

Disease management : Foot and mouth disease

Kamal QUAZBARY

March - August 2019

1 Introduction

Disease management has become an important field in public health since several decades. Outbreaks of fatal viral diseases has increased in different countries around the world. *Foot-and-mouth* disease (FMD) is a highly contagious famous one that has a lot of implications in animal farming. It can spread easily through contact with infected animals. It can affect different animal species such as cattle, water buffalo, sheep, goats, pigs, antelopes, deer, and bison. Humans are rarely infected.

The epidemic of FMD in the United Kingdom in 2001 resulted in 2000 cases of the disease in farms throughout the countryside, the county of Cumbria (North West of England) was the most seriously affected. To halt the disease, around ten million of animals were killed, which costs £8 billion to British agriculture. Numeric models to simulate the state of a World of animals infected by FMD have been implemented in different research papers. The Stefan Sellman's article "An optimized gridding approach for spatially explicit disease simulations" a standard SEIR (Susceptible - Exposed - Infectious - Removed) model with a gridding approach that reduces computational time.

2 Method

Farms, cells, World

The project simulates the evolution of a *World* of animals and the spread of a disease in this World at each time step (one day in our simulation). Animals are distributed in *farms* : we consider that each farm is an object determined with its *position* in the world (x, y), the number of the animals it contains.

An animal can be *susceptible* (not affected by the disease) or *infectious* (affected by the disease). As the animal becomes infected, we have to wait 9 days before we can report it. The animals becomes then *reported infectious*.

We consider in the simulation that all animals of the same farm are either susceptibles or infectious. That's why we have another farm's parameter : the *status* of a farm. We model then the status of a farm by an integer :

- 0 for a susceptible farm (a farm with susceptible animals)

- 1 for an infectious farm (a farm with infectious animals). The status of an infectious farm is then incremented at each time step. When a farm has a status greater or equal to 10, it becomes reported infectious.

Transmissibility model of the disease

The disease spreads among the animals each day depending on a disease model. In our simulation, each farm i has a *transmissibility* T_i and a *susceptibility* S_i . These parameters depend on the number of animals in the farm N_i : the more a farm contains animals, the more its ability to transmit the disease or to be infected :

$$\begin{aligned} T_i &= \Phi N_i^\tau, \\ S_i &= \psi N_i^\sigma \end{aligned}$$

where Φ , τ , ψ and σ are constants.

For an infectious farm i and a susceptible farm j , the probability of infection of j after one day is :

$$p_{ij} = 1 - \exp^{-S_j T_i K(d_{ij})},$$

where α , β and γ are constants, d_{ij} is the distance between the two farms and K a function given by :

$$K(d_{ij}) = \frac{\alpha}{1 + (\frac{d_{ij}}{\beta})^\gamma}$$

Iteration's algorithms

At each time step, we increment the status of all infected animals. We spread then the disease among the farms using a *pairwise* algorithm.

Pseudocode : Pairwise algorithm. p_{ij} is the probability of infection as defined between two farms i and j :

```
for each infectious node i
for each susceptible node j
R = uniform random number (0, 1)
if R ≤ pij, j is infected
```

Sellman's article presents a new gridding approach which is the *Conditional Entry Algorithm*. Farm's world is divided into cells and we calculate an over-estimated probability of infection. It is especially useful for a great number of farms. As we focus on a small part of the World, we will use pairwise algorithm in our simulation.

3 Simulation

Dataset and initial world

We use initially a World obtained from a dataset (high clustering random world - Sellman's paper). The data represents . For computational reasons, we focus only on a part of the world (farms with a position (x,y) satisfying $650000 \leq x \leq 660000$ and $750000 \leq y \leq 760000$) to build a set of 384 farms. We choose randomly one farm from the 384 farms to be infectious (status=1). Disease management begins after the first reported infectious, in the tenth day.

