

9-13-2018

Generating Strong Diversity of Opinions: Agent Models of Continuous Opinion Dynamics

Christopher W. Weimer

Follow this and additional works at: <https://scholar.afit.edu/etd>

Part of the [Operational Research Commons](#), and the [Social and Behavioral Sciences Commons](#)

Recommended Citation

Weimer, Christopher W., "Generating Strong Diversity of Opinions: Agent Models of Continuous Opinion Dynamics" (2018). *Theses and Dissertations*. 1956.

<https://scholar.afit.edu/etd/1956>

This Dissertation is brought to you for free and open access by the Student Graduate Works at AFIT Scholar. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of AFIT Scholar. For more information, please contact richard.mansfield@afit.edu.



**Generating Strong Diversity of Opinions:
Agent Models of Continuous Opinion Dynamics**

DISSERTATION

Christopher W. Weimer, Maj, USAF
AFIT-ENS-DS-18-S-044

**DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY**

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this document are those of the author and do not reflect the official policy or position of the United States Air Force, the United States Department of Defense or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENS-DS-18-S-044

GENERATING STRONG DIVERSITY OF OPINIONS:
AGENT MODELS OF CONTINUOUS OPINION DYNAMICS

DISSERTATION

Presented to the Faculty
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy in Operations Research

Christopher W. Weimer, BS, MS

Maj, USAF

13 September 2018

DISTRIBUTION STATEMENT A
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT-ENS-DS-18-S-044

GENERATING STRONG DIVERSITY OF OPINIONS:
AGENT MODELS OF CONTINUOUS OPINION DYNAMICS

Christopher W. Weimer, BS, MS
Maj, USAF

Committee Membership:

J. O. Miller, PhD
Chairman

Raymond R. Hill, PhD
Member

Douglas D. Hodson, PhD
Member

Abstract

Opinion dynamics is the study of how opinions in a group of individuals change over time. A goal of opinion dynamics modelers has long been to find a social science-based model that generates strong diversity — smooth, stable, possibly multi-modal distributions of opinions.

This research lays the foundations for and develops such a model. First, a taxonomy is developed to precisely describe agent schedules in an opinion dynamics model. The importance of scheduling is shown with applications to generalized forms of two models. Next, the meta-contrast influence field (MIF) model is defined. It is rooted in self-categorization theory and improves on the existing meta-contrast model by providing a properly scaled, continuous influence basis. Finally, the MIF-Local Repulsion (MIF-LR) model is developed and presented. This augments the MIF model with a formulation of uniqueness theory. The MIF-LR model generates strong diversity. An application of the model shows that partisan polarization can be explained by increased non-local social ties enabled by communications technology.

AFIT-ENS-DS-18-S-044

*To the memory of my grandmother, for whom every challenge was a puzzle to be
solved with impish glee*

Acknowledgements

First and foremost, I'd like to thank Dr. J.O. Miller for mentoring and advising me through both graduate degrees and for supporting my pursuit of a non-traditional research topic. You have helped me to grow both academically and professionally through a change in career fields that would not have been possible without you. My thanks also go to Dr. Ray Hill and Dr. Douglas Hodson for your support and mentorship. You have made this a tremendous experience.

Christopher W. Weimer

Contents

	Page
Abstract	iv
Acknowledgements	vi
List of Figures	xi
List of Tables	xiv
I. Introduction	1
1.1 Outline of Document	2
II. Literature Review	5
2.1 Modeling and simulation	5
2.1.1 Paradigms of simulation	6
2.2 Agent-based modeling and simulation	9
2.2.1 History	11
2.3 Opinion dynamics and related models	14
2.3.1 Cultural dynamics models	15
2.3.2 Discrete (binary-choice) opinion dynamics models	19
2.3.3 Continuous opinion dynamics models	27
2.3.4 Other modeling techniques that could inform new opinion dynamics models	46
2.4 Social science relating to opinion dynamics models	48
2.4.1 Choice shift	48
2.4.2 Interpersonal influence	56
2.4.3 Memory	58
III. Paper #1: Agent Scheduling in Opinion Dynamics: A Taxonomy and Comparison Using Generalized Models	60
3.1 Introduction	60
3.2 Synchrony, Actor type, Scale (SAS) scheduling taxonomy	64
3.2.1 Synchrony	65
3.2.2 Actor type	67
3.2.3 Scale	68
3.3 Generalized Repeated Averaging Model	70
3.3.1 Model definition	70
3.3.2 Parameter selection	75
3.3.3 Results	78
3.4 Generalized Bounded Confidence Model	82
3.4.1 Model definition	83

	Page
3.4.2	86
3.4.3	86
3.5	91
3.6	93
IV. Paper #2: Distilling Meta-contrast: The Meta-contrast Influence Field Model	94
4.1	94
4.2	95
4.2.1	97
4.3	99
4.3.1	99
4.3.2	101
4.3.3	107
4.4	108
4.5	113
4.6	115
V. Paper #3: Generating Strong Diversity with the Meta-contrast Influence Field Model	117
5.1	117
5.2	118
5.2.1	120
5.2.2	121
5.2.3	122
5.3	123
5.3.1	126
5.4	127
5.4.1	130
5.5	132
5.6	133
5.7	137
VI. Conclusion	139
6.1	140
References	143
Appendix A. Opinion Dynamics Models Summary	159

	Page
Appendix B. NetLogo Code: Replication of Carley (1991)	162
Appendix C. NetLogo Code: Replication of Mark (1998)	164
Appendix D. NetLogo Code: Replication of Axelrod (1997)	168
Appendix E. NetLogo Code: Voter Model	171
Appendix F. NetLogo Code: Replication of Nowak et al. (1990)	173
Appendix G. NetLogo Code: Agent-based implementation of Galam (1997)	176
Appendix H. NetLogo Code: Replication of Sznajd-Weron & Sznajd (2000)	179
Appendix I. NetLogo Code: Replication of Stauffer (2002)	182
Appendix J. NetLogo Code: Replication of Deffuant et al. (2000).....	185
Appendix K. NetLogo Code: Replication of opinion dynamics portion of Sun & Müller (2013)	189
Appendix L. NetLogo Code: Replication of Hegselmann & Krause (2002)	194
Appendix M. NetLogo Code: Replication of Deffuant et al. (2002).....	197
Appendix N. NetLogo Code: Replication of base model of Salzarulo (2006)	200
Appendix O. NetLogo Code: Replication of small-world network model of Salzarulo (2006)	204
Appendix P. NetLogo Code: Replication of Mäs & Flache (2013)	208
Appendix Q. NetLogo Code: Replication of Duggins (2017)	212
Appendix R. NetLogo Code: Generalized Repeated Averaging Model	216
Appendix S. NetLogo Code: Generalized Bounded Confidence Model	220
Appendix T. NetLogo Code: Meta-contrast Influence Field Model	226
Appendix U. NetLogo Code: Meta-contrast Model	231

	Page
Appendix V. NetLogo Code: Meta-contrast Influence Field - Local Repulsion Model	237
Appendix W. NetLogo Code: Meta-contrast Influence Field - Local Repulsion Model on U.S. Map	243
Vita	249

List of Figures

Figure		Page
1	Ideal-typical single-run distribution of cultures at convergence of Axelrod (1997) model for 10×10 , 20×20 , and 100×100 grid geographies	19
2	Approximately random walk behavior of a voter model leading to consensus	21
3	Outcomes of replication of Nowak-Szamrej-Latené model (Nowak et al. 1990) with limited range (left) and unlimited range (right)	23
4	Outcomes of replication of Nowak-Szamrej-Latené model using alternative formulae (Nowak et al. 1990) with limited range (left) and unlimited range (right)	23
5	Paired outcomes of G (left) and $ C $ (right) from Galam (1997) model using random order and random independent schedules	26
6	Ideal-typical results of base Deffuant-Weisbuch model converging to 2 groups and 1 wing, with $d = 0.25, \mu = 0.5$	30
7	Three instances of output from a model derived from Sun & Müller (2013), yielding convergence to moderate opinion (top), convergence to an extreme opinion (middle), and within-group convergence to opposing extreme opinions (bottom)	34
8	Ideal-typical results of base Hegselmann-Krause model converging to one group ($d = 0.25$, top) and two groups ($d = 0.20$, bottom)	36
9	Example outcome of meta-contrast model with $a = 0.08$ and $w = 0.36$, using a small-world network	40
10	Graphical view of flow of influence for three types of actor	69
11	Histogram of observed opinions at convergence for GRAM, where $(s, t) = (2, 1000)$	79
12	Histogram of observed opinions at convergence for GRAM, where $(s, t) = (\infty, 1000)$	79

Figure	Page
13	Variance in observed opinions at convergence for GRAM, where $(s, t) = (2, 1000)$, by primary actor type: Target (left), Source (middle), Group (right) 81
14	Variance in observed opinions at convergence for GRAM, where $(s, t) = (\infty, 1000)$, by primary actor type: Target (left), Source (right) 81
15	Mean observed opinions at convergence for GRAM 81
16	Histogram of observed opinions at convergence for GBCM, where $(s, t) = (2, 1000)$ and $d = 0.20$ 87
17	Histogram of observed opinions at convergence for GBCM, where $(s, t) = (\infty, 1000)$ and $d = 0.20$ 87
18	Mean observed opinions at convergence, averaged across clusters, per replicate of GBCM, where $d = 0.20$ 90
19	Mean number of clusters at convergence per replicate of GBCM, where $d = 0.20$ 90
20	Variance in number of observed clusters per replicate of GBCM, where $d = 0.20$ 91
21	Maximum values as function of w (left), and ratio of these values as a function of w (right) 103
22	Basis of influence when $a = 0.08, w = 0.36, X = \{x_0, 0.2, 0.8\}$ 105
23	MC prototypicality function when $a = 0.08, w = 0.36, X = \{x_0, 0.2, 0.2, 0.2, 0.2, 0.2, 0.8\}$ 106
24	Influence field at x_1 when $a = 0$ (left) and $a = 1$ (right), $w = 0.36, X = \{0, x_1\}$ 106
25	Mean number of clusters that form in the MC model (left) and MIF model (right) 109
26	Individual model runs where $w = 0.5$ and $a = 0.25$ (top), $a = 0.5$ (middle), and $a = 0.75$ (bottom) 110
27	Individual model run where $w = 0.15$ $a = 0.5$ 111
28	Normalized opinion variance for MIF model using a SWN 112

Figure	Page
29	Normalized opinion variance for MC model using a SWN 112
30	Shape of numerators of d_{intra} , d_{indiv} , and their sum for $w = 0.4, b = 0.5$ 127
31	Distribution of opinions after 2000 updates, single replicate (left) and 100 replicates (right) $w = 0.45$ 128
32	Distribution of opinions after 2000 updates, single replicate (left) and 100 replicates (right) $w = 0.23$ 129
33	Distribution of opinions after 2000 updates, single replicate (left) and 100 replicates (right) $w = 0.15$ 129
34	Distribution of opinions after 2000 updates, single replicate (left) and 100 replicates (right) $w = 0.12$ 129
35	Single-run trajectory of opinions over 2000 updates 130
36	Distribution of opinions after 3000 updates, varying $b \in \{0, 0.25, 0.5, 0.75, 1\}$ 131
37	MIF-LR results varying network randomness: single-run distribution (left), trajectory (center), and 100-run distribution (right) 136
38	Distributions resulting from 100 replicates of MIF-LR model varying initial conditions 141

List of Tables

Table		Page
1	Summary of SAS taxonomy by component	65
2	Schedules of opinion dynamics models using SAS taxonomy	66

GENERATING STRONG DIVERSITY OF OPINIONS:
AGENT MODELS OF CONTINUOUS OPINION DYNAMICS

I. Introduction

Opinion dynamics is the study of how opinions in a group of individuals change over time. This broad topic is of direct interest to fields as diverse as marketing, military influence operations, politics, and law. Opinion dynamics was originally in the sole domain of sociologists and social psychologists who draw correlations in behavior and provide theories for the causal structures but typically cannot verify the veracity of sometimes conflicting theories. More recent research has seen mathematical models of opinion dynamics developed by scientists from fields ranging from physics and mathematics to psychology and sociology.

Generative social science (GSS) provides an avenue to confirm or reject theories that imply a given theory is sufficient to generate realistic behavior. GSS is a simulation paradigm in which theories derived from social science are given mathematical formulation and simulated in a simplified population of software agents in order to test whether those theories are sufficient to generate realistic behaviors of interest (Epstein 2006). Despite decades of sociological and psychological theorizing related to opinion dynamics, the state of mathematical models of opinion dynamics is immature. Recently, the first continuous opinion dynamics model to generate realistic distributions of opinions was published (Duggins 2017), and it relies upon a complicated array of inputs and at least one assumption that is not strongly supported by the psychological literature. Furthermore, most opinion dynamics models in the literature do not justify critical assumptions relating to the order in which agents act,

despite the existence of path dependencies in complex patterns of interaction.

The following research contributes two new models to the opinion dynamics: one is a framework based upon social psychological literature that readily allows augmentation with other theoretic forces, and the other expands upon the first to provide the second known example of an opinion dynamics model that can generate realistic distributions of opinions. These distributions are characterized by strong diversity — diversity both within and between clusters of opinions — and can be formed as unimodal, bimodal, or multimodal distribution. It also provides a taxonomy for describing the order in which agents act in such a model along with generalized forms of models from the opinion dynamics literature that demonstrate the overwhelming effect changes in schedule can have upon behavior.

1.1 Outline of Document

This document is presented in a k-paper format. It begins with a detailed review of the literature relevant to the state of the art of agent-based opinion dynamics modeling. This includes both a survey of opinion dynamics models in the literature, with many of them replicated by code provided in the appendices, and a survey of the social scientific literature that relates directly to those models.

The following three chapters are constructed as independent journal articles, and therefore include relevant literature review sections along with associated methodology, results, discussion, and conclusions.

The first article defines a taxonomy for communicating the agent schedule of an opinion dynamics model, which specifies how many agents influence or are influenced by how many other agents in what order at each discrete step of the model. This fills a gap in communication and allows clear, precise, and concise description of a proposed model. The impact that may be caused by altering the schedule in two

often-used opinion dynamics models is demonstrated using generalized forms of those models.

The second article introduces the meta-contrast influence field (MIF) model to the literature. This expands on the meta-contrast (MC) model of Salzarulo (2006) in several ways. First, it equalizes the scale of previously imbalanced inputs into the prototypicality function. This prototypicality function computes how prototypical an agent will perceive other opinions to be, and it is the backbone of the MC model. The inter-group component of that prototypicality dominates the calculation in the MC formulation. Second, it implements a continuous source of influence based upon the derivative of the MC model's prototypicality function, which computed how prototypical an agent would perceive an opinion to be. Using the derivative, in effect, creates an influence field composed of competing forces acting upon the point of an agent's opinion. Finally, as a result of this field-based formulation, the schedule is altered to allow all agents to update in synchrony based upon the information available to them. The MIF model is a significant improvement upon the MC model, and its construction allows it to be a framework upon which more nuanced models of interpersonal influence can be built.

The third article updates the MIF model to include a drive for individualization based on uniqueness theory (Fromkin & Snyder 1980). By including local repulsive forces into the influence field, the meta-contrast field local repulsion (MIF-LR) model generates strong diversity of opinions. This outcome has been sought after for over 50 years since Abelson (1964) famously asked "what on earth one must assume in order to generate the bimodal outcome of community cleavage studies" [p. 153]. Examples of parameter settings that generate unimodal, bimodal, trimodal, and quattrimodal distributions are provided. An application that generates partisan polarization as an outcome of increasing random ties in social networks is also explored. These

distributions are stable with respect to time and repeatable between experiments. Furthermore, this is a parsimonious model with only 8 parameters, of which only 4 define agent behavior in a manner unique to this model and 3 define the social network itself. It is also firmly based in social psychological literature.

II. Literature Review

2.1 Modeling and simulation

A system is a closed set of entities interacting in the real world. A model is an abstraction of a system. This could be a physical model, such as a scaled model airplane, or a mathematical model, such as the relationship between force, mass, and acceleration integral to Newtonian physics. Mathematical models could be systems of equations that can be solved analytically or they could be computer models (or simulation models) that are implemented via software and typically have no analytical solution (Law 2007). For the purpose of this paper, I focus primarily upon computer modeling, the use of software to approximate a process seen in reality.

For a computer model, simulation is the act of using a model. This typically includes setting initial conditions for the model, implementing the rules of the model upon those conditions, and generating data from the outputs of that model. Simulation provides a method by which we can experiment on approximations of real physical systems that, for some reason, we cannot reasonably experiment upon directly. A common motivation for simulation is cost; it may be prohibitively expensive in terms of time, money, and/or manpower to experiment with a real-world system. In the social sciences, ethics may play a larger role; many experiments that might be scientifically of interest would be unethical to perform upon human participants.

Important distinctions between types of models relate to their use of random elements and time. Mathematical models may be either stochastic or deterministic: *stochastic* models implement random or pseudo-random elements so that outcomes vary for a given set of inputs, whereas *deterministic* models have only one outcome for a given set of inputs (Law 2007). Models may also be static or dynamic: *static* models do not include a time component, whereas *dynamic* models evolve over sim-

ulated time. How a dynamic model evolves over time can also vary: *discrete* models change at set points in time whereas *continuous* models change continuously, typically by implementing differential equations (Law 2007). Discrete models may further be distinguished between continuous-time and discrete-time models: *discrete-time* models are concerned only with the order of events and not the time that they occur, whereas *continuous-time* models are concerned with the time that events occur. For the purpose of this paper, I focus primarily upon stochastic, dynamic, discrete models.

2.1.1 Paradigms of simulation

These distinctions between types of mathematical models have led to the existence of distinct paradigms of dynamic simulation. Three of the most common paradigms are system dynamics, discrete event, and agent-based or individual-based models.

2.1.1.1 System dynamics

System dynamics (SD) models are concerned with continuous feedback loops between elements of a system. SD models are deterministic, dynamic, continuous-time models. A SD model is at its core a mathematical model comprised of systems of integral equations with continuous time, which constitute a continuous model (Kunc 2016). In practice, an analytical solution is often computationally intractable, and these systems of integral equations are solved approximately by simulating with discrete time steps that are sufficiently small to approximate continuity (Kunc 2016).

SD models are usually communicated in terms of stocks and flows. *Stocks* represent the state of the system at a given point in time, while *flows* represent the rate of change of a stock as a function of the stocks in the system. An example of a system amenable to SD modeling would be the fishing industry. Consider the stock of a type of fish and the stock of food available to the fish. There are several relevant flows as

well: the birth rate, the natural death rate, and the fishing rate are obvious examples. These flows are functions of the stocks of fish and food, while the stocks of fish and food are functions of the flows.

2.1.1.2 Discrete-event simulation

Discrete-event simulation (DES) models are concerned with systems whose state changes at discrete points in time (Banks et al. 2010). DES models are stochastic, dynamic, discrete, continuous-time models. A DES model consists of entities, processes, and a set of state variables. Entities may have their own sets of attribute variables that can vary between entities. During the simulation a future event list (FEL) and a clock variable are used to maintain the schedule in memory. Processes implement events known as *activities*, typically according to a function of state variables, attribute variables, and random variables. *Delays* may also occur as an entity must wait until a resource (a common state variable) is freed up by completion of another activity.

DES models are usually communicated in terms of a series of events. An example of a system amenable to DES modeling would be a shop queue. Entities (shoppers) arrive sporadically to the system when they complete their shopping, typically according to some probability distribution. They then wait in a queue until a resource (cashier) is available. Once that resource is available, the entity enters the process of checking out. The duration of this activity is a function of the number of items a shopper has chosen (the entity's attribute) and some random variation. The entity then departs the process and the system, freeing up a resource for another entity.

2.1.1.3 Agent/individual-based modeling

Agent- or individual-based models (ABM) are typically concerned with systems with interacting entities that communicate with their environment and other entities according to internal rules rather than an externally defined process. Agent-based modeling is the more common term, so it is used here, but individual-based modeling is a commonly used term in ecology (Railsback & Grimm 2011). ABMs are stochastic, dynamic, discrete, and typically discrete-time models. Hybrid models may be built that combine ABM with DES or SD models and introduce continuous-time modeling. The focus of ABM is typically emergence, which may be defined as “the arising of novel and coherent structures, patterns, and properties through the interactions of multiple distributed elements” (Wilensky & Rand 2015, p. 6).

ABMs are usually communicated and built in a bottom-up way by defining the agents first rather than any overarching process. This allows examinations of emergent behavior, a characteristic of complex systems which may be defined as higher-level behaviors that arise from lower-level entities’ localized behaviors (Bonabeau et al. 1995). An example of an emergent system amenable to ABM would be a flock of birds. Although the emergent behavior of flocking is easily observed and modeled from the process-overview top-down perspective, a more satisfying and predictive model can be built by implementing agents (commonly known as boids) that implement rules for collision avoidance, velocity matching, and flock centering (Reynolds 1987). By finding a set of rules that result in realistic emergent behavior, we can develop theories for the cognitive processes that result in flock behavior. This is a classical use of ABM.

2.2 Agent-based modeling and simulation

As has been noted by Axelrod & Tesfatsion (2006), agent-based modeling is particularly well-suited to the social sciences; as such, it is the primary focus of this document. Therefore, let us consider more deeply the history, current state, and continuing issues in ABM. First, a more formal definition of ABM is required.

There is no one agreed-upon definition for an agent or an agent-based model; in fact, this definition has been a point of significant contention between researchers. Most of this disagreement has centered around the degree of complexity an agent must possess in order to qualify as an agent. On the broad acceptance end of the spectrum, Bonabeau (2002) argues that ABM is more a mindset than a particular tool and that agents exist whenever a system is described in terms of its pieces. He specifically includes systems of differential equations, where those equations are defined at the individual level, within this definition. On the other side of the spectrum, Casti (1997) argues that an agent must possess adaptive behavior; they must be capable not only of following rules but also of changing those rules in response to their environment. In a nod to this disagreement, North & Macal (2007) make a distinction between adaptive ‘agents’ and non-adaptive ‘proto-agents.’

Most other definitions fall somewhere in between these two extremes, occasionally with concessions made to those at the poles. A helpful categorization of these definitions is provided by Macal (2016). Rather than define ABM as a whole, he distinguishes four separate definitions of ABMs with increasing complexity: individual, autonomous, interactive, and adaptive ABM. *Individual* ABM exists when “the agents in the model are represented individually and have diverse characteristics” (Macal 2016, p. 149), which is analogous to Bonabeau’s 2002 definition. In *autonomous* ABMs, “agents have internal behavior that allow them to be autonomous, able to sense whatever condition occurs within the model at any time, and to act

on the appropriate behavior in response” (Macal 2016, p. 149); this is similar to definitions by Gilbert & Troitzsch (2005), Macal & North (2014), Wilensky & Rand (2015), and C. W. Weimer et al. (2016). In *interactive* ABMs, “autonomous agents interact with other agents and with the environment” (Macal 2016, p. 150); this accommodates definitions by Epstein & Axtell (1996), Axelrod & Tesfatsion (2006), and Railsback & Grimm (2011). In *adaptive* ABM, “interacting, autonomous agents change their behaviors during the simulation, as agents learn, encounter novel situations, or as populations adjust their composition to include larger proportions of agents who have successfully adapted” (Macal 2016, p. 150); this meets the high bar set by Casti (1997). Although this is a very useful conceptualization, a weakness of this set of definitions is that cellular automata could be considered interactive ABMs.

In the field of opinion dynamics, which builds theoretical models of how opinions flow between individuals, there have been few attempts to lay claim to the term ABM, but for the sake of shared information in this research, there is value in considering them as such. The definition of adaptive ABM would be too restrictive to allow many of these models. On the other hand, the definition of individual ABM raises questions of whether DES and SD are actually different paradigms, as they can be defined at that level, so that definition seems not to be restrictive enough. Furthermore, calling agents autonomous is misleading and has led to arguments by nay-sayers; no agent is truly autonomous because it follows only the rules set by the modeler. A truly autonomous agent would require true artificial intelligence. However, a greater degree of autonomy is a defining feature of ABM. Another defining feature of ABM, which differentiates it from cellular automata, is the ability of agents to move within their environment. Therefore, for the purposes of this document, we will use the definition of ABM from C. W. Weimer et al. (2016):

An ABM is a simulation framework, using primarily the discrete-event scheduling paradigm, where the entities within the simulation have a greater degree of autonomy in movement and decision making than generally found in simulation models. (p. 67)

Some of the literature in the field of opinion dynamics, especially early work, does not fit this definition although it typically incorporates the individual-focused, bottom-up focus of ABM. Therefore, I will delineate between ABMs as defined above and models utilizing what Bonabeau (2002) called the “ABM mindset”:

The ABM mindset consists of describing a system from the perspective of its constituent units (p. 7280).

Models built with the ABM mindset include some systems of differential equations, Markov chain models, discrete-event simulations, cellular automata, and other types of models that would otherwise be excluded but spring from the same point of view and thus possess rules, properties, and insights that may be incorporated in full ABMs. For the purpose of this paper, the term ABM is specific to the former definition while the term agent extends over both definitions. Agent, individual, and entity are thus used interchangeably.

2.2.1 History

Agent-based modeling arose from the field of cellular automata, so they share an early history. Cellular automata are grids of entities, known as cells, that interact with their neighbors to modify their states. Typically, cells are arranged in a 1- or 2-dimensional grid, although they can theoretically exist in any number of dimensions, and their states are typically defined as binary, although again they can theoretically have any finite number of states (Fatès 2014). Each cell has identical rules of behavior and does not move. Typically, cells are updated synchronously and deterministically,

although exceptions exist in the form of stochastic and asynchronous CAs (see Fatès 2014).

Cellular automata (CA), and the theory of automata in general (which includes agents), owe their beginnings to the work of John von Neumann. His lectures and manuscripts through the 1940s and 50s were published posthumously in 1966 in a volume generally credited with popularizing the notion of cellular automata (von Neumann 1966). Unfortunately, von Neumann died before finishing a coherent theory of automata, but Stephen Wolfram has carried that torch in the area of CA. His collected works and more recent book on the topic detail practical uses on CAs and importantly exhaustively define the possible rules for a 1-dimensional CA whose actions depend solely on the states of their immediate neighbors, which he called elementary cellular automata and are commonly referenced according to his rule-numbering convention (Wolfram 1986, 2002).

One of the earliest and most well-known works in CA was formed shortly after von Neumann introduced the theory; John Conway combined CA with game theory to produce the Game of Life, brought into public consciousness by Gardner (1970). The Game of Life is a synchronous, deterministic CA (or in game theory terms, a zero-player game) with relatively simple rules that demonstrates complex emergent behavior without randomness. The wealth of stable and periodic outcomes is fascinating given its simplicity.

The natural growth of CA led to relaxation of rules. Relaxation of synchronous updating and determinism led to the fields of asynchronous and stochastic CA, examples of which are well-detailed by Fatès (2014). Stochastic asynchrony can come in the form of purely asynchronous updates, where one cell is updated at random at each step; α -asynchronous updates, where each cell has probability α of updating at each step; random order updates, where all cells are updated at each step in random

order; or other less common schedules. Deterministic asynchrony typically is in the form of fixed order updates, where each cell is updated in the same order at each step, although other possibilities such as incentive-based updates exist (see Page 1997).

Another relaxation of the rules of CA was to break the rules of immobility and homogeneity of cell rules. Mobile cells, especially with heterogeneous rules, fit our definition of agents; indeed, this is how agent-based modeling was born. Schelling (1971) is usually credited with the first agent-based model, which he operated by hand rather than by computer. He showed that even a mild intolerance for living in a highly mixed neighborhood results in large-scale housing segregation using a very simple homogeneous ABM. Further explorations of this model (Schelling 1978) were partially credited with Schelling winning the Nobel Prize in Economics in 2005. ABM was kicked off with distinction by a social scientist.

Interest in ABM subsided thereafter until computers grew powerful enough to handle the multitude of required calculations more quickly and easily. The next ABMs to reach popularity were Echo and Sugarscape in the mid-1990s. Echo is a model of evolution based upon characteristics, which may be considered elements of a gene describing an individual (Holland 1995). Holland is also credited with starting the field of genetic algorithms, which have become popular heuristic methods for optimization and illustrate that the perspective of ABM lends itself to use in problems beyond simulation. Sugarscape can be seen as an extension of Schelling's model; various distributions of resources over an environment (so-called sugarscapes) led to emergence of distinct cultures of individual agents (Epstein & Axtell 1996).

ABM's beginnings provide insight into the multi-disciplinary flavor that ABM has developed. As mentioned, Schelling was an economist and Nobel laureate. Holland was trained as a mathematician but served as both a Professor of psychology and a Professor of electrical engineering and computer science. Epstein's doctorate was

in political science, and he is currently a Professor of emergency medicine with joint appointments spanning the health, social, and mathematical sciences.

2.3 Opinion dynamics and related models

Building upon the social scientific roots of agent-based modeling, and owing to the nature of many social scientists to think using an agent-based mindset, a wealth of models examining opinion formation by individuals in a social environment have arisen¹. The emergent behavior of a group of agents forming their own individual opinions is of primary interest to these *opinion dynamics models*. This serves at least two purposes, one based in generative social science and another based in forecasting.

From a generative social science perspective (see Epstein 2006), opinion dynamics models are useful for identifying individual rules that can generate emergent behavior of interest. In other words, it can serve as a validation tool for social psychological theories of human opinion formation — if the theory, when applied, generates realistic group opinion dynamics, then it can be considered a sufficient condition. This validation is iterative; if a model shows that a social psychological theory is sufficient to generate realistic group behavior, the next step is to further validate the construction of the model with psychological experimentation.

From a forecasting perspective, there are innumerable applications that would benefit from a realistic model of how opinions might permeate a crowd and how one might influence these dynamics. In the business world, marketing is an example of a field that could be revolutionized by validated models of opinion dynamics. In a military context, military information support operations (formerly known as psychological operations) planners would gain a strong advantage from such a model

¹There are also opinion dynamics models that do not arise from the agent-based mindset. However, they are not of interest for the present research unless their insights spawned agent-based work.

through realistic war-gaming opportunities.

Early work related to opinion dynamics focused on a related concept of culture. These are relevant to the present research in their contributions to the opinion dynamics literature. This chapter will first examine these cultural dynamics models and follow up with an in-depth examination of a broad range, although certainly not an exhaustive list, of the most influential opinion dynamics models in the literature. Opinion dynamics models are broken up into discrete (i.e., binary choice) opinion models and continuous opinion models. A summary of all reviewed models in tabular form is given in Appendix A.

2.3.1 Cultural dynamics models

Culture is a difficult concept to precisely define. Recognizing this, but needing some frame of reference for his cultural dynamics model, Axelrod (1997, p. 204) defined culture as “the set of individual attributes that are subject to social influence.” Within this definition of culture, there is no distinction between culture and a set of discrete-valued opinions, which makes this work particularly relevant to the field of opinion dynamics. These models are all built with an agent-based mindset and are replicable using agent-based tools. However, if a geography is defined at all, it takes the form of a cellular automata model.

2.3.1.1 Carley (1991) group stability model

In one of the earliest cultural dynamics models, Carley (1991) modeled the spread of a set of facts through a group and the factors that influence group stability as groups absorb new individuals. The primary assumption of this model is that a pair of individuals are more likely to share information if they possess a greater amount of similarity; in particular, two individuals without completely disjoint sets of

facts will never interact. This assumption is this model's single greatest contribution to the opinion dynamics literature; *homophily* has been one of the most common assumptions in opinion dynamics models since it was encoded by Carley.

She found that in a connected graph, in which there is some direct or indirect path for communication between any two individuals, eventually all individuals share the full set of facts available to the group. However, in a disconnected graph, multiple stable groups form with unique sets of facts. Introducing a new member to this society, especially one that possesses facts from more than one of the stable groups, can cause significant turmoil in the short-term and eliminate sub-groups by further connecting the graph.

NetLogo 6.0.1 code replicating the Carley (1991) model is provided in Appendix B. N agents are initialized with a set of m facts randomly chosen from the M facts available to the group. Each tick, in random order, every agent chooses a partner (which may be themselves) from those that do not yet have a partner with probability proportional to the degree of information that they have in common. Once all pairs have been formed, each member of the pair shares one randomly selected piece of information with the other. After all information has been shared, every agent updates their set of facts. As expected, if the resultant social network is a connected graph, this model obtains perfect homogeneity.

2.3.1.2 Mark (1998) social differentiation model

Mark (1998) derived his model from that of Carley (1991) and generated a model with nearly opposite behavior. Like Carley, he assumes homophily, and his agents communicate in nearly the same manner as Carley's. The primary changes are knowledge creation and forgetting. Rather than only being capable of communicating existing information, there is a positive probability that two agents will interact and

generate a unique new fact. Additionally, agents have a limited span of memory; after a defined number of ticks, if a fact has not been communicated to or by an agent, that fact will be forgotten. With these additions, a population that is initialized to perfect homogeneity will differentiate into sub-groups that only interact among themselves.

Mark (1998) ran a 3×3 factorial experiment varying number of agents $N \in \{6, 50, 100\}$ and memory length $m \in \{3, 4, 5\}$. He found that the number of agents and memory length had primary and combined affects upon the number of sub-groups existing after 500 ticks, with the most sub-groups appearing with many agents with short memories and the fewest sub-groups appearing when agents had long memories regardless of the number of agents. He did not examine the dynamics that occur after 500 ticks.

NetLogo 6.0.1 code replicating the Mark (1998) model is provided in Appendix C. N agents are initialized with identical sets of 1 fact. Mark's model uses a different agent schedule than Carley's model did. Each tick, each agent picks a partner (which may be themselves) with probability proportional to the degree of information information that they have in common, regardless of whether that partner has already partnered with another agent. Once all partners have been chosen, the agents either share one fact randomly chosen from the set of facts available to one or both partners, or they generate a new fact. Generation of a new fact occurs with equal probability to selection of any one fact (i.e., if the set of facts available to one or both partners is of size k , each fact may be chosen or a new one generated with probability $1/(k+1)$). New facts may not be generated by an agent interacting with themselves, but a topic is communicated internally for the purpose of avoiding forgetting. Next, each agent forgets any fact that has not been communicated in the last m ticks. Contrary to the findings of Mark (1998), given sufficient time, these dynamics result in perfect social differentiation regardless of memory and number of agents; each agent eventually

forms their own sub-group. In one run with 100 agents and memory length 5, this occurred after 657,053 ticks; in another run with 6 agents and memory length 5, this occurred after 19,795 ticks. Both are far longer than the original paper’s stopping time of 500 ticks.

2.3.1.3 Axelrod (1997) cultural dissemination model

Axelrod (1997) aimed to propose a mechanism by which, despite normative influence between individuals, cultural diversity is maintained. While not directly a deviation of the Carley (1991) model, this model shares its implementation of homophily along with an assumption that different aspects of culture are not independent of one another. Agents’ cultures in this model are a fixed-length array of variables (features) which may take a limited number of values (traits). When two agents interact, with probability proportional to the cultural similarity between the two agents, the source agent influences the target agent by changing a single feature of the target’s culture to the trait matching its own.

Axelrod’s 1997 model is presented as an ABM, but it behaves as a 2-dimensional cellular automata model in which agents are influenced by their neighbors. The base model uses the von Neumann neighborhood (i.e., the four cells directly neighboring) but he also experiments with the Moore neighborhood (i.e., the eight cells including diagonal neighbors). Target agents are chosen randomly according to an asynchronous random independent schedule, and source agents are chosen randomly from the target’s neighbors. Axelrod (1997) found that a cultural majority tends to arise along with persistent minority clusters having completely dissimilar cultures. He also notes that the number of cultures decreases as the number of cultural features increases, increases with the number of traits per feature, decreases as the size of cell’s neighborhoods grows from von Neumann to Moore, and changes non-linearly with the size

of the modeling geography (increasing up to a certain size, then decreasing).

NetLogo 6.0.1 code replicating the Axelrod (1997) base model is provided in Appendix D. As the size of the geography increases, we can also observe that the size of minority cultures appears to decrease. Figure 1 shows single-run outcomes of the model with increasing numbers of cells. For a 10×10 grid, minority cultures include a 20-cell culture, a 4-cell culture, a dyad, and a singleton. The 20×20 grid includes a dyad and 4 singletons; the 100×100 grid includes only 2 singletons.

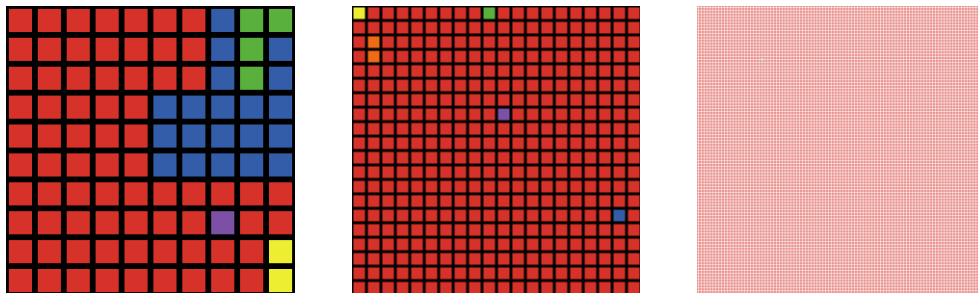


Figure 1. Ideal-typical single-run distribution of cultures at convergence of Axelrod (1997) model for 10×10 , 20×20 , and 100×100 grid geographies

2.3.2 Discrete (binary-choice) opinion dynamics models

Typically the term *discrete opinion model* refers to those models that treat opinion as a binary variable having values in either the set $\{0, 1\}$ or $\{-1, 1\}$. There exist models that are technically discrete opinion models, in that there exists a finite set of possible opinions that an agent may possess, but approximate a continuous opinion; these are usually considered continuous opinion models and are discussed in that section of the present document.

