A flexible load sharing system and implementation to anticipate and organise transfers based on ICU demand in the context of COVID-19 pandemic

Lucas Lacasa[1]; Robert Challen[2,3]; Ellen Brooks-Pollock[4]; and Leon Danon[5,6]

1. School of Mathematical Sciences, Queen Mary University of London, London E1 4NS, UK
2. EPSRC Centre for Predictive Modelling in Healthcare, University of Exeter, Exeter, Devon, UK.
3. Taunton and Somerset NHS Foundation Trust, Taunton, Somerset, UK.
4. Bristol Medical School: Population Health Sciences, University of Bristol, Bristol, UK
5. Data Science Institute, College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter, UK.
6. The Alan Turing Institute, British Library, 96 Euston Rd, London NW1 2DB, UK.

Contact: l.lacasa@qmul.ac.uk

Abstract

As the number of cases of COVID-19 continues to grow exponentially, local health services are likely to be overwhelmed with patients requiring intensive care. We develop and implement an algorithm to provide optimal re-routing strategies to transfer patients requiring Intensive Care Units (ICU) between hospitals within NHS trusts, constrained by feasibility of transfer. We coarse-grain the NHS system at the level of NHS trusts and, subsequently cover the whole set of geopositioned trusts to extract a 4-regular geometric graph which indicates, for a given trust, its four nearest neighbors. Estimates of weekly ICU demand can be extrapolated from an age structured epidemiological model by considering contagion-to-ICU likelihood estimates, and through random search optimisation we identify the best load sharing strategy. The cost function to minimise is based on the total number of ICU units above capacity and we implement and test two optimisation strategies. Our framework is flexible allowing for additional criteria, different cost functions, and this methodology is general enough that it can easily be extended to optimise other resources beyond ICU units or ventilators. Assuming a uniform ICU demand across trusts, we show that using our method it is possible to enable access to ICU treatment to up to 1000 cases nationally in a single step of the algorithm – leading to potentially saving a large percentage of these lives that would otherwise not have access to ICU if no load sharing was implemented.

I- Background

The coronavirus disease COVID-19 [1], whose outbreak was detected in China in December 2019 [2], has become pandemic and as of March 2020 is putting national health systems of different countries into significant levels of stress [3-6] (see [7] and references therein for a fully detailed thread of reports including the effect of non-pharmaceutical interventions in a number of countries, severity analysis, symptom progression, etc, elaborated by the Imperial College COVID-19 Response Team). It is expected that the ICU demand of several hospitals across the UK will surpass their nominal capacity, as is already happening in Spain [8]. The shortage of sanitary resources is unlikely to be limited to ICU units or ventilators, and other resources will face similar challenges. In anticipation of these scenarios, here we design and implement a simple and flexible load sharing procedure which we hope can help to alleviate the level of stress of healthcare systems and implement and test with information for the UK National Health Service (NHS). As a proof of concept, we focus on the problem of ICU demand and propose a routine strategy to transfer patients across the network. Similar strategies can also be followed to transfer ventilators and other resources.

II- Methods

2.1 The trust network

We coarse-grain data at the level of trusts, as the main units of NHS organisation. We have $N = 141$ trusts across the UK, and each trust corresponds to a conglomerate of hospitals. For each trust we provide a concrete geoposition in terms of the centroid of the convex polygon whose vertices are the hospitals belonging to that trust. While spatial coordinates are given in terms of latitude and longitude, we make a small angle approximation and accordingly interpret latitude and longitude as cartesian coordinates. In particular, under this approximation the centroid coordinates of trust i reduces to the arithmetic mean of the coordinates of each hospital in the trust

$$(x,y)_i = (\frac{1}{m}\sum_{j=1}^{m} lat(j),\ \frac{1}{m}\sum_{j=1}^{m} lon(j))_i\ ,$$

and the distance between two trusts corresponds to the Euclidean distance

$$d_{ij} = \|(x,y)_i\ -\ (x,y)_j\|_2 = \sqrt{(x_i - x_j)^2\ + (y_i - y_j)^2}$$

Once we have geopositioned each of the 141 NHS trusts, we assign a vertex to this spatial location and proceed to tessellate this set. Accordingly, we build a regular geometric graph with degree $k = 4$, where each vertex i is connected to the four closest vertices according to d_{ij} displayed above. The resulting graph models the NHS trust network, and each trust will only be allowed to transfer patients to the trusts in their topological neighborhood.

