Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 108C (2017) 205–214

Procedia Computer Science

International Conference on Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland

# A Serious Video Game To Support Decision Making On Refugee Aid Deployment Policy

Luis Eduardo Perez Estrada[1], Derek Groen[23], and Jose Emmanuel Ramirez-Marquez[1]

[1] School of Systems and Enterprises, Stevens Institute of Technology, Hoboken, NJ, USA
{lperezes, jmarquez}@stevens.edu
[2] Department of Computer Science, Brunel University London, Uxbridge, Middlesex, UK
Derek.Groen@brunel.ac.uk
[3] Centre for Computational Science, University College London, London, UK

**Abstract**

The success of refugee support operations depends on the ability of humanitarian organizations and governments to deploy aid effectively. These operations require that decisions on resource allocation are made as quickly as possible in order to respond to urgent crises and, by anticipating future developments, remain adequate as the situation evolves. Agent-based modeling and simulation has been used to understand the progression of past refugee crises, as well as a way to predict how new ones will unfold. In this work, we tackle the problem of refugee aid deployment as a variant of the Robust Facility Location Problem (RFLP). We present a serious video game that functions as an interface for an agent-based simulation run with data from past refugee crises. Having obtained good approximate solutions to the RFLP by implementing a game that frames the problem as a puzzle, we adapted its mechanics and interface to correspond to refugee situations. The game is intended to be played by both subject matter experts and the general public, as a way to crowd-source effective courses of action in these situations.


Peer-review under responsibility of the scientific committee of the International Conference on Computational Science

*Keywords:* Refugees, simulation, modeling, serious games

# 1 Introduction

## 1.1 Computer modeling for refugee crisis management

As of 2016, there are 65.3 million people in the world who are forcibly displaced from their homes, 21.3 million of them refugees. This is the largest amount of displaced people that has ever been recorded [1]. Assisting and relocating these refugees places tremendous stress on humanitarian organizations, as well as political and economic systems. The ability of support organizations to carry out their functions depends on their ability to deploy aid in the correct places at the appropriate time, which involves not only logistical prowess to attend present needs, but a forecast of the situation to allow for proactive action.

1877-0509 © 2017 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the scientific committee of the International Conference on Computational Science
10.1016/j.procs.2017.05.112

To predict how a crisis will unfold — or indeed, better understand the full complexity of past crises — many attempts have been made to model the movement of people in such emergencies [2] [3]. In the literature, agent-based modeling (ABM) and simulation appears to be the favored method to tackle these problems, from addressing situations in a single city [4], migration due to climate events [5], country-wide conflict [6], situations inside an individual refugee camp [7], or migration in general [8]. *Flee* [3], a network-based ABM specifically developed for the modeling of refugee crises, is used extensively in this work.

Good simulation models are invaluable as support for decision makers, particularly if the insights generated, though complex, are easily comprehensible by subject matter experts without requiring deep knowledge into the inner workings of the simulation itself. We believe experts in refugee crisis management would greatly benefit from a simulation system and interface that could help in directly exploring alternative policies and actions, as well as their ensuing consequences.

## 1.2 Games for solving problems

Interactive simulations and video games can be themselves a tool to tackle computationally hard problems, using human-based computation techniques. These methods consist of having a computer offload tasks to a human whenever an intractable problem is encountered. As such, the computer and human participant cooperate in finding a solution to a problem, having leveraged the strengths of each other. This approach has been used in applications such as speech processing, music generation and recognition, knowledge acquisition, information retrieval, and robotics [9].

When these problems are framed as games, some are able to attract a large enough player base that they can crowd-source solutions. Games like BioGames [10], Phylo [11], EyeWire [12] and FoldIt [13] have enabled thousands of players to participate in the solution of problems in biological and medical science.

This work entails the adaptation of a video game designed to tackle the Robust Facility Location Problem (RFLP) into an interactive simulation intended to assist policy and decision makers with examining the costs and humanitarian impacts of potential actions, as well as being distributable to a large audience in order to obtain crowd-sourced solutions to crisis scenarios.

