QGCrowd: Spatial Crowdsourcing Based on Qualification Analysis

Zekun Cai[1] Shuaiqi Cao[1] Tong Gou[1]
Ruiyuan Jiang[1] Peishi Yuan[1]

[1]University of Virginia, VA, USA
[1]{zc4hs, sc3ez, tg8be, rj4uf, py3nu}@virginia.edu

ABSTRACT

Crowdsourcing is a newly developed business term which refers to the process of obtaining needed services, ideas, or content from online communities. Any member of the crowd can complete an assigned task and be paid for their efforts. Meanwhile, with the popularity of mobile devices and improvement on wireless network quality, spatial crowdsourcing is rising as a new framework extending crowdsourcing from digital domain to the physical world. It involves the spatial information of tasks, and requires workers to complete tasks by traveling to specific locations. In this paper, we take expertise of workers into consideration and introduce an optimized method to complete the maximum tasks assignment while minimizing the cost for workers. The cost is defined according to the travel distance and worker's expertise on certain tasks. We build a flow network to define the relation of tasks and workers, and convert the task assignment to a maximum flow problem. For evaluation, we use both real-world and synthetic data set as input and test the scalability of our platform, as well as the effect of two constraints on workers. The result proves that the QGCrowd has significantly improved the overall accuracy of task completion while maximizing task assignment.

Keywords: QGCrowd, spatial crowdsourcing, maximum flow problem, spatial task assignment, qualification analysis

1. INTRODUCTION

In recent years, with the widespread use of multi-sensor smart phones become and the huge improvement of wireless network quality, every individual with a smart phone is enabled to act as a worker who collects and shares many types of data efficiently. The data could be texts, pictures, videos or audios. This contributes to the development of a new subclass of crowdsourcing called spatial crowdsourcing, which extends crowdsourcing beyond digital domain and connects it to tasks in the physical world[5]. In general, the main idea of spatial crowdsourcing is to make a set of spatial tasks be completed by a set of workers in an approriate manner. And workers are required to physically go to these locations to complete those tasks.

In consideration of the huge number of smart phones nowadays, spatial crowdsourcing has large potential to be applied in many domains such as tourism, journalism and urban planning[1]. For example, if someone needs some photos of celebration activities in various locations in a city in New Year Eve, she can submit a query or request to a spatial crowdsourcing(SC) server near him instead of going to all the locations by herself. The server crowdsources the task among available workers close to the locations of these activities. And once workers complete the task with their smart phones, the photos will be sent to the server. Finally, the requester receives these photos from the server.

In this paper, we focus on the problem of spatial crowdsourcing based on qualification analysis. The object is to let tasks to be completed by workers with higer expertise while maximizing the total assignment of tasks. The whole working process is described as below. Workers who are willing to perform tasks report their physical locations to the spatial crowdsourcing server. And requesters submit queries including the information of tasks and their location to the server. Instead of being selected by workers themselves, each task is assigned to workers by the server based on distance from it to available workers and these workers' expertise. There are several assumptions in this paper. Each task can only be assigned to one worker and the worker will complete the task voluntarily. It means that once a worker is assigned a task, she will accept it and finish it unconditionally. Rewards for completing tasks are not considered in this paper.

Tasks belong to different categories and workers have different levels of expertise for different tasks. Besides acheiving maximum task assignment, we also need to consider the expertise of a worker for a task, which is different from previous approaches of spatial crowdsourcing. Minimum distance is not a primary goal in this paper which is another difference. For the SC server, how to assign tasks to workers becomes much more complex.

While there has been much research work on crowdsourcing, few researchers paid attention to spatial crowdsourcing until recent years[17, 24, 26]. And most of the existing research work on spatial crowdsourcing concentrates on participatory sensing which is a concept of communities contributing sensory information to form a body of knowledge. It focuses on a single task and the challenges related to the task[6]. However, there is not too much work for workers in participatory sensing. Then GeoCrowd is deigned and developed by researchers led by Leyla Kazemi at the

University of Southern California[14]. It is a proof-of-concept, multi-purpose and generic spatial crowdsourcing system. Compared with those research work before, GeoCrowd can deal with many tasks simultaneously and assign max tasks to workers with min cost.

However, the system is based on several assumptions. All tasks belong to the same category and all the workers are equally qualified to complete these tasks successfully[20]. It may result in some problems if GeoCrowd is applied in real world. A task may be assigned to a worker who does not have corresponding skills or enough expertise. Although the task is completed, it is even worse than that the task is not completed. Because the requester may get wrong and unsatisfactory results which is useless. Also, the worker's time is wasted. It is below our expectation that tasks are completed by workers with as high expertise as possible. For iCrowd[], it is designed for crowdsourcing in digital domain. It is not a subclass of spatial crowdsourcing so it only takes expertise of workers into consideration while location and distance are not included in the paper.

We designed a more practical spatial crowdsourcing framework based on GeoCrowd. The main idea of our solution is to combine the advantages and frameworks of Geocrowd and iCrowd. Besides this, We made some changes to the assumptions in GeoCrowd that expertise of workers is also taken into consideration and given high priority while assigning tasks to workers. The expertise is involved in calculating the cost for a worker to complete a task while GeoCrowd simply use distance as cost. At every instance of time, the SC server receives new tasks from requesters and queries from workers, so it only has local knowledge of tasks. So we can achieve overall maximum task assignment by achieving maximum task assignment at every instance of time. And the problem can be reducible to a maximum task minimum cost problem. Workers and tasks can be represented as nodes in a graph. In addition, we use the edges to represent the assignment of tasks. By the Lowest Cost Priority(LSP) strategy, the weight of a edge is decided and calculated by the expertise of the worker and the distance between the worker and the task. Then we run experiments on both real-world and synthetic datasets to compare the performance of our solution with the approach in GeoCrowd. We also explore the effect of the two constraints on workers, maxT which is the max number of tasks a worker can be assigned and Region R which is the area a worker is willing to travel.

In our approach, the sum of accuracy of task completion is ensured. Most assigned tasks will be completed by workers who have enough expertise for these tasks. Also, our approach will have nearly the same performance with GeoCrowd in achieving maximum task assignment.

In future, we will optimize the current version in several aspects. Firstly, each task can be assigned to k workers in order to improve the accuracy of task completion further. Secondly, we will try to the privacy of workers and make the framework more secure and trustworthy. Also, rewards for completing tasks could be taken into consideration so the framework can be more practical.

The rest of the paper is organized as follows. The study of related work is given in Section 2. Section 3 sets the preliminaries our design including taxonomy of spatial crowdsourcing, problem definition, solution architecture and data structures. Our proposed solution is detailed in Section 4. And it is experimentally evaluated in Section 5. Finally, Section 7 concludes the paper.

2. RELATED WORK

Crowdsourcing is a large and rising area of both research and application. There has been many papers stating the topic. And in spatial crowdsourcing, there's also some existing techniques. In the following section, some of the research work about either crowdsourcing or spatial crowdsourcing is talked about.

In GeoCrowd[14], the main object is to maximize task assignment while minimizing overall distance that workers travel. The authors use three algorithms and compare their performance. Taking entropy of workers into consideration while assigning tasks results in more task assignment than other two algorithms because more tasks in low entropy area would be completed. Assigning higher priorities to closer tasks can reduce overall travel cost of workers. But since the mode of assigning tasks in GeoCrowd is self-incentivised and the system only assigns a single worker for a task, which indicates that the worker is completely trusted for finishing the task well[24]. However in real life, this is not likely going to happen. So we introduce the verification of worker skills on certain tasks in order to improve the quality of tasks done.

