# How Graph Theory Can Help Control Cattle Diseases

JESS ENRIGHT

I think that everyone can benefit from a little graph theory — in this article I argue that this includes cows. I'll describe several ways that a British cattle dataset can be interpreted as a graph, and outline some algorithmic work on optimally modifying these graphs to limit worst-case disease outbreaks.

As anyone who has spent time on a rail journey through the British countryside will know, there are a lot of sheep and cattle in Britain. A large amount of information is recorded about the management of these animals, including careful recording of their movements between farms and markets. I first encountered cattle trading datasets in 2011, when I started a postdoctoral appointment at the University of Glasgow as part of a quantitative epidemiology group, and since then I have joined many other researchers in thinking about how considering British cattle movements as a network or graph can help us detect and control disease.

While cattle movement datasets in Britain may not be big in the "big data" sense, they're also not small. At the most recent count, there were about 9.8 million cattle in the United Kingdom[1], held on over 75,000 agricultural holdings. Movements of these cattle between agricultural holdings are recorded and reported to British government, and collected into a central database. This extensive dataset is invaluable for modelling infectious diseases of cattle, as well as monitoring the operation of the industry as a whole. Because of the established importance of these animal movements in major outbreaks of foot-and-mouth disease, and as a risk factor for bovine tuberculosis, there has been a very large amount of modelling, statistical, and simulation work using the cattle movement database, including work from a network science perspective.

In this article, I'll outline some of the work that has been done on cattle movements in Britain, with a focus on how graph theory can contribute. First, I talk about the cattle movement dataset and several different ways of representing it as a graph, and spend some time on how we can incorporate temporal information into this graph. Then I'll move on to a discussion of how the properties of these graphs have let us make progress on optimal graph modification problems for limiting worst-case disease.

### From movements to graphs

The British Cattle Movement Service dataset contains movement records for individual animals, including the date of the movement, as well as births and deaths. It is one of my favourite datasets because it is large, detailed, has been collected for more than a decade, and is generally considered to be reliable.

When making a graph out of the cattle movement dataset, the first thing to decide is what entities will be vertices, and what contacts will be recorded as edges. The most common approach is to take the set of farms as the vertex set, with an edge between two farms if there has been animal trade between them over some time period of interest, but this is far from the only option. Sometimes a larger aggregation is more appropriate: for example, if we're looking at regional trade perhaps the counties of Britain should form the vertex set. Or, perhaps we need closer granularity, and take the set of individual bovines as the vertex set — of course, this may result in a larger graph than we are prepared to deal with, as at any given time there are approximately 9 million cattle living in Britain!

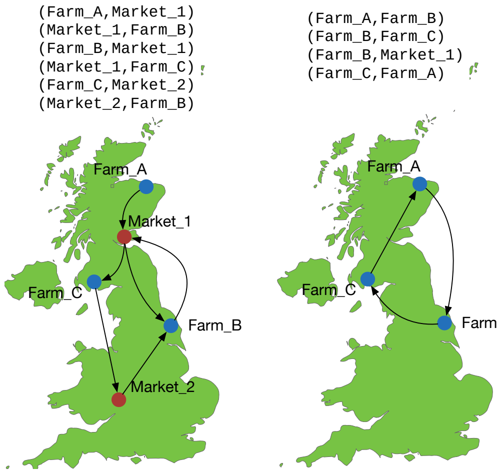
For the moment, let's say we take the set of agricultural holdings (including farms, markets, etc.) as our vertex set. What about edges? As I mentioned, a common choice is to link two agricultural holdings with an undirected edge if an animal has moved between them within some time window. This choice throws away a lot of potentially useful information, including the direction, weight, and timing of a contact. As methods and and available software have developed, these extra pieces of information are being more frequently used, and the cattle trading graph might be considered as a directed (as in the bottom left of Figure 1), weighted, or temporal graph. We'll return to the idea of a temporal graph later in this article.