2.2 Local load sharing model

The basic architecture of the *local* load sharing model is depicted in Figure 1. For each NHS trust, the algorithm takes `projected_ICU_demand` data aggregated at the NHS trust level, matches with its `baseline_ICU_capacity` (number of ICU beds aggregated for all the hospitals belonging to that trust), and generates a `local_stress` value for each trust. For those trusts where such stress is positive (meaning that there is demand that surpasses the capacity), the algorithm explores which neighboring trusts (extracted from the topological neighborhood of the trust under analysis) can accept a transfer.

If in the topological neighborhood of a given trust more than one receptor trust is available (i.e. has a negative `local_stress` and the distance between origin and destination is smaller than a certain maximally allowed transfer distance `d_max`), then the algorithm selects at random the receptor trust. Finally, a solidary load is shared to the receptor. This load is either 50% of the available capacity of the receptor trust, or the total excess demand of the origin trust, whichever is smaller.

Importantly, note that the algorithm can implement such analysis either in a **sequential** or **parallel** way. In the first case, the `projected_ICU_demand` of each trust is sequentially updated after each local load share is performed. This has the positive implication that no receptor can be overwhelmed from the simultaneous load sharing of different trusts. If the update is parallel, overwhelming receptor trusts can happen, but it is also true that more optimal redistributions are available.

The implementation code asks the user in which processing mode (sequential or parallel) the algorithm is run.

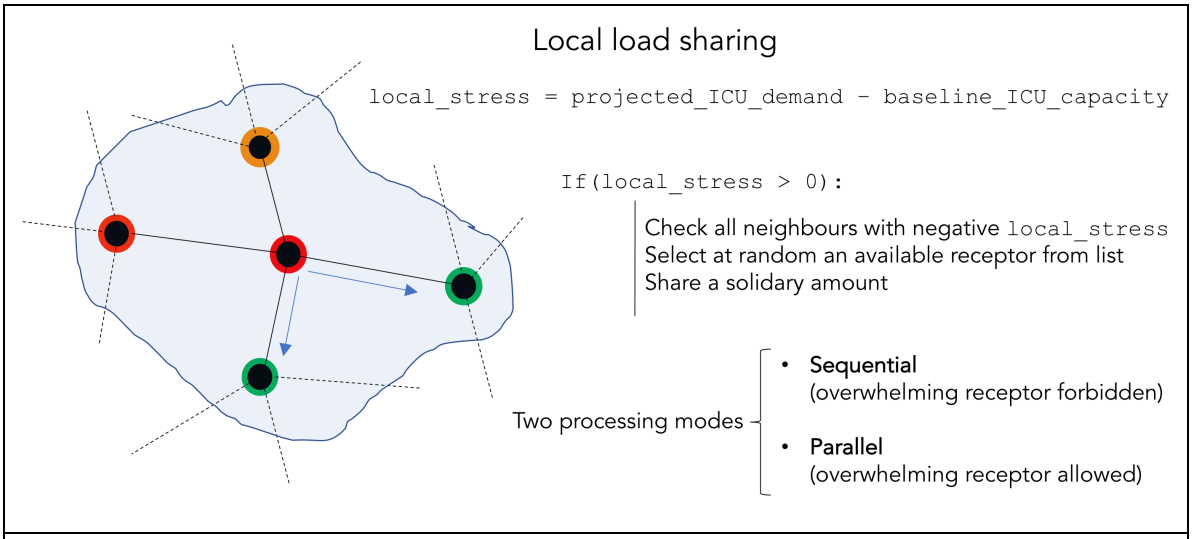

Figure 1: Scheme of the local load sharing model. Red and orange denotes an overwhelmed unit with varying levels of stress, green denotes a unit with capacity.

