Modeling Civil Conflict and Aid Delivery in Uganda

Peter J. Bull
Institute for Applied Computational Science
Harvard University
52 Oxford Street
Cambridge, MA 02139
bull@fas.harvard.edu

Isaac M. Slavitt
Institute for Applied Computational Science
Harvard University
52 Oxford Street
Cambridge, MA 02139
slavitt@fas.harvard.edu

Abstract

Using MCMC techniques, we model civil conflict in Uganda and optimize the limited relief aid that can be provided in these scenarios. We describe a method to simulate civil conflict events in space and time given historical data about these events. We also optimize the distribution of limited aid resources to refugees from these crises. Modeling aid delivery as a combination of the traveling salesman problem and the knapsack algorithm — two NP-hard problems — we find acceptable solutions using stochastic metaheuristics.

The Data

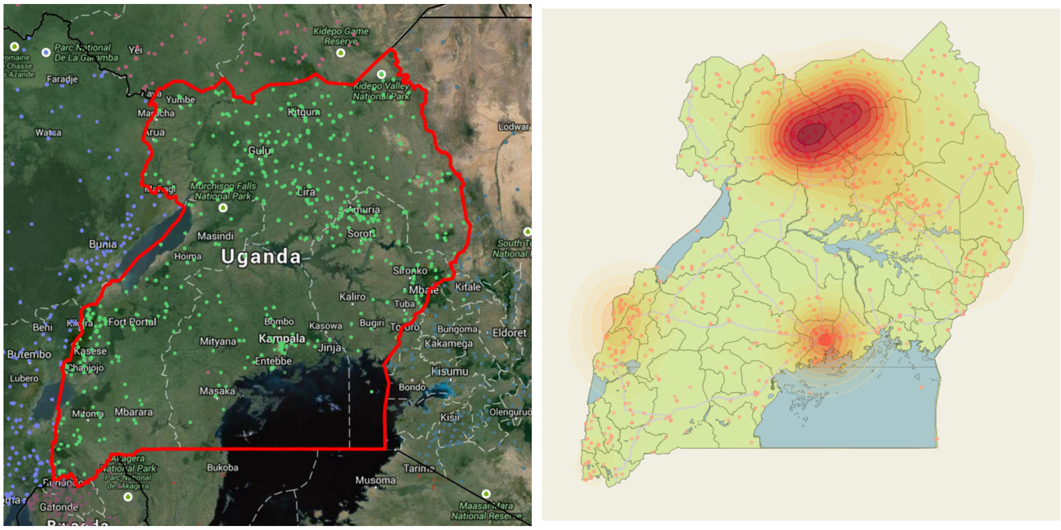
The data comes from ACLED (Armed Conflict Location and Event Data Project), which is a dataset with locations, dates, fatalities, motivation, actors involved, and other information about civil conflicts in Africa. Their collection of data on Uganda covers 1997-2013, and they have a real-time tracker of events reported in 2014.[1] The need for an understanding of these patterns of conflict is clear, as ACLED notes:

> This dataset codes the dates and locations of all reported political violence events in over 50 developing countries. Political violence includes events that occur within civil wars or periods of instability. Although civil war occurrence is decreasing across African countries, new forms of political violence are becoming more common.

Table 1: The ACLED Uganda Dataset

	GWNO	EVENT_DATE	TIME_PRECISION	EVENT_TYPE	LATITUDE	LONGITUDE	GEO_PRECIS	FATALITIES
0	500	1997-01-01	3	Battle-No change of territory	0.50000	32.00000	3	5
1	500	1997-01-01	3	Battle-No change of territory	2.76667	32.30556	3	4
2	500	1997-01-07	1	Battle-No change of territory	0.13580	30.36360	1	5
3	500	1997-01-08	1	Battle-No change of territory	0.18333	30.08333	3	2
4	500	1997-01-11	1	Violence against civilians	3.12583	32.91972	1	0

For Uganda, the dataset contains around 4,500 observations of civil violence. Each observation includes the data, location, an quantified estimate for how precise these measures are, a number of fatalities for the event, and the actors involved. Figure 1a shows these conflicts scattered on a political map of Uganda.