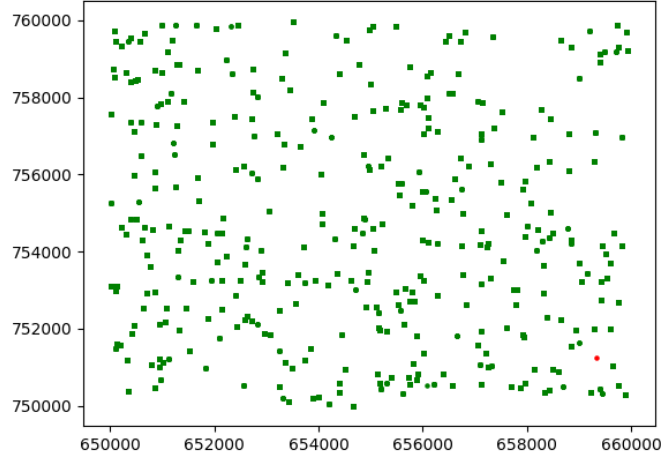


Figure 1: Initial state : each green point is a susceptible farm, the red point is the infectious farm

Parameters used

For the simulation, we fix the parameters as used in Sellman's article :

Parameter	Value
Φ	0.00083
τ	0.49
ψ	1
σ	0.2
α	1
β	1600
γ	4.6

Management and daily budget

We have a *daily budget*, which is a fixed number of farms that we can kill per day. The aim is to limit the spread of the disease among the world's farms, by killing as few farms as possible. We choose a budget of 3 farms per day in our simulation.

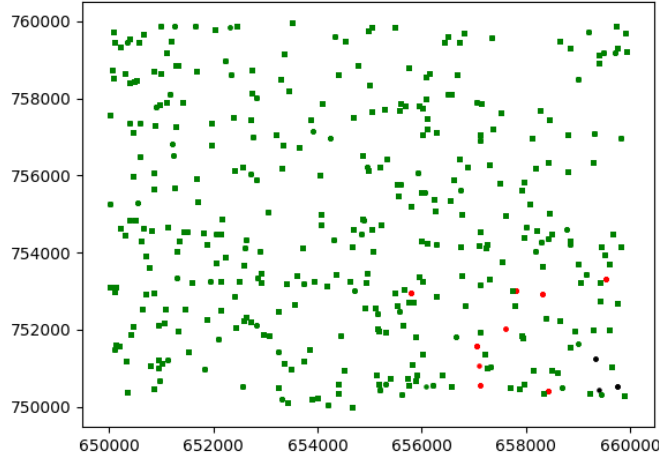


Figure 2: After 10 days : the disease has spread among several farms, the management begins

At each time step, we have to choose which farm will be killed by management. To this purpose, we have different strategies :

Strategy 1 : We firstly kill all the farms that are reported infectious. If there is left budget, for each susceptible farm, we calculate the probability that this farm becomes infected at next time step. We then sort susceptible farms by this probability and we kill the most probable susceptible farms until we exhaust the daily budget.

The probability of infection is calculated using the formula :

$$P = 1 - \prod_i (1 - p(i))$$

where i is a reported infectious farm, and $p(i)$ the probability that this farm affects the initial farm (as defined above in Method).

Strategy 2 : We rank susceptible farms upon their propensity to inflict further damage to other farms. To this purpose, we consider all susceptibles that are close to an infectious farm (we choose a radius of 100). For each susceptible, we simulate the iteration of the World during a fixed number of time steps with and without it, and then we calculate the number of infectious in each case. We kill then the susceptibles according to this difference until we exhaust the budget.

Strategy 3 : Firstly, we kill all the farms that are reported infectious. If there is left budget, we implement a ring cull of fixed diameter around the last farm reported infectious already killed, until we exhaust the budget. It's our control strategy.

In all strategies, the simulation lasts until there are no longer infectious animals in the world (reported or non reported).

Implementation and results

We run 100 simulations of the same World (extracted from the data set above). We calculate then after each day the number of farms killed by our management using strategy 1.