2.3 Random Search Optimisation

The basic local load sharing model is run for all trusts, and as a result a possible load sharing configuration is extracted, consisting in the specified origin and destination of all the packets of ICU units shared:

`Trust i shared x loads to trust j`

To assess the global impact of such load sharing configuration, we define the `global_stress` as

$$global\_stress = \sum_j \theta(local\_stress(j)),$$

where the sum runs over all trusts and $\theta(x) = x$ if $x > 0$ and $\theta(x) = 0$ otherwise. So essentially `global_stress` counts the total demand of ICU units in excess of capacity, in all those trusts which are projected to be overwhelmed.

Finally, the algorithm runs a total of 10^5 different realisations, and only keeps the run with smallest `global_stress.` By doing that, in each configuration the load sharing stochastically chooses a number of actions, and by randomly sampling the search space and keeping the configuration that minimises `global_stress,` the algorithm is globally optimising the load sharing configuration.

2.4 Input variables

Now we discuss the various input data required to run the local load sharing model:

`projected_ICU_demand`: This is an input data to the algorithm and could be estimated following a complex multi-step flow [9], which can be summarised as follows:

1. The projected number of new infections next week: This quantity can be informed in the first place from an epidemiological model [10] which provides predicted numbers of contagion at different spatial resolutions. Alternatively, or in the absence of such a model, it could be estimated from various sources of data, such as prescription data [11] or through direct questionnaires [12]. A post-processing of these numbers is then carried out, taking into account (i) age demographics and (ii) associated infection-to-ICU composed likelihood.
2. The projected number of patients already in the hospital which progress to ICU by next week: this number is estimated from real data of hospital admissions and average admission-to-UCI likelihood.
3. The projected number of patients already in ICU this week which will still require ICU next week: this number takes into account both the fatality ratio and the estimated discharge time.

As a proof of concept, in this work we assume different types of artificial ICU demands (uniform and heterogeneous distributions). We will test how the load sharing algorithm works under different demands.

`baseline_ICU_capacity:` This list is extracted from public available databases [13]. Note that this does not take into account surge capacity, that is expected to significantly increase the real ICU capacity of each trust.

III - Results

3.1 Single-share

In this first section we assume that each trust can only submit a unique load to a unique receptor trust, to be selected randomly from the trust's topological neighborhood.

3.1.1 Stress test with fixed, uniform-load ICU demand

As an illustration, we first analyse a stress test case where `projected_ICU_demand` is artificially set to a uniform value of 20 ICU beds per trust (i.e. all trusts receive a demand of 20 beds) whereas we set all `baseline_ICU_capacity` to its real value, and `d_max` $= \infty$. The histogram of `baseline_ICU_capacity` is reported in the left panel of Figure 2, whereas the histogram of `local_stress`, before and after the load sharing procedure is applied, is depicted in the right panel of the same figure (only the parallel mode is showcased). The procedure is capable of reducing the global stress of the system from an initial value of `global_stress = 611` ICU beds in excess in overwhelmed trusts, to a final value of `global_stress = 101` after the optimal load sharing is performed, i.e. a transfer and subsequent clearance of 510 ICU patients.