On the other hand, in the crowdsourcing framework called iCrowd[10], the focus is the qualification of workers for tasks in digital domain. There is an algorithm for evaluating the worker's expertise in certain area of tasks based on results' quality of tasks workers have completed and a prediction result for which tasks the worker is well acquainted with. For newly registered worker in the system, the system would virtually generate a task for the worker to complete without informing the worker and then grade her expertise based on the result of the task. The difference between iCrowd and our platform is that the tasks is restricted online tasks. There are maybe online query tasks or Q/A tasks with standard answer, which doesn't require the worker to go to a specific location to complete a task.

And there are many other works have been done in crowdsourcing, with different focuses and features. For example, some of them do research on privacy protection for workers, some of them are studying different algorithms for deploying tasks and some are doing research on different modes of crowdsourcing.

Before the upcoming of platforms like GeoCrowd, the concept of spatial crowdsourcing is mentioned in [1]. This paper mainly gives a overall design of the location-based crowdsourcing and we can call it a predecessor of spatial crowdsourcing. However the focus of that paper is on some phenomenon observed from the user study. They found some common principles like people like to pull job rather than be pushed to a job and would choose the one near their current location.

In [22], the author focuses on the privacy protection for worker identity and location. As the spatial tasks restriction, when a worker is performing a task, the physical location is revealed to the SC-server by the mobile device he/she is carrying. This is unavoidable. Along with the contact information and identity, it could be a privacy intrusion. But in this paper, it presents a third-party mechanism, which collects the privacy information of workers and give a sanitized release to the SC-server. In this way the identity as well as location is not available at SC-server as well as requesters. The downside of this method is that it is involved with decomposition of the information and requires multiple SC-servers. So it may result in additional fees and would not be so efficient as for scalable purpose.

In [8], it is focusing on the parameter of expiration time while

assigning tasks. The mode in this paper is worker selected tasks(WST), so the worker could adjust the time to complete the tasks according to their own schedule. In addition, it also explores the differences between suitable algorithms for small number of tasks and large number of tasks. The dynamic-programming and branch-and-bound strategies could be applied to small number of tasks for accurate calculation while approximation and progressive algorithms could be used in large number of tasks.

In [15], it also presents a method for evaluating workers' accuracy when completing tasks. There is a probability-determined algorithm named reputation score, which gives the probability of the worker performing the job correctly. Also, the task is assigned to a number of workers whose aggregate reputation satisfies the threshold of the task.

In [9], the authors talk about the spatial crowdsourcing platform. However, it focuses on not only the task matching but also the task scheduling with the expiration time for each task. In the experiment part, however, the authors mention that the three-phase matching method does not scale very well.

And for online crowdsourcing, there's also some articles about how to ensure accuracy of the tasks completed. In [12], the authors provide an algorithm to minimize the price that requesters have to pay for tasks while maximizing the overall accuracy of task completion. This paper focuses on developing new algorithms to replace majority voting for deciding a correct answer of a certain problem, under the situation that many workers are assigned for the same task.

3. PRELIMINARIES

3.1 Taxonomy

In this section, we present a tree to classify the spatial crowdsourcing modes. We present different levels of classification. The overall figure would be presented as in Figure. 1. The first level of the tree would be classification based on people's motivation. The second level would be two modes of task publishing. And finally comes the two ways of task assignment.

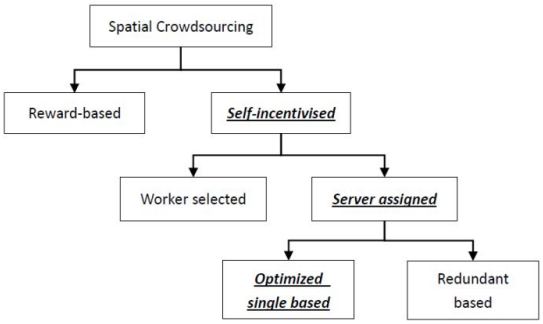


Figure 1: Taxonomy for spatial crowdsourcing. The focus of this paper is underlined and presented in italic.

Motivation based classification
Involved with the process of assigning different tasks to workers, the spatial crowdsourcing mode could be divided according to the motivation of the workers: Reward-based and Self-incentivised.

Reward-based spatial crowdsourcing:
With reward-based spatial crowdsourcing, the tasks are assigned with a price and for completing each task, the worker will receive a certain amount of money as a reward. As an example in [25], the workers would receive a small amount of reward after completing and sharing a sensing task.
Self-incentivised spatial crowdsourcing:
This class of spatial crowdsourcing refers to the tasks which workers complete voluntarily, without any actual reward. The tasks inside this class maybe easy to perform or could be automatically completed by the device. As an example in [18, 11], the "workers' voluntarily report traffic events (e.g., accidents and jams) by leveraging their sensor-equipped mobile devices. Here in this paper, we are doing research under this field.

Task publishing classification
For the process of assigning tasks to the workers, we can classify the spatial crowdsourcing to two modes: Worker selected tasks (WST) and Server Assigned Tasks (SAT).

Worker selected tasks(WST):
In this mode, the tasks are published by the SC (spatial crowdsourcing) server and workers are allowed to select the tasks they want to perform and then go to the location to perform the tasks. With this mode, the workers don't need to reveal their physical location to the server, which works as a privacy protection for the workers. But the drawback of this method is also pretty obvious. The overall assignments completed could be highly decreased because individuals fail to get the whole picture of all the tasks as well as the workers' location. Also, some tasks located at rather far locations have the probability of never being completed. As an example in [1], the workers are available for all the tasks and choose the ones in their neighborhood to complete.

Server assigned tasks(SAT):
Under this mode, the work of assigning tasks is completed by the SC server. The tasks are no longer revealed to the workers. On the contrast, volunteer workers have to report their physical locations to the SC server and after the combining the information of tasks location and worker location, the SC server assign the workers with the nearby tasks. In this mode, the workers could not reject the tasks assigned by the server. This mode does a good job in global optimization, for it is maximizing the overall assignments completed. As an example in [13], the paper proposes a framework for small campaigns. We are using this mode in this paper for its good performance in the big picture.

Task completion classification
After categorizing spatial crowdsourcing in terms of motivation and task publishing, we also need some classification for the process of completing tasks. Here we divide the modes by terms of how to verify the validity of the tasks: Single-based task assignment and Redundant-based task assignment.

Single-based task assignment:
In this mode, we assign one task to only one worker. And the worker is completely trusted for completing the task correctly. Based on the previous single-based task assignment, we add the process of accuracy check to ensure that the workers are qualified enough for this certain task. After applying this, the quality of completed tasks could be significantly improved. Some examples of the original single-based task assignment mode are [13, 4]. In our paper, we are using a much more optimized version of this mode.

Redundant-based task assignment:
In this mode, we are considering the completion of a single task

is involved with multiple workers. Especially for online query platforms with straight answers, the one gets the most votes would be used as the final answer. One famous example of the non-spatial redundant-based crowdsourcing is [16].

3.2 Terminology and definition

In this section, we define a set of terminologies in self-incentivised single-based spatial crowdsourcing with SAT mode. First, we define a ***spatial task*** formally:

DEFINITION 1. (SPATIAL TASK) *A task, which is denoted by t, is posted by an employer in a specific location waiting for online worker(s) to finish it. It is of the form <*lat, lng, max_w, t, tp*> where "lat" (latitude) and "lng"(longitude) is the location of that task. "*max_w*" represents for the maximum number of workers that can take this job. "t" is the time that the task generated at and "tp" is the type of the task.*

Note that if a task t is within a worker's working range, and then this task will be the candidate task that may be assigned to that worker. And for this framework, we set the max_w one because we assume that each task can only be accomplished by one worker. Besides, we assume that there are only 4 types of tasks, and one task can only be classified into one type (i.e. we have four types of tasks - "A","B","C" and "D", and for each single task, the type attribute can only be one of the four). In the following, we will introduce the definition of ***worker***.