2.3.2.1 Voter model

What is now known as the *voter model* was first proposed by Clifford & Sudbury (1973) as a stochastic model of conflict between opposing species. It was first

termed the voter model by Holley & Liggett (1975) when it was applied to the realm of American political opinion, forcing agents to decide between two opposing viewpoints. The classic voter model postulates agents existing on a 2-dimensional grid. In a voter model, agents randomly choose another agent to whom they are connected (e.g., neighbors in a grid) and take on their opinion. The voter model gained popularity due to its mathematical tractability; if we restrict time steps to those in which state changes occur and consider an infinite lattice grid with equal probabilities of choosing any neighbor, the model behaves as a random walk (Clifford & Sudbury 1973). Both Clifford & Sudbury (1973) and Holley & Liggett (1975) prove probabilistic outcomes of such a model. It has since been deeply researched and modified; these are considered beyond the scope of the present paper (for an excellent review of that work, see Castellano et al. 2009).

Although the voter model was not originally proposed as a simulation but as an analytic model, NetLogo 6.0.1 code for a cellular automata simulation of the voter model is given in Appendix E. Agents are chosen to act according to an asynchronous random independent schedule. Neighbors are chosen with equal likelihood, so it behaves approximately as a random walk, although the finite grid limits this interpretation and introduces absorbing states where all agents possess the same opinion. An ideal-typical development of the opinion dynamics of the voter model as it approaches polarized homogeneity is shown in Figure 2.

2.3.2.2 Nowak-Szamrej-Latané model

One of the earliest simulation models of opinion dynamics was a cellular automata model proposed by Nowak et al. (1990) that is based heavily upon social impact theory (Latané 1981). Social impact theory is discussed in more detail in Section 2.4.2. It specifies that the impact of others' opinions upon one's own is a function

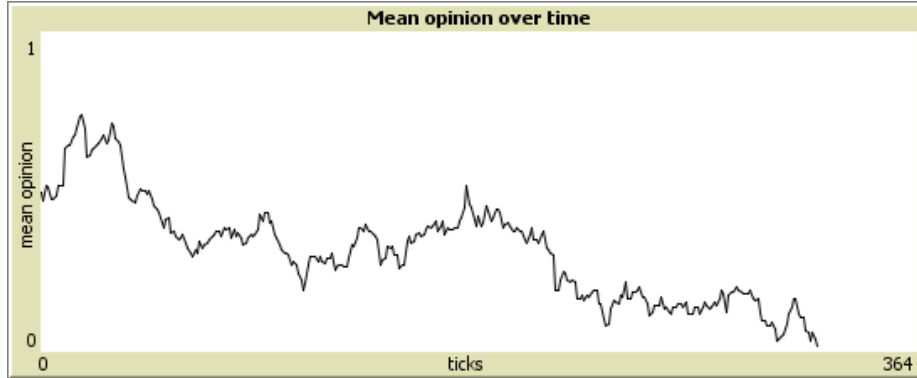


Figure 2. Approximately random walk behavior of a voter model leading to consensus

of the strength, immediacy, and quantity of others' arguments. Furthermore, others' impact diminishes as the number of influences increases according to a power law. This model uses an exponent of 0.5 for this function.

Nowak et al. introduce agents that possess an opinion ($o_i \in \{0, 1\}$), persuasiveness ($p_i \in [0, 1]$), supportiveness ($s_i \in [0, 1]$), and a location in a 2-dimensional grid of *social space* ($\{x, y\} \in \mathbb{Z}^2$). Persuasiveness is an agent's strength in changing the minds of others with dissenting opinions. Supportiveness is an agent's strength in convincing others with the same opinion not to change their opinion. Immediacy as defined by Latané (1981) is a function of the squared distance between two agents, in keeping with Latané's metaphor of a gravitational field. Persuasiveness and supportiveness are initialized randomly according to a uniform distribution over the relevant range.

All agents update their opinions synchronously at each tick (i.e., an agent's changed opinion is not communicated until the end of the tick). If the total persuasive influence of dissenting agents \hat{i}_p exceeds the total supportive influence of like-minded agents \hat{i}_s , an agent changes their opinion at the end of the tick. When an agent changes their opinion, they are randomly assigned new values for persuasiveness and supportiveness. By initializing the model with variously-sized minorities with respect to opinion, Nowak et al. (1990) found that minorities of sufficient size ($> 10\%$) were maintained but at a level lower than the initialized level; a pressure to conform existed

but did not dominate all others.

In the original paper, two formulae for impacts (and thus differing rules for cell updates) are included. The published results require that the cells change their opinion if the impact of persuasion is greater than the impact of social support, that is, $\hat{i}_p > \hat{i}_s$, where

$$\hat{i}_p = N_o^{\frac{1}{2}} \left(\frac{\sum (p_i/d_i^2)}{N_o} \right) \text{ and} \quad (1)$$

$$\hat{i}_s = N_s^{\frac{1}{2}} \left(\frac{\sum (s_i/d_i^2)}{N_s} \right). \quad (2)$$

N_o is the number of persuading (non-agreeing) cells, N_s is the number of supporting (agreeing) cells, and d_i is the distance to cell i from the cell under consideration with $\sqrt{2}$ added to avoid dividing by 0. To speed computation, the authors considered only those cells within a distance of 10.

An alternative formula for impacts is suggested later in the paper that the authors state that they came to prefer, but it was not used. Those formulae are

$$\hat{i}_p = \left(\sum ((p_i/d_i^2)^2) \right)^{\frac{1}{2}} \quad (3)$$

$$\hat{i}_s = \left(\sum ((s_i/d_i^2)^2) \right)^{\frac{1}{2}} \quad (4)$$

NetLogo 6.0.1 code that replicates the Nowak et al. (1990) model is provided in Appendix F. The results suggest that the computation-induced assumption of range limitation had an unintended effect of changing the qualitative outcome of the model. An ideal-typical result at convergence for the base model is shown in Figure 3 at left, and an unlimited-range deviation is shown at right. All runs are initialized with a randomly-placed 30% minority blue opinion. Using unlimited range, the mean persuasive effect is watered down by low-impact, long-distance cells resulting in a growth of minority opinion expanding from one corner.

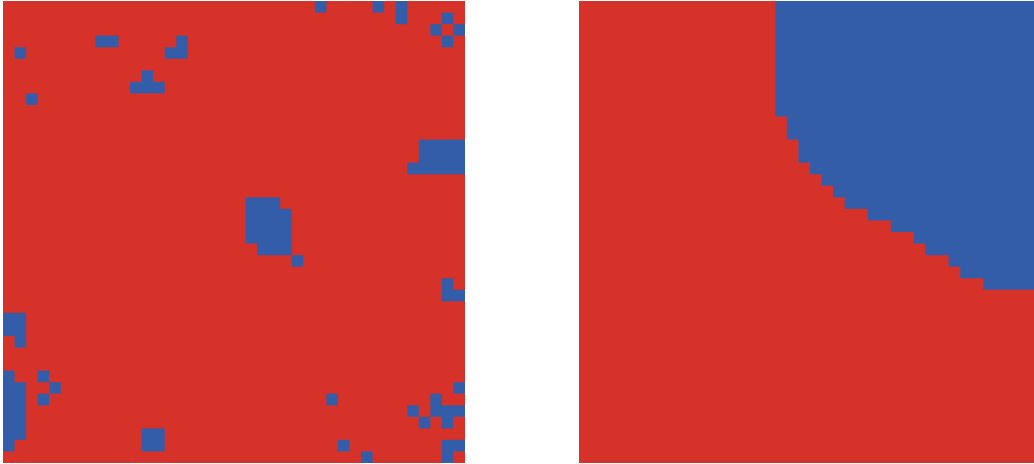


Figure 3. Outcomes of replication of Nowak-Szamrej-Latené model (Nowak et al. 1990) with limited range (left) and unlimited range (right)

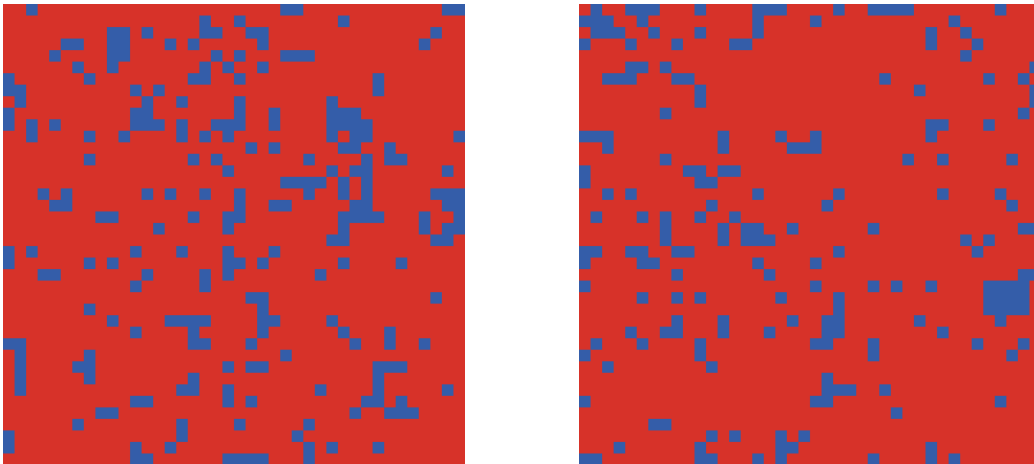


Figure 4. Outcomes of replication of Nowak-Szamrej-Latené model using alternative formulae (Nowak et al. 1990) with limited range (left) and unlimited range (right)

Using the alternative formulae given in Equations 3 and 4 results in another qualitative change model behavior. These formulae do not use a mean and thus are much more robust to range limitations as shown in Figure 4. They do not result in the same clustering effects that were the primary finding of the original paper, which find empirical support (Lewenstein et al. 1992). In fact, few cells ever switch using this formula, presumably due to the strength of influence upon oneself.

2.3.2.3 Sociophysics models of opinion dynamics

Sociophysics is a term developed in the early 1990s to describe the application of traditional physics tools and models to the problems of the social sciences (Monica & Bergenti 2017). While sociophysics models exist far outside of the realm of opinion dynamics, there is at least one influential discrete opinion dynamics model arising from sociophysics: the Sznajd model (Sznajd-Weron & Sznajd 2000; Stauffer et al. 2000; Stauffer 2002). However, sociophysics models in general have been criticized for lacking any real-world connection, or sufficient collaboration with social scientists, in their applications (Sobkowicz 2009).

Zero-temperature finite random-field Ising ferromagnetic model in an external magnetic field. The Sznajd model draws inspiration from Galam’s (1997) application of the zero-temperature finite random-field Ising ferromagnetic model in an external magnetic field to opinion dynamics. While the original article vacillates between using the agent-based mindset and a top-down systems-level mindset, the process is best described from an agent-based mindset. Each agent is assumed to be motivated to minimize total group conflict. This conflict has three components: interaction conflict (generated by disagreements among individuals), external social field conflict (generated by external pressures in one direction), and internal social field conflict (generated by disagreement between one’s opinion and

their internal pressures, i.e., cognitive dissonance). In other words, agents seek to maximize an objective function

$$G = I \cdot \sum_{i,j} c_i c_j + S \sum_{i=1}^N c_i + \sum_{i=1}^N S_i c_i \quad (5)$$

where N is the number of agents, I is the relative strength of interaction conflict, S is the strength and direction of the external social field, S_i is the strength and direction of an individual's internal social field, and $c_i \in \{-1, 1\}$ is individual i 's binary opinion.

It is unclear how individual agents implement this global maximization; the maximization of G is assumed to occur at a macro level without specifying an implementation at the micro level. For example, it is unclear in what order each agent should act to maximize the objective function at a given time. This is a weakness of the model that deserves some scrutiny.

NetLogo 6.0.1 code that implements the Galam (1997) model as a cellular automata model is provided in Appendix G. This model exists as a 33×33 grid of agents that, when they act, choose the state c_i that maximizes G as given in Equation 5. Agents' initial values for c_i are chosen at random from $\{-1, 1\}$, values for the internal social fields S_i are chosen randomly from the uniform distribution in $[0, 1]$, and the external social field is set to $S = 0$.

An experiment was conducted varying the agent schedule between synchronous, asynchronous random order, and asynchronous random independent using synchronized random number streams to force initialization states to be identical across 1000 replicates. Although random order and synchronous updates yielded nearly identical outcomes, as shown in Figure 5, the final objective function value G and final consensus opinion's magnitude $|C| = \sum_{i=1}^N c_i / N$ can vary between random independent and random order schedules. On average, the random independent schedule yielded

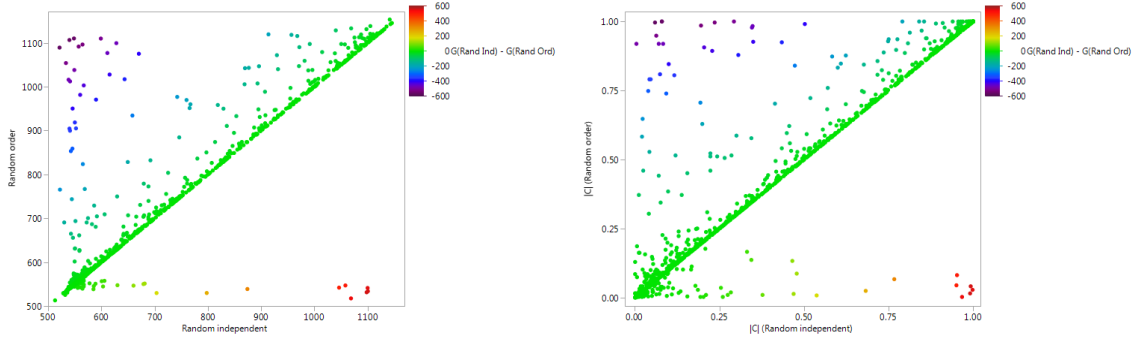


Figure 5. Paired outcomes of G (left) and $|C|$ (right) from Galam (1997) model using random order and random independent schedules

a value of G 16.6 higher (less conflict) and a value of $|C|$ 0.026 lower (less polarized) than the random order schedule. Since neither schedule dominates the other in terms of G , this shows that an agent-based implementation of the model does not always find the optimal G . As such, there is so reason to believe that an actual social system of people should be expected to find the optimal G .

Sznajd 1-dimensional model. Sznajd-Weron & Sznajd (2000) first proposed a far less complicated Ising field model of opinion dynamics as a 1-dimensional cellular automata model. What they called the *United we Stand, Divided we Fall (USDF)* ruleset implements both ferromagnetic and anti-ferromagnetic rules. A random pair of neighboring agents are chosen to influence their neighbors. The ferromagnetic rule stipulates that if both members of this pair have the same “Ising spin,” or opinion, then their neighbors will also take on this spin. The anti-ferromagnetic rule stipulates that if the pair has opposing spins, their neighbors will take on spins opposite them. There are three absorbing states of this model: all agents have positive spin, all agents have negative spin, or agents take on alternating spins. Random noise can be added (positive temperature in a physics sense) to overcome the stability of the system in these configurations. NetLogo 6.0.1 code replicating the Sznajd-Weron & Sznajd (2000) model, along with deviations to implement synchronous and asyn-

chronous random order agent schedules, is provided in Appendix H.

Sznajd 2-dimensional model. Stauffer et al. (2000) examined deviations from the original Sznajd model into 2 dimensions and experimented with several rulesets, finally settling upon the rules explained in Stauffer (2002). This is the model most commonly referred to as simply the Sznajd model (Castellano et al. 2009), and it uses only the *United we Stand* portion of the ruleset from Sznajd-Weron & Sznajd (2000). A random pair of neighboring agents are chosen to influence their neighbors. If both members of this pair have the same spin, they impart this spin on all 6 of their neighboring agents. Otherwise, no state change occurs. This eliminates the absorbing state in which agents have alternating spins; only total homogeneity results from this model, although the path by which consensus is reached may be of interest. NetLogo 6.0.1 code replicating the 2-dimensional Sznajd (Stauffer 2002) model, along with deviations to implement synchronous and asynchronous random order agent schedules, is provided in Appendix I.

2.3.3 Continuous opinion dynamics models

Continuous opinion dynamics models trace their origins to the work of DeGroot (1974), with a model commonly known as the *repeated averaging model*. This is also an early formulation of *social influence network theory* (Friedkin 1999). In this formulation, there are N agents with opinions $F_i, i = 1, \dots, N$. Each time step, agent i calculates its new opinion as a convex combination of the opinions of all N agents with weights $P_{ij}, j = 1, \dots, N$. In matrix form,

$$\mathbf{F}(t + 1) = \mathbf{F}(t)\mathbf{P} \tag{6}$$

where \mathbf{P} is an $N \times N$ stochastic matrix filled with influence weights and $\mathbf{F}(t)$ is the $1 \times N$ vector of agent opinions at time t . This is identical in formulation to a discrete-time Markov chain, and thus many properties can be derived similarly. In particular, if the associated Markov chain is ergodic (aperiodic and positive recurrent), then the system will reach consensus (DeGroot 1974). There are additional conditions when a periodic system will also reach consensus (R. L. Berger 1981). Furthermore, if consensus will eventually be reached, the consensus set of opinions $\boldsymbol{\pi}$ can be found by solving the system $\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{P}$.

This finding, while mathematically satisfying, underlies one of the most significant challenges in the use of continuous opinion dynamics models: the curse of monoculture. DeGroot (1974) assumes that only *positive influence* exists; that is, the weights given to the influence provided by others is strictly non-negative. Under this assumption, which is examined from a social psychological perspective in Section 2.4.2, unless a system is divided into non-communicating classes the system tends to converge to a single opinion. Continuous opinion dynamics models after DeGroot, then, try to explain mechanisms by which diversity may be maintained in light of this tendency toward convergence.

2.3.3.1 Bounded confidence models

One of the earliest and most influential approaches to generating sustained diversity is the use of bounded confidence. *Bounded confidence (BC)* models assume that two individuals can only influence one another if the absolute difference in their opinions is below a specified value. A confident individual would have a lower bound; they would be susceptible to influence by others with only slightly differing opinions. Less confident individuals would be susceptible to a wider range of influences. This serves to break the system into recurrent classes that do not communicate, thereby

making the associated Markov chain non-ergodic. Depending on the implementation, this confidence may be either homogeneous or heterogeneous, and it may remain static or change over time. In general, BC models are characterized by generation of *weak diversity*, which Duggins (2017) defines as “the convergence of opinions to a finite number of attractor states,” rather than *strong diversity*, which he defines as “a smooth distribution of opinions along a continuous ideological spectrum.”

Bounded confidence models can be split into two types: those based upon the Deffuant-Weisbuch model and those based upon the Hegselmann-Krause model. The Deffuant-Weisbuch model is the most well-studied bounded confidence opinion dynamics model (Castellano et al. 2009), as can be seen from the number of models derived thereof. Both this model and the Hegselmann-Krause model remain influential, largely as inputs to larger behavior models or as bases for comparison for newer opinion dynamics model.

Deffuant-Weisbuch model. Deffuant et al. (2000) first implemented bounded confidence as an agent-based model. Agents are randomly assigned values between 0 and 1 for their opinions, and they interact such that both individuals converge by an amount equal to $\mu \in (0, 0.5]$ times the difference between them. Mathematically, given agents i and j with opinions x_i and x_j ,

$$x_i \leftarrow x_i + \mu \cdot (x_j - x_i) \quad \text{if } |x_i - x_j| < d_i \quad (7)$$

$$x_j \leftarrow x_j + \mu \cdot (x_i - x_j) \quad \text{if } |x_i - x_j| < d_j \quad (8)$$

where d_i is the confidence bound for agent i . Their opinions can then be considered the 1-dimensional geography upon which agents move. The parameter μ in the Deffuant et al. (2000) model is approximately equivalent to P_{ij} in the DeGroot (1974) model with the following changes: (1) interactions are pair-wise, (2) $P_{ij} = 0$ if $|x_i - x_j| > d_j$,

and (3) P_{ij} is otherwise equal for all i, j . Random pairs of agents are chosen to interact at each step using a random independent schedule. Smaller values of d result in greater numbers of stable groups of opinions, while higher values of d result in convergence. The maximum number of stable groups is approximately $1/(2d)$. Within stable groups, opinions converge toward a single value (i.e., weak diversity is generated). NetLogo 6.0.1 code replicating the base model of Deffuant et al. (2000) is provided in Appendix J. Ideal-typical results of a model run with $d = 0.25, \mu = 0.5$ are shown in Figure 6; note that a single agent (what the authors call a *wing*, which is disregarded as a group in their results) remains outside of the main groups with opinion 1.

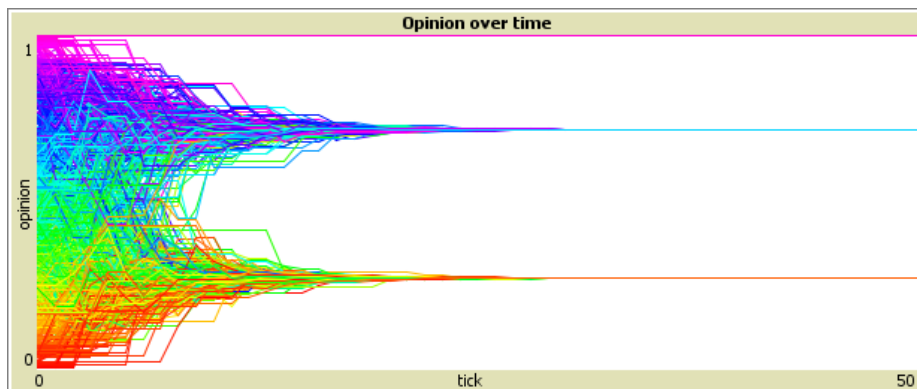


Figure 6. Ideal-typical results of base Deffuant-Weisbuch model converging to 2 groups and 1 wing, with $d = 0.25, \mu = 0.5$

The authors also consider two variants of their model in the original paper: social networks and vector opinions. Social networks are implemented by way of a 2-dimensional cellular automata geography. Only neighbors are considered for interaction rather than all other agents. This type of social network implementation, which is quite common, has been criticized for being unrealistic (Sobkowicz 2009). Additionally, Deffuant et al. (2000) note that this does not significantly alter the results.

The vector opinion implementation is similar in function to a gene in a genetic

algorithm. Opinion is set as an m -length vector of bits that represent binary opinions on specific issues. Agents must have a Hamming distance (number of unequal bits) no greater than $d \in \mathbb{Z}$ in order to interact. If they interact, equal bits are kept while each agent in the pair randomly selects a value for unequal bits. As in the base model, higher values of d yield consensus while lower values of d generate greater numbers of opinion clusters (Deffuant et al. 2000). Both variations generate weak diversity as well, and they have not been as influential as the base model.

Additional analysis of this model is performed by most of the same authors in a later work (Weisbuch et al. 2002). The two additions in this work involve heterogeneous assignment of d and values for d that decrease with the number of interactions. Heterogeneous assignment between two possible values of d appears to maintain the same number of clusters, in the long run, as predicted by the largest values of d and, in the short run, as predicted by the smallest values of d . Dynamically increasing confidence (decreasing d) with the number of interactions leads to slightly different clustering dynamics. For example, when 2 clusters form, they tend to be slightly more moderate clusters than would occur otherwise. Additionally, few wings are generated outside of clusters in this implementation.

Later work by Lorenz (2010) more deeply examines the dynamics of the Deffuant-Weisbuch model using heterogeneous confidence bounds. He finds that open-minded agents have the capacity to draw closed-minded agents from one cluster into another cluster, generating greater consensus and introducing more polarized average opinions in the population. Furthermore, he finds that the number of clusters decreases in this condition, even finding consensus despite all agents have confidence bounds smaller than that which would yield clustering in the base model.

Derivatives of the Deffuant-Weisbuch model. One fairly straightforward modification of the Deffuant-Weisbuch model is the *social judgment* model of

Jager & Amblard (2005). Whereas the Deffuant-Weisbuch model only allows positive influence between agents within bounds of confidence, the social judgment model adds negative influence between agents beyond further bounds of confidence. *Negative influence* assumes that an agent’s influence upon another agent that is sufficiently dissimilar will be in the opposite direction (i.e., after interaction their opinions will be more distant from one another). Put mathematically, given an acceptance threshold value of u_i and a rejection threshold of $t_i > u_i$, the change in an agent’s opinion as a result of interaction

$$\Delta x_i = \begin{cases} \mu \cdot (x_j - x_i) & \text{if } |x_i - x_j| < u_i \\ \mu \cdot (x_i - x_j) & \text{if } |x_i - x_j| > t_i \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

The results are similar to those of the Deffuant-Weisbuch model; varying the parameters can yield consensus formation as a population or consensus within a set of clusters.

A second modification of the Deffuant-Weisbuch model uses Bayesian updates of opinions and uncertainties by observing the actions of others. Martins et al. (2009) introduces this deviation, called the Continuous Opinion and Discrete Actions (CODA) model, and Martins (2009) further examines the model. CODA assumes that agents exhibit action (i.e., communicate an opinion) with valence equal to the rounded value of their opinion. Others observe this action and update their opinion (i.e., first moment) and, in some versions, uncertainty (i.e., second moment). Updating only the first moment results in behavior very similar to the base Deffuant-Weisbuch model, while adding updates of the second moment yields unimodal moderate opinions with some variance (Martins 2009). In other words, CODA either yields many clusters with no variance within clusters or one cluster with variance within that cluster;

unfortunately, it is not capable of yielding many clusters with variance (i.e., strong diversity).

Sun & Müller (2013) introduce another modification of the Deffuant-Weisbuch model embedded in a behavior model of land use decision-making in a region of China. This implements more realistic, small-world social networks among agents, heterogeneous convergence (μ) values, and confidence bounds that vary according to an agent’s opinion. Small-world networks, introduced by Milgram (1967), are characterized by minimally interconnected, small clusters of individuals such that the number of links that must be traversed to connect any two individuals is relatively small. Agents are initialized with μ drawn from a truncated normal distribution with mean 0.5, standard deviation 0.2, and bounded within the range $[0, 1]$. Additionally, confidence of an agent at any point in time can be calculated from the agent’s opinion:

$$d_i = 1 - 2 \cdot |x_i - 0.5| \tag{10}$$

where d_i is agent i ’s confidence bound and x_i is agent i ’s opinion.

These minor changes result in some interesting behavior that, by itself, has not yet been examined in any detail. This lack of investigation is likely due to its being embedded in a larger behavior model. A fairly shallow examination of possibilities shows that this can be an unpredictable model: utilizing the same input parameters, the three clusters may converge to a moderate opinion, an extreme opinion, or different and opposing extreme opinions. See Figure 7 for examples of all three behaviors. Varying the number of within-group and between-group social network connections or the distribution from which μ is drawn would likely have significant impacts upon the convergence behavior.

Hegselmann-Krause model. Hegselmann & Krause (2002) proposed a

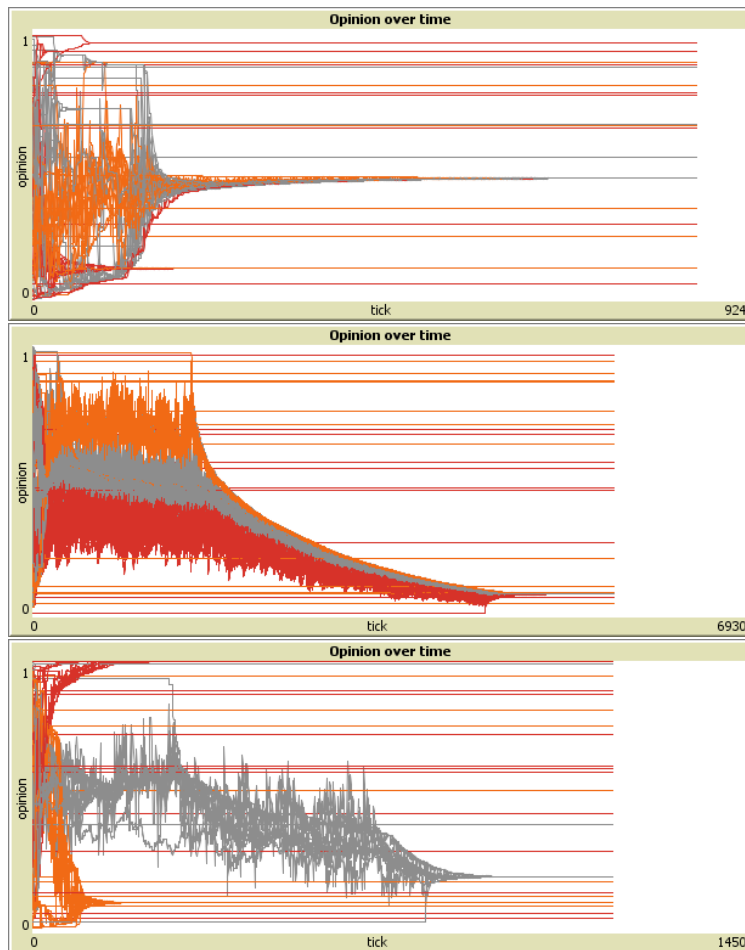


Figure 7. Three instances of output from a model derived from Sun & Müller (2013), yielding convergence to moderate opinion (top), convergence to an extreme opinion (middle), and within-group convergence to opposing extreme opinions (bottom)

bounded confidence opinion dynamics model that varies from the Deffuant-Weisbuch model primarily in the agents' schedules. In the Deffuant-Weisbuch model, at each step one random agent is selected to be influenced by one random agent within its bounds of confidence. In the Hegselmann-Krause model, at each step all agents take the mean opinion of all agents (including themselves) within their bounds of confidence, effectively giving equal weight to all sufficiently similar opinions including their own. Mathematically, given agent i with opinion x_i among N agents:

$$x_i \leftarrow |I(i, x_i)|^{-1} \cdot \sum_{j \in I(i, x_i)} x_j \quad (11)$$

$$I = \{j \in 1, \dots, N : -d \leq x_i - x_j \leq d\} \quad (12)$$

This corresponds with a synchronous schedule in selection of both source and target agents. This yields a much smoother convergence to fewer clusters as compared to the Deffuant-Weisbuch model. The Hegselmann-Krause model is equivalent to the DeGroot (1974) model where \mathbf{P} is redefined each step such that

$$P_{ij} = \begin{cases} |I(j, x_j)|^{-1} & \text{if } |x_i - x_j| < d \\ 0 & \text{otherwise} \end{cases}$$

The NetLogo 6.0.1 model given in Appendix L implements the Hegselmann-Krause model. An ideal-typical result with $d = 0.25$, analogous to that of Figure 6, is shown in Figure 8, along with an ideal-typical result with $d = 0.20$, which behaves more like the Deffuant-Weisbuch model with $d = 0.25$. Note that, when $d = 0.25$, a small group of agents with moderate opinions is caught between the influence of opposing groups and slowly moderates both of the larger groups. In the Deffuant-Weisbuch model, this moderate group would have randomly become a part of one of the larger

groups.

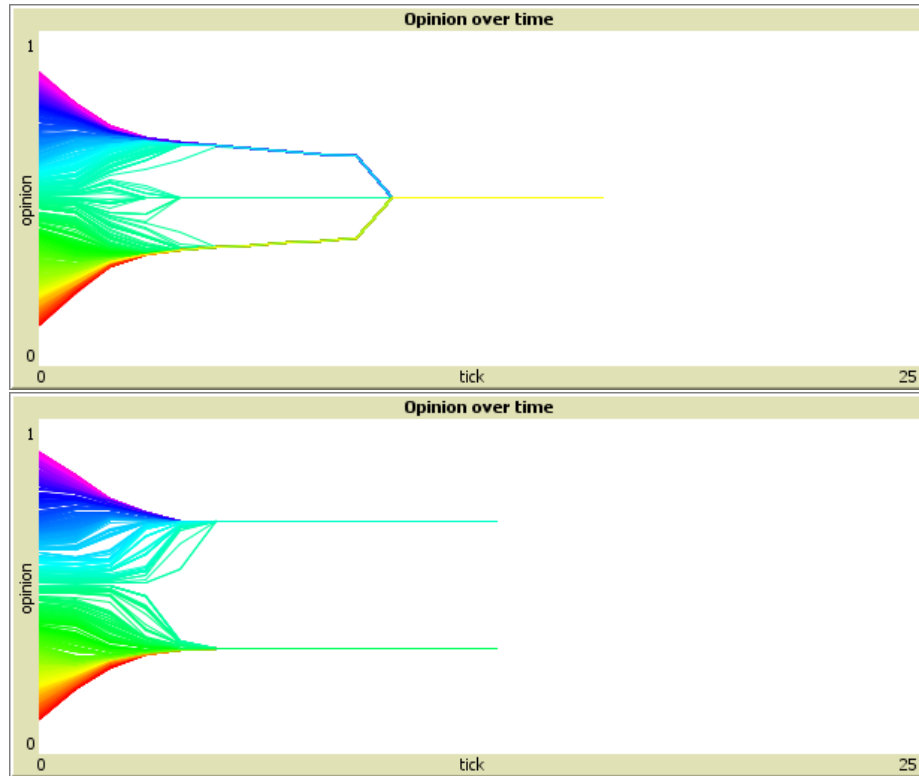


Figure 8. Ideal-typical results of base Hegselmann-Krause model converging to one group ($d = 0.25$, top) and two groups ($d = 0.20$, bottom)

Hegselmann & Krause (2002) also experiment with asymmetric bounds of confidence and find that asymmetric bounds of confidence yield asymmetric convergence behavior. That is, agents that are more open to dissenting opinions in one direction than another tend to move toward the opinion they are more open to.

Lorenz (2010) considered heterogeneous confidence bounds for the Hegselmann-Krause model as they did with the Deffuant-Weisbuch model and found that the same behavior applies to both. Heterogeneity of confidence bounds increases the likelihood of convergence to a consensus opinion, even when all confidence bounds are smaller than that which would be expected to yield consensus in the base model.

2.3.3.2 Alternatives to bounded confidence

While bounded confidence models are the basis for comparison for nearly every newer technique, many alternative models have been suggested that make up for shortcomings of the bounded confidence technique. What follows is a review of the best alternatives that have been proposed, though it is not exhaustive.

Relative agreement model. The relative agreement model was proposed by Deffuant et al. (2002) as an extension to the Deffuant-Weisbuch bounded confidence model in order to explain how initially extremist viewpoints (e.g., Nazism or fashion trends initially far outside of the norm) can become mainstream in a society. Whereas the bounded confidence model uses static confidence and dynamic opinion, the relative agreement model allows both an agent’s opinion and confidence to be dynamic as a result of social interactions. In this formulation, agent i ’s state is defined by both their opinion $x_i \in [-1, 1]$ and their uncertainty $u_i \in [0, 2]$. This state can be visualized as a line segment covering the range $[x_i - u_i, x_i + u_i]$.

As in the Deffuant-Weisbuch model, agents are chosen randomly to influence another random agent. Upon agent j attempting to influence agent i , agent i updates its opinion and uncertainty according to the following rules:

$$h_{ij} = \min(x_i + u_i, x_j + u_j) - \max(x_i - u_i, x_j - u_j) \quad (13)$$

$$x_i \leftarrow x_i + \mu \cdot \left(\frac{h_{ij}}{u_j} - 1 \right) \cdot (x_j - x_i) \quad (14)$$

$$u_i \leftarrow u_i + \mu \cdot \left(\frac{h_{ij}}{u_j} - 1 \right) \cdot (u_j - u_i) \quad (15)$$

In this way, the amount of influence agent j has upon agent i increases with the ratio of the line segment overlap h_{ij} to its uncertainty u_j . Thus confident agents are more influential and uncertain agents are more heavily influenced by others. Further-

more, agents whose line segments do not overlap cannot influence one another, which generates bounded confidence-like behavior.

In the base model, where all agents have identical initial uncertainty (and therefore, their uncertainty does not change with time), the relative agreement model behaves almost identically to the bounded confidence model, with the number of resultant clusters varying approximately with $\frac{1}{u}$. Adding high certainty ($u_i \approx 0$), extremist ($|x_i| \approx 1$) agents has the effect of polarizing an initially uniformly distributed population, fulfilling the desideratum of their modeling effort. NetLogo 6.0.1 code replicating this model is provided in Appendix M.

