numerous emerging platforms have utilized crowdsourced workforce, often via the use of mobile applications [5], such as ***Waze*** and ***Placemeter***. ***Waze*** is one of the most popular applications that gathers real-time traffic conditions evaluated and submitted by users based on their GPS locations [6], [7]. It helps mobile users to avoid routes with accidents, traffic, and hazards as well as sharing information such as gas prices in different stations. This allows users to save commuting time and gas money [8]. ***Placemeter*** is an online platform where users can upload videos of the streets anonymously in return for monetary incentives. The platform aggregates videos by users and use them to analyze pedestrian and traffic trends in the area. Using computer vision, the platform could study the effect of temporary events in a specific area, find opportunities for infrastructure development, or detect crowded and inactive areas [9].

The collected data, used for analyzing and decision making, reflects the wisdom of the crowds [10]–[12]. Thus, input from multiple workers is needed to reach a consensus [13]. However, one of the main challenges in MCS is the selection of workers that would achieve the best performance in terms of the Quality of Service (QoS) of the task.

Most selection algorithms in literature optimize one or more parameters in the selection of workers, such as the distance traveled, the QoS, or the number of workers involved [14]. The multi-objective optimization models make the selection problem even more challenging. In addition, in a multitasking framework, selecting multiple workers to serve multiple tasks in a large solution space is more efficient in terms of managing human resources, but it becomes considerably harder, since an increase in the number of tasks and the number of participants makes the solution combinations overly complex [15]. Moreover, The willingness of the workers in performing the assigned tasks are maximized when their preferences are considered during selection. In [16] and [17] the workers preferences are considered, however, the approaches are for single-task assignment.

Existing multi-task allocation frameworks only consider the tasks' requirements and constraints, without taking into account users' needs such as their satisfaction. User satisfaction is formulated as the workers' preferences or ranks for the available tasks in comparison to the actual order of their allocated tasks. Indeed, increasing the users' satisfaction motivates the workers to perform tasks which would have a positive impact on the QoS of the sensing outcome.

To address this shortcoming, in this work, a novel approach is proposed, which assigns workers to tasks considering both the task's requirements and the worker's preferences. A game theory approach, based on the one-to-many *college admissions problem* is proposed where stable matching between multiple workers and multiple tasks are formed based on the workers' preferences of tasks, as well as the tasks' preferences of workers. The main contribution of this work consists of:

- Adaptation of the Gale-Shapley one-to-many college admissions matching problem to a many-to-many matching problem, where multiple workers are selected to perform multiple tasks.
- A multi-worker selection mechanism that maximizes 1) the user satisfaction in the allocated tasks, 2) the completion confidence achieved by tasks, and 3) the overall QoS.

Simulations using real-life datasets are conducted and the results compared with a benchmark which uses a greedy-enhanced genetic algorithm (GGA-I) to select multiple workers for multiple tasks [15]. The results show that the proposed model outperforms the benchmark in terms of users' satisfaction, confidence, and achieved QoS of the tasks, without compromising the workers' traveling distance.

II. BACKGROUND AND RELATED WORK

A. BACKGROUND

MOBILE CROWD SOURCING

Mobile crowd sourcing/sensing is a paradigm that exploits connected smart phones of users in a predefined area of interest (AoI), to sense a phenomenon of interest. It usually involves asking the user to travel to the location of the sensing task [1]. The development of cellular mobile networks such as 4G and LTE, as well as the numerous embedded sensors in mobile phones such as GPS, temperature, and camera has made MCS attractive [2], [3], [18]. The users in an MCS system answer tasks requested by task publishers at the expense of their time, traveling cost, and the use of their device resources such as power or cellular data consumption, in return for an incentive determined by the task publisher [4], [19]. Incentives could be of different forms such as monetary payment, entertainment, or various services [20].

THE GALE-SHAPLEY GAME

In this work, the Gale-Shapley matching game is used to match multiple workers to their most preferred tasks. The game can be classified as a bipartite matching problem with two-sided preferences [21]. Participants in the game are divided into two independent sets, where those belonging to each set rank the members of the other set in an ordinal preference. The aim is to match two participants belonging to different sets with each other. The Gale-Shapley game is known to be a centralized matching algorithm in which the solution always starts from an empty set, and iteratively reach the stable matching solution [21]. The most commonly used stable matching problems are briefly discussed, as follows:

- The *Stable Marriage* problem is the classical one-to-one matching problem where participating agents are divided into a set of women and a set of men. Each participant ranks the members of the opposite sex in order of marriage preference [22]. The marriage between men and women is considered unstable if there exists a couple that are not paired together but would rather be matched together [23].
- The *Hospitals Residents* or the *College Admission* problem is the one-to-many extension of the stable marriage problem [21]. Here, the hospital/college can be matched with multiple residents/students, until a maximum capacity is reached. In college admission, each student proposes to their most preferred college. If the student is accepted, the matching is made; if not, the student removes the college from his or her list and proposes to the next preferred college. This continues until all students are admitted to colleges or the colleges have accepted enough students [24].

B. RELATED WORK

An extensive research has been carried out for allocation of workers in mobile crowd sourcing. Current research is geared more towards multi-task assignment [14], [16], [25] due to its proven efficiency in terms of using the available human resources over single-task assignment in a large-scale MCS [15]. In this section, the related work proposed for workers selection in participatory and opportunistic MCS systems are summarized.

A sequential-task assignment model is presented in an online real-time framework for opportunistic crowd sourcing [16]. The model uses human behavior factors for modeling task profile by requesting workers to tag similar objects, such as images, creating a keyword directory. The workability of the worker, or the probability of a worker accepting and finishing the task in the allotted time, is evaluated based on his or her interests in the keyword used to describe the task. In this work, the worker's interaction is required by showing interest in specific tasks in which she or he wishes to be selected.

Another framework where user interaction is required is presented in [17]. The workers are evaluated based on the credibility of interaction with other users. Mobile users sharing similar social relationships are clustered and based on evaluating the trust among the users, the task's route is established between the task requester and the task providers. This work, however, considers single-task assignment for a group of workers.

In [14], a sequential allocation framework is proposed. The quality of service of each task depends on the workers' locations, reputation, and the confidence in performing the tasks. This approach utilizes Particle Swam Optimization (PSO) algorithm to allocate multiple workers that maximize the QoS of a task within the task's response time. However, this approach neither considers multi-task workers' allocation, nor user preferences.

A multi-tasking worker selection, MTasker, is proposed in [26]. The approach uses descent greedy algorithm to maximize the QoS provided for individual task, while taking into consideration the minimum QoS and coverage threshold. However, in this model, the users do not travel to perform the tasks.