[1] https://www.gov.uk/government/statistics/farming-statistics-livestock-populations-at-1-december-2016-uk

The majority of cattle trades in Britain are conducted through a relatively small number of auction markets. This means that in the graph in which both farms and markets occur in the vertex set, there are a relatively small number of vertices (the markets) with very high degree, and many vertices of comparatively low degree. More than that, the structure of the graph is relatively simple: the majority of edges form a hub-and-spoke-like graph. This sort of structure can make many computationally difficult problems much easier to solve, but may not be appropriate for every disease setting.

A slow-spreading disease that requires close contact to spread between animals is unlikely to spread at a market, and so sometimes it may be more appropriate to consider a *market-stripped* version of this graph in which we remove the markets and record a movement of an animal from farm $u$ to farm $v$ via a market as an edge from $u$ to $v$ (bottom right of Figure 1).

```
Animal1,2018/01/01,Farm_A,Market_1
Animal1,2018/01/01,Market_1,Farm_B
Animal2,2018/01/01,Farm_A,Market_1
Animal2,2018/01/01,Market_1,Farm_B
Animal1,2018/02/01,Farm_B,Market_1
Animal1,2018/02/01,Market_1,Farm_C
Animal3,2018/01/15,Farm_C,Market_2
Animal3,2018/01/15,Market_2,Farm_B
```



```
(Farm_A,Market_1)
(Market_1,Farm_B)
(Farm_B,Market_1)
(Market_1,Farm_C)
(Farm_C,Market_2)
(Market_2,Farm_B)
```

```
(Farm_A,Farm_B)
(Farm_B,Farm_C)
(Farm_B,Market_1)
(Farm_C,Farm_A)
```

Figure 1. Examples of graphs derived from cattle movement records: at the top, a section of individual animal movements, showing the animal ID, date of movement, and source and destination locations. On the bottom left, a directed graph showing an edge from $u$ to $v$ if there is at least one animal movement from $u$ to $v$ in the set of movements. On the bottom right, a market-stripped graph, in which an animal's movement from farm $u$ to market $m$ to farm $v$ on a single day is recorded as an edge from $u$ to $v$. Edge sets for the two graphs are shown at the top of the maps.

### *Temporal graphs*

Leaving out the temporal information when creating a graph from cattle movements is a serious omission that can give us a misleading picture of transmission pathways of disease. It is no surprise that the timing of cattle trades is important in modelling disease. Consider the simple case of a path of three vertices, with different orderings of the edges.

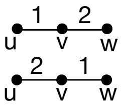


Figure 2. The same graph with two different assignments of times to edges. On the top, there is a temporally admissible path from $u$ to $w$, on the bottom there is not.

If the edge from $u$ to $v$ occurs before the edge from $v$ to $w$, then it's possible for an infection at $u$ to spread to $w$, and we say the path from $u$ to $v$ to $w$ is *temporally admissible*, whereas if the edges occur in the other order, then it is not. In a time-ignoring snapshot of more than a week of movements from the GB cattle trading dataset, approximately half of the static paths of three vertices and one fifth of the paths of four vertices are temporally feasible when timing is taken into account.

The detailed timing information in network datasets like the cattle movement dataset has partially motivated the increasing popularity of temporal or dynamic graphs both in algorithmic graph theory and in network science. While there are a variety of formalisms for temporal graphs, my favourite is one that fits well with the cattle movement data: a temporal graph is a pair $(G = (V, E), \Delta)$, where $G = (V, E)$ is a graph and $\Delta$ is a function from $E$ to sets of timesteps, with $\Delta(e)$ denoting the timesteps at which $e \in E$ occurs. We can include directions or weights in our temporal graph by allowing $G$ to be directed or weighted.

