

Distributed state networks

Concept. To define and develop an archetype simulation environment for distributed state networks. In our classification scheme, this archetype is defined by an arbitrary, bidirectional state partition graph topology and would make sense for simulations of human brain or traffic control networks and power or water grids. We will also discuss the typical ways in which the state of the system (and its various partitions) may only partially be observed in realistic examples, and analyse how best to deal with each situation. For the mathematically-inclined, this chapter will define the mapping of our formalism to distributed state networks. For the programmers, the software which is designed and described in this chapter can be found in the public Git repository here: <https://github.com/worldsoop/worldsoop>.

8.1 Defining the archetype

The distributed state network archetype refers to simulation environments whose bidirectional state partition graph topology is completely arbitrary, requiring no particular connectivity structure at all. Each state partition in this archetype typically refers to the same type of real-world node, object or sub-model. Due to the flexibility in topological structure, this archetype is well-suited to ‘network’ models of realistic phenomena. We have illustrated the state partition graph which fits into this category in Fig. 8.1.

Before discussing what kinds of data are typically available to infer states and parameters of environments in this archetype, it will be informative to consider the various real-world problem domains which apply. The particular subset of these domains which seem to fit well within this classification include:

- Computational models of human brain conditions, e.g., Parkinson’s disease [1], epilepsy [2], Alzheimer’s [3], etc., for deep brain stimulation control and other forms of treatment.
- Simulations of complex urban infrastructure networks to target various kinds of optimisation, e.g., traffic signal control [4], power dispatch [5] and water pipe maintainance [6].

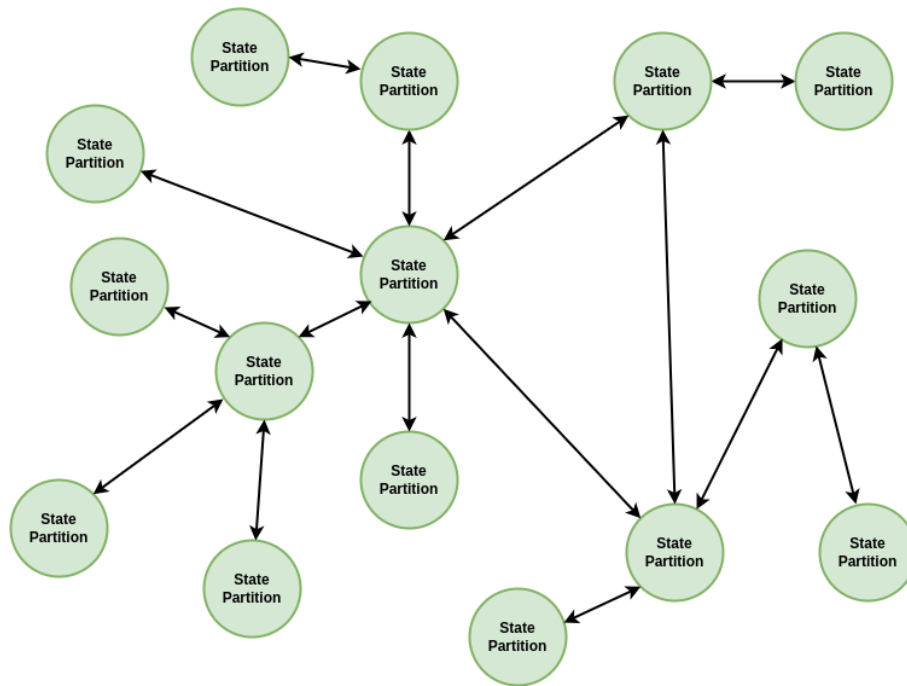


Figure 8.1: State partition graph topology for distributed state network archetypes.

8.2 Writing the code

Bibliography

- [1] M. Lu, X. Wei, Y. Che, J. Wang, and K. A. Loparo, “Application of reinforcement learning to deep brain stimulation in a computational model of parkinson’s disease,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 1, pp. 339–349, 2019.
- [2] J. Pineau, A. Guez, R. Vincent, G. Panuccio, and M. Avoli, “Treating epilepsy via adaptive neurostimulation: a reinforcement learning approach,” *International journal of neural systems*, vol. 19, no. 04, pp. 227–240, 2009.
- [3] K. Saboo, A. Choudhary, Y. Cao, G. Worrell, D. Jones, and R. Iyer, “Reinforcement learning based disease progression model for alzheimer’s disease,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 20 903–20 915, 2021.
- [4] K.-L. A. Yau, J. Qadir, H. L. Khoo, M. H. Ling, and P. Komisarczuk, “A survey on reinforcement learning models and algorithms for traffic signal control,” *ACM Computing Surveys (CSUR)*, vol. 50, no. 3, pp. 1–38, 2017.
- [5] Q. Li, X. Zhang, J. Guo, X. Shan, Z. Wang, Z. Li, and K. T. Chi, “Integrating reinforcement learning and optimal power dispatch to enhance power grid resilience,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 3, pp. 1402–1406, 2021.
- [6] Z. A. Bukhsh, H. Molegraaf, and N. Jansen, “A maintenance planning framework using online and offline deep reinforcement learning,” *Neural Computing and Applications*, pp. 1–12, 2023.