

# Truncated Lévy Flights to Fight Crime

Abram Schönfeldt

September 2019

This paper starts by reviewing research done on modelling opportunistic crime in residential areas and developments made in how criminal behaviour is modelled. It describes the development of code to run mean field simulations based on a model that uses truncated Lévy Flights<sup>1</sup> to model criminal mobility, and key results obtained from these simulations are compared to those obtained in the original paper. Finally, the model is adapted to model the behaviour of individual criminals and direct stochastic simulations are carried out. This is done to demonstrate a thorough understanding of the underlying model and to illustrate, by using individual criminals as opposed to a theoretical distribution of criminals, the relationship between repeat victimization and the development of predictable crime hot spots.

## Introduction

FOLLOWING a series of violent opportunistic crimes <sup>2,3</sup>, the student community of the University of Cape Town face frightening times. As one of these students, the spaces that I navigate daily, to and from university, feel vulnerable to the unpredictable. As a measure to improve student safety, there has been a noticeable increase in the number of security guards placed along routes from university campus to university residences. How do we know that the placement of these guards is optimal? The prospect of being able to understand the patterns of opportunistic crime is one of the primary motivations behind pursuing this topic. This understanding could then be harnessed to minimize the amount of opportunistic crime.

The main model that underpins this paper is one that considers residential burglary. The appeal of modelling residential burglary is partly simplicity. It involves stationary sites, so only the mobility patterns of the criminals have to be modelled. As we shall soon see, there have been a number of attempts at describing these mobility patterns. The seminal model, which will subsequently be referred to as the *UCLA model*<sup>4</sup> (a convention adopted in much of the literature), makes use of a random walk, biased towards areas of high attractiveness. The concept of attractiveness is partly defined by various underlying environmental properties, such as population density and site vacancy, and partly defined by the self-exciting nature of crime itself. In the context of the UCLA model, this self-exciting nature is explained with reference to 'empirically observed phenomena'<sup>5</sup> such as the broken windows effect, and the repeat and near-repeat vic-

<sup>1</sup> Chaohao Pan, Bo Li, Chuntian Wang, Yuqi Zhang, Nathan Geldner, Li Wang, and Andrea L. Bertozzi. Crime modeling with truncated lévy flights for residential burglary models. *Mathematical Models and Methods in Applied Sciences*, 28(09):1857–1880, May 2018. DOI: 10.1142/s0218202518400080

<sup>2</sup> Vincent Cruywagen. Chilling evidence of uyinene mrwetyana's last moments and how she fought back, Sep 2019. URL <https://www.iol.co.za/capeargus/news/chilling-evidence-of-uyinene-mrwetyanas-last-moments-2019-09-10>

<sup>3</sup> Jenna Etheridge. Clifton stabbing: 'he had warned his sisters to be safe' - slain uct student's family, Sep 2019. URL <https://www.news24.com/SouthAfrica/News/clifton-stabbing-he-had-warned-his-sisters-to-be-safe-20190910>

<sup>4</sup> M. B. Short, M. R. Dorsogna, V. B. Pasour, G. E. Tita, P. J. Brantingham, A. L. Bertozzi, and L. B. Chayes. A statistical model of criminal behavior. *Mathematical Models and Methods in Applied Sciences*, 18(suppo1):1249–1267, 2008. DOI: 10.1142/s0218202508003029

<sup>5</sup> Sorathan Chaturapruek, Jonah Breslau, Daniel Yazdi, Theodore Kolokolnikov, and Scott G. McCalla. Crime modeling with Lévy flights. *SIAM Journal on Applied Mathematics*, 73(4):1703–1720, 2013. DOI: 10.1137/120895408