DEFINITION 2. (WORKER) *A worker, which is denoted by* w*, is a people who is willing to volunteer to accept spatial task and to accomplish them. It is of the form <*ID*, lat,lng,* max_t*,* max_range*,* acc_A*,* acc_B*,* acc_C*,* acc_D*> where ID is the unique identifier of each worker. The "lat" and "lng", which form the exact location of* w*, and* max_t *is the maximum number of tasks that the worker willing to accept. The* max_range *can be utilized together to define the working range MBR(maximum boundary range). And the* acc_A*,* acc_B*,* acc_C *and* acc_D *are the accuracy for each type of task.*

Once we define the MBR of each worker, we can scan every task for each single worker to see if the location of the task is located in the MBR. If so, the task will be the candidate task for that worker. Under this circumstance, we now provide the definition of ***spatial task assign query***.

DEFINITION 3. (SPATIAL TASK ASSIGN QUERY) *A* ***spatial task assign query****, which is of the form <TaskSet, WorkerSet>, is a query q requiring an assignment of tasks of a specific time instance I for the input worker set. TaskSet is the total task set in time instance I, and WorkerSet is the total worker set.*

At a certain time instance, we obtain this query including available workers and unexpired tasks, and then according to the cost of each possible assignment of each task, we choose the minimum one and obtain the overall assignment with lowest total cost and highest total accuracy. Notice that we assume that we have fixed worker set and for every worker, he or she can accomplish the task within one time instance. That is, at each time instance, we have the same worker set. For fulfilling this requirement, we now define the ***Conditioned Maximum Task Assignment (CMTA)***.

DEFINITION 4. (CMTA) *Given a query q, let* T *be the number of assigned tasks for workers at time instance* I*. Let* C *be the cost of each assignment of every single task. Let* A *be the accuracy of each assignment of every single task. The maximum task assignment with minimum cost method is to assign the tasks with the total number of tasks (i.e.,*$|T|$*) is maximized and the total cost of tasks (i.e.,*$|C|$*) is minimized, together with the total accuracy (i.e.,*$|A|$*) is maximized.*

Note that this is an adaptive version of Maximum Task Assignment (MTA) method and has several conditions. Traditional MTA just take the distance as the cost and assign tasks to workers based on this factor. Once the accuracy is calculated, the total accuracy of one total assignment will not be optimized. Our new CMTA take the accuracy into consideration, and at the end we will obtain total accuracy that has the highest total accuracy.

3.3 Problem Definition

In this paper, we define a new framework QGCrowd. QGCrowd conbines the ideas from iCrowd and GeoCorwd. In the new framework, we define W as the worker set and T as the task set. In addition, we define w_i as a certain worker and t_i as a certain task. Besides the normal input, we add a factor $e_i$representing the expertise of each worker in different fields.

With two input datasets (i.e. workers and tasks), we can obtain two sets of data and then generate a graph G having all workers and tasks as nodes and edges. We also add two nodes and source and sink point. Edges from the source node to all the workers has a capacity that represents the maximum tasks that a worker is willing to accept. Edges from all the tasks to sink node represents for the required number of workers for that task. However, the cost of the edges from workers to tasks is changed. In our framework, the cost is weighted by two factors distance d and accuracy a instead of only the d.

Then the task assignment can be recalculated utilizing CMTA. After calculating the graph G, we obtain a new graph G' which has the maximum assignment (i.e.max_ass) with the highest summation of accuracies (i.e.$\sum_{n=1}^{i} e_i$).

3.4 Solution Architecture

As mentioned before, we basically combine the iCrowd and GeoCrowd frameworks together, and take the two factors together to calculate the new weight, and then assign tasks to workers with optimized maximum tasks assignment. Below is our basic framework.

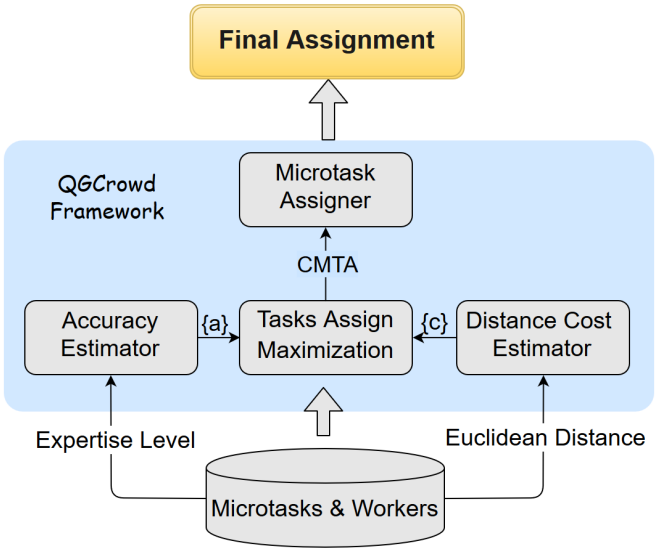


Figure 2: Basic Framework

From Figure.2, we can easily understand the whole processing procedure of QGCrowd framework. First, we will choose workers

and tasks based on the workers' MBR. This step removes the workers that can never be assigned tasks to and the tasks that can never be assigned to a worker. After this procedure, there will be a new set of workers and tasks. Then, we construct our graph based on datasets after the filtering procedure. In the end our new framework takes ***a*** and ***c*** as two parameter and utilize CMTA to calculate the final assignment.

3.5 Data Structures

Task Assignment Flow Structure

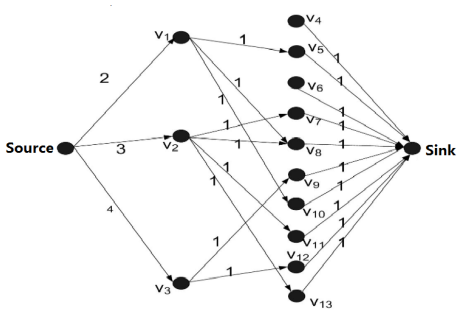


Figure 3: Task Assignment Flow Graph, G=(V,E)

Figure.3 is an example graph of task assignment. The edges between worker set(v_1, v_4, v_3) and task set(v_4 - v_{13}) represent the assignment of tasks, and the numbers stand for the weight of each edge. This problem can be regarded as a maximum flow problem.

4. METHODOLOGY

In this section, we first introduce the basic scenario of our framework. At first the requester will send spatial task query to the spatial crowdsourcing server including the start and the end time of the task, and the location of that task . After receiving the spatial crowdsourcing (SC) request from all the requesters, the SC server assign these queries to the available workers. In our framework, the SC-server will assign every spatial task to only one worker since we only focus on the single task assignment mode. A worker sends task inquiry to server to inform the server he is available. The tasks inquiry should include the location of the worker and the other constraints such as the max task which the worker is willing to take, and the maximum acceptable task region. Once the workers send their task inquiries, the server should assign to every worker a set of tasks which should satisfy their constraints. For the ideal case, all tasks could be assigned to all workers. However, this is not practical. Thus our goal is to maximize the overall assigned tasks[14].

In order to solve the MTA problem we proposed in the previous section, the SC-server should have a global knowledge of all the workers and task. However, the server is not able to be that clairvoyant, since it has to continuously distributed tasks in real time to workers without the prediction of the future due to the nature of worker and tasks. Therefore, the basic feasible idea is to optimize the task assignment at every instance of time. In the following, we propose two solutions to this problem. Our first approach tries to solve this problem by the reduce the situation to a model of maximum flow problem. Our second approach tries to take the expertise of workers and the travel cost of these workers into consideration and turns this CMTA problem into a minimum cost-maximum flow problem.