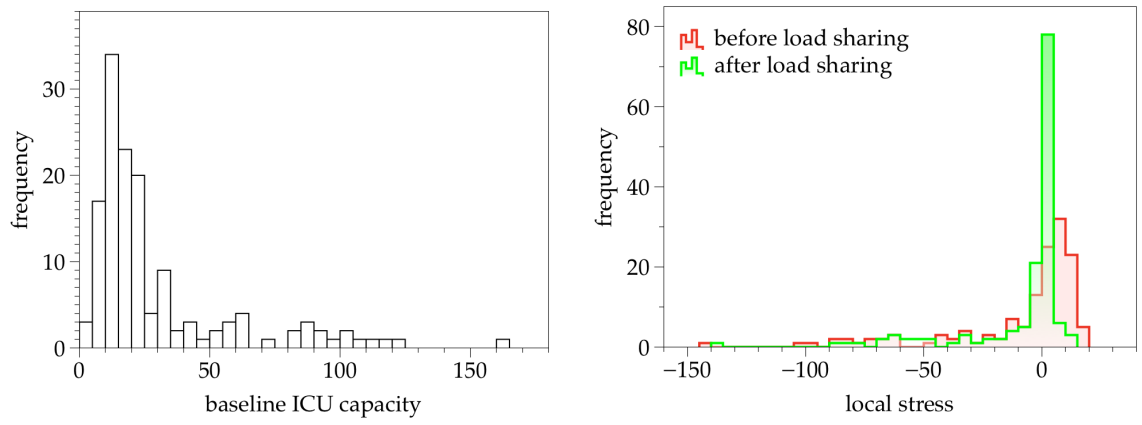

Figure 2: (Left panel) Histogram of the `baseline_ICU_capacity` (in number of beds) per trust. (Right panel) Illustration of the histogram of `local_stress` (expected demand of number of beds above capacity) per trust, before and after applying the load sharing procedure. In the synthetic example, all trusts are loaded with a uniform `projected_ICU_demand = 20`, whereas the `baseline_ICU_capacity` is the real one whose histogram is depicted in the left panel. Before the load sharing procedure, the `global_stress = 611`, whereas after the procedure, the new `global_stress = 101`, i.e. a reduction of a total of 510 ICU patients.

3.1.2 Pipeline of uniform-load stress tests

In a second step, we explore how the system behaves when the initial demand per trust varies. To do that, we consider a suite of stress tests and assume for each test that all trusts receive the same load --leading to a uniform demand per trust--, and we compute the `local_stress` before and after the load sharing procedure is applied. Accordingly, the `global_stress` of the whole system and the net reduction in the number of ICU beds in excess (in collapsed trusts) is also computed.