Meta-contrast model. The meta-contrast model was proposed by Salzarulo (2006) to implement interaction following the predictions of self-categorization theory (Turner et al. 1987), which is discussed in Section 2.4.1. Self-categorization theory predicts that an individual will associate with an in-group with higher probability if that group is cohesive and other individuals' opinions are separated from the in-group. This is measured by the difference between opinions within that group compared to the difference between that group and individuals outside of that group. Therefore, in a very diverse group, two relatively different individuals may become part of the same in-group; in a less diverse group, these same individuals might associate with different in-groups. Salzarulo implements this by first introducing the fuzzy membership function:

$$\mu(x, x_i) = e^{-\frac{(x-x_i)^2}{w^2}} \quad (16)$$

where $w \in [0, 1]$ is a parameter associated with the typical group width; higher values are expected to result in fewer groups while lower values are expected to result in more groups. Equation 16 is used to calculate the intra-category distance and the

inter-category distance for an opinion $x \in [0, 1]$ to the set of n agent opinions X :

$$d_{intra}(x, X) = \frac{\sum_{i=1}^n (\mu(x, x_i) \cdot (x - x_i)^2)}{\sum_{i=1}^n \mu(x, x_i)} \quad (17)$$

$$d_{inter}(x, X) = \frac{\sum_{i=1}^n ((1 - \mu(x, x_i)) \cdot (x - x_i)^2)}{\sum_{i=1}^n (1 - \mu(x, x_i))} \quad (18)$$

These distances in turn are used in the calculation of the prototypicality function of an opinion x to the set of n agent opinions X :

$$P(x, X) = a \cdot d_{inter}(x, X) - (1 - a) \cdot d_{intra}(x, X) \quad (19)$$

where a is another parameter that defines out-group aversion; higher values are expected to result in more groups while lower values are expected to result in fewer groups.

Agents update their opinions according to an asynchronous random independent schedule. An agent i updates their opinion by first finding the local maxima of the prototypicality function $P(x, X)$ where X is all agents to whom agent i is connected. Agent i interprets the local maximum nearest their own opinion as the prototypical opinion of their in-group. Agent i then finds agent j , the agent with opinion nearest the prototypical opinion, and adopts agent j 's opinion as their own. In the base model, agents are fully connected. NetLogo 6.0.1 code implementing the fully-connected model from Salzarulo (2006) is provided in Appendix N. The original paper also implements small-world social networks between agents that define the interactions. NetLogo 6.0.1 code implementing the small-world networks model from Salzarulo (2006) is provided in Appendix O.

Salzarulo (2006) explores the parameter space defined by $a \in [0, 0.3]$ and $w \in [0, 1]$ and find that, in the fully connected model, different parameters yield consensus (low a and high w) or fractioning into 2 or many groups (low a and low w). The results are similar for the small-world model, but the focus there is more on variance in opinions; the highest opinion variance is found in the parameter space that yields 2 groups in the fully connected model. The clustering in the small-world model is also spatially observable but less distinct than in other models explored thus far. See Figure 9 for an example of this.

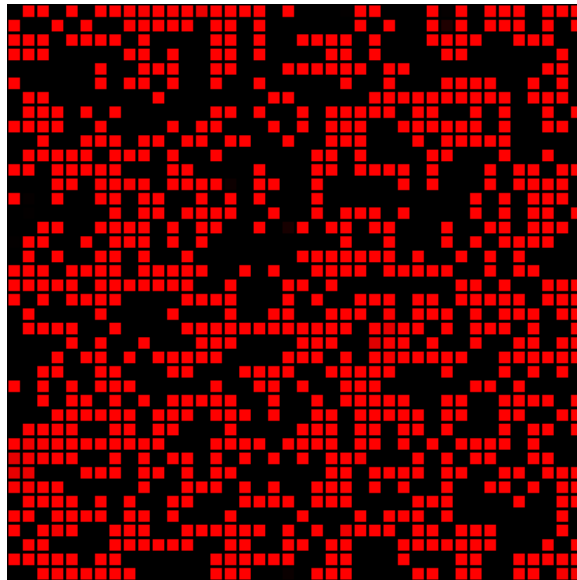


Figure 9. Example outcome of meta-contrast model with $a = 0.08$ and $w = 0.36$, using a small-world network

The meta-contrast model differs from other models in several important ways. First, it is based upon experimentally-grounded social psychological research. Second, it generates the effect of negative influence (two agents may move away from one another because of each other's presence) without actually implementing negative influence. Lastly, it can generate lasting diversity of opinions without relying upon disconnecting the social network.

Individuation by adaptive noise. Attempts to generate clustering from continuous opinion dynamics models by using random noise have been ineffective (Mäs et al. 2010). However, Mäs et al. (2010) introduced a model that uses noise in an adaptive capacity more rooted in sociological theory. The basic model is similar to the Hegselmann-Krause model, but instead of using a step function, influence decreases exponentially as the difference in opinion increases. Thus influence is homophilous; individuals with similar opinions have a greater effect than those with dissimilar opinions. The major addition is an adaptive noise component $\xi_i(t)$ that becomes stronger when there are many others with similar opinion and weaker otherwise; this is intended as a drive for individuation.

The mathematical form of the Mäs et al. (2010) model follows. Given a group of N agents, the opinion of agent i , $o_i \in [-1, 1]$, changes at each turn by

$$\Delta o_i = \frac{\sum_{\substack{j=1 \\ j \neq i}}^N (o_j(t) - o_i(t)) w_{ij}(t)}{\sum_{\substack{j=1 \\ j \neq i}}^N w_{ij}(t)} + \xi_i(t) \quad (20)$$

where the weight of influence between agents i and j , w_{ij} is defined by their opinions and the model parameter A where $A \geq 0$:

$$w_{ij}(t) = e^{-\frac{|o_j(t) - o_i(t)|}{A}} \quad (21)$$

High values of A represent openness and allow stronger influence by dissimilar others; low values represent confidence and allow little influence by dissimilar others. The noise component $\xi_i(t)$ is a random variate drawn from the normal distribution with

mean 0 and standard deviation $\theta_i(t)$, calculated as

$$\theta_i(t) = s \cdot \sum_{j=1}^N e^{-d_{ij}(t)} \quad (22)$$

where s is a model parameter representing the strength of individuation. In order to keep opinions within the defined bounds, the noise component is ignored if it would cause an opinion to go beyond the bounds. Agents are homogeneous in this model with respect to parameters A and s . Interestingly, although an agent is simultaneously influenced by all agents when it is updated, the agents are chosen to be influenced according to a random independent schedule. This makes the scheduling a hybrid of that used in the Deffuant-Weisbuch model and that used in the Hegselmann-Krause model; it is synchronous with respect to source agents and asynchronous random independent with respect to target agents.

Given appropriate parameterization, this model can cause opinions to converge to a single cluster, to maintain an approximately uniform distribution, or to vary between periods of multiple clusters and periods of approximate convergence. This is an appealing model due to its flexibility and sociological foundations. It is also interesting in that it maintains diversity without negative influence in a fully connected social network, even given starting conditions of perfect agreement.

A later work by Mäs et al. (2014) modifies the 2010 model by adding negative influence. This has the effect of making the clustering more predictable and stable at the cost of introducing negative influence. With negative influence, the distribution of opinions at initialization ceases to have an effect upon the number of clusters that emerge or the polarization present in those clusters.

Persuasive arguments. Persuasive argument theory, despite its popularity in the social psychology literature, is surprisingly underused in the opinion dynamics

modeling world. The one notable exception is provided by Mäs & Flache (2013). They call their model based on persuasive argument theory the argument-communication theory of bi-polarization (ACTB). The ACTB model is populated with N agents with opinions $o_i \in [-1, 1]$. Their world also contains P arguments in support of an opinion (pro arguments) and C arguments against an opinion (con arguments), making a pool of arguments l for which $c_l = 1$ for pro arguments and $c_l = -1$ for con arguments. Each agent stores $S_{i,t}$ distinct arguments in memory at each time step and holds the opinion

$$o_i = \frac{1}{S_{i,t}} \cdot \sum_{l=1}^{S_{i,t}} c_l r_{i,t,l} \quad (23)$$

where $r_{i,t,l} = 1$ if argument l is in agent i 's memory at time t , and $r_{i,t,l} = 0$ otherwise. This is not technically a continuous opinion, as only $(S_{i,t} + 1)$ opinions are possible. However, the article contends that these are continuous opinions, and they approximate continuous opinions, so they are included here in that section.

Agents are chosen to be influenced according to an asynchronous random independent schedule. Agent i is chosen to be influenced, and agent j is chosen from the population according to a multinomial distribution with probability proportional to the pair-wise similarity of opinions, calculated as

$$sim_{i,j,t} = \frac{1}{2} (2 - |o_{i,t} - o_{j,t}|) \quad (24)$$

such that $sim_{i,j,t} = 1$ when two agents have identical opinions and $sim_{i,j,t} = 0$ when two agents have opinions at opposite ends of the spectrum. An agent j has probability of being selected

$$p_{j,t} = \frac{(sim_{i,j,t})^h}{\sum_{\substack{k=1 \\ k \neq i}}^N (sim_{i,k,t})^h} \quad (25)$$

where h is a model parameter defining the strength of homophily. It is worth noting

that an agent has the highest probability of being influenced by another agent with the same opinion. If agent i has a polarized opinion ($|o_i| = 1$), then agent j with the opposite polarized opinion ($o_j = -o_i$) has zero probability of influencing them.

Source agent j then chooses one of the arguments in its memory to communicate to target agent i . If agent i already has that argument in memory, it becomes the most recent argument in agent i 's memory and no arguments are forgotten. If agent i does not have that argument in memory, it becomes the most recent argument in agent i 's memory and the least recent argument in memory is forgotten.

NetLogo 6.0.1 code implementing the model from Mäs & Flache (2013) is provided in Appendix P. Due to the implementation of homophily and the dwindling of the arguments in agents' memories over time, there are three convergence conditions. If all agents possess the same arguments in memory (regardless of order), the system has converged to consensus. If all agents possess polarized opinions ($|o_i| = 1, \forall i$), the system has converged to either consensus (if all agents possess the same opinion) or bi-polarization (if agents possess opinions in both poles).

ISC model. A desideratum of opinion dynamics modelers that has long eluded them has been the generation of strong diversity of continuous opinions at convergence. Recently, Duggins (2017) achieved that goal with the Influence, Susceptibility, and Conformity (ISC) model. This aims to be a more cognitively complex and realistic model of opinion dynamics by including heterogeneity among spatially distributed agents. Each agent i is initialized with an opinion $o_i \in [0, 100]$, intolerance $t_i \in [0, \infty)$, susceptibility $s_i \in [0, \infty)$, conformity $c_i \in (-\infty, \infty)$, and social range $r_i \in [0, \infty)^2$. Each value is drawn from a random normal distribution with pa-

²In the Python 2.7 simulation code provided in Duggins (2017), there was a typo in the code that set susceptibility to the conformity value. I informed the author of the typo, and he confirmed it was a mistake. Based on some informal testing, it appears not to change the qualitative performance of the model, but the author intends to follow up on it.

rameters varying as model parameters, and values falling outside of the bounds take the boundary values. These agents are placed at random (x, y) -coordinates within a 316×316 grid. The agents then form a social network by pairing every agent whose distance is within both members' social reach. At each time step, agents update their opinions in random order.

To update their opinion, agent i starts a discussion with the set of their immediate neighbors J in the social network by expressing their true opinion o_i . In random order, each neighbor j expresses an opinion that is tempered by the set of opinions D that have already been expressed in this discussion, expressing the value

$$e_j = o_j + \frac{c_j}{k_j} * \frac{1}{|D|} \sum_{o_k \in D} (o_k - o_j) \quad (26)$$

where k_j is a value for commitment calculated as

$$k_j = 1 + s_j \cdot \frac{|50 - o_j|}{50} \quad (27)$$

such that commitment increases with susceptibility and polarization of opinion. Influence is then calculated from the list of opinions expressed by others as

$$I_i = \frac{\sum_{j \in J} w_{ij} \cdot (e_j - o_i)}{\sum_{j \in J} |w_{ij}|} \quad (28)$$

where w_{ij} is the weight given that paired interaction, calculated as

$$w_{ij} = 1 - t_i \cdot \frac{e_j - o_i}{50} \quad (29)$$

NetLogo 6.0.1 code replicating the Duggins (2017) model is given in Appendix Q, with the noted bug corrected.

This model is capable of generating strong diversity in various distributions. The

original paper gives examples of bimodal distributions that span the opinion spectrum, unimodal distributions centered on moderate opinions, and collapse to polarized consensus. Furthermore, he is able to replicate the distributions of several opinions from real-world polls of political opinions. It is also fairly unique among opinion dynamics in that it uses a random order schedule.

2.3.4 Other modeling techniques that could inform new opinion dynamics models

Social influence network theory models. Social influence network theory was first mathematically modeled by DeGroot (1974) with the repeated averaging model, as discussed above. An expansion was proposed by Friedkin (1999) as a way to integrate social comparison theory, persuasive argument theory, self-categorization theory, and social decision scheme theory within the sociological (group-level) perspective rather than the psychological (individual-level) perspective. According to social influence network theory, opinions flow through a population as a function of pair-wise influence weights and susceptibility to influence of each individual. Mathematically, N agents update their opinions at time $t = 2, 3, \dots$ according to the following:

$$\mathbf{y}^{(t)} = \mathbf{A}\mathbf{W}\mathbf{y}^{(t-1)} + (\mathbf{I} - \mathbf{A})\mathbf{y}^{(1)} \quad (30)$$

where $\mathbf{y}^{(t)}$ is an $N \times 1$ vector of individual opinions at time t , \mathbf{A} is an $N \times N$ diagonal matrix defining each individual's susceptibility to influence, and \mathbf{W} is the $N \times N$ stochastic matrix of pair-wise influence weights. As this type of model implies synchronous updates of all individuals' opinions, this is also deterministic in nature. Thus we can find the consensus opinions,

$$\mathbf{y}^{(\infty)} = \mathbf{A}\mathbf{W}\mathbf{y}^{(\infty)} + (\mathbf{I} - \mathbf{A})\mathbf{y}^{(1)} = \mathbf{V}\mathbf{y}^{(1)} \quad (31)$$

where \mathbf{V} is a stochastic matrix. If $(\mathbf{I} - \mathbf{AW})$ is nonsingular, $\mathbf{V} = (\mathbf{I} - \mathbf{AW})^{-1}(\mathbf{I} - \mathbf{A})$.

If we assume that an individual's susceptibility to influence is really the weight that an individual gives to their own opinion (i.e., $w_{ii} = 1 - a_{ii}$), then the formulation can be simplified using an $N \times N$ stochastic matrix \mathbf{C} , $c_{ii} = 0, \forall i$. We can then set $\mathbf{W} = \mathbf{AC} + \mathbf{I} - \mathbf{A}$; this has the impact of distributing the weight remaining to other individuals a_{ii} proportionally according to the original \mathbf{W} matrix and simplifies the parameter space. Friedkin (1999) originally uses this simplification but omits it in later work (Friedkin 2001). This model has been used in at least one behavior model (Pires & Crooks 2017).

The primary difference between this and the much earlier DeGroot (1974) model is that in this formulation, for perpetuity, an individual's original opinion continues to strongly influence their later opinions. This concept has also been applied in a more agent-based method to a modified DeGroot model by Dandekar et al. (2013). Dandekar et al. (2013) use other methods to implement this concept, which they call *biased assimilation*, including PageRank algorithms. They find that these algorithms lead to polarization where unbiased methods do not and suggest that this methodology can be used to study the effects of personalized recommendation algorithms (e.g., Google searches, Facebook content) on societal polarization, a topic known as the *filter bubble* (Pariser 2011) that is of interest for the third paper.

Vector opinions and demographic faultlines. The use of vector opinions was briefly explored by Deffuant et al. (2000), but it seems to have gotten very little traction. Similarly, Salzarulo (2006) devotes a short paragraph (2.3) to discussing how his model could be easily modified to implement vector opinions but states that this is not the purpose of the paper. C. Weimer et al. (2013) used vector opinions of a form similar to that used by Deffuant et al. (2000) that allow communication to be similar to genetic recombination.

In the field of group stability modeling, there has been more use of discrete vector opinions. Flache & Mäs (2008a) and Flache & Mäs (2008b) use vectors mixing continuous, malleable opinions with binary, fixed demographic elements to examine the effects of demographic faultlines in team performance. Flache & Macy (2011) removes the demographic elements and embeds these agents in small-world social networks. This is the most in-depth use of vector opinions, using negative influence and homophily to develop weak diversity in the form of bi-polarization. Mäs et al. (2013) adds demographic elements back, along with small-world networks, to again examine team performance and group stability rather than opinion dynamics. In this form, they find that sub-groups initially separate and then find consensus.

2.4 Social science relating to opinion dynamics models

In this section, the social science literature relevant opinion dynamics models are briefly reviewed. The literature on interpersonal influence is vast; it is the primary focus of the field of social psychology. Additionally, the literature on group opinion dynamics, the domain of sociology, expands that corpus significantly. Therefore, this is not meant to be exhaustive with respect to influence; instead, only literature directly related to the models in the previous section or the research being proposed are presented.

2.4.1 Choice shift

Theories that underpin many opinion dynamics models arose from the experimental observations of *choice shift* in the forms of group polarization and depolarization. *Group polarization* is the tendency of a group's mean opinion, through deliberation, to become more extreme than the mean opinions prior to deliberation. *Group depolarization* is the tendency after deliberation of a group composed of two subgroups

with opposing viewpoints to approach a mean opinion less extreme than either subgroups' opinions prior to deliberation. Individual opinions also trend along with the groups in both cases.

Stoner (1968) ignited attempts to explain group polarization when he found a tendency for groups to accept more risk than their constituent individuals, a phenomenon that became known as the *risky shift*. He gave a group of students a scenario in which an individual is debating taking a risk by writing a novel. If successful, this would seriously bolster their career, but failure would mean a waste of time and effort for which they would never be paid. Students were asked what minimum chance of success the author should have in order to take the risk, then given the chance to discuss it as a group and select a chance of success as a group. The groups tended to suggest taking greater risk than the individuals composing that group had suggested prior to discussion; group discussion seemed to elicit riskier behavior.

Moscovici & Zavalloni (1969) proposed and showed that this phenomenon is not restricted to risky behavior but is a specific case of group polarization. They found that although their students in France held positive opinions of the French president and negative opinions of Americans, discussion actually strengthened students' attitudes in these topics. This has since been confirmed by others in hundreds of articles using different experimental techniques (Isenberg 1986). In the political science domain, it has gained enough traction to be raised to the status of the *Law of Group Polarization* (Sunstein 2002). In particular, group polarization has been shown to occur among ideologically like-minded individuals when discussing American politics (Schkade et al. 2010).

Group polarization research tended to focus on isolated groups that had a predisposition toward one side of an issue. Vinokur & Burnstein (1978) demonstrated group depolarization by examining what occurs when two such groups on opposite

sides of an issue meet to have a discussion. They found that in this case, the difference in mean opinion between the groups decreases. The mean opinion of all participants tends to become more polarized due to one side having more influence than the other, so examining solely the group mean would be misleading. It is notable that this depolarization outcome would not be predicted by bounded confidence models.

In the following section, five theories of choice shift are explored: social comparison theory, persuasive arguments theory, self-categorization theory, social decision scheme theory, and social influence network theory. These are each well summarized by Friedkin (1999). Social comparison theory represents the group-normative aspect of influence, which may be thought of as an emotional component. Persuasive arguments theory represents a more reasoned influence based on weighing the arguments for and against a point of view; this may be thought of as a logical component. Self-categorization theory represents group-normative influence along with a desire to distance oneself from other groups. As a normative influence theory, it can also be thought of as primarily emotional. Social decision scheme theory posits that group consensus opinions are some function of initial individual opinions without regard to any particular inter-personal influence mechanisms. Social influence network theory is intended as a consolidation of the other four theories.

2.4.1.1 Social comparison theory

Festinger (1954) first proposed *social comparison theory*. This theory represents a meta-theory driven by the assumption that humans are driven to evaluate their own opinions and abilities and that, in the absence of objective methods of self-evaluation, they will evaluate their opinions and abilities by comparing themselves to others. For the purpose of this discussion, discussion of the theory with respect to abilities is ignored; rather, the focus is on opinions. The drive to compare one's opinion to an-

other's opinion diminishes as the difference between those two individuals' opinions increases. This leads to an uncomfortable situation for individuals who are driven to evaluate their opinions but are surrounded by others with vastly differing opinions, motivating people to seek communities of like-minded individuals against whose opinions they may evaluate themselves. Furthermore, where diversity of opinion exists within a group, an individual will be motivated to decrease this diversity both by altering their own beliefs to be more in line with an opinion perceived as desirable in the group and by altering others' beliefs to be more in line with their own. This motivation varies in proportion to an individual's distance from the desirable opinion; those who are close to that opinion will seek to influence others and broaden the range of individuals with whom they compare themselves, whereas those who are farther from it will seek to change their own position to conform with the group and compare themselves with a narrow range of individuals.

Social comparison theory can explain group polarization, as explained by Sanders & Baron (1977), due to shifts in the opinion that is perceived as desirable in the group. In a new group, an individual will tend to moderate their true opinions when expressing their point of view. As discussion continues, however, the fear of being perceived as too extreme is tempered by the presence of others with similar opinions. In fact, the perception may change such that the opinion perceived as desirable in the group is more extreme than any individual's initial opinion, leading to group polarization.

Social comparison theory is at the root of many opinion dynamics models, even though it may not be explicitly stated. The assumption inherent to bounded confidence and repeated averaging models is that those individuals sufficiently similar to oneself provide positive influence; this is a normative effect as predicted by social comparison theory. The assumption that those perceived as too dissimilar to oneself, and

thereby not members of the individual's self-identified group, provide no influence is also in keeping with social comparison theory. However, to my knowledge, the difference between expressed and true opinions is only present in Duggins's (2017) model. In his model, the opposite trend occurs; initially expressed opinions are untempered by others' opinions whereas later expressed opinions are strongly moderated toward the average opinion expressed up to that point.

2.4.1.2 Persuasive arguments theory

Persuasive arguments theory was proposed by Vinokur & Burnstein (1974) as an alternative explanation for choice shift. This assumes that there exists a culturally-relevant pool of persuasive arguments for risky behavior and cautious behavior. Initial attitudes will then reflect the strength and proportion of arguments in each direction. Diversity reflects the fact that not all persuasive arguments are known to every individual. Groups in which individuals know relatively few overlapping arguments are most prone to shifting their opinions from the initial attitude, as discussion brings more of these persuasive arguments into one another's awareness.

Persuasive argument theory is supported by the phenomenon of group depolarization observed by Vinokur & Burnstein (1978). Two opposite groups placed into a position where they could interact would be expected by social comparison theory to bi-polarize by interacting nearly exclusively within their own group and behaving as if the other group were not present. Instead, the groups deliberated together and, both as groups and as individuals, approached more moderate opinions than their initial opinions.

A meta-analysis of experiments in support of both social comparison theory and persuasive arguments theory found that the impact of both theories acted in concert to predict opinion shifts (Isenberg 1986). The author notes that the effect size observed

in support of persuasive arguments tends to be larger than those in support of social comparison theory. The fact that the two theories work together is generally accepted; even authors such as Sanders & Baron (1977) explicitly defending social comparison theory argue that both theories work together.

Despite persuasive arguments theory being fairly well accepted by social scientists, I am aware of only one opinion dynamics model with it explicitly implemented (Mäs & Flache 2013). Genetic implementations of opinion (Deffuant et al. 2000; C. Weimer et al. 2013) can also be considered to be using persuasive arguments theory although it is not explicitly stated.

2.4.1.3 Self-categorization theory

Self-categorization theory grew out of *social identity theory*. Tajfel & Turner (1979) proposed social identity theory, which predicts that humans will naturally categorize the people with whom they interact into groups, then identify with certain groups (in-groups) and contrast themselves with other groups (out-groups). Experiments have supported the idea that individuals in groups tend to modify their behavior in order to avoid signaling belonging to an out-group (J. Berger & Heath 2008), which may serve as a distancing function between groups that grow too similar (or, in opinion dynamics modeling terms, preserve clusters). Turner & Oakes (1986) noted that individuals take on an identity that is defined more at the individual or at the group level depending upon the situation.

Turner et al. (1987) expanded this concept into *self-categorization theory*, which proposes that individuals shift their opinions toward a perceived “prototypical” opinion of their in-group. This prototypicality is an opinion that best defines their group while providing separation from the opinions held by others in out-groups. This stands in contrast with persuasive arguments theory by predicting that the persua-

siveness of an argument is not an attribute associated with that argument but rather a function of how closely it is considered to hold to the prototypical position of their in-group.

Self-categorization theory is explicitly modeled by Salzarulo (2006). This model shows how self-categorization theory can explain both group polarization and group depolarization depending on the distribution of initial opinions and the construction of the prototypicality function.

2.4.1.4 Social decision scheme theory

Social decision scheme theory was first proposed by Davis (1973) to explain how groups may come to decide among a set of possible choices that are proposed by individuals in that group when voting is not performed but rather a decision must be reached by consensus. This posits that some combinatorial process exists that links the distribution of initial opinions of the group to a final opinion. In well-defined cases this might be the majority opinion, for example; in less-defined cases it might be the median, mean, most extreme, or any other rule.

An example of social decision scheme theory in action, presented by Zuber et al. (1992), looked at sequential choices by subjects given various information regarding others' first choices. In this condition, a median-opinion social decision scheme outperformed persuasive arguments in predicting the distribution of choices after differing information was provided.

Social decision scheme theory is not explicitly stated as the basis for an opinion dynamics model, to my knowledge. However, in the sense that every agent-based model implements some combinatorial function based on the initial distribution of opinions (perhaps in addition to other factors), every agent-based opinion dynamics model adheres to this theory. Perhaps the closest adherent is the repeated averaging

model (DeGroot 1974), which implements a well-defined, deterministic function to the initial distribution of opinions to find consensus if consensus exists.

2.4.1.5 Social influence network theory

Social influence network theory was first proposed by French (1956). It describes individuals as emanating an influence field that affects those to whom they are directly connected in a social network. The strength of this influence field varies with social power and inversely with the difference in their opinions. Under this theory, individuals shift their opinions until the combined effect of all influence fields is zero. A more general form of this theory was implemented by DeGroot (1974), whose model is described in detail above, which showed that this stable point in a connected social network is population consensus.

According to Friedkin (1986), this model is no longer considered useful due to advancements in social psychology since 1956, but he provided a modernized version of social influence network theory to replace it. The basis of the updated form of social influence network theory is the equation:

$$m_{i(t+1)} = \sum_{j=1}^N w_{ij} m_{j(t)} \text{ for } i = 1, \dots, N \quad (32)$$

where $m_{i(t)}$ is the opinion of agent i at time t , $w_{ij} \in [0, 1]$ is the weight of influence from agent j on agent i , and N is the size of the population. Opinion $m_{i(t)}$ is a column vector of real-valued scalars. The sum of all weights of influence on a specific agent $\sum_{j=1}^N w_{ij} = 1, \forall i$. Furthermore, every agent has positive influence upon themselves, $w_{ii} > 0, \forall i$, and every influence is bi-directional, $w_{ij} > 0 \leftrightarrow w_{ji} > 0$. This is fundamentally the same as the DeGroot (1974) model, but the network structure approach implied by the restrictions on values allows for different analyses.

Social influence network theory has been implemented in the opinion dynamics

models of Friedkin (1999), Friedkin (2001), Dandekar et al. (2013), and Pires & Crooks (2017).

2.4.2 Interpersonal influence

Interpersonal influence is a large field and the focus of social psychology, so a complete review of interpersonal influence is impractical. However, there are several themes that arise in the literature of opinion dynamics models and that therefore should be addressed. In particular, this section reviews research on homophily, social impact theory, negative influence, and memory.

2.4.2.1 Homophily

One of the most common assumptions in opinion dynamics models is that of *homophily*, increased influence between agents that share attributes in common. This typically comes in one of two forms: an increased probability of interaction for individuals that share similar opinions (used in models with dyadic interactions, including Deffuant et al. 2000, 2002; Weisbuch et al. 2002; Mäs & Flache 2013; Duggins 2017); or weight of influence being a decreasing function with respect to the difference between two agents' opinions (used in models with group interactions, including Hegselmann & Krause 2002; Mäs et al. 2010, 2014).

Both similarity and, relatedly, perceived attractiveness have been shown to increase the effects of influence attempts by an individual (Berscheid 1966). Furthermore, Berscheid (1966) showed that similarity in attributes perceived to be relevant to the interaction have a far stronger impact than similarities in irrelevant attributes. This distinction is important for modeling opinion dynamics, because it implies that one need not model irrelevant attributes in order to accurately model homophily. It also implies that using a set of relevant opinions (such as may be used in a vector opin-

ion implementation) is more appropriate than a single opinion. Another examination by D. Abrams et al. (1990) found that perception of belonging to the same in-group, from a self-categorization theory perspective, is the primary determinant of the degree of influence achieved during communication. This group membership-focused implementation of homophily is not implemented directly in any opinion dynamics model to my knowledge, but it is implicitly used in bounded confidence models. A more detailed review on the history of homophily is presented by McPherson et al. (2001), but it is sufficient to further state that homophily is well documented and observed, with support that backs its inclusion in most opinion dynamics models.

2.4.2.2 Social impact theory

Social impact theory (Latané 1981) is an attempt to codify mathematically how influence between two individuals varies. Social impact theory is composed of three principles. First, the degree of social impact upon an individual from an interaction (I) is an increasing function of the strength (S) and immediacy (I) of the interaction along with the number of sources present (N) — $I = f(SIN)$. Second, the marginal influence of each additional source diminishes according to a power law — $I = sN^t$, where s is a scaling constant and t is some power such that $0 < t < 1$. Lastly, the social impact of an individual diminishes as the number of targets increases according to a power law — $I = sN^{-t}$ for outward influence. Nowak et al. (1990) and Mäs et al. (2014) both implement social impact theory and note that experimental results seem to support that $t \approx 0.5$.

2.4.2.3 Negative influence

Positive influence, whereby interaction pushes individuals to have more similar opinions, is fairly ubiquitous among opinion dynamics models. Negative influence,

whereby interaction pushes individuals with sufficiently dissimilar opinions to further separate, is less ubiquitous but observable for example in the models of Jager & Amblard (2005), Mäs et al. (2014), and Duggins (2017). For modelers desiring an outcome of diverse opinions, the presence of negative influence is a tempting assumption. Furthermore, there has been some evidence that negative influence occurs. Berscheid (1966), for example, found that negative influence occurred when two individuals who are dissimilar with respect to traits that are relevant to the discussion interacted. From the self-categorization theory perspective results have been mixed; D. Abrams et al. (1990) did not observe any negative influence between individuals from separate groups, but Hogg et al. (1990) observed separation of opinions between groups. The latter outcome can be explained in ways that do not rely upon negative influence; meta-contrast as modeled by Salzarulo (2006) would predict a similar outcome. Additionally, a recent experiment by Takács et al. (2016) found no evidence of negative influence but rather decreased influence as similarity decreased. These mixed results suggest that, if negative influence exists, its effect is much smaller than that of positive influence. Opinion dynamics modelers should, then, use negative influence with caution and never implement it in a way that suggests it is comparable in strength to positive influence.

2.4.3 Memory

It is surprising that, of the opinion dynamics reviewed above, only two had an explicit implementation of memory: those of Mark (1998) and Mäs & Flache (2013). These implementations do not have strong justifications in social science, however, so it is worth looking briefly at the social scientific theories in this area.

Asch (1946) first found in experiments that, given a list of words describing an individual, people tend to be more influenced by the first words that they are given.

This relatively strong impact of first impressions is known as the *primacy effect* or *anchoring*. On the other hand, Miller & Campbell (1959) found that more recently encountered information has the stronger effect if a delay existed between the presentation of two pieces of information. This effect is termed the *recency effect*.

By their nature, these effects are difficult to disentangle from one another, particularly in the context of complex opinions. However, it is much simpler to assess the relative strength of primacy and recency effects upon memory. In a serial recall task, a participant is presented with a list of things to remember; these might be numbers, letters, words, images, or other things depending on the experiment. The participant is later asked to remember that list in the correct order. In a memory context, the primacy effect seems to be stronger than the recency effect although they clearly interact (Jahnke 1963), even among lower primates (Sands & Wright 1980).

Neither primacy nor recency are modeled by Mark (1998) or Mäs & Flache (2013). Mark (1998) assumes a firm time horizon for memory within which all information is equal; a fact is remembered for a limited period of time and is then forgotten completely, and any facts in memory have an equal impact upon that agent's identity. Mäs & Flache (2013) similarly assumes a fixed number of persuasive arguments can be maintained in memory; attempts to remember a new argument result in forgetting an old argument, but all arguments in memory have equal impact upon opinion.

III. Paper #1: Agent Scheduling in Opinion Dynamics: A Taxonomy and Comparison Using Generalized Models

3.1 Introduction

The field of opinion dynamics (OD) seeks to model the mechanisms by which opinions spread through a population. Within this context, opinions may be loosely defined as Axelrod (1997) defined culture: they are any attributes that may be altered by social influence. Recent papers have extensively covered the dominant models, ideal-typical results, and challenges facing the field of OD modeling (Sîrbu et al. 2017; Flache et al. 2017). OD is inherently a multi-disciplinary field; first authors of influential papers in the field have degrees spanning cognitive science (Deffuant et al. 2000, 2002), management science (Dandekar et al. 2013), mathematics (Holley & Liggett 1975; Salzarulo 2006; Lorenz 2010), philosophy (Hegselmann & Krause 2002), psychology (Nowak et al. 1990), physics (Galam 1997; Sznajd-Weron & Sznajd 2000; Stauffer et al. 2000; Castellano et al. 2009; Martins 2009; Martins et al. 2009), political science (Axelrod 1997), social sciences (Carley 1991; Jager & Amblard 2005), sociology (Mark 1998; Friedkin 2001; Mäs et al. 2010; Mäs & Flache 2013; Mäs et al. 2013, 2014), and statistics (DeGroot 1974) to name only a few.

OD modeling lends itself to the use of what Bonabeau (2002) called the agent-based mindset, where one describes "a system from the perspective of its constituent units" (p. 7280). Using this mindset, modelers can define the rules of interaction between individual agents within the model and allow social influence to propagate throughout the system. This allows a modeler to build complex models from relatively simple rules that are based upon theories developed in the psychological and sociological literature. If these outputs are realistic, the modeler has proven that the proposed rules are sufficient to generate realistic emergent behavior. In other words,

that behavior can be explained by the proposed mechanism. This is best stated by Epstein (1999) in his motto for generative social science: "if you didn't grow it, you didn't explain its emergence" (p. 43). For the purpose of this paper, agents are any objects that populate a model implementing an agent-based mindset. This is consistent with individual agent-based models (ABM) in the taxonomy proposed by Macal (2016).

While it is possible to build continuous-time OD models, discrete-time models are more common and are thus the focus of the present paper. Discrete-time OD models map the set of N agents' opinions at time t to the set of those agents' opinions at time $t + 1$. Continuous OD models are those that map $\mathbb{R}^N \rightarrow \mathbb{R}^N$, with individual opinions drawn from a continuous range of values, typically $[0, 1]$ or $[-1, 1]$. These maps may take the form of a linear transformation as seen in the repeated averaging models (Harary 1959; Abelson 1964; DeGroot 1974), a deterministic non-linear transformation as seen in the Hegselmann-Krause (HK) bounded confidence model (Hegselmann & Krause 2002), or a stochastically-varied non-linear transformation as seen in the Deffuant-Weisbuch (DW) bounded confidence model (Deffuant et al. 2000). They may take as inputs only the vector of opinions and any model parameters, as in the basic forms of those listed above, or take additional inputs from agent characteristics such as uncertainty (Deffuant et al. 2002), vectors of arguments (Mäs & Flache 2013), or locations and personality traits (Duggins 2017).

In discrete OD models, individual opinions are drawn from a discrete set of values, typically $\{0, 1\}$ or $\{-1, 1\}$. These usually implement non-linear transformations with additional inputs from agent characteristics, often including their geographic position. Examples of discrete OD models that take this form are the voter model (Holley & Liggett 1975), the social impact theory model (Nowak et al. 1990), and the Sznajd model (Sznajd-Weron & Sznajd 2000; Stauffer et al. 2000). One discrete OD model

that breaks this form is the Ising field model of Galam (1997), which solves a global optimization problem without an agent-based implementation despite defining its elements using an agent-based mindset.

The agent-based mindset, for all its strengths, can yield weaknesses. One, which explains its emergence in recent decades as a viable method of modeling, is the amount of computation required. Interactions and associated computations tend to increase exponentially as the number of agents increases. Another potential weakness is the tendency to ignore system-level elements of behavior when focusing upon agent-level behavior.

One system-level behavior of an OD model that may have a significant effect upon dynamics is the agent schedule: which agent(s), in what order, influence (or are influenced by) which other agents in each discrete step of time. OD models have used various schedules with little justification. The Sznajd model has 2 agents simultaneously influencing their immediate neighbors in 1 dimension (Sznajd-Weron & Sznajd 2000; Stauffer et al. 2000). The repeated averaging models have all agents simultaneously being influenced by all others to whom they are connected (Harary 1959; Abelson 1964; DeGroot 1974). The bounded confidence models have 1 pair of agents influencing one another simultaneously in the DW model (Deffuant et al. 2000) or all agents simultaneously being influenced by all others in the HK model (Hegselmann & Krause 2002), subject to confidence constraints.

Research into the impact of varying agent schedules is limited. Cellular automata (CA) researchers have examined the effects of varying agent schedules upon CA behavior (Page 1997; Cornforth et al. 2005) and proposed schedules based on probabilistic sets of cells acting each turn and imperfect communication of states between cells at each turn (Bouré et al. 2012). A thorough survey of existing work on the impact of scheduling upon CA behavior is provided by Fatès (2014).