timization effect.<sup>6</sup> The broken windows effect links visible traces of past crimes to the increase in likelihood of future crimes. Other than these visible traces, there is also the acquired experience of having committed a crime in an area. Once a criminal has been successful in burglarising one house in a neighbourhood, they have a sense of the sort of valuables to expect across other houses in the network, what sort of security they will likely encounter, as well as other details that give them a sense of the relative ease or difficulty of carrying out burglaries in the area. Repeat and near-repeat victimization are the terms we use to describe this tendency of residential burglars to return to places that have previously been burgled. Following the UCLA model, several adaptations to the model have been considered. These adaptations primarily involved considering different mobility patterns to for the criminals. While the UCLA paper made use of biased random walks, subsequent papers investigate more accurate ways of describing human mobility patterns. A 2013 paper by Chaturapruek et al. introduced Lévy Flights to model the mobility of criminals. Lévy Flights have been observed as a more accurate way of describing the mobility patterns of animals foraging in an environment since it permits long jumps.<sup>7</sup> These long jumps do not occur in biased random walks. This project starts by considering the 2018 paper by Pan et al. which makes use of a truncated Lévy Flight.

### *Truncated Lévy Flight Model*

#### *Explanation of the Model*

This particular variation of the UCLA model makes use of Truncated Lévy Flights to model criminal behaviour. This will be referred to as the *TLF model*. We start by familiarising ourselves with the components of the TLF model.

34 *sites* are placed at constant intervals,  $L$ , along a one-dimensional grid. This grid has periodic boundary conditions, which means that a criminal moving out of one side of the grid will reappear at the beginning of the other side of the grid. We want to record the mean number of criminals  $n_k(t)$  as well as the attractiveness  $A_k(t)$  of each site  $k$  at each time step. As will soon be discussed, the initial values of attractiveness at each site persists as what is called the 'static background term.'

Attractiveness is defined by two terms. A static background term,  $A_k^0$ , which represents the inherent appeal of site  $k$ , and a dynamic term,  $B_k(t)$  which captures the broken windows effect and the impact of repeat and near-repeat victimization. These phenomena are briefly discussed in the introduction of the UCLA model. The equations for

<sup>6</sup> James Q. Wilson George L. Kelling. Broken windows, March 1982. URL <https://www.theatlantic.com/magazine/archive/1982/03/broken-windows/304465/>

<sup>7</sup> Injong Rhee, Minsu Shin, Seongik Hong, Kyunghan Lee, Seong Joon Kim, and Song Chong. On the levy-walk nature of human mobility. *IEEE/ACM Transactions on Networking*, 19(3):630–643, 2011. DOI: 10.1109/tnet.2011.2120618

Sites represent residential houses in this model

Attractiveness as defined from the perspective of the criminal. I imagine unoccupied homes with low walls are particularly attractive

The *broken windows effect* - visible traces of past crimes may increase the likelihood of future crimes  
*Repeat victimization* - the tendency of residential burglars to return to places that have previously been burgled

$A_k(t)$  and  $B_k(t)$  are given by:

$$A_k(t) = A_k^0 + B_k(t) \quad (1)$$

$$B_k(t + \delta t) = [(1 - \eta)B_k(t) + \frac{\eta}{2}(B_{k-1}(t) + B_{k+1}(t))](1 - \omega\delta t) + \theta\delta t A_k(t)n_k(t) \quad (2)$$

The constants  $\eta$ ,  $\theta$ , and  $\omega$  and the term  $\delta t A_k(t)n_k(t)$  are explained in Table 1. The term  $\eta/2(B_{k-1}(t) + B_{k+1}(t))$  takes the average of the dynamic terms of the sites neighbouring site  $k$  and is used to model how, if a crime happened in one of these neighbouring sites, the increased attractiveness following that crime would 'carry over' to site  $k$ .

Constant	Explanation
$\eta \in (0, 1)$	The strength of the repeat victimization effect
$\theta$	The amount that local attractiveness increases following a burglary
$\omega$	The rate of decay of increased attractiveness inspired by recent burglaries
$\delta t A_k(t)n_k(t)$	The expected number of burglaries in the time interval $(t, t + \delta t]$ at site $k$ .