4.1 Basic Strategy for Maximum Task Assignment

At each instance of working time, the spatial crowdsourcing server holds a set of tasks from the requesters, and also a set of task inquiries from the workers. The server's target is to maximize the overall assignment by solving the maximum task assignment instance problem for every instance of time. This means this strategy only optimize the local performance which does not definitely lead to the globally optimal. Given a set of online workers $Wi = \{w1, w2, ...\}$, and a set of tasks $Ti = \{t1, t2, ...\}$ at time instance Si, the server assigns maximum number of tasks in Ti to workers in Wi for every instance Si[14]. We refer to this problem as MTA problem.

Since the worker need to physically move to the location to perform the task a "work region" R should be defined as a constraint to limit the acceptable working distance for particular worker. That will avoid "long distance" task assignment. Also, a maximum number of tasks $maxT$ is set for each worker. Thus, each worker would perform at most $maxT$ tasks, which should all be inside its work region R. The following figure shows a simple example of workers and tasks in this scenario.

To formalize and solve this problem, we transform this problem into a maximum flow problem. For time instance Si with $Wi = \{w1, w2, ...\}$ as the set of online workers and $Ti = \{t1, t2, ...\}$ as the set of available spatial tasks, let $G_i = (V, E)$ be the flow network in which V represent vertices and E represents set of edges. There are $W_i + T_i + 2$ vertices. We create a virtual source node and a virtual destination node to the Graph. The indices of these two nodes are setted respectively to $V_{W_i+T_i}$ and $V_{W_i+T_i} + 1$. Each worker and each task map to a vertex in set V. Workers represent the first i nodes in set V, and tasks represent the rest node in set V. The set E contains the $W_i + T_i + X$ edges. W_i stands for the number of edges between the dummy source node and workers. We set the capacity of these edges to be $maxT$ because every worker can only perform at maximum $maxT$ tasks and this number varies from worker to worker. T_i represents the edges between the tasks and the destination node. The capacity of these edges are set to be one, since every task can be assigned to only one worker. X means the number of edges between the worker set and the task set. For every worker w_j in W_i we connect this node with all the nodes in the task sets which satisfied the spatial constraints R_j.

With this model, this maximum task assignment instance problem can be reduced to the maximum flow problem. Figure 4 shows an example scenario. In this figure, w1 can only accept tasks within his spatial region (i.e., t2, t5, and t7),w2 can take task $t4, t5, t8, t10$ while $w3$ can take task $t6t9$. Figure 3 shows the corresponding flow network Graph $G(V, E)$ generated from the scenario in Figure 4. In this figure, the vertex mapped from w1 can transfer flow to three vertices mapped from those tasks (i.e., v5, v8, and v10). Moreover, w1 is only willing to accept two tasks since $maxT1 = 2$. Therefore, the capacity of the edge $(src, v1)$ is 2. Finally, the capacity of all the edges connecting the vertices mapped from spatial tasks (i.e., v4..v13) to the destination vertex dest are 1, since every spatial task is to be assigned to one worker[14]. By reducing the MTA problem into maximum flow problem, the algorithm which computes the maximum flow in the network can be used to solve the MTA problem. One of the well-known method to solve this problem is Ford-Fulkerson algorithm.

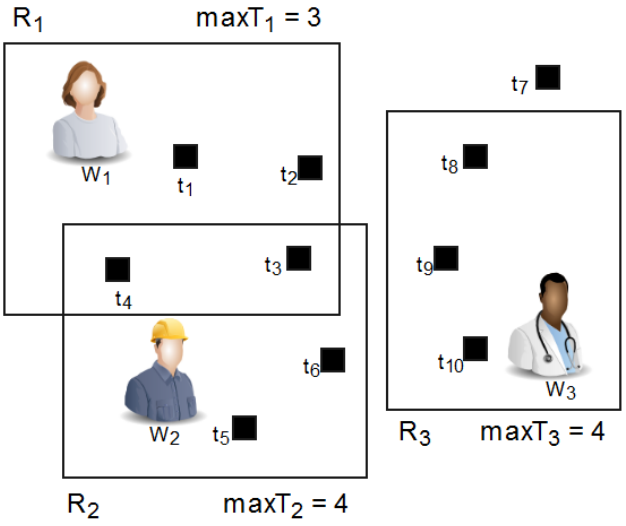


Figure 4: An example of Workers and Tasks

The idea behind this algorithm is that it first find a residual flow network graph based on the current graph. Then it keep finding the Augmenting path based on the updated residual flow graph utill no augmenting path can be found. The Augmenting Path can be found by using the Breadth Frist Search (BFS) or Depth First Search(DFS). In our implementation, we use the former one, which is known as Edmonds−Karp algorithm. It is a variation of the Ford-Fulkerson algorithm with guranteed termination and a runtime independent of the maxium flow value which runs in $O(VE^2)$ time. The augmenting path it found must be the shortest path with available capacity. Each augmenting path can be found in $O(E)$ time. After that, maximum flow can be achieved. Based on this maximum flow model, the Lowest Cost Strategy is applied, which consider the travel distance and worker's expertise as factors of a "cost" system to optimize the task assignment. The details of Lowest Cost Strategy will be discussed in sectione 4.2

4.2 Lowest Cost Priority Strategy

As for spatial crowdsourcing, the travel cost is critical because workers are required to move physically to the location and perform the task. Although, the assignment process prohibits workers from taking the task outside his spatial area, it does not assign these workers with the nearest task. Thus, we incorporate the travel cost of the workers into the process.

Meanwhile, we believe that workers' expertise and reputation should also be taken into consideration. So far, we assume that all tasks are of the same type and all workers are trustable and have the same ability. However, the truth is that every worker is different. And the motivation for considering expertise is intuitive, since workers are more likely to complete the tasks that they have higher expertise for. For instance, a photographer will be a better candidate to perform the task of taking a high quality picture than a chef. Our goal in this strategy is to maximize the overall task assignment at every time instance while minimizing the travel cost of the workers and maximizing the expertise matching.

So we combine these two factors to introduce a hybrid of cost and formally define this strategy. Before we introduce the concept of cost, we define the travel cost and the equation to combine worker's expertise and the distance between the worker and the tasks. We define the travel cost between a worker W and a spatial task t in terms of the Euclidean distance between them, and denoted by $d(w, t)$. The calculation of this Euclidan distance is by the unit of longtitude and latitude. Thus, the Euclidean distance may be normalized to simplify the calculation of the cost.

In order to take the expertise into consideration, we will relax the definition and the data structure of Spatial Task Query, Worker. For every Spatial Task, we will add an attribute called task type denoted by ty. As for Worker, we add a set of four different skills, denoted by $acc_A, acc_B, acc_C, acc_D$. Besides the skills, every worker has a corresponding expertise score for each skill and the score is represented by a decimal number between 0 and 10. Worker W with skill $acc_X \in ty$ means that the worker has the expertise to perform the task with type acc_X. So for every Task Inquiry it should also include the worker's expertise. Noted that every task can only has one type, while the worker can have many skills.

DEFINITION 5. (COST) *For every match of form $< w, t >$, we define a score value, which indicates how close the worker and task is and how well the worker w performs the task based on workers' expertise. For a given task, let D be the distance between the worker and task. Also let the S be the expertise score. We will also use two weight factors α and β to adjust the performance of this strategy. The score for a worker on a specific task can be computed as follows:*

$$Score(t) = \frac{1}{\alpha \cdot S} \times \beta D$$

This cost is a hybrid cost. Since our goal does not change, this problem can still be turned into the maximum task assignment problem. As we have proved that the maximum task assignment instance problems is reducible to the maximum flow problem. For this problem we prove that the minimum-cost task assignment is a minimum-cost max flow problem. Let $G_i = (V, E)$ be the flow network graph constructed for the maximum task assignment problem. For every task t_j , and worker w_i , let (w_i, t_j) be the edge between them. We associate to (w_i, t_j) the cost of t_{ij}.(i.e., $c(w_i, t_j) = Cost(t_{ij})$). Furthermore, the cost of other edges in E should be 0 by default. Thus, by finding the minimum-cost maximum flow in graph G_i, we have assigned the maximum number of tasks with the minimum cost. Also, in order to know the expertise mathcing score of every task, we associate worker's accuracy on this particular task to every edge between the worker set and the task set.