(a) Civil conflicts in Uganda 1997-2013.

(b) Conflicts with Matérn smoothing.

Figure 1: Civil conflicts in Uganda.

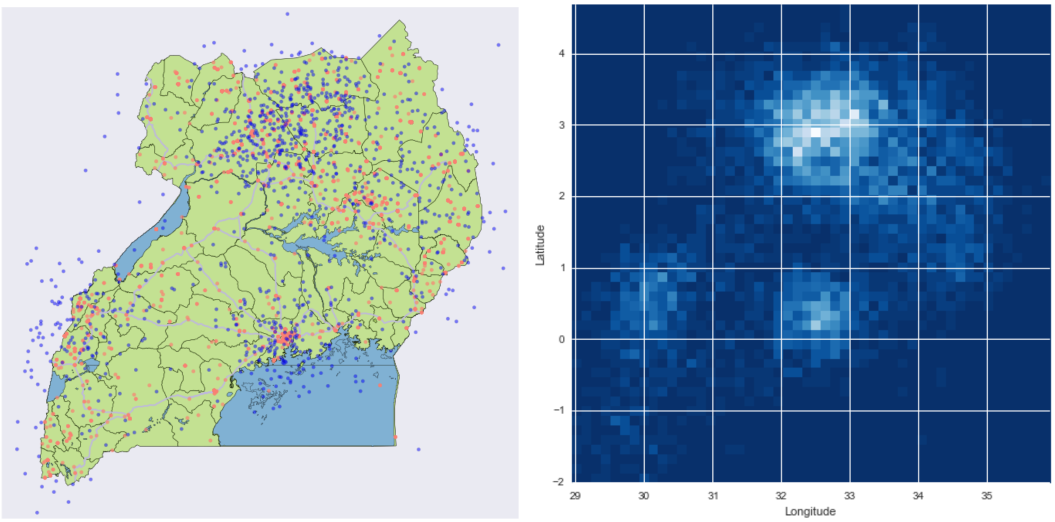
Modeling civil conflict

Events in space

The events of violence and civil conflict that are available from the ACLED dataset are coded with both a latitude and a longitude. One assumption we make is that these events are distributed due to underlying causes such as population centers, road access, historical land ownership, and other features that make conflicts more or less likely. It is difficult to get the data and create a generative model for these probabilities, so we make the assumption that the distribution of the data is representative of these factors. In effect, we treat the entire country of Uganda as a probability distribution from which geospatial conflict events could be sampled. We took historical conflict location data from the entire ACLED data set and smoothed it using a Matérn covariance function. Figure 1b shows this smoothing applied to the same conflicts depicted in 1a.

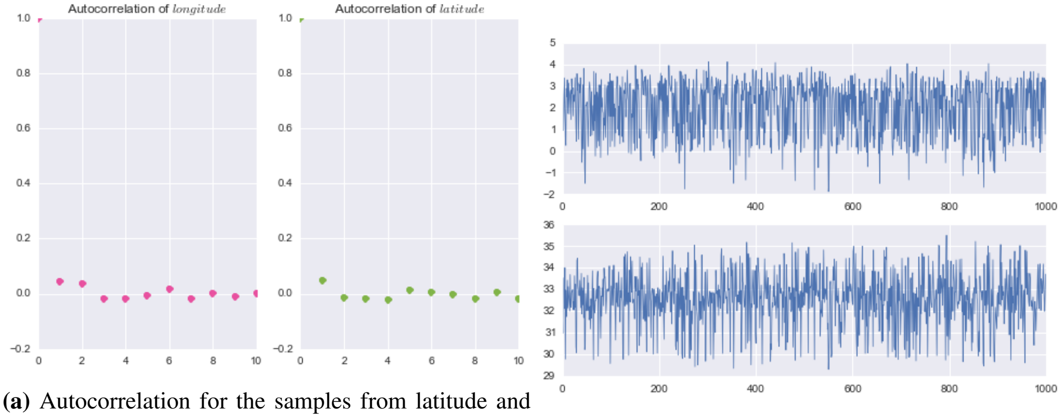
We used this smooth as a kernel-density esitmate. This estimate (i.e., the empirical distribution of the conflict data), has a complex functional form which makes it challenging to sample from. However, it is simple for a given coordinate to get the probability of an event. Given this property of our smooth, we can apply monte-carlo sampling techniques to generate samples from this probability distribution. Visualizing the distribution, we see that it is multi-modal with regions of low density between the modes. Because of these properties, we opted to use slice sampling to generate draws from the distribution. Figure 2a shows the first 1,000 samples from this probability distribution (note, as some point we could write code to throw away samples that occur over water or outside of country boundaries). Figure 2b shows the distribution of the samples as a two-dimensional histogram.