Motivated by the presence of urgent spatial tasks, greedy-enhanced genetic algorithm based on intentional movement (GGA-I) is proposed in [15]. The workers are greedily allocated to their closest tasks, then using genetic algorithm on the greedy selection, the approach iteratively allocates workers for each task that will minimize the traveled distance. However, it does not consider the QoS provided for the tasks or the selected workers' preferences.

Moreover, in [25], workers allocation for spatial tasks located in under-crowded areas is investigated. The proposed approach relies on the maximum flow minimum cost (MCMF) network problem. First, the tasks are combined in task sets then the traveling distance for each worker to complete each set is computed by forming TSPs (traveling salesman problems). The path with the minimum distance is greedily selected for each worker. This approach, however, does not consider the workers' preferences or satisfaction during selection.

In summary, some approaches consider certain parameters and disregard others based on the application of the MCS system. In addition, to the best of our knowledge, none of the multi-tasking MCS systems considers worker satisfaction or user preferences during the selection. Table 1 provides a comparison for the different parameters considered by the allocation approaches presented in this section.

TABLE 1. Summary of the parameters considered in related-work.

Solution	QoS	Reputation	Distance /time	Multi-task	User Preferences	Scalability
[16]	✓	×	×	×	✓	✓
[17]	✓	✓	✓	×	✓	✓
[14]	✓	✓	✓	×	×	✓
[26]	✓	×	×	✓	×	✓
[15]	×	×	✓	✓	×	×
[25]	×	×	✓	✓	×	×

III. PROPOSED APPROACH

A. ILLUSTRATIVE EXAMPLE

In this section, a running illustrative example for the problem of allocating multiple workers to multiple tasks is discussed. An AoI, with different locations for tasks and workers, is laid out as shown in Fig. 1.

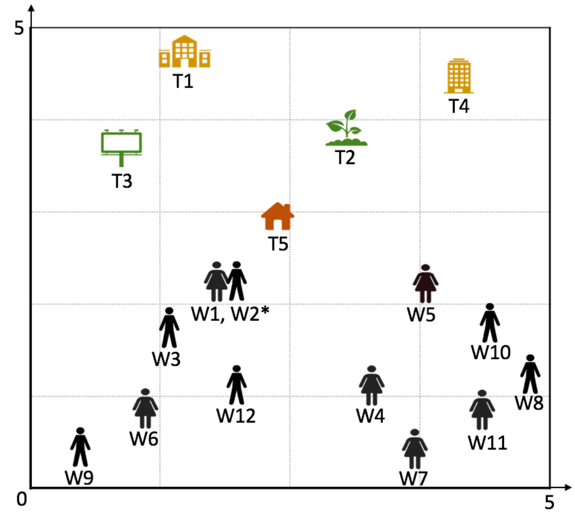

FIGURE 1. Illustrative example for the proposed approach. * W1 and W2 are at the same position and would provide equal QoS.

In this example, 5 tasks and 12 workers are available in an MCS system. Each task has a capacity of maximum 5 workers and workers have a capacity of 5 tasks. The problem here is not only allocating workers that will maximize the QoS for the task, but also to maximize the workers' satisfaction. This is done by taking into consideration the preferences of the workers for the available tasks, as well as the preferred order of assignment.

For simplicity, we assume that all the five workers meet the constraints of all available sensing tasks. In other words, every worker is eligible to perform any task.

B. MODEL DESCRIPTION

In this section, a game theoretical approach is proposed for the multi-worker multi-task allocation problem maximizing the workers' satisfaction, the tasks' completion confidence and QoS. The game is based on the college admission problem by Gale and Shapley. The players of the game are divided into two independent sets; a set $W = \{w_1, \ldots, w_n\}$ for workers and a set $T = \{t_1, \ldots, t_n\}$ for tasks. Each $t_j \in T$ is defined as a tuple in the form of $t_j =< L_j^T, Q_j^T, R_j^T, K_j^T, TC_j >$ where L_j^T is the task's GPS location in latitude-longitude coordinates, Q_j^T and R_j^T are the minimum QoS and worker's reputation requirements of the task respectively, K_j^T indicates the maximum number of workers required to perform the task, and TC_j is the time constraint to fulfill the task. Each $w_i \in W$ is defined as a tuple in the form of $w_i =< L_i^W, R_i^W, D_i^W >$, where L_i^W is the worker's current GPS location in latitude-longitude coordinates, R_i^W and D_i^W are the worker's reputation calculated from their previously assigned tasks, and the maximum distance that he or she are willing to travel, respectively. Table 2 lists the symbols used in this paper.

Based on the tasks' and workers' constraints, there exists *acceptable pairs*, $E \subseteq W \times T$, where for each pair the task

TABLE 2. List of symbols used.

Symbols	Definition
L_j^T	Location of task j
Q_j^T	Minimum QoS requirement of task j
R_j^T	Minimum reputation requirement of task j
K_j^T	Maximum number of workers required for task j
TC_j	The time constraint for task j
L_i^W	Location of worker i
R_i^W	Reputation of worker i
D_i^W	Maximum traveling distance constraint set by worker i
E	Acceptable pairs $\subseteq W \times T$
$A(t_j)$	Set of acceptable workers by task j
$A(w_i)$	Set of acceptable tasks by worker i
M	Set of assigned pairs $\subseteq E$
r	The current selection round
r_i^j	The round at which the assignment $(w_i, t_j) \in M$ is made
$T_{i,r}^j$	Time needed for worker i to reach task j at round r
$QoS_{i,r}^j$	The QoS provided to task j by worker i at round r
$C_{i,r}^j$	The confidence provided to task j by worker i at round r
$S_{i,r_i^j}^j$	The normalized satisfaction score of worker i with task j at round r_i^j

meets the constraints of the worker and vice versa. For each task, the *acceptable set of workers*, $A(t_j)$, is the set of workers that meet the R_j^T and Q_j^T constraints.

$$A(t_j) = \{w_i \in W : (w_i, t_j) \in E\}. \quad (1)$$

Likewise, for each worker, the *acceptable set of tasks*, $A(w_i)$, is the set of tasks that meet the D_i^W constraint.

$$A(w_i) = \{t_j \in T : (w_i, t_j) \in E\}. \quad (2)$$

Each task and worker creates a separate *preference list* in which they rank $A(t_j)$ and $A(w_i)$, respectively. For instance, t_1 is said to prefer w_1 to w_2 if w_1 precedes w_2 in its *preference list*.