Table 1: Explanations of the constants associated with the dynamic term  $B_k(t)$  in equation 2

Having established how attractiveness changes as burglaries are committed, we need to now explain how this affects the movement of criminals from site to site. In the TLF model, the probability of a criminal visiting site  $i$  is the weight of site  $i$  over the sum of the weights of all the possible sites that the criminal can jump to, i.e.  $q_{k \rightarrow i}(t)$ , the probability a criminal jumps from site  $k$  to  $i$ , is given by:

$$q_{k \rightarrow i}(t) = \frac{w_{k \rightarrow i}(t)}{\sum_{\substack{j \in \mathbb{Z} \\ j \neq k}} w_{k \rightarrow j}(t)}, k \neq i \quad (3)$$

These weights,  $w_{k \rightarrow i}$  are determined by the attractiveness, as well as the distance to site  $i$

$$w_{k \rightarrow i} = \begin{cases} \frac{A_i(t)}{l^\mu |i - k|^\mu}, & 1 \leq |i - k| \leq L \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

Since a truncated Lévy flight is being used, the list of possible sites that a criminal at site  $k$  can jump to is limited to all the sites that have indexes  $i$  that are within  $L \in \mathbb{N}$  hops. When  $L \rightarrow \infty$  we obtain a Lévy flight, and when  $L = 1$ , we obtain a biased random walk. The other constant that plays a role in criminal mobility is  $\mu \in (1, 3)$

It follows that sites with higher attractiveness are weighted more, so criminals are more likely to move to more attractive sites and sites that are further away are weighted less, so criminals are less likely to make long hops

which models the underlying power law of the Lévy flight. Within the context of this application to residential burglary, it determines the relative range of mobility of a criminal. Professional criminals will be capable of a greater range of mobility, which is reflected in a lower value of  $\mu$ , in comparison to an amateur criminal.

The final equation in the TLF model ties together the previous ones and governs how the distribution of criminals changes over time.  $n_k(t + \delta t)$ , the expected number of criminals at site  $k$  in the next time step, is given by:

$$n_k(t + \delta t) = \sum_{\substack{i \in \mathbb{Z} \\ 1 \leq |i-k| \leq L}} [1 - A_i(t)\delta t] n_i(t) q_{i \rightarrow k}(t) + \gamma \delta t \quad (5)$$

where  $\gamma$  is the rate at which new agents are replaced at each site.  $[1 - A_i(t)\delta t] n_i(t) q_{i \rightarrow k}(t)$  represents the expected number of criminals in the time step  $[t, t + \delta t)$  who do not commit a crime at their site, and consequently move to site  $k$ .

### *Translating the model into code*

We start by defining all the relevant intervals, time steps, grid spacing, etc. alluded to in the previous section.

We also need to define the initial number of criminals and inherent attractiveness of each site, for which we use the Python library NumPy<sup>8</sup> and initialize these arrays as  $n = \text{np.ones}(\text{no\_sites})$  and  $A\_0 = 1 - 0.4 * \text{np.cos}(4 * \text{np.pi} * x)$ . Functions  $\text{updateA}()$  and  $\text{updateB}()$  are defined to simplify the process of updating the arrays  $A$  and  $B$  at each time step and are derived from equations 1 and 2. Making use of NumPy means that updating the array  $A$  can be achieved by simply adding the arrays  $A\_0 + B$ . The array  $B$ , capturing the dynamic term of each site at each time step, is updated by looping through the sites one by one and determining

$$B\_next[k] = ((1 - \eta) * B[k] + (\eta/2) * (B[k-1] + B[k+1])) * (1 - \omega * dt) + \theta * dt * A[k] * n[k]$$

The only complication in updating  $B$  comes from the boundary conditions which requires the first and last site to be updated separately.