## 1.3 Robust facility location problem

The facility location problem (FLP) is a combinatorial optimization problem that involves choosing where a set of *facilities* are to be placed in order to minimize the costs of servicing a set of *demand centers*. Each demand center has its own magnitude of *demand*. A solution candidate is evaluated by its *total cost*: the sum of the operating cost of having those facilities open, and the cost of transport between a demand center and its closest facility. If the facilities are modeled as having unlimited capacity, an instance of the problem is of the *uncapacitated* variant. If limits in capacity are considered, then it is *capacitated* [14].

The basic formulation of the FLP considers static scenarios, where the parameters of the problem such as the location of the demand centers and their magnitude, remain constant. If these conditions change with time, the problem becomes *dynamic*. Solutions to this kind of problem can have one of two objectives: finding a solution that will stay in place and must remain cost-effective for a length of time despite changing circumstances; or optimizing costs while allowing for the relocation of facilities, considering the costs of doing so [15] [16].

The FLP has been proven to be NP-complete [17]. In order to obtain sufficiently good solutions in reasonable time frames, it is usually tackled with heuristics, approximations [18], or meta-heuristics like genetic algorithms [19].

The *robust* variant of the problem (RFLP [19]) considers the possibility of failure of facilities, and the re-assignment of demand centers to facilities to resume normal operations. The RFLP is a multi-objective optimization problem, where a trade-off is considered between the cost of a solution before any failures occur, and the cost of the same solution after a number of facilities have failed. For example, while a redundant facility will increase the total cost of a solution when all facilities function as normal, that redundancy might save costs after a failure, since the demand will continue to be addressed efficiently.

### 1.4 Refugee crisis and refugee movements

A common trait of forced migrants is an immediate need for safety and survival. When a large crisis emerges, these migrants therefore will go on the move, searching for places where these basic needs can be fulfilled. The exact movement patterns of forced migrants depend on a very wide range of conditions, but in the case of refugees approximate predictions of their preferred destinations can be made using simulations [3]. Destinations of forced migrants can differ, as some people will end up staying within their home country (Internally Displaced People), while others may have the means to find safety and survival on their own, without external help. Within this work we look specifically at forced migrants that *lack* the means to survive on their own, and that require additional resources and support (e.g., in refugee camps).

### 1.5 Formulating refugee crisis management as a dynamic RFLP instance

The problem of modeling and analyzing refugee crisis situations shares many similarities with the RFLP, from the perspective of giving the refugees the attention they require as efficiently as possible. Simpler formulations of the FLP have been applied in the context of natural disasters [20], and as a part of the larger field of humanitarian logistics [21].

In this context, the needs of refugees (shelter, food, water, security, etc.) is considered as demand that needs to be satisfied. Refugees are located at geographical locations — cities or regions — that correspond to the demand centers that need to be attended. The more people found at a given location, the higher the magnitude of the demand there.

For this purpose, the facilities are the ones established for assisting refugees to fulfill their needs, in particular, refugee camps. Given a crisis scenario, solving the problem corresponds to choosing the locations where refugee camps will be opened and operated.

Many of variables to be minimized in a refugee crisis, namely quick and efficient attention to the people that need it (expressed as geographical proximity from affected areas to a camp), as well as maximizing the amount of people that can be assisted (expressed as an efficient use of resources), correspond to the optimality criteria of the FLP. It is therefore desirable in both cases that facilities be placed where they will be most efficient at attending demand.

Since the amount of people affected and requiring aid is time-dependent, the problem is modeled as a dynamic instance of the FLP.

In addition, the changing landscape of available locations on which to establish facilities can be thought of as a time-dependent potential for facility failure, which the RFLP can tackle. For example, if during a crisis a border between an affected country and its neighbor is suddenly closed to refugees, facilities located in the neighboring country may be modeled as having failed and rendered disabled.

There are, however, some important differences between the formulation of these two problems.

While facilities for the assistance of refugees may have a fixed capacity, due to the difficulty inherent in estimating affected populations in emergency situations [22], they are often overpopulated, to the detriment of the living conditions of their occupants [23]. A simple capacitated FLP is not able to model such a situation. Fortunately, variants of the FLP that consider "soft" limits on facility capacity, which incur in penalties when exceeded, have been studied and have been shown to be adequately solvable [24].