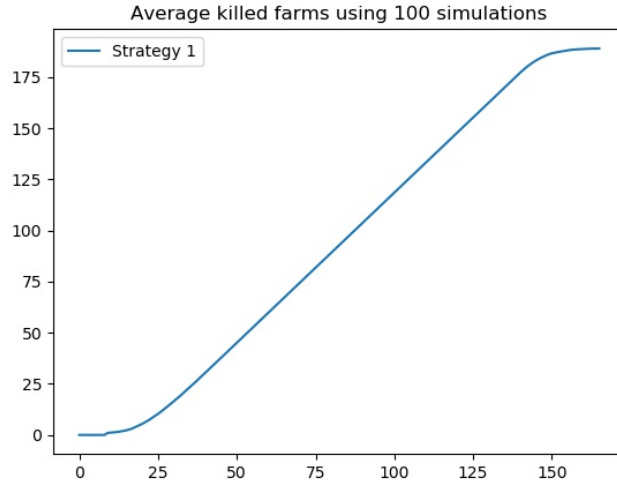


Figure 3: Results of 100 simulations of the same initial world of 384 farms

We remark that we kill an average of 190 animals from 384 (50 % of farms saved) with strategy 1 (Figure 3). The disease disappears after 150 days approximately. During the 10 first days, we can see that we don't kill any farm : all the infectious animals are non reported. After the first reported farm, we kill between 1 and 3 farms daily. When the disease spreads enough (30 days), we kill 3 farms per day.

Using strategy 3, the average number of killed animals is 280 from 384 (27 % of farms saved) (Figure 4). The disease disappears also after 150 days. We kill around 100 farms more than in strategy 1.

Effect of the parameter α

We modify the value of the parameter α (used in the disease model) from 1 to 10, and simulate the average number of farms killed (Figure 5). The disease spreads more in this situation, because the probability of infection increases. We then kill all the farms in the two situations. Here, we can't save any farm using both strategies.

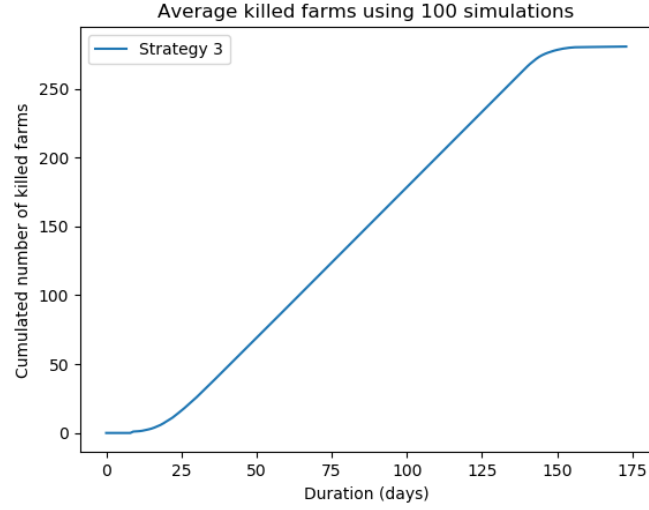


Figure 4: Results of 100 simulations of the same initial world of 384 farms

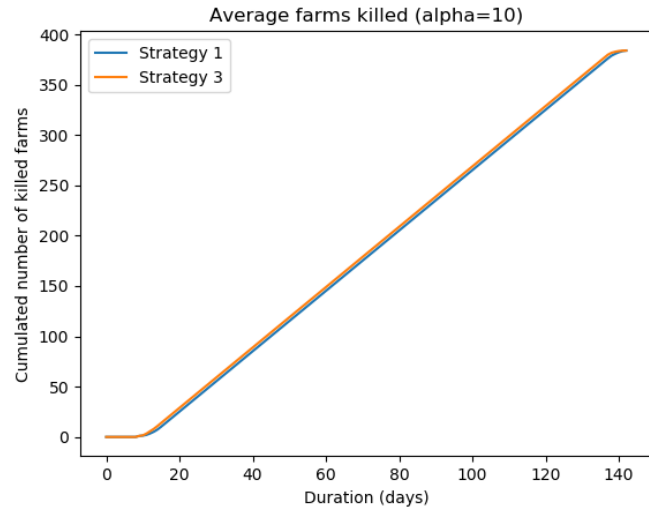


Figure 5: With $\alpha=10$, the disease spreads more easily. We kill all the animals in the 2 strategies

Effect of the number of animals

We keep the same data set World, and we modify the number of animals by multiplying the initial number by a parameter δ .

