



Please cite the Published Version

Coates, Adam, Han, Liangxiu  and Kleerekoper, Anthony  (2018) A Unified Opinion Framework Simulator. In: 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), 10 July 2018 - 15 July 2018, Stockholm, Sweden.

Publisher: International Foundation for Autonomous Agents and MultiAgent Systems (IFAAMAS)

Version: Published Version

Downloaded from: <https://e-space.mmu.ac.uk/620629/>

Usage rights:  In Copyright

Additional Information: This is a conference paper which was originally presented at 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018)

Enquiries:

If you have questions about this document, contact openresearch@mmu.ac.uk. Please include the URL of the record in e-space. If you believe that your, or a third party's rights have been compromised through this document please see our Take Down policy (available from <https://www.mmu.ac.uk/library/using-the-library/policies-and-guidelines>)

A Unified Opinion Framework Simulator

Demonstration

Adam Coates
 Manchester Metropolitan University
 Manchester, UK
 adam.coates@stu.mmu.ac.uk

Liangxiu Han
 Manchester Metropolitan University
 Manchester, UK
 l.han@mmu.ac.uk

Anthony Kleerekoper
 Manchester Metropolitan University
 Manchester, UK
 a.kleerekoper@mmu.ac.uk

ACM Reference Format:

Adam Coates, Liangxiu Han, and Anthony Kleerekoper. 2018. A Unified Opinion Framework Simulator. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10-15, 2018*, IFAAMAS, 3 pages.

research. We developed a unifying framework such that many existing models can be decomposed into a set of independent rules. We identified the following rules:

We identified the following rules:

- Structural Rules - these determine the underlying graph structure (e.g. Lattice, Complete etc) and the distribution of initial values.

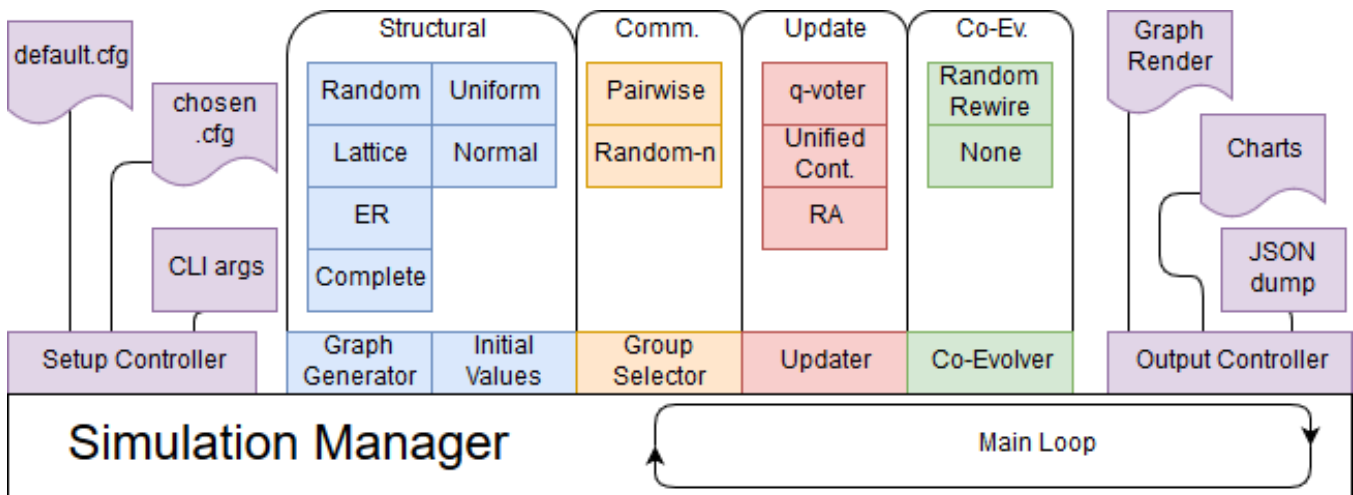


Figure 1: Simulation Control Flow

1 INTRODUCTION

In opinion dynamics, we see a wide range of different models for simulating the spread of opinions and influence over a social network, starting with the introduction of the voter model by Clifford and Sudbury [2]. At present, researchers wishing to explore and compare models must program each model from scratch. This presents three main problems: the expense of time that could be better spent in experimentation, the dangers of imperfectly replicating a model, and the hindrance this poses to direct comparison of models. For example, Meadows and Cliff wanted to propose a new opinion dynamics model and compare to Deffuant *et al*'s Relative Agreement (RA) model. They needed to implement RA from scratch but failed to replicate previous results because of an imperfect implementation (see [6] and [5]). This wasted much researcher time and effort.