Due to the delicate and complex nature of these situations, a straightforward cost function that accounts for every aspect, such as the humanitarian consequences and political costs, cannot be formulated. While there is certainly a monetary cost of operating refugee camps, it is only one facet of the overall problem. Numerous additional considerations in a refugee crisis cannot be easily encoded as a numerical optimization problem, notably ones that are geopolitical in nature. For these issues, we defer to the experts that are presented with choices by our interactive simulation.

# 2 Design and Implementation

## 2.1 Video game for solving the RFLP

This work is based on a game that aims to provide solutions to the Robust Facility Location Problem. The User Interface (UI) for the game presents the player with a 2D visualization of demand centers and their connections to their assigned facility. The game is scored by the total cost, which is the sum of: the cost of having the facilities open; and the distance cost, incurred by having facilities far away from the demand centers. Figure 1 shows an example of the visualization the game uses for an RFLP instance.

In the game, demand centers are represented by blue spots, shaded according to the magnitude of the demand. Facilities are represented by black squares. Lines connecting demand centers to the closest facility are shaded according to their contribution to the distance cost, with darker lines being more expensive assignments than lighter ones.

The player has control over whether facilities are opened at each of the demand centers. Whenever a failure occurs, the player must reassign orphaned demand centers to another functioning facility, in order to maintain a complete solution. The game is made up of a succession of short levels, progressing from tutorial levels meant to teach players the mechanics of the game, to scenarios based on real problem instances, where their performance is recorded and used to construct a solution.

The player toggles whether a facility is open at a given demand center by tapping on a spot. To connect an orphaned demand center to a new facility, the player drags their finger from a square to a spot, drawing a line between them.

The game is implemented in the Python programming language, using *Kivy* [25], a framework for multi-touch applications. Playtesting was performed on an iPad, since the interface was designed with direct manipulation in mind.

From the data gathered in a pilot test of the game, we were able to approximate the Pareto set for the Swain [26] and London [27] datasets, approaching, and in some cases outperforming, a genetic algorithm. Figure 2, taken from [28], shows the comparison between the Pareto set obtained for distance before failures by the human players, and the algorithmic approach.

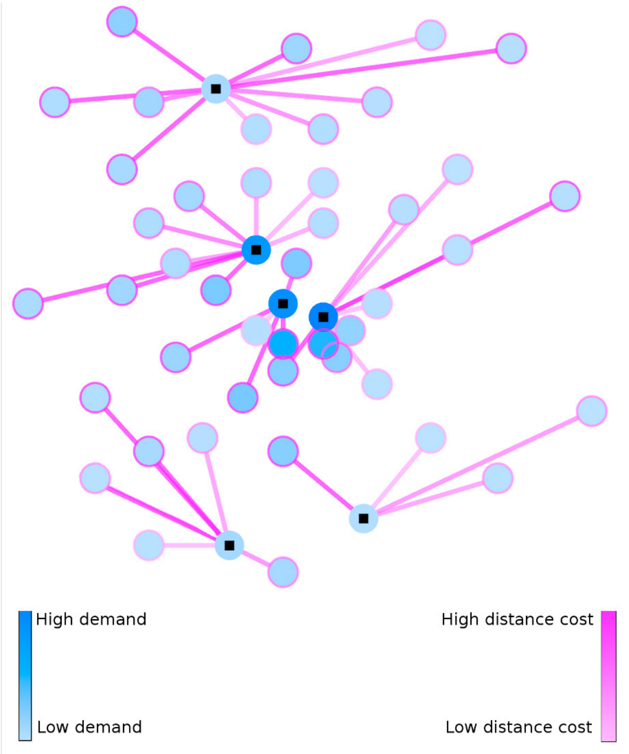


Figure 1: Interface for the original RFLP game showing an instance of the problem. Demand centers (in blue) are shaded according to the magnitude of their demand. Assignments from demand centers to facilities (in purple) are shaded according to their cost.

