**Improving speed of models for improved real-world decision-making**

Jason Thompson1, Haifeng Zhao1, Sachith Seneviratne1, Rohan Byrne1, Rajith Vidanaarachichi2, Roderick McClure2

Transport, Health and Urban Design Research Laboratory

University of Melbourne (UoM)

Victoria, Australia

Faculty of Health and Medicine

University of New England

New South Wales, Australia

**Abstract.** The sudden onset of the COVID-19 global health crisis and associated economic and social fall-out has highlighted the importance of speed in modeling emergency scenarios so that robust, reliable evidence can be placed in policy and decision-makers’ hands as swiftly as possible. For computational social scientists who are building complex policy models but who lack ready access to high-performance computing facilities, such time-pressure can hinder effective engagement. Popular and accessible agent-based modeling platforms such as NetLogo can be fast to develop, but slow to run when exploring broad parameter spaces on individual workstations. However, while deployment on high-performance computing (HPC) clusters can achieve marked performance improvements, transferring models from workstations to HPC clusters can also be a technically challenging and time-consuming task. In this paper we present a set of generic templates that can be used and adapted by NetLogo users who have access to HPC clusters but require additional support for deploying their models on such infrastructure. We show that model run-time speed improvements of between 200x and 400x over desktop machines are possible using 1) a benchmark ‘wolf-sheep predation’ model in addition to 2) an example drawn from our own work modeling the spread of COVID-19 in Victoria, Australia. We describe how a focus on improving model speed is non-trivial for model development and discuss its practical importance for improved policy and decision-making in the real world. We provide all associated documentation in a linked git repository.

**Keywords:** Agent-Based Model, High Performance Computing, Policy, Decision-making

1. Background

In 2020, Australia’s twin crises of the catastrophic ‘Black Summer’ bushfires and Covid-19 pandemic provided stark examples of crises that can be classified as X-events[1]; critical systems failures and crises that are at once extreme, sudden, novel, rare, surprising and disastrous. By definition, X-events have a relatively short unfolding time, but their impact is significant and may last decades or longer. As such, X-events hold important ramifications for the function of society and the intersection of science and policy.

A common feature of X-events is their association with the design and evolution of human, sociotechnical systems[2].They may be exacerbated by human systems, emerge as a result of activity within human systems, or are crises ofabstract, but critical human-designed systems that societies rely upon to function ‘normally'. These systems could include health care, banking and financial systems, communications, political, transport, economic and insurance systems.

As the world becomes increasingly connected through technological, social, geographic, and economic ties, the frequency of X-events is expected to accelerate[2, 3]. This is bad news. In Australia alone the bushfire and pandemic-related crises[4] absorbed hundreds of billions of dollars in direct and indirect costs[5], plunged it into the worst economic crisis since the Great Depression, and destroyed environment, lives, livelihoods, and property at unprecedented scale[6]. In turn, the effects of both crises reverberated across numerous associated systems, exposing significant weaknesses and generating fresh crises within healthcare[7, 8], housing[9], education[10], transport[11], economics[12], finance[13], environment[6], politics[14], and industrial relations systems[15], each requiring their own adjustments and policy responses. Reasonably, many of these consequent effects were unforseen because no empirical record of their occurrence existed prior to the event.

In the early stages of the COVID-19 pandemic, the dearth of historical record and available data compromised scientists’ typical means of gathering evidence about the world as well as their capacity to build models to deal with the crises and its corrolaries[16]. Because there was limited science to draw upon, it also compromised the application of evidence-based policy[17], which promotes ‘following the science’[18, 19].

