Noname manuscript No.
(will be inserted by the editor)

Crowdsourcing Planar Facility Location Allocation Problems

Mohammad Allahbakhsh · Saeed Arbabi · Mohammadreza Galavii · Florian Daniel · Boualem Benatallah

Received: date / Accepted: date

Abstract Facility location allocation is key to success of urban design, mainly in designing transport systems, finding locations for warehouse, fire stations and so on. The problem of determining locations of k facilities so that provides service to n customers, also known as p-median problems, is one of the well-known $\mathcal{NP}$-hard problems. Several heuristics have been proposed to solve location allocation problems, each of which has several limitations such as accuracy, time and flexibility, besides their advantages. In this paper, we propose to solve the p-median problems using crowdsourcing and gamification techniques. We present a crowdsourced game, called SolveIt, which employs wisdom and intelligence of the crowd to solve location allocation problems. We have presented a data model for representing p-median problems, designed and implemented the game and tested it using gold standards generated using a genetic algorithm tool. We have also compared the results obtained from SolveIt with the results of a well-known approach called Cooper. The evaluations show the accuracy and superiority of the results obtained from SolveIt players. We have also discussed the limitations and possible applications of the proposed approach.

Keywords Location Allocation · p-median · Planar · Crowdsourcig · Gamification · SolveIt

M. Allahbakhsh, S. Arbabi and M. Galavii
University of Zabol, Iran
E-mail: {allahbakhsh,sarbabi,mohammad.galavii}@uoz.ac.ir

F. Daniel
Politecnico di Milano, Italy
E-mail: florian.daniel@polimi.it

M. Allahbakhsh, B. Benatallah
The University of New South Wales, Australia
E-mail: {mallahbakhsh, boualem}@cse.unsw.edu.au

1 Introduction

The location allocation problem, also referred to as p-median problem, is to find suitable locations for p facilities so as to provide a specific service to a set of n customers [13]. The problem is believed to date back to 17^{th} century where the French mathematician Pierre de Fermat(1601-1665) posed his well-known geometric problem as: How to find a point in an Euclidean plane to minimise the sum of its distance from three given points [35]. The problem was extended later to a multi-facility version in which the problem is finding locations on a plane for p facilities to give service to n demand points. This problem is also called *continuous p-median* or the *multi-source Weber* problem [17,5,13]. Hakimi reduced the planar problem to a p-median problem on a graph [24,26]. Hakimi's proposed problem is to find locations of facilities on a discrete set of graph nodes, rather than on a plane. This reduction makes the problem more tractable and easier to solve.

The p-median problems are proved to be $\mathcal{NP}$-hard on a general graph/plane [41, 17,5]. Hence, several heuristic solutions have been proposed to solve the problem [41]. Most of the proposed techniques try to reduce the planar problem to a discrete setting in which the locations are selected out of a finite set of candidate locations. Even research works that claim to solve the planar problems reduce the plane to a set or a mesh of points [41,17,5]. The main reason for such a reduction is to decrease the number of candidate locations from an infinite set to a finite set.

In this research we propose a crowdsourced online game to solve continuous p-median problems without reduction and on a plane. Crowdsourcing is an emerging distributed technique for solving problems that rely on the wisdom of the crowd as well as on human intelligence. There are several problems that are very easy to do for a human being, but very hard or even impossible for a computer to do [1]. Image tagging, audio transcription and identifying objects in photos are examples of such tasks [15]. Crowdsourcing proposes to harness human intelligence to solve these problems. So, in crowdsourcing terminology these tasks are called Human Intelligence Tasks or simply HITS [36,1]. As the main asset in crowdsourcing is people contributions, it is very important to make people motivated to participate. Different types of incentives can be considered in a crowdsourcing task, mainly extrinsic (e.g., monetary) and intrinsic (e.g., fun). Research shows that intrinsic motivations can result in higher quality contributions [42]. One of the most important intrinsic motivations is entertainment, the basis on which online games are built [39]. In online crowdsourcing games people contribute for the purpose of entertainment, altruism, etc. Based on this, researchers have proposed to leverage games for solving serious problems, such as image tagging [46] and protein folding [11]. That is why these games are called serious game[28].

We call our proposed game SolveIt, meaning that you use this game to solve location allocation problems. In the proposed game the p-median problem is converted to a simple drag-and-drop game, in which the demand points, also called customers, are fixed in terms of their locations. But players can change the location of facilities so as to minimize overall distance. The main contribution of this game is that it relies on human intelligence to find the optimal location for the facilities. When you give the game to a large crowd of people and provide them with interesting incentives, they will do heir best to find the optimal solution. We do not have any idea about the best place. However, research shows that the quality of the crowd contributions is comparable to those of experts [34]. Therefore,

the best contribution among the crowd is expected to have an acceptable level of quality. To design such a game, we first introduce a graph data model to represent a generic location allocation problem, and upon this data model, we build our game. In summary the main contributions of the paper are as follows:

- We propose a graph data model for a better understanding and representation of location allocation problems. The graph data model consists of nodes, which are facilities, and customers and edges which are the coverage relationships between facilities and customers.
- We propose an online crowdsourcing game, called SolveIt, to solve planar p-median problems. The proposed game leverages the human intelligence and the crowd wisdom to solve $\mathcal{NP}$-hard problems that, generally, are approximated using heuristic algorithms.
- We evaluate the proposed game in a real world environment. We ask a group of 40 students in the University of Zabol, to participate in the game and solve the proposed problems. The contributions received from the crowd are compared to those obtained from a specific implementation of genetic algorithms, which is being used as gold standard. Also, we compare the SolveIt results with results of one of the well-known approaches called Cooper method. The comparison of results shows the high accuracy and superiority of the estimations obtained from the proposed game.

The rest of the paper is organized as follows. In the Section 2, the data model is presented. We propose our game in Section 3. The implementation details are presented in Section 3. Performance evaluation and comparison results are brought in Section 5. In the Section 7, we study the related literature. In the Section 6 a discussion on the game and its pros and cons is presented; and finally, we conclude in Section 8.

2 Principles and Basic Concepts

In this section, we first describe an example scenario, to show one of the sample applications of the proposed model and to simplify the understanding of the problem that is being tackled in this research. Then we present a graph data model for representation of P-Median problems, as well as some basic notations.

2.1 Motivating Scenarios

Choosing polling stations in elections is a typical location problem. In most elections, e.g., presidential elections, people need to approach polling stations to cast their votes. The government needs to open, say p, polling stations to collect people's votes. The location of each voting station can be chosen from a finite set of available locations. The p locations should be selected so as to minimize the distance that each voter travels to reach the polling station. It is almost impossible to decrease the distance for all voters at the same time. In other words, when the distance is decreased for one voter, the distance may increase for some others. So, we define the location selection policy more precisely as: the locations should be

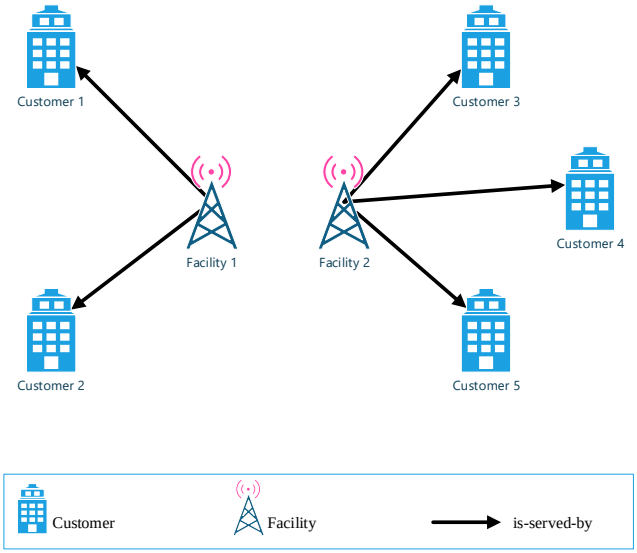


Fig. 1: A sample representation of a p-median graph.

selected so as to minimize the average distance each voter walks, i.e., to minimize the sum of the distances all voters walk.