## 2.2 Serious game for refugee aid deployment

The serious video game focused on the refugee situation simulation is played from the perspective of policy makers, giving the player control over refugee camps and their management. The objective is to find configurations that minimize the cost function, while avoiding situations that the player considers undesirable. While some game actions available to players are carried over directly from the original RFLP game, others are more specific to the refugee simulation.

The player, as in the original game, has contol over opening or closing facilities at eligible locations. The need to manage the simulation of a refugee situation introduces new mechanics: players are able to modify the capacity of individual facilities, in order to expand or reduce the amount of people that they can support and observing the consequences of such a policy in the overall budget and living condition metrics; and players may set movement policies, aimed at moving some part of the population from one facility to another.

Internally, *Flee* is used to compute the location of every refugee at every timestep (see [3] for details). Like the RFLP game, *Flee* is written in Python, which makes the communication between the two components straightforward. While the code for the game manages the graphics and interaction, *Flee* performs the simulation itself.

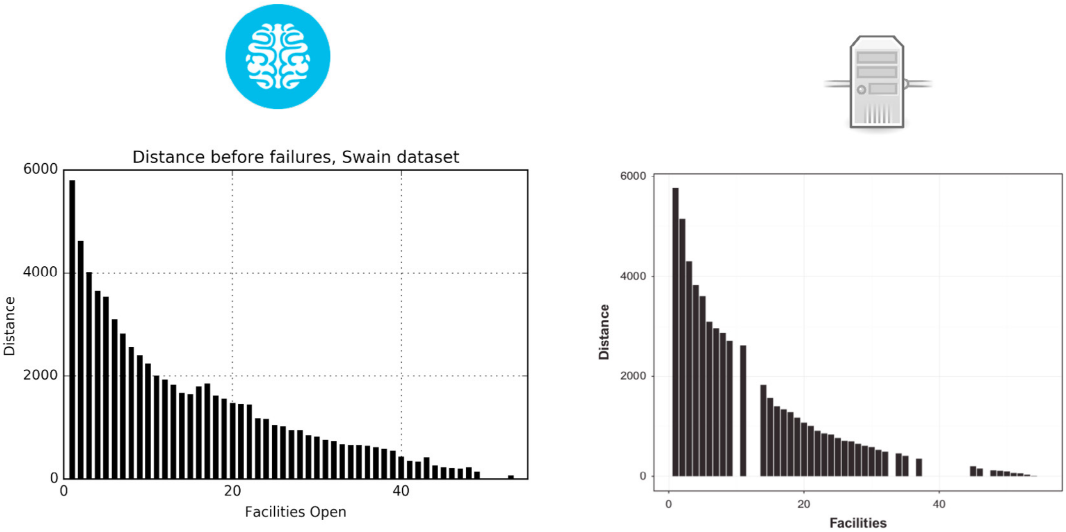


Figure 2: Pareto set approximation for the Swain dataset obtained by human players of the original RFLP video game (left), and a genetic algorithm (right).

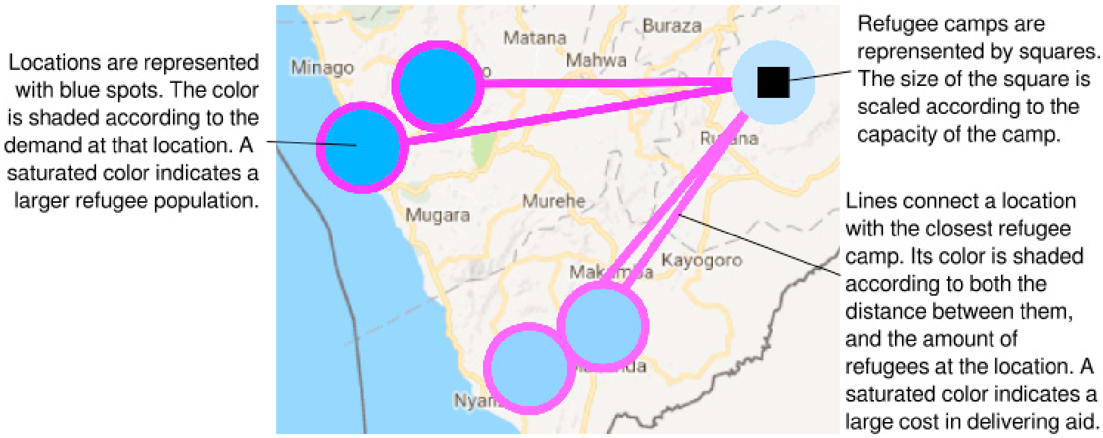


Figure 3: Elements of the visualization for the refugee aid deployment problem.