C. ASSIGNMENT METHODOLOGY

Given the set of available tasks and the set of available workers, tasks are paired with workers that maximize the confidence of completing the tasks while providing highest possible satisfaction for the selected workers. The selection is divided into rounds, r, where each task can be performed by more than one worker, while each worker can only perform one task at a given round. For every r, each task proposes to its most preferred worker and an assignment M is made, where $M \subset E$. If a worker w_i receives more than one proposal, which happens if more than one task prefers the same worker, the final assignment is $(w_i, t_j) \in M$ where t_j is the highest ranked task in the *preference list* of the worker w_i. The other tasks will move on to their next preferred workers. This process continues until all tasks have reached their full capacity.

The *preference list* of a worker is created by sorting the tasks in $A(w_i)$ based on a task-related factor, e.g. location, nature of the task, incentives, etc. In this work, we assume that workers sort the tasks in their ascending order of their proximity relatively to his or her location. On the other hand, the *preference list* of a task is created by sorting the workers in $A(t_j)$ in their descending order of their QoS relatively to the task. Every worker in $A(t_j)$ satisfies $R_i^W \geq R_j^T$.

Table 3 presents the preference list of the workers in our illustrative example. This order is obtained based on the distance the worker has to travel to the task. It can be seen from Table 3 that all the workers initially prefer task 5 the most, since it is the closest task to them. Similarly, the workers' *preference list* of each task is given in Table 4. This order is obtained based on the QoS achieved by the workers. For instance, task 5 prefers worker 6 the most, as he or she would provide the highest QoS for this task.

TABLE 3. Preference list for each worker.

Worker	Preference list
1	$< 5, 3, 2, 1, 4 >$
2	$< 5, 3, 2, 1, 4 >$
3	$< 5, 3, 2, 1, 4 >$
4	$< 5, 2, 4, 3, 1 >$
5	$< 5, 2, 4, 3, 1 >$
6	$< 5, 3, 2, 4, 1 >$
7	$< 5, 2, 4, 3, 1 >$
8	$< 5, 2, 4, 3, 1 >$
9	$< 5, 3, 2, 1, 4 >$
10	$< 5, 2, 4, 3, 1 >$
11	$< 5, 2, 4, 3, 1 >$
12	$< 5, 2, 3, 4, 1 >$

TABLE 4. Preference list for each task.

Task	Preference list
1	$< 6, 3, 11, 9, 10, 8, 12, 5, 1, 2, 7, 4 >$
2	$< 6, 11, 10, 3, 8, 9, 5, 12, 1, 2, 7, 4 >$
3	$< 6, 3, 9, 11, 10, 12, 1, 2, 8, 5, 7, 4 >$
4	$< 11, 6, 10, 8, 3, 9, 5, 12, 7, 4, 1, 2 >$
5	$< 6, 3, 1, 2, 11, 10, 12, 9, 5, 8, 7, 4 >$

The QoS of a worker i to a task j is evaluated based on the models in [14] and [27], where it reflects the probability with which the worker is likely to perform the task. This depends on the reputation, the confidence, and the traveling time of the worker. Hence, at a given round r, $QoS_{i,r}^j$ is computed as in (3). The $QoS_{i,r}^j$ will be a score between 0 and 1 for the task considered.

$$QoS_{i,r}^j = R_i^W \times C_{i,r}^j \times \tau_{i,r}^j \quad (3)$$

The reputation of a worker, R_i^W, is defined as the ratio of successfully completed tasks to the number of tasks he or she has been previously assigned. $C_{i,r}^j$ denotes the confidence of the worker attempting the task at round r. The definition of confidence depends on the application of the MCS system. In this work, the confidence is represented as the degree at which the worker's preferences are met, i.e. the earlier the task is in the worker's *preference list*, the higher his or her confidence to fulfill that task. Hence, $C_{i,r}^j$ is evaluated as:

$$C_{i,r}^j = 1/rank_{i,r}(t_j) \quad (4)$$

where $rank_{i,r}$ is the rank of worker i at round r. $\tau_{i,r}^{j}$ is a decreasing function with respect to time; which reduces the QoS provided to the task by the worker when the time needed for a worker to finish the task increases. It is modeled based on [14] and is computed as:

$$\tau_{i,r}^{j} = 1 - \max(0, \min[\log_{TC_j}(T_{i,r}^{j}), 1]) \tag{5}$$

where TC_j is the time constraint of the task and $T_{i,r}^{j}$ is the time needed, in seconds, for the worker to reach the task at round r. In this work, we assume that the sensing time is minimal. $T_{i,r}^{j}$ is calculated as the Euclidean distance between the user's coordinates to the task's coordinates over the average speed of the worker at a round r. Assuming the traffic condition is known, the average speed of the worker can be estimated. Since the traveling distance and time are directly related, the traveled distance is considered part of the QoS equation.

The user satisfaction is defined by how the order of the assigned task deviates from the rank in the user's preference list. $S_{i,r_i^j}^{j}$ is the normalized user satisfaction at the round where the assignment of the worker w_i to the task t_j is made ($(w_i, t_j) \in M$), denoted by r_i^j. It is evaluated as:

$$S_{i,r_i^j}^{j} = -\frac{|rank_{i,r_i^j}(t_j) - order_{i,r_i^j}(t_j)|}{n_i - 1} \tag{6}$$

where n_i is the total number of tasks in the user's preference list, $rank_{i,r_i^j}(t_j)$ is the rank of the task t_j in the *preference list* of worker w_i, and $order_{i,r_i^j}(t_j)$ is the actual order of the task t_j in the assignments of the worker w_j at round r_i^j.

The proposed selection algorithm is presented in Algorithm 1 and is detailed as follows:

Step 1) The dataset of the participants and the tasks are used as input. The dataset of the participants consists of their IDs, location, reputation, and traveling distance constraints, whereas, the dataset of the tasks consists of their IDs, location, minimum QoS requirement, minimum reputation requirement, maximum workers capacity, and time constraint.

Step 2) **Lines** 1 **to 3:** Calculate the distance between all the workers and tasks, and add the tasks that are at a distance less than the distance constraint from the worker's location to the set acceptable list of that worker, $A(w_i)$. Additionally, create the *preference list* for the worker by ranking the acceptable tasks in $A(w_i)$ in a strict ascending order of their distance from the worker, as listed in Table 3.

Step 3) **Lines** 4 **to 7:** Add the workers that meet the minimum reputation requirement to the set of acceptable list of the task, $A(t_j)$. Also, calculate $QoS_{i,r}^{j}$ by each worker for each task, and create the *preference list* for the task by ranking the acceptable workers in a strict descending order of $QoS_{i,r}^{j}$, as listed in Table 4.