When slice sampling, there are a number of parameter choices that are imporant. We want to choose our rectangle widths, burn-in, and thinning parameters appropriately. In testing, a thinning value of 10 reduced autocorrelation to less than 0.1 at a time lag of 1. We can see this result in Figure 3a. We used the Gelman-Rubin potential scale reduction factor [2] to determine if we were mixing well. Generally, a value less than 1.1 indicates good mixing. In both of our dimensions, the Gelman-Rubin statistic was less than this threshold. We also calculated the Gewecke statistic, which is used to indicate convergence. A value less than 2 indicates convergence and for our draws, this statisic was $\ll 1$. We can also examine convergence by looking at the trace plots for the samples. As we see in Figure 3b these appear stationary.



(a) Blue dots are samples from the empirical distribution.

(b) 2d histogram of slice samples from the empirical distribution.



(a) Autocorrelation for the samples from latitude and longitude

(b) Trace plots for draws from the latitude and longitude

Events in time

As a modeling assumption, we seperate the dimensions of space and time as being independent. To model events in time across the country, we use an autoregressive Poisson-GLM. While standard autoregressive models create a linear relation between a future value and a previous value, the Poisson GLM let's us have a linear relation between previous data and the mean of a Poisson distribution. This will allow us to retain the probabilistic interpretation of the events in time.

In order to model events using a Poisson distribution, we must discretize our time dimension. We opted for month-long increments. The thinking behind this decision is that we want to use sample draws to run our aid optimizations. If a model like this were to be used in planning for future conflicts, having a month-long window for a plan seems like a good balance between precision and logistic concerns.

TODO: Finish code and incorporate results.

Putting it together

We can now use our draws in space and time to generate a simulation of further conflicts in Uganda. These simulated scenarios will be the basis of our adaptive aid-placement and allocation optimization. A model of conflict events in space and time, along with optimized aid-delivery could help organizations such as the Red Cross to order supplies, allocate staff and volunteers, and develop infrastructure.

Optimizing Resource Placement and Allocation

In the second part of this project, we use the first section as both inspiration for the aid delivery analogy and also as a source of randomly sampled data points representing geospatially distributed conflicts.

The Traveling Salesman Problem

One question of particular interest that we can tackle using techniques from the class is how to route emergency aid to locations where it is needed. For concreteness, let's postulate a Red Cross medical or food supply caravan that originates from the organization's in-country headquarters. This caravan wishes to visit all n emergent locations in order to deliver needed supplies. They wish to do so in the most efficient manner possible.

This is fundamentally an optimization problem, and one that is well known – it was first described in 1932 by Karl Menger (shortly after his year here at Harvard as a visiting lecturer) and has been studied extensively ever since.[3] Here is the traditional convex optimization specification of the problem:[4]

$$\begin{aligned}
&\min \sum_{i=0}^{n} \sum_{j\neq i, j=0}^{n} c_{ij} x_{ij} && \\
&\text{s.t.} && \\
&\quad x_{ij} \in \{0,1\} && i,j = 0, \cdots, n \\
&\quad \sum_{i=0, i\neq j}^{n} x_{ij} = 1 && j = 0, \cdots, n \\
&\quad \sum_{j=0, j\neq i}^{n} x_{ij} = 1 && i = 0, \cdots, n \\
&\quad u_i - u_j + n x_{ij} \leq n-1 && 1 \leq i \neq j \leq n
\end{aligned}$$

As is clear from the specification, this is an integer linear program (ILP) where:

- x_{ij} is a binary decision variable indicating whether we go from location i to location j.
- c_{ij} is the distance between location i and location j.
- The objective function is the sum of the distances for routes that we decide to take.
- The final constraint ensures that all locations are visited once and only once.

