applied sciences

MDPI

Article

Complementing Solutions for Facility Location Optimization via Video Game Crowdsourcing and Machine Learning Approach

Mariano Vargas-Santiago [1,†], Diana A. León-Velasco [2,*,†], Ricardo Marcelín Jiménez [1,†] and Luis Alberto Morales-Rosales [3,†]

[1] Department of Electrical Engineering, Universidad Autónoma Metropolitana, Unidad Iztapalapa, Ciudad de México 09340, Mexico; mariano.v.santiago@gmail.com (M.V.-S.); rmarcelin@izt.uam.mx (R.M.J.)
[2] Department of Applied Mathematics and Systems, Universidad Autónoma Metropolitana, Unidad Cuajimalpa, Ciudad de México 05348, Mexico
[3] CONACYT-Universidad Michoacana de San Nicolas de Hidalgo, Facultad de Ingeniería Civil, Morelia 58000, Mexico; lamorales@conacyt.mx
* Correspondence: dleon@cua.uam.mx
† These authors contributed equally to this work.

Abstract: The facility location problem (FLP) is a complex optimization problem that has been widely researched and applied in industry. In this research, we proposed two innovative approaches to complement the limitations of traditional methods, such as heuristics, metaheuristics, and genetic algorithms. The first approach involves utilizing crowdsourcing through video game players to obtain improved solutions, filling the gap in existing research on crowdsourcing for FLP. The second approach leverages machine learning techniques, specifically prediction methods, to provide an efficient exploration of the solution space. Our findings indicate that machine learning techniques can complement existing solutions by providing a more comprehensive approach to solving FLP and filling gaps in the solution space. Furthermore, machine learning predictive models are efficient for decision making and provide quick insights into the system's behavior. In conclusion, this research contributes to the advancement of problem-solving techniques and has potential implications for solving a wide range of complex, NP-hard problems in various domains.

Keywords: optimization; facility location problems; genetic algorithms; predictive models

Citation: Vargas-Santiago, M.; León-Velasco, D.A.; Marcelín Jiménez, R.; Morales-Rosales, L.A. Complementing Solutions for Facility Location Optimization via Video Game Crowdsourcing and Machine Learning Approach. *Appl. Sci.* **2023**, *13*, 4884. https://doi.org/10.3390/app13084884

Academic Editors: Chuan-Ming Liu and Wei-Shinn Ku

Received: 10 February 2023
Revised: 22 March 2023
Accepted: 23 March 2023
Published: 13 April 2023

1. Introduction

Given the ubiquity of computationally hard, complex, or difficult problems, using human insight and intuition allows us to improve the traditional algorithmic methods [1,2]. In this work, we applied this concept to the optimization problem known as the facility location problem (FLP), utilizing video games as the means to gather human input on instances of it.

The FLP consists of, given a set of demand centers and potential locations for opening facilities, choosing a subset of the potential locations. Opening each facility has a cost associated, and servicing demand centers from faraway facilities is costly as well. Strategies may vary from having few facilities that service distant demand centers to having numerous facilities that service demand centers from short distances. The subset must be chosen to minimize the total cost. This problem has been proven to be NP-hard [3].

The utilization of crowd computing techniques is employed as a method to gather large amounts of human input for problem instances. Crowd computing is defined as a strategy that allows a collective group of individuals, rather than solely computers, to perform productive computations and aggregate the results in order to solve a problem. A comprehensive discussion about the understanding of crowdsourcing in science was proposed by Lenart-Gansiniec et al. [4] since it is a topic that is relevant for generating

scientific knowledge and has been used to solve problems for business, the public, and non-governmental sectors.

The FLP is a common issue that frequently occurs in the fields of logistics and operations research. Despite its ubiquitousness, the complexity of the problem often renders obtaining optimal solutions for substantial instances within practical time limits challenging. For this reason, these problems are often tackled with heuristics [5], meta-heuristics, or stochastic approaches [6]. In particular, problems where a robust solution is needed—one that can endure a disruption and remain efficient—have been approached with genetic algorithms [7,8]. The aim of this study is to demonstrate the feasibility of incorporating human intuition and insight into solving complex problems such as the FLP, which frequently arise in logistics and operations research. The goal is to reduce transportation costs, enhance robustness in FLP, and allocate resources effectively. However, the task of acquiring knowledge from human participants is challenging due to the difficulty in articulating the thought processes that lead to a conclusion. People may have an intuitive sense of the solution to a particular problem instance, but their thought processes are not easily expressed in a concrete algorithm or heuristic. To address this challenge, a video game was employed to gather potential solutions from human participants. This game presents participants with multiple instances of optimization problems, aiming to aggregate their strategies into a unified algorithm or heuristic.

The objective of this study is to demonstrate the feasibility of a hybrid human–computer approach in addressing challenging problems. A comparison between the results of this hybrid approach and traditional algorithmic methods, specifically genetic algorithms, was made concerning the facility location problem. No prior expertise or training is required for the human participants since the proposed video game is designed to be accessible to the general public. This study includes the design and implementation of a video game specifically for the FLP, as well as the data analysis and interpretation of the results obtained from individuals playing the game.

While the widespread distribution of the game is not within the scope of this work, the game's design is intended to be suitable for such an objective, including the aim of increasing player engagement and gather more data per player. The design of the video game was intended to fulfill dual purposes, including providing basic training on the concepts of the problem and collecting candidate solutions. A critical aspect of the game design was that no prior knowledge or training was required to play and progress. This design choice was made for several reasons: to eliminate the costs and time associated with training participants beforehand; to ensure a broad distribution of the game to a large audience, which would be challenging to train individually; and to prevent the infusion of our own ideas, strategies, and assumptions related to the problem into the players since their unaltered perceptions and intuitions were the primary objective.

In the existing literature, various approaches have been proposed to solve the FLP problem, including the use of heuristics, metaheuristics, and genetic algorithms [7–11]. Among these, genetic algorithms are the most widely used method. However, the computational resources and execution time required to apply genetic algorithms can be substantial. The optimization problem of the FLP is known to be NP-hard, making it challenging to determine the optimality of the solutions obtained from these algorithms. Thus, there is a need for alternative methods that can reasonably approximate the solutions obtained from various sources, thereby reducing the computational resources and time required for problem-solving.

In addition, we aim to obtain approximate solutions for the Pareto front, where gaps exist in the solution space explored by genetic algorithms. To achieve this, various predictive models were employed. The calculation of the Pareto front through genetic algorithms is a computationally intensive task, which can result in the intractability of combinatorial optimization problems. This is due to the need to search through all possible combinations to find the solution with the minimum value. However, genetic algorithms

may often become trapped in a local optimum, which may also be dominated by a previous solution on the Pareto front.

The obtained experimental results demonstrate that reasonable approximations of the Pareto front generated by genetic algorithms can be obtained. However, despite being sophisticated solutions, they may not cover all points in some cases. To address this limitation, it is necessary to provide decision makers with tools or methods that present a complete set of solutions for the Pareto front for a given dataset of interest. To achieve this, we complemented the incomplete solutions generated by genetic algorithms by utilizing machine learning principles, specifically by applying various predictive models. The results indicate that reasonable solutions can be obtained, but the precision of the approximations depends on the performance of the predictive model used. This study utilized cubic splines, exponential regressions, K-nearest neighbors (KNN), and multiple imputations by chained equations (MICE) as machine learning principles.

This paper makes the following contributions:

1. A video game through which the general public can learn about a multi-objective optimization problem, namely the FLP. While playing, people learn about multi-objective optimization.
2. The use of crowd computing as a means to collect solutions to complex problems. These crowdsourced solutions complement the coverage of popular methods, such as genetic algorithms.
3. Fast and efficient machine learning principles based on predictive models complementing the genetic algorithm, even for large instance problems. Decision makers who require immediate solutions can benefit from these approximations.

The organization of this work is structured as follows: Section 2 provides the necessary background information relevant to the problem being addressed, as well as the datasets utilized in the experiments. Section 3 discusses the relevant previous work. Section 4 outlines the implementation of the proposed video game. Section 5 presents the use of machine-learning-based predictive models. The analysis of the results is presented in Section 6. Finally, the conclusion and recommendations for future research are included in Section 7.

2. Background

The facility location problem is a combinatorial optimization problem pertaining to operations, distribution, and logistics. The problem involves finding the optimal placement of facilities in a set of demand centers to minimize the total cost. The demand centers represent locations with varying degrees of demand for a service, and facilities must be placed at these locations to service the demand. The cost of operating a facility depends on the distance between the facility and its corresponding demand center, as well as the fixed cost of operating the facility. A solution to the FLP is a set of facility locations that minimize the total cost, where each demand center can have at most one facility within it.

Examples of the FLP can be found in various real-world scenarios. For example, in the context of city management, the FLP can arise in the placement of fire department facilities, where they must consider the costs of maintaining equipment, firefighters, and the response time. In this case, the demand centers can be the neighborhoods within the city, each with a different demand value corresponding to the neighborhood's population. Another example is the arrangement of networked sensors in a space to minimize the number of sensors required while maximizing the coverage area [12,13].

Formally, the FLP can be defined as three distinct multi-objective optimization problems [7], where:

D: Set of all demand centers;
F: Set of all candidate facilities;
G: Set of open facilities;
W: Set of open facilities that did not fail;

P(W): Power set of W;
f_i: Cost of opening a facility at location i;
c_{ij}: Cost of assigning demand center j to facility i;
y_i: Binary decision variable that indicates whether facility i is open;
x_{ij}: Binary decision variable that indicates whether location j is assigned to facility i;
u_{ij} : Binary decision variable that indicates whether location j is assigned to facility i after failures have occurred;
v_i: Binary decision variable that indicates whether open facility i has failed.

1. First subproblem:
   (a) $minimize_y \sum_{i \in F} f_i y_i$;
   (b) $minimize_x \sum_{i \in F} \sum_{j \in D} d_j c_{ij} x_{ij}$.

   The first objective function minimizes the cost of opening facilities, and the second objective function minimizes the total distance without failures. These two objectives allow decision makers to understand the impact of one objective on another (i.e., how opening more facilities reduces the total distance).
2. Second subproblem:
   (a) $minimize_v \sum_{m \in G} f_m v_m$;
   (b) $maximize_u \sum_{k \in W} \sum_{j \in D} d_j c_{kj} u_{kj}$;
   (c) subject to:
      i. $y_m - v_m \leq k$;
      ii. $v_m \in \{0, 1\} \forall m \in G$;
      iii. $u_{kj} \leq x_{kj} \forall k \in W, j \in D$;
      iv. $\sum_{k \in W} u_{kj} = 1, \forall j \in D$.