Step 4) **Lines** 8 **to 20:** Each task proposes to their most preferred worker. Some workers may not get any proposals, while others may get one or more proposals. The workers that did not get any proposal will remain unpaired, and the workers who got only one proposal from one task will be paired with that task, only if it is on their *preference list*. On the other hand, the workers that got multiple proposals from multiple tasks will be assigned to the task with higher precedence in their *preference list*. In the example, tasks 1, 2, 3, and 5 propose to W6, while task 4 proposes to W11. Since W6 and W11 received one or more proposals, W6 is to be paired with T5 and W11 is to be paired with T4, which are their most preferred tasks. The other workers do not get any proposals at this stage.

Each task then proposes to its next preferred worker. A task may propose to a worker who is already paired with another

Algorithm 1 Gale-Shapley Matching Game Selection Algorithm

Input: set of tasks (T), set of participants (W)
Output: Assignment set for all tasks and participants (M)
Let n_t be the number of tasks available, n_w be the number of participants available, and let $M = \emptyset$
1: Calculate distance d_i^j between every worker-task pair
2: **if** $d_i^j < D_i^W$ **then** add t_j to $A(w_i)$
3: *worker_pref* (w_i) = sort tasks in ascending order of d_i^j
4: **if** $R_i^j > R_j^T$ **then** add w_i to $A(t_j)$
5: $r = 1$
6: Calculate $QoS_{i,r}^{t}$ based on (3) for every worker-task pair
7: *task_pref* (t_j) = sort workers in descending order of $QoS_{i,r}^{j}$
8: **while** ($r \leq n_t$) & ($\exists$ tasks that still has unfulfilled capacity) **do**
9: initialize all $t \in T$ and $w \in W$ to free
10: **while** $\exists$ free t that still has w to propose to **do**
11: t proposes to w
12: **if** w is free **then**
13: $M := M \bigcup (w, t)$
14: **else if** $\exists$ task t' such as $(w, t') \in M$ **then**
15: **if** w prefers t to t' **then**
16: $M := M \bigcup (w, t)$
17: $M := M - (w, t')$
18: **end if**
19: **end if**
20: **end while**
21: **for** $i = 1$ **to** n_w **do**
22: **if** $(w_i, t) \in M$ **then**
23: update worker's location
24: remove the paired t and w_i from $A(w_i)$ and $A(t)$, resp.
25: update *worker_pref* (w_i)
26: **end if**
27: **end for**
28: **for** $j = 1$ **to** n_t **do**
29: update *task_pref* (t_j)
30: **end for**
31: $r = r + 1$
32: **end**

task, in this case, if the worker prefers the current proposal more than the former, a new assignment is made based on this proposal and the previous assignment for the worker is undone. This also frees a space in the capacity of the formerly paired task. For example, W6 will not accept any other proposals because he or she is already paired with their most preferred task. However, W11 will keep accepting other proposals that precede T4 in the *preference list*. This step continues until all the tasks have proposed to the workers in their *preference lists*.

Step 5) **Lines 21 to 32:** So far, multiple workers can perform the same task, but a worker can only perform one task, i.e. one-many matching. To adapt the algorithm to allow multitasking, i.e. many-many matching, the selection process is divided into time slots or rounds. Steps 2-4 represent one round. For the second round, the locations of the workers that were assigned in the previous iteration are updated to the location of the currently assigned tasks, and the paired tasks and workers are removed from each others' set of acceptable lists to avoid duplicated assignments. This causes changes in the *preference list* for the workers leading into changes in the *preference list* of the tasks, due to the changes in the QoS of the tasks.

Step 6) Steps 1-5 are repeated until the number of iterations exceeds the number of tasks available or all the tasks have fulfilled their available capacities. Finally, the output is all the stable matching between each worker and the assigned tasks where workers are requested to perform those tasks in the order of the rounds. The $QoS^j_{i,r}$ provided to a task by an assigned worker is calculated using (3), where confidence is calculated based on the updated *preference list* for the worker. It is worth mentioning that the aforementioned algorithm works even when the number of available workers, n_w, is less than the number of available tasks, n_t.

The final QoS^j for the task j is calculated using the collective $QoS^j_{i,r}$ for its assigned workers, g_j. This is computed as the probability of at least one successful reading from the assigned workers. QoS^j must be greater than or equal to Q^T_j and it is evaluated as:

$$QoS^j = 1 - \prod_{i=1}^{g_j}[1 - QoS^j_{i,r^j_i}] \tag{7}$$

The total QoS (TQ), the total confidence (TC), and the total user satisfaction (TS) achieved by the workers selected for all the available tasks are assessed. These are computed as given in (8), (9), and (10), respectively.

$$TQ = \sum_{j=1}^{n_t} QoS^j \tag{8}$$

$$TC = \sum_{j=1}^{n_t}\sum_{i=1}^{g_j} C^j_{i,r^j_i} \tag{9}$$

$$TS = 100 + \sum_{j=1}^{n_t}\sum_{i=1}^{g_j} S^j_{i,r^j_i} \tag{10}$$

where n_t is the number of available tasks and 100 is the highest total satisfaction score, which occurs when the rank of the task is equal to the order of the task for all the workers, i.e. S^j_{i,r^j_i} is zero.

The time complexity for the Gale-Shapley matching game selection (GSMS) algorithm is $O(n_t^2\, n_w) + O(n_t n_w \log(n_w))$. $O(n_t^2\, n_w))$ denotes the complexity of the many-many matching algorithm, and $O(n_t n_w \log(n_w))$ is the complexity of sorting the preferences of the tasks.

Finally, it is worth mentioning that there are two possible conflicting scenarios during the selection.

- *Scenario 1:* If two workers have the same QoS score relative to a task, then the one which is closer to the task is selected, provided that the reputation of the worker is higher than the minimum reputation.
- *Scenario 2:* If two workers are at the same distance from a task, then the worker with higher confidence and reputation scores will be selected.

IV. SIMULATION RESULTS

In this section, the proposed selection algorithm, GSMS, is compared with the Greedy-enhanced Genetic Algorithm based on Intentional movement (GGA-I) [15] in terms of the user satisfaction, confidence and the QoS achieved by the tasks. Both GGA-I and GSMS are multi-worker multitasking selection approaches. However, GGA-I selects workers solely based on the tasks' requirements. On the other hand, GSMS selects workers based on their preferences as well as the task's requirements. The GGA-I algorithm is adjusted to select workers based on highest expected QoS to the tasks, instead of the shortest distance to be traveled to the tasks. Finally, the collective QoS of the assigned workers for a task is computed. Table 5 summarizes the modifications that were made to the GGA-I approach to fairly compare it with GSMS.