The world of ABM has focused much less upon this impact. The earliest ABMs, run without computers, did not adhere to any schedule strictly (Schelling 1971). In many of the most influential ABMs, all agents act in random order (examples include Holland 1995; Epstein & Axtell 1996; Epstein 2006), and books detailing the use of ABM continue to use this schedule without explanation (Gilbert & Troitzsch 2005; North & Macal 2007). Introductory tutorials on ABM tend to avoid the question of agent scheduling entirely (Bonabeau 2002; Axelrod & Tesfatsion 2006; Macal & North 2014; Macal 2016; C. W. Weimer et al. 2016). Textbooks address the difference between synchronous and asynchronous update schedules without acknowledging the variety of schedules that fit into those broad categories (Railsback & Grimm 2011; Wilensky & Rand 2015). None of these resources delve into the depth of schedules that may exist.

Research into the impact of scheduling upon ABMs is limited to a small handful of articles. Caron-Lormier et al. (2008) used a basic ecological ABM to show a significant difference in the behavior of the model when switching between two schedules. Fatès & Chevrier (2010) compared the behavior of a basic ABM based upon various deconfliction rules paired with a particular synchronous schedule and found significant differences. Bonnell et al. (2016) used a basic foraging model to examine the combined effects of cell size, cell heterogeneity, and two specific schedules upon patterns of behavior; they found significant non-linear effects and interactions between these inputs.

Notably in OD modeling, Urbig et al. (2008) generalized the DW and HK bounded confidence models into a single model where all agents are simultaneously influenced by up to m others simultaneously. They found that the general behavior of the models is qualitatively similar, although the value of m did have various effects on the specifics of that behavior. However, direction of influence was not addressed; agents

were influenced by other agents' opinions without reciprocating influence, making the generalized model adaptable to replicate the HK model but not the DW model.

The Overview, Design concepts, Details (ODD) protocol (Grimm et al. 2010) encourages ABM modelers to explicitly specify the schedule used in the model. More recently, Collins et al. (2015) called for development of descriptive standards for agent-based models, but as yet no standard exists with which to communicate agent schedules in a way that is adequate for OD modeling. Some attempts at this do exist; one example focused specifically upon cellular automata is that of Cornforth et al. (2005). However, it is insufficient to adequately explain an OD model's schedule due to not addressing direction of influence and adhering to only the extremes of scale in which one or all agents act per time step.

The purpose of this paper is threefold: (1) to build a taxonomy for agent scheduling that is adequate for describing OD model schedules, (2) to demonstrate the potential impact of various schedules using influential continuous OD models, and (3) to discuss social interpretations of schedule choices. This taxonomy also serves to potentially unite disparate models implementing similar assumptions, such as the bounded confidence models, into a single set of parameters as recently called for by Flache et al. (2017).

3.2 Synchrony, Actor type, Scale (SAS) scheduling taxonomy

The Synchrony, Actor type, Scale (SAS) taxonomy is a concise method of communicating the schedule of an OD model. The components should be reported in order as shown in the summary in Table 1. Synchrony relates to whether states are continuously updated as agents act and has two options: synchronous and asynchronous. Actor type relates to the direction of influence and has four options: target, source, group, and mixed. Scale relates to the number of actors chosen for each role per time

Table 1. Summary of SAS taxonomy by component

Synchrony	Actor type	Scale
Synchronous or Asynchronous	Target	} (s, t)
	Source	
	Group	} (r, s, t, u)
	Mixed	

step and has 2 parameters (or 4 if actor type is mixed). Each component is explained in detail below. Table 2 lists the schedules of many OD models in the literature using this taxonomy.

3.2.1 Synchrony

Synchrony refers to whether updates to each agent’s state occur in parallel or in series. A model in which all agent updates occur in parallel would be called *synchronous*. A model in which some or all agent updates occur in series would be called *asynchronous*.

Let \mathcal{A} be a set of ordered source-target pairs (i, j) , where $i \neq j$, chosen to exhibit influence in a given time step in an OD model. Let $f_{\mathcal{A}} : \mathbb{R}^N \rightarrow \mathbb{R}^N$ be the function that maps a vector of N opinions to the vector of N opinions after the pairs in \mathcal{A} exhibit their influence synchronously. This synchronous model cannot be broken down further. A single time step of an asynchronous model, to the contrary, can be broken into a repeated application of $f_{\{(i,j)\}}$, in some order, for each (i, j) pair in \mathcal{A} . Synchronous and asynchronous models are therefore identical when \mathcal{A} consists of no more than one (i, j) pair, i.e., when at most one interaction occurs per time step.

Synchrony is best illustrated using a repeated averaging model. Let $\mathbf{x}_t \in \mathbb{R}^N$ be the row vector of opinions at time t . Let w_{ij} be the weight of influence from agent i to agent j , where $w_{ii} = 0$. An OD model can be formulated as an $N \times N$ matrix \mathbf{P}

Table 2. Schedules of opinion dynamics models using SAS taxonomy

OD Model	Schedule Used
DeGroot (1974)	Synchronous Target (∞, ∞)
Holley & Liggett (1975)	Synchronous Target $(1, \infty)$
Nowak et al. (1990)	Synchronous Target $(233, \infty)$
Carley (1991)	Synchronous Group $(2, \infty)$
Axelrod (1997)	Asynchronous Target $(4, 1)$
Mark (1998)	Synchronous Group $(2, \infty)$
Friedkin (1999)	Synchronous Target (∞, ∞)
Deffuant et al. (2000)	Synchronous Group $(2, 1)$
Sznajd-Weron & Sznajd (2000)	Synchronous Source $(2, 1)$
Stauffer et al. (2000)	Synchronous Source $(2, 3)$
Friedkin (2001)	Synchronous Target (∞, ∞)
Hegselmann & Krause (2002)	Synchronous Target (∞, ∞)
Jager & Amblard (2005)	Synchronous Group $(2, 1)$
Salzarulo (2006)	Asynchronous Target $(\infty, 1)$
Martins et al. (2009)	Asynchronous Target $(1, 1)$
Mäs et al. (2010)	Synchronous Target $(\infty, 1)$
Lorenz (2010)	Synchronous Target (∞, ∞)
Dandekar et al. (2013)	Synchronous Target (∞, ∞)
Mäs et al. (2013)	Asynchronous Target $(1, 1)$
Mäs & Flache (2013)	Asynchronous Target $(1, 1)$
Mäs et al. (2014)	Synchronous Target $(\infty, 1)$
Duggins (2017)	Asynchronous Target (∞, ∞)

where

$$\mathbf{x}_{t+1} = \mathbf{x}_t \mathbf{P}$$

The synchronous model's matrix would be defined by

$$P_{ij}^A = \begin{cases} 0 & \text{if } i \neq j, (i, j) \notin \mathcal{A} \\ w_{ij} & \text{if } i \neq j, (i, j) \in \mathcal{A} \\ 1 - \sum_{k=1}^N w_{ik} & \text{if } i = j \end{cases}$$

To generate an asynchronous model, let $\mathbf{P}^{(i,j)}$ be the $N \times N$ matrix defining a synchronous model where $\mathcal{A} = \{(i, j)\}$. Let $\mathcal{A}^{(l)}$ be the l th pair drawn in some order

from \mathcal{A} . Then the asynchronous model is defined by

$$\mathbf{P} = \mathbf{P}^{\mathcal{A}^{(1)}} \mathbf{P}^{\mathcal{A}^{(2)}} \dots \mathbf{P}^{\mathcal{A}^{(|\mathcal{A}|)}}$$

A synchronous model's update in an object-oriented programming language may be written as two steps: (1) update temporary opinions for each agent using the permanent opinions of other agents and (2) update permanent opinions to match temporary opinions. This avoids cascading effects within a single time step such that agent i 's opinion at time t has no effect on agent j 's opinion at time $t+1$ unless agent i directly influences agent j .

An asynchronous model updates permanent opinions directly after each interaction. This allows cascading effects within a single time step such that agent i 's opinion may influence agent j 's opinion without directly influencing agent j . This occurs when there is an unbroken chain of influence between agent i and agent j in the order of agents' actions.

3.2.2 Actor type

In the context of the SAS taxonomy, the *primary actor* in an OD model is the entity that is chosen to act. In an agent-based model, this is the agent or group of agents that executes code directly. Primary actors may be paired with other agents in the course of this action. An agent paired in such a way is a *secondary actor*. In a model not explicitly coded as an agent-based model, the primary actor is implicitly identified as performing some action by the mathematical formulation.

In defining a model built from an agent-based mindset, three primary actor types exist: *source agents* who influence others when they act, *target agents* who are influenced by others when they act, and *groups of agents* who mutually influence one another. The choice of agent type impacts the schedule both in how pairing of agents

is performed and, for asynchronous schedules, the order in which influence occurs. Figure 10 shows examples of possible pairings based on actor types. Additionally, it is possible for actors to be of *mixed type*; various “breeds” of agent may act differently, for example, or an agent’s action may vary as a function of the model state.

If primary actors are source agents, some set of secondary actors (targets) is chosen for each primary actor. In an asynchronous model, this source influences all of its targets before another source acts. If primary actors are target agents, some set of secondary actors (sources) is chosen for each primary actor. In an asynchronous model, this target is influenced by all of its sources before another target acts.

If primary actors are groups of agents, there are no secondary actors; each group exhibits influence between member agents as defined by the model. Each group action may be represented as an OD model executed upon a subset of agents, so these schedules may be further refined using the SAS taxonomy. Deconfliction rules may be required for synchronous models when one agent may be chosen to act as a member of multiple groups.

If primary actors are of mixed types, some or all of the above actor types exist and act within the model. It must be specified whether actors of a given type act before actors of another type, representing sequential applications of multiple OD models within a time step, or they act in mixed order, representing a truly mixed OD model.

Regardless of primary actor type, it is vital for the modeler to further detail the order in which agents take action. Random order is common, but potential alternative ordering techniques are without limit.

3.2.3 Scale

In this context, scale refers to the number of agents contained in sets of primary actors, secondary actors, and groups. For a model using source actors, no more than

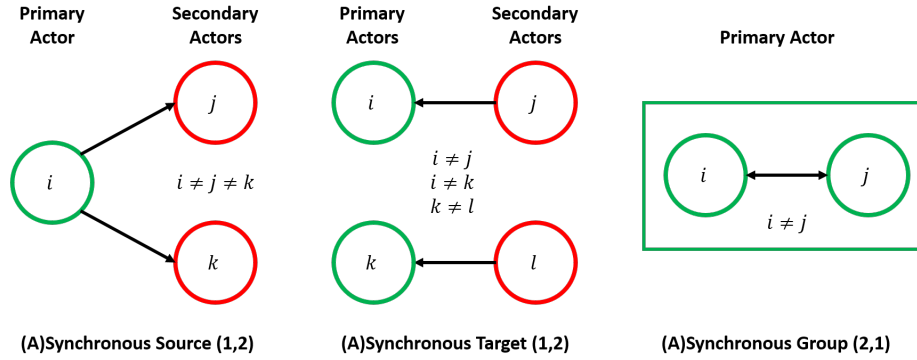


Figure 10. Graphical view of flow of influence for three types of actor

s source agents are chosen as primary actors, with no more than t target agents selected as secondary actors for each primary actor. For a model using target actors, no more than t target agents are chosen as primary actors, with no more than s source agents selected as secondary actors for each. For a model using group actors, no more than t groups of no more than s agents each are chosen to influence one another. For a model with mixed-type actors, no more than r agents are chosen to act as primary actors. If a primary actor acts as a target, it is influenced by no more than s secondary actors; if it acts as a source, it influences no more than t secondary actors; and if it is a group, it consists of no more than u agents. Parameters should be listed in alphabetical order.

If the model imposes scaling limits below the number of agents in the model, related parameters should be given as positive integers. Otherwise, parameters should be reported as ∞ to communicate the scale is not limited. This allows direct comparison between otherwise-identical models of differing population sizes. Similarly, if individual agents are heterogeneous with respect to scale parameters, the largest scale parameter should be reported along with more detail regarding the heterogeneous values.

3.3 Generalized Repeated Averaging Model

To demonstrate the use of the SAS taxonomy, and to demonstrate potential differences that may arise in model outputs as a result of differing schedules, we examine two models generalized from those available in the literature. The first of these is the repeated averaging model.

3.3.1 Model definition

As originally presented by Harary (1959), the repeated averaging model uses a linear transformation upon the vector of agents' opinions to perform discrete-time updates. This linear transformation takes the form of an $N \times N$ right-stochastic matrix of pair-wise weights between agents. DeGroot (1974) and R. L. Berger (1981) proved that this model tends to converge under reasonable conditions. We limit our model such that these conditions are met.

The generalized repeated averaging model (GRAM) removes the requirement that the model's matrix be pre-determined and static. For ease of communication, we use the transposed form of earlier models. Let \mathbf{x}_t be the row vector of agent opinions at time t and \mathbf{P} be the left-stochastic matrix defining the OD model from time t to time $t + 1$. Then,

$$\mathbf{x}_{t+1} = \mathbf{x}_t \mathbf{P}$$

This matrix is further restricted to have diagonal elements of value $(1 - \mu)$, where $\mu \in (0, 1)$ is a convergence parameter analogous to that in the Deffuant-Weisbuch model (Deffuant et al. 2000). An individual agent, if influenced during time t , grants μ influence to others while maintaining $(1 - \mu)$ self-influence. This ensures that the model converges and allows direct comparison between varied schedules.

\mathbf{P} may vary over time as different primary and secondary actors are chosen and, in

the case of asynchronous schedules, as the order of actors changes. Actors are always selected at random from the pool of available agents. As all agents are equally likely to be chosen at each time step, the social network is a complete graph. However, depending on schedule parameters, every pair of agents need not interact at each time step.

Further definition of the GRAM varies by synchrony and actor type. For precision in defining the model, it is defined in matrix form. However, all simulations were performed using an agent-based implementation in NetLogo 6.0 (Wilensky 1999). That implementation can be found in Appendix R.

3.3.1.1 Synchronous

Let us first consider the model associated with a Synchronous Target (s, t) schedule. Let \mathcal{T} be the set of t target agents randomly chosen as primary actors, or the set of all agents if $t \geq N$. For each agent $j \in \mathcal{T}$, let \mathcal{S}_j be the set of s source agents randomly chosen to influence agent j , or the set of all other agents if $s \geq N - 1$. The matrix associated with the model appears as an $N \times N$ identity matrix where columns with indices in \mathcal{T} are modified. It may be defined element-wise by

$$P_{ij} = \begin{cases} 0 & \text{if } i \neq j, j \notin \mathcal{T} \\ 1 & \text{if } i = j, j \notin \mathcal{T} \\ \frac{\mu}{|\mathcal{S}_j|} & \text{if } i \neq j, j \in \mathcal{T}, i \in \mathcal{S}_j \\ 1 - \mu & \text{if } i = j, j \in \mathcal{T}, \\ 0 & \text{otherwise} \end{cases} \quad (33)$$

For the Synchronous Source (s, t) schedule, let \mathcal{S} be the set of s source agents randomly chosen as primary actors at time t , or the set of all agents if $s \geq N$. For

each agent $i \in \mathcal{S}$, let \mathcal{T}_i be the set of t target agents randomly chosen to be influenced by agent i , or the set of all agents if $t \geq N - 1$. By extension, let \mathcal{S}_j be the set of agents i for which $j \in \mathcal{T}_i$. The matrix associated with this model appears as an identity matrix where rows with indices in \mathcal{S} are modified. It is defined by

$$P_{ij} = \begin{cases} 0 & \text{if } i \notin \mathcal{S}, j \neq i \\ 1 & \text{if } i \notin \mathcal{S}, j = i \\ \frac{\mu}{|\mathcal{S}_j|} & \text{if } i \in \mathcal{S}, j \neq i, j \in \mathcal{T}_i \\ 1 - \mu & \text{if } i \in \mathcal{S}, j = i \\ 0 & \text{otherwise} \end{cases} \quad (34)$$

For the Synchronous Group (s, t) schedule, we assume that repeat influences may not occur; that is, if two agents are members of multiple groups, they only influence each other once. Let \mathcal{A} be the set of t s -tuples of agents chosen as primary actors, or the set of all possible s -tuples of agents if $t \geq \binom{N}{s}$. Let \mathcal{T} be the set of all agents belonging to one or more groups in \mathcal{A} . For each agent j , let \mathcal{S}_j be the set of all agents belonging to one or more groups in \mathcal{A} that also contain agent j . The matrix associated with this model is defined by

$$P_{ij} = \begin{cases} 0 & \text{if } i \neq j, j \notin \mathcal{T} \\ 1 & \text{if } i = j, j \notin \mathcal{T} \\ \frac{\mu}{|\mathcal{S}_j|} & \text{if } i \neq j, i \in \mathcal{S}_j, j \in \mathcal{T} \\ 1 - \mu & \text{if } i = j, j \in \mathcal{T} \\ 0 & \text{otherwise} \end{cases} \quad (35)$$

Mixed actor types are not used in the GRAM. The multitude of options make

direct comparison to other actor types impossible. Furthermore, we are not aware of any OD models that currently utilize mixed actor types.

3.3.1.2 Asynchronous

Asynchronous schedules are more complicated to define as a result of their iterative nature. Let us first define the matrix $\mathbf{P}^{(i,j)}$, which is used for both source and target actor types. This is the matrix associated with a Synchronous Target (1,1) GRAM where $\mathcal{T} = \{j\}$, $\mathcal{S}_j = \{i\}$, $\mu = \mu^*$.

To solve for the appropriate value of μ^* , we must ensure that $(1 - \mu)$ influence remains assigned to the agent's initial opinion at the end of the time step, potentially after several iterative updates. Consider agent j , with a set of source agents \mathcal{S}_j that will serially influence agent j in some order. Let $\mathcal{S}_j^{(k)}$ be the k th element drawn from \mathcal{S}_j . The opinion of agent j , x_j , after the k th interaction becomes

$$x_j \leftarrow (1 - \mu^*) \cdot x_j + \mu^* \cdot x_{\mathcal{S}_j^{(k)}} \quad (36)$$

where $x_{\mathcal{S}_j^{(k)}}$ is that agent's opinion when the interaction occurs, which may have been modified since the beginning of the time step. Therefore, after all interactions the opinion becomes

$$x_j \leftarrow (1 - \mu^*)^{|\mathcal{S}_j|} \cdot x_j + \sum_{k=1}^{|\mathcal{S}_j|} \left((1 - \mu^*)^{|\mathcal{S}_j| - k} \cdot \mu^* \cdot x_{\mathcal{S}_j^{(k)}} \right) \quad (37)$$

In order to maintain $(1 - \mu)$ self-influence, we then have that

$$(1 - \mu^*)^{|\mathcal{S}_j|} = (1 - \mu) \implies \mu^* = 1 - (1 - \mu)^{\frac{1}{|\mathcal{S}_j|}} \quad (38)$$

The matrix $\mathbf{P}^{(i,j)}$ is therefore defined as

$$P_{kl}^{(i,j)} = \begin{cases} 0 & \text{if } k \neq l, l \neq j \\ 1 & \text{if } k = l, l \neq j \\ 1 - (1 - \mu)^{\frac{1}{|\mathcal{S}_j|}} & \text{if } k \neq l, l = j \\ (1 - \mu)^{\frac{1}{|\mathcal{S}_j|}} & \text{if } k = l = j \\ 0 & \text{otherwise} \end{cases} \quad (39)$$

We may now define the Asynchronous Target (s, t) GRAM. Let \mathcal{T} be the set of t target agents randomly chosen as primary actors, or the set of all agents if $t \geq N$. For each agent $j \in \mathcal{T}$, let \mathcal{S}_j be the set of s source agents randomly chosen to influence agent j , or the set of all other agents if $s \geq N - 1$. Let $\mathcal{T}^{(l)}$ be the l th element drawn from \mathcal{T} , and let $\mathcal{S}_j^{(k)}$ be the k th element drawn from \mathcal{S}_j . The matrix associated with the model, then, is the ordered product of $\mathbf{P}^{(i,j)}$ matrices.

$$\mathbf{P} = \prod_{l=1}^{|\mathcal{T}|} \prod_{k=1}^{|\mathcal{S}_l|} \mathbf{P}^{(\mathcal{S}_l^{(k)}, \mathcal{T}^{(l)})} \quad (40)$$

For the Asynchronous Source (s, t) GRAM, let \mathcal{S} be the set of s source agents randomly chosen as primary actors, or the set of all agents if $s \geq N$. For each agent $i \in \mathcal{S}$, let \mathcal{T}_i be the set of t target agents randomly chosen to be influenced by agent i , or the set of all other agents if $t \geq N - 1$. By extension, let \mathcal{S}_j be the set of agents i for which $j \in \mathcal{T}_i$. Let $\mathcal{S}^{(k)}$ be the k th element drawn from \mathcal{S} , and let $\mathcal{T}_i^{(l)}$ be the l th element drawn from \mathcal{T}_i . The matrix associated with the model is again the ordered product of $\mathbf{P}^{(i,j)}$ matrices.

$$\mathbf{P} = \prod_{k=1}^{|\mathcal{S}|} \prod_{l=1}^{|\mathcal{T}_k|} \mathbf{P}^{(\mathcal{S}^{(k)}, \mathcal{T}_{\mathcal{S}^{(k)}}^{(l)})} \quad (41)$$

For the Asynchronous Group (s, t) GRAM, let \mathcal{A} be the set of t s -tuples of agents chosen as primary actors, or the set of all possible s -tuples of agents if $t \geq \binom{N}{s}$. Let $\mathcal{A}^{(k)}$ be the k th element drawn from \mathcal{A} . Let $\mathbf{P}^{\mathcal{A}^{(k)}}$ be the matrix associated with a Synchronous Group $(s, 1)$ GRAM where $\mathcal{A}^{(k)}$ is the chosen primary actor according to Equation 35. The Asynchronous Group (s, t) GRAM's associated matrix is the ordered product of $\mathbf{P}^{\mathcal{A}^{(k)}}$ matrices.

$$\mathbf{P} = \prod_{k=1}^{|\mathcal{A}|} \mathbf{P}^{\mathcal{A}^{(k)}} \quad (42)$$

3.3.2 Parameter selection

Parameters for the GRAM that remain in need of values are s , t , μ , and N . The intent of the present experiment is to motivate experimenters to explicitly state their schedule use by showing that scheduling choices can have a significant impact upon the outcome of an OD model. Therefore, a quantity of interest is the influence that particular agents have based on their order in an asynchronous model.

In an Asynchronous Target (s, t) schedule, let the coefficient associated with agent i 's influence upon agent j be denoted λ_i . As shown in Equation 37,

$$\lambda_i = (1 - \mu^*)^{|\mathcal{S}_j| - k} \cdot \mu^* = (1 - \mu)^{\frac{|\mathcal{S}_j| - k}{|\mathcal{S}_j|}} \cdot \left(1 - (1 - \mu)^{\frac{1}{|\mathcal{S}_j|}}\right) \quad (43)$$

where k is the order in which agent i was drawn from \mathcal{S}_j . It follows that the absolute difference between the influence of the last source and that of the first source is

$$\lambda_{|\mathcal{S}_j|} - \lambda_1 = 2 - \mu - (1 - \mu)^{\frac{1}{|\mathcal{S}_j|}} - (1 - \mu)^{1 - \frac{1}{|\mathcal{S}_j|}} \quad (44)$$

For a set value of μ , this is a decreasing function of $|\mathcal{S}_j|$ when $|\mathcal{S}_j| \geq 2$ and equals 0 when $|\mathcal{S}_j| = 1$. Therefore, the greatest absolute difference in influence is observed

when $s = |\mathcal{S}_j| = 2$.

$$\max_{|\mathcal{S}_j|} \lambda_{|\mathcal{S}_j|} - \lambda_1 = \left(1 - \sqrt{1 - \mu}\right)^2 \quad (45)$$

This function of $\mu \in (0, 1)$ approaches 0 as $\mu \rightarrow 0^+$ and approaches 1 as $\mu \rightarrow 1^-$. Therefore, the largest absolute difference in influence exists when $s = 2$ and μ is large.

Using the same schedule, another measure of interest would be the ratio of the last agent's influence to that of the first. From Equation 43 we can calculate this ratio.

$$\frac{\lambda_{|\mathcal{S}_j|}}{\lambda_1} = (1 - \mu)^{\frac{1}{|\mathcal{S}_j|} - 1} \quad (46)$$

Because $\mu \in (0, 1)$, for a set value μ , this is an increasing function of $|\mathcal{S}_j|$ with value 1 when $|\mathcal{S}_j| = 1$ that approaches $(1 - \mu)^{-1}$ as $|\mathcal{S}_j| \rightarrow \infty$. For a set value of $|\mathcal{S}_j|$, this is an increasing function that approaches 0 as $\mu \rightarrow 0^+$ and approaches ∞ as $\mu \rightarrow 1^-$. Therefore, the greatest influence ratio is observed when $s = \infty$ and μ is large.

In an Asynchronous Source (s, t) schedule, the precise influence of an agent is not analytically tractable because effects cascade depending upon the nature of the social network defined by primary and secondary actor selection. Some observations related to order may still be made, however. The first primary actor shifts the mean opinion in the model toward their opinion upon action. If one of that actor's secondary actors is a later primary actor, that shift further affects the mean. The probability of this occurring increases with t . Therefore, we expect the influence of an agent i to decrease with k , the order in which agent j is drawn from \mathcal{S} . Furthermore, we expect this variation in influence to increase with t . Because the opinion of source agent i is unchanging, the order in which agents are drawn from \mathcal{T}_i has no effect.

In an Asynchronous Group (s, t) schedule, as defined in the GRAM, cascading

effects behave similarly to those in the Asynchronous Source (s, t) schedule. Therefore, we expect the influence of agents in group k , where k is the order in which that group of agents is drawn from \mathcal{A} , to decrease as k increases. The probability of cascading effects increases with t , again, so we also expect the difference in influence to increase with t . Note that when $s = \infty$, only one group is possible (all agents), so the Asynchronous Group (∞, t) GRAM is identical to the Synchronous Target (∞, ∞) GRAM, which is itself equivalent to both the Synchronous Source (∞, ∞) and the Synchronous Group (∞, ∞) GRAM.

These results suggest that s should be varied between 2 and ∞ for Source and Target actor types and held at 2 for Group actors. Because there is no reason to believe that t would have a significant effect for Source actors while high values of t elicit the most effect for Target and Group actors, t should be set high for those actor types. Setting $t \geq 1000$ results in $s \cdot t$ interactions for Source and Target actors, while the number of interactions increases up to $\binom{N}{s}$ for Group actors, so setting $t = 1000$ allows for the most direct comparisons between actor types by keeping total interactions equal. Furthermore, although we expect the greatest magnitude of effect when μ is high, a range of values should be examined to assess the robustness of these findings. The number of agents, N , should be set to a high value, but computation time increases exponentially with N . Informal pilot runs varying N indicated $N = 1000$ is a good compromise value. Thus, all simulations are performed with $t = 1000, N = 1000$, s varied between 2 and 1000, and μ varied from 0.01 to 0.99 at increments of 0.01.

The order effects also suggest a manner of observing the impact that schedule may have by biasing that order. Randomly ordered agent actions should be expected to obscure any variance in influence when a large number of agents are used. However, there is no reason to believe that humans are influenced in random order by those with

whom they interact. Indeed, the dynamics of individual conversations is a relatively unexplored area in opinion dynamics (Duggins (2017) is a notable exception here). We seek to show the effect that scheduling may have by comparing models in which agent order is randomized and biased forms, in which agent order is sorted to bias opinions toward one extreme. In biased forms of GRAM, we opted to bias opinions toward 0. When primary actors are targets, sources act in order of decreasing opinion as Equations 44 and 46 show that later sources have higher influence. When primary actors are sources, they act in order of increasing opinions as early actors propagate greater cascading influence. For the same reason, when primary actors are groups in biased models, they act in increasing order of the mean opinions of agents within the group.

3.3.3 Results

The GRAM was run for 1000 replications for each set of parameters, beginning with a set of $N = 1000$ agents initialized to uniformly distributed opinions.

Normalized histograms are shown in Figures 11 and 12. For each set of parameters (seen as a column within a plot), the histogram was constructed with bins of 0.01 and frequencies normalized such that a value of 0 is observed only if no outcome resulted in that opinion and a value of 1 is observed only if that opinion was the most frequently observed outcome for that set of parameters. Each dot in these plots is colored according to the normalized frequency, with red being the most common outcome and blue being unseen outcomes.

For $s = 2$, all three primary actor types (target, source, group) were examined with and without synchrony. When schedules were asynchronous, both unbiased and biased forms were run as described above. Figure 11 shows the results as a normalized histogram. This broad view shows that there are differences between schedules

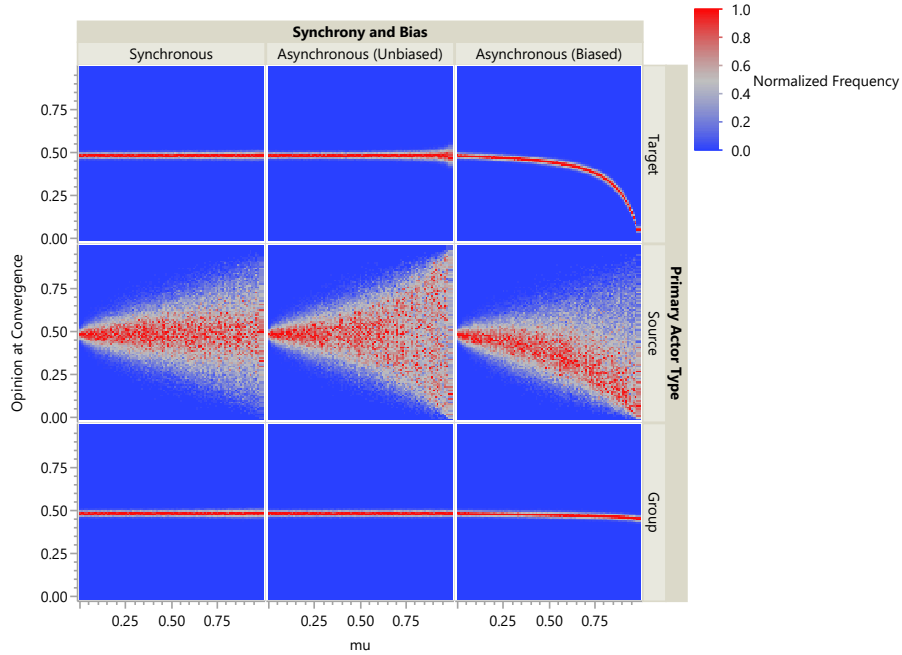


Figure 11. Histogram of observed opinions at convergence for GRAM, where $(s, t) = (2, 1000)$

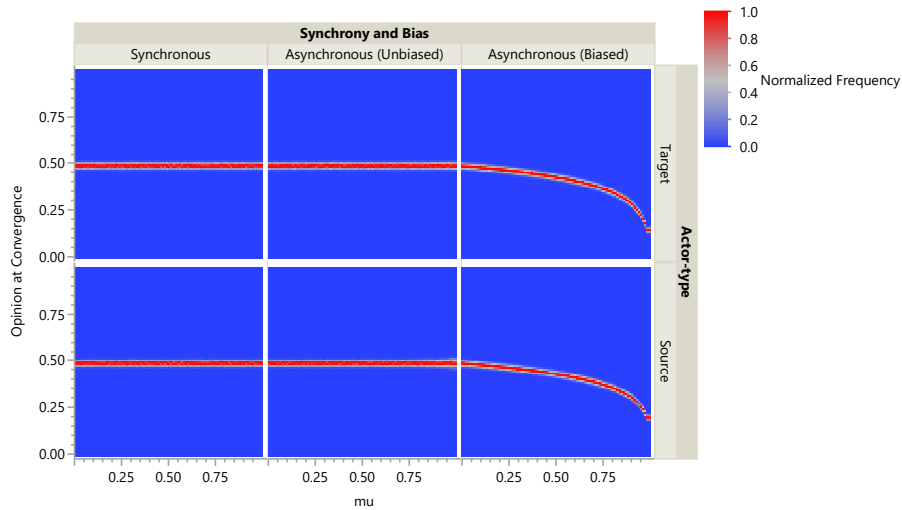


Figure 12. Histogram of observed opinions at convergence for GRAM, where $(s, t) = (\infty, 1000)$

with respect to the degree of biasing exhibited in biased models and the variance in observed opinions at convergence. Both source and target primary actors exhibited noticeable differences in variance for high values of μ when changing synchrony. For all synchrony and biases, source primary actors generate drastically higher variance in

opinions at convergence than target or group primary actors. Group primary actors exhibit the weakest biasing with relatively small variance even for high μ .

For $s = \infty$, only source and target primary actors were simulated; as the action for group primary actors is defined, those models would be equivalent for both synchronous and asynchronous schedules. Furthermore, for synchronous schedules, source and target primary actors result in identical models; therefore, only target primary actors were simulated, although the results show them in both positions for ease of comparison. When schedules were asynchronous, both unbiased and biased forms were again run. Figure 12 shows the results as a normalized histogram. A larger value of s appears from a broad view to have eliminated the differences between schedules. A closer view is warranted.

Figures 13 (for $s = 2$) and 14 (for $s = \infty$) plot the variance observed for each parameter set across 1000 replicates. Note that scales vary between plots when in Figure 13 due to the drastic differences in scale between actor types that mostly disappear for $s = \infty$. The variance in observed opinions for the Source (2, 1000) schedules stands out as being significantly different from that observed with other schedules. This is likely due to the strong influence that the first agents chosen have upon every other agent. However, closer examination shows that all schedules exhibit unique patterns of variance as μ is varied. Interestingly, increasing μ beyond a point for some schedules actually has the effect of decreasing variance. For the biased Asynchronous Target (2, 1000), biased Asynchronous Target (∞ , 1000), and biased Asynchronous Source (∞ , 1000) models, this is likely the result of the strength of schedule bias overcoming other sources of variance. For the Asynchronous Group (2, 1000) schedules, we suspect that the cohesive impact of high values of μ decreases the variance based upon early interactions of agents with disparate opinions that moderate both agents.

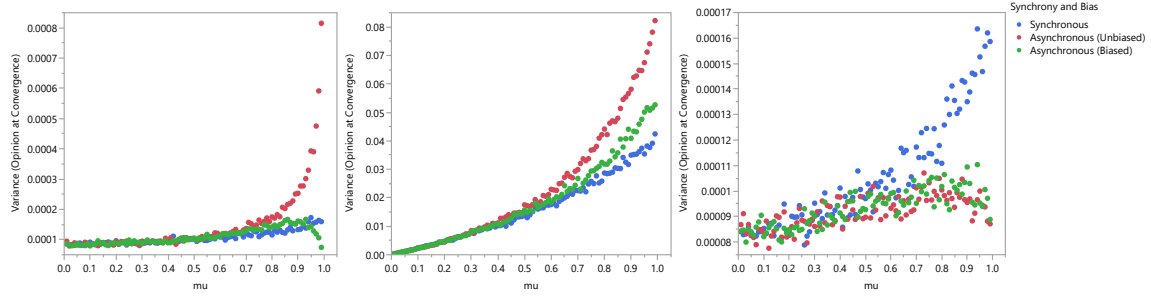


Figure 13. Variance in observed opinions at convergence for GRAM, where $(s, t) = (2, 1000)$, by primary actor type: Target (left), Source (middle), Group (right)

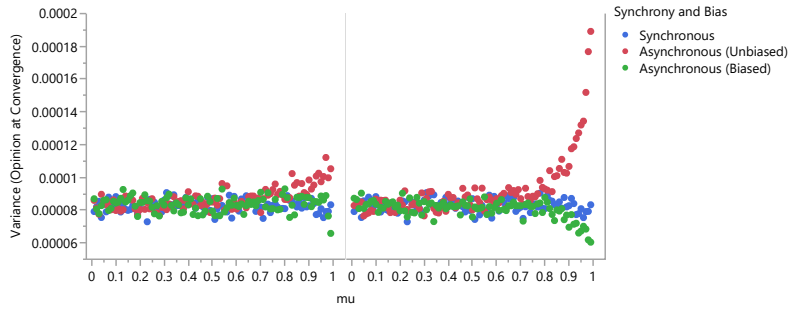


Figure 14. Variance in observed opinions at convergence for GRAM, where $(s, t) = (\infty, 1000)$, by primary actor type: Target (left), Source (right)

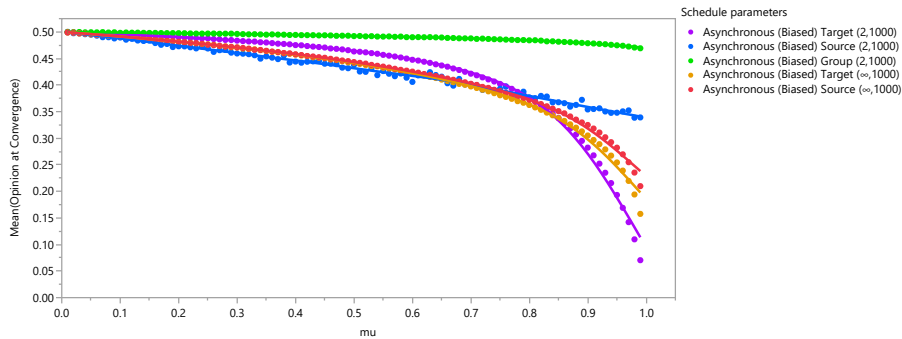


Figure 15. Mean observed opinions at convergence for GRAM

The biasing effect of ordering primary actors according to their opinion can be clearly observed in the results in Figure 15, a plot of the mean observed outcomes across all 1000 replicates for each parameter setting for which bias is included. As expected, it is strongest when the convergence parameter μ is high, but its effect is clearly observable even for low to moderate values. In these conditions, when $\mu < 0.65$, the schedule that exhibits the strongest bias is the Asynchronous Source

(2, 1000). In this schedule, the mean opinion at convergence decreases in a nearly linear fashion as μ increases, while other schedules exhibit more nonlinear effects. When $0.65 \leq \mu \leq 0.85$, the strongest bias is exhibited in the Asynchronous Target $(\infty, 1000)$ schedule. For $\mu > 0.85$, the strongest bias is in the Asynchronous Target (2, 1000) schedule. For all values of μ , biasing is weakest with an Asynchronous Group (2, 1000) schedule.