   The first objective function minimizes the number of facilities that fail. The second objective function maximizes the distance after failures. Constraint *i* ensures that only open facilities can fail. Constraint *ii* ensures that v is a binary decision variable. Constraint *iii* ensures that demand centers are reassigned to open facilities, and constraint *iv* ensures that demand centers are reassigned.
3. Third subproblem:
   (a) $minimize_x \sum_{i \in F} \sum_{j \in D} d_j c_{ij} x_{ij}$;
   (b) $minimize_u \sum_{k \in W} \sum_{j \in D} d_j c_{kj} u_{kj}$.

   The first objective function minimizes the distance before failure and the second objective function minimizes the distance after failure.

Datasets

The primary inputs for the FLP problem are the set of demand centers and candidate facilities. The data sets used in this study include the Swain and London datasets (referred to as the benchmark in the FLP literature), which are depicted in Figures 1 and 2, respectively. The coordinates of the demand centers are displayed in each figure, with the size of each point proportional to the demand value. Each point also signifies the potential location to open a facility. The weighted Euclidean distance was used to calculate the total distance between a demand center and an open facility.

The Swain dataset, comprising 55 nodes with location coordinates and user population data, has been a valuable resource for researchers in the transportation network optimization domain. This dataset, which provides an approximation of air traveler distribution based on origin and destination data in the Washington, DC area during the 1960s [14], has been utilized in recent studies to carry out simulations and solve the FLP. The Swain dataset has been a popular choice among researchers due to its ability to support a search

tree guided process, making it well-suited for benchmarking and evaluating algorithmic solutions to the FLP. For instance, Church et al. [15] recently used the Swain dataset in proposing several approaches to address the p-median problem, comparing their results to the Swain dataset. The continued use of this dataset in recent research highlights its enduring value and utility in the scientific community.

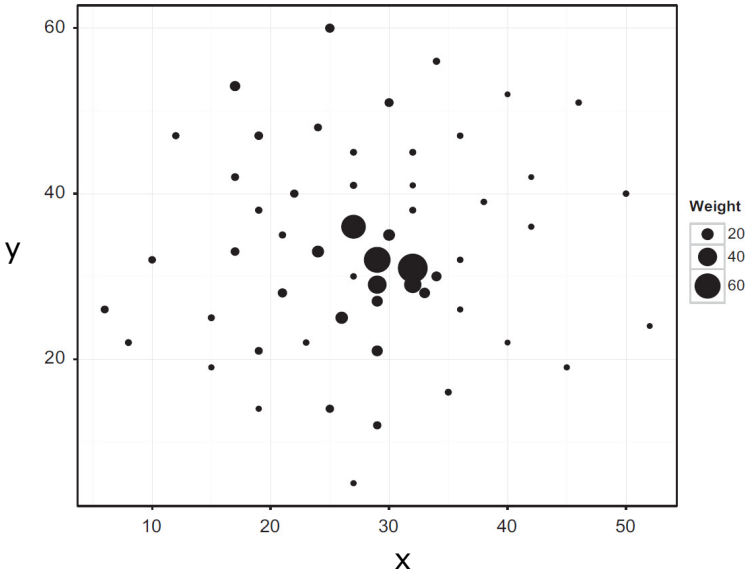

Figure 1. Swain's data set is one of the most used sets in the literature and benchmarks.

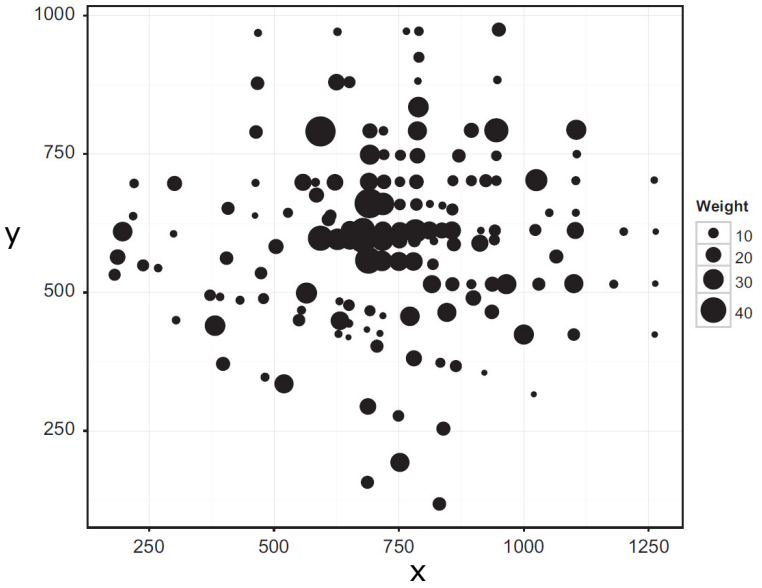

Figure 2. London data set is one of the most used sets in the literature to represent problems with large instances of facilities.

The use of Swain and London datasets is still relevant in ongoing research due to several reasons. First, these datasets serve as benchmark problems in optimization and machine learning, allowing researchers to evaluate and compare the performance of different algorithms and models. The Swain dataset in particular has been used to study several optimization problems, such as facility location, clustering, and graph partitioning.

Second, the Swain dataset is a well-known benchmark problem in the field of community detection, which is the task of identifying densely connected subgraphs in a network. This problem is of great importance in various applications, such as social network analysis,

epidemiology, and transportation planning. Due to its large size and heterogeneous structure, the Swain dataset provides a realistic testbed for community detection algorithms.

Third, the Swain dataset, as well as the London dataset, is often used in transportation planning research to model the flow of people and goods between different locations. Despite being based on data from the 1960s, these datasets still provide valuable insights into the fundamental principles of transportation systems and can be used to develop and test new models and algorithms.

The Swain (Figure 1) and London (Figure 2) datasets remain relevant in ongoing research due to their importance as benchmark problems in optimization, community detection, and transportation planning. Their use allows researchers to develop and evaluate new algorithms and models and to gain insights into the underlying principles of complex systems.

The Swain and London datasets were utilized in our study due to their ability to provide simulations of populations with varying levels of demand and population density. These datasets contain location coordinates denoted by 'x' and 'y' values, which can be substituted with actual geospatial data such as building, hospital, or park locations for analysis. Additionally, the datasets include a 'weight' field that represents population density, with higher weights indicating a greater population concentration in a given demand center. We utilized Euclidean distance calculations to simulate travel costs for individuals or groups seeking to access facilities located within demand centers.

3. Related Works

This research proposes the application of video games as a means of solving optimization problems in a non-monetary context, where participant motivation is driven by entertainment value. To ensure the reliability and validity of user inputs, controlled playtests were conducted, and game levels were designed to be brief in duration, thus minimizing the required participation time for each dataset while maintaining user engagement. The optimization problem was presented in the context of a game and a tutorial was provided to reduce the cognitive load required for understanding the problem, further enhancing player engagement.

In this section, we provide a summary of the concepts of human-based computation, video games with a purpose, and crowdsourcing, as these techniques were combined in the current work.

3.1. Human-Based Computation

In the literature, previous research has been conducted to address the integration of human and computer interaction in solving various optimization problems. For instance, the capacitated vehicle routing with time windows problem has been explored through the use of an interactive graphical interface [2]. Additionally, studies have been conducted to evaluate the effectiveness of combining the strengths of both the human and computer participants [16]. Results have shown that a combination of human and computer agents in teams is more effective than a team of only humans in military command and control tasks [17]. Interactive genetic algorithms have been applied to information retrieval problems [18] and in software design [19].

Compared to the interface-based approach, using a video game has several advantages. Implementing the interface-based approach requires the user's training, which acts as the guide and operator of the algorithm. This requirement limits the pool of potential subjects. On the other hand, a video game, with its level-based structure, eliminates the need for training and reduces the cognitive load on the player. The player's sole focus is on progressing through the levels of the game without the added responsibility of directing the algorithm.

3.2. Crowd Computing

Crowd computing, also known as citizen science, is a novel approach that leverages the abilities of ordinary individuals to solve diverse problems. This concept is characterized by the gathering of vast amounts of data, their analysis, and the processing of information to identify effective solutions to these problems [20–22]. Crowd computing is similar to cloud computing, as it seeks to be an accessible, dynamic, and available computational resource. However, it differs from cloud computing in that it utilizes human cognitive abilities rather than central processing units to carry out its computations.

We acknowledge that crowd computing and cloud computing are two different paradigms for solving optimization problems. Although both approaches utilize a large number of computers for computation, they have fundamental differences. Crowd computing involves leveraging human intelligence to solve complex problems that cannot be addressed by computers alone. This approach requires the recruitment of a large number of individuals who perform small tasks that are then combined to solve a larger problem. Crowd computing is typically used for tasks that require human interpretation and processing of data, such as image labeling, data entry, and transcription. In contrast, cloud computing involves the use of a network of computers to provide on-demand computing resources for data storage, data processing, and software development. The data and applications are stored on remote servers, which can be accessed over the Internet. This approach is suitable for tasks that require large-scale computation and data processing.

The motivation of a large number of participants to solve complex problems is often incentivized through the provision of monetary rewards. However, negotiating appropriate compensation, promoting collaboration among participants, and effectively managing communication between workers and organizers can pose challenges [23]. For instance, mobile crowd-sensing applications have faced reluctance from participants due to perceived high energy and bandwidth consumption costs [24].

Aside from monetary compensation, entertainment can also motivate individuals to participate in crowd computing. Problem-solving tasks, particularly in the form of puzzle games, can be intellectually challenging and provide a sense of satisfaction upon completion. Studies have demonstrated that incorporating puzzle elements into computer-based tasks can engage young people more productively with technology [25].

It is proposed that framing complex computational problems as engaging puzzle games can result in developing a crowd-computing platform capable of massive data acquisition. The motivation for participation in such games is often entertainment, and several examples of popular crowdsourcing games exist, such as Trivia Crack, King of Thieves, and Super Mario Maker by Nintendo. These games often induce player engagement through social pressure to try the game, as evidenced by their commercial success.

It has been observed that various phenomena can be conceptualized as implementations of the principles of crowd computing, including the popularity of social news websites such as Reddit and Digg [26], the analysis of semantic meaning derived from crowdsourced tags assigned to books [27], and the influence that audience participation has on the creation, dissemination, curation, and financing of music [28]. Crowds can be utilized for both formulating features and as sources of data for machine science applications [29].