In the absence of complete evidence, simulation modeling can provide a useful theoretical and practical bridge for scientists and policy-makers, alike. For example, a branch of simulation - computational social science - is the discipline of representing communities, societies and social phenomena through the generation of tangible, observable, but computer-generated artificial or ‘synthetic’ societies. By authentically representing both known (e.g., from evidence) and proposed critical features, structures, and mechanisms of interaction among agents within artificial societies, phenomena representing realistic potential crises befalling a society can be generated from the bottom-up[20]. Similarly, if crises within artificial societies can be generated, so too can policy solutions that prevent those same crises from arising. Simulation modeling of this type (primarily through agent-based modeling (ABM)) has demonstrated great utility across the world during the COVID-19 pandemic. It has proven to be a flexible, robust, and transparent tool that has provided valuable insight into the spatial, biological, and economic effects of the crisis and potential policy remedies[e.g., 21, 22-29].

However, despite greater awareness of ABM and its strengths, challenges remain. Very simply, the current time it takes to develop, analyse, and iterate trusted models of artificial societies is often too long to make them useful to policy-makers. This delay can result in either 1) disengagement by time-poor policy-makers who require faster answers to ‘what-if?’ questions than is currently possible, or 2) the real-world crisis moves on to a new phase that is outside the scope of the current model. In both cases, ‘the science’ has failed to keep pace with decision-makers’ needs [30].

There is therefore an urgent need for science and policy to connect better when faced with novel crises (e.g., COVID-19, environmental crises and natural disasters) requiring up-to date information and fast decision-making. Our own experience in working with policy-makers in both development and analysis of important social policy models demonstrates that the utilisation of HPC clusters is central to achieving this goal. That is, once a set of policy-settings are agreed upon, the ability to run experiments, analyse, and feed-back outcomes quickly is critical.

In this paper, we demonstrate the advantages for modelers working at the intersection of computational social science and policy-making of deploying existing policy models developed in NetLogo[31] on HPC clusters featuring parallel computing infrastructure. Our aim is to assist social science researchers make a transition to using HPC clusters by providing a generic framework for adaptation by individual users through a set of step-by-step instructions and scripts. These can be used as-is or modified with the assistance of local expertise to suit researchers’ own HPC environments.

1. Method

To demonstrate improvements in speed associated with the deployment of policy-models on HPC clusters, we used two examples. Firstly, the benchmark ‘Wolf-sheep predation’ model drawn from the standard NetLogo Models library. Secondly, we used a model developed in consultation with the Department of Health in Victoria, Australia to estimate risk associated with easing social restrictions after that state’s 2nd wave of SARS-CoV-2 infections in 2020[21, 28, 32]. To demonstrate both compute and real-time performance differences between various HPC set-ups, we also compared run-times for the benchmark model when allocated to the HPC across 1, 2, 4, 8, 16 or 32 cores.

In example 1, we used the standard ‘wolf-sheep predation’ representation with minor adjustments so that it runs in NetLogo BehaviourSpace. Changes to the model include the addition of a global variable ‘repetitions’ on line 3 of the model code (see Section 2.1, below), as well as the removal of the text pop-up warning on line 59 that halts the model when it reached a ‘max-sheep’ threshold.

A BehaviourSpace function was then created called ‘HPC\_Experiment’. This function included 100 random numbers under ‘repetitions’, and also included 5 levels across each of the variables: ‘wolf-gain-from-food’, ‘wolf-reproduce’, ‘sheep-gain-from-food’, ‘grass-regrowth-time’, and ‘sheep-reproduce’ for a total of 312,500 individual model runs. The maximum time-step limit for each run was set to 150.

Example 2 compared a single scenario of 100 model runs under Policy 4 (aggressive elimination) from the authors’ published COVID-19 epidemiological model[21, 28, 32]. We ran the model for a total of 1500 time-steps, equivalent to 1500 model days.

Both examples were first run on one of the author’s laptop (Intel® Core™, 4 cores, i7-7700HQ CPU @ 2.80GHz, 32GB RAM, Windows 10, 64bit OS). It was then deployed using the ‘snowy’ HPC partition on the University of Melbourne’s ‘Spartan’ HPC cluster[33] using 8 cores per task. This is a traditional cluster with a high-speed interconnect in one partition as well as an alternative queue that uses virtual machines with a common image. Computing jobs are submitted to a Slurm workload manager specifying which partition they would like to operate on (e.g., in our case, ‘Snowy’). Step-by-step information on how to prepare existing NetLogo models for deployment on the HPC, as well as the example benchmark model is contained at (<https://github.com/melbhz/netlogo-hpc>). A brief description of the procedure follows.