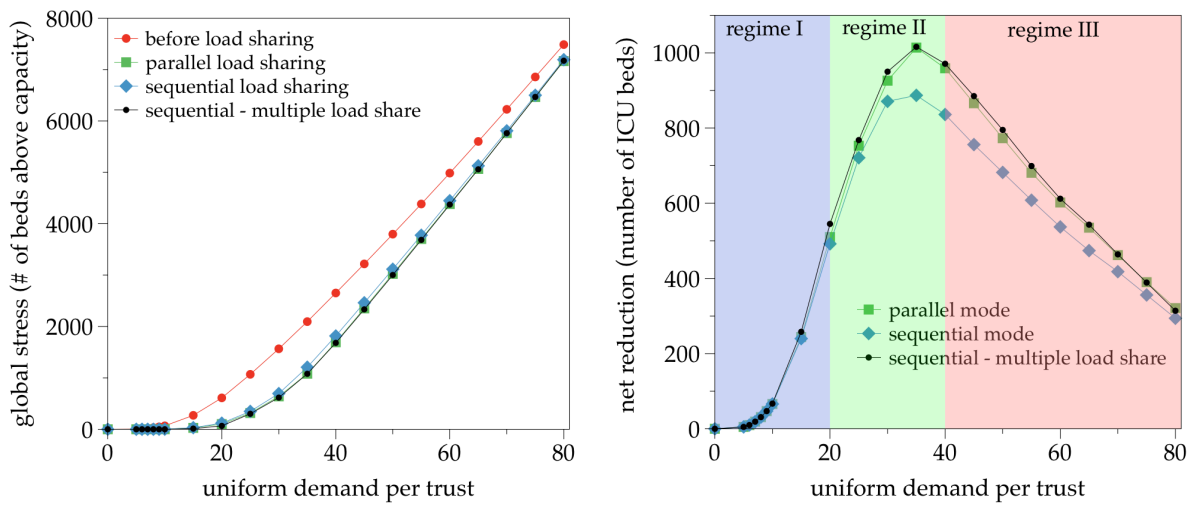

Figure 3: Response of the system (in terms of `global_stress` *and net reduction in number of ICU beds) after the load sharing procedure is applied, as a function of the initial demand per trust (uniform demand across trusts). Different lines correspond to different modes: absence of load sharing (red); single-share, parallel mode (green); single-share, sequential mode (blue); multiple-share, sequential mode (black). Results are similar across modes. We can see three regimes: an initial regime where the load sharing procedure easily removes all signs of overwhelming, a second regime where although the procedure cannot remove all signs of overwhelming, the net reduction is maximised, and a third regime where the load sharing procedure is less and less efficient due to the fact that the whole system is overwhelmed.*

Results are depicted, for both sequential and parallel mode, in Figure 3. In the left panel we plot the `global_stress` before and after the load sharing procedure is applied, as a function of the demand initially loaded uniformly for all trusts. As expected, the curves increase when the demand per trust is uniformly increased. At the beginning (for a uniform demand between 0 and 20 ICU beds per trust), the load sharing procedure works very well and completely removes any sign of overwhelming of the system (i.e. keeping the `global_stress` around zero). When the demand per trust increases further we enter in a second regime (between 20 and 40 ICU beds per trust) where the system shows serious signs of

overwhelming but the load sharing procedure removes a large portion of it. If the demand per trust increases above 40 ICU beds, the whole system is vastly overwhelmed, and the load sharing procedure becomes less and less efficient and the resulting net reduction decreases. Results are systematically better for the parallel mode than the sequential mode, but as previously mentioned, this comes at the expense of inevitably overwhelming some receptor trusts. Sequential mode still provides very good results and preclude receptor trusts from being overwhelmed.

3.2 Multiple-share

In this last section we relax the single-share assumption and allow each trust to share multiple loads to various receptor trusts, selected randomly from the trust's topological neighborhood. We only consider this option in the 'sequential processing mode', where real values of `local_stress` are updated in a sequential way as load sharing is performed.

In the uniform-load stress test, enabling a multiple-share option in the sequential mode provides an improvement in the net reduction of cases as compared with the single-share case, however such improvement is not massive (see Figure 3 for a comparison), and essentially puts the multiple-share sequential mode in a similar footing than the single-share parallel mode (but at the same time guaranteeing that no receptor gets overwhelmed). This result is easy to interpret: there is not an enormous gain in being able to share loads to different receptors (vs one receptor), because on average this possibility will only be useful in a handful of cases. In other words, this result is a byproduct of artificially imposing a uniform-load.
Something different is expected to happen if the initial demand on each node is not uniform. Suppose, for instance, that we have a few trusts that are extremely overwhelmed, and could in principle share loads with several receptors (more than one available receptor in its topological neighborhood), but suppose that those receptors are small trusts with only a small number of available ICU beds. In that case, a single-share approach is clearly deficient, but a multiple-share approach could indeed provide a notable improvement. We illustrate this case in the following section.

3.2.1 Heterogeneous-load stress test

Instead of loading a uniform demand in each trust, we now proceed to load a demand which is heterogeneous, where we only overwhelm 'large' trusts. Concretely, if the trust has a `baseline_ICU_capacity` larger than a certain threshold τ, then we set an initial value for `projected_ICU_demand` for this trust equivalent to 120% its `baseline_ICU_capacity` (i.e. this trust is overwhelmed with an excess of 20%). Similarly, for those trusts whose `baseline_ICU_capacity` is smaller than the threshold τ, we set an initial `projected_ICU_demand` for these trusts equivalent to 80% of their corresponding `baseline_ICU_capacity`.