Other applications of p-Median location problem are finding optimal locations for local services such as fire services stations, emergency medical services, etc. For example, assume that a company gives services to a wide region, so it needs to establish a number of, say p, local stores, facilities or warehouses, so that it delivers goods to customers in each region from its local store. To decrease delivery costs, the company should set locations of local stores, so that the overall delivery distance is minimized. It means that the company needs a p-median problem in which facility locations are the p optimal points that should be found.

2.2 Problem Formulation and Data Model

In this research, we propose to crowdsource location problems using an online game. In other words, we aim to rely on human intelligence to solve such complicated problems, instead of employing sophisticated mathematical optimization solutions. Assume that in an online platform, there exist N_Y p-median problems, denoted by $Y = \{y_n | 1 \leq n \leq N_Y\}$, to be solved. As explained earlier, each individual problem is tackled as a multi-player game, which means, one game will be defined corresponding to each problem. Hence, there are also N_Y games, which are depicted by $G = \{g_k | 1 \leq k \leq N_Y\}$. Assume that there are N_π people, also told players, registered in the online platform who might be willing to contribute to games. Let's denote people, or more precisely players, with $\Pi = \{\pi_m | 1 \leq m \leq N_\pi\}$, and the people who have contributed to a specific game, say g by $\Pi_g = \{\pi | \pi \rightarrow g\}$,

where $\pi \rightarrow g$ means the player π has contributed to the game g. Let R be the set of all contributions of all players to all games, R_g be the set of all contributions provided for the game g, and R_π^g the contribution of player π to the game g.

In order to simplify representing and understanding of such systems containing problems, games, players, etc., and to assist with solving this kind of problems, we present a graph data model (referred to as p-Median Problem model or shortly PMP in the followings). We use this graph data model to represent and organize entities and relationships amongst them, in a location allocation problem. We propose to represent a p-median problem with an attributed directed graph $G = (V, E)$, where V is the set of nodes representing entities and $E \subset V \times V$ is the set of edges representing relationships between nodes. A sample representation of a PMP is depicted in Figure. 1. An entity is an independent object with a unique identity.

There are two types of entities in PMP: *customers* and *facilities*, and one type of relationship between entities named *is-served-by*.

Customer. A customer is a person, a group of people, an organization or even a geographical region which is being served as a unique whole. We assume that there are N customers in a PMP graph and denote them by $C = \{c_i | 1 \leq i \leq N\}$. A customer is identified by a unique id number, i.e., c_{id}. A customer also has a weight which represents the importance, urgency of needs or priority of the customer. The weight is a positive number and is represented by c_w. Moreover, each customer resides in a specific location. The location of a customer, denoted by c_L, contains a set of information which uniquely specifies where the customer resides on the plane, map, etc. We assume that the location of customers is specified by the Cartesian coordinates of the customer on the map. The location of customers are fixed, and do not change, when the location problem is being solved.

Facility. A facility is an equipment, device, service, person, etc., that acts as a local service provider. We assume that there are p facilities in a PMP graph that are denoted by $F = \{f_j | 1 \leq j \leq p\}$. In addition to a unique id represented by f_{id}, a facility has a coverage rage. The coverage range of the facility f, denoted by f_ρ, is a positive number reflecting the furthest distance in which the service provided by f is available. Similar to customers, each facility also has a location, which is denoted by f_L. Location of facilities are subject to changes, and this change directly impacts the quality of the final solution of the problem. Facility locations are recommended by players who contribute to a given game. We also use notation f_L^π to represent the location that has been recommended by the player π for the facility f in a given game.

is-served-by. While there might be several types of relationships between nodes in a generic location problem graph, we identify only one type of relationships in PMP graph, called *is-served-by*. The existence of an is-served-by relationship between a facility f and the customer c, means that the c is being served by the facility f. More precisely, $c \xrightarrow{is-served-by} f$ means that the distance between c and f is less than f_ρ, so c is being served by facility f. The *is-served-by* is a directed relation which starts from the customer and ends on a facility. To keep the data model generic, we assume that PMP is an attributed graph in which *is-served-by*

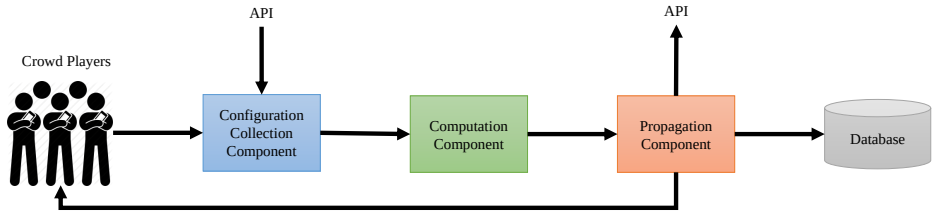


Fig. 2: The overall architecture of SolveIt.

can have a set of attributes. These attributes can represent information such as why/when a node is served by a facility. These attributes also might be extracted from the correlations between attributes of the relationship itself, or attributes of the nodes connected by the relationship [40]. However, in this work, for simplicity of understanding, we consider relationships without attributes.

According to this data model, we can formalize the motivating scenario, presented in Section 2.1, as follows. Voters are customers who need to be served, and polling stations are facilities that are supposed to serve customers. The facilities should be selected so that all customer are served and each customer is served only by one facility. When a voter casts her vote in a polling station she *is-served-by* that polling station. So, an *is-served-by* relationship is established between the voter and the station.

3 Proposed Game - SolveIt

3.1 Overview and Architecture

In this section, we propose a game for solving $\mathcal{NP}$-hard p-median problems. As the game is inspired by Foldit game [11], it is called *SolveIt*. The main objective of presenting this game is to show how crwodsourcing games can be employed to solve sophisticated $\mathcal{NP}$-hard problems, here specifically, facility location allocation optimization problems.

As depicted in the Figure 2, the overall architecture of SolveIt consists of three main components: (i) *Configuration Collection* Component, (ii) *Computation* Component, and (iii) *Propagation* Component. More details are presented in the following subsections.

3.2 Configuration Collection Component

The first component of the architecture is the configuration collection component. This component is responsible for collecting the configuration of a given p-median problem, which is being defined as a game, as proposed in Section 2.2. The configuration of a game g, referred to as g_κ, generally consist of two sets of locations. The first set of locations in the g_κ contains locations of all customers, i.e., $\Lambda_C = \{c_L | \forall c \in C\}$.

The second set of locations is the set which contains locations recommended for all facilities in a given game. As stated earlier, each player has her own recommendations for the location of facilities. Assume that the set of locations recommended for all facilities in a given game by a particular player π is $\Lambda_F^\pi = \{f_L | \forall f \in F\}$. Then, the set of all facility locations that should be collected for the game g is:

$$\Lambda_F^g = \bigcup_{\pi \to g} \Lambda_F^\pi \tag{1}$$

In Equation 1, $\pi \to g$ means that the player π has contributed to the game g. Finally the configuration of the game g, that should be collected, i.e., g_κ, is built as follows:

$$g_\kappa = \Lambda_F^g \bigcup \Lambda_C \tag{2}$$

As shown in Figure 2, configuration of a game might be collected through a user interface or an API. In our research, players recommend solutions through the provided graphical user interface.

3.3 Computation Component

The second component of the SolveIt architecture is the computation component. This component is responsible for computing all required metrics. There are two types of metrics that should be computed in SolveIt: *overall distance* and *ranks*.

3.3.1 Overall Distance

The overall distance between customers and facilities is the main metric upon which the SolveIt game and all solutions for p-median problems are built. This distance is computed for each configuration which is collected through the configuration collection component, each of which corresponds to one of the players contributing to the game.