TABLE 5. Modifications made to GGA-I approach to fairly compare with GSMS.

GGA-I	Modified GGA-I
Input: tasks and participants dataset. Let q be the maximum number of tasks per worker.	
1:Compute distance matrix among all tasks and workers.	1:Compute **QoS** matrix for all workers to tasks **using** (3) and **create** *preference lists*.
2:**While** $T \neq \emptyset$ and $W \neq \emptyset$ **do**	2:**While** $T \neq \emptyset$ and $W \neq \emptyset$ **do**
3: Assign the tuple $< t, w >$ with minimal distance in all tuples.	3: Assign the tuple $< t, w >$ with **maximal** QoS in all tuples.
4: Eliminate tasks that have full capacity and workers that have q tasks.	4: Eliminate tasks that have full capacity and workers that have q tasks.
5: Update distance matrix.	5: Update **QoS** matrix.
6: **end while**	6: **end while**
	7: **Compute QoS and user satisfaction based on the** ***prerefence lists*** **and eq. 7**.

In the simulations, the QoS, the confidence, and the user satisfaction for a task-worker pair are computed for both approaches, using (3), (4), and (6), respectively. The aggregated QoS for a task is computed using (7). Finally, the total

QoS, confidence, and user satisfaction for the available tasks are computed using (8), (9), and (10), respectively.

The aim of the comparison is to show how the proposed selection approach meets the preferences of the workers by improving user satisfaction and the tasks by improving the completion confidence and QoS. The approaches are first compared on a small scale using the illustrative example, then on a larger scale using a real life dataset.

A. ILLUSTRATIVE EXAMPLE RESULTS

This section provides the results for the illustrative example given in section III-A. In Table 6, the final assignment for every worker using the proposed GSMS approach is compared against the assignment using GGA-I, as discussed in section II-B and Table 5. Both GSMS and GGA-I consider the reputation, the confidence, and the traveling time by workers in the selection. However, unlike GGA-I, GSMS considers the workers' satisfaction by design. The total QoS of a task by all assigned workers is computed using (7).

TABLE 6. GSMS vs. GGA-I: Task allocations for each worker.

	W1	W2	W3
GSMS	$<5,2,4,1,3>$	$<5,2,4,1,3>$	$<5,2,4,1,3>$
GGA-I	$<5>$	$<5>$	$<5,2,4,1,3>$
	W4	**W5**	**W6**
GSMS	$<\emptyset>$	$<\emptyset>$	$<5,2,4,1,3>$
GGA-I	$<\emptyset>$	$<\emptyset>$	$<5,2,4,1,3>$
	W7	**W8**	**W9**
GSMS	$<\emptyset>$	$<\emptyset>$	$<\emptyset>$
GGA-I	$<\emptyset>$	$<\emptyset>$	$<3,1,2,4>$
	W10	**W11**	**W12**
GSMS	$<\emptyset>$	$<5,2,4,1,3>$	$<\emptyset>$
GGA-I	$<2,4,1,3>$	$<5,2,4,1,3>$	$<\emptyset>$

It can be seen that, using GSMS, all assigned workers got the first task in their *preference list*. While all workers preferred T5 as their first task, T5 was allocated to its first 5 preferred workers, workers 6, 3, 1, 2, and 11. Based on the assignments made, the total confidence, user satisfaction, and QoS for each task are compared for both approaches in Table 7.

TABLE 7. GSMS vs. GGA-I: Confidence, user satisfaction, and QoS.

	GSMS			GGA-I		
	Confidence	Satisfaction	QoS	Confidence	Satisfaction	QoS
T1	5.0	100	0.85	1.1	98.5	0.34
T2	5.0	100	0.90	2.0	99.25	0.57
T3	5.0	100	0.88	2.0	98.0	0.55
T4	5.0	100	0.94	1.32	98.75	0.50
T5	5.0	100	0.87	5.0	100	0.87
Avg.	5	100	0.89	2.28	98.9	0.57

This example illustrates how GSMS considers the satisfaction of the users in the selection by taking into account their preferences of the tasks. It is evident from the selection results, in Table 7, that the Gale-Shapley matching game is able to improve the average user satisfaction, confidence and QoS. Since the presented example is in small scale, the difference in the overall user satisfaction is marginal.

B. SIMULATION RESULTS USING REAL LIFE DATASETS

In this section, simulations are conducted using real-life dataset with large number of tasks and compared with the benchmark, GGA-I. For fair comparison, GSMS and GGA-I use the same population size, number of tasks and maximum group size.

1) DATASET

The Sarwat Foursquare dataset [28], [29] for a social networking application containing data about users, their social connections, check-ins and venues, is used in this work. The workers' records considered in the simulations are: *User ID, Longitude* and *Latitude*. The Stack Exchange Data Dump[1] real-life dataset is used to assign reputation to participants, whereas the *Distance Constraint* of the workers are randomly generated.

The tasks' dataset consists of the following: *Task ID, Longitude* and *Latitude, Minimum Reputation, Minimum QoS, Worker Capacity*, and *Time Constraint*. The tasks' records are randomly generated, and their locations are within an area of 5 $km \times 5$ km, which is defined as the AoI.

2) EVALUATION AND RESULTS

The proposed approach is evaluated and its performance is compared under the availability of i) different number of tasks, ii) different number of workers in the MCS system. In all simulations, the maximum number of workers performing each task is set to 10 workers and a set of 10 simulations for each number of tasks is taken to average out the results.

FIRST CASE: DIFFERENT NUMBER OF AVAILABLE TASKS

In this case, 600 participants are selected from the dataset based on their locations, i.e. participants whose locations are within the defined AoI. In these simulations, the number of available tasks increases gradually from 10 to 200, and their locations are randomly generated within the AoI.

