Simulating Forced Migration with the FLEE Agent-based Modelling Environment: Preliminary Results

Tyler Amos

9 May 2018

Abstract

This is a summary of initial results for a simulation of forced migration, specifically internal displacement. The case study used is Iraq in the period January 2017 through to April 2018.

Can agent-based simulation accurately predict the volume and geographic distribution of internal displacement? This paper uses the FLEE agent-based modelling environment to study internal displacement in Iraq from January 2017 through to April 2018. This section describes the data sources and cleaning algorithms, the FLEE environment and the basic ruleset for agents, and initial results of parameter optimization.

Data

The principal data sources for this analysis are: (i) records of the volume and geographic distribution of internally displaced people collected by the International Organization for Migration (IOM); (ii) records of violent incidents from the Armed Conflict Location and Event Database (ACLED), and; (iii) spatial data for populated locations from the United Nations Office for the Coordination of Humanitarian Affairs (UNOCHA).[1]

IOM Displacement Tracking Matrix (DTM)

IOM conducts regular surveys of the location and number of displaced households in Iraq. Surveys are conducted approximately every two weeks as part of IOM's assessment system.[2] For this simulation, the IDP Master Lists for rounds 84 (November 29, 2017) through 91 (April 30, 2018) were used. On average, each round of the survey has 2,436 cases.

[1] Links to download all data files are available in the Appendix.
[2] See Migration (n.d.) for details of the survey methodology.

IDP Master Lists have consistent formats across rounds, which motivated their selection over more granular round-specific reports. Each Master List provides updated figures of the number of households at each reported location, as well as adding new locations as they are surveyed. Master Lists were downloaded as MS Excel documents and converted to CSV format.

Armed Conflict Location and Event Database (ACLED)

ACLED is an initiative which catalogues incidents of violence across a number of countries. ACLED data are frequently used by researchers studying conflict/crisis. For this simulation, ACLED data for Iraq were accessed from the Humanitarian Data Exchange portal's live update link as a CSV. (Exchange, n.d.)

For each event, ACLED records the approximate location, date and time, estimated fatalities, a short description, as well as other features. This simulation uses the approximate location and the time in the simulation environment. Events with no estimated fatalities were removed from the data under the assumption they are not indicative of a level of conflict sufficient to provoke new displacements. In the complete dataset there are 6,354 cases, 3,730 of which involve at least one fatality. Of these, 704 fit into the specified time period and were used in the simulation.

Populated Locations in Iraq

Spatial data on the location of 23,991 populated places in Iraq was collected from UNOCHA via the HumData portal in ShapeFile format. (Coordination of Humanitarian Affairs, n.d.) This dataset was chosen specifically as it is compatible with the Displacement Tracking Matrix and is derived from IOM's internal placename database. It provides the names and locations of not only official settlements, but neighbourhoods and other unofficial locations, allowing for regional centroids to be weighted by density of settlements, not area.

Source	Usage	Start	End
IOM	IDP Location	2017-11-29	2018-04-30
IOM	IDP Population	2017-22-29	2018-04-30
ACLED	Location Type	2017-01-01	2018-04-28
UNOCHA	Network Node Location	NA	NA

Data Joining and Aggregation[3]

To align the three data sources in the desired format, a series of aggregation and spatial join operations were performed. The spatial geometries of IDP, event, and populated places were transformed into representative polygons using the GeoPandas convex hull functionality after being aggregated to the level of one of Iraq's 18 first-level sub-national administrative regions (governorate/province). On these transformed datasets, two sets of spatial joins were used. First,

[3]Algorithms used to perform these operations are available in the Appendix.

the populated locations were joined with the observed IDP populations data. Population numbers were summed by region to establish the initial and future populations of each location. Second, the populated locations, aggregated to the regional level and represented geometrically as a convex hull, were joined with ACLED records involving at least one fatality. Population location geometries were then transformed into centroids for use as node locations. Lastly, the distance between these centroids was calculated using the GeoPandas `distance` function.