In developing simulators for models, we noted a similar underlying structure between many of the models in active use within

- Communication Rules - these determine which agents interact with which other agents and whether these are individuals or groups.
- Update Rules - these determine if and how each agent updates its own opinion (and other internal parameters), after interacting with other agents.
- Co-evolutionary Rules - these determine how the structure itself changes in response to changing opinions.

Because the rules are independent, new models can be implemented by creating only one rule, rather than rewriting all the other rules as well. For example, a new model may only be new in that it has a new update rule. Using our unifying framework, the cost of implementing the new model is reduced to just the cost of implementing the new update rule. The new model can then be directly and accurately compared to other models by keeping all rules the same and just switching the update rule.

In this demonstration paper, we introduce our implementation of the unified framework. For more on the framework itself see [1].

Figure 1 shows the control flow of the simulator and how the provided modules map to the rule components of the unified framework. The simulator is written in Python and is designed to be easily extensible, system-independent and simple to use.

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10-15, 2018, Stockholm, Sweden. © 2018 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

Listing 1: The entire config file required to create the heatmap shown in Fig. 2

```

1 [graph]
2 alg = complete
3 n = 500
4 [group]
5 alg = pairwise
6 [update]
7 alg = relative_agreement
8 iterations_each = 400
9 [analysis]
10 alg = heatmap
11 [test]
12 initial-uncertainty = 0.2, 2.0, 20
13 initial-extreme = 0.01, 0.3, 20
    
```

2 PROVIDED MODULES

Several modules are provided for the user. The main modules are briefly described here.

Structural Rules. Structural rules are divided into two parts: Graph Generator and Initial Values. The following graphs are available: Complete, Erdős–Rényi graphs (Random), 2D Lattice, Barabási – Albert graphs (scale-free and small-world), and generic scale-free and small-world. Initial Values can either be uniformly distributed or follow a Normal distribution, capped to be within -1 and $+1$.

Communication Rules. The modules for these rules define which agents can communicate with others. We call this group selection. The Random- n group selection module uses the selection process of Urbig et al. [7]. Given an agent, it returns n neighbours of that agent. The Pairwise module is provided as a convenient shorthand for Random- n with $n = 2$, because this is a commonly used choice.

Update Rules. The two main update modules are capable of emulating many of the existing models, via a number of parameters. The q-Voter model emulates discrete models using a variant of the voter model [2], and the Unified Continuous model (see our other work [1]) produces the continuous models using a variant of bounded confidence [4]. There are also modules for popular rules such as Relative Agreement [3].

Co-Evolutionary Rules. Research into co-evolutionary rules is still in the early stages and not many rules have been proposed. The default is for no co-evolution, however, we also provide a Random Rewire co-evolutionary module that takes a pair of agents, severs the edge between them, and then establishes a new edge from the first agent to a randomly chosen node in the graph.

2.1 Running an Experiment

Experiments are conducted by linking together provided modules, as specified in a configuration file. New models can be implemented and added to those already available via APIs. As an example of the ease with which experiments can be run, and as a validation of our simulator, Deffuant *et al.*'s famous heatmap of the Relative

Agreement model (shown in Fig. 2) was generated with the 13-line config file shown in Listing 1.