In the example of Figure 3, let $Cost(t_{ij})$ be the Cost for a worker to perform a specific task. Thus, for every edge between worker and the task will have a specific value. In order to find the maximum flow problem we can use the well-known Ford-Fulkerson or any other algorithm. After that, the score of the flow can be minimized by using the linear programming tech since all constraints are linear. Let $G_i = (V, E)$ be the flow network graph constructed, every edge $(u, v) \in E$ has flow $f(u, v) >= 0$, capacity $c(u, v) > 0$, and $s(u, v) >= 0$. We can use the Ford-Fulkerson algorithm to find the maximum flow F_{max}. The goal is to minimize the total cost of the flow, which can be calculated as follows:

$$\sum_{(u,v)\in E} f(u, v) \times s(u, v)$$

It has the following constraints which are defined for the maximum flow problem.

$$f(u, v) <= c(u, v)$$

$$f(u, v) = -f(u, v)$$

$$\sum_{w \in V} f(u,w) = 0 \quad iff \quad u \neq src \; or \; dst$$

$$\sum_{w \in V} f(src,w) = f_{max}$$

$$\sum_{w \in V} f(w,dst) = f_{max}$$

Since all constraints are linear, this problem can solved by linear programming algorithm in polynomial time. Thus, we solve the Maximum task assignment problem by computing the maximum flow in which the cost for every edge is defined by the cost of the tasks.

5. EXPERIMENTAL EVALUATION

We executed several experiments on both real-world and synthetic dataset to evaluate the performance of both Geocrowd algorithm and our spatial crowdsourcing task assignment algorithm based on qualification analysis. By comparing the average travel cost, average accuracy, total number of assigned tasks, the improvements of our algorithm on expertise matching are shown. In the following, we firstly introduce the generation of synthetic workload, and them discuss the experimental methodology. Finally, we present our experimental results.

5.1 Synthetic Workload Generation

This project aims to achieve the Spatial Crowdsourcing task Assignment. The spatial data is the most crucial part in this dataset. Since the public real-world datasets is limited and hard to be accessed, a synthetic workload generator is designed to produce dataset for this experimentation. In our project, two ways are chosen to generate the training data. Firstly, we evaluated the properties of some real-world geosocial dataset, and build the generator which supports synthetic random spatial data generation with certain distributions. Secondly, we directly sample spatial information, which includes latitude and longitude, as the spatial data of tasks and workers, from the Gowalla dataset. Based on the spatial information, we also applied the synthetic task's type and other information attributes to form the tasks and workers.

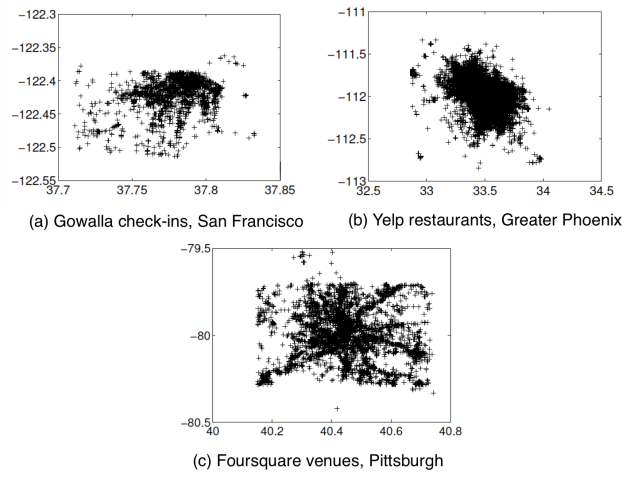


Figure 5: Distributions of points of interest in real-world datasets[21]

Spatial Distributions of Workers and Tasks

To explore the properties of real-world geosocial datasets, we studied the dataset collected by Gowalla, Yelp and Foursquare. The following figure shows the distributions of the data points in three datasets.

We treat the users in the three datasets are spatial workers, and checking in a spot is equivalent to completing a spatial task at that location. The distributions in Gowalla and Yelp are more close to Gaussian distribution, while the Foursquare data is close to the Uniform distribution. Therefore, our workload generator will apply the Mixture distribution, where half of the data points are sampled from Uniform and the other half from Gaussian, which can successfully simulate the downtown area than rural area. The following figure shows the distribution of our synthetic data.

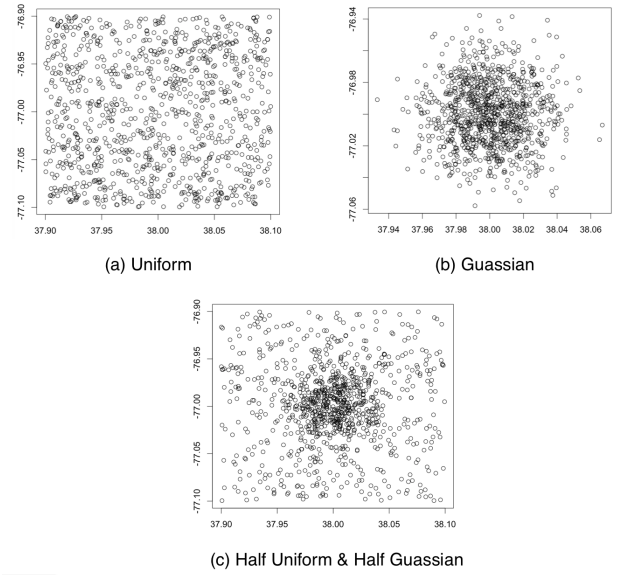


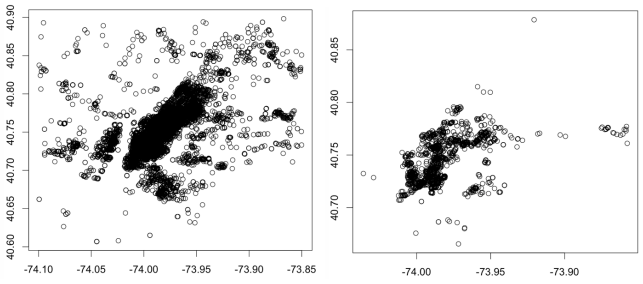
Figure 6: Distribution of Synthetic Data

Real World Spatial data of Workers and Tasks

Based on the studying of three real world dataset, we pick the Gowalla data [19] to generate the real world spatial information for workers and tasks. This Gowalla dataset record the check in spatial information of its user. By restrict the ranges of latitude and longitude, we can isolate the users from certain area. In a real spatial crowdsourcing system, the tasks are posted by actual users. Therefore, it is reasonable to use the check in location to simulate the task's location. In this project, we sample the Gowalla user's spatial data which are in New York and Los Angeles, and synthetize two training datasets. The New York dataset conations 16801 tasks and 1500 workers, which are located at the region of latitude N 40.6° to N 40.9°, and longitude W 73.85 °to W 74.1°. The Los Angeles dataset contains 26394 tasks and 2000 workers, which are located at the region of latitude N 33.5° to N 34.2°, and longitude W 117.8° to W 118.7°. The locations of workers and tasks in two datasets are shown as following. It can be seen that the distribution matches real population distribution well.