* 1. Description of the procedure for deploying NetLogo models on the HPC

Firstly, NetLogo must be installed to run on the HPC cluster. Next a NetLogo model must be created to match the format required for deployment (described below and in the documentation). Any NetLogo dependencies and plug-in packages (e.g., rngs, GIS, etc.) should be copied to the same folder to the NetLogo model or can be placed in the ‘extensions’ folder in the NetLogo extensions directory.

Regardless of the type of model being run, it must contain a named experiment within NetLogo’s BehaviourSpace function that contains a dummy input variable that bears no consequence to the function of the model (e.g., ‘repetitions’ from the example above). This dummy variable should contain a parameter space equal to the number of individual runs desired for each unique parameter combination. For example, the dummy variable ‘repetitions’ could contain a list of integers in the list [1:100]. Then, when combined with 3 policy setting choices on (for example) a real variable #1 (low, med and high) and 5 choices on (for example) a real variable #2 (very low, low, medium, high, very high), this creates 100 x 3 x 5 = 1500 individual model runs containing 100 runs of 15 separate policy combinations.

Using the set of templates available at (https://github.com/melbhz/netlogo-hpc), the user can open the ‘create\_xmls.sh’ file, altering the highlighted inputs to match their own NetLogo file location, the unique BehaviourSpace model name, and their unique output file name location(s). Running this file in the command line will create a folder containing 100 separate .xml files that will be submitted as independent jobs across the HPC cluster and contain the full complement of parameter combinations. Computation speed in this case is ideally about 200 times faster (for example, if moving from 4 cores on a desktop machine to 8 cores on the HPC), however the actual computation time will depend on the number of CPU cores available to the experimenter on the HPC cluster at any time. As a practical example, during semester when students or other staff are also using your HPC cluster for their own work, you can expect a longer duration between deploying your model and having results returned because your job may be ‘queued’ behind others as you wait for computing resources to become available (see Results for further details).

Next, the experimenter can open and revise the ‘submit\_jobarray.slurm’ file, again altering highlighted details related to file locations, input and output folders, as well as SBATCH settings unique to their own HPC environment. In our own case, we set: the number of nodes per job (we recommend 1), the desired partition on the HPC, the job name, the maximum run time for any job, and the desired CPUs per job (e.g.,8). In our specific case at the University of Melbourne, computing jobs that contain an additional ‘critical’ status are given priority over other jobs in the queue. Readers may also be able to request similar priority status from your local HPC administrators if applicable.

1. Results

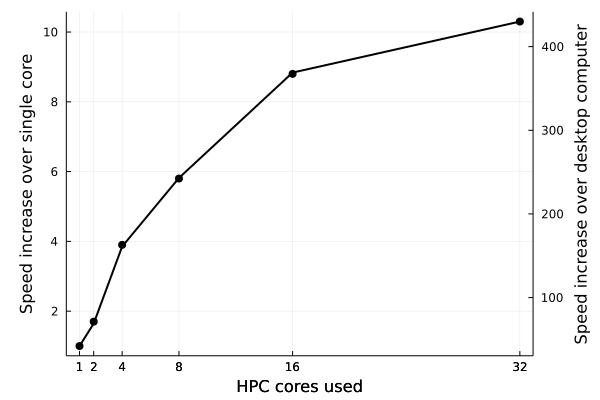
Run time for our benchmark ‘wolf-sheep predation’ model showed that when deployed on the author’s desktop computer, the complete set 312,500 trials was completed in 3 hours and 35 minutes. When deployed on the HPC cluster using 8 cores, CPU time was instead 5 minutes and 57 seconds, while job wall-clock time was 53 seconds. Deploying on the HPC cluster resulted in a 243x increase in real-time speed (See Figure 1).