The overall distance is the sum of the distances between all customers and their serving facilities, according to a submitted configuration. More precisely, assume that the customer c is served by the facility f, which is denoted by $c \to f$. Let's denote the pairwise distance between c and f by d_{cf}. The pairwise distance is a positive number and can be computed based on several principles. In a simple Euclidean space, the pairwise distance between a customer and a facility is the length of the straight line connecting the two points. In a geographical map, the pairwise distance between two points can be calculated based on the length of the roads, the sailing path, a flight path or simply a walk between the two points. It can also be computed based on other similarity and distance metrics such as Jaccard distance between sets, vectors, etc. Based on the pairwise distances between facilities and customers, the overall distance is calculated as follows:

$$D = \sum_{c \to f} d_{cf} \tag{3}$$

The distance computed using the Equation 3 is not weighted and is suitable just for the cases in which facilities have no weights or have the same weights. As we explained in Section 2.2, in our proposed model each customer has an associated weight that reflects its importance, urgency, priority, etc. This means that some customers are more important than others, and when locating facilities on the plane, the priority should be given to such customers.

Assume that, in a configuration submitted by the player π for the game g, the customer c, with the weight of c_w is served by the facility f which is in the distance d_{cf} from c. The overall weighted distance of all facilities from the customers, in such a configuration, is denoted by Δ_g^π and computed as follows.

$$\Delta_g^\pi = \sum_{c \to f} d_{cf} \times c_w \tag{4}$$

In Equation 4, the Δ_g^π is the overall distance calculated for a configuration collected from the player π for a given game g. The distance is a positive number. Since the goal of the game is to minimize this distance, the smaller the distance, the better.

3.3.2 Ranks

In a crowdsourced game like SolveIt, a large number of players are expected to contribute to the game. When a player changes his/her recommended configuration, the distance computed for her configuration will also change, based on the Equation 4. On the other hand, the reward that will be given to a player depends on such a distance. As we deal with a minimization problem, the smaller distances might receive higher rewards, or even, just the smallest distance will be chosen as the winner of the game, and receive the reward. So, it is necessary to compare the distances obtained by players, and compute and assign each player a rank, to show her where she stands amongst all players of the game.

Players' ranks are the second group of metrics computed by the computation component of the SolveIt. Several aggregation techniques can be used while building the final outcome of crowdsourcing tasks [12]. In this game we chose to reward the best answer, who is the player with the smallest overall distance. Hence, assume that the Δ_g is the set of all distances computed for the game g, the winner of the game is specified as follows:

$$Winner = \pi \text{ , where } \Delta_g^\pi \text{ is } min(\Delta_g) \tag{5}$$

The game g might have several winners if more than on player provide same solutions.

3.4 Propagation Component

The propagation component is responsible for propagation of several types of data and information. First of all, the propagation component is responsible for rank propagation. When players contribute to a game, they need to know their ranks. Being informed of their ranks, players might get motivated to put more effort to

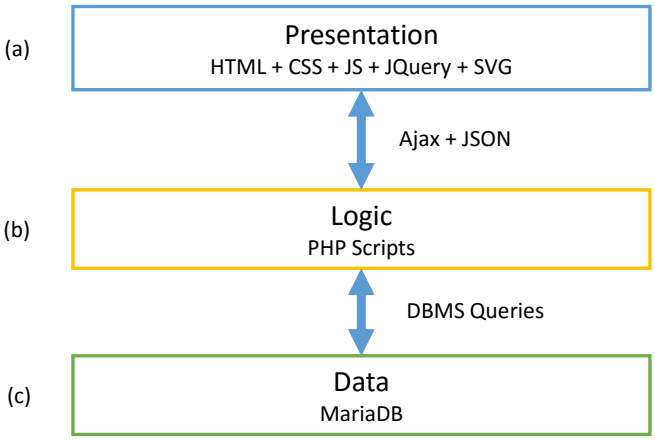


Fig. 3: Overall Architecture of SolveIt Implementation.

reach a better place, as several research evidences show that providing players with informative items and online shepherding can improve the quality of their outcomes [16,12]. Ranks are also the key metrics in specifying the winner of the games, so it is crucial to have them clearly available to all players, or whoever interested.

In addition to the ranks, the history of the users, their recommended configurations, game histories and any other type of information can become available through the propagation component of the game. As illustrated in Figure 2, the propagation can be done using the user interface of the game, or through an API.

4 SolveIt Implementation

We have implemented SolveIt as an online web based crowdsourcing game, available on the web site of the WebTech Lab at The University of Zabol[1]. SolveIt also has a page on github[2] in which the code of SolveIt is openly available.

The overall architecture of SolveIt implementation is represented in Figure. 3. The proposed architecture consists of three layers: (i) *presentation*, (ii) *logic* and (iii) *data*.

4.1 Presentation

The first layer of the game is the presentation layer. This layer, as depicted in Figure 3(a), is the client side of the game and is responsible for providing players, i.e., crowd workers, with an adequate user interface, through which, players can contribute to the games. The presentation layer is developed using HTML, CSS, JavaScript, jQuery and SVG.

[1] http:// webtech.uoz.ac.ir/projects/solveit
[2] https://github.com/webtechuoz/SolveIt

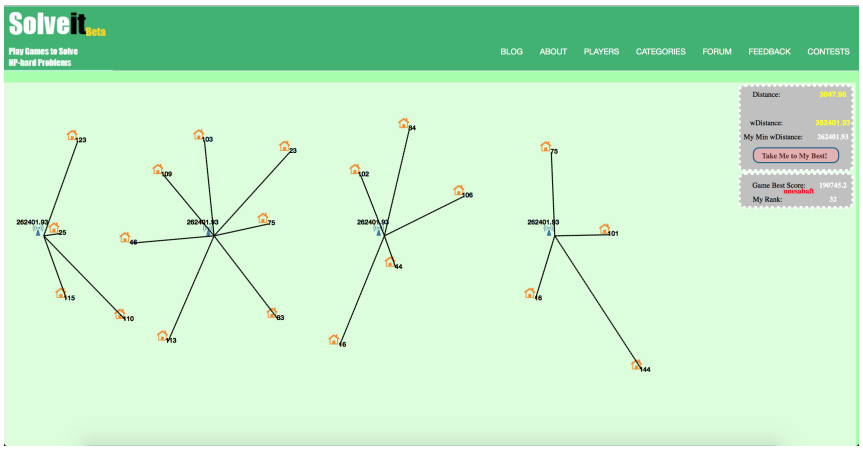


Fig. 4: User Interface of SolveIt.

A sample screenshot of the web interface of SolveIt is represented in Figure 4. Via the game interface, players can register in the game or use their Facebook credentials to login. They can also find information on how to play the game, and try a demo game, before entering the real games. When a player, say π, starts the game g, she faces a screen on which there are N fixed points and p moveable points. The fixed points (house shapes) represent the customers and moveable points (WiFi tower shapes) represent facilities. The player can grab facilities and move them around the screen. Each move changes the overall distance of facilities from customers. This distance is shown on the right corner of the game to help players to see their current computed distance. The game has also a dashboard on which players can see a set of metrics. They can see their current distance, their best distance they ever obtained in the current game, their rank between the players who have contributed to the game so far, and the distance/id of the best player so far. The game also has the feature to take a player to the best configuration she has ever reached in a game. This feature is useful to help players to always try for better answers, while they are confident that they do not loose their best chance. The final recommended configuration for each player is the last configuration in which she has left the game. The user interface component relies on Ajax and JSON technologies to communicate with the server side, that is, the logic component.

4.2 Logic

The second layer of the game is the logic layer, which is implemented using PHP[3] ver. 7. This layer is responsible for analysing and responding to the requests received from the presentation layer. When a request is received from client, the logic layer interprets the request to make sure that it is a genuine and valid request. Then, using the data stored in the data layer, the request is responded accord-

[3] http://php.net/

Table 1: Game setup

Game	No. of Facilities	No. of Customers	Solved By	No. of Participants
$C_{15}F_2$	2	15	Students	40
$C_{15}F_4$	4	15	Experts	2
$C_{15}F_6$	6	15	Experts	2
$C_{15}F_8$	8	15	Experts	2
$C_{20}F_2$	2	20	Experts	2
$C_{20}F_4$	4	20	Students	40
$C_{20}F_6$	6	20	Experts	2
$C_{20}F_8$	8	20	Experts	2
$C_{25}F_2$	2	25	Experts	2
$C_{25}F_4$	4	25	Experts	2
$C_{25}F_6$	6	25	Students	40
$C_{25}F_8$	8	25	Experts	2
$C_{30}F_2$	2	30	Experts	2
$C_{30}F_4$	4	30	Experts	2
$C_{30}F_6$	6	30	Experts	2
$C_{30}F_8$	8	30	Students	40

ingly. The communications between the logic layer and the data layer takes place based on DBMS queries. All distance and ranks computations are performed in this layer.