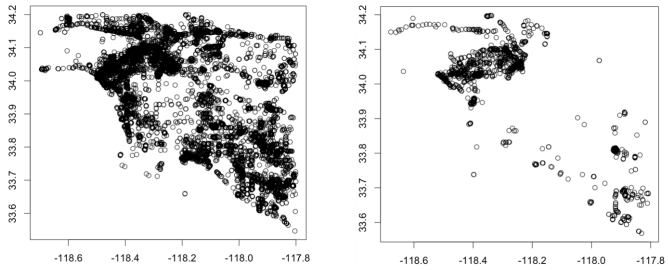
Task Types and Worker's Expertise

Since this project aims to improve the task assignment accuracy based on geocrowd frame, the matching between task's type and worker's expertise is important. We set a group of task types. For each task, we randomly assign a type to it for simulating the real crowdsourcing dataset. Each worker should have its expertise on some particular aspects or types of work. We use 0 to 10 scale to represent the worker's expertise, and the scores follow the Gaussian distribution.



(1) Tasks in New York (2) Workers in New York

Figure 7: Workers and Tasks in New York



(1) Tasks in Los Angeles (2) Workers in Los Angeles

Figure 8: Workers and Tasks in Los Angeles

5.2 Experimental Setup

For the synthetic data, it is from the designed work and task datasets generator. The worker dataset has position, acceptable working range, maximum tasks and proficiency as attributes. And the tasks dataset has position, required number of workers, duration time and type as attributes.

We have tried to find a better real-world data which should be compatible with our projects and assumptions. However, at present, we only have the datasets from Stanford Network Analysis Project[19] and Yelp[27]. Both of them lack some attributes that we need. Since, we are not able to find appropriate dataset from Internet, we modify the existing ones according to our experiment scenarios to obtain the datasets we need for the project evaluation.

Our approach is implemented in Java and we use a laptop with core-i7 processor and Window 10 operating system to run the experiments.

For all the experiments, we have following assumptions. Firstly, all workers perform tasks voluntarily without requiring any rewards. Secondly, we assume the experiment area is plain which means the same distance has the same travel cost. Thirdly, we assume that the expertise of a worker for a type of tasks can be represented by number from 1 to 10.

5.3 Performance evaluation

We performed three sets of experiments to evaluate the scalabitlity of our proposed solution by varying the number of spatial tasks. We also evaluted the influence of workers's constrains such as maximum region R and max-Task $max - T$ on the peformance of our solution. In all our experimentss, we varied the number of tasks from 5k to 20k, and we use the 10k as the default value for our task numbers. In addition, max-Task is set between 1 to 10 unless mentioned otherwise. The spatial region R is set between 0.05 to 0.1 of the entrie area. For the entrie experiment, the distance is calculated in the unit of longtitude and latitude and the maximum lontitude and latitude are set both to be 0.2 for both of the synthetic datasets and real datasets. The corresponding distance in the kilometer form is 17 kilometers. The spatial area is set to 17 square kilometers. We only conducted the first experiment on the real world set, for the rest of experiments we use the synthetic data. Whatsmore, for each of our experiments , we ran several cases and recorded the average of the resullts.

For the baseline, we choose the original GeoCrowd platform and compare the performance of QGCrowd and GeoCrowd in some parameters. As we are introducing the accuracy of task assignment based on the qualification of the workers, we also add the parameter of accuracy in GeoCrowd for comparison.

Besides the baseline comparison between QGCrowd and GeoCrowd, we also present 3 other sets of evaluation results for demonstrating other factors that may effect the performance of our platform.

	WorkerSum	TaskSum	Assigned TasksSum	DistanceSum	ExpertiseSum
Synthetic-1: NonAcc	1000	10000	1253	30.45	9856.51
Synthetic-1: WithAcc	1000	10000	1253	39.94	15442.24
Synthetic-2: NonAcc	1000	20000	2181	60.13	21392.26
Synthetic-2: WithAcc	1000	20000	2181	77.55	33054.57

(a) Results of Synthetic Datasets

	WorkerSum	TaskSum	Assigned TasksSum	DistanceSum	ExpertiseSum
New York: NonAcc	6896	16801	1700	41.37	15775.37
New York: WithAcc	6896	16801	1700	48.83	27699.28
Los Angeles: NonACC	2000	26394	1292	326.74	9593.82
Los Angeles: WithACC	2000	26394	1292	370.74	18704.41

(b) Results of Realworld Datasets

Figure 9: Comparison between GeoCrowd and QGCrowd

5.3.1 QGCrowd vs GeoCrowd

In this part, after putting the parameter of accuracy into GeoCrowd system, we run the platform using real world and synthetic datasets to test the performance of these two platforms. The basic idea is to compare 4 parameters about two platforms. They are illustrated as followings:

1. Total accuracy: The parameter is calculated by adding all the accuracy of assigned tasks. In this particular comparison, we are adding the accuracy of workers in one of the regions of A,B,C and D.

2. Tasks assigned: This parameter is used to test that after taking the accuracy into consideration, whether the QGCrowd could assign the same number of tasks as GeoCrowd.

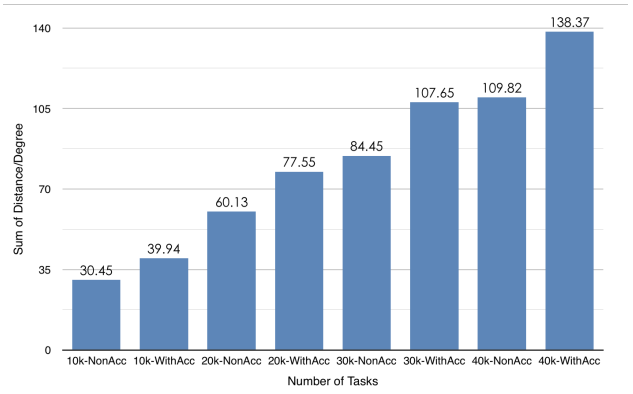
3. Total distance: This parameter is used to test whether the QGCrowd is still assigning tasks that are located near the workers.

4. Running time: This parameter is used to test the latency of QGCrowd.

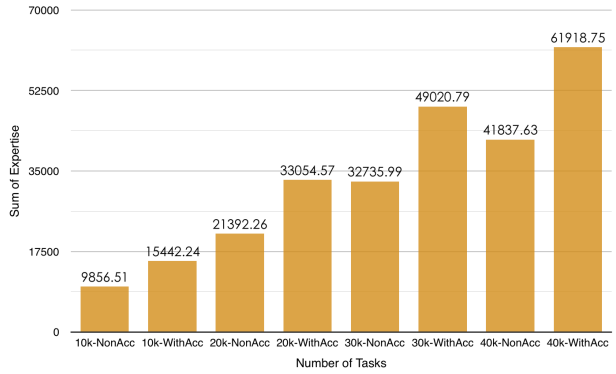
Using the real world and synthetic datasets, we take these parameters after running it on QGCrowd and GeoCrowd. The result is shown in the following Figure 9.

From the table, we can see that under the same experimental setup, the total accuracy of synthetic data is improved by 55.60% in average, while the total distance cost is only increased by 30.07% in average. For the real-world data, we can know that the total accuracy is increased by 85.28%, while the total distance cost is only increased by 15.75% in average.

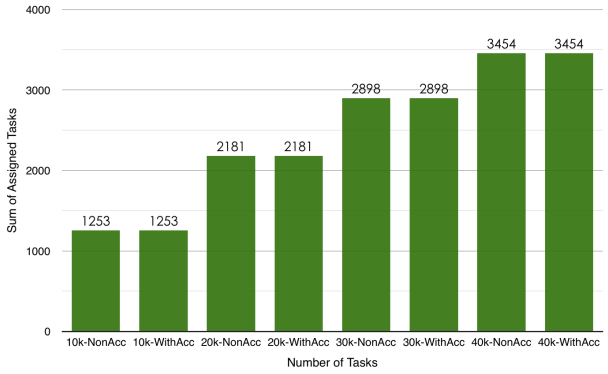
In conclusion, our QGCrowd is able to assign the same number of tasks to workers, while significantly increasing the total accuracy. It also did a good job in assigning the tasks to its nearby workers and doesn't increase much distance cost over the original GeoCrowd.