## 2.3 User interface design

The basic visualization of the problem remains unchanged, with spots representing geographical regions (demand centers), and squares representing refugee camps (facilites). The size of a square representing a refugee camp is scaled according to its population capacity. These elements are shown overlaid on a map of the country or region in which the particular crisis ocurred. Figure 3 details the elements of the visualization.

Players are able to select a spot, which displays a panel of information about the corresponding location. Using this panel, the player can increase or decrease the refugee population capacity at that location. A camp may by closed by lowering the capacity to zero. A red badge indicates refugee camps that are overpopulated, which would indicate a negative impact on the quality of life of its inhabitants. By using the "move" button, the players may put in place a movement policy in order to shift the population from one location to another.

Figure 4 shows the proposed interface for the game, displaying data from the 2015 crisis in Burundi [29].

# 3 Conclusions and Future work

We have presented the design for a video game aimed at optimizing the efficiency of humanitarian organizations at providing aid during refugee crises. Simply by playing the game, the player is able to explore the possible actions that could be taken and policies that could be enacted, and watch their consequences as the situation evolves. In doing so, we have formulated the problem of refugee aid deployment as a softly capacitated, dynamic and robust facility location problem.

Beyond individual exploration of scenarios, as the simulation becomes more sophisticated and realistic, this game has potential for use as a demonstration of real world scenarios and for training people in the enterprise of humanitarian logistics.

Even though the situations represented in the game are highly complex, we believe the design of the game is simple enough it can be used to find approximate solutions even when used by untrained individuals as we did with the original RFLP game.

# Acknowledgements

We thank Diana Suleimenova at the Department of Computer Science, Brunel University, London, for collecting the locations for the Burundi model shown in Figures 3 and 4.

# References

[1] "Figures at a Glance." UNHCR. Accessed January 26, 2017. `http://www.unhcr.org/en-us/figures-at-a-glance.html`.

[2] Edwards, Scott. "Computational tools in predicting and assessing forced migration." Journal of Refugee Studies 21, no. 3 (2008): 347-359.

[3] Groen, Derek. "Simulating refugee movements: Where would you go?" Procedia Computer Science 80 (2016): 2251-2255.

[4] Sokolowski, John A., Catherine M. Banks, and Reginald L. Hayes. "Modeling population displacement in the syrian city of aleppo." In Proceedings of the 2014 Winter Simulation Conference, pp. 252-263. IEEE Press, 2014.

[5] Entwisle, Barbara, Nathalie E. Williams, Ashton M. Verdery, Ronald R. Rindfuss, Stephen J. Walsh, George P. Malanson, Peter J. Mucha et al. "Climate shocks and migration: An agent-based modeling approach." Population and environment 38, no. 1 (2016): 47-71.

[6] Latek, Maciej M., Seyed M. Mussavi Rizi, and Armando Geller. "Verification through calibration: an approach and a case study of a model of conflict in Syria." In Simulation Conference (WSC), 2013 Winter, pp. 1649-1660. IEEE, 2013.