The quality of contributions by participants in crowd computing poses a challenge, particularly when untrained individuals generate data. To determine the most appropriate data to be taken into consideration, a ranking system based on weights can be implemented, where the data submitted by participants with the highest classification are assigned greater weights. Alternatively, a voting system can be established, where a group of scientists assigns scores to the data. This approach generates profiles of the skills of each scientist based on dimensions such as accuracy, completeness, reputation, relevance, and others [30]. The trustworthiness of a participant's rating can be established by evaluating the objectivity of participants and the degree of consensus among them [31]. The integration of conflicting preferences of participants in solving problems can be facilitated through the use of social choice theory, enabling the selection of a single solution from a set of candidates [32].

In the context of control systems with significant safety considerations, such as the operation of heavy machinery, physiological data can be utilized to estimate the operator's mental workload, thereby providing insights into the operator's mental state during the task of collaboration with a computer [33]. This information can be used to determine if a task is overly simple or excessively challenging for a particular participant.

Recent works have developed new ways of solving allocation problems with crowdsourcing and gamification techniques. For instance, Allahbakhsh et al. [34] presented a solution to solve the p-median problems using crowdsourcing and gamification techniques. They developed a crowdsourced game called SolveIt, which employs the wisdom and intelligence of the crowd to solve location allocation problems. SolveIt uses the attention technique by showing the winner's name on the scoreboard and a competition technique to increase players' motivation. They proposed a graph data model to represent the location allocation problems. They asked a group of 40 students at the University of Zabol to participate in the game and solve the proposed problems. The contributions received from the crowd were compared to those obtained from a specific implementation of genetic algorithms, which was used as the gold standard.

Jiang et al. [35] focused on the problem of multiple cooperative task allocation (MCTA). They used real-life relationships among users on the social network and proposed group-oriented cooperative crowd-sensing to solve the MCTA problem. They covered the solutions via group-oriented cooperation with three phases while achieving a good task cooperation quality. In phase 1, they selected a subset of users on the social network as initial leaders and directly pushed sensing tasks to them. For phase 2, they used the leaders to search for their socially connected users to model groups. Phase 3 presented the process of group-oriented task allocation for solving the MCTA problem.

Moreover, a crowdsourcing strategy and the quantum have also been utilized. For instance, Minghui Xu et al. [36] used quantum crowdsourcing schemes, in which the welfare of the requestor or worker can be maximized because quantum players share the extended strategy space and the addition of entanglement offers a new method of depicting fine-grained relationships between players. To address problems of task allocation, they presented a quantum game model for quota-oriented crowdsourcing games.

3.3. Games for Solving Problems

Games with a purpose have been developed to utilize individuals' problem-solving skills for addressing computationally challenging problems [37]. These games focus on various domains, such as biology, biochemistry, and linguistics. An example of such a game is BioGames [38], which is an image-matching game that trains its players to recognize malaria-infected red blood cells. This not only provides an engaging training program for student medical personnel but also serves as a means for crowdsourcing labeling data to train machine learning algorithms. The results of BioGames demonstrate the potential of a crowd of untrained individuals to achieve a disease diagnosis with an accuracy comparable to that of an expert.

The game Phylo [21] leverages the concept of color matching to facilitate the optimization of nucleotide sequence alignment, thus minimizing the number of mutations needed to produce a different species from an ancestor. This process enables the generation of phylogenetic trees, thereby providing deeper insights into the evolutionary relationships between species with sequenced DNA. The problem of multiple sequence alignment has been shown to be NP-complete [39]; however, Phylo serves as an example of how such problems can be transformed into engaging games.

Foldit [40] and EteRNA are two games designed to facilitate the discovery of protein folding and RNA folding mechanisms, respectively. After acquiring a basic understanding of the rules, non-expert players can predict the folding of complex molecules. The players of EteRNA vote on each other's predictions, which are later synthesized and evaluated for accuracy. Foldit allows players to automate their common sequences of actions through a feature called recipes, and through observing these recipes, new strategies have been

discovered that improve the performance of existing algorithms [20]. This demonstrates the potential of crowd computing to uncover new algorithms and heuristics for complex problems. Foldit further encodes and conveys the insights gained by players by providing them with a scripting language.

EyeWire [41,42] is a game aimed at mapping the outlines of neurons in 3D space from 2D images obtained from a retina. The player is tasked with tracing the path of a neuron in 3D space by following the edges of the 2D images. The game complements algorithmic approaches by directing the player's focus to areas where the algorithm is uncertain, thereby facilitating the generation of complete maps of neurons and their connections from images. EyeWire represents a successful example of human–computer cooperation in solving a problem, utilizing the strengths of each in areas where the other is ineffective.

Google Image Labeler (GIL), which is based on the game called ESP [43], employs a strategy of using human players to label images obtained from the web. The gameplay involves two players, who are unable to communicate with each other, attempting to propose matching labels for an image. When both players agree on the same label, it is then utilized as metadata to enhance the image search service of the company. Similar to GIL's goals and gameplay, TagATune [44,45] aimed at obtaining metadata for audio files, such as the genre, type of instruments being played, and gender of the vocalists. Both GIL and TagATune incorporate a multiplayer component to increase their entertainment value and attract more players, which also serves as a validation mechanism. With the players incentivized to agree on a label but unable to communicate except through the game's mechanics, poor solutions can be easily detected and disregarded, whether submitted intentionally or not.

The focus of several games, including JeuxDeMots [46], OnToGalaxy [47], Verbosity [48], Phrase Detectives [49], and ZombiLingo [50], is on establishing relationships between words and acquiring common sense knowledge of them. Additionally, entries on WikiData, a database supporting Wikipedia and other WikiMedia services, can be augmented, cleaned up, and edited through playing WikiData: The Game.

In a study, it was found that transforming a path-finding problem for the real-time control of robotic arms into a maze-like interactive interface could improve the operator performance [51]. This approach framed the problem in the configuration space of the robot, rather than physical space, allowing the human operator to find a path between two points while avoiding obstacles.

4. Game Implementation Details

4.1. Programming Language and Deployment

The FLP model was implemented as a video game using the Python programming language and the Kivy framework. Kivy was selected for its support for multiple platforms, including all major operating systems, and its hardware-accelerated OpenGL-based rendering system. The game was playtested on an iPad Air 2 running iOS 9.3, as the game's interface was designed primarily for direct manipulation through a touchscreen. Although Python is not a language directly supported for development on iOS, Kivy allows for deployment onto iOS devices. This was achieved by including custom Python code into an XCode project using tools provided by the Kivy project, which was then compiled, packaged, signed, and copied onto the development device for execution. The successful submission of Kivy-based games to the Apple App Store demonstrates that they can be used as a distribution method.

4.2. Facility Location Problem Game Design

Similarity with the genetic algorithm (GA) approach: the design and functionality of the first game are based on the study presented in [7]. The authors utilized genetic algorithms to solve three distinct optimization problems within the context of the FLP. These sub-problems encompass:

1. A minimization of the cost associated with opening facilities and the cost of distances between facilities and demand centers. This represents a multi-objective formulation of the FLP.
2. Determining the worst possible failure combination; that is, the set of opened facilities that result in the greatest total distance in case of failure. This sub-problem aims to evaluate the proposed solutions from the previous step and assess their performance in the context of the robust facility location problem (RFLP).
3. Finding optimal solutions to the RFLP, which involves balancing between the solution's optimality in the absence of any facility failures and its optimality after facility failures have occurred. This multi-objective optimization problem provides decision makers with options for determining the appropriate trade-off for a particular domain.

The game design includes two stages that parallel the breakdown of the FLP problem. Each stage is represented by a category of levels played in consecutive order, with each level corresponding to a single instance of the problem. The following are the two stages of the game:

1. Proposed Solution to Instances of FLP: The players are tasked with proposing a solution, which consists of a set of facilities to be opened, for instances of the FLP.
2. Improvement in Total Cost: The players are presented with solutions provided by other players and are asked to improve the total cost by choosing a set of open facilities. In this stage, the game provides a starting point for the players.

The first stage aims to capture the players' initial impressions and immediate intuitions regarding the best solution for each particular instance. It is equivalent to conducting a global search with low granularity to identify interesting neighborhoods in the solution space.

The second stage of the game explores solution space in more detail by presenting players with solutions obtained from previous iterations of stage one and previous playthroughs of the game. This stage aims to identify small incremental changes to existing solutions that reduce the total cost. The design of this stage is based on the concept of a local search with high granularity around the solutions obtained in previous iterations.

Up to this point, the game has gathered data on potential solutions to the FLP. These data include information on the impact of failures on candidate solutions as well as new assignments of demand centers to facilities made to counteract such failures. Figure 3 shows the user interface of the FLP video on an iPad Air 2. The FLP is modeled as a multi-objective optimization problem in the GA approach, reflecting the trade-off between having more facilities open versus the distance from demand centers to facilities. However, in the game, a single cost is associated with opening a facility to compute a total cost for each solution candidate. This implicit assumption of the equivalent cost may bias the players' solutions and make them focus on minimizing the total cost through a fixed number of open facilities. To counteract this potential bias, some levels feature a randomly chosen limit on the number of facilities that can be opened, encouraging the players to explore the entire solution space. This distinction between separate stages allows for cooperation between the GA approach and the game, as the crowdsourced results from the game can be incorporated into the algorithm.

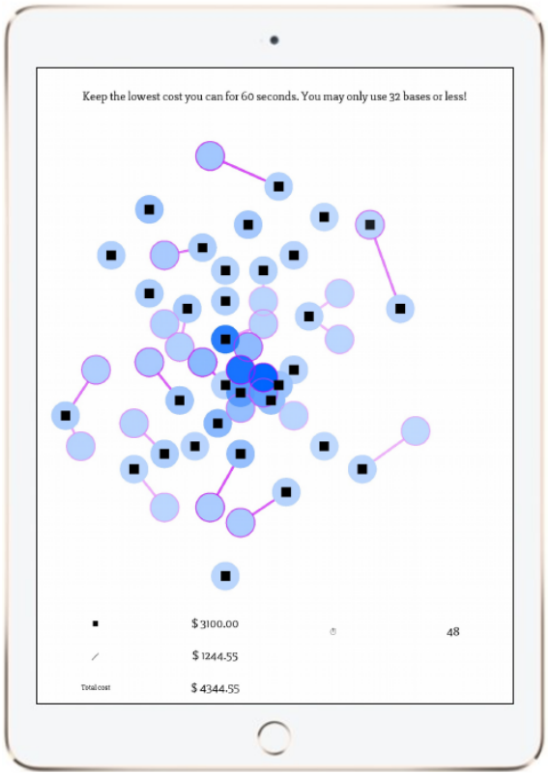

Figure 3. FLP video game showing the user interface.