4.3 Data

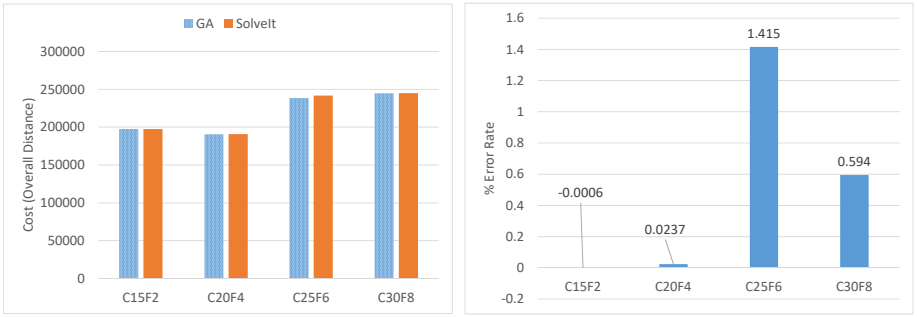
The data layer is responsible for data management. This layer, keeps all data about the games, players, facilities, customers, contributions, and so on. The MariaDB ver. 10.1.16 is leveraged as the DBMS in this layer[4].

5 Experimentation and Evaluation

In this section, we evaluate the performance of the game in a real world experimentation and assess the accuracy of the results obtained from the SolveIt game. Recall that the planar p-Median problems are $\mathcal{NP}$-hard [41,17,5]. Hence, the proposed solutions for solving this kind of problems are generally heuristic approximation solutions, trying to obtain results of acceptable accuracy in a reasonable running time. Several pieces of research shows that for small to medium sized location allocation problems genetic algorithms (GA) show a better performance than the other heuristic approaches [30,31]. Although, when the size of the problem increases the running time of GA approaches increases very fast, but for medium to small problems, which are the subject of this research too, both the accuracy and time complexity of GA approaches are better than other methods. Therefore, we select the GA approach, in order to assess accuracy of the results generated by our proposed game.

We also use the Cooper [9,10,3,6,6] approach to compare the accuracy of the results obtained form SolveIt. The approach proposed by Cooper is an iterative approach that alternates between the relocating facilities and reallocating customers

[4] http://mariadb.org/



(a) Distance (b) Error Rate

Fig. 5: Comparison of SolveIt results with GA gold standards

to facilities until no further relocations are possible. That is why the Cooper approach is also known as alternate model. Cooper divides a p-median problem into p problems of 1-median size and solves them using a classic approach proposed by Weiszfeld [45]. In the literature, Cooper approach is one of the heuristics that also has been used for solving continuous p-median problems [6]. We compare the overall costs obtained from Cooper and SolveIt approaches as well as their error rates to show the credibility of our proposed game.

5.1 Game Setup

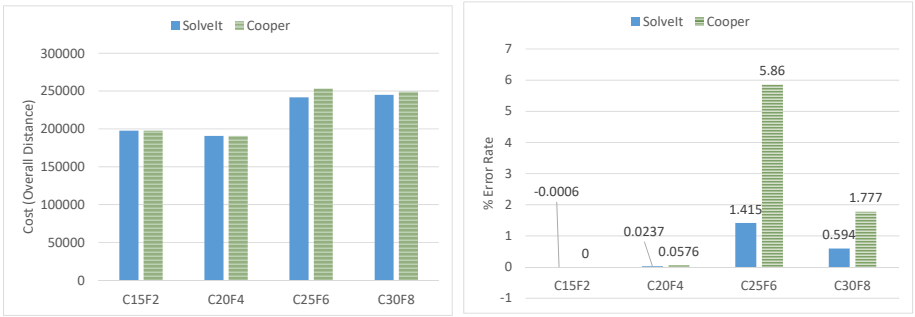
To assess the accuracy of the results obtained by SolveIt, we have developed 16 games, each of which have different numbers of customers and facilities. Let us denote each game by C_iF_j, where i is the number of customers, and j the number of facilities. The config of the defined games is proposed in the Table 1.

We asked a group of 40 university students to participate in games $C_{15}F_2$, $C_{20}F_4$, $C_{25}F_6$ and $C_{30}F_8$ and solve them using the SolveIt game. The students were in the third year of bachelor of computer engineering course, but were not familiar with the concepts of facility location allocation. The best answer, out of the players' contributions to each game, is selected as the result of the game.

We also asked two domain experts to solve other games in the Table 1. The chosen domain experts both hold a master in computer engineering and are familiar with facility location allocation and p-$Median$ problems. Again, the best answer is selected as the game result.

To be able to check the accuracy of the results obtained from the crowd and experts, we need a gold standard to compare the results with. We used a GA algorithm, as explained earlier in this section, to obtain these gold standard vales [30]. The GA results are also used to assess accuracy of the Cooper approach. We developed the GA and Cooper algorithm using Matlab® on a machine running Windows® 10, with $8GB$ of RAM. We apply the GA approach to the games defined in Table 1 and record the results.

To make sure that GA does not remain in a local optimum and generates accurate results, we let the GA approach take 1000 iterations and then choose its



(a) Distance (b) Error Rate

Fig. 6: Comparison between SolveIt and Cooper Results

best result as the result of one run of the algorithm. We repeat this process for 100 rounds, and then the best result found by the algorithm is selected as the GA result. More precisely, the GA result is the best result selected out of $100,000$ results generated by the GA approach. Recall that these problems are $\mathcal{NP}$-hard and whatever solution we choose for comparison reason, should be a heuristic, and these problems, by their nature, do not have exact answers. Therefore, we tried to make sure that the answers chosen as the gold standards are enough accurate by over-repeating and picking the best answer.

5.2 Evaluation Metric

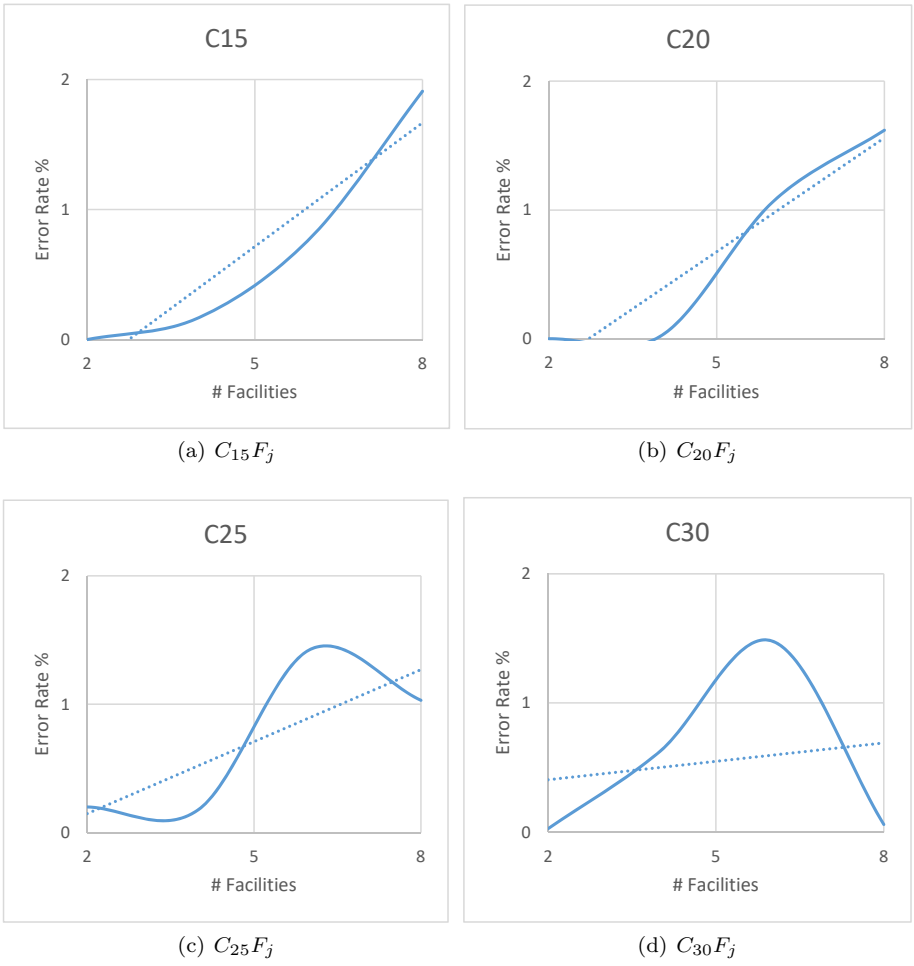
The main evaluation metric by which we assess accuracy of SolveIt results is the *error rate*. The error rate of the results depends on the distance between the result obtained by a human or Cooper and the gold standard obtained from the GA. Let's denote the gold standard with *gold*, human result with *human* and Cooper results with *Cooper*. The error rate of the result is calcuated as follows:

$$error = \frac{(humanorCooper) - gold}{\lambda} \times 100 \tag{6}$$

In Equation 6, the λ is the average of the best results obtained by SolveIt/Cooper and the gold standard.

5.3 Evaluation Results

In order to evaluate SolveIt, we first compare the results obtained by SolveIt with the gold standards obtained form GA. As shown in Figure 5(a), the best overall distances obtained by the SolveIt players for the 4 played games is very close to the gold standards. Particularly, in the $C_{15}F_2$ game SolveIt beats the gold standard and is a little bit better than GA results. In Figure 5(b), the error rate for the SolveIt is compared based on the Equation 6. In all the four cases the error rate is



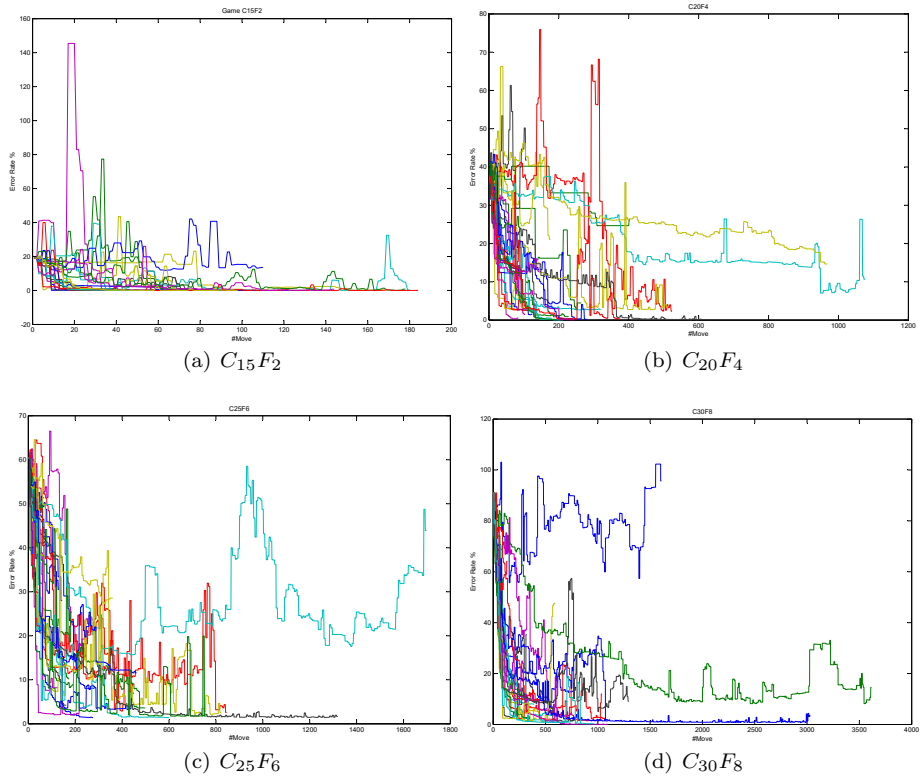
(a) $C_{15}F_j$ (b) $C_{20}F_j$

(c) $C_{25}F_j$ (d) $C_{30}F_j$

Fig. 7: Comparison of SolveIt results with GA approach ($j \in \{2, 4, 6, 8\}$)

less than 1.5%. It means that the difference between the SolveIt result and the gold standard is less than two percent of the average of the two results. More precisely, for the games $C_{15}F_2$, $C_{20}F_4$ and $C_{30}F_8$ the error rate is almost zero, i.e., -0.00065, 0.024 and 0.059 percent, respectively. As the game $C_{25}F_6$ was a little bit tricky for the players, the error rate is a little bit higher, i.e., 1.41 percent, which is still very low. So, the results show that the accuracy of the best results obtained from the players is highly accurate and dependable.

As depicted in Figure 6, we also compare the overall costs as well as the error rates of the results received from SolveIt and Cooper to show the credibility of our proposed approach. Figure 6(a) shows that in almost all cases, the overall cost obtained from SolveIt is smaller than the Cooper results. Particularly, when the number of customers and the facilities increase, the difference between the obtained costs from the two approaches increases as well. In terms of error rates,



(a) $C_{15}F_2$ (b) $C_{20}F_4$

(c) $C_{25}F_6$ (d) $C_{30}F_8$

Fig. 8: Error Rate of the Players

SolveIt shows a better performance. As shown in Figure 6(b), the error rate of results obtained from SolveIt is always better than the Cooper approach. Again, when the size of the problem, i.e., the number of customers and facilities, increases, a more notable difference between error rates is witnessed. Therefore, the results shows the superiority of our proposed model over one of the well-known solutions for p-median problems.

As mentioned earlier, we designed 16 games and asked a crowd of students to solve four of them and the rest are solved by domain experts. The accuracy of the results obtained for all 16 games is depicted in Figure 7. Again, as one can see for all games the error rate of the results is always less than two percent. It means that even when we do not tap into the crowd wisdom and just employ few domain experts to solve the games, the results are still accurate enough. The other inference from this figure can be that the accuracy of human results decreases when the number of facilities increases. The trend line of the four charts in this figures support this inference. We know from the literature that this is the case for almost all other heuristic approaches [30,31]. So, this behaviour can be accepted form SolveIt as well.

We have also tracked the error rate of all crowd players while they have been playing SolveIt. The results are illustrated in Figure 8. In Figures 8(a), 8(b), 8(c)

and 8(d), the vertical axis shows the error rate of a player and the horizontal axis shows the move number of the player. A move is any change made to the arrangement of facilities on the screen. Different games takes players different number of moves, depending on the number of facilities and customers in the game. Every line in the Figures 8(a), 8(b), 8(c) and 8(d) shows unique user and evaluation of his/her error rate in the time. As one can see, in almost all games, excepting for few outliers, error rate of users decreases rapidly and gets close to zero. The other observation is that, by increasing the number of facilities, the number of steps/moves, and consequently the time, that a player must take to get close to the solution, increases. Since we conduct our experiments in a controlled area in which players have to finish games in a specific period of time, we do not bring any time comparisons.

6 Discussion

The experimentation results and comparison with the gold standards we obtained from the developed genetic algorithm, shows the promising performance of SolveIt. The results in comparison with GA are reasonably accurate and the behaviour is predictable. Also, in comparison with the Cooper approach, SolveIt game shows a more promising performance. But we know that there are many other heuristic techniques proposed for solving p-Median problems. In the following we discuss how SolveIt can be justified in the domain, what types of problems are better candidates to be solved by SolveIt, and what are its pros and cons.

6.1 Contributions

The most important components of a location allocation approach are the followings:

1. The decision space X which describe where facilities may be placed.
   (a) *Continuous location problems*: The decision space is the space $\mathbb{R}^d$ or some subspace of it.
   (b) *Network location problems*: The clients and the facilities are points on a graph. Depending on the model, facilities may either be placed only at vertices of the graph or also on edges.
   (c) *Discrete location problems*: A discrete set of pre-specified locations where facilities can be placed is given.
2. The number of facilities to be placed, i.e., p.
3. The distance measure d, which is needed to measure the distance between customers and facilities.
   (a) Distance based on metrics, such as the L_q metrics family.
   (b) Network distance in network locations models, which is calculated based on shortest path distances with respect to given edge lengths.
4. Weights.
5. The decision maker's objective function.