Next, we translate the component of the model that dictates the mobility patterns of the criminals, i.e. equations 3 and 4. The weight function,  $w(k, i)$ , is defined first. This function returns weights proportional to the attractiveness  $A[i]$  of the destination site at index  $i$ , and inversely proportional to the amount of jumps required to get to site  $i$ . Again, due to periodic boundary conditions, there are two possible directions that a criminal can move in if they are jumping from their current site to a destination site. The code for the weight

Here, the sum ends before the first +

```
dt = 0.01
no_sites = 60
#domain
dom = [0,1]
#grid lattice spacing
l = (dom[1]-dom[0])/no_sites
#site locations
x = np.arange(dom[0], dom[1], l)
```

<sup>8</sup> Travis E Oliphant. *A guide to NumPy*, volume 1. Trelgol Publishing USA, 2006

function has to firstly determine which of the two possible directions results in the shorter jump. It then determines whether the destination is within  $L$  jumps of the current site, and if it is, it returns  $A[i] / ((1/\mu) * (hop/\mu))$ . Otherwise, it returns 0. Due to the way the code has been implemented, we never expect a weight of zero to be returned. Only neighbours within  $L$  jumps of the current site are looped over. This is illustrated in  $q(k, i)$  which loops over all the neighbours to the left and to the right of the current site within  $L$  jumps, summing together their weighting functions.  $q(k, i)$  then returns the weight of the destination site  $i$  divided by the sum calculated in the previous step.

The implementation of equation 5 is done similarly to  $q(k, i)$ . The sites  $i$  neighbouring site  $k$  within  $L$  jumps are looped over and the expected number of criminals at site  $i$  that did not commit crimes and that chose to move to site  $k$   $((1-A[i]*dt)*n[i]*q(i, k))$  are added to the expected number of criminals at site  $k$  for the next time step.

```
def q(k,i):
    denom = 0
    for j in range(1,L+1):
        denom+=w(k,(k+j)%no_sites)
        denom+=w(k,(k-j)%no_sites )
    return w(k,i)/denom
```

## Results

Using a truncation length of  $L=3$  and the arrays  $(A_0, B, n)$  initialized above, the TLF model was simulated for 8 seconds (or  $8/dt$  time steps). For each time step, the following simple procedure was carried out: The mean number of criminals and attractiveness of each

- 1 Call `updatedB()`, `updatedA()` and `updateN()`,  
and stores these values as `B_next`, `A_next` and `n_next`
- 2 Set `B, A, n = B_next, A_next, n_next`

site after 8 seconds are shown in figure1. Comparing these results to those obtained by Pan et al. is done by inspection. The ticks on the y-axis of their plots are somewhat sparse. For their plot of attractiveness at each site after 8 seconds with a truncation length of 3, they obtain the same shape as the one in figure1 with the absolute maximums of their curves just below 9, and their absolute minimums lying just above  $2/5^{th}$  of the way between 1 and 9 (did I mention the sparsity of the ticks) which visually corresponds to what we see in figure1. Similarly, for their plot of the mean criminals at each site, the shape and absolute minimums and maximums of their curve corresponds to what we see in figure 1. To compare the mean number of criminals and attractiveness for simulations of varying duration and for different truncation lengths, follow the link the Appendix to the Notebook and change the variables  $L$  and  $T$ . These comparisons were conducted were conducted for  $L = 1, 3, 7$  and  $T = 8, 20$ , and in each case the observed curves were unidentifiable different from the