(a) Summation of Distance Cost



(b) Summation of Expertise



(c) Final Task Assignment

Figure 10: The scalability of QGCrowd

5.3.2 Scalability of QGCrowd

Next we are going to analyze the ability of scaling in QGCrowd. Here we are utilizing different scales of synthetic datasets.

From 5k, we gradually increase the number of tasks requested to 10k,15k and 20k. While the other parameters stay the same(worker set: 1k, MaxT: [0, 10], Region R: 0.1 * 0.1(1 percent of the whole area)), we are able to test whether the outcome of the QGCrowd are changing accordingly. For the output, we take the same 4 parameters to test its performance. Notice that for better comparison, we also present the non-accuracy result, which is the result from original GeoCrowd.

From Figure 10, the overall performance of QGCrowd is good. And there are a few unexpected results need to be mentioned. For the total distance in 10k assignment input, it grows even faster than the 15k assignment input. As explanation, we think it may be due to when the assignment number is 15k, there are more assignments that may be nearer for workers to complete and result in the distance boost.

5.3.3 Effect of MaxT on QGCrowd

In this part, we are testing the effect of MaxT, which is the max number of tasks a worker is willing to perform. Same as the previous section, we are setting all other parameters to be the same and only changing the MaxT from 5 to 20, meaning that a worker's max number of tasks he is willing to take at a time.

The result is shown in Figure 11 (shown in next page). After obtaining the result, we have observed that there are some unexpected results besides the normal results. For the normal results, we could see that the number of assigned tasks is increasing as the MaxT is increasing. But the sum of expertise is decreasing as MaxT is getting larger, which should not be the expected case.

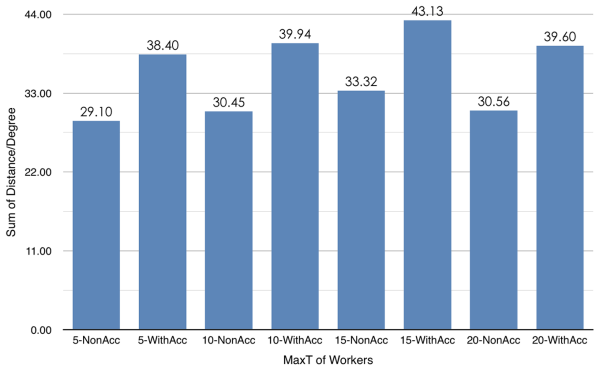
After some consideration and test running, we found that the possible reason for this strange phenomenon is: As MaxT is increasing, the server may assign more tasks to a single worker due to his travel cost to the certain location is lower, which would result in the lower overall cost (distance/accuracy). In other words, the same task may be assigned to someone who is nearer to the location but got less accuracy. This would result in the situation that there are few workers working on more assignments and the sum of accuracy is not guaranteed to increase.

5.3.4 Effect of Spatial Region Constraints on QGCrowd

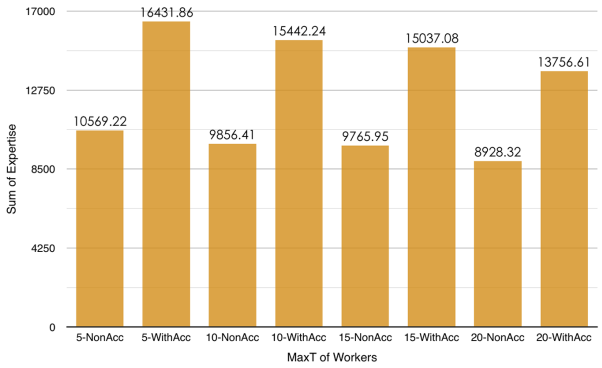
In this final experiment, we measure the impact of varying the spatial region constraints on the performance of our solution. The maximum regions constraints spans from $[0.01 - 0.10]$ percentage to $[0.01 - 0.40]$ percentage of the entire region.

As we can see from the Figure 12 (shown in next page), the parameters like distance cost, expertise and number of tasks assigned are changing within the expectations. As the willing-to-travel distance of each worker is increased, the total number of task assignment increases accordingly, together with the increment of the total expertise and distance cost.

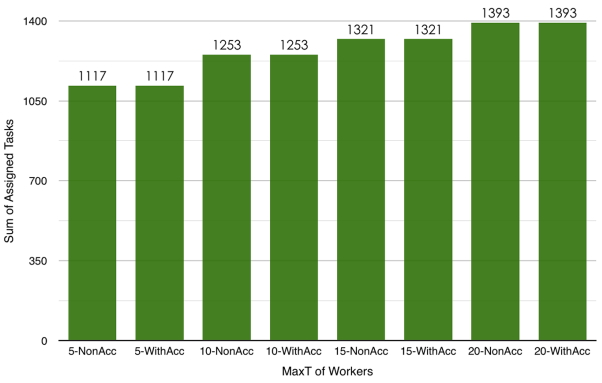
Compared to GeoCrowd, we obtain the same number of task assignment. In addition, we also achieved 52.24% more of accuracy summation in average, while the distance cost is increased by 28.40%. So we can conclude that QGCrowd has better overall performance than GeoCrowd.



(a) Summation of Distance Cost

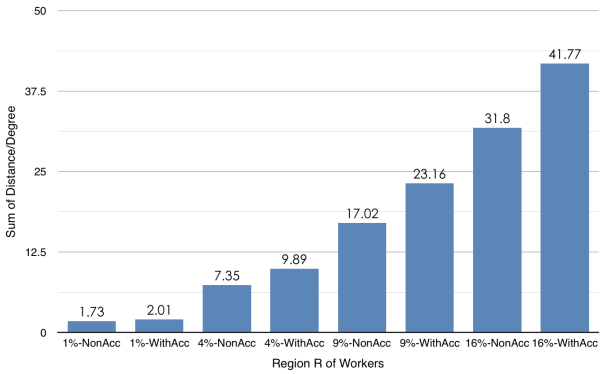


(b) Summation of Expertise

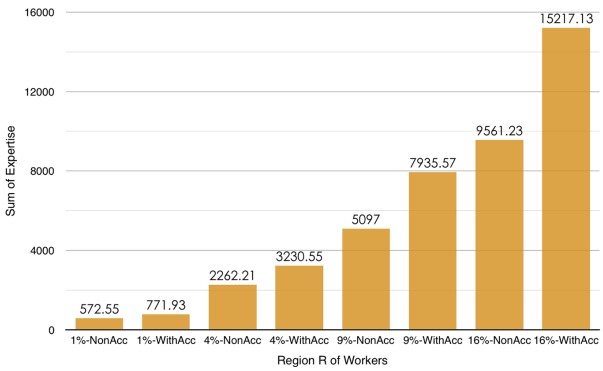


(c) Final Task Assignment

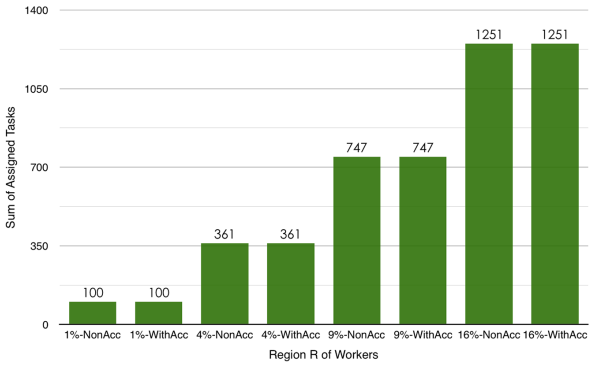
Figure 11: Effect of maxT Constraints on QGCrowd



(a) Summation of Distance Cost



(b) Summation of Expertise



(c) Final Task Assignment

Figure 12: Effect of Spatial Region Constraints on QGCrowd

6. FUTURE WORK

In this paper, we mainly focus on the classification of self-incentivised, server-assigned and single worker based mode. In possible extension of the ongoing work, we could do more research on the other branches of the taxonomy, each with their own advantages.