As described earlier, p-median problems turn out to be $\mathcal{NP}$-hard in most settings (continuous and network cases), and for most distance functions. Typically,

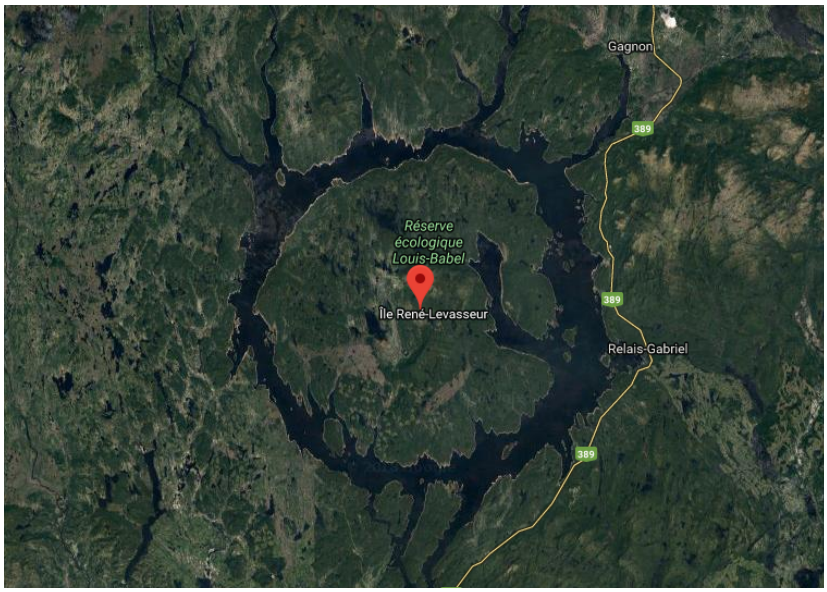


Fig. 9: Manicougan Lake reservoir, an example of areas with complex maps (obtained from Google Maps).

polynomial time algorithms exist for solving the problem for fixed values of p, but these are not often practical due to the dependence of the running time on p.

When it comes to continuous cases, the problem gets much harder to tackle, especially when p is an arbitrary input parameter. That is why we propose a solution to the continuous p-median problems. It is notable that SolveIt can solve all types of location allocation problems, thanks to its reliance on the human intelligence. We chose to solve continuous problems to show that as our proposed approach works for the harder case, it obviously will work for the simpler ones.

Furthermore, SolveIt is a new approach to solving p-Median problems using the wisdom of the crowd. A wide variety of solutions have been proposed for solving p-Median problems, but, to the best of our knowledge, none of them relies on human intelligence and crowd wisdom. Many research evidences show that relying on the crowd for solving problems can lead to better quality with lower costs. Even the quality of the results, under specific circumstances, is comparable to those obtained from domain experts [1, 12].

Moreover, motivating users using gamification can even lead to obtaining results with even higher qualities than domain experts' contributions. Different types of motivations can be used in gamification, such as rewards, attention and attribution, achievement, competition, etc. [28]. In this game we use attention technique by showing the name of the winner on the scoreboard. We also use competition technique to increase the motivation of players. These can lead to more dependable results, comparing to other heuristics.

Next, crowdsourcing is usually suitable for tasks that are too hard for computers but very simple for human. These tasks are also called Human Intelligence Tasks (HITs). Locating objects in images and photo tagging are examples of such

problems [15]. Location allocation problems in some cases can become HITs. Problems in which the customers are distributed so that the colonies of nodes can be easily identified using human visual perception. In such cases, the time and the cost spent for solving the problem using crowdsourcing techniques will be far lower than other techniques. Also, there are cases in which, several parameters such as distance, cost, time, geographical barriers, forbidden areas, etc., should be taken into account when solving the problem. In such cases, humans are more likely to perform better than machines, due to their visual perception.

To bring a more clear example, assume that a telecommunication company is going to provide mobile coverage for the Manicouagan Reservoir, an annular lake in central Quebec, Canada, and a tourist attraction in the region. As depicted in the Figure 9, the area is full of rivers and lakes, with haphazard borderlines. In such a case, to be able to use automated heuristics such as genetic algorithms, etc., one should define these lakes and rivers in the form of obstacles and forbidden areas [21] on the map. It is evident that defining such stochastic areas in the computer applications is almost impossible, but it is very easy for a human to distinguish rivers and lakes from lands, in order to recommend facility locations.

SolveIt, in its current form, might have difficulties with solving problems in which the number of facilities and locations is too high and customers are scattered on the plane without any visual order, which is not the case in most real-world problems. However, this approach can be adopted in a very broad range of problems, particularly optimization problems.

Last but not the least, incentives are very important in the success of games. In SolveIt, all players receive attribution and also a certificate of contribution and appreciation from the *Vice Chancellor for Research*. The winners, also, receive a 16 GB USB flash, as a prize. Generally speaking, the motivation of players might depend on the type of the problem that is being solved. For commercial problems a monetary reward might work better than an intrinsic one, while in some other cases only attribution or giving points, badges, etc., might be enough. Also, the type of incentives might depend on the type of players, i.e., the age, income, expertise, etc. So, the type of incentives is more problem and player dependent, and varies from game to game.

6.2 Threats to Validity

One of the major challenges that SolveIt as a crowdsourcing game might face is the validity of the crowd contributions. Crowd workers/contributors might provide low quality contributions for several reasons such as lack of expertise or misunderstanding. They might even show dishonest behaviour due to their personal or group interests or benefits. So, the quality of the contributors/workers is one of the threats to quality. This can be addressed by employing quality assurance techniques, such as reputation management, to make sure that suitable players are employed [12].

The main advantage of crowdsourcing games lies in their ability to harvest the wisdom of the crowd. In our experiment we used 40 participants (university students) in a lab setting without using an openly accessible, commercial crowdsourcing platform to recruit participants. While this may contrast with a more

conventional interpretation of crowdsourcing, we consider the number of participants big enough to draw conclusions on the effectiveness of the game design. On the other hand, making the game accessible to an open crowd via commercial platforms is straightforward and requires just redirecting them to the game from within the crowdsourcing platform.

A threat to the validity of the results described in this article that may arise is failing to recruit enough number of players. When the number of players decreases in a game, the chance to reach the optimum point for the game also decreases. To overcome this challenge, it is thus imperative to use suitable incentives to make potential players interested in playing the game. Incentives are problem and player dependent, hence, should be selected carefully. Non-suitable incentives can lead to less suitable workers or low quality contributions, according to literature [12,1]

7 Related Work

As we propose a crowdsourcing game-theoretic solution for solving p-median problems, we study the related literature in two parts: *location problems* and *online crowdsourcing games*.

7.1 Location Problems

Modern location theory is usually said to have begun with Weber's treatise (*Über den Standort von Industrien*) on the location of industries. The general location problem is: given a set of facility locations and a set of customers who are served from the facilities, thus the questions are:

- Which facilities should be used?
- Which customers should be served from which facilities so as to minimize the total cost of serving all the customers?

The field of location theory consists of four primary problems: the *p-median problem*, the *p-center problem*, the *uncapacitated facility location problem (UFLP)* and the *quadratic assignment problem (QAP)* [38]. A detailed description of location problems can be found in Mirchandani and Francis [38] and Drezner and Hamacher [18]. We now define the terms of p-median and p-center as follows:

(p-median problem) Given a metric space $(X, ||.||)$ with distance function $||.|| : X \times X \to \mathbb{R}^+$. Let finitely many points $P_1, P_2, ..., P_n \in X$ and multipliers $w_1, w_2, ..., w_n \in \mathbb{R}$ be given. Determine p new points $u_1, u_2, ..., u_p \in X$, called p-median, such that

$$\sum_{i=1}^{n} w_i \min_{j=1,...,p} ||P_i - u_j||$$

becomes minimum.