1. In designing the user interface (UI) for the game, it was decided to maintain abstraction and avoid referencing specific real-world elements in the representation of demand centers, facilities, weights, and assignments. This approach was chosen to allow the game to remain independent of the various domains in which the problem instances may arise. The use of metaphors, while effective for datasets that share similar structures with a given domain, may not be applicable to datasets that differ significantly.
   To achieve this, the problem instance is displayed using only basic geometric shapes. The distance between demand centers is depicted as the Euclidean distance between the shapes. The weight parameter is indicated through color, and the assignment is shown with straight lines.
   The FLP is visualized on a two-dimensional white canvas where the demand centers are represented as circular spots with their locations specified by the dataset. The facilities are displayed as black squares inscribed in circles representing the demand centers. The demand centers are all depicted in a blue hue, with the saturation of the blue color varying depending on the weight of the demand center, with lighter blue representing a low demand and darker blue representing a high demand. The assignment between a demand center and a facility is represented by a straight line connecting the two, with the color saturation reflecting the cost of the assignment, which is a function of the distance between the center and the facility, and the demand center's weight; see Figure 3. The borders of unassigned demand centers blink to attract the players' attention, and failed facilities are indicated by blinking, highlighting their former presence as a facility at that location.
   The interface for the game's first and second stages requires the player to make a binary decision for each demand center as to whether to open a facility or not. The player may select a demand center without a facility by tapping it, and deselect a demand center with a facility by tapping it again. This design is intended to resemble the behavior of checkbox controls commonly found in graphical user interfaces.

For stages featuring failed facilities, the player is required to allocate demand centers to facilities manually. To achieve this, the player must initiate a tap-and-drag operation from the facility to the demand center, establishing a connection between the two. During this process, the player's finger or cursor movement is accompanied by a line connecting the facility to the finger or cursor, visually indicating the extension of the facility's coverage to new demand centers.

2. Gameplay: The solution approach to the FLP involves the utilization of a video game consisting of multiple levels that are sequentially presented to the player. These levels are either intended to instruct the player on the mechanics of the game or to provide a real dataset for the player to generate solution candidates for the FLP. The FLP datasets are characterized by a set of demand centers, each described by their spatial location on a two-dimensional plane and the magnitude of their demand, represented by a weight. At certain levels, the player may interact with solution candidates generated by other players, which may include an assignment of facilities to demand centers, a mapping of demand centers to facilities, or a set of failed facilities.
The tutorial levels conclude when the player successfully executes the action that the game is instructing. On the other hand, the conclusion of a dataset level is limited by a time constraint since it is not feasible to determine the optimal solution for an arbitrary dataset without solving the underlying optimization problem.
The implementation of fast-paced gameplay with rapidly changing levels serves several purposes. Firstly, crowd-sourced activities tend to yield better results when kept concise. Secondly, the game's design maximizes the computation performed by each player by exposing them to as many distinct datasets as possible in a limited time frame, leading to a wide range of solutions for a single instance. Finally, the progression of levels in the game provides a sense of reward for the player, preventing frustration that might arise from being stuck on a level with limited room for improvement.
The presentation of the stages in a sequential manner allows for a natural progression of gameplay and the introduction of new mechanics that build upon the concepts and elements of the previous stages.

5. Machine Learning Principles Based on Predictive Methods

In this section, we present an overview of genetic algorithms. The implementation of the algorithms developed by Hernandez et al. in Python is provided, as described in their publication [7]. The authors utilize multi-objective evolutionary algorithms (MOEAs) to address the optimization challenges posed by the facility location problem (FLP). MOEAs are particularly suitable for dealing with optimization problems that exhibit non-continuous, non-convex, and non-linear objectives and constraints. They are also useful in combinatorial optimization problems where optimal solutions may not be guaranteed and the solution space is vast [52]. MOEAs can achieve near-optimal solutions by efficiently exploring only a portion of the solution space. These algorithms are based on the evolutionary process, where the most advantageous traits of a population are identified and utilized to produce the next generation of descendants that are better than their predecessors.

The authors Hernandez et al. [7] utilized two multi-objective evolutionary algorithms (MOEAs)—NSGA-II and MOPSDA—to solve the optimization problems of the FLP. These algorithms are advantageous in combinatorial optimization problems due to their ability to explore the solution space and find near-optimal solutions efficiently. The evolutionary algorithms are based on the principle of genetic inheritance, where the best traits of the population are passed on to the next generation to create offspring with improved characteristics.

In the implementation, a chromosome was utilized to represent the decision variable y, where the ith position y_i of the chromosome represents the opening or closing of the ith distribution center. The chromosome length is equal to the number of distribution centers. Each chromosome was evaluated based on two objectives: the number of facilities and

the total distance. These objectives were used to generate solutions for the Pareto set by allowing the two algorithms to explore different parts of the solution space.

In order to predict the solutions to incomplete algorithmic problems, we will analyze various predictive models, which are foundational to machine learning. These models include cubic spline interpolation, exponential regression, k-nearest neighbors (KNNs), and multiple imputation by chained equations (MICE).

Cubic spline interpolation involves constructing a piecewise polynomial function of the third order that passes through all of the discrete points obtained through the genetic algorithm. The purpose of this method is to estimate a continuous function that fits the given data points as closely as possible. Exponential regression is a statistical method that involves finding an exponential function that best fits the data obtained. This method aims to model the relationship between the dependent and independent variables in the data. KNN is a non-parametric method used for classification and regression. It classifies a new data point based on its k-nearest neighbors in the training data. The prediction for the new data point is based on the majority class or mean of the k-nearest neighbors. MICE is a method used for imputing missing values in a dataset. It involves using regression models to estimate the missing values and then combining the results from multiple imputations to produce a single set of estimates.

Regarding the machine learning methods for estimating missing values in the solution space of the Pareto set after resolving the facility location problem (FLP), we consider an input feature 'x' consisting of 'n' facility allocations and an output target 'y' comprising 'm' distances between the facilities and the demand locations. The output targets contain missing values, which can be imputed using machine learning techniques such as KNN and MICE. These methods leverage the available data to estimate the missing values in the output target 'y', enabling an accurate and comprehensive characterization of the Pareto set's solution space.

Note that the classification of methods such as cubic splines and exponential regression as machine learning methods is a topic of ongoing discussion among researchers. While some argue that they do not meet the criteria for machine learning methods due to their univariate interpolation nature, others contend that they do satisfy the fundamental principles of machine learning. At the core of machine learning is the creation of models that can learn from data to make predictions or decisions. This involves exposing the model to a large dataset, enabling it to identify patterns and relationships that may not be discernible to humans. Once the model has been trained, it can be utilized to predict or classify new instances. Although methods such as cubic splines and exponential regression do not involve the use of sophisticated mathematical models or complex algorithms, they still utilize data to interpolate missing values or data points. They can learn from the data to identify underlying patterns or trends, facilitating accurate approximations of the underlying function. In our study, cubic splines and exponential regression were classified as machine learning techniques due to their capacity to forecast data in the FLP and their potential applicability to analogous problems.

5.1. Cubic Spline Interpolation

Cubic spline interpolation aims to provide accurate approximations with low-degree polynomials when the number of data points is large to avoid oscillations and excessive fluctuations in the interval. This is achieved by dividing the interval into subintervals and constructing a piecewise approximation using several cubic polynomials.

Suppose that we have n data; that is, (x_1, y_1), (x_2, y_2), ..., (x_n, y_n) with $x_1 < x_2 < \cdots < x_n$. The interpolating polynomial is as follows:

$$P(x) = \begin{cases} p_1(x), & x \in [x_1, x_2] \\ p_2(x), & x \in [x_2, x_3] \\ \vdots & \\ p_{n-1}(x), & x \in [x_{n-1}, x_n] \end{cases} \quad (1)$$

where $p_i(x)$ is of the form $a_i + b_i x + c_i x^2 + d_i x^3$ with a_i, b_i, c_i, d_i constants; that is, a polynomial of degree less than or equal to three, corresponding to the interval $[x_i, x_{i+1}]$. In addition, $P(x)$ satisfies the following conditions:

- Continuity; that is,

$$p_i(x_{i+1}) = y_{i+1} = p_{i+1}(x_{i+1}) \quad \forall i = 1, 2, \ldots n-2.$$

- It is twice differentiable (class C^2); that is

$$p_i'(x_{i+1}) = p_{i+1}'(x_{i+1}) \quad \forall i = 1, 2, \ldots n-2.$$

$$p_i''(x_{i+1}) = p_{i+1}''(x_{i+1}) \quad \forall i = 1, 2, \ldots n-2.$$

- $p_1(x_1) = y_1$ y $p_{n-1}(x_n) = y_n$.
- The second derivatives at their boundary are zero; that is, $p_1''(x_1) = 0$ y $p_{n-1}''(x_n) = 0$.

Suppose that, with the genetic algorithm, we obtain n data, with $n < number\ of\ instances$. The idea is to obtain the missing data using an interpolating (dashed) polynomial.

It is worth noting that the spline interpolation does not provide control over the slope of the curves between subintervals. Hence, if the points obtained through the genetic algorithm vary, it can result in a complete change in the interpolation curve. To mitigate this limitation, exponential regression was utilized. This approach leads to a slight improvement in the Pareto front solutions as it results in fewer dominated solutions compared to those obtained through cubic spline interpolation.

5.2. Exponential Regression

Exponential regression is a type of statistical analysis used to model the relationship between an independent variable (x) and a dependent variable (y) when the relationship between the two is non-linear. The exponential function is defined as $y = ae^{(bx)}$, where "a" is the amplitude of the curve and "b" is the exponential growth rate.

Exponential regression involves determining the best-fit parameters, denoted by a and b, for a set of data points through a number of techniques, such as the least squares method or maximum likelihood estimation. The exponential regression method can then be used to predict future values of the dependent variable using new values of the independent variable once the best-fit parameters have been determined.

6. Experimental Results

Our experiments were divided into three distinct parts. In the first part, we presented the results obtained by using genetic algorithms alone. In the second part, we demonstrated the improvement in the solution space achieved through the implementation of a crowdsourcing approach presented as a video game compared to the results obtained through genetic algorithms alone. Finally, in the third part, we demonstrated the efficacy of using predictive models to complement the Pareto solution space obtained through genetic algorithms; see Figure 4 for details.