These joins and aggregation produced four datasets: (i) population centres represented as a point, with starting IDP populations (nodes); (ii) final observed IDP populations by population centre; (iii) conflict locations and their date of observation, and; (iv) distance between all population centres (edges).

Missingness

The validation period used in this iteration of the simulation uses observed values from round 91 (late April 2018) to calculate error rates. Round 91 of the Displacement Tracking Matrix does not contain IDP totals for all governorates. The error rate used is calculated based on those governorates for which there are observed values. In future iterations, alternative error calculations will be explored.

Models and Methods

The FLEE Agent-based Modelling Environment

FLEE is a purpose-built Agent-based Modelling (ABM) environment for simulating the flow of people. (Suleimenova, Bell, and Groen 2017) The initial development of the environment has focused on modelling forced displacement, specifically refugee movements. In FLEE, agents traverse a network where each node represents a town, camp, or conflict. Agents follow a series of rules in order to determine where they will travel where conflict and distant locations are less likely to be selected, and non-conflict and proximate locations are more likely.

In this iteration of the simulation environment, each agent represents a household (family). At each step of the ecosystem, in this case a 2-week period, agents navigate the ecosystem according to a set of rules inspired by the gravity model of migration. In short, under the gravity model the relative attractiveness of a location is a function of the population size of the destination location and the distance to that location. In this simulation, the population size is the number of internally displaced people, and the distance is the euclidean distance between points as calculated by the GeoPandas `distance` function. A fixed number of agents (100) are added to locations at random once per step. Agents at those locations then decide to stay or move based on the population of their current location and the distance to other locations, as in the gravity model. In this simulation, there were seven possible parameters which could be adjusted (see below).

Parameters Varied In the Simulation

Name	Description
`CampWeight`	The factor by which camps 'attract' agents.
`ConflictWeight`	Reduction factor for camps.
`MinMoveSpeed`	Minimum distance an agent covers in one step.
`MaxMoveSpeed`	Maximum distance an agent covers in one step.
`ConflictMoveChance`	Default probability for leaving a conflict zone.
`CampMoveChance`	Default probability for leaving a camp.
`DefaultMoveChance`	Default probability for leaving any location.

Parameters and Constructs

Name	Construct (Gravity Analogy)
`CampWeight`	The attractive power of camps (mass)
`ConflictWeight`	The repellent factor of conflicts (mass)
`ConflictMoveChance`	Agents' decision to leave a conflict.
`CampMoveChance`	Agents' decision to leave a camp.
`DefaultMoveChance`	Agents' decision to leave any given location

Apart from these parameters, the simulation was set so that agents introduced added to existing populations (`TakeRefugeesFromPopulation = False`), camp weights were dynamically calculated based on the agent population in the camp at each step (`UseDynamicCampWeights = True`), agent awareness was limited to their location (`AwarenessLevel = 1`), IDP mode was enabled (`UseIDPMode = True`), and agents did not accumulate knowledge about the network over time (`UseDynamicAwareness = False`).

The ecosystem was initialized with locations drawn from the processed list of locations (i) above. The distances between nodes were used to create links between locations (iv). Throughout the simulation, if a location was included in the list of conflict locations for that step, the location was changed to a 'conflict' zone, and the weights agents use to implement their decision function were re-calculated. To evaluate the appropriateness of the simulation parameters, an error function was calculated from the difference in proportions of displaced people in each governorate predicted by the simulation and the true observed proportions of IDPs in each governorate.

Results

Algorithmic Optimization

Algorithmic optimization of the simulation parameters was done using two different algorithms commonly applied to discrete simulations. These algorithms were chosen for their ability to produce a global optimum. Both algorithms were implemented through the `scipy.optimize` library. The objective function used was a mean error calculated from comparing the governorate-level distribution of agents and observed IDPs in the final step of the simulation (round 91 of the IOM DTM).

The first algorithm, the basin hopper algorithm,[4] produced a mean error of 0.09, 0.10 and 0.10 for 3, 5, and 10 iterations, respectively. The optimized parameters, however, include values which are not meaningful, such as negative probabilities.[5] The second algorithm, brute force, was not able to converge in a computationally tractable time period. Future iterations of this simulation will explore parallel processing as a possible solution to this issue, as brute force algorithms will allow greater control over the permitted simulation parameters, thus avoiding the issue of optimizations producing invalid parameters. The potential use of parallelization will be expanded upon in the next section.