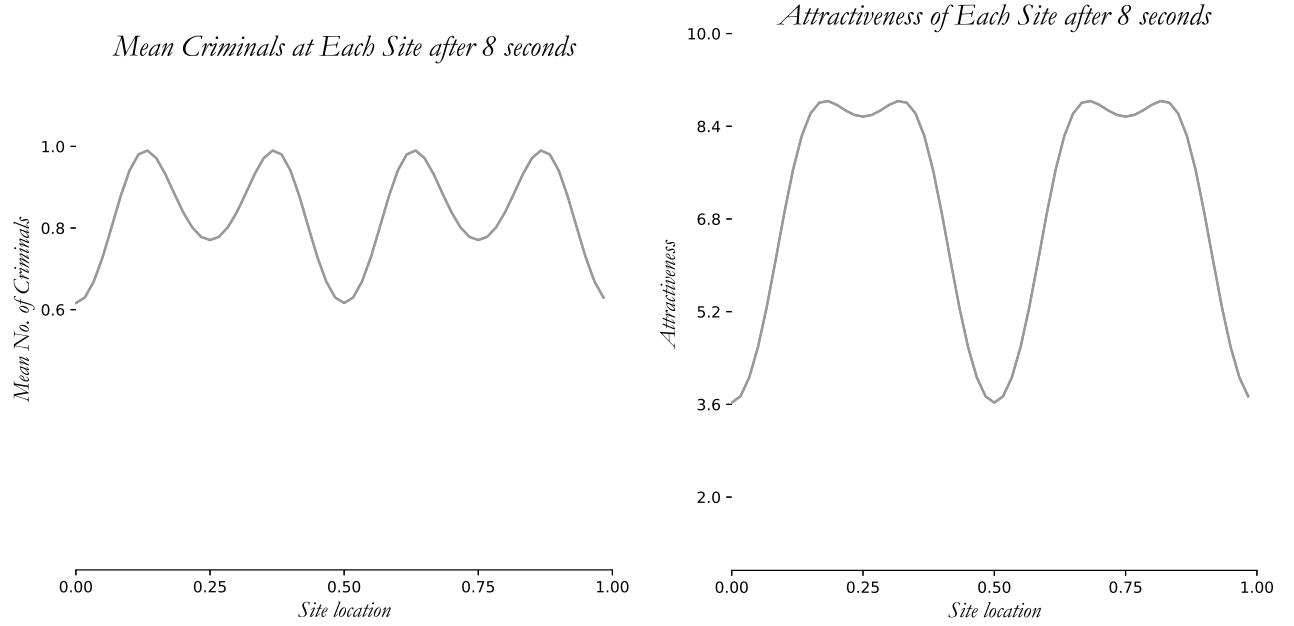


Figure 1: Mean number of criminals and attractiveness of each site at  $t = 8$  with truncation length  $L = 3$

expected curves.

### *Direct Stochastic Simulation of Criminal Behaviour*

#### *Modifying the Model*

The TLF model makes use of a flowchart to summarize the decisions that a criminals make each time the system is updated. The essence of this flow is:

1. Calculate the burglary probability  $p_k(t)$
2. (a) Burgle  $\implies$  remove burglar from the grid  
 (b) Do not burgle  $\implies$  calculate  $q_{k \rightarrow i}$  and move burglar to selected site
3. Update the dynamic component of attractiveness
4. Place new criminals at rate  $\gamma$

This flow is understood a lot more intuitively by picturing the behaviour of individual criminals. There are other reasons to adapt the model to simulate the behaviours of individual agents. It presents an opportunity to assign individual criminals their own mobility ranges, their own paces of moving through an environment and their own level of receptivity to attractiveness. We develop a very simple model

based the TLF model which we then use to carry our direct stochastic simulations of criminal behaviour.

We again work on a one dimensional grid with periodic boundary conditions, and with discrete time steps. However, I did make subtle tweaks to the initial setup of the sites. I placed the sites 20m apart, and worked with 16 sites. This corresponds to two blocks of 8 houses spaced 20m apart. This setup was inspired by the layout of the street I grew up on (where we experienced one traumatic break-in) where properties are approximately 20m wide.

Since we are now tracking the movements of distinct criminals, an object to contain the data associated with each criminal had to be created. This object, named 'Criminal' contains the fields:

idno: an id number to distinguish between multiple criminals

dest: the index of the site the burglar is moving towards

path: the sequence of x-coordinates to get to the destination site

The dynamic term  $B_k$  now changes according to the actual number of crimes committed in the time period  $[t, t + \delta t]$ . We denote actual number of crimes as  $C_k$ , which gives us the following modification to equation 2:

$$B_k(t + \delta t) = [(1 - \eta)B_k(t) + \frac{\eta}{2}(B_{k-1}(t) + B_{k+1}(t))](1 - \omega\delta t) + \theta C_k$$

Since we have moved away from calculating the expected number of criminals committing a crime at each time step, we now have to come up with a way to determine, if a criminal is a site, whether they commit the crime or not. In the long term, the frequency of committing crimes should match up with what we previously had. The probability of a criminal committing a crime is given by  $p_k(t) = 1 - e^{-A_k(t)\delta t}$ . We make use of uniformly distributed random numbers between 0 and 1, and a cutoff value in order to make the decision of whether the criminal commits the crime or not. This cutoff is determined by the  $p_k$ . For instance, if  $p_k = 0.3$  then for any random number generated  $\leq 0.3$ , we would say that the criminal commits the crime and remove them from the grid.