In order to ensure a statistically significant sample size for our experiments, we conducted 20 iterations of the experiment. The sample size was determined using Cochran's sample size formula [53]:

$$n = \frac{t^2 pq}{d^2}$$

$$n = \frac{(1.282)^2(0.5)(0.5)}{(0.145)^2} = 19.54 \approx 20$$

The statistical sample size was determined using Cochran's sample size formula, with a selected alpha level of 0.05 in each tail, resulting in a value of 1.282. This provides a confidence interval of 80%. The estimate of variance, calculated as $(p)(q)$, was 0.25, assuming an equal probability for a player to provide good or bad results. The acceptable margin of error for the portion being estimated was set to 0.145, which the researcher was willing to accept as a 14.5% error. As a result, we conducted 20 iterations of the experiment to obtain a statistically significant sample.

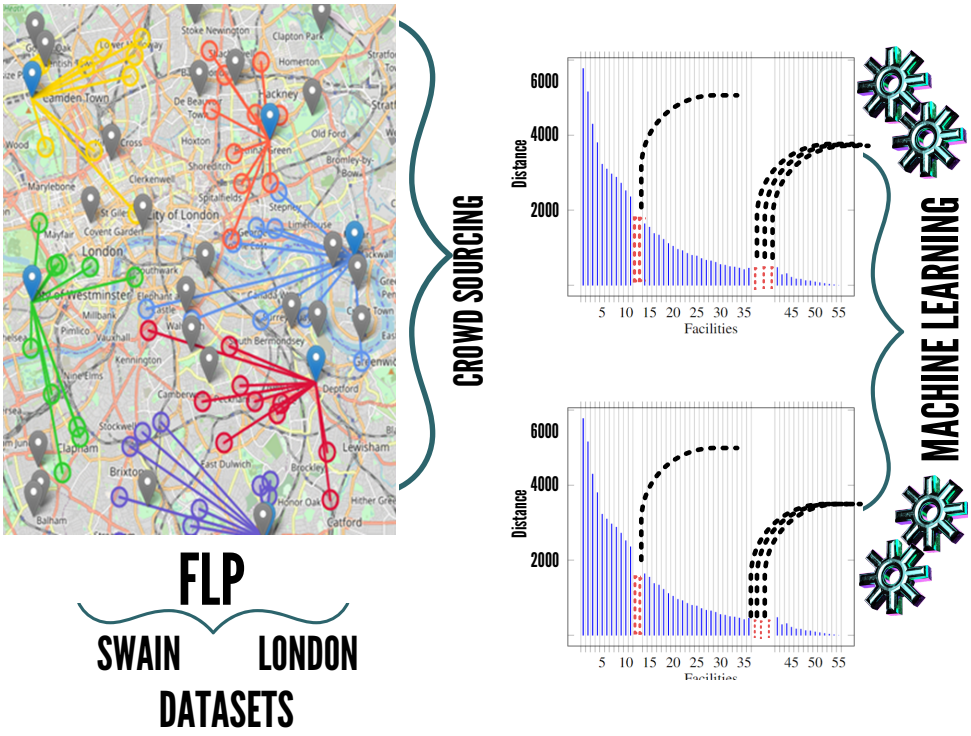

Figure 4. Experimental setup for solving the FLP and for complementing genetic algorithms.

In order to ensure a statistically significant sample, we conducted 20 runs of the genetic algorithms for both the Swain and London datasets. Additionally, we recruited 20 different players to solve each problem once, resulting in a total of 20 executions of the experiment. This approach was taken to ensure that the results were reliable and statistically valid. In the context of the current investigation, a total of 20 game players, consisting of both graduate and undergraduate students, were recruited as participants. The undergraduate cohort was characterized by a diverse range of academic pursuits, spanning multiple disciplines, such as computer science, mathematics, biology, social sciences, and law, among others. The age range of undergraduate participants was 19 to 23. In addition, graduate students from two distinct programs, namely computer science and applied mathematics, were also included in the study. The age range of graduate participants was between 30 to 35 years. The experimental tests were administered in a face-to-face format, during which the participants were instructed to follow the guidelines provided by the video game without any verbal clarification. This approach was chosen to ensure the replication of the

user experience of a game obtained through digital distribution. Subsequent to each play, the video game automatically recorded and stored the resulting data in a private cloud.

On the other hand, both statistical and machine learning models offer several advantages that we leveraged in this work, such as:

- Statistical and machine learning models only need to be executed once, significantly reducing the time required to generate a response that can be analyzed by decision makers.
- Both statistical and machine learning models can be tuned/adjusted to obtain precise approximations.
- It is not necessary to dedicate significant computing time to obtain good approximations.
- Human intervention is minimal, as only an expert is required to execute and obtain the approximations.

These advantages allowed us to efficiently generate accurate approximations with minimal human intervention, making our approach a practical and effective solution for decision-making processes. It is noteworthy that the human subjects in the study could play the video game without the need for prior training or knowledge of the underlying problems, and they did not experience any stress during the experiment. The participants were requested to engage in gameplay during their allocated break-time for school activities.

6.1. Solving the FLP Using Genetic Algorithms

The genetic algorithms used for solving the problems presented in Section 2 are parameterized as follows: a random seed is initially generated, and half of the problem instances are assigned to the MOPSDA algorithm, while the other half is assigned to the NSGA-II algorithm. The population size is set to 100, and the number of generations is fixed at 100. The number of runs is also set to 5. These multi-objective evolutionary algorithms (MOEAs) are used because they can handle non-continuous, non-convex, and/or non-linear objectives/constraints, as well as problems where the objective function is not explicitly known. The NSGA-II and MOPSDA algorithms are utilized to obtain near-optimal solutions by efficiently exploring a fraction of the entire solution space. The MOEAs are based on the process of evolution, where the best traits of a population are identified and used to generate the next generation or replace the population.

To solve the first subproblem of Section 2, the optimal Pareto set is approximated using both NSGA-II and MOPSDA, and the resulting Pareto sets are merged into one. The decision variable y is represented using a chromosome, where position i has the bit y_i. The chromosome length is equal to the number of distribution centers ($|F|$), and element y_i is equal to 1 if the facility is open, and 0 otherwise. Each chromosome is evaluated using two figures of merit: one to obtain the number of facilities and the other to obtain the total distance (Figures 5 and 6).

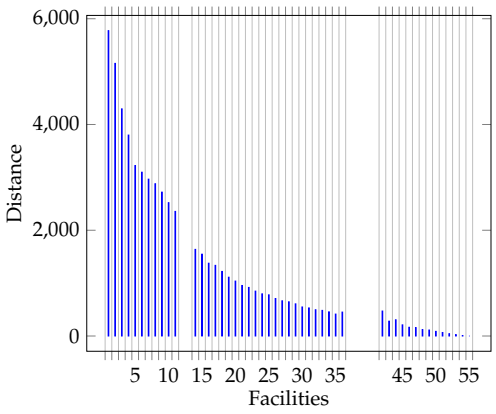

Figure 5. Computed Pareto set for the Swain dataset using genetic algorithms.

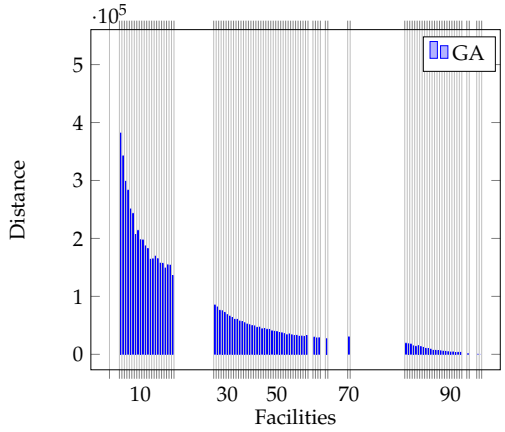

Figure 6. Computed Pareto set for the London dataset using genetic algorithms for a large instance problem.

We chose the optimal solutions obtained from 20 independent runs of the genetic algorithms illustrated in Figure 5. This figure depicts the Pareto set obtained for the Swain dataset and presents a reasonable representation of the Pareto set. However, the genetic algorithms fail to find solutions for all possible instances. This results in three potential approaches to addressing the missing solutions. The first option is to re-run the genetic algorithms (as we carried out 20 times). Nevertheless, this is computationally expensive and consumes substantial resources (approximately 30 min for the Swain dataset and 3.5 h for the London dataset). The second option involves utilizing a crowdsourcing approach, as proposed, to fill the gaps left by the genetic algorithms. Finally, as demonstrated in the following sections, the third alternative uses machine learning for predictive purposes as a more efficient and timely solution to fill the gaps.

We chose the optimal solutions obtained from 20 independent runs of the genetic algorithms illustrated in Figure 6. This figure depicts the Pareto set obtained for the London dataset and highlights the incompleteness of the solutions found by the Pareto set. Furthermore, NP-hard problems can become computationally intractable for larger instances of the FLP or other complex problems. As a result, it is imperative to have alternative approaches, such as the one proposed in this study, to estimate the solutions that would be obtained after multiple iterations of the genetic algorithms. This is particularly relevant for larger instances of NP-hard problems, as they require increased computational resources.

The Pareto set is defined as the set of solutions that cannot be improved in at least one dimension without worsening in another dimension. However, in some cases, the relationships between different solution dimensions may not be monotonic; that is, there may be solutions that are not strictly dominated but do not belong to the Pareto set. In these non-monotonic situations, solutions may not have a clear order among themselves, making the identification of the Pareto set more challenging. Additionally, a simple post-processing of the output of the genetic algorithm may not be sufficient to resolve these non-monotonic situations.

One way to address non-monotonic situations is through the use of genetic algorithms that employ more advanced techniques, such as neighborhood search, parameter adaptation, and diversified selection. These methods can help to identify non-monotonic solutions and resolve the problems that arise in these situations, yet this is beyond the scope of this paper. Here, we focused on the process of finding efficient solutions that complement those found by genetic algorithms.

6.2. Crowdsourcing the FLP in Video Game Plays

The results obtained through the implementation of the genetic algorithms reveal the existence of gaps in the solution space. This is due to the lack of agreement between the two utilized genetic algorithms (MOPSDA and NGSA) in determining the optimal solution, as they both tend to converge toward local minima. The incompleteness of the solutions for

various Pareto front instances is therefore considered as a drawback. To mitigate this issue, a re-execution of the genetic algorithm could be performed with the hope of obtaining solutions for the missing data, although there is no guarantee that the algorithm would be able to fill in the gaps in subsequent runs.