Selected Optimization Results (3 Iterations)

Mean Error: 0.096

Name	Optimized Value
`CampWeight`	4.266
`ConflictWeight`	-0.007
`MinMoveSpeed`	10.158
`MaxMoveSpeed`	10.562
`ConflictMoveChance`	-0.090
`CampMoveChance`	-0.207
`DefaultMoveChance`	-0.662

Heuristic Optimization

As an alternative to algorithmic optimization, parameters were entered by hand, based upon simple heuristics (e.g., the chance of leaving a conflict zone is greater than the chance of leaving a non-conflict zone). These results are summarized below. What is notable from an initial review of these results is that the simulation does not appear to be very sensitive to parameters, error is consistently between 8 and 10%.

[4]See https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.basinhopping.html

[5]See Appendix for details of both algorithms' results.

Selected Heuristically-Defined Parameterizations

Case 1:

Mean Error: 0.086

Name	Value
`CampWeight`	100
`ConflictWeight`	0.5
`MinMoveSpeed`	300
`MaxMoveSpeed`	1000
`ConflictMoveChance`	0.1
`CampMoveChance`	0.4
`DefaultMoveChance`	0.1

Case 2:

Mean Error: 0.089

Name	Value
`CampWeight`	2
`ConflictWeight`	0.5
`MinMoveSpeed`	100
`MaxMoveSpeed`	1000
`ConflictMoveChance`	0.1
`CampMoveChance`	0.9
`DefaultMoveChance`	0.8

Case 3:

Mean Error: 0.085

Name	Value
`CampWeight`	100
`ConflictWeight`	0.1
`MinMoveSpeed`	100
`MaxMoveSpeed`	1000
`ConflictMoveChance`	0.8
`CampMoveChance`	0.4
`DefaultMoveChance`	0.1

Case 4:

Mean Error: 0.086

Name	Value
`CampWeight`	1
`ConflictWeight`	0.5
`MinMoveSpeed`	1
`MaxMoveSpeed`	1000
`ConflictMoveChance`	0.1
`CampMoveChance`	0.9
`DefaultMoveChance`	0.8

Further Steps

In the next iteration of this simulation, additional optimization algorithms will be employed and compared to heuristic parameterizations (as above). In order to implement certain algorithms, such as brute force, tools such as multi-thread and parallel processing will be explored. Different and more granular validation measures will be employed to better understand the robustness of the simulation to new test data. In the next week, new data will be available from ACLED and IOM which will help to futher test the performance of the simulation.

Appendix

Data Sources

IOM Displacement Tracking Matrix: http://iraqdtm.iom.int

Populated Places in Iraq: https://data.humdata.org/dataset/settlements-villages-towns-cities

Iraq Administrative Boundaries: https://data.humdata.org/dataset/iraq-admin-level-1-boundaries[6]

ACLED Event Data: https://www.acleddata.com/data/

ACLED Data is also available with live updates from: www.data.humdata.org

Algorithms

Aggregation of Populated Locations

```
generate spatial geometries of all populated locations
dissolve geometries by Governorate
convert geometries to convex hulls
spatial join with second dataset (if needed)
```

Creation of new geometries

```
load non-spatial data with requisite spatial features (lat, lon)
for each lat, lon pair
    create geometry
assign new geometries as a feature of the dataset
```

Calculation of edge lengths

```
load dissolved geometries of Populated Locations
convert geometries to centroids (a single point)
for each row
    calculate the distance between the row's geometry and all other geometries
    convert distances to an integer value
    for each trip
        if the start and end are equal
            move to the next trip
        if the trip has not yet been seen
            add the trip to a master trip list
            write the start, end, and distance to a file
```

Identification of Conflict Zones

[6] This data was used for visualization purposes only.

```
load ACLED conflict data
remove cases with no fatalities
create new geometries (above)
sort by event date
convert event dates to integers
remove cases with event dates before the lower date bound
remove cases with event dates after the upper date bound
assign round number, equivalent to step in simulation
write location names and steps to file
```

Creation of test dataset

```
aggregate population locations
load final observed dataset
create geometries for final observed dataset
spatial join observed counts with population locations
sum observed counts by governorate
write to file
```

Links to Code

Code used to produce the simulation environment, clean, and represent data are available at: https://github.com/tamos/MACS30200proj.