Answering complex, realistic questions

Packing the aid truck - adding the Knapsack Problem

We extend the TSP into a multi-objective optimization problem where *the contents of the aid trucks* also have an optimization component. Therein lies the knapsack problem: subject to a volume or weight constraint, and given that different locations might have very different needs such as food, vaccinations, or emergent medical supplies, *which supplies do we pack on the trucks*?

Here's the unbounded[1] version of the knapsack problem:

$$
\begin{aligned}
&\max \sum_{i=1}^{n} v_i x_i \\
&\text{s.t.} \\
&\quad x_i \in \mathbb{Z} \\
&\quad x_i \geq 0 \\
&\quad \sum_{i=1}^{n} w_i x_i \leq W
\end{aligned}
$$

In this formulation:

- x_i is a zero or positive integer decision variable indicating how many units of item i we load on the truck.
- v_i is the utility we get from bringing along item i.
- w_i is the weight of item i.
- W is the maximum weight the truck can carry.

The problem, of course, is that brute force solution of the TSP is $\mathcal{O}(n!)$. Traditional, deterministic algorithm approaches such as branch-and-bound or branch-and-cut are still impractical for larger numbers of nodes. In many cases, exhaustive search for global optimality is not even particularly helpful as long as the solution found is good enough. We will use simulated annealing (SA) to get acceptable solutions to the TSP (c.f. the class lectures and homework problem).

Finding the optimal site for the resupply location

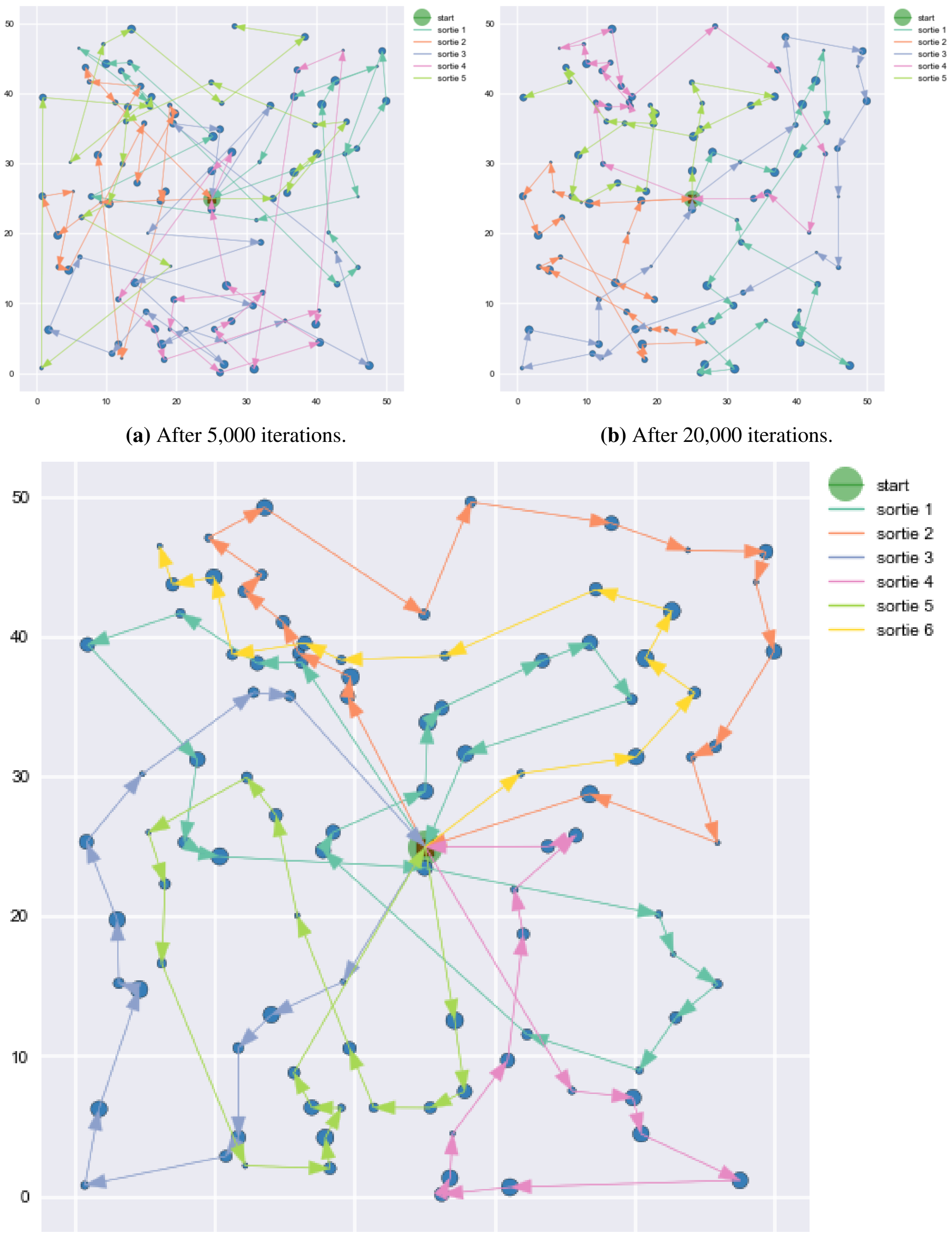
TODO: At the moment, our code draws random cities. We need to update this code to use the samples from our slice sampling instead.