Because there are far more tools available for dealing with static graphs, there have been several attempts to capture the dynamic nature of trades in static graphs. Vernon and Keeling [9] describe a variety of static graphs derived from the cattle movements with the intention of capturing the dynamic nature of the graph. In common with many others, their primary tool is effectively a sequence of graphs, with one defined at each appropriate time step (in the case of the cattle movements, usually this timestep is a day).

Kim and Anderson [8] use a now-popular method that creates a static directed network in which the vertex set is formed of multiple copies of vertices from the temporal graph, with a vertex duplicated at each appropriate timestep, and edges going forward in time.

Heath et al. [7] use a line graph-like method that incorporates explicit information about an infectious period for a particular infectious disease of interest to produce a static graph that captures some of the key dynamic information: essentially, they create a graph in which the vertices are pairs of trades and days at which those trades occurred, and there is an edge from one trade/time pair to another if disease that spreads over the first could subsequently move over the second.

Formally, given a directed temporal graph $(G = (V, E), \Delta)$ and an infectious period $\delta_I$, they produce a new directed graph $H = (P, F)$, in which the vertex set $P$ is composed of $(e, t)$ where $e \in E$ and $t \in \Delta(e)$, and the edge set $F$ contains an edge $((e, t_e) \rightarrow (f, t_f))$ where $e = (u \rightarrow v)$ and $f = (v \rightarrow w)$ if and only if $t_e < \mathcal{T}(f)$ and $t_f - t_e \leq \delta_I$. This static representation captures the directionality both of the original graph, and of time.

### *Seasonal trends*

Great Britain has (somewhat) distinct seasons, and so, as you might expect, there are seasonal patterns in the cattle movement network. We can use a limited, but easy-to-calculate measure of changes in the network over time: to compare two network snapshots at different times, calculate the proportion of edges in the graph that exist at both time-steps. If we consider month-long snapshots of the cattle trading graph, then we see a scallop-edged plot in Figure 3. For any given month, the most similar months are multiples of 12 months previous, and the least similar are those in opposite seasons. For example, April of this year is most similar to April in previous years, and most dissimilar to November. In addition to the seasonal effect, we see a gradual decrease in similarity with an increase in temporal distance — the farther in the past we look, the more dissimilar the graph is. As you might expect, we see higher similarity scores at higher levels of aggregation: when we consider only county-to-county level edges, we see much more similarity than at the farm-to-farm level of detail.

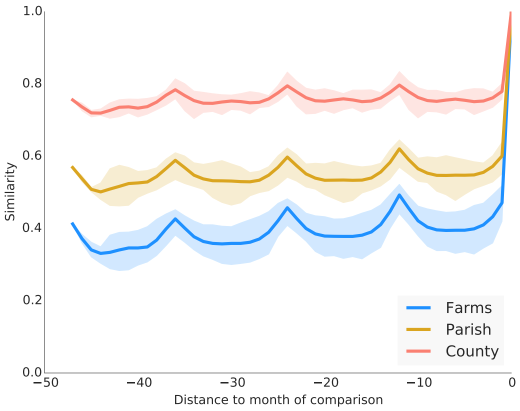


Figure 3. From [1]: A plot of the mean similarity (calculated using a method of set similarity counting equivalent to the Jacquard similarity) of cattle movements in Great Britain between each month of 2011 and months up to four years previous. The blue line shows similarity of unaggregated movements, the yellow movements to parishes, and the pink movements to counties. The background shaded envelopes are the minimal areas that include the similarity plots of each of the individual months in 2011 (the solid line is the mean over all 12 months).

## Modification of livestock movement graphs

A significant volume of work has shown that the strategic removal of important vertices or edges (by a variety of measures) is far more effective in decreasing simulated disease outbreak size than random interventions — this is pleasingly logical. For example, it is not surprising that in a network with a relatively small number of high-degree vertices (often dealers or markets in the cattle network), removing those vertices will limit disease spread on the network more effectively than removing random farms. Gates and Woolhouse [6] provide evidence of this effect for edges, using a simple edge centrality measure, and use this observation to motivate a heuristic rewiring method to modify the cattle trading network with the aim of limiting disease spread. Roughly speaking, Gates and Woolhouse select edges with a high potential to spread disease across the network, and rewire them using a heuristic matching process that preserves in- and out-degree. Their approach gives networks with lower simulated endemic disease prevalence than the baseline real-data networks.