Fig. 2 demonstrates the comparison between GSMS and GGA-I for the total confidence achieved by all the available

[1] https://archive.org/details/stackexchange

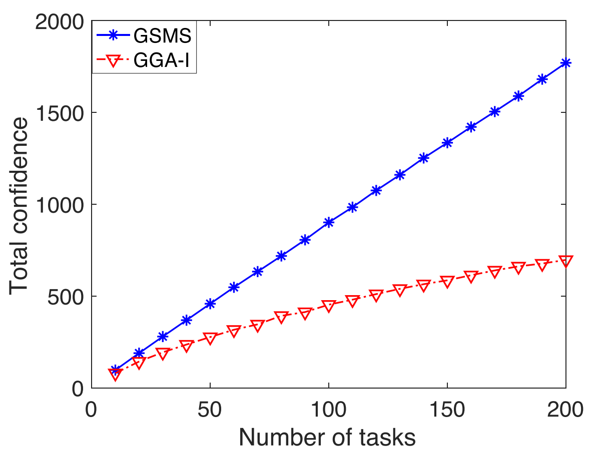

FIGURE 2. The total confidence achieved by tasks vs. the number of tasks available.

tasks. It can be seen that GSMS achieved up to 154% higher total confidence when compared with GGA-I. This indicates that GSMS is able to allocate workers, from the pool of available participants, to the tasks they have higher preference for, given the constraints, Q_j^T and R_j^T.

While the QoS of the tasks depend on the confidence achieved by tasks, it also depends on the traveling time and the reputation of the workers as given by (3). Fig. 3 demonstrates that GSMS improves the total QoS achieved by all tasks by up to 49% when compared with GGA-I.

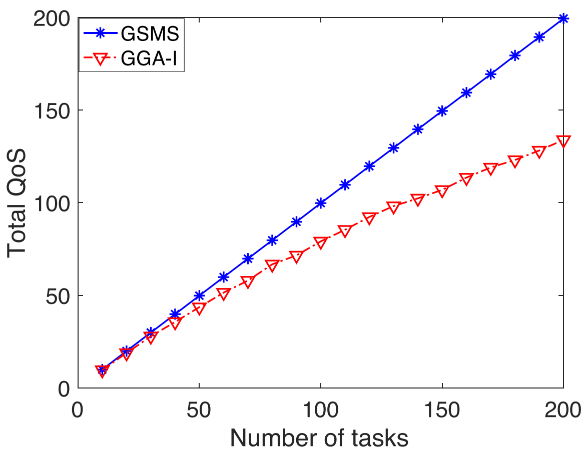

FIGURE 3. The total QoS achieved for all tasks vs. the number of tasks available.

Subsequently, the user satisfaction is compared for both GSMS and GGA-I in Fig. 4. GSMS improved the user satisfaction when compared with the benchmark, even with the increasing number of tasks. The reason for this is that Gale-Shapley considers the workers' preferences in selection, in terms of allocated tasks and their allocation order. The difference in user satisfaction between GSMS and GGA-I becomes more evident as the number of tasks increases. It can be seen that the user satisfaction is decreasing in GGA-I, as it becomes more challenging to fulfill the preferences of both workers and tasks when more tasks are available in an MCS.

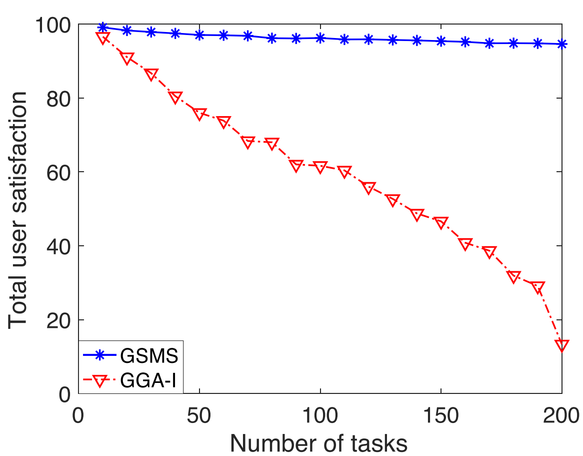

FIGURE 4. The total user satisfaction achieved by tasks vs. the number of tasks available.

However, this is not the case with GSMS, since it is constantly checking the preferences of the workers, which proves that the proposed approach is scalable and is able to satisfy the workers even in a packed AoI.

Finally, the total distance traveled by assigned workers is compared for both GSMS and GGA-I in Fig. 5. It can be concluded that while the proposed approach, GSMS, improved the confidence of the tasks and the overall QoS, the total distance traveled by the workers is comparable to the benchmark, GGA-I.

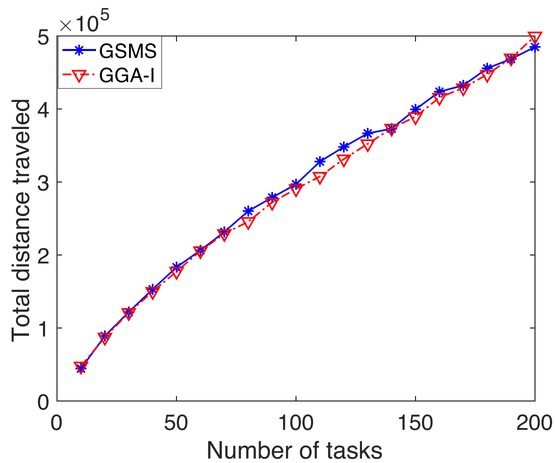

FIGURE 5. The total distance traveled by assigned workers vs. the number of tasks available.

SECOND CASE: DIFFERENT NUMBER OF AVAILABLE WORKERS

In this case, the number of participants is increased gradually from 100 to 600 participants, which are randomly selected from the dataset. In these simulations, 30 tasks are randomly generated in the AoI.

The total confidence of the completed tasks is compared for a varying number of available workers, while keeping the number of tasks fixed. Fig. 6 shows that GSMS increased the confidence of the tasks by up to 117% when compared with

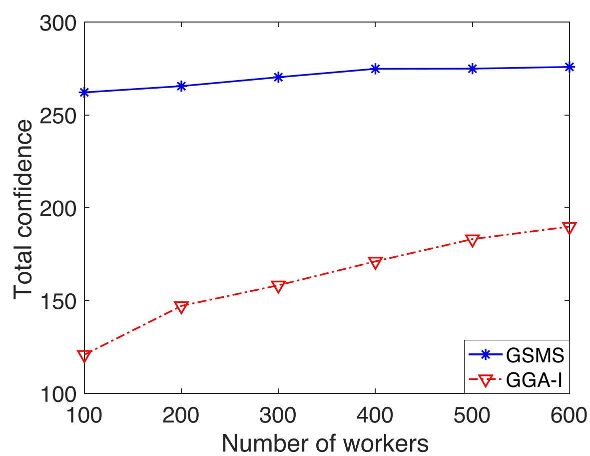

FIGURE 6. The total confidence achieved by tasks vs. the number of workers available.

the benchmark, GGA-I. In addition, the total QoS achieved by tasks is compared in Fig. 7, where the results show that GSMS increased the total QoS by up to 43% when compared with GGA-I. It is worth noting that in both Fig. 6 and Fig. 7, the total confidence and the total QoS for GSMS remains almost constant with increasing number of workers.