3.4 Generalized Bounded Confidence Model

While the GRAM is sufficient to demonstrate differences that may arise in model outputs as a result of schedule selection, there is value in presenting results for a more complex model in more common contemporary usage. The bounded confidence OD models presented by Deffuant et al. (2000) (the Deffuant-Weisbuch or DW model) and that presented by Hegselmann & Krause (2002) (the Hegselmann-Krause or HK model) have been heavily studied and reused. The two are very similar to one another; the primary difference relates to schedule. Using the SAS taxonomy, the DW model uses a Synchronous Group (2, 1) schedule while the HK model uses a Synchronous Target (∞, ∞) schedule.

The generalized bounded confidence model (GBCM) is a generalized model that includes the DW, but not the HK model, as a special case. Taking an average of all agents within confidence threshold d , as the HK model does, sets individual values of μ to $\frac{\mathcal{S}_i - 1}{\mathcal{S}_j}$. Thus, including the HK model perfectly into the GBCM would negate the ability to compare between schedules as μ would become heterogeneous between agents, and its scale would vary with actor type and scale parameters. A near analogue exists, however, using the same schedule but fixing μ as a homogeneous input parameter.

The GBCM differs from the GRAM in one key way: after initializing sets \mathcal{S}_j

and/or \mathcal{T}_i , they are filtered to only include secondary actors whose opinions are within d of the primary actor, where d is the confidence threshold. In order to include the base DW and HK models and enable direct comparisons between schedules, d is homogeneous across the model.

3.4.1 Model definition

The above informal definition of the model is useful for a conceptual overview of the GBCM but insufficiently precise to replicate the model. Let \mathbf{x}_t again be the row vector of N agent opinions at time t and \mathbf{P} be the left-stochastic $N \times N$ matrix defining the OD model from time t to $t + 1$. Then,

$$\mathbf{x}_{t+1} = \mathbf{x}_t \mathbf{P}$$

As with the GRAM, \mathbf{P} is further restricted to have diagonal elements of value $(1 - \mu)$, where $\mu \in (0, 1)$ is a convergence parameter. An individual agent, if influenced during time t , grants μ influence to others while maintaining $(1 - \mu)$ self-influence. This form ensures the clusters converge and allows direct comparison between schedules.

Again, \mathbf{P} may vary over time as sets of primary and secondary actors are chosen and, in the case of asynchronous schedules, as the order of actors changes. Actors are always initially selected at random from the pool of available agents, regardless of their opinions.

Further definition of the GBCM varies by synchrony and actor type. For precision in defining the model, it is defined in matrix form. However, all simulations were performed using an agent-based implementation in NetLogo 6.0. That implementation can be found in Appendix S.

3.4.1.1 Synchronous

For the Synchronous Target (s, t) schedule, let \mathcal{T} be the set of t target agents randomly chosen as primary actors, or the set of all agents if $t \geq N$. For each agent $j \in \mathcal{T}$, let \mathcal{S}_j^* be the set of s source agents randomly chosen to potentially influence agent j , or the set of all other agents if $s \geq N - 1$. Let \mathcal{I}_j be the set of agents i for whom $|x_i - x_j| \leq d$. The set $\mathcal{S}_j = \mathcal{S}_j^* \cap \mathcal{I}_j$ becomes the filtered set of eligible sources. The matrix \mathbf{P} associated with the model is the $N \times N$ matrix derived from \mathcal{T} and \mathcal{S}_j that is defined by Equation 33.

For the Synchronous Source (s, t) schedule, let \mathcal{S} be the set of s source agents randomly chosen as primary actors at time t , or the set of all agents if $s \geq N$. For each agent $i \in \mathcal{S}$, let \mathcal{T}_i^* be the set of t target agents randomly chosen to potentially be influenced by agent i , or the set of all other agents if $t \geq N - 1$. Let \mathcal{I}_i be the set of agents j for whom $|x_i - x_j| \leq d$. The set $\mathcal{T}_i = \mathcal{T}_i^* \cap \mathcal{I}_i$ becomes the filtered set of eligible targets. By extension, let \mathcal{S}_j be the set of agents i for which $j \in \mathcal{T}_i$. The matrix \mathbf{P} associated with the model is the $N \times N$ matrix derived from \mathcal{S} and \mathcal{S}_j that is defined by Equation 34.

For the Synchronous Group (s, t) schedule, we again assume that repeat influences may not occur, as in the GRAM. Let \mathcal{A} be the set of t s -tuples of agents initially chosen as primary actors, or the set of all possible s -tuples of agents if $t \geq \binom{N}{s}$. Let \mathcal{T} be the set of all agents belonging to one or more groups in \mathcal{A} . For each agent j , let \mathcal{S}_j^* be the set of all agents belonging to one or more groups in \mathcal{A} that also contain j . Let \mathcal{I}_j be the set of agents i for whom $|x_i - x_j| \leq d$. The set $\mathcal{S}_j = \mathcal{S}_j^* \cap \mathcal{I}_j$ becomes the filtered set of eligible sources. The matrix associated with the model is the $N \times N$ matrix derived from \mathcal{T} and \mathcal{S}_j that is defined by Equation 35.

As in the GRAM, mixed actor types are not used in the GBCM.

3.4.1.2 Asynchronous

Asynchronous schedules with source and target primary actors are defined using the $\mathbf{P}^{(i,j)}$ matrices defined in Equation 39 in the GBCM as in the GRAM.

For the Asynchronous Target (s, t) schedule, let \mathcal{T} be the set of t target agents randomly chosen as primary actors, or the set of all agents if $t \geq N$. For each agent $j \in \mathcal{T}$, let \mathcal{S}_j^* be the set of s source agents randomly chosen to potentially influence agent j , or the set of all other agents if $s \geq N - 1$. Let \mathcal{I}_j be the set of agents i for whom $|x_i - x_j| \leq d$. Let $\mathcal{T}^{(l)}$ be the l th element drawn from \mathcal{T} , and let $\mathcal{S}_j^{(k)}$ be the k th element drawn from \mathcal{S}_j . The set $\mathcal{S}_j = \mathcal{S}_j^* \cap \mathcal{I}_j$ becomes the filtered set of eligible sources. The matrix \mathbf{P} associated with the model is the $N \times N$ matrix defined by Equation 40.

For the Asynchronous Source (s, t) schedule, let \mathcal{S} be the set of s source agents randomly chosen as primary actors at time t , or the set of all agents if $s \geq N$. For each agent $i \in \mathcal{S}$, let \mathcal{T}_i^* be the set of t target agents randomly chosen to potentially be influenced by agent i , or the set of all other agents if $t \geq N - 1$. Let \mathcal{I}_i be the set of agents j for whom $|x_i - x_j| \leq d$. The set $\mathcal{T}_i = \mathcal{T}_i^* \cap \mathcal{I}_i$ becomes the filtered set of eligible targets. By extension, let \mathcal{S}_j be the set of agents i for which $j \in \mathcal{T}_i$. Let $\mathcal{S}^{(k)}$ be the k th element drawn from \mathcal{S} , and let $\mathcal{T}_i^{(l)}$ be the l th element drawn from \mathcal{T}_i . The matrix \mathbf{P} associated with the model is the $N \times N$ matrix defined by Equation 41.

For the Asynchronous Group (s, t) schedule, let \mathcal{A}^* be the set of t s -tuples of agents initially chosen as primary actors, or the set of all possible s -tuples of agents if $t \geq \binom{N}{s}$. Let $\mathcal{A}^{(k)}$ be the k th element drawn from \mathcal{A} . Let $\mathbf{P}^{\mathcal{A}^{(k)}}$ be the matrix associated with a Synchronous Group $(s, 1)$ GBCM where $\mathcal{A}^{(k)}$ is the chosen primary actor according to Equation 35. The Asynchronous Group (s, t) GBCM's associated matrix is the ordered product of $\mathbf{P}^{\mathcal{A}^{(k)}}$ matrices given in Equation 42.

3.4.2 Parameter selection

Parameters for the GBCM that are in need of values are s , t , μ , d , and N . Calculations for an agent's influence are identical to those for the GRAM, with the exception that $|\mathcal{S}_j| \leq s$ in GBCM. When $s = 2$, it is likely for many agents that $|\mathcal{S}_j| = 1$, so we expect to see the biasing effect reduced in GBCM compared to GRAM. In particular, when the schedule is Asynchronous (biased) Source (2, 1000), only the set of agents with opinions within d of both randomly chosen primary actors will be affected by order biasing. This implies that such agents would have opinions within $2d$ of one another. Because opinion shift is proportional to the distance between source and target opinions, these interactions will result in relatively small shifts. Thus we expect the biasing effect for the Asynchronous (biased) Source (2, 1000) schedule to be very weak.

In order to best show potential differences between schedules, it is most effective to use a value of d for which the results are well-known. Both Deffuant et al. (2000) and Hegselmann & Krause (2002) examine their models in-depth with $d = 0.20$, so we follow suit with the same value. In both models the prototypical outcome for this model is two distinct clusters with the possibility of small clusters at the opinion poles (called "wings") or between the major clusters. In their analyses, these smaller clusters were ignored, but we will include them.

All simulations are performed with $t = 1000$, $N = 1000$, $d = 0.20$, s varied between 2 and 1000, and μ varied from 0.01 to 0.99 at increments of 0.01. Aside from d , which is not defined in the GRAM, these are the same values used in the GRAM.

3.4.3 Results

The GBCM was run for 1000 replications for each set of parameters, beginning with a set of $N = 1000$ agents initialized to uniformly distributed opinions.

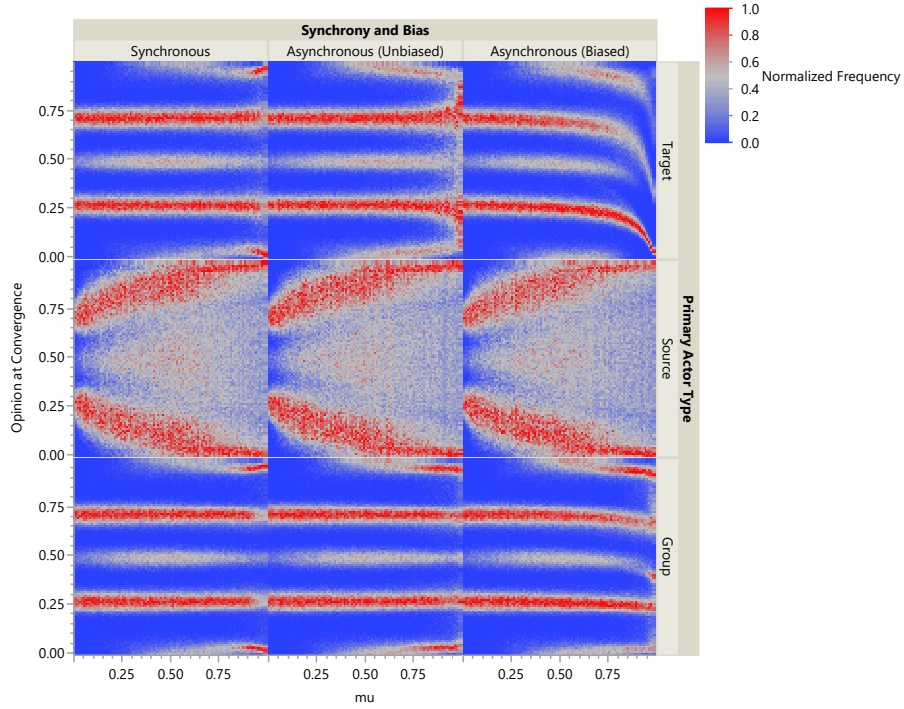


Figure 16. Histogram of observed opinions at convergence for GBCM, where $(s, t) = (2, 1000)$ and $d = 0.20$

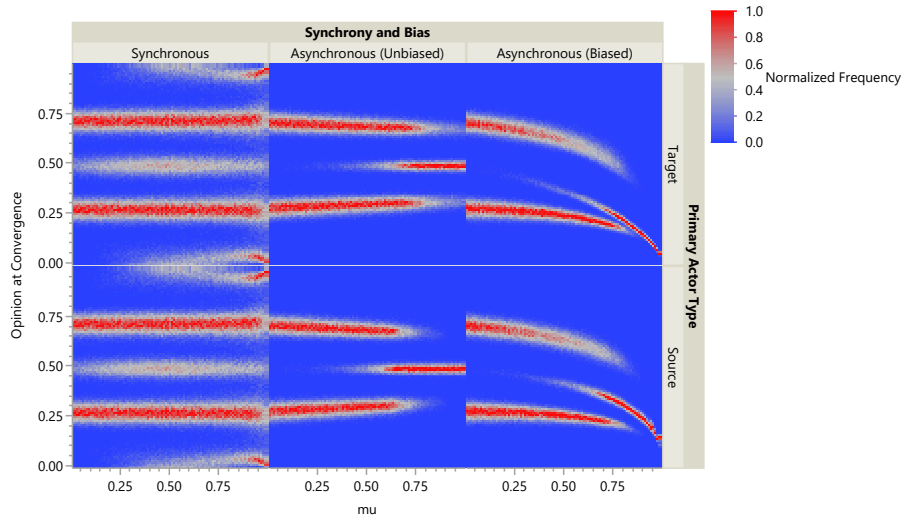


Figure 17. Histogram of observed opinions at convergence for GBCM, where $(s, t) = (\infty, 1000)$ and $d = 0.20$

Normalized histograms are shown in Figures 16 and 17. For these plots, the mean opinion within each cluster at convergence is treated as a separate outcome, regardless of how many agents were in that cluster. For each set of parameters

(seen as a column within a plot), the histogram was constructed with bins of 0.01 and frequencies normalized such that a value of 0 is observed only if no outcome resulted in that opinion and a value of 1 is observed only if that opinion was the most frequently observed outcome for that set of parameters. Each dot in these plots is colored according to the normalized frequency, with red being the most common outcome and blue being unseen outcomes.

For $s = 2$, all three primary actor types (target, source, group) were examined with and without synchrony. When schedules were asynchronous, both unbiased and biased forms were run. Figure 16 shows the results as a normalized histogram. This broad overview shows several interesting outcomes. As with the GRAM, source actors yield much higher variance in opinions in the model at convergence. Furthermore, as μ increases for these schedules, the location of the most common clusters diverge toward the opinion poles. No obvious differences exist between schedules utilizing source actors with $s = 2$. For both target and group actors, there appears to be a difference in variance in cluster opinions in the synchronous and asynchronous schedules. Biasing effects appear mild for group actors but much stronger for target actors.

For $s = \infty$, only source and target primary actors were simulated; as the action for group primary actors is defined, those models would be equivalent for both synchronous and asynchronous schedules. Furthermore, for synchronous schedules, source and target primary actors result in identical models; therefore, only target primary actors were simulated, although the results shown in Figure 17 show them in both positions for ease of comparison. While the GRAM showed the differences between schedules to be less drastic for $s = \infty$, the GBCM shows the opposite trend. Synchronous schedules exhibit similar distributions of cluster opinions with $s = \infty$ to those observed when $s = 2$. Asynchronous schedules, however, have no instances

of extreme “wings” occurring across all values of μ for either primary actor type, and only 9 observed instances of opinion convergence when $\mu < 0.25$. Instead, cluster opinions shift gradually toward a moderate opinion until they are near enough to converge when μ is high. This pattern is observed for both unbiased and biased schedules, although “moderate” opinion is significantly affected by μ for biased schedules for both actor types. It is also the opposite pattern to that observed for source actors when $s = 2$, in which clusters tended to diverge as μ increased.

In order to compare the biasing effect across schedules, Figure 18 plots the mean opinion of all agents at convergence with the same parameters, regardless of cluster membership. Unlike with GRAM, GBCM biasing effects appear to have roughly the same pattern of increasing with respect to μ , allowing schedules to be rank-ordered by their biasing effect. As predicted, the Asynchronous Source (2, 1000) schedule exhibited the weakest biasing effect; mean opinion decreases by 0.0075 as μ increases, which is visually imperceptible. From weakest to strongest, schedules are Asynchronous Source (2, 1000), Group (2, 1000), Target (2, 1000), Source (∞ , 1000), Target (∞ , 1000).

The feature that distinguishes bounded confidence models from repeated averaging models is the presence of distinct opinion clusters at convergence. As such, for these models, a primary measure of interest is the number of clusters that exist at convergence within a single replicate. Figure 19 shows the mean number of clusters of any size that exist for each schedule and μ parameter across 1000 replicates. At low values of μ , all schedules result in approximately 2 clusters. When $s = \infty$, the number of clusters decreases as μ increases for all schedules. Of these, only the Asynchronous (Unbiased) Target (∞ , 1000) schedule does not reach a single cluster when μ is sufficiently high. When $s = 2$, the number of clusters initially increases as μ increases for all schedules. The number of clusters in the Synchronous Source

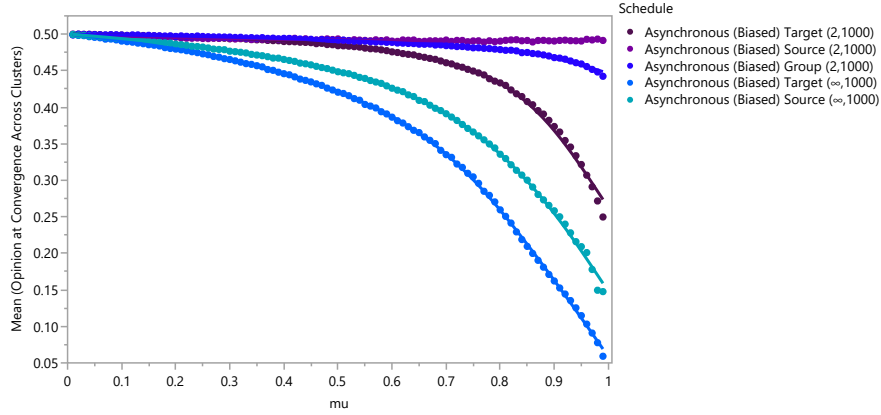


Figure 18. Mean observed opinions at convergence, averaged across clusters, per replicate of GBCM, where $d = 0.20$

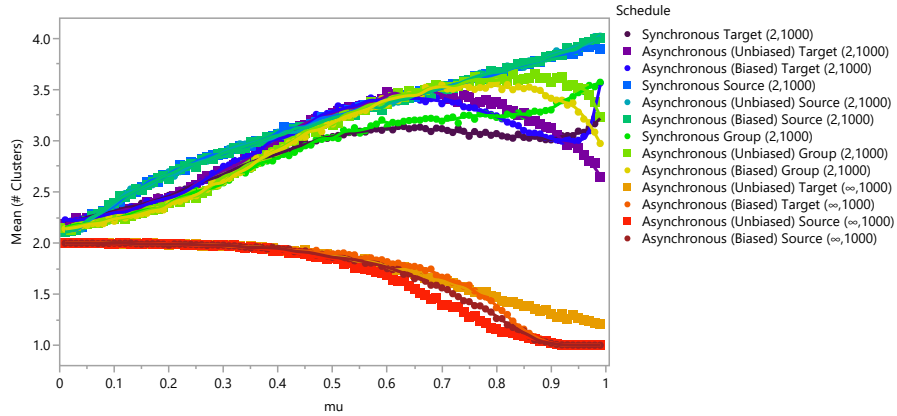


Figure 19. Mean number of clusters at convergence per replicate of GBCM, where $d = 0.20$

(2, 1000), Asynchronous (Unbiased and Biased) Source (2, 1000), and Synchronous Group (2, 1000) schedules increase as a function of μ across its entire range, while the function is non-monotonic for the remaining 5 schedules.

The variance in number of observed clusters between replicates, shown in Figure 20 also shows varying patterns for each schedule. For most schedules, the variance increases with μ to a point, then decreases. The Asynchronous (Biased) Target (2, 1000) schedule is unique in resuming an increase at the highest values of μ , while the Synchronous and Asynchronous (Unbiased and Biased) Source (2, 1000) schedules stand out with variance that increases drastically up to $\mu \approx 0.15$ and continuing to increase

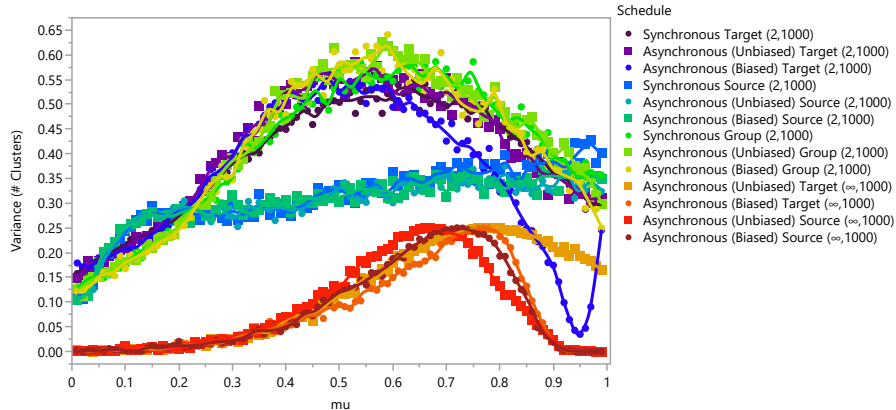


Figure 20. Variance in number of observed clusters per replicate of GBCM, where $d = 0.20$

approximately linearly for the remaining range of μ .

3.5 Discussion

The clearest outcome for both the GRAM and GBCM is that varying the schedule of agent interactions can have significant impacts upon the emergent behavior observed. This is the primary outcome intended to be demonstrated by these models, and it suggests that schedules should be more clearly stated and justified in OD models. A clear and inclusive taxonomy such as SAS makes these discussions far easier.

In the GRAM, varying synchrony in all cases altered the relationship between the convergence parameter μ and the variance in opinions observed at convergence. Varying the primary actor type had drastic effects upon that variance for $s = 2$, while that effect largely disappeared for $s = \infty$.

In the GBCM, the patterns of where emergent clusters were located varied with synchrony, especially for high values of μ . For $s = 2$ and moderate values of μ , Asynchronous schedules were more likely to generate clusters toward the poles of the opinion spectrum (i.e., “wings”) than synchronous schedules. For $s = \infty$, only syn-

chronous schedules ever resulted in these wings. The pattern of cluster locations was changed drastically by varying synchrony in this case. For all synchronous schedules, with sufficiently high values of μ , wings were more likely to be observed than clusters at more moderate locations; these models increasingly generated bi-polarization. For $s = 2$ and source-type primary actors, this effect was more clearly observed regardless of synchrony and bias; the most-observed clusters shifted toward the poles as μ was increased.

The biased models were biased solely by affecting the order in which agents were chosen to act. No schedule completely eliminated the effect of biased ordering, but in both models target actors were most affected by this ordering regardless of synchrony and parameters. This suggests that further research should be made into the dynamics of conversations; the assumption inherent to OD models that individuals speak with others in random order should be questioned. For example, it may be true that individuals tend to initiate conversations with like-minded others but are motivated to interject in existing conversations when they hear opinions that differ moderately from their own. This would imply a non-random order that may affect emergent behaviors.

The schedules in this model do not exist in a technical vacuum; there are social implications for each element. Synchrony indicates whether individuals change their minds in the course of an interaction, or if their opinion shifts after the interaction is completed during a period of reflection. Scale parameters indicate the social activity levels of an individual by defining the how many interactions they may have. This may vary by time, topic, and individual. Primary actor type reflects how an individual approaches interactions – as a learner (target), as a teacher (source), or as a collaborator (group). An individual may have different motivations, and therefore different approaches, depending upon the situation. This mixture may also vary

depending on the status of that individual; writers for media outlets and individuals with many followers on social media may act as sources far more often than others.

An interesting and counterintuitive outcome was observed in the GBCM with source actors for $s = 2$, where increasing the convergence parameter μ increased the separation between the most observed clusters. The convergence parameter could alternatively be considered “susceptibility to influence,” as it is related to how much others’ opinions affect one’s own. This suggests that the presence of primarily source-type actors has a bi-polarizing influence. Partisan news sources and social media have been examined for their effects on bi-polarization (see, for example, Levendusky 2013; Lee et al. 2014; Messing & Westwood 2014; Bakshy et al. 2015). This suggests a mechanism for such a phenomenon.

3.6 Conclusion

The SAS taxonomy represents a step toward creating a common language with which opinion dynamics researchers can compare models and discuss the social assumptions inherent to those schedules. This paper defines the taxonomy and examines variants of two generalized models using the mechanisms of repeated averaging and bounded confidence.

The outcomes of these models executed under various schedules show that varying any element of the schedule as defined can significantly affect emergent behavior in even relatively simple models. Existing models rely primarily upon randomization to order actions and choose agent type and synchrony without significant justification, but these can be interesting and useful inputs to a model. Furthermore, openly discussing these aspects of models can allow greater standardization and repeatability of models.

IV. Paper #2: Distilling Meta-contrast: The Meta-contrast Influence Field Model

4.1 Introduction

Opinion dynamics or sociophysics models are mathematical formulations for the evolution of opinions within groups. They have been criticized for being insufficiently based in reality and social science (Sobkowicz 2009). The meta-contrast (MC) model (Salzarulo 2006) is firmly rooted in the social science literature; it is an attempt to mathematically formalize the meta-contrast principle of self-categorization theory (Turner et al. 1987). This principle posits that individuals divide others into in-groups and out-groups, and they seek to exhibit an opinion prototypical of their in-group while avoiding opinions prototypical of out-groups. The MC model has been heavily cited (102 citations at the time of writing), but no known work has attempted to improve upon or extend the model.

The meta-contrast influence field (MIF) model developed herein expands on the MC model of Salzarulo (2006) in several ways. First, it equalizes the scale of previously imbalanced inputs into the prototypicality function. This prototypicality function computes how prototypical an agent will perceive other opinions to be, and it is the backbone of the MC model. The inter-group component of that prototypicality dominates the calculation in the MC formulation. Second, it implements a continuous source of influence based upon the derivative of the MC model's prototypicality function, which computed how prototypical an agent would perceive an opinion to be. Using the derivative, in effect, creates an influence field composed of competing forces acting upon the point of an agent's opinion. Finally, as a result of this field-based formulation, the schedule is altered to allow all agents to update in synchrony based upon the information available to them. The MIF model is a significant improvement

upon the MC model, and its construction allows it to be a framework upon which more nuanced models of interpersonal influence can be built.

4.2 Background

Flache et al. (2017) provide an excellent overview of the state of the art of continuous opinion dynamics models, but a historical overview of the models that have influenced the meta-contrast model motivates a way forward. A continuous opinion dynamics model is one that implements opinion as a continuous value, typically in the range $[0, 1]$ or $[-1, 1]$, rather than as one of a set of discrete values representing various choices.

Continuous opinion dynamics models trace their heritage to the *repeated averaging* model. The conceptual framework behind the repeated averaging model began with French’s work on a theory of social power (French 1956; French & Raven 1959) and were given more rigorous mathematical form by Harary (1959), Abelson (1964), and DeGroot (1974). These models examine social networks in which an individual takes as its opinion a convex combination of the opinions they observe, with weights determined by the degree of social power that others have over them. DeGroot (1974) and R. L. Berger (1981) proved that such models generate consensus if the matrix of weights has the form of an ergodic, non-periodic Markov chain, as well as under some other conditions. This condition is met by any connected social network in which no individual has total power over another’s opinion. As a result, nearly any realistic social structure generates a consensus opinion over time using this model. This led Abelson (1964) to famously ask “what on earth one must assume in order to generate the bimodal outcome of community cleavage studies” [p. 153].

One of the earliest and most influential approaches to generating such a bimodal outcome is the use of bounded confidence. *Bounded confidence (BC)* models assume

that two individuals can only influence one another if the absolute difference in their opinions is below a specified value (Deffuant et al. 2000; Hegselmann & Krause 2002). A confident individual has a lower value and thus a smaller range of opinions by which they may be influenced. Less confident individuals would be susceptible to a wider range of influences. From a Markov chain perspective, this serves to break the system into recurrent classes that do not communicate, thereby making the associated Markov chain non-ergodic and keeping consensus from occurring. This model has been examined for both homogeneous and heterogeneous confidence values, and heterogeneity makes consensus more likely (Weisbuch et al. 2002; Lorenz 2010). The behavior generated by BC models is characterized by *weak diversity*, which Duggins (2017) defines as “the convergence of opinions to a finite number of attractor states.”

The BC model is not based in any one social scientific theory but upon the principle that individuals are more likely to interact if they are more similar, or homophily, a concept which has strong support in the literature (McPherson et al. 2001). However, even weak connections between dissimilar individuals are sufficient to make the associated Markov chain ergodic and therefore generate consensus.

The social judgment model of Jager & Amblard (2005) is based explicitly upon social judgment theory (Sherif & Hovland 1961). This model implements the assumption of bounded confidence, while also adding a social rejection zone of influence. If two individuals are sufficiently similar in opinion, they will attract one another (i.e., impart positive influence), and if they are sufficiently different, they will repel one another (i.e., impart negative influence). This model generates weak diversity similar to the behavior generated by BC models, except that in models that result in at least 2 clusters, 2 of those clusters will have fully polarized opinions 0 and 1.

The assumption of negative influence has been criticized in light of mixed evidence that such a phenomenon exists (Takács et al. 2016), although other studies

indicate that negative influence may be nearly as strong as positive in certain situations (Hilmert et al. 2006). As influence is calculated as a proportion of the distance between two opinions, negative influence triggered by this formulation is always stronger than positive influence. No known evidence exists to support this, so there is reason to be skeptical of this formulation.

4.2.1 Meta-contrast model

The meta-contrast (MC) model was proposed by Salzarulo (2006) to implement the principle of meta-contrast in self-categorization theory (Turner et al. 1987). The meta-contrast principle predicts that an individual is more likely to associate with an in-group if that group is cohesive and other groups' opinions are distinct from those of the in-group. Therefore, in a setting with diverse opinions two individuals may become part of the same in-group while in less diverse context these same individuals might associate with different in-groups.

The MC model proposes a formula for calculating the degree to which a given opinion is perceived as prototypical of any group's opinion. This prototypicality has two components: an intra-group component that generates a region of increased prototypicality near each observed opinion and an inter-group component that generates increased prototypicality far from each observed opinion. The degree to which each component impacts the prototypicality calculation is determined by a model parameter. An individual is then attracted to highly prototypical opinions that are similar to their own. Specifically, the nearest local maximum on the prototypicality curve generated by their surroundings is perceived as the in-group opinion, and the individual with the nearest opinion to the in-group opinion becomes the source of positive influence. The individual takes this source opinion as its own.

Many papers in the literature referencing the MC model state that it implements

both positive and negative influence (Flache & Mäs 2008a; Mäs & Flache 2013; Groeber et al. 2014; Huet & Deffuant 2010; Takács et al. 2016; Kurahashi-Nakamura et al. 2016; Krueger et al. 2017). This is not precisely accurate; only positive influence occurs. It is considered an example of negative influence because the presence of a dissimilar individual may push the prototypical opinion farther from that individual. This generates an effect similar to negative influence by disconnecting the social network and selecting sources farther from the outgroup member. Despite this, the MC model does have the potential to address the criticism of the social judgment model that negative influence outweighs positive influence by allowing the modeler to choose the proportion of prototypicality generated by intra- and inter-group components.

In the base MC model, during each turn, one randomly chosen agent updates its opinion as described above based on observation of all others' opinions. Behavior generated by the MC model is similar to that generated by the social judgment model: one or more clusters of agents with homogeneous behavior emerge. If more than one cluster is generated, two of those clusters move toward the poles at 0 and 1.

One deviation from the base model implements a small-world network (SWN) rather than a fully connected network. SWNs, introduced by Milgram (1967) as a description of real-world social ties, are characterized by minimally interconnected, small clusters of individuals where the number of links that must be traversed to connect any two individuals is small. A popular method for generating SWNs is to first generate a ring network where a number of nearest neighbors are connected, then randomly rewire these links with some probability (Watts & Strogatz 1998). Salzarulo (2006) instead starts with a 2-dimensional grid of agents connected to their 8-agent Moore neighborhood, then randomly rewiring those connections. This generates behavior similar to the fully-connected base model except that clusters do not necessarily fully converge.

4.3 Method

4.3.1 Meta-contrast information field model definition

The meta-contrast influence field (MIF) model is based upon the MC model. The system of interest is defined by N agents each possessing opinions $x_i \in [0, 1]$ for $i = 1, \dots, N$. These opinions are initialized to random values within the defined range. The network of agents is fully connected unless otherwise stated; that is, every pair of agents in the population is connected.

An agent assesses the prototypicality of opinions according to the prototypicality function used by the MC model with an added multiplier to equalize the scale of intra- and inter-group effects. The rationale for these changes are explained in detail in section 4.3.2. From the perspective of agent i , the prototypicality of an arbitrary opinion x , given the set of opinions X of agents to whom agent i is connected, is defined by the equation

$$P(x, X) = a \cdot \lambda \cdot d_{\text{inter}}(x, X) - (1 - a) \cdot d_{\text{intra}}(x, X) \quad (47)$$

where a is the model parameter defining the proportion of influence caused by inter-group effects (generated by the desire to distinguish one's group from outgroups), λ is a scaling factor defined by

$$\lambda = \frac{w^2}{e - e^{1 - \frac{1}{w^2}}} \quad (48)$$

w is the model parameter defining the breadth of opinions one is willing to attribute to a single group (i.e., group width), $d_{\text{inter}}(x, X)$ is the inter-group component of

opinion prototypicality

$$d_{\text{inter}}(x, X) = \frac{\sum_{i=1}^{|X|} \left((1 - \mu(x, x_i)) \cdot (x - x_i)^2 \right)}{\sum_{i=1}^{|X|} (1 - \mu(x, x_i))}$$

$d_{\text{intra}}(x, X)$ is the intra-group component of opinion prototypicality

$$d_{\text{intra}}(x, X) = \frac{\sum_{i=1}^{|X|} (\mu(x, x_i) \cdot (x - x_i)^2)}{\sum_{i=1}^{|X|} \mu(x, x_i)}$$

and $\mu(x, x_i)$ is the fuzzy membership function defining the perceived degree to which opinions x and x_i belong in the same group

$$\mu(x, x_i) = e^{-\frac{(x-x_i)^2}{w^2}}. \quad (49)$$

Agents are influenced by the influence field generated by others' observed opinions. For agent i , this is a function of the gradient of the prototypicality function at their current opinion, x_i . At each update, an agent updates their opinion

$$x_i \leftarrow x_i + k \cdot \frac{\delta}{\delta x} P(x_i, X) \quad (50)$$

where k is a convergence parameter, 0 if this would result in an opinion less than 0, or 1 if this would result in an opinion greater than 1. When using the fuzzy membership function defined by Equation 49, the gradient is given by

$$\frac{\delta}{\delta x} P(x, X) = a \cdot \lambda \cdot \frac{\delta}{\delta x} d_{\text{inter}}(x, X) - (1 - a) \cdot \frac{\delta}{\delta x} d_{\text{intra}}(x, X) \quad (51)$$

$$\frac{\delta}{\delta x} d_{\text{intra}}(x, X) = 2 \left(\frac{w^2 \sum_{i=1}^{|X|} (x - x_i) \mu(x, x_i) - \sum_{i=1}^{|X|} (x - x_i)^3 \mu(x, x_i)}{w^2 \sum_{i=1}^{|X|} \mu(x, x_i)} + \frac{\sum_{i=1}^{|X|} (x - x_i) \mu(x, x_i) \cdot \sum_{i=1}^{|X|} (x - x_i)^2 \mu(x, x_i)}{w^2 \left(\sum_{i=1}^{|X|} \mu(x, x_i) \right)^2} \right)$$

$$\frac{\delta}{\delta x} d_{\text{inter}}(x, X) = 2 \left(\frac{w^2 \sum_{i=1}^{|X|} (x - x_i) (1 - \mu(x, x_i)) + \sum_{i=1}^{|X|} (x - x_i)^3 \mu(x, x_i)}{w^2 \sum_{i=1}^{|X|} (1 - \mu(x, x_i))} - \frac{\sum_{i=1}^{|X|} (x - x_i) \mu(x, x_i) \cdot \sum_{i=1}^{|X|} (x - x_i)^2 (1 - \mu(x, x_i))}{w^2 \left(\sum_{i=1}^{|X|} (1 - \mu(x, x_i)) \right)^2} \right).$$

The base MIF model uses a synchronous target (∞, ∞) model, according to the synchrony, actor type, scale (SAS) taxonomy (see Chapter III). This assumes that all agents adjust their opinions continuously according to the influence field they perceive. Values of k should then be small enough that time steps approximate continuous time.