Performance improvements when deployed on the HPC were similarly improved for our real-world COVID-19 policy model. When run on the author’s laptop, job completion took 2 hours, 26 minutes. By comparison, CPU time for the same model deployed on the HPC using 8 cores was 26 minutes and 16 seconds, while job wall-clock time was 3 minutes, 17 seconds. Deploying the policy model on the HPC cluster resulted in a 44.5x increase in real-time speed.

Finally, comparison of performance improvements gained on the HPC cluster when the number of cores allocated to the benchmark task was manipulated indicated diminishing speed returns with increased allocated cores. As shown in Figure 1, allocation of 8 cores (as used in the examples above) produced a 5.8x speed increase over using a single core. Increasing to 16 cores increased performance by a further 51% to 8.8x over the single core allocation. However, allocating 32 cores then only improved performance by an additional 17% to 10.3x the speed of a single core (see Figure 1).

It should also be noted that while wall-clock time was significantly improved in the 16 and 32 core conditions, in practice, allocation of the jobs to the HPC partition were delayed by 3 to 4 hours because the busy HPC cluster needed to wait for sufficient allocation space to become available before deploying the model – this was despite being provided with priority access on the network. This highlights (as described in Section 2.1) that it is important to be mindful of the trade-off between time gained through efficiency of processing vs HPC cluster access when making decisions about how and when to deploy models on individual – and possibly congested – HPC infrastructure.

It is also important to consider that this experiment did not incorporate any run-time parallelization load-balancing. The processing to be carried out was split at initialization to run across the allocated number of cores in a static manner. A more dynamic allocation of cores at run-time would potentially improve the diminishing returns observed in Figure 1. This is because during dynamic allocation the remaining processing is split across cores as soon as they finish the last workload assigned, whereas in static allocation it is possible that a core that has finished its workload may idle until the end of the job while other cores finish theirs. In general, the decision for deploying on HPC infrastructure can be informed by such characteristic curves. The optimal setting for deployment needs to weigh the benefits of increased speed versus the costs and availability of cores on the HPC infrastructure.



**Figure 1.** Comparative performance (speed) improvements gained over a single core allocation on the HPC cluster (left axis) and over the author’s desktop computer (right axis), and for the benchmark ‘wolf-sheep predation’ model.

1. Discussion

A hindrance to the uptake of computational social science and synthetic societies research programs to date has been the speed at which ABMs can be developed, explored, and run with results fed back to policy-makers. This was highlighted in the early stages of the COVID-19 pandemic where highly influential models in both the UK and Australia were adapted from existing influenza models rather than built as bespoke representations[34]. It shows that in urgent, unexpected crises where answers are demanded in minutes or hours rather than days (e.g., X-events), the capacity to ramp-up model speed and analysis is critical.

In our own work with the Victorian Department of Health during the second wave COVID-19 crisis in 2020[27, 28], extreme time pressure was exerted to match the timeframe of Victorian ‘Crisis Cabinet’ deliberations. Had our research group not been able to meet policy-makers’ timelines, advice would likely have been omitted, resulting in poorer decision-making, or decision-making that would have at least incorporated less information than more. After requested input parameter adjustments from Departmental officials covering wide parameter sweeps and the exploration of consequently large phase spaces, many model versions were run ‘overnight’ or over 24 hours split across multiple individual machines. This delayed timely provision of advice back to Government and consequent decision-making, which affected millions of Victorians’ lives as they waited for advice on when and how social restrictions would be lifted in response to declining COVID-19 infection rates. It also provided motivation to build and share these HPC templates for adaptation and use by other researchers lest the value of insights provided by computational social scientists and other simulation modelers (e.g., epidemiological forecasters) be neglected on the basis of that advice being too slow to produce.