The probabilities  $q_{k \rightarrow i}(t)$  that the criminals will move from site  $k$  to site  $i$  are as before. Again, we employ a similar strategy as before to translate these probabilities of moving to each site to a discrete decision to move to a single site. We now define multiple cutoff regions, each associated with one of the possible sites that the criminal can move to, where the size of these regions is proportional to the probabilities obtained from the function  $q(k, i)$ . This is implemented in pseudo code below:

```
p = 1 - np.exp(-A*dt)
def burgle(k):
    if rnd.random() <= p[k]: return True
    return False
```

```

def hop(k):
    choice = random number in [0,1]
    region = 0
    for i in possible sites
        region += q(k,i)
    if choice <= region:
        return i

```

We also define speed, the speed at which criminals move between sites, which, for the purpose of these simulations, will be in meters per second. We assume that the average walking speed of a person is about 1.4 meters per second. This means that criminals change their position by  $1.4 \cdot dt$  m every time step.

We now have all the tools to implement the first 3 steps of the flow described at the beginning of this section, we are left with implementing the last step - the rate of introducing criminals at each site. Realistically, we would expect about only 1 or 2 criminals to appear at a site every hour. This translates to being  $2/(60 \cdot 60) = 1/1800$  every second. There are  $1/dt$  time steps in every second, so in a given time step, we expect  $dt/1800$  to appear at a site. We use `numpy.random.poisson` to generate the number of criminals to introduce to each site at each time step.

## Results

Simulations were run for an hour, and the number of crimes committed at each site at each time step was summed up. The dynamic part  $B_k$  of the attraction term  $A_k$  was also summed up over each time step, and then scaled down by the number of seconds that the hour long simulation ran for (3600 seconds). These two results are shown in figures 2 and 3. The purpose of juxtaposing these two results was to investigate whether we can observe the crime hotspots documented in the UCLA and the TLF models. The correlation between a dynamic term that increases the attractiveness of a neighbourhood of sites following a crime, and the total crimes seems fairly trivial, and indeed, we do observe elevated values for the dynamic term at sites that experienced larger amounts of crimes during our simulation.

Finally, and this was a fairly tedious process, a static visual representation of a few seconds of these simulations was captured in figure 4. These *extended snapshots* were captured by making use of a series of dots with lighter colours to show the direction and rate of displacement of the various criminal agents. The darkest dot denotes its most recent position, and the faintest - its least recent one. You are also able to notice some the changes in directions. For instance,