NetLogo 6.0 (Wilensky 1999) code implementing the MIF model is given in Appendix T.

4.3.2 Changes from the meta-contrast model

To enable direct comparison of results between the MIF model and the MC model, a replication of the MC model is used that adheres to the rules specified in the text of Salzarulo (2006). In this model, during each step, one agent updates its opinion to

match that of another agent. This source agent is the one with opinion most similar to the nearest opinion with a locally maximal prototypicality. The local maximum is obtained to an accuracy of 0.01. This differs slightly from the code used in the original paper, in which a gradient-based search was used that does not guarantee that the nearest local maximum is chosen. The Netlogo 6.0 code for the MC model is given in Appendix U.

The MIF model implements three major changes to the MC model: scaling the $d_{\text{inter}}(x, X)$ by λ in the calculation of prototypicality, updating opinion as a function of the gradient, and modifying the agent schedule. These changes exist to rectify weaknesses in the original MC model; the rationale behind those changes is given below.

4.3.2.1 Scaling factor

The form of the prototypicality function (Equation 47) suggests that prototypicality is a weighted average of two components: $d_{\text{inter}}(x, X)$ and $d_{\text{intra}}(x, X)$. One might assume that both components are of a comparable scale, but Salzarulo (2006) notes that for $a \geq 0.3$ the effect of $d_{\text{inter}}(x, X)$ dominates the prototypicality function. This disparity is illustrated by comparing the individual agent's contribution to each function's numerator, which are functions of the distance from their opinion x_i to x . Let an individual's contribution to the numerator of d_{intra} be

$$\hat{d}_{\text{intra}}(|x - x_i|) = e^{-\frac{(x-x_i)^2}{w^2}} \cdot (x - x_i)^2$$

The maximum value of this function occurs when $|x - x_i| = w$, yielding a maximum contribution of

$$\max_{|x-x_i|} \hat{d}_{\text{intra}}(x - x_i) = e^{-\frac{w^2}{w^2}} \cdot w^2 = \frac{w^2}{e}$$

Let an individual's contribution to the numerator of d_{inter} be

$$\hat{d}_{\text{inter}}(|x - x_i|) = \left(1 - e^{-\frac{(x-x_i)^2}{w^2}}\right) \cdot (x - x_i)^2$$

The maximum value of this function occurs when $|x - x_i| = 1$, yielding a maximum contribution of

$$\max_{|x-x_i|} \hat{d}_{\text{inter}}(x - x_i) = \left(1 - e^{-\frac{1}{w^2}}\right) \cdot 1^2 = 1 - e^{-\frac{1}{w^2}}$$

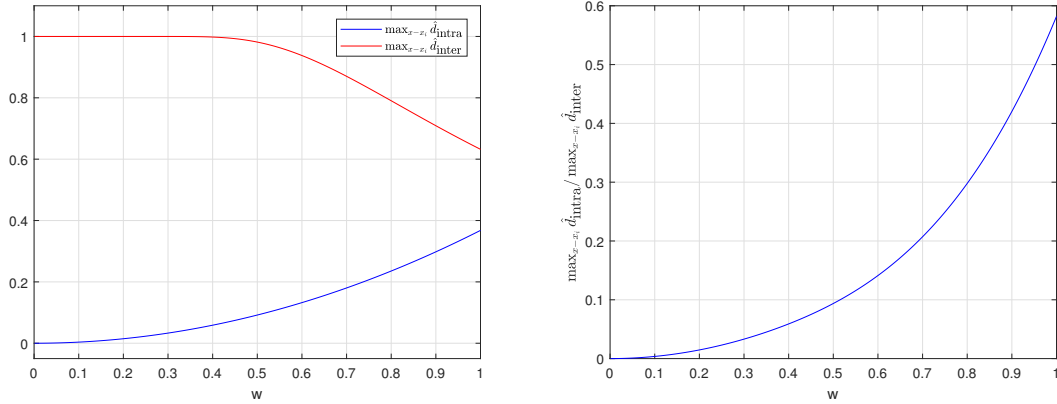


Figure 21. Maximum values as function of w (left), and ratio of these values as a function of w (right)

Figure 21 shows each of these maximum values as a function of w and the ratio of $\max_{x-x_i} \hat{d}_{\text{intra}}$ to $\max_{x-x_i} \hat{d}_{\text{inter}}$. It is clear that there is a large disparity between the ranges of $d_{\text{inter}}(x, X)$ and $d_{\text{intra}}(x, X)$, especially for low values of w . To equalize this, the MIF multiplies $d_{\text{inter}}(x, X)$ by the ratio of component maxima λ defined by Equation 48.

4.3.2.2 Agent updates

Most continuous opinion dynamics models use as their core updating mechanism some weighted sum of agent opinions. Prior to the MIF, this has remained fun-

damentally unchanged since the original repeated averaging models, although the calculation of weights varies between models. The repeated averaging model of DeGroot (1974) uses fixed non-negative weights summing to 1. Deffuant et al. (2000) and Hegselmann & Krause (2002) use averages of a subset of opinions within confidence bounds. The relative agreement model (Deffuant et al. 2002) calculates non-negative weights summing to 1 based upon uncertainty levels. Jager & Amblard (2005) use both positive and negative weights. The influence, susceptibility, and conformity (ISC) model (Duggins 2017) calculates weights based upon agent characteristics and conversational dynamics. The MC model is no different; as used by Salzarulo (2006) it imparts the source agent’s opinion fully upon the target agent, thereby giving a weight of 1 to the chosen source agent’s opinion and a weight of 0 to all others.

The prototypicality function of the MC model suggests an alternative mechanism, which the MIF model uses, when it is recast as a utility function. From this perspective, individuals should seek to hold opinions prototypical of their in-group. Therefore the derivative of this function can be considered a social pressure pushing them toward group conformity.

Using the derivative of the prototypicality function, rather than any subset of agents’ opinions, avoids non-continuous breaks in the the value upon which any change in opinion is based, or the *influence basis*. Consider the case of $X = \{x_0, 0.2, 0.8\}$. In the MC model, the basis of influence for agent 0 is $x_* - x_0$ where x_* is the nearest prototypical opinion. Four cases may occur: (1) x_1 is the nearest prototypical opinion, (2) x_2 is the nearest prototypical opinion, (3) x_0 is the nearest prototypical opinion, or (4) $x_0 = 0.5$ and a tie exists for nearest prototypical opinion. For parameters $a = 0.08$ and $w = 0.36$, (1) occurs for $x_0 \in [0, 0.043] \cap [0.2, 0.5)$, (2) occurs for $x_0 \in (0.5, 0.8] \cap [0.957, 1]$, (3) occurs for $x_0 \in (0.043, 0.2) \cap (0.8, 0.957)$, and (4) occurs for $x_0 = 0$. The corresponding basis of influence for the MC model along

with the derivative of P are shown in Figure 22. Discontinuities in the MC model's influence basis exist where $x_0 \in \{0.043, 0.5, 0.957\}$, and tie-breaking rules must be implemented to determine influence at these points. The gradient, shown in blue, is continuous. Furthermore, tie-breaking is not required; an agent caught perfectly between two prototypical opinions will be influenced equally by both and experience zero net influence.

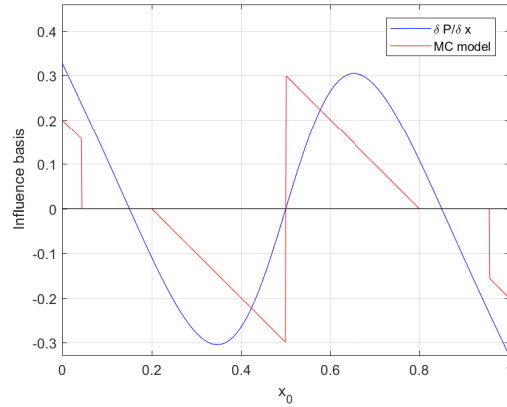


Figure 22. Basis of influence when $a = 0.08$, $w = 0.36$, $X = \{x_0, 0.2, 0.8\}$

The gradient as a basis of influence also eliminates ambiguity in source opinion selection. The MC model, as described by Salzarulo (2006), identifies the prototypical in-group opinion as the opinion closest to the nearest local maximum. However, in the associated code, a gradient-based search for local maximum is utilized. In the case where identified groups are of uneven sizes, this may lead to the farther local maximum, although it might be considered the group that exerts the most influence. Consider the case of $X = \{x_0, 0.2, 0.2, 0.2, 0.2, 0.2, 0.8\}$. Figure 23 shows the associated prototypicality curve for agent 0 when $a = 0.08$ and $w = 0.36$. If $x_0 = 0.6$, a gradient-based search will result in influence toward 0.2 rather than 0.8, despite the nearest prototypical opinion being near 1. This seems appropriate as the larger group should be expected to exert stronger influence, although it contrasts with the model as described. The gradient maintains this direction of influence.

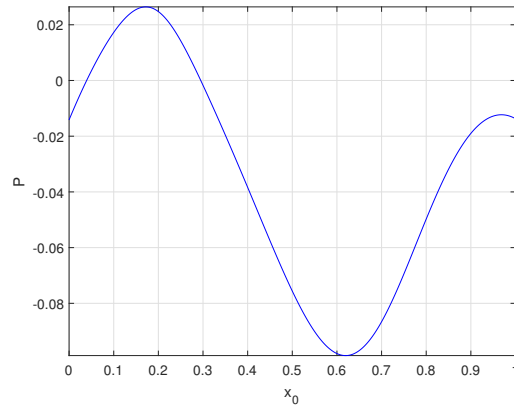


Figure 23. MC prototypicality function when $a = 0.08, w = 0.36, X = \{x_0, 0.2, 0.2, 0.2, 0.2, 0.2, 0.8\}$

The MIF influence field exhibits both positive and negative influence except when $a = 1$, in which case only negative influence exists. This can be seen in Figure 24, which plots the force exerted by an agent with opinion $x_0 = 0$ upon an agent with opinion x_1 when those two agents are interacting in isolation. At left of Figure 24, only the intra-group component is exerting force. At right of Figure 24, only the inter-group component is exerting force. Positive values represent a repulsive force from the source at 0 (i.e., negative influence) and negative values represent an attractive force toward the source at 0 (i.e., positive influence).

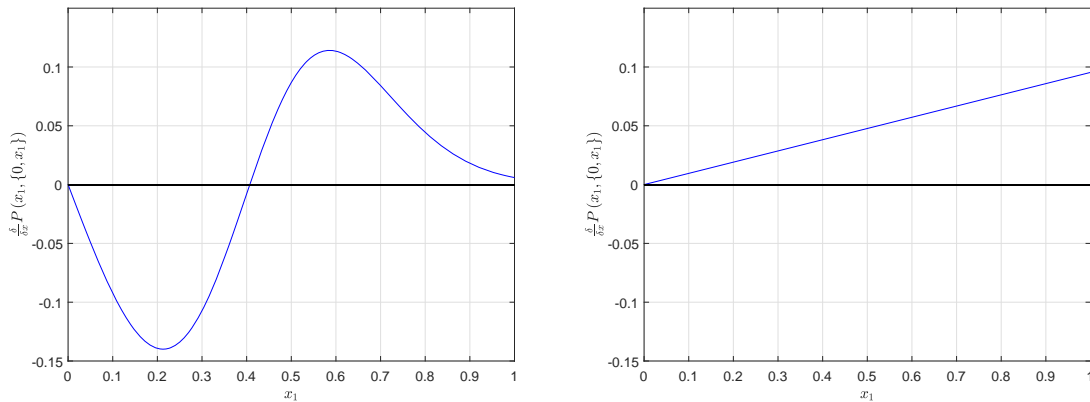


Figure 24. Influence field at x_1 when $a = 0$ (left) and $a = 1$ (right), $w = 0.36, X = \{0, x_1\}$

The gradient allows for influence based upon a single calculation rather than

performing a search for local maxima. This calculation is certainly more complex than average-based opinion dynamics models, particularly for a fully connected social network, but the increase in complexity yields more justifiable and flexible behavior.

4.3.2.3 Agent schedule

The base MC model uses an asynchronous target $(1, 1)$ schedule according to the SAS taxonomy (see Chapter III). This models sequential conversations in which high status individuals convince lower-status individuals who perceive that they are members of the same in-group. The MIF model considers agents to be continuously influenced by the influence fields that they are exposed to over time, rather than influenced at discrete times through conversation events. Therefore, a better schedule is the synchronous target (∞, ∞) schedule, in which all agents are simultaneously influenced by those to whom they are connected at every time step.

4.3.3 Small-world network variant

A strength of the MC model is that agents use the full set of observed opinions to inform their opinion updates, so an important variant to examine is one in which the social network is not fully connected and agents do not possess perfect information. The small-world network (SWN) variant of the MIF model uses the method of Watts & Strogatz (1998) to build the social network, which remains static over the course of the model run. Agents are placed into a ring network and connected with the c closest neighbors in each direction ($k = 2 \cdot c$ in the original paper's notation). Each of these connections is randomly rewired with probability p . Salzarulo (2006) used a similar method on a 2-dimensional grid of agents utilizing the 8-agent Moore neighborhood. All SWN experiments used for this paper take a fixed value $c = 8$. The MC model replication used for comparison also uses the Watts-Strogatz method, rather than the

similar method used in the original paper. These variants are built into the same code as the base model (Appendix T for MIF and Appendix U for MC).

4.4 Results

The base model was run 100 times for each set of parameters with a varied from 0 to 1 at increments of 0.01, w varied from 0.01 to 1 at increments of 0.01, and k fixed at 0.2. N was fixed at 100 as in the experiments performed by Salzarulo (2006). The primary response variable of interest is the number of clusters that form. Thus, the model is considered to have sufficiently converged when clusters have formed of width no greater than $\frac{w}{2}$, with separation of no less than w between clusters. The MC model used for comparison is stopped when the set of opinions in the population is no larger than the set of local maxima in the prototypicality curve.

Figure 25 shows the mean number of clusters that form for the base MC model and the base MIF model. Each dot in the plots represents the mean number of clusters observed over all 100 replications using a logarithmic color scale to ensure variation between both small and large numbers of clusters is visible. The dark blue region in the top-left of each plot represents convergence to consensus; a single cluster is formed. The region in the lower-left represents the opposite behavior, formation of a large number of clusters. Clustering behavior observable in the MC model is maintained in the MIF model, but the effect of changing a is diminished significantly, and additional interaction between the effects of a and w are induced by the λ multiplier.

The within-run dynamics over time can be characterized by three overlapping phases: (1) initial moderation of extreme opinions, (2) consolidation of opinions within clusters, and (3) separation of clusters if more than one remains. As a increases, the strength of phase 1 is diminished. This is well illustrated in the case where $w = 0.5$. Figure 26 shows three individual runs, where each line is the opinion

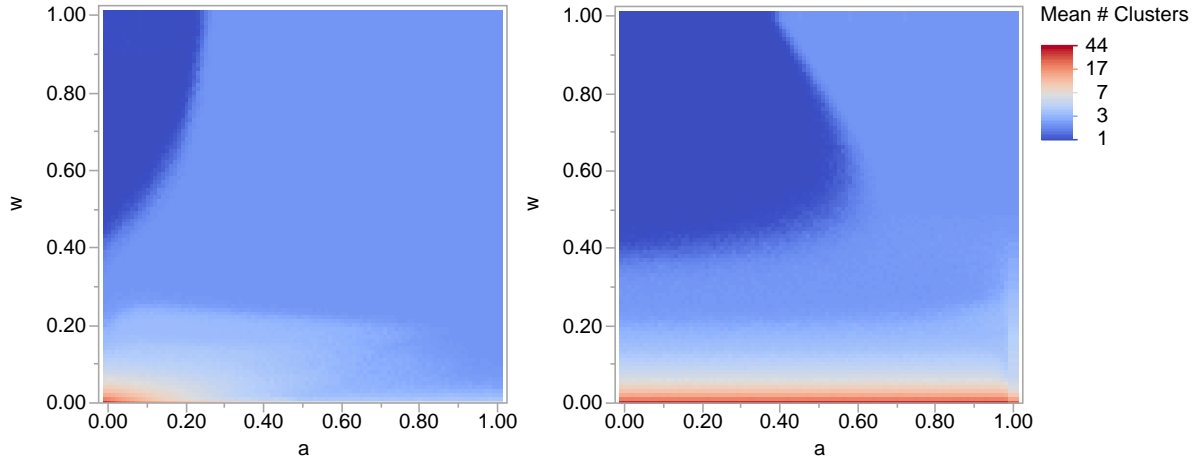


Figure 25. Mean number of clusters that form in the MC model (left) and MIF model (right)

of an agent over time. When $a = 0.25$, phase 1 is strong enough to form a single cluster, and phase 3 does not occur. When $a = 0.5$, all three phases occur: initially, extreme opinions are pulled toward the center, then clusters consolidate and are separated resulting in bi-polarization. This outcome is not inevitable for these parameters but depends upon minor variations in the initial conditions. When $a = 0.75$, the effect of phase 1 is far weaker and the range of opinions barely diminishes before phase 3 begins. Similar dynamics exist for other values of w but with more clusters forming. Figure 27 shows an individual run for $w = 0.15$ and $a = 0.5$ in which the three phases can be observed in the formation of four clusters.

The time for a single run of the MIF model had a mean of 1.52 seconds and a median of 0.85 seconds, with individual runs ranging from 0.08 to 806 seconds. The time for a single run of the MC model has a mean of 12.0 seconds and a median of 11.4 seconds, with individual runs ranging from 4.38 seconds to 235 seconds. All instances of the mean time to compute being greater for the MIF model than the MC model have parameters $a \geq 0.95$ and $w \leq 0.25$. In this region of the parameter space, the MIF model converged slowly.

The SWN variant was also run 100 times for each set of parameters with a varied

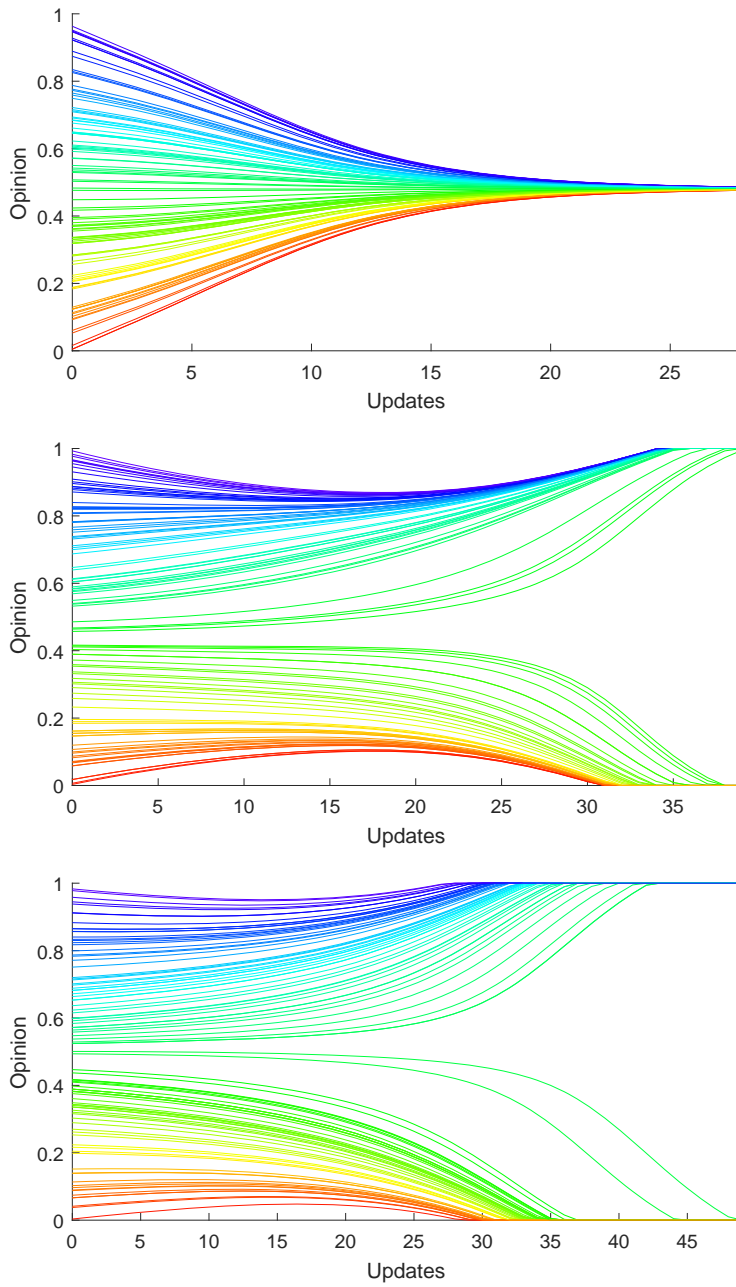


Figure 26. Individual model runs where $w = 0.5$ and $a = 0.25$ (top), $a = 0.5$ (middle), and $a = 0.75$ (bottom)

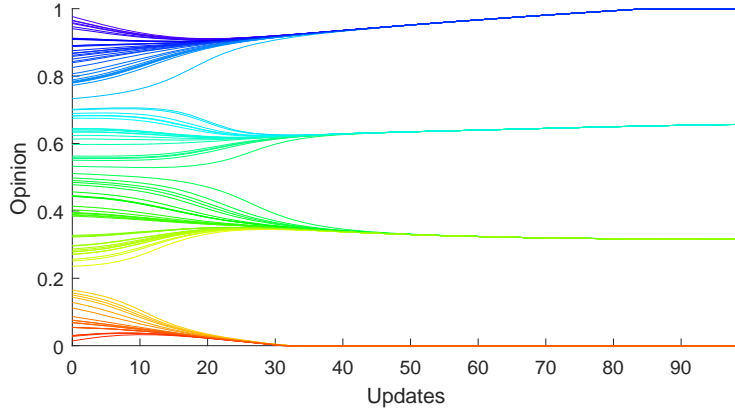


Figure 27. Individual model run where $w = 0.15$ $a = 0.5$

from 0 to 1 at increments of 0.01, w varied from 0.01 to 1 at increments of 0.01, and p varied between values $\{0, 0.05, 0.1\}$. Other parameters were fixed: $N = 100$, $k = 0.2$, $c = 8$. The SWN model may not converge to clusters in the same manner as the base model as a result of individual agents experiencing differing contexts, so the model is stopped when an update would result in no agent changing their opinion by more than $\frac{w}{100}$. This is intended to capture the bulk of the clustering behavior in the model to enable comparison with the MC model and between parameters, as only small adjustments relative to group width are being made. However, it may not fully capture long-run behavior that may take hundreds of thousands of updates to reach. The MC model SWN variant is stopped when the range of agent opinions decreases below 0.01, two clusters have formed with identical opinions, or after 1500 updates (an average of 15 per agent), whichever occurs first. Salzarulo (2006) used 1500 updates as the stopping criterion for the MC model, so it is used here as well.

Figure 28 shows the mean normalized variance in opinions at the end of a model run. At initialization, opinions are taken from the uniform distribution in $[0, 1]$, which has a variance of $\frac{1}{12}$. The relative variance, then, is 12 times the observed variance of opinions. This allows observation of under what conditions opinion variance increased or decreased and by how much. The outcome of complete bi-polarization into equal-

sized groups with opinions at the extremes has the maximum normalized variance of 3, while the consensus outcome has the minimum normalized variance of 0. This method was also used by Salzarulo (2006) in examining the MC model’s behavior in a SWN. Figure 29 shows results over the same parameter set using the MC model.

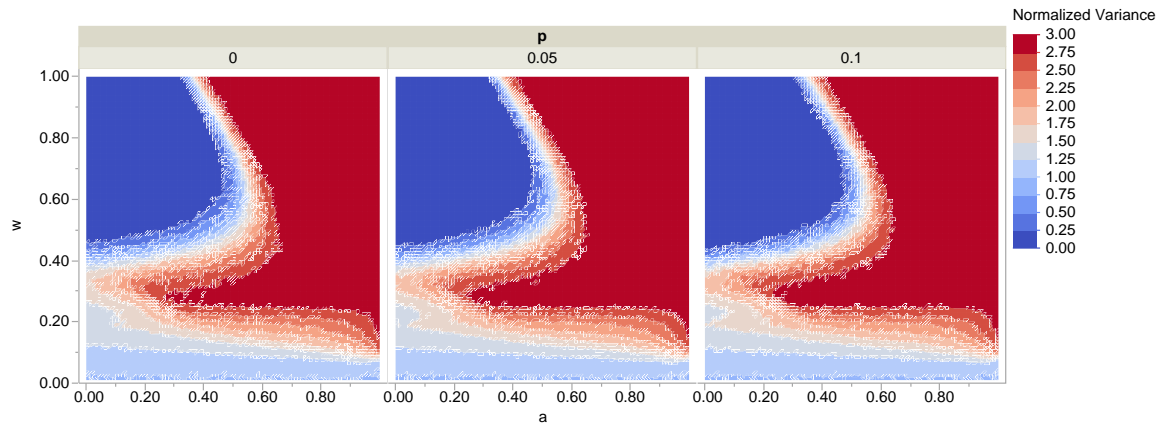


Figure 28. Normalized opinion variance for MIF model using a SWN

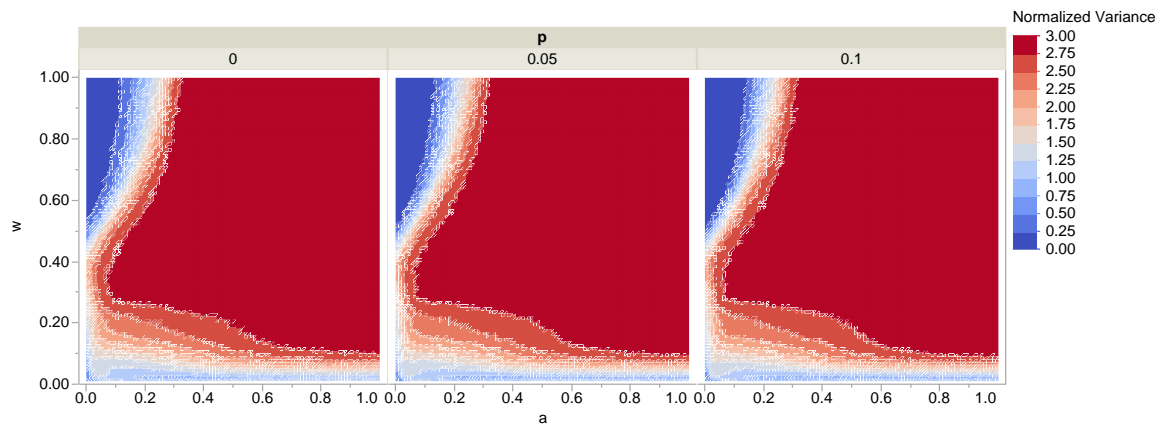


Figure 29. Normalized opinion variance for MC model using a SWN

There is very little change observed between the ring network ($p = 0$) and various random small-world networks ($p = 0.05, p = 0.1$) in either the MIF or MC model. Similar outcomes were reported by Salzarulo (2006). In the MIF (Figure 28), bi-polarization is not a common outcome in a small-world network when $a < 0.4$. In the range $0.4 \leq a \leq 0.65$, the probability of that outcome is highly contingent upon

w . As w increases, the change required in a to move from a region of guaranteed consensus to a region of guaranteed bi-polarization decreases significantly. The MC model generates bi-polarization for nearly all of the parameter space. This is a result of the scaling factor decreasing the magnitude of inter-group pressure to a level comparable with the magnitude of intra-group pressure.

A closer examination of data for model runs where $w = 0.5$ showcases the impact of the scale modifier applied in the MIF model. It seems appropriate that a group width of $w = 0.5$ should result in consensus when a is low; all agents are within opinion 0.5 of those agents with opinion 0.5. It also seems appropriate that bi-polarization should occur when a is high; agents with moderate opinions should be pushed away from whichever side exerts greater force while w is too large for intermediate clusters to exist. Both of these outcomes are observed in both the MC and MIF models. However, it seems reasonable for the region in which $a \approx 0.5$ to be unstable as both competing forces should approximately cancel one another out. This is exactly what is observed in the central region of the MIF model (see Figure 28). This unstable region in the MC model occurs for $a < 0.2$ (see Figure 29) because the inter-group pressure is dominant in that formulation of prototypicality.

4.5 Discussion

One of the strongest aspects of the MC model is its capability to specify the relative impact of intra- and inter-group components of prototypicality. This gives the MC model the flexibility to model situations in which only intra-group pressures apply, situations in which only inter-group pressures apply, or anything in between. However, the implementation of the MC model muddled the interpretation of the parameter a that defined the relative impact of each component and only implemented positive influence, although source selection yielded results that might be expected

from a negative influence model.

The MIF model strengthens this capability of the MC model. The MIF model addresses the imbalance of scale between the intra- and inter-group components of prototypicality calculation. This imbalance has a real practical effect. If one were to measure the relative impact of each type of pressure experimentally, it would not be possible to parameterize the relevant MC model for validation. The MIF model could be parameterized in this way. This is an area to be explored in more detail in future research with this model.

The desirable clustering behavior observed in the MC model remains in the MIF model. The MC model deterministically selects source agents based on prototypical opinion calculations, which serves to disconnect the social network and allow for the generation of weak diversity. The MIF model uses all connected agents as sources for the influence field, reconnecting the social network and instead using negative influence to generate weak diversity.

The computation required in the MIF model is reduced to a single calculation of $\frac{\delta}{\delta x}P(x, X)$ rather than a costly search for local maxima of $P(x, X)$ as in the MC model. Using the described criteria for determining convergence of the model and finding local maxima on the prototypicality curve to an accuracy of 0.01, the MIF model reduced the mean computation time over the entire parameter space of the MC model by 87% and the median time by 81%. This eases the computational burden of implementing meta-contrast and makes larger models with more agents computationally feasible.

The MIF model is a first step toward building a more comprehensive model of opinion dynamics. The construction of the MIF model makes extension using additional components possible in a way not easily done with other opinion dynamics models. Prototypicality is viewed as a utility function, and the MIF model uses its

gradient as an influence field. Additional components can easily be added to the utility function, with corresponding gradients added to the influence field. Thus, future research can expand on the MIF model by adding other sources of utility that motivate opinion change, each contributing another competing force into the influence field.

The MIF model, like the MC model, is fundamentally rooted in the meta-contrast principle of self-categorization theory. While Turner & Oakes (1986) argued against the distinction between normative and informational influence, there is evidence that informational influence should be modeled separately (Isenberg 1986). An area for future research is to combine this normative model with a model of informative influence such as the Argument-Communication Theory of Bi-polarization (ACTB) model of Mäs & Flache (2013). This could yield a more complete model of opinion dynamics.

4.6 Conclusion

The meta-contrast (MC) model is heavily rooted in social science theory. Its behavior is characterized by weak diversity — sustained clusters with homogeneous opinions within clusters. It has nonetheless remained unused in the opinion dynamics modeling field, likely because it has a high computational cost and because it has been dismissed as yielding results identical to the social judgment model.

The meta-contrast influence field (MIF) model introduced herein breathes new life into the MC model and makes significant improvements by using a continuous basis of influence derived from the prototypicality gradient, addressing the relative scale of intra- and inter-group contributions to opinion prototypicality, and reducing computation time by a mean of 87% over the entire parameter space. The resultant model has a closer tie to reality and is far more usable while retaining all desirable qualitative behavior.

By moving to a continuous basis of influence the MIF model, unlike the MC model, becomes a model that implements both positive and negative influence. The parameter a controls the relative magnitude of intra- and inter-group forces so that experimentally-derived estimates can be used to model a given situation. This allows the MIF model to be more interpretable, as the parameters have clear meaning. The MIF model maintains the desirable behavior of the MC model; when $a = 0$ weak diversity is generated without using negative influence or disconnecting the social network. These improvements also come with significant savings in computation.

The MIF model alone is a significant improvement on its own, but the strongest impact of this research comes from the ability for this model to be extended and combined with other models. Influence in the MIF is a sum of forces exerted by other agents. Additional forces can literally be simply added to this sum. The modular construction of the MIF model makes extension using additional components possible in a way not easily done with other opinion dynamics models, opening an avenue for extension and modification of the MIF to incorporate other forces that motivate influence and opinion change.

V. Paper #3: Generating Strong Diversity with the Meta-contrast Influence Field Model

5.1 Introduction

Models of social influence, known as opinion dynamics models, have long sought an explanatory mechanism for the diversity of opinions observed in society (Abelson 1964; Axelrod 1997; Flache et al. 2017). Models that generate diversity between, but not within, groups have been criticized for making assumptions beyond what social scientific evidence justifies (Mäs et al. 2010). The harder problem of generating diversity both between and within groups has been solved by a single known model, the Influence, Susceptibility, and Conformity (ISC) model, which relies heavily upon randomness and heterogeneity to achieve this goal (Duggins 2017).

While generating diversity of opinions in such a way is a worthwhile goal of its own, opinion dynamics models are inherently forms of what Epstein (1999) calls generative social science. Within this paradigm, the goal of modeling is to provide a set of rules that generate a desired complex behavior. Parsimony is key to this type of modeling, as a complex answer does little to explain the underlying mechanisms driving behavior.

The present paper introduces the meta-contrast influence field with local repulsion (MIF-LR) model, a parsimonious model of opinion dynamics that can generate a wealth of opinion distributions. It is rooted in theories from the social scientific literature, specifically self-categorization theory (Turner et al. 1987) and uniqueness theory (Fromkin & Snyder 1980). Self-categorization theory is used to define the rules of cluster formation, while uniqueness theory motivates local repulsion amongst agents to maintain diversity within clusters.

The resulting opinion distributions suggest that individuals' competing drives to

conform within in-groups, appear unique within those in-groups, and differentiate between in-groups and out-groups is sufficient to explain opinion diversity in a small-world network. This is further applied to a political science topic that has generated much attention: partisan polarization of society. We are able to generate this polarization by simply increasing the randomness of connections within a social network, proving the generative sufficiency of this mechanism of polarization.

5.2 Background

“The formation of persistent opinion clusters is such a difficult puzzle that all attempts to explain them had to make assumptions that are difficult to justify by empirical evidence.” (Mäs et al. 2010, p. 3)

A mathematical model that explains the diversity of opinions in the populations has been an elusive goal of opinion dynamics modelers. The earliest continuous opinion dynamics models were based upon repeated averaging of the opinions in one’s social network based upon pair-wise social power (French 1956; Harary 1959). These linear models have been proven to converge within a broad set of conditions that replicate most real-world social networks (Abelson 1964; DeGroot 1974; R. L. Berger 1981).

Abelson (1964) insightfully identified three ways in which diversity might be obtained in a continuous opinion dynamics model, each of which have been applied in modern models. First, the social network might not be connected, in which case each group would converge separately. Second, negative influence may exist that pushes an individual’s opinion away from another’s opinion. Third, contact rates and pair-wise effect rates may change over time.

Non-linear opinion dynamics models have been developed in the last 20 years that make use of Abelson’s suggestions. These all follow his third suggestion; the rates of

contact and/or effect between pairs of individuals varies over time.

Bounded confidence models (Deffuant et al. 2000; Hegselmann & Krause 2002) posit that only opinions similar to one's own provide influence. This makes use of Abelson's first suggestion by gradually breaking the connectivity of the social network into groups with a width defined by the range of opinions an individual considers meaningful. These models generate what Duggins (2017) defines as weak diversity — within a group, opinions converge to consensus.

The relative agreement model (Deffuant et al. 2002) takes an alternative approach to disconnecting the social network over time. Individuals update their confidence as well as their opinions over time based upon interactions with others. With time, individuals gain sufficient confidence in their opinions to decrease the set of opinions that may impact their own, gradually disconnecting the network. This also generates weak diversity of opinions.

An extension of the bounded confidence model is the social judgment model (Jager & Amblard 2005). This adds a region within which Abelson's second suggestion holds. As in the bounded confidence models, opinions similar to one's own induce positive, or attractive, influence. To this, Jager & Amblard add a region in which, if two individuals' opinions are sufficiently different, the resultant influence is negative, or repulsive. This generates weak diversity of opinions, and if two or more clusters arise, two clusters move to the most extreme positions.

Many other models have been developed that implement similar rules in varying ways. Others have cataloged them extensively (see Flache et al. 2017; Sîrbu et al. 2017), and they all have in common that they generate weak diversity of opinions with one exception. Furthermore, they have been criticized for relying upon assumptions that are not well supported (Mäs et al. 2010). In particular, any formulation that completely disconnects a social network fails to reflect the real world.

5.2.1 Influence, Susceptibility, and Conformity (ISC) Model

The Influence, Susceptibility, and Conformity (ISC) model is the first known opinion dynamics model to have generated strong diversity (Duggins 2017), or diversity both between and within clusters. Agents in this model have heterogeneous values for internal, psychologically-based parameters of susceptibility, conformity, tolerance, and social reach. Agents are connected to one another by proximity along with some random rewiring to form a small-world social network in a manner similar to that of Watts & Strogatz (1998). At each interaction, an agent initiates a conversation within its social network that occurs in random order. Within this conversation, agents modify their expressed opinion toward the average opinion that has been expressed so far, with dynamics varying according to their internal parameters. The resultant influence may be positive or negative with magnitude dependent upon the distance between the average expressed opinion and the target agent's opinion along with internal parameters. Negative influence occurs when the difference between these opinions is large and is a function of that distance. This causes negative influence to have the potential to be much stronger than positive influence. This can generate a diverse set of opinion distributions that are sustained over time.