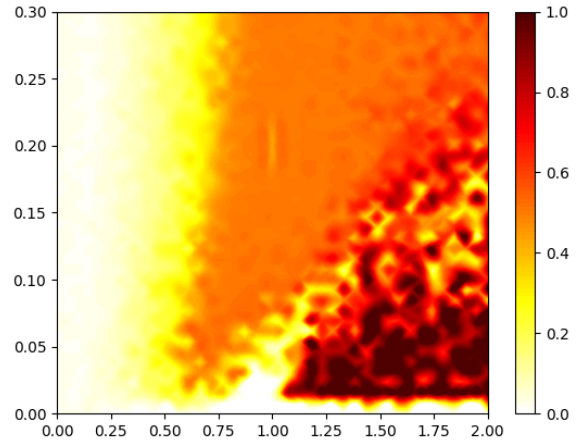


Figure 2: Replicating Deffuant et al’s famous heatmap [3] with our simulator requires only the few lines in Listing 1.

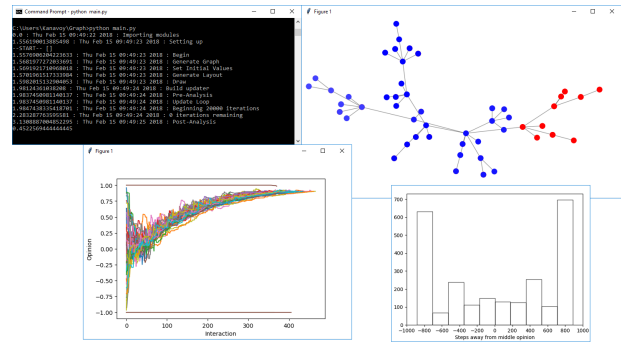


Figure 3: Examples of the outputs and visualisations currently available with our simulator

2.2 Output

The results of a simulation can be output in a number of ways. Firstly, the underlying graph and any updates can be dumped as a JSON output. Secondly, the graph can be displayed with nodes coloured by their opinion. The display is regularly updated during run-time to show changes to the graph and opinions. A user can also provide analysis modules which collect data after a simulation completes and can produce output. So far, our simulator contains three such modules capable of producing a heatmap of Deffuant’s y metric, a trace of the changes to opinions over time for all agents and a histogram of the distribution of agent opinions at the end of the simulation. Examples of all four outputs are shown in Fig. 3.

3 VIDEO URL

A video demonstration is available at <https://youtu.be/egU8zc8ADf0>

REFERENCES

- [1] Liangxiu Han Adam Coates and Anthony Kleerekoper. 2018. A Unified Framework for Opinion Dynamics. (2018).
- [2] Peter Clifford and Aidan Sudbury. 1973. A model for spatial conflict. *Biometrika* 60, 3 (1973), 581–588. DOI : <http://dx.doi.org/10.1093/biomet/60.3.581>
- [3] Frédéric; Weisbuch Gérard; Faure Thierry Deffuant, Guillaume; Amblard. 2002. How can extremism prevail? A study based on the relative agreement interaction model. *Journal of Artificial Societies and Social Simulation* 5, 4 (2002).
- [4] Guillaume Deffuant, David Neau, Frederic Amblard, and Gérard Weisbuch. 2000. Mixing beliefs among interacting agents. *Advances in Complex Systems* 03, 01n04 (1 2000), 87–98. DOI : <http://dx.doi.org/10.1142/S0219525900000078>
- [5] Guillaume Deffuant, Gérard Weisbuch, Frederic Amblard, and Thierry Faure. 2013. The Results of Meadows and Cliff Are Wrong Because They Compute Indicator y Before Model Convergence. *Journal of Artificial Societies and Social Simulation* 16, 1 (2013), 11. DOI : <http://dx.doi.org/10.18564/jasss.2211>
- [6] Michael Meadows and Dave Cliff. 2012. Reexamining the Relative Agreement Model of Opinion Dynamics. *Journal of Artificial Societies and Social Simulation* 15, 4 (2012), 4. DOI : <http://dx.doi.org/10.18564/jasss.2083>
- [7] Diemo Urbig, Jan Lorenz, and Herzberg. 2008. Opinion dynamics: The effect of the number of peers met at once. *Journal of Artificial Societies and Social Simulation* 11, 4 (2008).