First thing we could improve on this project is the protection of the privacy of workers. Under the ongoing mode, workers are reporting their current locations whenever they set their profile to be online. This may result in some privacy crisis for the workers[3]. When the worker's information is combined with their physical locations, the information could be so convincing that there could be potential frauds happening to the people based on those information[23].

Next thing we could do better is the some reward mechanism. Since the mode we are developing in this paper is self-incentivised and people are volunteerily performing the tasks, the workers may not be so motivated and the tasks quality could be influenced in a possible bad way. The appropriated reward mechanism could improve on this problem and attract more people to join the trend of becoming a worker[7, 2]. The form of the reward doesn't necessarily have to be money-based. We could work on some cooperation with some authorizations and give the workers some kind of certificate after completing the task.

Last but not least, an actually very important way we could improve the system is to develop the redundant mode of completing tasks. Since we are trying to address spatial crowdsourcing in this platform, so the quality might not have a significant improvement

when the same task is completed by multiple workers. But if there are some tasks that are out of the capability of a single worker and require the workers to cooperate for a single task, the redundant mode would be really necessary.

Besides the branches of modes that we could look into, a user friendly interface of our platform should also be developed and put into real use for testing its performance.

7. CONCLUSION

In this piece of work, we introduce spatial crowdsourcing and formally define the task assignment problem as a maximum task assignment problem. Besides, the basic greedy maximum flow solution, and nearest neighbour priority strategy, we introduced a new strategy in distributing tasks among different workers. We call it lowest cost strategy in which cost is a hybrid of worker's proficiency and the travel cost of the worker. In our experiment on both real datasets and synthetic datasets, we show the advantage of our idea in maximizing the overall task assignment and improving the quality of results of finished tasks. There is always a tradeoff between workers' travel cost and the quality of finished tasks' results, nevertheless the travel cost does not increase too much while the total accuracy of the task has been improved significantly.

In our experiments on both real-world data and synthetic data, QGCrowd can improve the total expertise score by 34.97% at least, while the overall distance cost is also increased by at most 31.35%.

The main contribution of this work is on the formulation of the workers' expertise and combined it into the framework of spatial crowdsourcing. Through our experiments, we learn that the basic solution which turns the spatial task assignment problem into a maximum flow problem is far from enough. Instead, we should use a heuristic strategy that considering the distances traveled by the workers and workers' expertise in specific task types.

8. REFERENCES

[1] F. Alt, A. S. Shirazi, A. Schmidt, U. Kramer, and Z. Nawaz. Location-based crowdsourcing: extending crowdsourcing to the real world. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, pages 13–22. ACM, 2010.

[2] I. Boutsis and V. Kalogeraki. On task assignment for real-time reliable crowdsourcing. In *Distributed Computing Systems (ICDCS), 2014 IEEE 34th International Conference on*, pages 1–10. IEEE, 2014.

[3] M. F. Bulut, Y. S. Yilmaz, and M. Demirbas. Crowdsourcing location-based queries. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, pages 513–518. IEEE, 2011.

[4] C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, and N. Triandopoulos. Anonysense: privacy-aware people-centric sensing. In *Proceedings of the 6th international conference on Mobile systems, applications, and services*, pages 211–224. ACM, 2008.

[5] J. Cranshaw, E. Toch, J. Hong, A. Kittur, and N. Sadeh. Bridging the gap between physical location and online social networks. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, pages 119–128. ACM, 2010.

[6] H. Dang, T. Nguyen, and H. To. Maximum complex task assignment: Towards tasks correlation in spatial crowdsourcing. In *Proceedings of International Conference on Information Integration and Web-based Applications & Services*, page 77. ACM, 2013.

[7] K.-H. Dang and K.-T. Cao. Towards reward-based spatial crowdsourcing. In *Control, Automation and Information Sciences (ICCAIS), 2013 International Conference on*, pages 363–368. IEEE, 2013.

[8] D. Deng, C. Shahabi, and U. Demiryurek. Maximizing the number of worker's self-selected tasks in spatial crowdsourcing. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 324–333. ACM, 2013.

[9] D. Deng, C. Shahabi, and L. Zhu. Task matching and scheduling for multiple workers in spatial crowdsourcing, 2015.

[10] J. Fan, G. Li, B. C. Ooi, K.-l. Tan, and J. Feng. icrowd: An adaptive crowdsourcing framework. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1015–1030. ACM, 2015.

[11] C. for embedded networked sensing (cens). `http://urban.cens.ucla.edu/projects/`.

[12] D. R. Karger, S. Oh, and D. Shah. Budget-optimal task allocation for reliable crowdsourcing systems. *Operations Research*, 62(1):1–24, 2014.

[13] L. Kazemi and C. Shahabi. A privacy-aware framework for participatory sensing. *ACM SIGKDD Explorations Newsletter*, 13(1):43–51, 2011.

[14] L. Kazemi and C. Shahabi. Geocrowd: enabling query answering with spatial crowdsourcing. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 189–198. ACM, 2012.

[15] L. Kazemi, C. Shahabi, and L. Chen. Geotrucrowd: trustworthy query answering with spatial crowdsourcing. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 314–323. ACM, 2013.

[16] A. mechanical turk. `http://www.mturk.com`.

[17] Y. S. Y. Muhammed Fatih Bulut and M. Demirbas. Crowdsourcing location-based queries. *In Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, pages 513–518, 2011.

[18] U. of california berkeley. `http://traffic.berkeley.edu/`.

[19] S. N. A. Project. Gowalla. `http://snap.stanford.edu/data/loc-gowalla.html`.

[20] C. Shahabi. Towards a generic framework for trustworthy spatial crowdsourcing. In *Proceedings of the 12th International ACM Workshop on Data Engineering for Wireless and Mobile Acess*, pages 1–4. ACM, 2013.

[21] H. To, M. Asghari, D. Deng, and C. Shahabi. Scawg: A toolbox for generating synthetic workload for spatial crowdsourcing. In *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, pages 1–6. IEEE, 2016.

[22] H. To, G. Ghinita, and C. Shahabi. A framework for protecting worker location privacy in spatial crowdsourcing. *Proceedings of the VLDB Endowment*, 7(10):919–930, 2014.

[23] H. To, G. Ghinita, and C. Shahabi. Privgeocrowd: A toolbox for studying private spatial crowdsourcing. In *Data Engineering (ICDE), 2015 IEEE 31st International Conference on*, pages 1404–1407. IEEE, 2015.

[24] H. To, C. Shahabi, and L. Kazemi. A server-assigned spatial crowdsourcing framework. *ACM Transactions on Spatial*

Algorithms and Systems, 1(1):2, 2015.

[25] X. Xie, H. Chen, and H. Wu. Bargain-based stimulation mechanism for selfish mobile nodes in participatory sensing network. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON'09. 6th Annual IEEE Communications Society Conference on*, pages 1–9. IEEE, 2009.

[26] T. Yan, M. Marzilli, R. Holmes, D. Ganesan, and M. Corner. mcrowd: a platform for mobile crowdsourcing. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, pages 347–348. ACM, 2009.

[27] Yelp. Yelp Dataset Challenge. `https://www.yelp.com/dataset_challenge`.