With a parameter $\delta=2$, we kill more animals (384 instead of 280). It means that with a high number of animals in each farm, the disease spreads more easily among from one farm to another, which is an expected result.

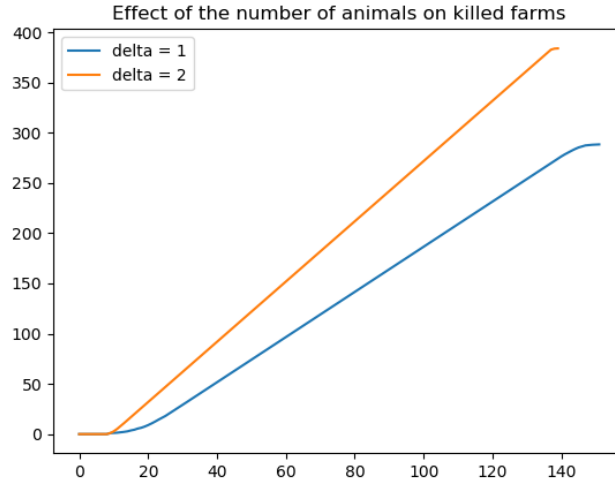


Figure 6: Orange curve is a simulation of farms with more animals than blue curve

Strategy 2 : implementation

In strategy 2, we focus more on the propensity of a farm to inflict further damage (by causing more infections) rather than the risk of becoming infected.

The idea is to determine at each iteration what are the susceptible farms whose presence causes a lot of damage if the next time steps and kill it. For this purpose, we introduce a function *Simulate World* :

Pseudo code of Simulate World :

For each susceptible farm that is close to an infectious (radius=100) :

W_1 = copy of initial world W

W_2 = copy of initial world W from which we remove the susceptible

W_1 = Iteration (W_1) for T time-steps (T is usually 5)

W_2 = Iteration (W_2) for T time-steps

return Killed Farms (W_2) - Killed Farms (W_1)

Once we have this result for each susceptible, we rank all susceptibles and kill the most "dangerous" ones using the initial World W until we exhaust the daily budget.

It is important to work on the copies W_1 and W_2 in order not to modify the real world W.

Strategy 2 : work done

During the different attempts to simulate the World, we got 0 as a result of the *Simulate World* function with different susceptible farms. We built a small World of 10 animals (Figure 7) : in this world, we have an infections farm, a susceptible farm close to it (the 'dangerous' one) and some other susceptibles.

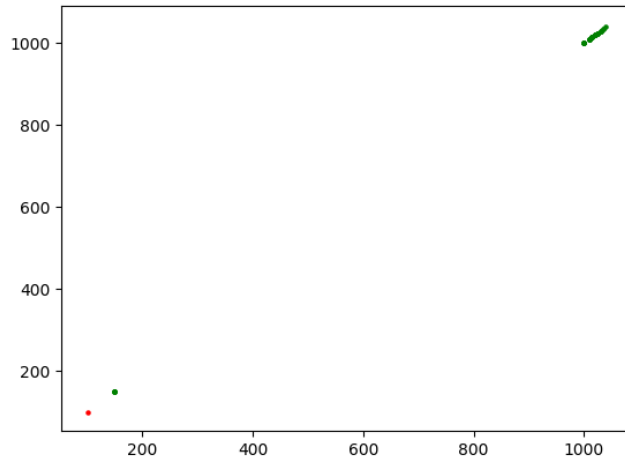


Figure 7: 10 farms world to test the Simualte World function

4 Conclusion

It seems that Strategy 1 gives a satisfying result : almost 50% of farms are saved in the simulation. To go further, we can imagine some other ways to apply the strategy 2, and to explain it didn't work in the different cases.

Another idea will be to hold the daily budget at the beginning when the disease is under control, and use it afterwards.

In the case of huge worlds, the next step will be to implement the Conditional Entry Algorithm, which is better for the computation.

5 References

Stefan Sellman. *Need for speed: An optimized gridding approach for spatially explicit disease simulations*, 2018