To address the aforementioned problem, the FLP was modeled as a video game and crowdsourced to a number of individuals. The contributions obtained from the crowd were found to be adequate in filling in the gaps left by the genetic algorithms. The results of this approach are presented in the following figures.

The FLP is a type of optimization problem that involves multiple objectives. The solution to this problem involves finding a set of solutions, called the Pareto set, which cannot be improved in one objective without negatively impacting another objective. To evaluate the quality of the solution, the aggregate data from players is measured based on how closely it approximates the Pareto set as determined by alternative methods. The optimal responses used in our analysis were selected from a dataset containing results from 20 different players, as illustrated in the accompanying Figure 7. The selection process involved choosing the most accurate and reliable responses, ensuring that our analysis was based on high-quality data. As can be observed from the figures, the results obtained through crowdsourcing—that is, through the engagement of individuals playing a video game modeling the FLP problem—are comparable to those obtained through sophisticated optimization tools such as genetic algorithms. Figure 7 demonstrate that gamers follow the trend in the Pareto front. When they play optimally, they effectively enhance the results obtained through genetic algorithms. Furthermore, the intuition and perspective of gamers enable them to make trade-offs between short-term benefits and long-term goals, thus contributing to better solutions.

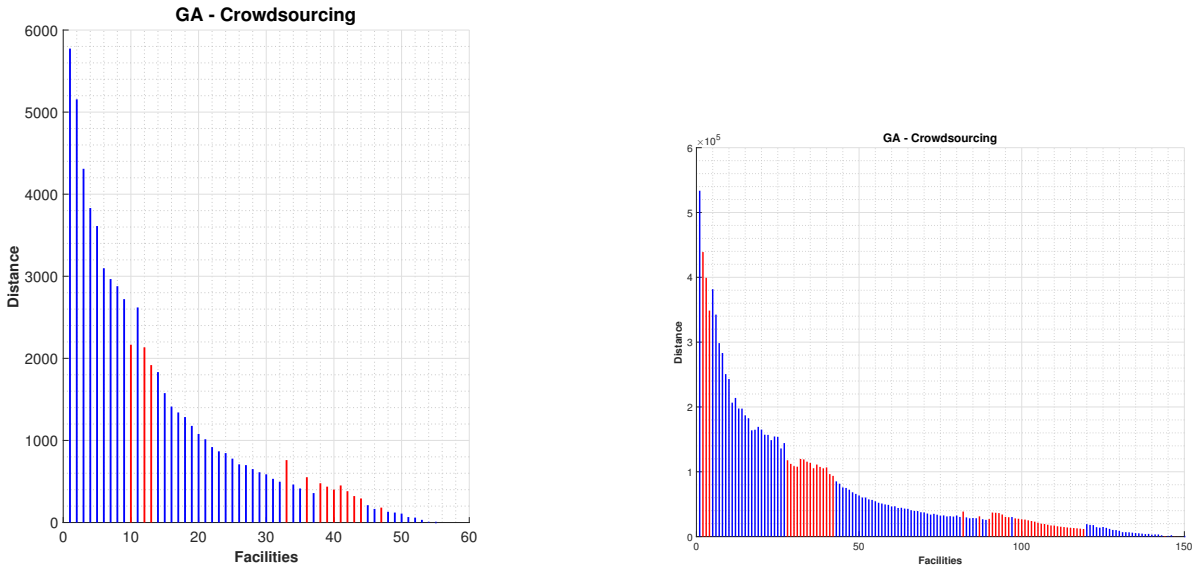

Figure 7. The Pareto set after solving the FLP complemented via crowdsourcing for: the Swain dataset (**left**), and the London dataset (**right**).

The biggest advantage of crowdsourcing can be observed in problems with a high number of facilities, as is the case for the London dataset. In this instance, the genetic algorithms generated significant gaps in the solution space, whereas crowdsourced solutions matched and even improved upon the solutions generated by genetic algorithms. Re-executing the genetic algorithms to fill these gaps would consume a substantial amount of computational resources, in addition to the time required for the genetic algorithms to generate the complete Pareto front. The re-run of the genetic algorithm for the London dataset is estimated to take approximately two to three hours, without any guarantee of filling the previously encountered gaps.

However, a limitation of the crowdsourcing approach is that it is challenging to objectively quantify the duration required to construct and design video games. While an adept individual possessing programming skills and familiarity with multi-objective optimization problems (MOPs) may complete the coding process within a few weeks, an unskilled individual could require several months. Despite this, after the game has been constructed and developed, it can be conveniently adapted for other relevant datasets and utilized as a simulation model capable of swiftly responding to emergency scenarios.

The interaction between humans and computers in carrying out crowdsourcing activities can complement the Pareto search space, as the solutions obtained by the crowdsourcing approach can be superior to those generated by genetic algorithms in many cases, as shown in Figure 7. To further enhance the quality of solutions, a hybrid combination of crowdsourcing and genetic algorithms could be explored, leveraging the strengths of both approaches. Our empirical study indicates that crowdsourcing can offer a sufficient number of solutions to fill the gaps presented by genetic algorithms. Additionally, the use of machine learning techniques could provide insights into the heuristics employed by users to solve computationally challenging problems. As a proposed future direction, we suggest that combining the best solutions from crowdsourcing with those generated by genetic algorithms could reduce the convergence time or improve the quality of the solutions. However, we must acknowledge that reducing the convergence time of genetic algorithms lies beyond the scope of this work.

The crowdsourcing-based approach has a drawback in that not all user responses carry significant value. Moreover, some players may not comprehend the video game instantly, especially those without engineering or mathematical backgrounds, resulting in suboptimal solutions for multi-objective optimization problems (MOPs). The research was conducted in a university setting with a diverse audience, thereby leading to a lack of expertise in MOPs. However, despite these limitations, we observed a statistically significant enhancement in the solutions, even though not all players contributed with high-quality answers.

6.3. Solving the FLP Using Predictive Models

The third alternative is to use predictive methods to estimate the solutions for incomplete data. Given a dataset with gaps or missing information, this approach aims to approximate the solutions that would have been obtained after multiple iterations of the algorithms.

One of the critical considerations for researchers, corporations, and decision makers is the time required to solve complex problems. For instance, in the case of an emergency situation (such as a natural disaster), decision makers must respond quickly and make decisions on the placement of shelters (facilities) while taking into account the proximity of the affected population to ensure their safety.

The use of machine learning techniques has demonstrated the capability of generating approximate or comparable results to those produced by sophisticated methods such as genetic algorithms, as evident from Figures 8–11. Implementing predictive machine learning models can provide a significant advantage for decision makers in emergency situations, as they enable an almost instantaneous reaction. In the context of the present study, this means that decision makers can quickly allocate refugee facilities to attend to people affected by emergencies within a city or state. The only prerequisites for such rapid responses are the availability of relevant data and the execution of fitting curves or machine learning models. The decision makers can then visualize the Pareto front produced by the various predictive methods, as depicted in Figures 8–11, and determine the most suitable solution based on the circumstances.

Figures 8 and 9 present the results of the complete Pareto front achieved through the utilization of interpolation techniques such as cubic splines and exponential regression, respectively. It can be observed that the utilization of exponential regression leads to a more accurate approximation of the Pareto front compared to cubic splines. This is due to

the fundamental nature of the regression method, which aims to fit the data trend in the most optimal manner. Conversely, the cubic splines technique necessitates the solutions to pass through the given points, which results in more oscillatory solutions and a higher number of dominated solutions.

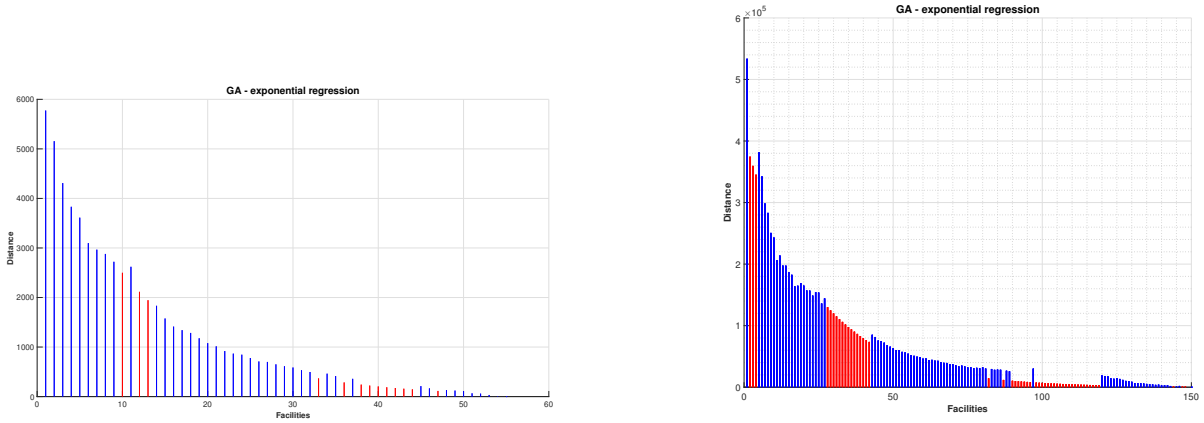

Figure 8. The Pareto set after solving the FLP complemented via exponential regression for: the Swain dataset (**left**), and the London dataset (**right**).