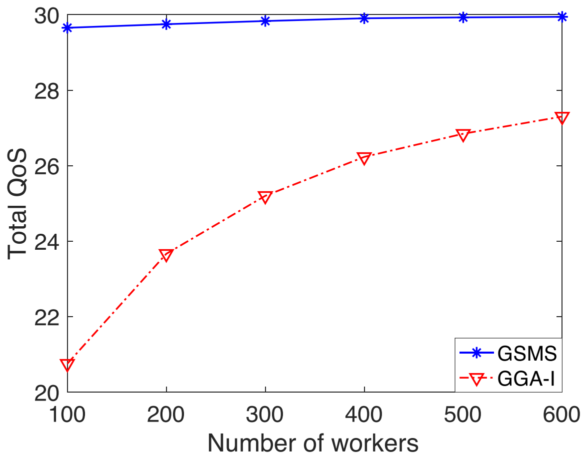

FIGURE 7. The total QoS achieved for all tasks vs. the number of workers available.

Fig. 8 shows the comparison of the total user satisfaction for both GSMS and GGA-I. It is evident that the total user satisfaction achieved through GSMS is always higher than that of GGA-I by a per centum reaching 42%. It can be seen that the total confidence, QoS, and user satisfaction are improving with increasing the number of workers. This is due to the fact that a large number of available workers increases the probability to allocate workers to their preferred tasks.

Finally, the total distance traveled by workers for the 30 tasks is compared for different number of workers available in Fig. 9. Results show that GGA-I improves the traveling distance by a maximum of 4%. This means that the improvement in the QoS and user satisfaction for GSMS is mainly caused by the improvement in confidence at the expense of a mere 4% increase in traveled distance. It is also

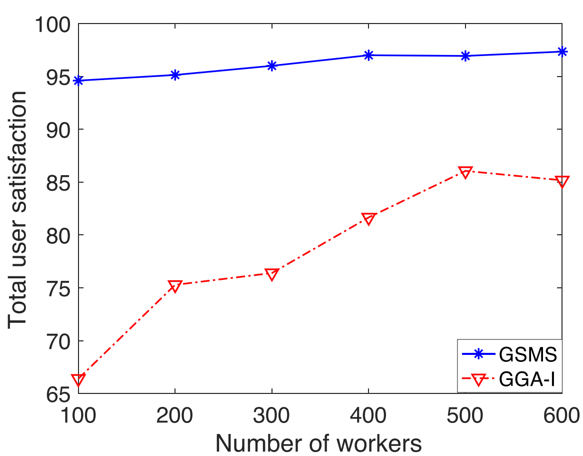

FIGURE 8. The total user satisfaction achieved by tasks vs. the number of workers available.

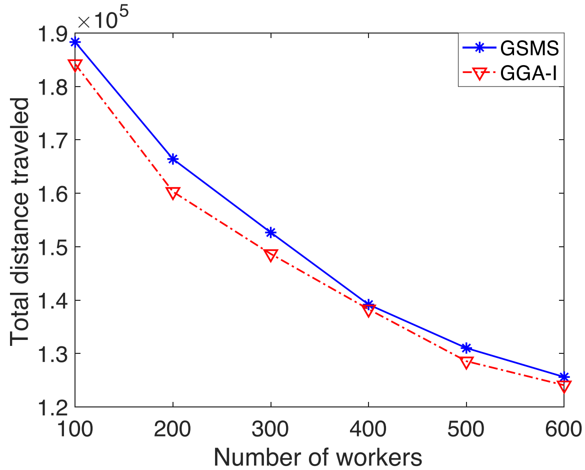

FIGURE 9. The total distance traveled by assigned workers vs. the number of workers available.

clear that as the number of workers increases, the total distance traveled to complete the tasks decreases. This is because the availability of more workers increases the probability of having workers close to the tasks.

V. CONCLUSION

A novel approach, Gale-Shapley Matching Game Selection (GSMS), which is based on game theory, is proposed to solve the multi-worker multi-tasking allocation problem in mobile crowd sourcing. The proposed model adapts the college admission problem by Gale-Shapley to allocate workers to multiple tasks based on both the workers' preferences and the tasks' preferences. Simulations using real dataset demonstrate that our approach outperforms the existing benchmark in terms of achieved confidence level, QoS, and user satisfaction. For instance, for different number of tasks available, GSMS increases the confidence of the tasks by a per centum reaching 154%, the QoS by up to 49%, and the total user satisfaction by up to 6 times, while having comparable results for the traveled distance, when compared to the benchmark. Additionally, similar trend was seen for varying number of workers where the total confidence of the tasks increased by a percentage as high as 117%, the total QoS of the tasks by up to 43%, and the total user satisfaction by up to 42%, at the expense of only marginal increase in the traveled distance. Overall, GSMS showed higher efficiency and scalability for different numbers of available tasks and workers, thus showing the efficacy of the proposed model in multi-worker multi-task MCS applications.

REFERENCES

[1] J. Pedersen *et al.*, "Conceptual foundations of crowdsourcing: A review of IS research," in *Proc. 46th Hawaii Int. Conf. Syst. Sci. (HICSS)*, Jan. 2013, pp. 579–588.

[2] R. Zhou, Z. Li, and C. Wu, "A truthful online mechanism for location-aware tasks in mobile crowd sensing," *IEEE Trans. Mobile Comput.*, vol. 17, no. 8, pp. 1737–1749, Aug. 2018.

[3] J. Xu, Z. Rao, L. Xu, D. Yang, and T. Li, "Mobile crowd sensing via online communities: Incentive mechanisms for multiple cooperative tasks," in *Proc. IEEE 14th Int. Conf. Mobile Ad Hoc Sensor Syst. (MASS)*, Oct. 2017, pp. 171–179.

[4] S. Luo, Y. Sun, Z. Wen, and Y. Ji, "C2: Truthful incentive mechanism for multiple cooperative tasks in mobile cloud," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–6.

[5] G. Merlino, S. Arkoulis, S. Distefano, C. Papagianni, A. Puliafito, and S. Papavassiliou, "Mobile crowdsensing as a service: A platform for applications on top of sensing clouds," *Future Gener. Comput. Syst.*, vol. 56, pp. 623–639, Mar. 2016.

[6] V. R. Neto, D. S. Medeiros, and M. E. M. Campista, "Analysis of mobile user behavior in vehicular social networks," in *Proc. 7th Int. Conf. Netw. Future (NOF)*, Nov. 2016, pp. 1–5.