(p-center problem) Given a metric space $(X, ||.||)$ with distance function $||.|| : X \times X \to \mathbb{R}^+$. Let finitely many points $P_1, P_2, ..., P_n \in X$ and multipliers $w_1, w_2, ..., w_n \in \mathbb{R}$ be given. Determine p new points $u_1, u_2, ..., u_p \in X$, called p-center, such that

$$\max_{1 \le i \le n} w_i \min_{1 \le j \le p} ||P_i - u_j||$$

becomes minimum.

7.2 Discrete Median and Center Problems

Given a connected graph $G = (V, E)$ with vertex set $V = \{v_1, v_2..., v_n\}$, edge set E with $|E| = m$, a distance function $d : V \times V \rightarrow \mathbb{R}^+$, and a constant $p \leq n$. Every vertex has some weight w_i, $i = 1, 2, ..., n$. In the classical case the weights w_i, $i = 1, 2, ..., n$, are positive. Assume that all edges have a positive length l_j, $j = 1, 2, ..., m$.

Classical p-Median and p-Center Problems

Hakimi [25,27] introduced the following discrete p-median and p-center problem. Determine p facilities $u_1, u_2, ...u_p$, called p-median, to minimize the sum of the distances from each customer to its closest facility:

$$\sum_{i=1}^{n} w_i \min_{j=1,...,p} d(v_i, u_j).$$

This problem is a well-known $\mathcal{NP}$-hard problem [33]. Hakimi showed that there is always a collection of p vertices that minimizes the objective. The objective of the p-center problem is to locate p new facilities $x_1, x_2, ..., x_p$, called centers, on G in order to minimize the maximum weighted distance between a node and its nearest facility,

$$\min_{\substack{X \subseteq G \\ |X|=p}} \max_{1 \leq i \leq n} w_i \min_{1 \leq j \leq p} d(v_i, x_j).$$

For the p-median problem in networks with nonnegative weights it has been proved by Hakimi [25] that the search for an optimal solution of the p-median problem can be restricted to the set of vertices. This property is known as *vertex optimality property*. The vertex optimality property does not hold for p-center problems even for the case $p = 1$ and if all weights are nonnegative. For the network case the $\mathcal{NP}$-hardness of the p-median and p-center problems has been shown by Kariv and Hakimi [32,33]. Both problems remain $\mathcal{NP}$-hard even for the special case where all weights w_i and all edge lengths are equal to one. The p-center problem also remains $\mathcal{NP}$-hard if the variant is considered where the facilities are only allowed to be placed on vertices.

Obnoxious and Semi-Obnoxious p-Median and p-Center Problems

A location problem in which all clients are associated with negative weights is called obnoxious facility location problem. If the facilities are obnoxious only for a subset of the clients, then the resulting problem class is referred to as semi-obnoxious facility location problems. For the case $p \geq 2$ different types of objective functions for semi-obnoxious p-median problems can be investigated. Burkard, Çela and Dollani [7] deal with two different models. In the first model (P_1) the sum of the minimum weighted distances over all $X \subseteq G$ with $|X| = p$ is minimized

$$F_1(X) = \sum_{i=1}^{n} \min_{1 \leq j \leq p} (w_i d(x_i, v_i))$$

In the second model (P_2) the sum of weighted minimum distances over all $X \subseteq G$ with $|X| = p$ is minimized

$$F_2(X) = \sum_{i=1}^{n} w_i \min_{1 \leq j \leq p} d(x_i, v_i)$$

If all weights w_i, $0 \leq i \leq n$, are positive, then $F_1(X) = F_2(X)$ for all X and both problems are identical to the classical p-median problem. For $p = 1$ both objective functions F_1 and F_2 coincide with the objective function of the 1-median problem.

The literature contains results for obnoxious p-center location problems and related problems with several different types of objective functions. For more details on variants of obnoxious p-center problems we refer the reader to the comprehensive survey paper of Cappanera [8]. Semi-obnoxious p-center problems for case $p \geq 2$ are an attractive topic of research.

7.3 Online Crowdsourcing Games

In the area of incentive design, one of the most popular developments in recent years has been titled as gamification. Gamification is defined as a process of enhancing services with (motivational) affordances in order to invoke gameful experiences and further behavioral outcomes [29]. Tomnod[5] is an example which uses images taken by Digital Globe satellites to pinpoint objects and places in the aftermath of natural disasters and man-made catastrophes. Their most recent search (launched in March 2014) is focused on finding the missing Malaysia Airlines plane, flight MH370, in the Indian Ocean. Genes in Space [6] is a mobile game that uses the collective force of players to analyze real genetic data to help with cancer research. Smorball [7] is another game in which, players are presented with phrases from scanned pages in the Biodiversity Heritage Library and they are asked to type the words they see as quickly and accurately as possible. The result makes historic literature more usable for institutions, scholars, educators, and the public.

Several incentive mechanisms can be employed in human centric participatory systems [20,43]. More specifically, there has recently been work addressing incentives of crowdsourcing contests from game-theoretic perspectives by modeling these contests as auctions [14,4,37,43]. In an auction, an auctioneer requests bids and provides a good in exchange for payment from a winner selected by the auctioneer. A widely used form among these auctions is all-pay auction in which every bidder must pay regardless of whether they win the prize, which is awarded to the highest bidder as in a conventional auction. In an all-pay auction, the Nash Equilibrium is such that each bidder plays a mixed strategy and their expected pay-off is zero. In another research inspired by Nash Equilibrium, a game-theoretic approach has been proposed to improve quality of contributions in crowdsourcing tasks based on real-time feedbacks and agreements received from other players [2].

[5] http://www.tomnod.com/
[6] http://genesinspace.org/
[7] http://smorballgame.org/

Gamification has also been used in crowdsourcing [39]. In [19], authors have proposed a game called Wordsmith. Wordsmith is a single player game in which players attend in an image labeling task. Players receive feedback based on their contributions. While feedbacks in Wordsmith motivate users to improve their behaviour, they do not have the opportunity to change their mind, and this is how our proposed game is different from Wordsmith. In another similar game called Game of Words [22], authors propose a game for collecting relevant keywords about a specific location. Game of Words is a single player game in which players cannot change their mind and do not receive any online support, while our proposed game tries to improve quality by relying on the opinions of two players and giving them feedback and the opportunity to change their choice. Moreover, authors in [23,44] rely on the crowdsourcing to solve maximization optimization problems.

In this paper, we propose a game, called SolveIt, for solving planar p-Median problems. SolveIt benefits from both crowdsourcing and gamification advantages to retain and motivate workers. To the best of our knowledge, this is the first time that the idea of using crowdsourcing games for solving location allocation problems is proposed.

8 Conclusion

In this paper, we propose to solve planar P-Median problems as a popular location allocation problem using crowdsourcing and gamification techniques. We present a game called SolveIt by which location allocation problems are converted into online games and crowd workers are asked to solve them while they play games. The performance of the game in terms of accuracy of the results is evaluated and the results show the viable performance of the game.

Although SolveIt is proposed for location allocation problems, the intuition behind the game is applicable to a broad range of problems, namely optimization problems. The advantages of crowdsourcing techniques, combined with the benefits of gamification and the data model and formulations proposed in this research, makes SolveIt a good candidate for solving problems in domains such as clustering, linear optimization, graph traversal and processing, and so on, the areas in which we plan to focus more in our future research.

References

1. M. Allahbakhsh, B. Benatallah, A. Ignjatovic, H. R. Motahari-Nezhad, E. Bertino, and S. Dustdar. Quality control in crowdsourcing systems: Issues and directions. *IEEE Internet Computing*, 17(2):76–81, 2013.
2. Mohammad Allahbakhsh, Haleh Amintoosi, and Salil S. Kanhere. A game-theoretic approach to quality improvement in crowdsourcing tasks. In Amin Beheshti, Mustafa Hashmi, Hai Dong, and Wei Emma Zhang, editors, *Service Research and Innovation*, pages 116–130, Cham, 2018. Springer International Publishing.
3. Alexander N Antamoshkin and Lev A Kazakovtsev. Random search algorithm for the p-median problem. *Informatica*, 37(3), 2013.
4. Nikolay Archak and Arun Sundararajan. Optimal design of crowdsourcing contests. *ICIS 2009 Proceedings*, page 200, 2009.
5. Jack Brimberg and Zvi Drezner. A new heuristic for solving the p-median problem in the plane. *Computers & Operations Research*, 40(1):427–437, 2013.