Figures

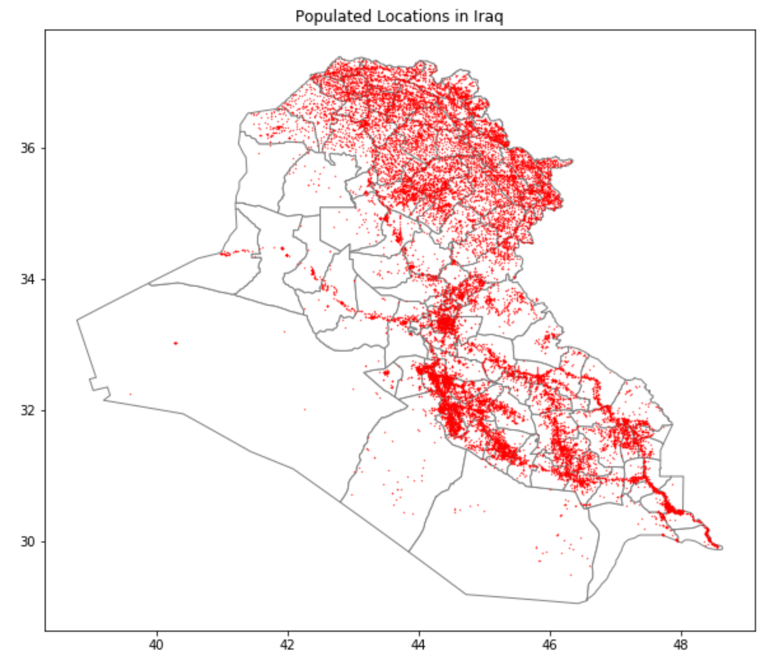

Figure 1: Iraq Populated Places with level 1 administrative boundaries. Source: UNOCHA

Basin Hopper Optimized Parameters

Starting parameter vector of: 5, 0.2, 10, 10, 0.1, 0.1, 0.1

Iterations: 3

Mean Error: 0.0967734444771941

Optimized Parameters: 4.266, -0.00702500877, 10.1585, 10.562191, -0.0907899883, -0.207080421, -0.662580836

Iterations: 5

Mean Error: 0.10395007399424917

Optimized Parameters: 5.3530009, 0.65435315, 9.71979426, 9.41641239, 1.93755486, -0.19553126, 0.69430158

Iterations: 10

Mean Error: 0.10150157686490097

Optimized Parameters: 5.17107326, -0.39410599, 9.92363267, 10.17238293, -0.81284913, 0.67581873, 0.37396598

Brute Force Optimized Parameters

Mean Error: Not Obtained

Optimized Parameters: Not Obtained

Permutations of parameter values were defined in a space of: (0,1,0.1) x (0, 1, 0.1) x (0, 100, 10) x (0, 1000, 100) x (0, 1.0, 0.1) x (0, 1.0, 0.1) x (0, 1.0, 0.1)

Where each tuple represents: (lower bound, upper bound, step value).

References

Coordination of Humanitarian Affairs, United Nations Office for. n.d. "Iraq - Settlements (villages, towns, cities)." https://data.humdata.org/dataset/settlements-villages-towns-cities.

Exchange, Humanitarian Data. n.d. "Iraq - Conflict Data." https://data.humdata.org/dataset/acled-data-for-iraq.

Migration, International Organization for. n.d. "Methodology." http://iraqdtm.iom.int/Methodology.aspx.

Suleimenova, Diana, David Bell, and Derek Groen. 2017. "A Generalized Simulation Development Approach for Predicting Refugee Destinations." *Scientific Reports* 7 (1). Nature Publishing Group:13377.