[7] S. Lin, J. Zhang, and L. Ying, "Waze-inspired spectrum discovery via smartphone sensing data fusion," in *Proc. 16th Int. Symp. Modeling Optim. Mobile, Ad Hoc, Wireless Netw. (WiOpt)*, May 2018, pp. 1–8.

[8] *Free Driving Directions, Traffic Reports and GPS Navigation App by Waze*. Accessed: Sep. 28, 2018. [Online]. Available: https://www.waze.com/

[9] *Quantify the World With Placemeter*. Accessed: Sep. 28, 2018. [Online]. Available: http://www.placemeter.com/solutions/smart-cities

[10] K. Ota, M. Dong, J. Gui, and A. Liu, "QUOIN: Incentive mechanisms for crowd sensing networks," *IEEE Netw.*, vol. 32, no. 2, pp. 114–119, Mar./Apr. 2018.

[11] P.-Y. Chen, C.-W. Lien, F.-J. Chu, P.-S. Ting, and S.-M. Cheng, "Supervised collective classification for crowdsourcing," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2015, pp. 1–6.

[12] J. Surowiecki *et al.*, "The wisdom of crowds," *Amer. J. Phys.*, vol. 75, no. 2, pp. 190–192, 2007.

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

[14] R. Estrada, R. Mizouni, H. Otrok, A. Ouali, and J. Bentahar, "A crowd-sensing framework for allocation of time-constrained and location-based tasks," *IEEE Trans. Services Comput.*, to be published, doi: 10.1109/TSC.2017.2725835.

[15] B. Guo, Y. Liu, W. Wu, Z. Yu, and Q. Han, "ActiveCrowd: A framework for optimized multitask allocation in mobile crowdsensing systems," *IEEE Trans. Human-Mach. Syst.*, vol. 47, no. 3, pp. 392–403, Jun. 2017.

[16] L. Pu, X. Chen, J. Xu, and X. Fu, "Crowd foraging: A QoS-oriented self-organized mobile crowdsourcing framework over opportunistic networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 4, pp. 848–862, Apr. 2017.

[17] J. An, X. Gui, Z. Wang, J. Yang, and X. He, "A crowdsourcing assignment model based on mobile crowd sensing in the Internet of Things," *IEEE Internet Things J.*, vol. 2, no. 5, pp. 358–369, Oct. 2015.

[18] X. Zhu, J. An, M. Yang, L. Xiang, Q. Yang, and X. Gui, "A fair incentive mechanism for crowdsourcing in crowd sensing," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1364–1372, Dec. 2016.

[19] R. Azzam, R. Mizouni, H. Otrok, S. Singh, and A. Ouali, "A stability-based group recruitment system for continuous mobile crowd sensing," *Comput. Commun.*, vol. 119, pp. 1–14, Apr. 2018.

[20] R. Azzam, R. Mizouni, H. Otrok, A. Ouali, and S. Singh, "GRS: A group-based recruitment system for mobile crowd sensing," *J. Netw. Comput. Appl.*, vol. 72, pp. 38–50, Sep. 2016.

[21] D. F. Manlove, *Algorithmics of Matching Under Preferences*, vol. 2. Singapore: World Scientific, 2013.

[22] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *Amer. Math. Monthly*, vol. 69, no. 1, pp. 9–15, Jan. 1962.

[23] D. Gusfield and R. W. Irving, *The Stable Marriage Problem: Structure and Algorithms*. Cambridge, MA, USA: MIT Press, 1989.

[24] Y. Gu, Y. Zhang, M. Pan, and Z. Han, "Student admission matching based content-cache allocation," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2015, pp. 2179–2184.

[25] Y. Liu, B. Guo, Y. Wang, W. Wu, Z. Yu, and D. Zhang, "TaskMe: Multi-task allocation in mobile crowd sensing," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2016, pp. 403–414.

[26] J. Wang *et al.*, "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.

[27] H. Yu, C. Miao, Z. Shen, and C. Leung, "Quality and budget aware task allocation for spatial crowdsourcing," in *Proc. Int. Conf. Auton. Agents Multiagent Syst.*, 2015, pp. 1689–1690.

[28] J. J. Levandoski, M. Sarwat, A. Eldawy, and M. F. Mokbel, "LARS: A location-aware recommender system," in *Proc. IEEE 28th Int. Conf. Data Eng. (ICDE)*, Apr. 2012, pp. 450–461.

[29] M. Sarwat, J. J. Levandoski, A. Eldawy, and M. F. Mokbel, "LARS*: An efficient and scalable location-aware recommender system," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 6, pp. 1384–1399, Jun. 2014.

MENATALLA ABOUOUF received the B.Sc. and M.Sc. degrees in electrical and computer engineering from the Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates, where she is currently a Research Engineer with the Center on Cyber-Physical Systems. Her research interests include crowd sourcing, semiconductor devices, and democratization of IoT.

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

HADI OTROK received the Ph.D. degree in electrical and computer engineering from Concordia University, Montreal, Canada. He is currently an Associate Professor with the Electrical and Computer Engineering Department, Khalifa University, an Affiliate Associate Professor with the Concordia Institute for Information Systems Engineering, Concordia University, and an Affiliate Associate Professor with the Electrical Department, École de Technologie Supérieure, Montreal. He is an expert in the domain of computer and network security, web services, ad hoc networks, application of game theory, and cloud security. He is a Senior Member of the IEEE. He has co-chaired several committees at various IEEE conferences. He is also an Associate Editor of the *Ad Hoc networks* (Elsevier), the IEEE Communications Letters, and *Wireless Communications and Mobile Computing* (Wiley).

RABEB MIZOUNI received the M.Sc. and Ph.D. degrees in electrical and computer engineering from Concordia University, Montreal, Canada, in 2007 and 2002, respectively. She is currently an Associate Professor of electrical and computer engineering with Khalifa University. She is interested in the deployment of context aware mobile applications, crowd sensing, software product line, and cloud computing.

ANIS OUALI received the B.Sc. degree in computer engineering from the L'Ecole Nationale des Sciences de l'Informatique, Tunisia, in 2000, the M.Sc. degree in computer science from the Université du Québec à Montréal, Canada, in 2004, and the Ph.D. degree with the Electrical and Computer Engineering Department, Concordia University, Montreal, Canada, in 2011. In 2010, he joined the Emirates ICT Innovation Center, where he is currently with the Network Optimization Team that focuses on solving network design-related problem. His research interests include P2P networks for video streaming, distributed computing, and content adaptation.