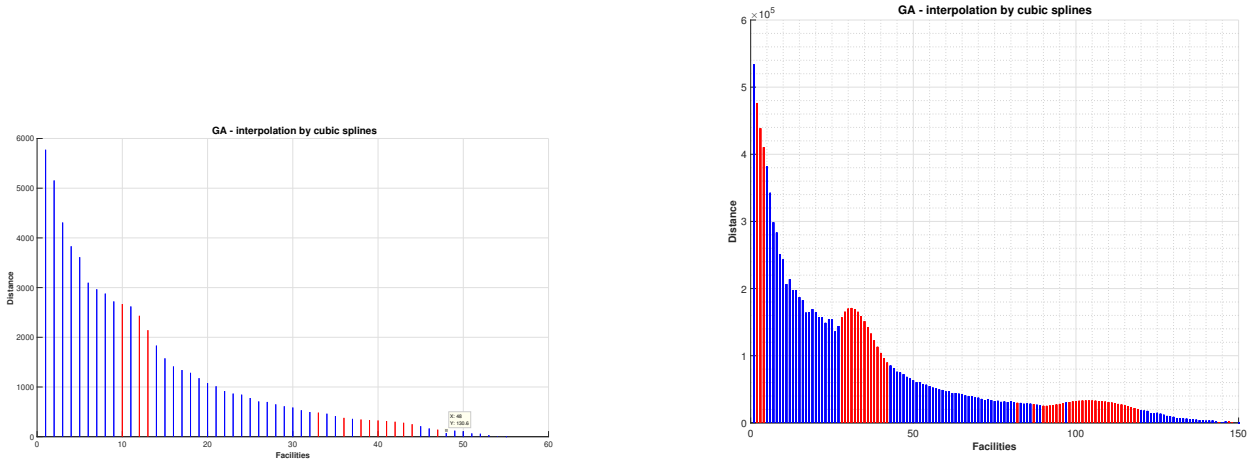

Figure 9. The Pareto set after solving the FLP complemented via cubic spline for: the Swain dataset (**left**), and the London dataset (**right**).

Figures 10 and 11 present the results obtained through the utilization of k-nearest neighbors (KNNs) and multiple imputation by chained equations (MICE), respectively. KNN is a supervised machine learning algorithm primarily used for solving classification and regression problems, whereas MICE is a statistical method for dealing with missing data. The results obtained through both methods may be similar as they employ similar strategies, such as using the values of similar observations or utilizing a model to predict missing values based on the available data.

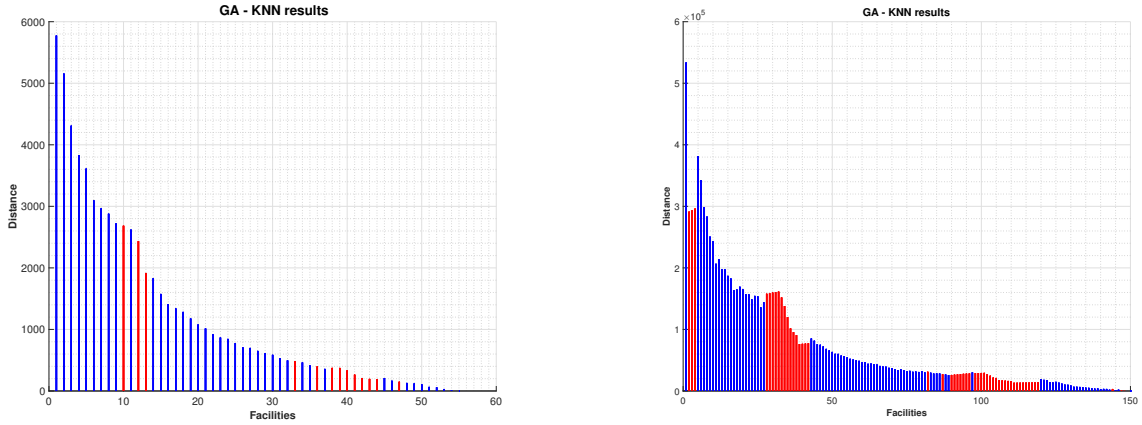

Figure 10. The Pareto set after solving the FLP complemented via KNN for: the Swain dataset (**left**), and the London dataset (**right**).

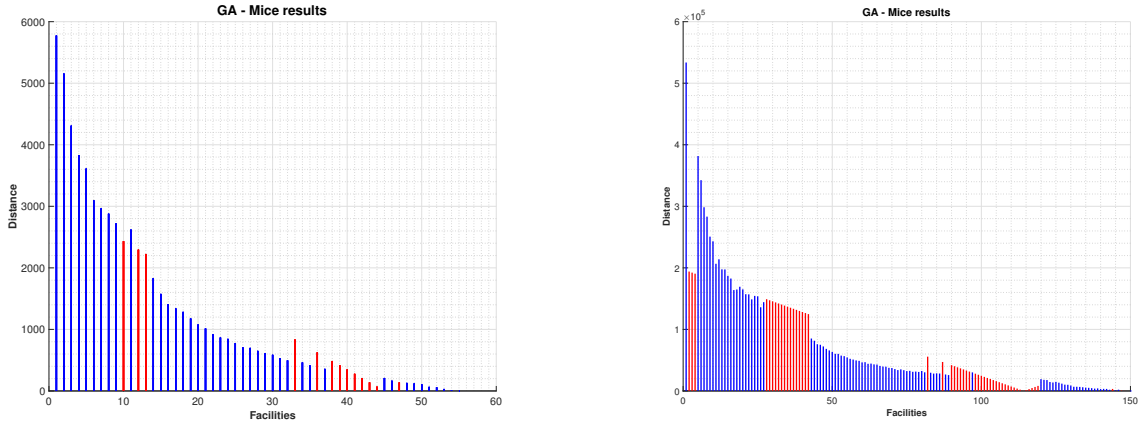

Figure 11. The Pareto set after solving the FLP complemented via MICE for: the Swain dataset (**left**), and the London dataset (**right**).

As depicted in Figure 11, the multiple imputation by chained equations (MICE) method utilizes a statistical model to generate several imputed datasets, which are then combined to produce a continuous curve. On the other hand, as shown in Figure 10, the k-nearest neighbors (KNNs) algorithm imputes missing data based on the information of its nearest neighbors, which results in a steeper or "noisier" curve due to the non-parametric nature of the algorithm.

7. Conclusions

In this paper, we used some known machine learning techniques that can be used as predictive models to make predictions about future events or outcomes based on historical data. One of the main advantages that we found after using machine learning for prediction was that we could handle large and complex datasets efficiently. In addition, our findings show that, as new data become available, we can easily provide decision makers with useful information. However, it is worth noting that the accuracy of predictions made using machine learning models can vary depending on the quality and relevance of the training data (in our case, truncated datasets), as well as the specific algorithm used.

We acknowledge that there exists a broader range of machine learning techniques that could be utilized in the present study, such as random forest, support vector machines, and decision trees. However, we must note that the successful implementation of these techniques requires a set of features and output values. In contrast, the current study only contains one input feature and one output value, with missing data in some cases.

Therefore, to ensure the appropriate selection of machine learning techniques, a thorough investigation was undertaken to identify approaches that were suitable for the specific case at hand. Furthermore, it should be emphasized that the utilization of techniques that require multiple features and outputs may lead to model overfitting, as well as an increased computational complexity, resulting in a poor model performance. Thus, the use of appropriate techniques is crucial for the development of accurate and reliable predictive models.

We plan to analyze and extract the players' strategies in future work. We believe that doing so can allow for more efficient solutions. Our purpose is to have enough knowledge of people gathered to implement hybrid approaches, i.e., a combination of human–computer interactions, where some parts of an NP-hard problem would be solved using algorithms and other parts of the problem using human knowledge.

Author Contributions: Conceptualization, M.V.-S., D.A.L.-V. and R.M.J.; formal analysis, M.V.-S., D.A.L.-V. and L.A.M.-R.; investigation, M.V.-S., R.M.J. and L.A.M.-R.; methodology, M.V.-S., D.A.L.-V. and R.M.J.; software, M.V.-S., D.A.L.-V. and L.A.M.-R.; supervision, D.A.L.-V. and R.M.J.; validation, D.A.L.-V. and M.V.-S.; writing—original draft, M.V.-S.; writing—review and editing, D.A.L.-V., R.M.J. and L.A.M.-R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by the project "Optimization via crowd computing for scientific computing problems analyzing through machine learning" of the National Council of Science and Technology (CONACYT) for Mariano Vargas Santiago (CVU number 417439). Besides, it was partially supported by the Conacyt project 613 "Cyber-Physical Systems for the Development of Intelligent Transport Systems".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing is not applicable to this article.

Acknowledgments: The authors express their sincere gratitude to the players for their invaluable participation and collaboration in this study. Additionally, the authors extend their appreciation to the undergraduate and graduate students of Universidad Autónoma Metropolitana, Unidad Cuajimalpa, for generously devoting their time and efforts towards the accomplishment of this research endeavor. The authors acknowledge that their contributions have significantly enhanced the quality and validity of this study.

Conflicts of Interest: The authors declare no conflict of interest. The founders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

FLP	Facility Location Problem
GA	Genetic Algorithm
KNN	K-Nearest Neighbors
MICE	Multiple Imputation by Chained Equations
MOEAs	Multi-Objective Evolutionary Algorithms
MOPSDA	Multi-Objective Probabilistic Solution Discovery Algorithm
NGSA	Non-dominated Sorting Genetic Algorithm
RFLP	Robust Facility Location Problem

References

1. Takagi, H. Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation. *Proc. IEEE* **2001**, *89*, 1275–1296. [CrossRef]
2. Anderson, D.; Anderson, E.; Lesh, N.; Marks, J.; Mirtich, B.; Ratajczak, D.; Ryall, K. Human-guided simple search. In Proceedings of the AAAI/IAAI, Austin, TX, USA, 30 July–3 August 2000; pp. 209–216.
3. Megiddo, N.; Tamir, A. On the complexity of locating linear facilities in the plane. *Oper. Res. Lett.* **1982**, *1*, 194–197. [CrossRef]