We then apply the load sharing procedure in the sequential mode and compare the net reduction of the global level of stress (number of ICU patients that can be efficiently transferred) for the single-share and

the multiple-share options. In Figure 4 we plot these results as a function of the threshold τ, indeed finding that the multiple-share option is much more efficient in this case, as expected.

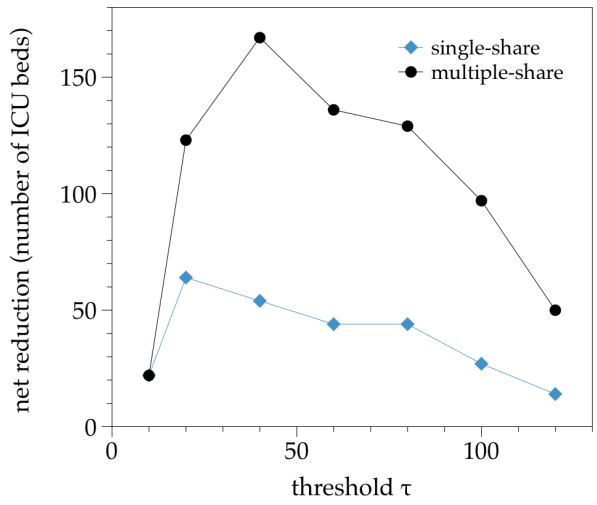

Figure 4: Net reduction offered by the sequential load sharing procedure vs the threshold τ (see the text), for a single-share and a multiple-share option. The multiple-share option clearly outperforms the single-share one in this case.

IV - Discussion

The COVID-19 pandemic is putting under stress the national health systems of several countries. Under this scenario, it is important to devise strategies to share the distributed capacity of hospitals: not only in terms of the number of ICU beds, as a matter of fact this is extensible to the overall capacity (critical care, acute capacity, number of ventilators, etc). Here we have detailed an algorithm and implementation for load sharing at the level of NHS trusts. All data and codes, as well as further versions, are and will be available at https://github.com/lucaslacasa/loadsharing . We have presented a proof of concept algorithm and implementation and showed that this procedure works well and can de-collapse the whole NHS network for a range of scenarios. The random search optimisation layer allows to explore non-intuitive load sharing configurations which go beyond the trivial solution to share load with the neighbor with highest capacity, a strategy which might be locally optimal but also might be leading to a global response far away from the global optimum. We have depicted and studied several options, such as sequential vs parallel update mode, and compared results of single-share (where a trust can only share load with a

single receptor) or multiple-share (where the trust can share parts of the load with different receptors of its neighborhood).

In the context of COVID-19 pandemic, we think that adopting a load sharing strategy is likely to be beneficial when (i) the whole system is not overwhelmed, and (ii) the projected ICU demand can be accurately estimated, and (iii) the facilities exist to transfer patients between ICU departments. This is likely to be the case (i) at the very beginning of the exponential growth phase, (ii) in situations with full lockdown where the demand is on decline and only some trusts are overwhelmed, or in general (iii) when the epidemic curve is on decline and not all trusts are overwhelmed. On the other hand, when the system is already fully overwhelmed or soon-to-be, this strategy is likely to be inefficient.

From a clinical point of view, an important point to consider is whether the load sharing can be activated at the ICU stage – potentially leading to transferring highly unstable patients who require ambulance with ICU equipment as well as trained personnel – or if, in anticipation to this, transfer needs to be planned at the point of hospitalisation (admission). In the latter scenario, planning needs to further take into account not only baseline ICU capacity, but overall capacity, also factoring in the estimated lag between admission to hospital and necessity of ventilator, which for COVID-19 is currently estimated at about 2 to 3 days. In a similar vein, note that this work considers the transfer of ICU patients, however a similar approach could be followed if the load to be shared is not patients but ventilators (the units to be shared are not ICU patients but ventilators, so transfer simply happens in the opposite direction, from receptor to origin). Assuming the receptor has both room and personnel to handle additional ventilators, this alternative would indeed eliminate the burden on transferring highly unstable patients and the associated resources required to make such transfers.