Returning to the concept of X-events, the rapid production and analysis of models facilitated through deployment on HPC clusters is not only potentially advantageous for faster, more informed decision-making, but also for the speed and evolution of models, themselves. Because results are available more quickly, HPC powered ABMs can be iterated and evolve with faster turn-around time and are more likely able to match real-time developments of X-events (e.g., natural disasters including bushfires and floods) in matter of minutes by incorporating new information and data as it comes to hand. This is more likely to enable ABM simulation to be relevant in the face of rapidly unfolding crises that contain surprising or unexpected developments [1, 35].

1. Conclusions

Simulation models used in supporting important public policy and decision-making should be robust[36], but it is also important to recognise that sometimes decision-makers cannot wait for complete evidence before acting, especially in unfolding health crises or natural disasters[37]. The speed at which supporting synthetic evidence created through simulation modeling can be produced, analysed and presented is critical if science is to ‘keep pace’ with the world and delivers evidence in a form that directly addresses policy-makers’ real-time requirements[30]. The sooner evidence can be presented, the sooner it has a chance to be incorporated into decision-making, and the greater chance it has to positively affect the course of strategy, policy direction, action and outcomes. In addition to other documented performance improvement measures that can be achieved in simulation modeling platforms[38], the utilisation of HPC clusters can assist to bring the production and presentation of important evidence generated by simulation modelers forward in time.

References

1. Casti, J.L., *X-events: The collapse of everything*. 2012: Harper Collins.

2. Walsh, M.G., et al., *Whence the next pandemic? The intersecting global geography of the animal-human interface, poor health systems and air transit centrality reveals conduits for high-impact spillover.* One Health, 2020: p. 100177.

3. de Ruiter, M.C., et al., *Why we can no longer ignore consecutive disasters.* Earth's Future, 2020. **8**(3): p. e2019EF001425.

4. Flannery, T., *The megafires and pandemic expose the lies that frustrate action on climate change*, in *Fire, Flood, and Plague - essays about 2020*. 2020, The Guardian: Australia.

5. Commonwealth of Australia, *Budget 2020-21*. 2020: Canberra, Australia.

6. Binskin, M., A. Bennett, and A. Macintosh, *Royal Commission into National Natural Disaster Arrangements*. 2020, Royal Commission into Natural Disaster Arrangements: Australia.

7. MacIntyre, C.R. and D.J. Heslop, *Public health, health systems and palliation planning for COVID-19 on an exponential timeline.* The Medical Journal of Australia, 2020. **1**.

8. Blecher, G., G.A. Blashki, and S. Judkins, *Crisis as opportunity: how COVID-19 will reshape the Australian health system.* Med. J. Aust, 2020.

9. Power, E.R., D. Rogers, and J. Kadi, *Public housing and COVID-19: contestation, challenge and change.* International Journal of Housing Policy, 2020. **20**(3): p. 313-319.

10. Drane, C., L. Vernon, and S. O’Shea, *The impact of ‘learning at home’on the educational outcomes of vulnerable children in Australia during the COVID-19 pandemic*. 2020, National Centre for Student Equity in Higher Education, Curtin University ….

11. Beck, M.J. and D.A. Hensher, *Insights into the impact of COVID-19 on household travel and activities in Australia–The early days of easing restrictions.* Transport policy, 2020. **99**: p. 95-119.

12. O’Sullivan, D., M. Rahamathulla, and M. Pawar, *The impact and implications of COVID-19: An Australian perspective.* The International Journal of Community and Social Development, 2020. **2**(2): p. 134-151.

13. Andrew, J., et al., *Australia's COVID-19 public budgeting response: the straitjacket of neoliberalism.* Journal of Public Budgeting, Accounting & Financial Management, 2020.

14. Greer, S.L., et al., *The comparative politics of COVID-19: The need to understand government responses.* Global Public Health, 2020. **15**(9): p. 1413-1416.

15. van Barneveld, K., et al., *The COVID-19 pandemic: Lessons on building more equal and sustainable societies.* The Economic and Labour Relations Review, 2020. **31**(2): p. 133-157.

16. von Borzyskowski, I., et al., *Data science and AI in the age of COVID-19*. 2020, The Alan Turing Institute: London, United Kingdom.