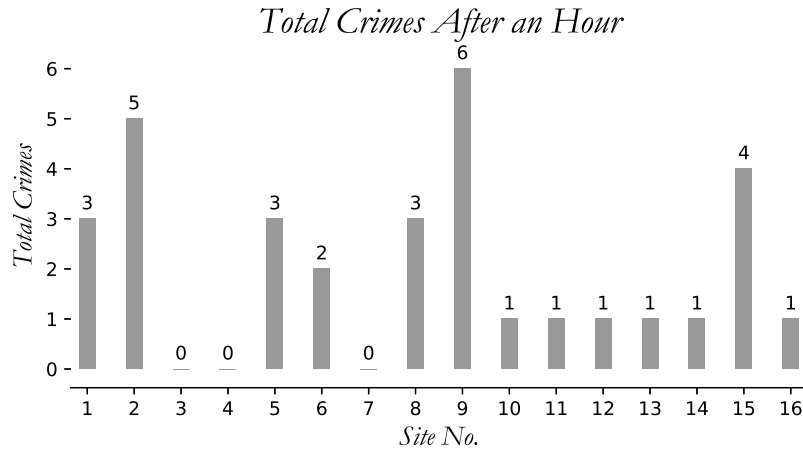


Figure 2: This is some mean values

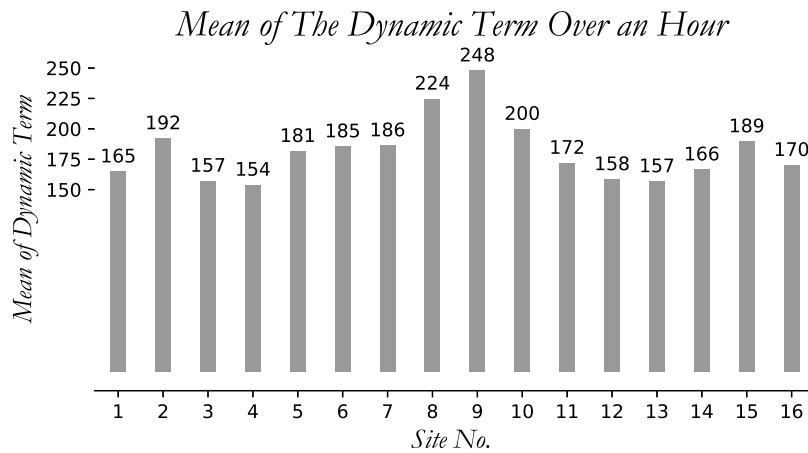


Figure 3: This is some mean values

the criminal between the sites at 0 and 20 and the criminal between the sites at 260 and 280. Each criminal is assigned their own *lane* and as criminals commit burglaries and are removed, new criminals fill in the empty lanes. We can thus also the number of filled lanes as a way to measure the number of criminals present in the simulation at any given time. Red *xs* were used to mark the locations of each of the crimes that were committed during the simulation. When a criminal commits a crime, a red *x* appears in their place which persists until the end of the simulation. The total number of crimes can thus be obtained from tallying all the red *xs*, and total crimes for each site can also be obtained by tallying the number of *xs* above each site. The simulation in figure4 was run over 30 minutes with 'screen shots' taken every 5 seconds, and starts with just one criminal at the second site (from the left). The static component of the attractiveness was

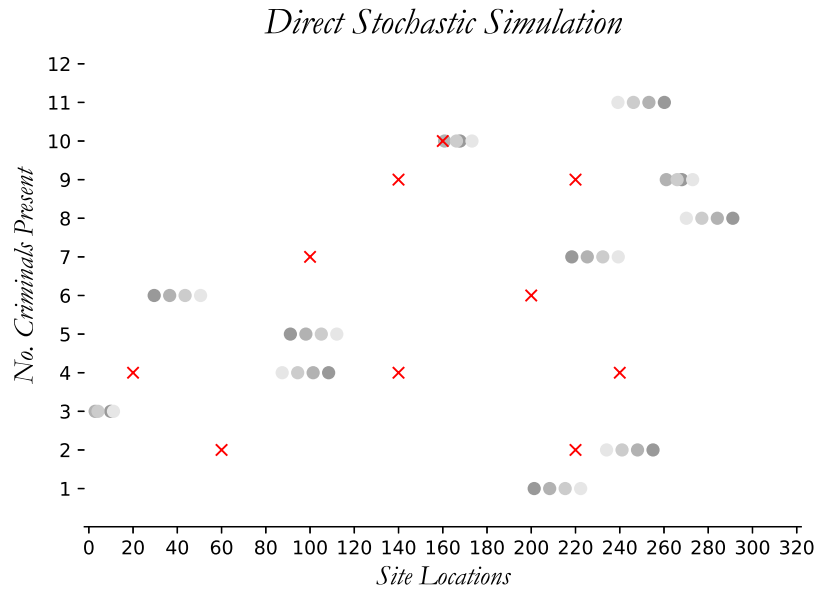


Figure 4: This is some mean values

initialised to:

```
A_0 = 17-16*np.cos(4*np.pi*x)
```

To view the time lapse of the 30 minute simulation, follow the link in the appendix. It is certainly a lot more successful at illustrating the haphazard sequence of short and long jumps that the criminals follow as they move from site to site.