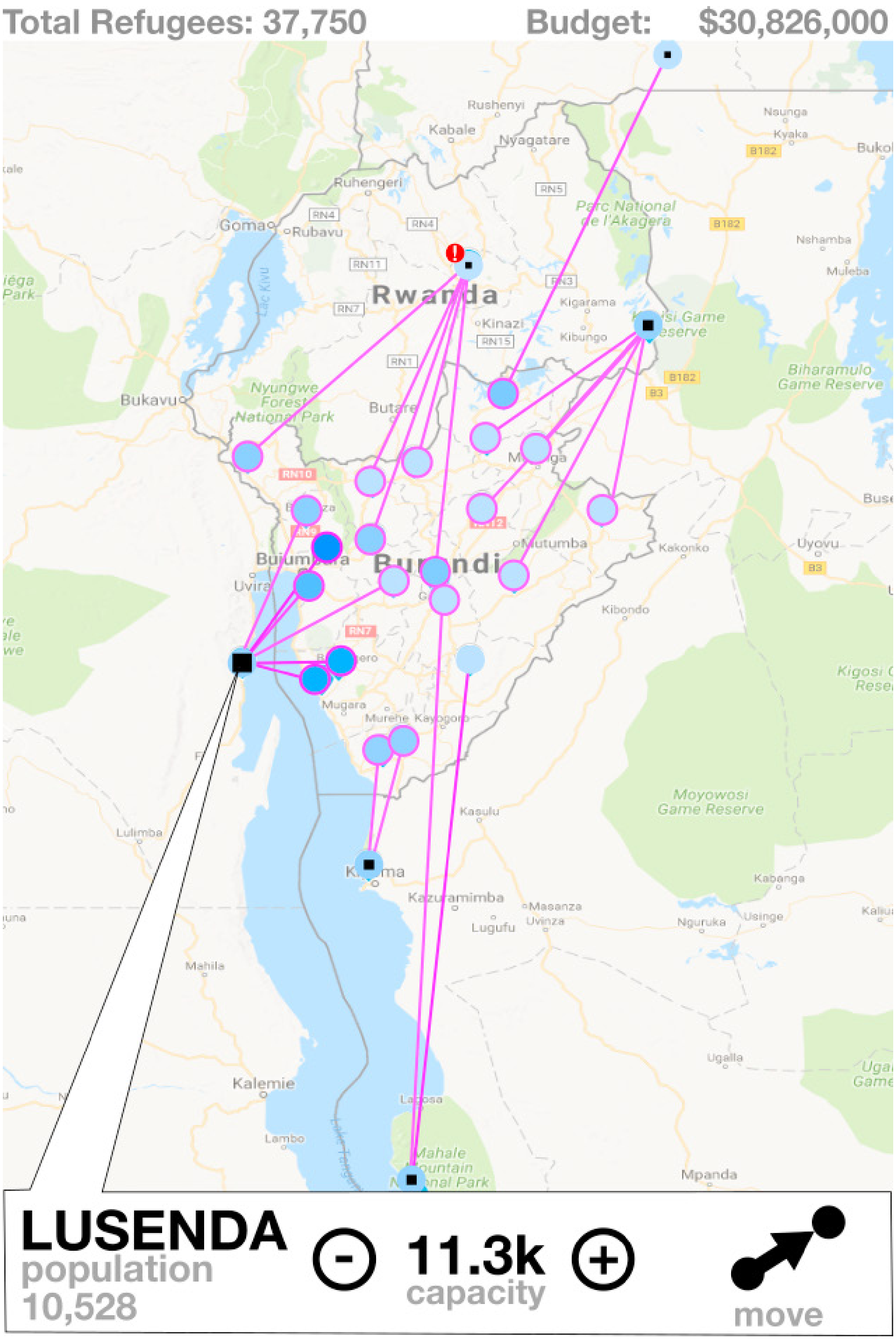


Figure 4: Interface for the Refugee Aid Deployment game, displaying the actions available to the player. By selecting a refugee camp, the player can choose to alter its capacity, or enact a movement policy to another location.

[7] Johnson, Rachel T., Thorsten A. Lampe, and Stephan Seichter. "Calibration of an agent-based simulation model depicting a refugee camp scenario." In Winter Simulation Conference, pp. 1778-1786. Winter Simulation Conference, 2009.

[8] Disney, George, Arkadiusz Wiśniowski, Jonathan J. Forster, Peter WF Smith, and Jakub Bijak. "Evaluation of existing migration forecasting methods and models." ESRC Centre for Population Change, University of Southampton, 2015.

[9] Takagi, Hideyuki. "Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation." Proceedings of the IEEE 89, no. 9 (2001): 1275-1296.