6. Jack Brimberg, Pierre Hansen, Nenad Mladenović, and Eric D Taillard. Improvements and comparison of heuristics for solving the uncapacitated multisource weber problem. *Operations Research*, 48(3):444–460, 2000.
7. Rainer E Burkard, Eranda Çela, and Helidon Dollani. 2-medians in trees with pos/neg weights. *Discrete Applied Mathematics*, 105(1):51–71, 2000.
8. Paola Cappanera. A survey on obnoxious facility location problems, 1999.
9. Leon Cooper. Location-allocation problems. *Operations research*, 11(3):331–343, 1963.
10. Leon Cooper. Heuristic methods for location-allocation problems. *SIAM review*, 6(1):37–53, 1964.
11. Seth Cooper, Firas Khatib, Adrien Treuille, Janos Barbero, Jeehyung Lee, Michael Beenen, Andrew Leaver-Fay, David Baker, Zoran Popović, et al. Predicting protein structures with a multiplayer online game. *Nature*, 466(7307):756–760, 2010.
12. Florian Daniel, Pavel Kucherbaev, Cinzia Cappiello, Boualem Benatallah, and Mohammad Allahbakhsh. Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions. *ACM Comput. Surv.*, 51(1):7:1–7:40, January 2018.
13. Mark S. Daskin and Kayse Lee Maass. *The p-Median Problem*, pages 21–45. Springer International Publishing, Cham, 2015.
14. Dominic DiPalantino and Milan Vojnovic. Crowdsourcing and all-pay auctions. In *Proceedings of the 10th ACM conference on Electronic commerce*, pages 119–128. ACM, 2009.
15. Anhai Doan, Raghu Ramakrishnan, and Alon Y Halevy. Crowdsourcing systems on the world-wide web. *Communications of the ACM*, 54(4):86–96, 2011.
16. Steven Dow, Anand Kulkarni, Scott Klemmer, and Björn Hartmann. Shepherding the crowd yields better work. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 1013–1022. ACM, 2012.
17. Zvi Drezner, Jack Brimberg, Nenad Mladenović, and Said Salhi. New heuristic algorithms for solving the planar p-median problem. *Computers & Operations Research*, 62:296–304, 2015.
18. Zvi Drezner and Horst W Hamacher. *Facility location: applications and theory*. Springer Science & Business Media, 2001.
19. Oluwaseyi Feyisetan, Elena Simperl, Max Van Kleek, and Nigel Shadbolt. Improving paid microtasks through gamification and adaptive furtherance incentives. In *Proceedings of the 24th International Conference on World Wide Web*, pages 333–343. ACM, 2015.
20. H. Gao, C. H. Liu, W. Wang, J. Zhao, Z. Song, X. Su, J. Crowcroft, and K. K. Leung. A survey of incentive mechanisms for participatory sensing. *IEEE Communications Surveys Tutorials*, 17(2):918–943, Secondquarter 2015.
21. Mitsuo Gen, Runwei Cheng, and Lin Lin. *Network models and optimization: Multiobjective genetic algorithm approach*. Springer Science & Business Media, 2008.
22. Jorge Goncalves, Simo Hosio, Denzil Ferreira, and Vassilis Kostakos. Game of words: Tagging places through crowdsourcing on public displays. In *Proceedings of the 2014 Conference on Designing Interactive Systems*, DIS '14, pages 705–714, New York, NY, USA, 2014. ACM.
23. Stephen Guo, Aditya Parameswaran, and Hector Garcia-Molina. So who won?: Dynamic max discovery with the crowd. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, SIGMOD '12, pages 385–396, New York, NY, USA, 2012. ACM.
24. S Louis Hakimi. Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations research*, 12(3):450–459, 1964.
25. S Louis Hakimi. Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations research*, 12(3):450–459, 1964.
26. S Louis Hakimi. Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Operations Research*, 13(3):462–475, 1965.
27. S Louis Hakimi. Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Operations Research*, 13(3):462–475, 1965.
28. Juho Hamari, Jonna Koivisto, and Harri Sarsa. Does gamification work?–a literature review of empirical studies on gamification. In *System Sciences (HICSS), 2014 47th Hawaii International Conference on*, pages 3025–3034. IEEE, 2014.
29. Kai Huotari and Juho Hamari. Defining gamification: a service marketing perspective. In *Proceeding of the 16th International Academic MindTrek Conference*, pages 17–22. ACM, 2012.
30. Jorge H. Jaramillo, Joy Bhadury, and Rajan Batta. On the use of genetic algorithms to solve location problems. *Computers & Operations Research*, 29(6):761 – 779, 2002. Location Analysis.

31. Hongzhong Jia, Fernando Ordonez, and Maged M Dessouky. Solution approaches for facility location of medical supplies for large-scale emergencies. *Computers & Industrial Engineering*, 52(2):257–276, 2007.
32. Oded Kariv and S Louis Hakimi. An algorithmic approach to network location problems. i: The p-centers. *SIAM Journal on Applied Mathematics*, 37(3):513–538, 1979.
33. Oded Kariv and S Louis Hakimi. An algorithmic approach to network location problems. ii: The p-medians. *SIAM Journal on Applied Mathematics*, 37(3):539–560, 1979.
34. Aniket Kittur, Boris Smus, Susheel Khamkar, and Robert E Kraut. Crowdforge: Crowdsourcing complex work. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 43–52. ACM, 2011.
35. Gilbert Laporte, Stefan Nickel, and Francisco Saldanha da Gama. *Introduction to Location Science*, pages 1–18. Springer International Publishing, Cham, 2015.
36. Greg Little, Lydia B Chilton, Max Goldman, and Robert C Miller. Turkit: human computation algorithms on mechanical turk. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 57–66. ACM, 2010.
37. Tie Luo, Salil S Kanhere, Sajal K Das, and Hwee-Pink Tan. Optimal prizes for all-pay contests in heterogeneous crowdsourcing. In *Mobile Ad Hoc and Sensor Systems (MASS), 2014 IEEE 11th International Conference on*, pages 136–144. IEEE, 2014.
38. Pitu B Mirchandani. The p-median problem and generalizations. *Discrete location theory*, 1:55–117, 1990.
39. Benedikt Morschheuser, Juho Hamari, and Jonna Koivisto. Gamification in crowdsourcing: a review. In *2016 49th Hawaii International Conference on System Sciences (HICSS)*, pages 4375–4384. IEEE, 2016.
40. Joseph J. Pfeiffer, III, Sebastian Moreno, Timothy La Fond, Jennifer Neville, and Brian Gallagher. Attributed graph models: Modeling network structure with correlated attributes. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 831–842, New York, NY, USA, 2014. ACM.
41. Josh Reese. Solution methods for the p-median problem: An annotated bibliography. *Networks*, 48(3):125–142, 2006.
42. Jakob Rogstadius, Vassilis Kostakos, Aniket Kittur, Boris Smus, Jim Laredo, and Maja Vukovic. An assessment of intrinsic and extrinsic motivation on task performance in crowdsourcing markets. In *ICWSM*, 2011.
43. Stefan Stieglitz, Christoph Lattemann, Susanne Robra-Bissantz, Rüdiger Zarnekow, and Tobias Brockmann. *Gamification*. Springer, 2017.
44. Petros Venetis, Hector Garcia-Molina, Kerui Huang, and Neoklis Polyzotis. Max algorithms in crowdsourcing environments. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 989–998, New York, NY, USA, 2012. ACM.
45. Endre Weiszfeld. Sur le point pour lequel la somme des distances de n points donnés est minimum. *Tohoku Mathematical Journal, First Series*, 43:355–386, 1937.
46. Peter Welinder and Pietro Perona. Online crowdsourcing: rating annotators and obtaining cost-effective labels. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 25–32. IEEE, 2010.