Duggins (2017) deserves credit for identifying a model that can create strong diversity, but it strays significantly from the principle of parsimony in order to do so. Both heterogeneous parameter values and the dynamics of a conversation are defined by randomness. It is therefore unclear how consistently a distribution may be generated by a given set of parameters. These processes are based in social science and well justified, but a more parsimonious model is desirable if it can also generate strong diversity.

5.2.2 Meta-contrast Influence Field (MIF) Model

The meta-contrast influence field (MIF) model (see Chapter IV) serves as a framework that allows a modeler to add potentially conflicting forces of influence to an agent's behavior. The base model uses the prototypicality function of the meta-contrast model (Salzarulo 2006) as a utility function and assumes that opinions will be updated in accordance with the derivative of the utility function. This derivative, then, serves as the net force acting upon the point of the agent's opinion as a result of a field emanating from all others within their social network.

These forces take two forms, the intra-group and inter-group components, representing components of the meta-contrast principle of self-categorization theory (Turner et al. 1987). The intra-group component represents the agent's desire to categorize themselves and their social network into in-groups and out-groups. It causes opinions common to a cluster of others to be perceived as prototypical (the in-group), as well as opinions far from such a cluster (the out-group). The inter-group component represents the agent's desire to distinguish their own in-group from perceived out-groups. It increases prototypicality of opinions distant from observed opinions, which has the effect of pushing the opinions perceived to be prototypical of each group apart from one another.

In the MIF formulation, the intra-group component results in both positive and negative influence while the inter-group results only in negative influence. The MIF model, then, is an example of Abelson's second and third suggestions; it includes negative influence and the forces change over time. There is some controversy regarding the empirical support for negative influence. A recent study by Takács et al. (2016) found no evidence of negative influence in their study and cautioned against relying upon it as a model foundation. Others have found strong evidence of negative influence (Hilmert et al. 2006). On a macro scale, however, group members consis-

tently change behavior to signal a group identity consistent with their in-group and distinctive from out-groups (J. Berger & Heath 2007, 2008). This is consistent with the meta-contrast principle.

5.2.3 Drive for Individualization

One interesting finding by Takács et al. (2016) was that negative influence was observed between individuals with closely aligned opinions. This would be predicted by uniqueness theory, which postulates that over-similarity to others can generate a negative emotional reaction (Fromkin & Snyder 1980). Chan et al. (2012) found support for an interaction between desire to both belong to an in-group and be unique within that group.

Formulations including a desire for uniqueness are largely absent from the opinion dynamics modeling literature. The one exception appears to be that of the Durkheimian opinion dynamics model of Mäs et al. (2010). This is inspired by Durkheim’s theory of social integration (Durkheim 1933), which similarly argued that society was formed by contrasting forces to conform and individualize. This formulation fundamentally uses a repeated averaging model augmented with an adaptive noise component. This adaptive noise is a random fluctuation in opinions drawn from a normal distribution with mean 0 and with standard deviation

$$s \sum_{j=1}^N e^{-|x_i - x_j|}$$

where s is a model parameter, N is the number of agents in the model, and x_i is the opinion of agent i . In this way, the noise parameter has greater variation when agents were densely packed than when they were dispersed. This results in clusters forming and dissolving without stabilizing as time progresses. A drive for individualization is therefore useful for explaining cluster generation, but randomness in its formulation

yields temporal instability.

5.3 Method

The meta-contrast influence field local repulsion (MIF-LR) model is based upon the meta-contrast influence field (MIF) model. The MIF model and its rationale is described in detail in Chapter IV. It takes a variant of the prototypicality function used in the meta-contrast (MC) model (Salzarulo 2006), which calculates the perceived prototypicality of opinion x based upon observations of a set of opinions X , and redefines it as a utility function. This modification of the MIF model adds a non-prototypicality element to that utility function representing a drive for individualization, which makes it a broader function of opinion desirability than simple prototypicality.

NetLogo 6.0 (Wilensky 1999) code implementing the MIF-LR model is provided in Appendix V.

The system of interest is defined by N agents each possessing opinions $x_i \in [0, 1]$ for $i = 1, \dots, N$. These opinions are initialized to random values within the defined range. Agents are placed within a small world network generated using the method of Watts & Strogatz (1998). All agents are placed in a ring network and connected with c nearest neighbors in each direction ($2 \cdot c$ total), and connections are rewired to another agent chosen at random with probability p .

The MIF model's prototypicality function contained two components: an intra-group (d_{intra}) and an inter-group (d_{inter}). Taking into account relevant scalar multi-

pliers, these have the values

$$d_{\text{intra}}(x, X) = -\frac{\sum_{i=1}^{|X|} (x - x_i)^2 \cdot e^{-\frac{(x-x_i)^2}{w^2}}}{\sum_{i=1}^{|X|} e^{-\frac{(x-x_i)^2}{w^2}}}$$

$$d_{\text{inter}}(x, X) = \frac{\frac{w^2}{e+e} \cdot \sum_{i=1}^{|X|} (x - x_i)^2 \cdot \left(1 - e^{-\frac{(x-x_i)^2}{w^2}}\right)}{\sum_{i=1}^{|X|} \left(1 - e^{-\frac{(x-x_i)^2}{w^2}}\right)}$$

where X is the set of observed opinions and w is a parameter defining group width.

The MIF-LR extends the prototypicality function with an extra component: a drive for individualization (d_{indiv}).

$$d_{\text{indiv}}(x, X) = -\frac{\frac{w^2}{e} \cdot \sum_{i=1}^{|X|} e^{-\frac{(x-x_i)^2}{(v \cdot w)^2}}}{\sum_{i=1}^{|X|} e^{-\frac{(x-x_i)^2}{w^2}}}$$

where $v \in (0, 1)$ defines the relative width of a repulsive force stemming from a desire to be unique. The multiplier $\frac{w^2}{e}$ is used to equalize the relative scale of d_{intra} , the numerator of which has a maximum value of $\frac{e}{w^2}$, with that of d_{indiv} , the numerator of which has a maximum value of 1.

These components combine to define the perceived desirability of opinion x based upon observations of the set of opinions X available to an agent. Letting a be the proportion of desirability attributable to inter-group dynamics and b be the proportion of intra-group desirability attributable to a drive for individualization, the desirability of opinion x is

$$D(x, X) = a \cdot d_{\text{inter}}(x, X) + (1 - a) \cdot (b \cdot d_{\text{indiv}}(x, X) + (1 - b) \cdot d_{\text{intra}}(x, X)).$$

The influence field generated at x by the perception of desirability is defined by the derivative of the desirability function

$$\frac{\delta}{\delta x}D(x, X) = a \cdot \frac{\delta}{\delta x}d_{\text{inter}}(x, X) + (1-a) \cdot \left(b \cdot \frac{\delta}{\delta x}d_{\text{indiv}}(x, X) + (1-b) \cdot \frac{\delta}{\delta x}d_{\text{intra}}(x, X) \right)$$

with component derivatives defined by

$$\begin{aligned} \frac{\delta}{\delta x}d_{\text{inter}}(x, X) &= \frac{2}{e + e^{\frac{w^2-1}{w^2}}} \cdot \left(\frac{w^2 \sum_{i=1}^{|X|} (x - x_i)(1 - \mu(x, x_i)) + \sum_{i=1}^{|X|} (x - x_i)^3 \mu(x, x_i)}{\sum_{i=1}^{|X|} (1 - \mu(x, x_i))} \right. \\ &\quad \left. - \frac{\sum_{i=1}^{|X|} (x - x_i) \mu(x, x_i) \cdot \sum_{i=1}^{|X|} (x - x_i)^2 (1 - \mu(x, x_i))}{\left(\sum_{i=1}^{|X|} (1 - \mu(x, x_i)) \right)^2} \right) \\ \frac{\delta}{\delta x}d_{\text{intra}}(x, X) &= \frac{-2}{w^2} \left(\frac{w^2 \sum_{i=1}^{|X|} (x - x_i) \mu(x, x_i) - \sum_{i=1}^{|X|} (x - x_i)^3 \mu(x, x_i)}{\sum_{i=1}^{|X|} \mu(x, x_i)} \right. \\ &\quad \left. + \frac{\sum_{i=1}^{|X|} (x - x_i) \mu(x, x_i) \cdot \sum_{i=1}^{|X|} (x - x_i)^2 \mu(x, x_i)}{\left(\sum_{i=1}^{|X|} \mu(x, x_i) \right)^2} \right) \\ \frac{\delta}{\delta x}d_{\text{indiv}}(x, X) &= \frac{-2w^2}{e} \cdot \left(\frac{\sum_{i=1}^{|X|} (x - x_i) \mu(x, x_i) \cdot \sum_{i=1}^{|X|} \mu^*(x, x_i)}{w^2 \cdot \left(\sum_{i=1}^{|X|} \mu(x, x_i) \right)^2} - \frac{\sum_{i=1}^{|X|} \mu^*(x, x_i)}{v^2 \cdot \sum_{i=1}^{|X|} \mu(x, x_i)} \right). \end{aligned}$$

Agents in the MIF-LR model update their opinions much like the agents in the MIF model. Using a Synchronous Target (∞, ∞) schedule, per the SAS taxonomy

(see Chapter III), all agents $i = 1, \dots, N$ update their opinions according to

$$x_i \leftarrow x_i + k \cdot \frac{\delta}{\delta x} D(x, X)$$

where k is a convergence parameter that should be set low enough to approximate continuous time. If k is set too high, the modeler will observe agents overshooting their preferred opinion and oscillating with each update.

5.3.1 Effect of adding a drive for individualization

The drive for individualization, d_{indiv} , serves as a modifier to the original intra-group effect of meta-contrast, d_{intra} . They share a denominator, so they may be compared using the numerator alone without altering the shape of the effect in the full model. Figure 30 shows the shapes of each component and their sum with $b = 0.5$, to keep their scales equal, and $v = 0.10, 0.25, 0.5$, and 0.75 as noted above each plot. Without the drive for individualization, d_{intra} creates local maxima at $|x - x_i| = 0$ and at $|x - x_i| = 1$. Considering the derivative, this generates positive influence for $0 < |x - x_i| < w$ and negative influence for $|x - x_i| > w$. Without the d_{intra} component, d_{indiv} generates a local maximum at $|x - x_i| = 1$. Its derivative generates negative influence for all $|x - x_i| > 0$, although its magnitude diminishes rapidly.

The sum of both components shows a more interesting pattern. For low to moderate values of $v \lesssim 0.6$, the sum generates a local maximum at some point $0 < |x - x_i| < w$ and another at $|x - x_i| = 1$. This has the effect of generating negative influence for those with very similar opinions, positive influence for those with moderately dissimilar opinions, and negative influence for those with very dissimilar opinions. These three groups correspond with in-group members whose similarity causes a desire to appear unique, other in-group members, and out-group members respectively. The lower the value of v , the stronger the repulsive force against sim-

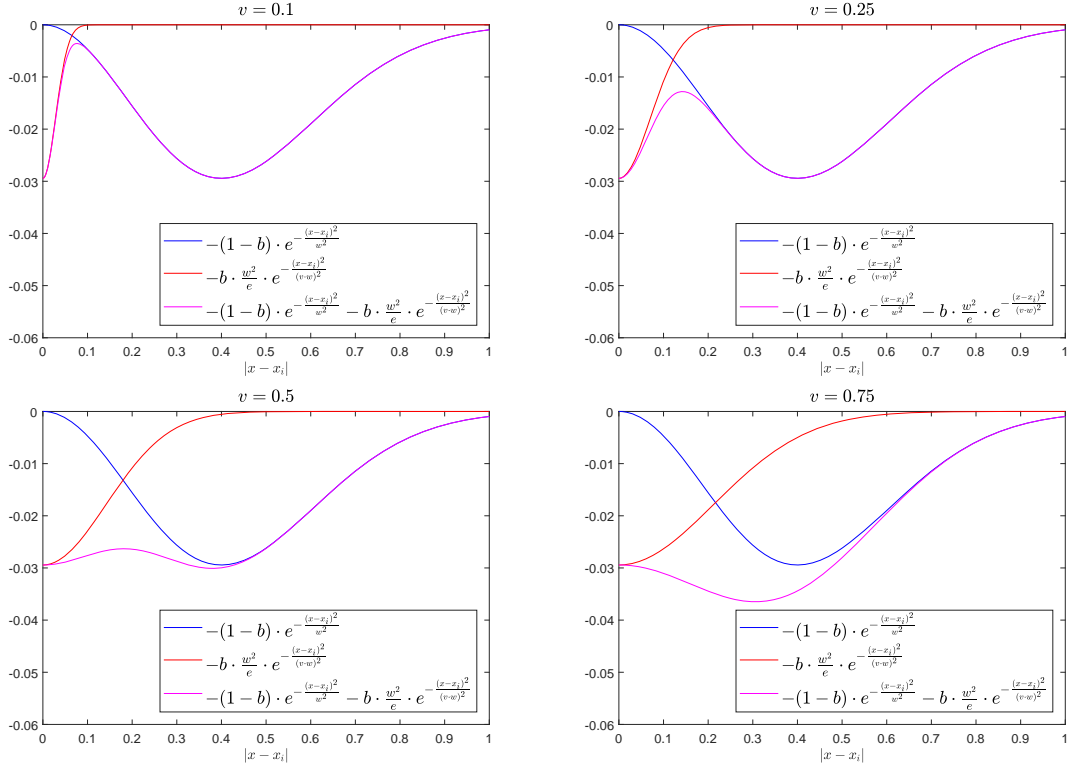


Figure 30. Shape of numerators of d_{intra} , d_{indiv} , and their sum for $w = 0.4$, $b = 0.5$

ilar others but the narrower the range of this negative influence. For higher values of v , the shape is similar to that of d_{intra} alone, except that the local maximum at $|x - x_i| = w$ is moved toward 0.

5.4 Results

The desired outcome of the MIF-LR model is the capability of generating a broad range of opinion distributions that are stable over time and consistently generated by a particular set of parameters. To demonstrate this capability, parameter sets were identified that generate unimodal, bimodal, trimodal, and quatrmodal distributions. This outcome was achieved while only varying a single variable: w . The other parameters are set to $a = 0$, $b = 0.50$, $c = 8$, $N = 1000$, $p = 0.05$, $v = 0.25$. Each model was run for 2000 updates.

The results shown in Figure 31 come from setting $w = 0.45$. The distribution on the left shows the 1000 opinions resulting from a single model run for 2000 updates. The distribution on the right shows the 100,000 individual opinions from 100 model runs for 2000 updates. A clear unimodal distribution has formed in the single run, and that distribution is reflected in the multi-run behavior showing that the outcome is repeatable with those parameters. The long tails appear to have created very small clusters at the extreme opinion values of 0 and 1, with less than 0.1% of opinions at each of those values.

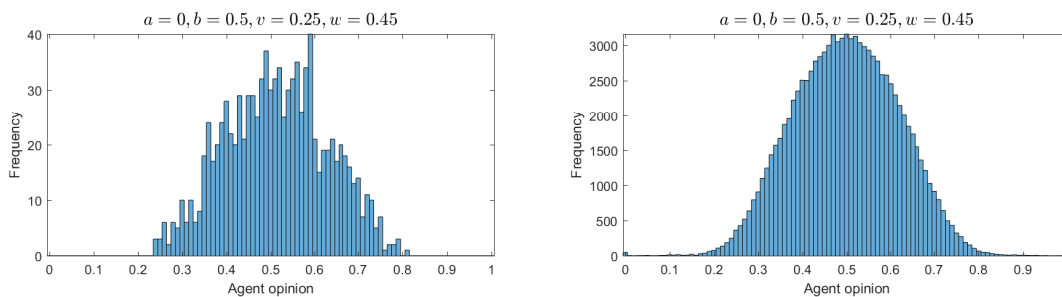


Figure 31. Distribution of opinions after 2000 updates, single replicate (left) and 100 replicates (right) $w = 0.45$

The results shown in Figure 32 come from setting $w = 0.23$. By decreasing the group width, a bimodal distribution has formed in the single run. Again, that distribution is reflected in the multi-run behavior. Approximately 3.5% of agents over 100 runs are clustered at each extreme opinion as the extreme tails of group cannot shift past 0 or 1. This point density interrupts the smooth distribution of opinions near the poles slightly as the local repulsion there adds up.

The results shown in Figure 33 come from setting $w = 0.15$. Further decreasing the group width results in a trimodal distribution being formed in the single run. Again, that distribution is reflected in the multi-run behavior. Approximately 5% of agents over 100 runs are clustered at each extreme opinion. The density at those points adds further disruption to the distribution of opinions near the poles.

The results shown in Figure 34 come from setting $w = 0.12$. A quatrmodal

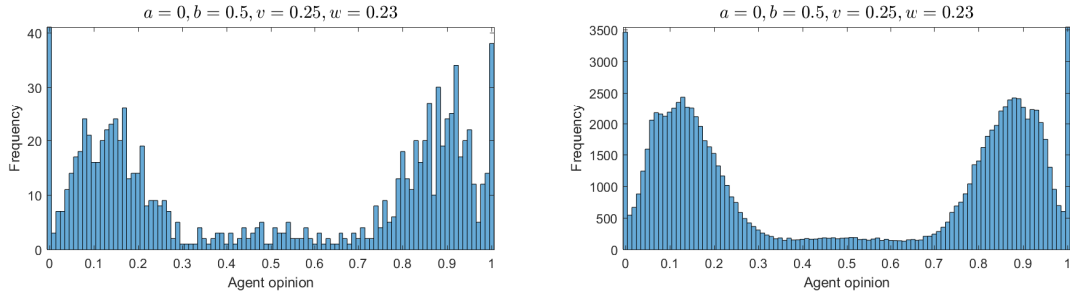


Figure 32. Distribution of opinions after 2000 updates, single replicate (left) and 100 replicates (right) $w = 0.23$

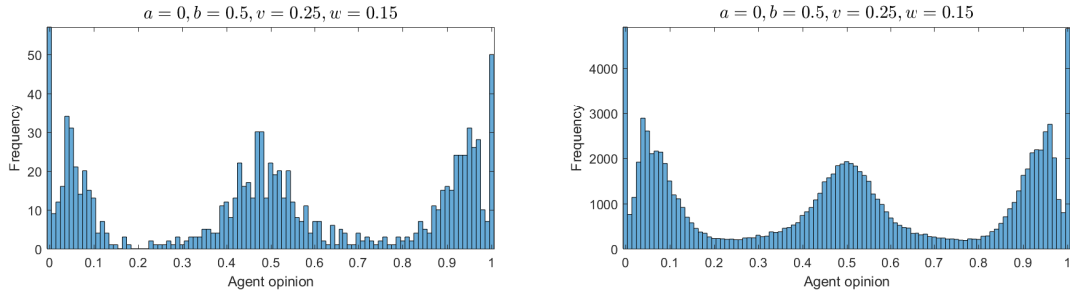


Figure 33. Distribution of opinions after 2000 updates, single replicate (left) and 100 replicates (right) $w = 0.15$

distribution has formed in the single run, with that distribution being again reflected in the multi-run behavior. Approximately 5% of agents over 100 runs are again clustered at each extreme opinion. The pattern of disrupted opinion distribution at the poles appears nearly identical to the trimodal results.

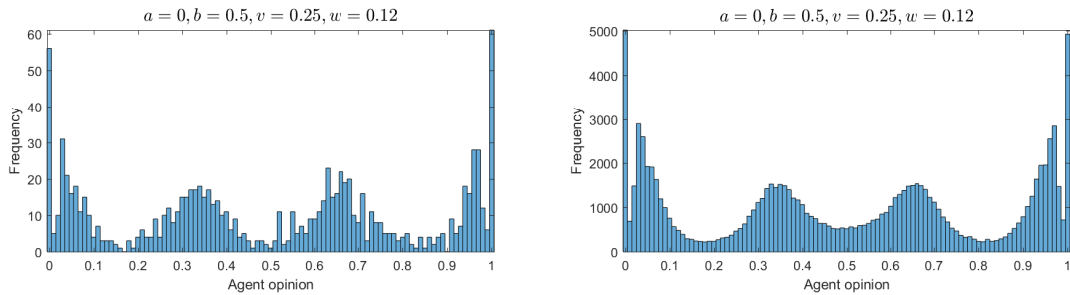


Figure 34. Distribution of opinions after 2000 updates, single replicate (left) and 100 replicates (right) $w = 0.12$

The stopping criterion of 2000 updates was chosen to ensure that sufficient time had elapsed that opinions were no longer in shift. Figure 35 shows the development

over time of the opinion distributions shown at left in Figures 31–34. Each line is drawn by one of 1000 agents with color corresponding to the opinion with which they were randomly initialized. This shows that the impact of running the model beyond 2000 updates is unlikely to significantly change distributions.

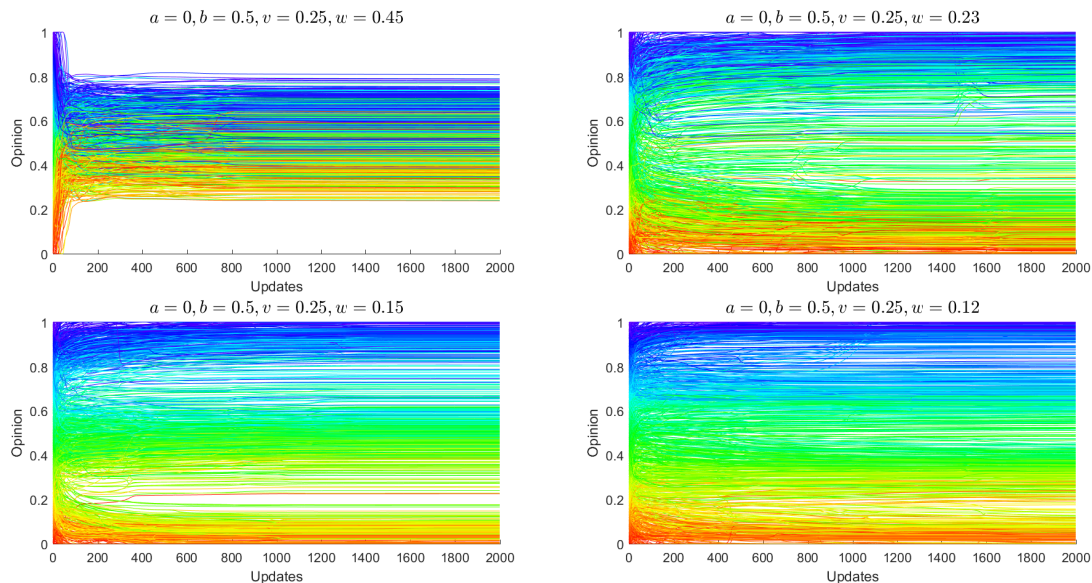


Figure 35. Single-run trajectory of opinions over 2000 updates

5.4.1 Effect of varying relative strength of individualization

The balance between d_{intra} and d_{div} yields interesting dynamics that can also shape the distribution of opinions. This balance is characterized by the parameter b . Taking the quatrmodal example above as a starting point, Figure 36 shows the distribution of opinion of 100 replicates for $a = 0, v = 0.25, w = 0.12$ and b varied between 0, 0.25, 0.5, 0.75, and 1. The stopping criterion was increased to 3000 updates for this experiment to ensure clustering had completed for the slower $b = 0$ case.

Increasing the strength of the individualization force has the effect of widening clusters, which can cause clusters to combine and decrease the number of modes observed in the final distribution. For $b \leq 0.5$ this results in a quatrmodal distribution, but $b = 0.75$ forms a trimodal distribution and pure individualization at $b = 1.0$ forms

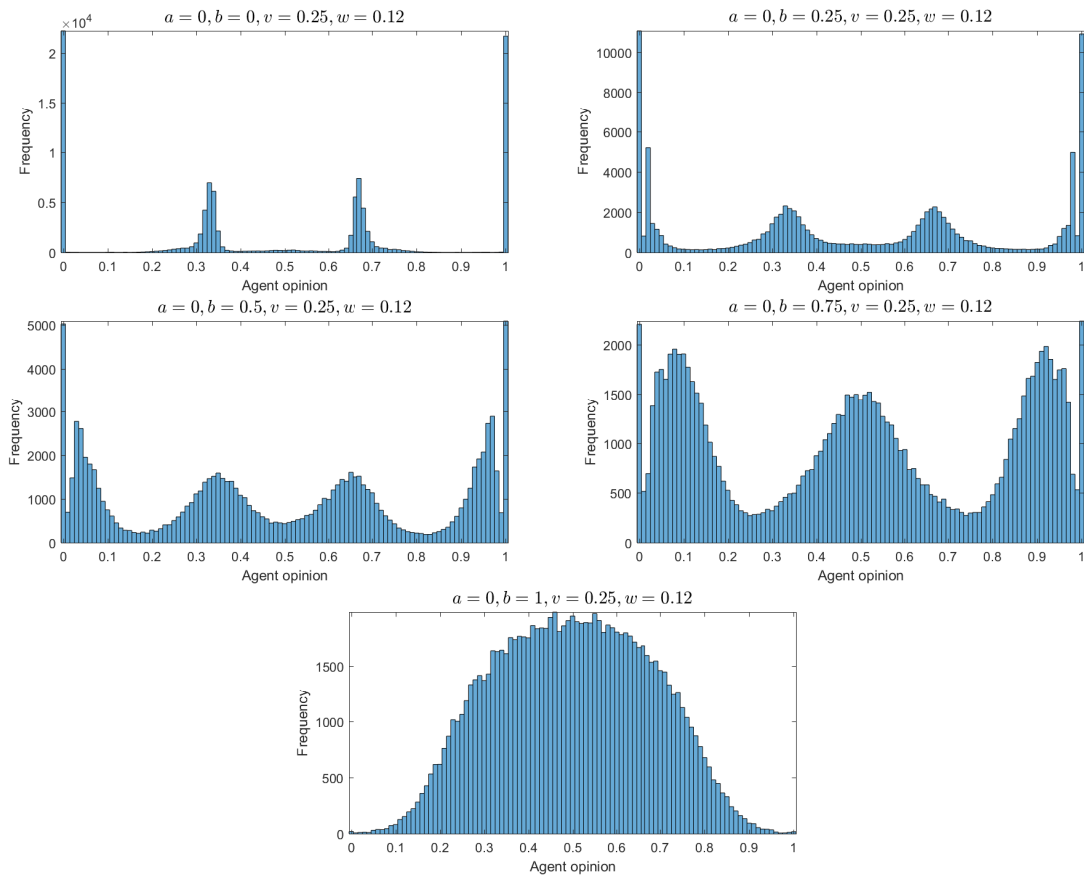


Figure 36. Distribution of opinions after 3000 updates, varying $b \in \{0, 0.25, 0.5, 0.75, 1\}$

a broad unimodal distribution. This fits with intuition; as the desire to form groups is overcome by the desire to appear unique, group membership dissolves.

5.5 Discussion

“It seems that some relaxation of the principle of parsimony might be required. Perhaps Occam’s razor should be replaced by Occam’s lawnmower.” (Abelson 1964, p. 160)

The MIF-LR model is the second known opinion dynamics model to generate sustained strong diversity of continuous opinions. The first, the ISC model (Duggins 2017), relies upon heterogeneous agents and defined distributions of parameters for every agent. When homogeneous agents are used, only weak diversity can be obtained. The MIF-LR achieves this goal with far greater parsimony and homogeneous agents. Of 8 parameters, 3 are directly related to initializing the social network (N, c, p) , 1 is a convergence parameter that must be set to a sufficiently small value (k) , leaving only 4 parameters unique to the model that impact the distribution (a, b, v, w) .

The MIF-LR model also makes little use of randomness; development of opinions is a deterministic chaotic process instead. The only random element in the model is the initial opinion set of agents. As a result, all outcomes are approximately symmetrical. To achieve non-symmetrical results, either a non-symmetrical set of initial conditions must be used, or an external force must be introduced into the model. This is not a weakness of the model but a strength; the MIF model is constructed to allow for additional internal or external forces to be added with ease.

The potential uses of the MIF-LR model in future research are myriad. From a generativist perspective, it serves as validation for self-categorization theory and uniqueness theory. Modelers may also find a parameter set that generates a particular opinion distributions of interest, such as those derived from polling data, and forecast

how societal changes in out-group aversion, social network connectedness, or in-group width might impact those distributions over time.

The MIF framework upon which the MIF-LR model is built also allows for extension of the model based on more specific and individualized situations. There are certainly forces of influence at play in society beyond self-categorization and uniqueness that may be added.

In an attempt to maintain the minimum required complexity, however, a broad set of distributions of opinions are generated while varying only w in the model. Furthermore, this parameter and its effects are readily interpreted; as the perceived width of a group decreases, a greater number of groups arise, each of which has a membership with a diverse set of opinions centered on one value.

Varying the strength of the individualization force relative to the intra-group force, in the form of the parameter b , has the effect of dissolving group membership until, for very high values of b , a broad unimodal distribution of opinions is achieved. The intra-group force and the individualization force act upon the distribution much as the competing forces of surface tension and surface friction on a pool of liquid. As surface tension overcomes surface friction, distinct droplets appear. As surface friction overcomes surface tension, a puddle forms.

Outgroup aversion, the component of the MIF model that generates pure negative influence with strength that increases with the distance between two agents' opinions, is not used in any of the examples shown. Group formation, then, is attainable with only intra-group forces that diminish with distance.

5.6 A Political Science Application: Generating Partisan Polarization

While the ability to generate a broad spectrum of opinion distributions is itself a useful contribution to opinion dynamics literature, a generative model is especially

useful if it can explain the mechanisms behind behaviors observed in the real world. A trend in opinion distributions that has received ample attention recently is an increase in political polarization, particularly in the United States but more recently worldwide. There is some disagreement in the political science field about whether polarization is restricted to political elites (Fiorina & Abrams 2008), political elites and those who identify with partisan labels (Evans 2003), or political elites and the population at large (Abramowitz & Saunders 2008).

Much of this debate centers around the definition of polarization. S. J. Abrams & Fiorina (2015) distinguishes between mass polarization, which suggests that moderates have disappeared and groups have moved toward the poles, and partisan polarization (or party sorting), which suggests that groups have become less diverse and more separated from other groups. They argue that the evidence for mass polarization is weak but that the evidence for partisan polarization is unambiguous. Pew Research Center (2012) also find that this partisan polarization has accelerated through the 1990s and 2000s. For the purposes of this application, we are more focused upon partisan polarization than mass polarization.

The mechanisms behind this increased partisan polarization is hotly debated. One popular explanation is that algorithmically-selected social media and search results create personal “filter bubbles” or “echo chambers” in which an individual’s opinions are reinforced by the filtered information fed to us via the internet (Pariser 2011). The evidence for this is weak at best. Bakshy et al. (2015) found that algorithmically ranked social media actually expose individuals to viewpoints conflicting with their own, and that personal choices rather than algorithms limited exposure to such viewpoints. Furthermore, Nguyen et al. (2014) found that those who actually consumed algorithmically-selected content accessed a broader set of content than those who did not. An alternative explanation comes from Levendusky (2013), who argues that the

increase in partisan media availability leads to polarization first in the extremes, with influences cascading to the rest of the populace. He provides experimental data that supports this conclusion. Each of these proposed mechanisms rely upon changes in the content individuals consume.

An alternative mechanism is suggested by the time frame over which increased polarization has been observed; Abramowitz & Saunders (2008) estimates that polarization began to grow in the 1970s and Pew Research Center (2012) finds this trend accelerating from the 1990s until today. These years have seen the increased availability of communication and travel to the mass public. One effect of increased availability of communication and travel, from a small-world network perspective, may be an increase in randomness in connections. A more random network has an increased probability that social network contacts of one individual are not themselves directly connected.

In the MIF-LR results displayed thus far, a small-world network is based upon each agent being initially connected to the nearest $c = 8$ agents in each direction along a ring network. These connections are rewired to random agents with probability $p = 0.05$. This value of c was chosen to allow comparison with the results of the MIF model (see Chapter IV). The probability of rewiring, p , is set to a value that Watts & Strogatz (1998) showed gained most of the path-shortening effect desired in a small-world network without significantly diminishing clustering.

The MIF-LR model allows an examination of the predicted qualitative change in opinion distributions achieved by increasing the randomness of a social network. Discussion of political polarization tends to assume three groups: liberals, conservatives, and moderates (see, for example, S. J. Abrams & Fiorina 2015). To replicate this tri-modal distribution, w is fixed to 0.18, a value that generates a tri-modal distribution with strong intermingling for $p = 0$. As before, other parameters are fixed

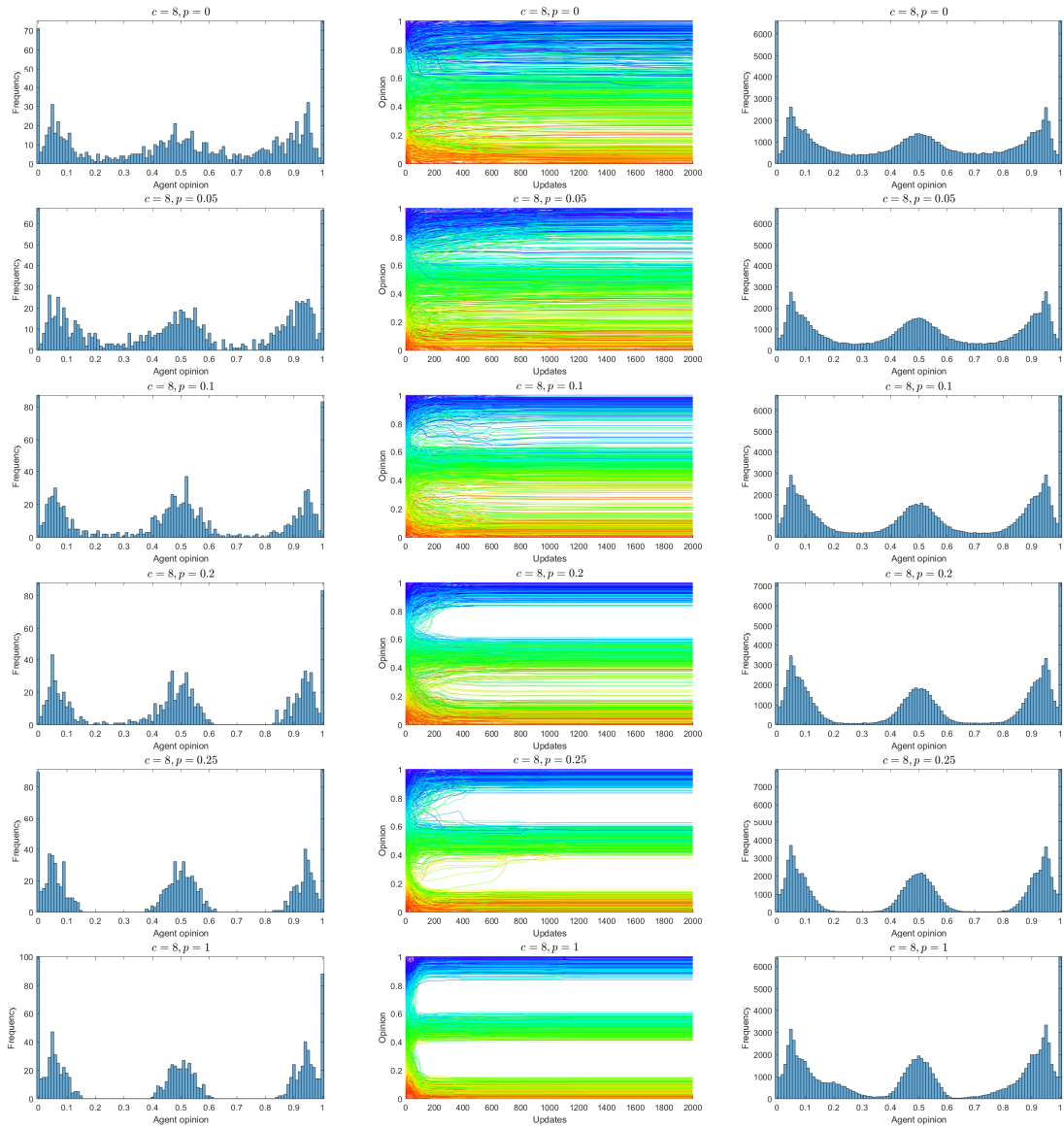


Figure 37. MIF-LR results varying network randomness: single-run distribution (left), trajectory (center), and 100-run distribution (right)

to $a = 0, b = 0.5, c = 8, N = 1000$, and $v = 0.25$. Figure 37 shows distributions resulting from varying the network from a ring network ($p = 0$), among small-world networks ($p \in \{0.05, 0.10, 0.20, 0.25\}$) up to a purely random network ($p = 1$). Partisan polarization clearly increased with even small increases in network randomness. The number of agents converging to an opinion between clusters diminishes as p increases; for $p \geq 0.25$ the clusters no longer blended together in between at all. In a purely random network, the partisan polarization effect was strong enough to dissolve the moderate cluster in 24 of 100 runs.