Given that removing edges and vertices strategically can make a large difference in expected disease prevalence, we might address the idea in a more formal algorithmic sense and ask: given a graph and a budget for edge or vertex removal, what is the best possible choice? As a first attempt at answering this question, Kitty Meeks and I [3] focussed on removing edges to limit the maximum connected component size in a graph, as maximum component size is an upper bound on the largest possible outbreak size.

In general, this problem is NP-complete, so we expect that it cannot be solved efficiently on general graphs. However, if we restrict the graphs we are interested in to a limited class, we can make progress toward an efficient algorithm (by which we mean one that takes computational time asymptotically bounded by a polynomial function of the size of the input).

We focussed our attention on graphs of limited treewidth, which are graphs admitting a certain type of decomposition (see “Treewidth”) for two reasons: first, the tree decompositions of graphs with limited treewidth have a strong history of supporting efficient algorithms, particularly via several key metatheorems that promise the existence of polynomial-time algorithms for problems that can be encoded in specified logical frameworks, and secondly (and possibly more remarkably), a static aggregation of the cattle trading graph in Scotland has relatively low treewidth.

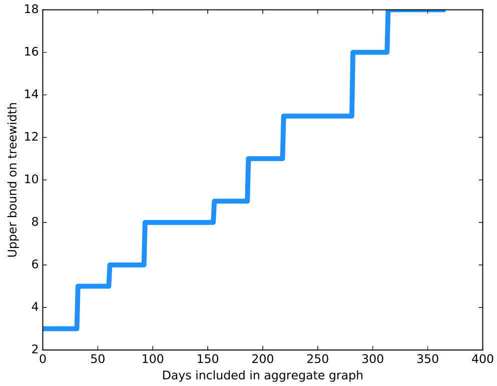


Figure 4. From [3]: a plot of treewidth of the static market-stripped graph derived from cattle trades in Scotland over increasing portions of 2013.

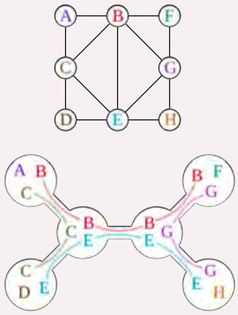
We created a series of aggregations of the Scottish cattle movement graph with markets stripped over the year 2013. For each day in 2013, we create a graph of all movements (ignoring direction and timing) from the beginning of the year to that day, and calculated and plotted an upper bound on the treewidth of that graph. Because the graph grows over the year, we expect the treewidth to be non-decreasing, and this is what we see in Figure 4. The entire year of movements has a treewidth of at most 18, and the first six months a treewidth under 10. The running time of a tree-decomposition-based algorithm is typically exponential in the treewidth of the graph — a running time that is exponential in a treewidth of 10 is perhaps not ideal, but is generally feasible, as we discovered in a number of related computational experiments [3].

## Treewidth

A *tree decomposition* is an assignment of all vertices of a graph $G = (V, E)$ to bags at nodes of a tree that must follow two rules:

- for every vertex $v \in V$, the set of nodes of the tree that are assigned $v$ must induce a connected subtree of the tree, and
- every pair of vertices $u, v$ that are adjacent in $G$ must co-occur in at least one bag on the tree

The width of such a decomposition is one less than the largest number of vertices assigned to a single bag. The *treewidth* of a graph is the smallest width of a tree decomposition over all possible tree decompositions of that graph. Here is an example of a valid tree decomposition, reproduced from the wikimedia commons [5].