TODO: We are working on the code that incorporates finding the optimal location of HQ.

Factoring in travel time and loss from service delays

Since we have draws in space and time, we want to update our cost function. As refugees build up, injuries go untreated, and citizens become more desperate for supplies, sending the aid truck to a particular location becomes more important. We want to incorporate this idea by including a

[1] Often, this problem is formulated such that you can only bring one of each item, but that doesn't make sense here. We want to be able to bring as many types of each type of aid as we think necessary, and we'll assume that as many as desired are available to load on the trucks before starting out from HQ.



(a) After 5,000 iterations.

(b) After 20,000 iterations.

(c) After 100,000 iterations.

Figure 4: Routing for the TSP/Knapsack hybrid.

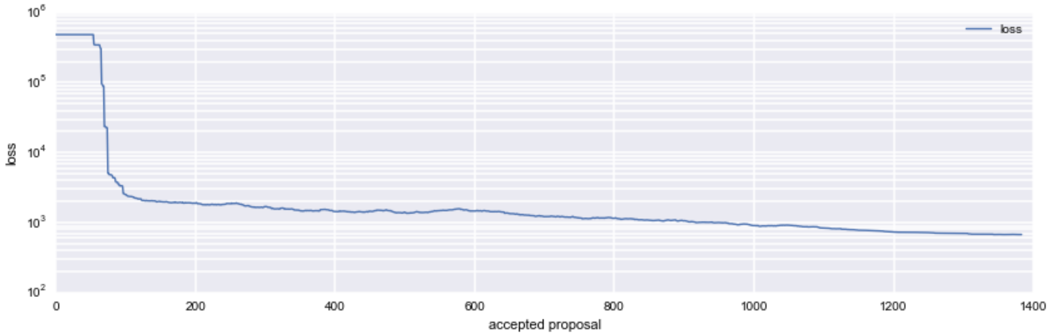


Figure 5: Loss function acceptances over 100,000 iterations.

time-sice-event term in our cost function. As time passes and the site remains unvisited, it becomes more important for the aid truck to visit these locations.

TODO: Incorporate this code and results.

References

[1] C. Raleigh, A. Linke, H. Hegre, and J. Karlsen, “Introducing ACLED-armed conflict location and event data,” *Journal of Peace Research*, vol. 47, no. 5, pp. 1–10, 2010.

[2] G. A. and D. Rubin, “Inference from iterative simulation using multiple sequences,” *Statistical Science*, vol. 7, p. 457511, 1992.

[3] K. Menger, “Das botenproblem,” *Ergebnisse eines Mathematischen Kolloquiums*, vol. 2, pp. 11–12, 1932.

[4] W. Winston, *Operations Research: Applications and Algorithms*. Thomson Brooks/Cole, 2004.