### *Conclusion*

In this paper, an understanding of how we can use agent based statistical models to model opportunistic crime in residential areas was developed. Simulations of TLF model were successfully reproduced. A simple model based off of the TLF model that models individual criminal behaviour was used to run stochastic simulations. There are many extensions to this project that are worth pursuing. These include, but are not limited to:

- adapting the TLF model involving law enforcement agents and run direct simulations
- adapting the TLF model to a 2-dimensional grid as was done with the original UCLA model
- mapping the sites to an existing crime-ridden area and comparing how effective the model is in a real context

- giving individual criminals different values of  $\mu$  and  $L$  to represent the different ‘classes’ of criminals
- moving from static sites to dynamic people as targets for criminals to model crimes such as muggings

### *Acknowledgments*

To my supervisor, Prof. Nora Alexeeva, thank you for humouring my long periods of silence, and for your time and interest. To my laptop, I thank you for your long periods of silence and making me less reliant on local storage (and material possessions). To the cloud for providing the infrastructure that facilitated this pseudo-Buddhist transformation. To Kahla for letting me use her laptop when I actually did need one.

### *References*

- Sorathan Chaturapruek, Jonah Breslau, Daniel Yazdi, Theodore Kolokolnikov, and Scott G. McCalla. Crime modeling with lévy flights. *SIAM Journal on Applied Mathematics*, 73(4):1703–1720, 2013. DOI: 10.1137/120895408.
- Vincent Cruywagen. Chilling evidence of uyinene mrwetyana’s last moments and how she fought back, Sep 2019. URL <https://www.iol.co.za/capeargus/news/chilling-evidence-of-uyinene-mrwetyanas-last-moments-and-how-she-fought-back-31755766>.
- Jenna Etheridge. Clifton stabbing: ‘he had warned his sisters to be safe’ - slain uct student’s family, Sep 2019. URL <https://www.news24.com/SouthAfrica/News/clifton-stabbing-he-had-warned-his-sisters-to-be-safe-slain-uct-students-family-20190930>.
- James Q. Wilson George L. Kelling. Broken windows, March 1982. URL <https://www.theatlantic.com/magazine/archive/1982/03/broken-windows/304465/>.
- Travis E Oliphant. *A guide to NumPy*, volume 1. Trelgol Publishing USA, 2006.
- Chaohao Pan, Bo Li, Chuntian Wang, Yuqi Zhang, Nathan Geldner, Li Wang, and Andrea L. Bertozzi. Crime modeling with truncated lévy flights for residential burglary models. *Mathematical Models and Methods in Applied Sciences*, 28(09):1857–1880, May 2018. DOI: 10.1142/S0218202518400080.

Injong Rhee, Minsu Shin, Seongik Hong, Kyunghan Lee, Seong Joon Kim, and Song Chong. On the levy-walk nature of human mobility. *IEEE/ACM Transactions on Networking*, 19(3):630–643, 2011. DOI: 10.1109/tnet.2011.2120618.

M. B. Short, M. R. Dorsogna, V. B. Pasour, G. E. Tita, P. J. Brantingham, A. L. Bertozzi, and L. B. Chayes. A statistical model of criminal behavior. *Mathematical Models and Methods in Applied Sciences*, 18(supp01):1249–1267, 2008. DOI: 10.1142/S0218202508003029.

## Appendix

To watch one of the simulations, follow this link: <https://drive.google.com/file/d/1-4fU15ELtbMdwkCZUVxpSyTZd424XDN5/view?usp=sharing>

To read through the Notebooks associated with this project, follow these links:

Mean field simulation:

<https://colab.research.google.com/drive/1RDHddScr-SpQhaGgxdhox9vwiAnxwMXW>

Direct stochastic simulation:

<https://colab.research.google.com/drive/1oRA-9NC1WRcDwFMRLp76ldPoNPKb1ego>