[10] Mavandadi, Sam, Steve Feng, Frank Yu, Stoyan Dimitrov, Richard Yu, and Aydogan Ozcan. "BioGames: a platform for crowd-sourced biomedical image analysis and telediagnosis." GAMES FOR HEALTH: Research, Development, and Clinical Applications 1, no. 5 (2012): 373-376.

[11] Kawrykow, Alexander, Gary Roumanis, Alfred Kam, Daniel Kwak, Clarence Leung, Chu Wu, Eleyine Zarour, Luis Sarmenta, Mathieu Blanchette, and Jérôme Waldispühl. "Phylo: A citizen science approach for improving multiple sequence alignment." PloS one 7, no. 3 (2012): e31362.

[12] Marx, Vivien. "Neuroscience waves to the crowd." Nature methods 10, no. 11 (2013): 1069-1074.

[13] Cooper, Seth, Firas Khatib, Adrien Treuille, Janos Barbero, Jeehyung Lee, Michael Beenen, Andrew Leaver-Fay, David Baker, and Zoran Popović. "Predicting protein structures with a multiplayer online game." Nature 466, no. 7307 (2010): 756-760.

[14] Cornuéjols, Gérard, George L. Nemhauser, and Lairemce A. Wolsey. The uncapacitated facility location problem. No. MSRR-493. Carnegie-mellon univ pittsburgh pa management sciences research group, 1983.

[15] Wesolowsky, George O. "Dynamic facility location." Management Science 19, no. 11 (1973): 1241-1248.

[16] Farahani, Reza Zanjirani, Maryam Abedian, and Sara Sharahi. "Dynamic facility location problem." In Facility Location, pp. 347-372. Physica-Verlag HD, 2009.

[17] Megiddo, Nimrod, and Arie Tamir. "On the complexity of locating linear facilities in the plane." Operations research letters 1, no. 5 (1982): 194-197.

[18] Shmoys, David B., Éva Tardos, and Karen Aardal. "Approximation algorithms for facility location problems." In Proceedings of the twenty-ninth annual ACM symposium on Theory of computing, pp. 265-274. ACM, 1997.

[19] Hernandez, Ivan, Jose Emmanuel Ramirez-Marquez, Chase Rainwater, Edward Pohl, and Hugh Medal. "Robust facility location: Hedging against failures." Reliability Engineering & System Safety 123 (2014): 73-80.

[20] Görmez, N., M. Köksalan, and F. S. Salman. "Locating disaster response facilities in Istanbul." Journal of the Operational Research Society 62, no. 7 (2011): 1239-1252.

[21] Apte, Aruna. "Humanitarian logistics: A new field of research and action." Foundations and Trends® in Technology, Information and Operations Management 3, no. 1 (2010): 1-100.

[22] Noji, Eric K. "Estimating population size in emergencies." Bulletin of the World Health Organization 83, no. 3 (2005): 164-164.

[23] Garfi, Marianna, Simona Tondelli, and A. Bonoli. "Multi-criteria decision analysis for waste management in Saharawi refugee camps." Waste management 29, no. 10 (2009): 2729-2739.

[24] Harkness, Joseph, and Charles ReVelle. "Facility location with increasing production costs." European Journal of Operational Research 145, no. 1 (2003): 1-13.

[25] "Kivy: Cross-platform Python Framework for NUI Development," Kivy Organization, accesed December 1, 2015, `http://kivy.org`

[26] Swain RW. "A decomposition algorithm for a class of facility location problems." Ph.D. thesis, Cornell University, Ithaca, NY, USA, 1971.

[27] Goodchild, Michael F., and Valerian T. Noronha. Location-allocation for small computers. Vol. 8. Department of Geography, University of Iowa, 1983.

[28] Luis Eduardo Perez Estrada, Dante Gama Dessavre, Jose Emmanuel Ramirez-Marquez, and J. Octavio Gutierrez-Garcia. "Optimization via Crowd Computing: A Video Game Approach." Submitted to IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2017.

[29] "Situation Burundi Situation". UNHCR. Accessed January 26, 2017. `https://data2.unhcr.org/en/situations/burundi`.