17. Holmes, D., et al., *Deconstructing the evidence‐based discourse in health sciences: truth, power and fascism.* International Journal of Evidence‐Based Healthcare, 2006. **4**(3): p. 180-186.

18. Mercuri, M., *Just follow the science: A government response to a pandemic*. 2020, Wiley Online Library.

19. Abbasi, K., *Covid-19: politicisation, “corruption,” and suppression of science.* BMJ, 2020. **371**: p. m4425.

20. Epstein, J.M., *Generative social science: Studies in agent-based computational modeling*. 2006: Princeton University Press.

21. Blakely, T., et al., *The probability of the 6‐week lockdown in Victoria (commencing 9 July 2020) achieving elimination of community transmission of SARS‐CoV‐2.* Medical Journal of Australia, 2020. **213**(8): p. 349-351 e1.

22. Blakely, T., et al., *Association of Simulated COVID-19 Policy Responses for Social Restrictions and Lockdowns With Health-Adjusted Life-Years and Costs in Victoria, Australia.* JAMA Health Forum, 2021. **2**(7): p. e211749-e211749.

23. Kerr, C.C., et al., *Covasim: an agent-based model of COVID-19 dynamics and interventions.* medRxiv, 2020: p. 2020.05.10.20097469.

24. Abeysuriya, R., et al., *Estimating risks associated with early reopening in Victoria.* 2021.

25. Chang, S.L., et al., *Modelling transmission and control of the COVID-19 pandemic in Australia.* arXiv preprint arXiv:2003.10218, 2020.

26. Rockett, R.J., et al., *Revealing COVID-19 transmission in Australia by SARS-CoV-2 genome sequencing and agent-based modeling.* Nature Medicine, 2020. **26**(9): p. 1398-1404.

27. State Government of Victoria, *Emerging from lockdown: Evidence, modelling, outputs and assumptions*, D.o.H.a.H. Services, Editor. 2020, State Government of Victoria: Melbourne, Victoria.

28. State Government of Victoria, *Emerging from lockdown – model*, D.o.H.a.H. Services, Editor. 2020, State Government of Victoria: Melbourne, Victoria.

29. Milne, G.J., et al., *A modelling analysis of the effectiveness of second wave COVID-19 response strategies in Australia.* Scientific reports, 2021. **11**(1): p. 1-10.

30. Fischhoff, B., *Making Decisions in a COVID-19 World.* JAMA, 2020.

31. Wilensky, U., *NetLogo Version 6.2.0*. 2021, Centre for Connected Learning and Computer-Based Modeling, Northwestern University: United States of America.

32. Thompson, J., et al., *The Estimated Likelihood of Eliminating the SARS-CoV-2 Pandemic in Australia and New Zealand Under Current Public Health Policy Settings: An Agent-Based-SEIR Modelling Approach.* Available at SSRN 3588074, 2020.

33. Lafayette, L. and B. Wiebelt. *Spartan and NEMO: Two HPC-Cloud Hybrid Implementations*. in *2017 IEEE 13th International Conference on e-Science (e-Science)*. 2017.

34. Squazzoni, F., et al., *Computational Models That Matter During a Global Pandemic Outbreak: A Call to Action.* Journal of Artificial Societies and Social Simulation, 2020. **23**(2): p. 10.

35. Wilenius, M. and J. Casti, *Seizing the X-events. The sixth K-wave and the shocks that may upend it.* Technological Forecasting and Social Change, 2015. **94**: p. 335-349.

36. Calder, M., et al., *Computational modelling for decision-making: where, why, what, who and how.* Royal Society Open Science, 2018. **5**(6).

37. Thompson, J., R. McClure, and A. de Silva, *A complex systems approach for understanding the effect of policy and management interventions on health system performance.* Social‐Behavioral Modeling for Complex Systems, 2019: p. 809-831.

38. Railsback, S.F., et al., *Improving Execution Speed of Models Implemented in NetLogo.* Journal of Artificial Societies and Social Simulation, 2017. **20**(1): p. 3.