4. Lenart-Gansiniec, R.; Czakon, W.; Sułkowski, Ł.; Pocek, J. Understanding crowdsourcing in science. *Rev. Manag. Sci.* **2022**, 1–34. [CrossRef]
5. Graham, R.L. Bounds on multiprocessing timing anomalies. *SIAM J. Appl. Math.* **1969**, *17*, 416–429. [CrossRef]
6. Pinedo, M.; Schrage, L. Stochastic shop scheduling: A survey. In *Deterministic and Stochastic Scheduling*; Springer: Berlin/Heidelberg, Germany, 1982; pp. 181–196.
7. Hernandez, I.; Ramirez-Marquez, J.E.; Rainwater, C.; Pohl, E.; Medal, H. Robust facility location: Hedging against failures. *Reliab. Eng. Syst. Saf.* **2014**, *123*, 73–80. [CrossRef]
8. Chen, C.H.; Chou, J.H. Multiobjective optimization of airline crew roster recovery problems under disruption conditions. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *47*, 133–144. [CrossRef]
9. Pliego-Marugán, A.; Pinar-Pérez, J.M.; Ruiz-Hernández, D. A Metaheuristic Approach for Quantifying the Effects of the Structural Complexity in Facility Location Problems. In Proceedings of the International Conference on Management Science and Engineering Management, Chisinau, Moldova, 30 July–2 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 45–56.
10. Das, S.K.; Roy, S.K.; Weber, G.W. An exact and a heuristic approach for the transportation-p-facility location problem. *Comput. Manag. Sci.* **2020**, *17*, 389–407. [CrossRef]
11. Ahmadi-Javid, A.; Seyedi, P.; Syam, S.S. A survey of healthcare facility location. *Comput. Oper. Res.* **2017**, *79*, 223–263. [CrossRef]
12. Vlasenko, I.; Nikolaidis, I.; Stroulia, E. The smart-condo: Optimizing sensor placement for indoor localization. *IEEE Trans. Syst. Man Cybern. Syst.* **2015**, *45*, 436–453. [CrossRef]
13. Frank, C.; Römer, K. Distributed facility location algorithms for flexible configuration of wireless sensor networks. In Proceedings of the International Conference on Distributed Computing in Sensor Systems, Santa Fe, NM, USA, 18–20 June 2007; Springer: Berlin/Heidelberg, Germany, 2007; pp. 124–141.
14. Swain, R.W. *A Decomposition Algorithm for a Class of Facility Location Problems*; Technical Report; Cornell University: Ithaca, NY, USA, 1971.
15. Church, R.L.; Baez, C.A. Generating optimal and near-optimal solutions to facility location problems. *Environ. Plan. B Urban Anal. City Sci.* **2020**, *47*, 1014–1030. [CrossRef]
16. Scott, S.D.; Lesh, N.; Klau, G.W. Investigating human-computer optimization. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Minneapolis, MN, USA, 20–25 April 2002; pp. 155–162.
17. Fan, X.; McNeese, M.; Sun, B.; Hanratty, T.; Allender, L.; Yen, J. Human–agent collaboration for time-stressed multicontext decision making. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2010**, *40*, 306–320. [CrossRef]
18. Cho, S.B.; Lee, J.Y. A human-oriented image retrieval system using interactive genetic algorithm. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2002**, *32*, 452–458.
19. Simons, C.L.; Parmee, I.C. Elegant object-oriented software design via interactive, evolutionary computation. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2012**, *42*, 1797–1805. [CrossRef]
20. Khatib, F.; Cooper, S.; Tyka, M.D.; Xu, K.; Makedon, I.; Popović, Z.; Baker, D. Algorithm discovery by protein folding game players. *Proc. Natl. Acad. Sci. USA* **2011**, *108*, 18949–18953. [CrossRef]
21. Kawrykow, A.; Roumanis, G.; Kam, A.; Kwak, D.; Leung, C.; Wu, C.; Zarour, E.; Sarmenta, L.; Blanchette, M.; Waldispühl, J.; et al. Phylo: A citizen science approach for improving multiple sequence alignment. *PLoS ONE* **2012**, *7*, e31362. [CrossRef]
22. Muhammadi, J.; Rabiee, H.R. Crowd computing: A survey. *arXiv* **2013**, arXiv:1301.2774.
23. Wang, W.; Jiang, J.; An, B.; Jiang, Y.; Chen, B. Toward efficient team formation for crowdsourcing in noncooperative social networks. *IEEE Trans. Cybern.* **2017**, *47*, 4208–4222. [CrossRef]
24. Wang, L.; Zhang, D.; Yan, Z.; Xiong, H.; Xie, B. effSense: A novel mobile crowd-sensing framework for energy-efficient and cost-effective data uploading. *IEEE Trans. Syst. Man Cybern. Syst.* **2015**, *45*, 1549–1563. [CrossRef]
25. Gorriz, C.M.; Medina, C. Engaging girls with computers through software games. *Commun. ACM* **2000**, *43*, 42. [CrossRef]
26. Schneider, D.; de Souza, J.; Lucas, E.M. Towards a typology of social news apps from a Crowd Computing perspective. In Proceedings of the 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC), San Diego, CA, USA, 5–8 October 2014; pp. 1134–1140.
27. Agreste, S.; De Meo, P.; Ferrara, E.; Piccolo, S.; Provetti, A. Analysis of a heterogeneous social network of humans and cultural objects. *IEEE Trans. Syst. Man Cybern. Syst.* **2015**, *45*, 559–570. [CrossRef]
28. Gomes, C.; Schneider, D.; Moraes, K.; De Souza, J. Crowdsourcing for music: Survey and taxonomy. In Proceedings of the 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Seoul, Republic of Korea, 14–17 October 2012; pp. 832–839.
29. Bongard, J.C.; Hines, P.D.; Conger, D.; Hurd, P.; Lu, Z. Crowdsourcing predictors of behavioral outcomes. *IEEE Trans. Syst. Man Cybern. Syst.* **2013**, *43*, 176–185. [CrossRef]
30. Antelio, M.; Esteves, M.G.P.; Schneider, D.; de Souza, J.M. Qualitocracy: A data quality collaborative framework applied to citizen science. In Proceedings of the 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Seoul, Republic of Korea, 14–17 October 2012; pp. 931–936.
31. Oh, H.K.; Kim, S.W.; Park, S.; Zhou, M. Can you trust online ratings? A mutual reinforcement model for trustworthy online rating systems. *IEEE Trans. Syst. Man Cybern. Syst.* **2015**, *45*, 1564–1576. [CrossRef]
32. Gören, S.; Baccouche, A.; Pierreval, H. A framework to incorporate decision-maker preferences into simulation optimization to support collaborative design. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *47*, 229–237. [CrossRef]

33. Zhang, J.; Yin, Z.; Wang, R. Recognition of mental workload levels under complex human–machine collaboration by using physiological features and adaptive support vector machines. *IEEE Trans. Hum. Mach. Syst.* **2015**, *45*, 200–214. [CrossRef]
34. Allahbakhsh, M.; Arbabi, S.; Galavii, M.; Daniel, F.; Benatallah, B. Crowdsourcing planar facility location allocation problems. *Computing* **2019**, *101*, 237–261. [CrossRef]
35. Jiang, J.; An, B.; Jiang, Y.; Zhang, C.; Bu, Z.; Cao, J. Group-Oriented Task Allocation for Crowdsourcing in Social Networks. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 4417–4432. [CrossRef]
36. Xu, M.; Wang, S.; Hu, Q.; Sheng, H.; Cheng, X. Quantum analysis on task allocation and quality control for crowdsourcing with homogeneous workers. *IEEE Trans. Netw. Sci. Eng.* **2020**, *7*, 2830–2839. [CrossRef]
37. Von Ahn, L. Games with a purpose. *Computer* **2006**, *39*, 92–94. [CrossRef]
38. Mavandadi, S.; Feng, S.; Yu, F.; Dimitrov, S.; Yu, R.; Ozcan, A. BioGames: A platform for crowd-sourced biomedical image analysis and telediagnosis. *Games Health Res. Dev. Clin. Appl.* **2012**, *1*, 373–376. [CrossRef]
39. Wang, L.; Jiang, T. On the complexity of multiple sequence alignment. *J. Comput. Biol.* **1994**, *1*, 337–348. [CrossRef]
40. Cooper, S.; Khatib, F.; Treuille, A.; Barbero, J.; Lee, J.; Beenen, M.; Leaver-Fay, A.; Baker, D.; Popović, Z.; Foldit players. Predicting protein structures with a multiplayer online game. *Nature* **2010**, *466*, 756. [CrossRef]
41. Kim, J.S.; Greene, M.J.; Zlateski, A.; Lee, K.; Richardson, M.; Turaga, S.C.; Purcaro, M.; Balkam, M.; Robinson, A.; Behabadi, B.F.; et al. Space–time wiring specificity supports direction selectivity in the retina. *Nature* **2014**, *509*, 331. [CrossRef]
42. Marx, V. Neuroscience waves to the crowd. *Nat. Methods* **2013**, *10*, 1069–1074. [CrossRef]
43. Von Ahn, L. Human computation. In Proceedings of the 2008 IEEE 24th International Conference on Data Engineering, Vancouver, BC, Canada, 7–12 May 2008; pp. 1–2.
44. Law, E.L.; Von Ahn, L.; Dannenberg, R.B.; Crawford, M. TagATune: A Game for Music and Sound Annotation. In Proceedings of the 8th International Conference on Music Information Retrieval, Vienna, Austria, 23–27 September 2007; Volume 3, p. 2.
45. Law, E.; West, K.; Mandel, M.I.; Bay, M.; Downie, J.S. Evaluation of Algorithms Using Games: The Case of Music Tagging. In *ISMIR*; Austrian Computer Society: Wien, Austria, 2009; pp. 387–392.
46. Lafourcade, M. Making people play for Lexical Acquisition with the JeuxDeMots prototype. In Proceedings of the SNLP'07: 7th international Symposium on Natural Language Processing, Chonburi, Thailand, 5–7 October 2007; p. 7.
47. Krause, M.; Takhtamysheva, A.; Wittstock, M.; Malaka, R. Frontiers of a paradigm: Exploring human computation with digital games. In Proceedings of the ACM sigkdd Workshop on Human Computation, Washington, DC, USA, 25 July 2010; pp. 22–25.
48. Speer, R.; Havasi, C.; Surana, H. Using verbosity: Common sense data from games with a purpose. In Proceedings of the Twenty-Third International FLAIRS Conference, Daytona Beach, FL, USA, 19–21 May 2010.
49. Chamberlain, J.; Poesio, M.; Kruschwitz, U. Phrase detectives: A web-based collaborative annotation game. In Proceedings of the International Conference on Semantic Systems (I-Semantics' 08), Graz, Austria, 3–5 September 2008; pp. 42–49.
50. Fort, K.; Guillaume, B.; Chastant, H. Creating Zombilingo, a Game with A Purpose for dependency syntax annotation. In Proceedings of the Gamification for Information Retrieval (GamifIR'14) Workshop, Amsterdam, The Netherlands, 13 April 2014.
51. Ivanisevic, I.; Lumelsky, V.J. Configuration space as a means for augmenting human performance in teleoperation tasks. *IEEE Trans. Syst. Man Cybern. Part B* **2000**, *30*, 471–484. [CrossRef]
52. Smith, J.C.; Taskin, Z.C. A tutorial guide to mixed-integer programming models and solution techniques. In *Optimization in Medicine and Biology*; Taylor & Francis: Oxfordshire, UK, 2008 ; pp. 521–548.
53. Kotrlik, J.; Higgins, C. Organizational research: Determining appropriate sample size in survey research appropriate sample size in survey research. *Inf. Technol. Learn. Perform. J.* **2001**, *19*, 43.