This work is of course subject to several limitations which we hope will be addressed in future iterations. First of all, the baseline ICU demand does not take into account surge capacity, that is expected to significantly increase the real ICU capacity of each trust. Also, the optimisation process implemented here is based on a stochastic search and as such there is no guarantee that the suggested configuration is the global optimum. Other refined methods such as hill climbing, genetic algorithms or simulated annealing can be used. Finally, we have assumed that the number of ambulances or the human resources are not a constraint, and that there are enough vehicles to transfer ICU patients effectively and enough qualified personnel to handle them. All these limitations can easily be addressed by suitably extending the specifications of the algorithm.

Code and data

https://github.com/lucaslacasa/loadsharing

Funding

LL gratefully acknowledges the financial support of the EPSRC via Early Career Fellowship EP/P01660X/1. RCh gratefully acknowledges the financial support of the EPSRC via grant

EP/N014391/1 and NHS England, Global Digital Exemplar programme. LD gratefully acknowledges the financial support of The Alan Turing Institute under the EPSRC grant EP/N510129/1.

References

1. https://www.who.int/emergencies/diseases/novel-coronavirus-2019/technical-guidance/naming-the-coronavirus-disease-(covid-2019)-and-the-virus-that-causes-it.
2. Huang, C. et al. Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. *Lancet* 395, 497–506 (2020).
3. Remuzzi, Andrea, and Giuseppe Remuzzi. "COVID-19 and Italy: what next?." The Lancet (2020).
4. Grasselli G, Pesenti A, Cecconi M. Critical Care Utilization for the COVID-19 Outbreak in Lombardy, Italy: Early Experience and Forecast During an Emergency Response. JAMA. Published online March 13, 2020. doi:10.1001/jama.2020.4031
5. Xie, J., Tong, Z., Guan, X. et al. Critical care crisis and some recommendations during the COVID-19 epidemic in China. Intensive Care Med (2020). https://doi.org/10.1007/s00134-020-05979-7
6. Arabi, Y.M., Murthy, S. & Webb, S. COVID-19: a novel coronavirus and a novel challenge for critical care. Intensive Care Med (2020). https://doi.org/10.1007/s00134-020-05955-1
7. Imperial College COVID-19 Response Team, COVID-19 Reports, https://www.imperial.ac.uk/mrc-global-infectious-disease-analysis/covid-19/
8. Alex Arenas, Wesley Cota, Jesús Gómez-Gardeñes, Sergio Gómez, Clara Granell, Joan T. Matamalas, David Soriano-Panos, Benjamin Steinegger, A mathematical model for the spatiotemporal epidemic spreading of COVID19, *medRxiv* 2020.03.21.20040022; doi: https://doi.org/10.1101/2020.03.21.20040022
9. R. Challen et al, Estimating local ICU demand from disease incidence in COVID-19, in preparation.
10. Danon, L., Brooks-Pollock, E., Bailey, M., & Keeling, M. J. (2020). A spatial model of CoVID-19 transmission in England and Wales: early spread and peak timing. *medRxiv*, doi: https://doi.org/10.1101/2020.02.12.20022566
11. L. Lacasa, R. Challen, S. Hendricks, E. Brooks-Pollock, L. Danon, Potential for monitoring COVID-19 cases via primary care prescription data, *in preparation*.
12. Data can be retrieved and processed from apps and other surveillance systems such as centralised webpages where citizens submit their symptoms. These questionnaires, coupled with a classification algorithm, can estimate the number of latent infected people in a certain region or postcode.
13. NHS England. Critical Care Bed Capacity and Urgent Operations Cancelled. https://www.england.nhs.uk/statistics/statistical-work-areas/critical-care-capacity/