Note that each subtree corresponding to nodes containing a single vertex is connected, and every pair of vertices that are adjacent occur together at a node at least once. This decomposition has width 2.

### *Incorporating temporal information — Reachability sets*

Where the maximum component size is an upper bound on the size of the largest outbreak on a static and undirected graph, the maximum reachability set size serves the same role in a temporal graph, either directed or undirected.

If $(G = (V, E), \Delta)$ is a temporal graph, then we say vertex $v \in V$ is *reachable* from vertex $u \in V$ if there is a path of edges from $u$ to $v$ such that every edge occurs temporally after the one before it in the path: that is, the path goes forward in time. The *reachability set* of a vertex $v$ is the set of vertices that are reachable from $v$, including $v$ itself.

Recently, we have been investigating several approaches to modifying temporal graphs to limit the size of the maximum reachability set, including edge deletion (as we did to limit component size in static graphs), and assigning or re-ordering the times of edges (with George Mertzios and Viktor Zamaraev). This is very much work in progress, with current results reported in [2, 4].

### Concluding thoughts

I have only talked about cattle movements here, but there are a wide variety of agricultural datasets that I've enjoyed viewing as graphs. I believe that graph theory has a lot to contribute to the analysis and modelling of the processes and systems that produce these datasets.

When we were trying to find small edge deletions to limit component sizes in the cattle trading graphs in Scotland, we were lucky that the graphs had low treewidth, which we were able to exploit to devise and implement an algorithm. In the future, several of my collaborators and I are planning to investigate why these graphs have low treewidth, and what other properties and classes arise in data-derived graphs that we can exploit for the design of efficient algorithms. There are some obvious candidates — for example, graphs derived from geographical proximity are often planar, or may be well-described with random geometric models.

There is a lot of work to be done on optimisation problems on these agriculturally-derived graphs, particularly on graphs that incorporate direction or temporal information. While there has been a recent boom of activity in the algorithmics of temporal graphs, there is still an enormous gulf between the large collection of algorithmic machinery available for static graphs, and that available for temporal graphs, even when the temporal graph is fully specified. Data-driven problems like those I've encountered working with cattle movements can benefit significantly from these recent and coming advances, as well as jump-starting work on theoretically appealing algorithmic questions — I'm looking forward to being inspired by cows for years to come!

### FURTHER READING

[1] J. Enright and R.R. Kao, Epidemics on dynamic networks, Epidemics, in press.

[2] J. Enright and K. Meeks, Changing times to optimise reachability in temporal graphs, CoRR abs/1802.05905 (2018).

[3] J. Enright and K. Meeks, Deleting edges to restrict the size of an epidemic: A new application for treewidth, Algorithmica 80 (2018) 1857–1889.

[4] J. Enright, et al. Deleting edges to restrict the size of an epidemic in temporal networks, CoRR abs/1805.06836 (2018).

[5] D. Epstein, A graph and its tree decomposition, 2007.

[6] M.C. Gates and M.E.J. Woolhouse, Controlling infectious disease through the targeted manipulation of contact network structure, Epidemics 12 (2015), 11–19, Papers arising from Epidemics 4.

[7] M.F. Heath, M. Vernon, and C. Webb, Construction of networks with intrinsic temporal structure from UK cattle movement data, BMC Veterinary Research 4 (2008) 11.

[8] H. Kim and R. Anderson, Temporal node centrality in complex networks, Physical Review E 85 (2012) 026107+.

[9] M.C. Vernon and M.J. Keeling, Representing the UK's cattle herd as static and dynamic networks, Proceedings of the Royal Society B: Biological Sciences 276 (2009) 469–476.

### Jess Enright

Jess Enright is a lecturer in Computational Methods and Agrifood Systems at the University of Edinburgh. Her main research interests are in graph theory and network epidemiology, but she's also a big fan of combinatorial games. She was very surprised when she found herself working with cattle and sheep data, and has recently branched to fish. She wrote the first draft of this article on a delayed train.