This demonstrates an unintuitive generative explanation based on the MIF-LR model; increased randomness of social network connections alone is sufficient to generate partisan polarization. In other words, no media bias or personalized content need be blamed for increased partisanship; the expansion of social network ties beyond the limits of geography may be at the root of this development. In a ring network as in society without communication technology, neighbors are likely to have similar opinions. This amplifies the effect of the drive for uniqueness. Randomness in the network, as induced by the ability to communicate at a distance, increases connections outside of these geographic clusters, stifling the compounding effect of the drive for uniqueness within these clusters. At the extreme, a purely random network has no geographic clusters because location in the original ring no longer holds any meaning.

5.7 Conclusion

The MIF-LR model is the second known opinion dynamics model to generate the desired behavior of strong diversity. This is accomplished without relying upon randomness or excessively complicated interaction rules. Examples of distributions with 1–4 modes are shown based on the variation of only a single model parameter.

The effect of varying the relative strength of group formation and individualization forces is also examined, with individualization widening and potentially combining clusters as its strength increases.

The MIF-LR model's capability to generate these distributions informs not only the opinion dynamics modeling field but the social sciences as well. This mathematical formulation shows that the interplay between self-categorization theory and uniqueness theory is sufficient to explain the emergence of sustained, multi-modal opinion distributions in society. This serves as an unprecedented mathematical validation of these social psychological theories.

Epstein (1999) stated that the motto of generative social science is "if you didn't grow it, you didn't explain its emergence" [p.43]. Several explanations have been offered to explain the emergent behavior of partisan polarization observed in recent decades, but we are unaware of another example satisfactorily growing that behavior. The MIF-LR demonstrated a generative explanation for this increased polarization. By increasing the randomness of a small-world network, the MIF-LR model generated increased polarization. This displays a mechanism that can generate this behavior that has thus far been ignored: communication technology has increased the randomness of individuals' social ties, diminishing the compounding effect of the desire for uniqueness of geographically clustered groups.

The MIF-LR is a strong contribution to these fields for both its modeling outcomes and its explanatory power. Being based in the MIF framework, it can easily be augmented with greater psychological realism in future research. One such augmentation that would seem to enhance the explanatory power of the model would be direct modeling of information, facts, and arguments.

VI. Conclusion

Presented herein are three distinct contributions to the literature relating to opinion dynamics models, each presented as a separate articles in Chapters III–V.

Chapter III defined the synchrony, actor type, scale (SAS) taxonomy, a means of communicating the agent schedule of an opinion dynamics model. This filled a lingering gap in communication and allowed for proposed models to be described more clearly, precisely, and concisely. The changes in emergent behavior that may be caused by altering the schedule in two often-used opinion dynamics models were demonstrated using generalized forms of those models.

Chapter IV introduced the meta-contrast influence field (MIF) model to the literature. This expanded on the meta-contrast (MC) model of Salzarulo (2006) in several ways. First, it equalized the scale of previously imbalanced inputs into the prototypicality function. This prototypicality function computed how prototypical an agent will perceive other opinions to be, and it was the backbone of the MC model. The inter-group component of that prototypicality dominated the calculation in the MC formulation. Second, it implemented a continuous basis of influence calculated from the derivative of the MC model’s prototypicality function, which computed how prototypical an agent would perceive an opinion to be. Using the derivative, in effect, created an influence field composed of competing forces acting upon the point of an agent’s opinion. Finally, as a result of this field-based formulation, the schedule is altered to allow all agents to update in synchrony based upon the information available to them. These improvements came with a mean 87% decrease in required computation time, which is a significant contribution all its own. The MIF model is a significant enhancement of the MC model, and its construction allows it to be a framework upon which more nuanced models of interpersonal influence can be built.

Chapter V represents by far the largest contribution to the field in this dissertation.

It updated the MIF model to include a drive for individualization based on uniqueness theory. By including local repulsive forces into the influence field, the meta-contrast field local repulsion (MIF-LR) model generated strong diversity of opinions. This outcome has been sought after for over 50 years since Abelson (1964) famously asked “what on earth one must assume in order to generate the bimodal outcome of community cleavage studies” [p. 153]. Examples of parameter settings that generate unimodal, bimodal, trimodal, and quattrimodal distributions are provided. An application that generates partisan polarization as an outcome of increasing random ties in social networks is also explored. These distributions are stable with respect to time and repeatable between experiments. Furthermore, this is a parsimonious model with only 8 parameters, of which only 4 define agent behavior in a manner unique to this model, 1 needs only to be sufficiently small to avoid erratic behavior, and 3 define the social network itself. It is also firmly based in social psychological literature. This model has the potential to revolutionize opinion dynamics modeling and be the gold standard in the field.

6.1 Future Research

There are many opportunities to expand upon the MIF-LR model or further explore its behavior. Two such avenues of expansion have already had some groundwork laid that suggests promising results. First, the topic of initial conditions is not addressed in Chapter V. All results are based upon an initial distribution drawn from the uniform (0,1) distribution. This is partially a result of push-back from referees in reviewing an earlier version of the paper in Chapter III when using other initial conditions, and it is partially to allow more apples-to-apples comparisons with earlier opinion dynamics models that use the same initial conditions. However, many other models show very limited robustness to deviations in initial conditions. In particu-

lar, bounded confidence models are strongly affected by initial conditions with lower variance.

Initial results of using initial conditions drawn from the normal distribution with mean 0.5 and standard deviation s are shown in Figure 38. Each histogram is based on 100 replicates. These results vary the parameter a (fixed by column) along with s (fixed by row), as negative influence between groups has the potential to increase the variance in the opinion distribution and allow clusters to form. Other parameters are fixed: $b = 0.5, c = 8, k = 0.2, N = 1000, p = 0.05, v = 0.25, w = 0.12$. This shows that a standard deviation of $s = 0.2$ is sufficient to generate clustering in this example, even with $a = 0$. By setting $a \geq 0.25$, clustering is observed for $s = 0.15$. Other early results suggest that increasing a may cause a unimodal distribution based on $w = 0.45$ to split into a bimodal distribution with modes near 0.5 on either side.

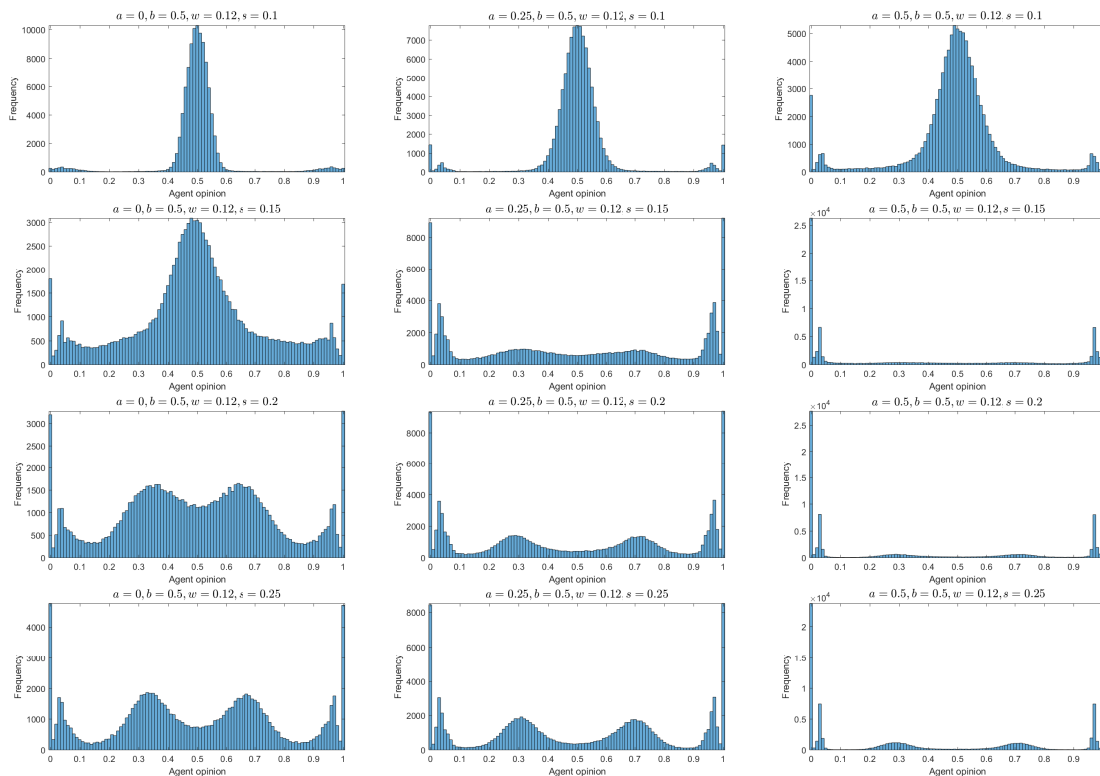


Figure 38. Distributions resulting from 100 replicates of MIF-LR model varying initial conditions

Another intriguing area for future research would be to determine a mechanism by which the urban-rural split observable in election maps can be generated. To accomplish this, agents could be placed randomly according to population density data that is available to 1 arc-second (approximately 1 square kilometer) fidelity (Columbia University 2005). NetLogo code accomplishing this task and implementing the MIF-LR model with non-random distance-based network within it is available in Appendix W, and the accompanying data file with population density data is available from the author. A method for generating a small-world network based upon this framework is not known to exist in the literature. This alone would be a useful contribution. Operating the MIF-LR model within such a network could yield very useful generative explanations for emergent behavior of interest in our society.

References

- Abelson, R. P. (1964). Mathematical models of the distribution of attitudes under controversy. In N. Frederiksen & L. L. Thurstone (Eds.), *Contributions to mathematical psychology* (pp. 142–160). New York, NY: Holt, Rinehart, and Winston.
- Abramowitz, A. I., & Saunders, K. L. (2008). Is polarization a myth? *The Journal of Politics*, *70*(2), 542–555.
- Abrams, D., Wetherell, M., Cochrane, S., Hogg, M. A., & Turner, J. C. (1990). Knowing what to think by knowing who you are: Self-categorization and the nature of norm formation, conformity and group polarization. *British Journal of Social Psychology*, *29*(2), 97–119.
- Abrams, S. J., & Fiorina, M. P. (2015). Party Sorting: The Foundations of Polarized Politics. In J. A. Thurber & A. Yoshinaka (Eds.), *American gridlock: The sources, character, and impact of political polarization* (pp. 113–129). New York: Cambridge University Press.
- Asch, S. E. (1946). Forming impressions of personality. *The Journal of Abnormal and Social Psychology*, *41*(3), 258–290. doi: 10.1037/h0055756
- Axelrod, R. (1997). The dissemination of culture: A model with local convergence and global polarization. *Journal of Conflict Resolution*, *41*(2), 203–226. doi: 10.1177/0022002797041002001
- Axelrod, R., & Tesfatsion, L. (2006). A guide for newcomers to agent-based modeling in the social sciences. In L. Tesfatsion & K. L. Judd (Eds.), *Handbook of computational economics* (Vol. 2, pp. 1647–1659). Elsevier B.V. doi: 10.1016/S1574-0021(05)02044-7

- Bakshy, E., Messing, S., & Adamic, L. A. (2015). Exposure to ideologically diverse news and opinion on Facebook. *Science*, *348*(6239), 1130–1132. doi: 10.1126/science.aaa1160
- Banks, J., Carson, J. S. I., Nelson, B. L., & Nicol, D. M. (2010). *Discrete-Event System Simulation* (Fifth ed.). Upper Saddle River, NJ: Prentice Hall.
- Berger, J., & Heath, C. (2007). Where consumers diverge from others: Identity signaling and product domains. *Journal of Consumer Research*, *34*(2), 121–134. doi: 10.1086/519142
- Berger, J., & Heath, C. (2008). Who drives divergence? Identity signaling, outgroup dissimilarity, and the abandonment of cultural tastes. *Journal of Personality and Social Psychology*, *95*(3), 593–607.
- Berger, R. L. (1981). A necessary and sufficient condition for reaching a consensus using DeGroot's method. *Journal of the American Statistical Association*, *76*(374), 415.
- Berscheid, E. (1966). Opinion change and communicator-communicatee similarity and dissimilarity. *Journal of Personality and Social Psychology*, *4*(6), 670–680.
- Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, *99*(Supplement 3), 7280–7287. doi: 10.1073/pnas.082080899
- Bonabeau, E., Dessalles, J.-L., & Grumbach, A. (1995). Characterizing emergent phenomena (2): A conceptual framework. *Revue Internationale De Systémique*, *9*(3), 347–371.
- Bonnell, T. R., Chapman, C. A., & Sengupta, R. (2016). Interaction between scale and scheduling choices in simulations of spatial agents. *International Journal of*

- Geographical Information Science*, 30(10), 2075–2088. doi: 10.1080/13658816.2016.1158822
- Bouré, O., Fatès, N., & Chevrier, V. (2012). Probing robustness of cellular automata through variations of asynchronous updating. *Natural Computing*, 11(4), 553–564. doi: 10.1007/s11047-012-9340-y
- Carley, K. M. (1991). A theory of group stability. *American Sociological Review*, 56(3), 331–354. Retrieved from <http://www.jstor.org/stable/2096108>
- Caron-Lormier, G., Humphry, R. W., Bohan, D. A., Hawes, C., & Thorbek, P. (2008). Asynchronous and synchronous updating in individual-based models. *Ecological Modelling*, 212(3-4), 522–527. doi: 10.1016/j.ecolmodel.2007.10.049
- Castellano, C., Fortunato, S., & Loreto, V. (2009). Statistical physics of social dynamics. *Reviews of Modern Physics*, 81(2), 591–646.
- Casti, J. L. (1997). *Would-Be Worlds: How Simulation is Changing the World of Science*. New York, NY: Wiley.
- Chan, C., Berger, J., & Van Boven, L. (2012). Identifiable but not identical: Combining social identity and uniqueness motives in choice. *Journal of Consumer Research*, 39(3), 561–573. doi: 10.1086/664804
- Clifford, P., & Sudbury, A. (1973). A model for spatial conflict. *Biometrika*, 60(3), 581–588. doi: 10.1093/biomet/60.3.581
- Collins, A., Petty, M., Vernon-Bido, D., & Sherfey, S. (2015). A Call to Arms: Standards for Agent-Based Modeling and Simulation. *Journal of Artificial Societies and Social Simulation*, 18(3), 2–12. doi: 10.18564/jasss.2838

- Columbia University. (2005). *Gridded Population of the World Version 3 (GPWv3)*. Retrieved from <http://sedac.ciesin.columbia.edu/data/collection/gpw-v3>
- Cornforth, D., Green, D. G., & Newth, D. (2005). Ordered asynchronous processes in multi-agent systems. *Physica D: Nonlinear Phenomena*, *204*(1-2), 70–82. doi: 10.1016/j.physd.2005.04.005
- Dandekar, P., Goel, A., & Lee, D. T. (2013). Biased assimilation, homophily, and the dynamics of polarization. *Proceedings of the National Academy of Sciences of the United States of America*, *110*(15), 5791–5796. doi: 10.1073/pnas.1217220110
- Davis, J. H. (1973). Group decision and social interaction: A theory of social decision schemes. *Psychological Review*, *80*(2), 97–125. doi: 10.1037/h0033951
- Deffuant, G., Amblard, F., Weisbuch, G., & Faure, T. (2002). How can extremism prevail? A study based on the relative agreement interaction model. *Journal of Artificial Societies and Social Simulation*, *5*(4), 1. Retrieved from <http://jasss.soc.surrey.ac.uk/5/4/1.html>
- Deffuant, G., Neau, D., Amblard, F., & Weisbuch, G. (2000). Mixing beliefs among interacting agents. *Advances in Complex Systems*, *3*(1-4), 87–98.
- DeGroot, M. H. (1974). Reaching a consensus. *Journal of the American Statistical Association*, *69*(345), 118–121.
- Duggins, P. (2017). A psychologically-motivated model of opinion change with applications to American politics. *Journal of Artificial Societies and Social Simulation*, *20*(1), 13. doi: 10.18564/jasss.3316
- Durkheim, E. (1933). *The Division of Labor in Society* (G. Simpson, Trans.). Glencoe, IL: The Free Press.

- Epstein, J. M. (1999). Agent-based computational models and generative social science. *Complexity*, 4(5), 41–60. doi: 10.1002/(SICI)1099-0526(199905/06)4:5<41::AID-CPLX9>3.3.CO;2-6
- Epstein, J. M. (2006). *Generative Social Science: Studies in Agent-based Computational Modeling*. Princeton, NJ: Princeton University Press.
- Epstein, J. M., & Axtell, R. (1996). *Growing Artificial Societies: Social Science From the Bottom Up*. Washington, DC: Brookings Institution Press.
- Evans, J. H. (2003). Have Americans' attitudes become more polarized? An update. *Social Science Quarterly*, 84(1), 71–90.
- Fatès, N. (2014). A guided tour of asynchronous cellular automata. *Journal of Cellular Automata*, 9(5-6), 387–416. doi: 10.1007/978-3-642-40867-0_2
- Fatès, N., & Chevrier, V. (2010). How important are updating schemes in multi-agent systems? An illustration on a multi-turmite model. In *Proceedings of the 9th international conference on autonomous agents and multiagent systems* (pp. 533–540). Toronto: International Foundation for Autonomous Agents and Multiagent Systems. Retrieved from <http://dl.acm.org/citation.cfm?id=1838282>
- Festinger, L. (1954). A theory of social comparison processes. *Human Relations*, 7(2), 117–140.
- Fiorina, M. P., & Abrams, S. J. (2008). Political polarization in the American public. *Annual Review of Political Science*, 11(1), 563–588.
- Flache, A., & Macy, M. W. (2011). Small worlds and cultural polarization. *The Journal of Mathematical Sociology*, 35(1-3), 146–176.

- Flache, A., & Mäs, M. (2008a). How to get the timing right. A computational model of the effects of the timing of contacts on team cohesion in demographically diverse teams. *Computational and Mathematical Organization Theory*, *14*(1), 23–51. doi: 10.1007/s10588-008-9019-1
- Flache, A., & Mäs, M. (2008b). Why do faultlines matter? A computational model of how strong demographic faultlines undermine team cohesion. *Simulation Modelling Practice and Theory*, *16*(2), 175–191.
- Flache, A., Mäs, M., Feliciani, T., Chattoe-Brown, E., Deffuant, G., Huet, S., & Lorenz, J. (2017). Models of social influence: Towards the next frontiers. *Journal of Artificial Societies and Social Simulation*, *20*(4), 2. doi: 10.18564/jasss.3521
- French, J. R. P. (1956). A formal theory of social power. *Psychological Review*, *63*(3), 181–194. doi: 10.1037/h0046123
- French, J. R. P., & Raven, B. (1959). The bases of social power. In D. Cartwright (Ed.), *Studies in social power* (pp. 150–167). Ann Arbor, MI: University of Michigan.
- Friedkin, N. E. (1986). A formal theory of social power. *The Journal of Mathematical Sociology*, *12*(2), 103–126. doi: 10.1080/0022250X.1986.9990008
- Friedkin, N. E. (1999). Choice shift and group polarization. *American Sociological Review*, *64*(6), 856–875. Retrieved from <http://www.jstor.org/stable/2657407>
- Friedkin, N. E. (2001). Norm formation in social influence networks. *Social Networks*, *23*(3), 167–189. doi: 10.1016/S0378-8733(01)00036-3
- Fromkin, H. L., & Snyder, C. R. (1980). The search for uniqueness and valuation of scarcity. In K. J. Gergen, M. S. Greenberg, & R. H. Willis (Eds.), *Social exchange* (pp. 57–75). Boston, MA: Springer US. doi: 10.1007/978-1-4613-3087-5_3

- Galam, S. (1997). Rational group decision making: A random field Ising model at $T = 0$. *Physica A: Statistical Mechanics and its Applications*, *238*, 66–80. doi: 10.1016/S0378-4371(96)00456-6
- Gardner, M. (1970). Mathematical games. *Scientific American*, *223*, 120–123.
- Gilbert, N., & Troitzsch, K. (2005). *Simulation for the Social Scientist*. New York, NY: Open University Press.
- Grimm, V., Berger, U., DeAngelis, D. L., Polhill, J. G., Giske, J., & Railsback, S. F. (2010). The ODD protocol: A review and first update. *Ecological Modelling*, *221*(23), 2760–2768. doi: 10.1016/j.ecolmodel.2010.08.019
- Groeber, P., Lorenz, J., & Schweitzer, F. (2014). Dissonance minimization as a microfoundation of social influence in models of opinion formation. *The Journal of Mathematical Sociology*, *38*(3), 147–174. doi: 10.1080/0022250X.2012.724486
- Harary, F. (1959). A criterion for unanimity in French's theory of social power. In D. Cartwright (Ed.), *Studies in social power* (pp. 168–182). Ann Arbor, MI: University of Michigan.
- Hegselmann, R., & Krause, U. (2002). Opinion dynamics and bounded confidence: Models, analysis and simulation. *Journal of Artificial Societies and Social Simulation*, *5*(3), 2. Retrieved from <http://jasss.soc.surrey.ac.uk/5/3/2.html>
- Hilmert, C. J., Kulik, J. A., & Christenfeld, N. J. S. (2006). Positive and negative opinion modeling: The influence of another's similarity and dissimilarity. *Journal of Personality and Social Psychology*, *90*(3), 440–452. doi: 10.1037/0022-3514.90.3.44

- Hogg, M. A., Turner, J. C., & Davidson, B. (1990). Polarized norms and social frames of reference: A test of the self-categorization theory of group polarization. *Basic and Applied Social Psychology, 11*(1), 77–100. doi: 10.1207/s15324834basp1101_6
- Holland, J. H. (1995). *Hidden Order: How Adaptation Builds Complexity*. New York, NY: Basic Books.
- Holley, R. A., & Liggett, T. M. (1975). Ergodic theorems for weakly interacting infinite systems and the voter model. *The Annals of Probability, 3*(4), 643–663.
- Huet, S., & Deffuant, G. (2010). Openness leads to opinion stability and narrowness to volatility. *Advances in Complex Systems, 13*(03), 405–23. Retrieved from <http://arxiv.org/abs/1404.7284>
- Isenberg, D. J. (1986). Group polarization: A critical review and meta-analysis. *Journal of Personality and Social Psychology, 50*(6), 1141–1151. doi: 10.1037/0022-3514.50.6.1141
- Jager, W., & Amblard, F. (2005). Uniformity, Bipolarization and Pluriformity Captured as Generic Stylized Behavior with an Agent-Based Simulation Model of Attitude Change. *Computational & Mathematical Organization Theory, 10*(4), 295–303. doi: 10.1007/s10588-005-6282-2
- Jahnke, J. C. (1963). Serial position effects in immediate serial recall. *Journal of Verbal Learning and Verbal Behavior, 2*(3), 284–287. doi: 10.1016/S0022-5371(63)80095-X
- Krueger, T., Szwabiński, J., & Weron, T. (2017). Conformity, anticonformity and polarization of opinions: Insights from a mathematical model of opinion dynamics. *Entropy, 19*(7), 371. doi: 10.3390/e19070371

- Kunc, M. (2016). System dynamics: A behavioral modeling method. In T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, & S. E. Chick (Eds.), *Proceedings of the 2016 winter simulation conference* (pp. 53–64). Arlington, VA: IEEE. doi: 10.1109/WSC.2016.7822079
- Kurahashi-Nakamura, T., Mäs, M., & Lorenz, J. (2016). Robust Clustering in Generalized Bounded Confidence Models. *Journal of Artificial Societies and Social Simulation*, 19(4), 7. doi: 10.18564/jasss.3220
- Latané, B. (1981). The psychology of social impact. *American Psychologist*, 36(4), 343–356.
- Law, A. M. (2007). *Simulation Modeling and Analysis* (Fourth ed.). New York, NY: McGraw-Hill.
- Lee, J. K., Choi, J., Kim, C., & Kim, Y. (2014). Social media, network heterogeneity, and opinion polarization. *Journal of Communication*, 64(4), 702–722. doi: 10.1111/jcom.12077
- Levendusky, M. S. (2013). Why do partisan media polarize viewers? *American Journal of Political Science*, 57(3), 611–623.
- Lewenstein, M., Nowak, A., & Latané, B. (1992). Statistical mechanics of social impact. *Physical Review A*, 45(2), 763–776. doi: 10.1103/PhysRevA.45.763
- Lorenz, J. (2010). Heterogeneous bounds of confidence: Meet, discuss and find consensus! *Complexity*, 15(4), 43–52.
- Macal, C. M. (2016). Everything you need to know about agent-based modelling and simulation. *Journal of Simulation*, 10(2), 144–156. doi: 10.1057/jos.2016.7

- Macal, C. M., & North, M. J. (2014). Introductory tutorial: Agent-based modeling and simulation. In *Proceedings of the 2014 winter simulation conference* (pp. 6–20). IEEE. doi: 10.1109/WSC.2014.7019874
- Mark, N. (1998). Beyond individual differences: Social differentiation from first principles. *American Sociological Review*, *63*(3), 309–330. Retrieved from <http://www.jstor.org/stable/2657552>
- Martins, A. C. R. (2009). Bayesian updating rules in continuous opinion dynamics models. *Journal of Statistical Mechanics: Theory and Experiment*, *2009*(02), P02017. doi: 10.1088/1742-5468/2009/02/P02017
- Martins, A. C. R., Pereira, C. d. B., & Vicente, R. (2009). An opinion dynamics model for the diffusion of innovations. *Physica A: Statistical Mechanics and its Applications*, *388*(15-16), 3225–3232.
- Mäs, M., & Flache, A. (2013). Differentiation without distancing: Explaining bipolarization of opinions without negative influence. *PLoS ONE*, *8*(11), e74516. doi: 10.1371/journal.pone.0074516
- Mäs, M., Flache, A., & Helbing, D. (2010). Individualization as driving force of clustering phenomena in humans. *PLoS Computational Biology*, *6*(10), e1000959. doi: 10.1371/journal.pcbi.1000959
- Mäs, M., Flache, A., & Kitts, J. A. (2014). Cultural integration and differentiation in groups and organizations. In V. Dignum & F. Dignum (Eds.), *Perspectives on culture and agent-based simulations: Integrating cultures* (pp. 71–90). Cham: Springer International Publishing.
- Mäs, M., Flache, A., Takács, K., & Jehn, K. A. (2013). In the short term we divide, in the long term we unite: Demographic crisscrossing and the effects of

- faultlines on subgroup polarization. *Organization Science*, 24(3), 716–736. doi: 10.1287/orsc.1120.0767
- McPherson, M., Smith-Lovin, L., & Cook, J. M. (2001). Birds of a Feather: Homophily in Social Networks. *Annual Review of Sociology*, 27(1), 415–444.
- Messing, S., & Westwood, S. J. (2014). Selective Exposure in the Age of Social Media. *Communication Research*, 41(8), 1042–1063. doi: 10.1177/0093650212466406
- Milgram, S. (1967). The small-world problem. *Psychology Today*, 1(1), 61–67.
- Miller, N., & Campbell, D. T. (1959). Recency and primacy in persuasion as a function of the timing of speeches and measurements. *The Journal of Abnormal and Social Psychology*, 59(1), 1–9. doi: 10.1037/h0049330
- Monica, S., & Bergenti, F. (2017). An analytic study of opinion dynamics in multi-agent systems. *Computers & Mathematics with Applications*, 73(10), 2272–2284. doi: 10.1016/j.camwa.2017.03.008
- Moscovici, S., & Zavalloni, M. (1969). The group as a polarizer of attitudes. *Journal of Personality and Social Psychology*, 12(2), 125–135. doi: 10.1037/h0027568
- Nguyen, T. T., Hui, P.-M., Harper, F. M., Terveen, L., & Konstan, J. A. (2014). Exploring the filter bubble. In *Proceedings of the 23rd international conference on world wide web* (pp. 677–686). New York, NY: ACM Press.
- North, M. J., & Macal, C. M. (2007). *Managing Business Complexity: Discovering Strategic Solutions with Agent-based Modeling and Simulation*. Oxford University Press.

- Nowak, A., Szamrej, J., & Latané, B. (1990). From private attitude to public opinion: A dynamic theory of social impact. *Psychological Review*, *97*(3), 362–376. doi: 10.1037/0033-295X.97.3.362
- Page, S. E. (1997). On incentives and updating in agent based models. *Computational Economics*, *10*(1), 67–87. doi: 10.1023/A:1008625524072
- Pariser, E. (2011). *The Filter Bubble: How the New Personalized Web is Changing What We Read and How We Think*. New York, NY: The Penguin Press.
- Pew Research Center. (2012). *Partisan polarization surges in Bush, Obama years*. Washington, DC. Retrieved from <http://www.people-press.org/2012/06/04/partisan-polarization-surges-in-bush-obama-years/>
- Pires, B., & Crooks, A. T. (2017). Modeling the emergence of riots: A geosimulation approach. *Computers, Environment and Urban Systems*, *61*(A), 66–80. doi: 10.1016/j.compenvurbsys.2016.09.003
- Railsback, S. F., & Grimm, V. (2011). *Agent-based and Individual-based Modeling: A Practical Introduction*. Princeton, NJ: Princeton University Press.
- Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH Computer Graphics*, *21*(4), 25–34. doi: 10.1145/37402.37406
- Salzarulo, L. (2006). A continuous opinion dynamics model based on the principle of meta-contrast. *Journal of Artificial Societies and Social Simulation*, *9*(1), 13. Retrieved from <http://jasss.soc.surrey.ac.uk/9/1/13.html>
- Sanders, G. S., & Baron, R. S. (1977). Is social comparison irrelevant for producing choice shifts? *Journal of Experimental Social Psychology*, *13*(4), 303–314. doi: 10.1016/0022-1031(77)90001-4

- Sands, S., & Wright, A. (1980). Primate memory: Retention of serial list items by a rhesus monkey. *Science*, *209*(4459), 938–940. doi: 10.1126/science.6773143
- Schelling, T. C. (1971). Dynamic models of segregation. *The Journal of Mathematical Sociology*, *1*(2), 143–186. doi: 10.1080/0022250X.1971.9989794
- Schelling, T. C. (1978). *Micromotives and Macrobehavior*. New York, NY: W.W. Norton.
- Schkade, D., Sunstein, C. R., & Hastie, R. (2010). When deliberation produces extremism. *Critical Review*, *22*(2-3), 227–252. doi: 10.1080/08913811.2010.508634
- Sherif, M., & Hovland, C. I. (1961). *Social Judgment: Assimilation and Contrast Effects in Communication and Attitude Change*. New Haven, CT: Yale University Press.
- Sîrbu, A., Loreto, V., Servedio, V. D. P., & Tria, F. (2017). Opinion Dynamics: Models, Extensions and External Effects. In V. Loreto et al. (Eds.), *Participatory sensing, opinions and collective awareness* (pp. 363–401). Cham, Switzerland: Springer International Publishing. doi: 10.1007/978-3-319-25658-0_17
- Sobkowicz, P. (2009). Modelling opinion formation with physics tools: Call for closer link with reality. *Journal of Artificial Societies and Social Simulation*, *12*(1), 11. Retrieved from <http://jasss.soc.surrey.ac.uk/12/1/11.html>
- Stauffer, D. (2002). Sociophysics: the Sznajd model and its applications. *Computer Physics Communications*, *146*(1), 93–98. doi: 10.1016/S0010-4655(02)00439-3
- Stauffer, D., Sousa, A. O., & Moss de Oliveira, S. (2000). Generalization to square lattice of Sznajd sociophysics model. *International Journal of Modern Physics C*, *11*(6), 1239–1245. doi: 10.1142/S012918310000105X

- Stoner, J. A. (1968). Risky and cautious shifts in group decisions: The influence of widely held values. *Journal of Experimental Social Psychology*, 4(4), 442–459.
- Sun, Z., & Müller, D. (2013). A framework for modeling payments for ecosystem services with agent-based models, Bayesian belief networks and opinion dynamics models. *Environmental Modelling & Software*, 45, 15–28. doi: 10.1016/j.envsoft.2012.06.007
- Sunstein, C. R. (2002). The law of group polarization. *Journal of Political Philosophy*, 10(2), 175–195. doi: 10.1111/1467-9760.00148
- Sznajd-Weron, K., & Sznajd, J. (2000). Opinion evolution in closed community. *International Journal of Modern Physics C*, 11(6), 1157–1165. doi: 10.1142/S0129183100000936
- Tajfel, H., & Turner, J. (1979). An integrative theory of intergroup conflict. In W. G. Austin & S. Worchel (Eds.), *The social psychology of intergroup relations* (pp. 33–47). Monterey: Brooks Cole.
- Takács, K., Flache, A., & Mäs, M. (2016). Discrepancy and disliking do not induce negative opinion shifts. *PLOS ONE*, 11(6), e0157948. doi: 10.1371/journal.pone.0157948
- Turner, J. C., Hogg, M. A., Oakes, P. J., Reicher, S. D., & Wetherell, M. S. (1987). *Rediscovering the Social Group: A Self-Categorization Theory*. Oxford, UK: Basil Blackwell.
- Turner, J. C., & Oakes, P. J. (1986). The significance of the social identity concept for social psychology with reference to individualism, interactionism and social influence. *British Journal of Social Psychology*, 25(3), 237–252. doi: 10.1111/j.2044-8309.1986.tb00732.x

- Urbig, D., Lorenz, J., & Herzberg, H. (2008). Opinion dynamics: The effect of the number of peers met at once. *Journal of Artificial Societies and Social Simulation*, *11*(2), 4. Retrieved from <http://jasss.soc.surrey.ac.uk/11/2/4.html>
- Vinokur, A., & Burnstein, E. (1974). Effects of partially shared persuasive arguments on group-induced shifts: A group-problem-solving approach. *Journal of Personality and Social Psychology*, *29*(3), 305–315. doi: 10.1037/h0036010
- Vinokur, A., & Burnstein, E. (1978). Depolarization of attitudes in groups. *Journal of Personality and Social Psychology*, *36*(8), 872–885. doi: 10.1037/0022-3514.36.8.872
- von Neumann, J. (1966). *Theory of Self-Reproducing Automata* (A. W. Burks, Ed.). Urbana, IL: University of Illinois Press.
- Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of ‘small-world’ networks. *Nature*, *393*, 440–442. doi: 10.1038/30918
- Weimer, C., Miller, J. O., Friend, M., & Miller, J. (2013). Forecasting effects of MISO actions: An ABM methodology. In *Proceedings of the 2013 winter simulation conference* (pp. 2762–2771). IEEE. doi: 10.1109/WSC.2013.6721647
- Weimer, C. W., Miller, J. O., & Hill, R. R. (2016). Agent-based modeling: An introduction and primer. In *Proceedings of the 2016 winter simulation conference* (pp. 65–79). IEEE Press. doi: 10.1109/WSC.2016.7822080
- Weisbuch, G., Deffuant, G., Amblard, F., & Nadal, J.-P. (2002). Meet, discuss, and segregate! *Complexity*, *7*(3), 55–63. doi: 10.1002/cplx.10031
- Wilensky, U. (1999). *NetLogo*. Evanston, IL: Center for Connected Learning and Computer-Based Modeling, Northwestern University. Retrieved from <http://ccl.northwestern.edu/netlogo/>

Wilensky, U., & Rand, W. (2015). *An Introduction to Agent-Based Modeling: Modeling Natural, Social, and Engineered Complex Systems with NetLogo*. Cambridge, MA: MIT Press.

Wolfram, S. (1986). *Theory and Applications of Cellular Automata*. Singapore: World Scientific.

Wolfram, S. (2002). *A New Kind of Science*. Champaign, IL: Wolfram Media.

Zuber, J. A., Crott, H. W., & Werner, J. (1992). Choice shift and group polarization: An analysis of the status of arguments and social decision schemes. *Journal of Personality and Social Psychology*, *62*(1), 50–61. doi: 10.1037/0022-3514.62.1.50

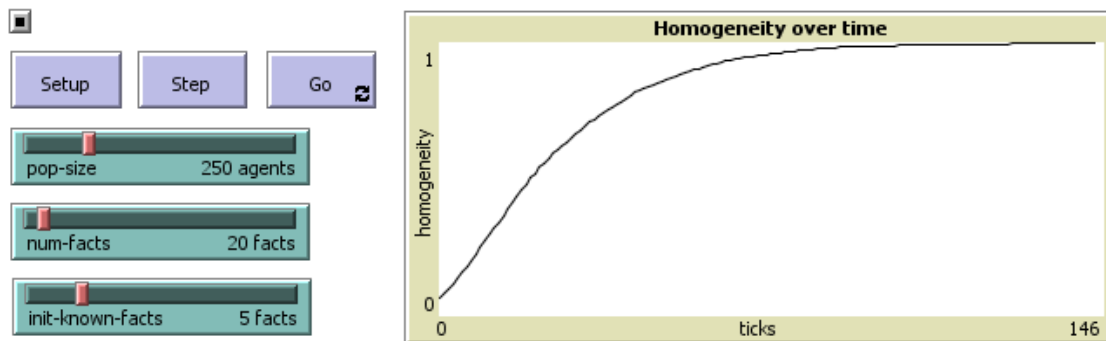
Appendix A. Opinion Dynamics Models Summary

Publication	Cultural Dynamics	Discrete Opinion	Continuous Opinion	Vector Opinion	Fixed Attributes	Cellular Automata	Agent-based Model	Bounded Confidence	Persuasive Argument	Social Comparison	Self-Categorization	Social Influence Network	Negative Influence	Individuation	Synchronous	Random Order	Random Independent
Axelrod (1997)	•					•											•
Carley (1991)	•																
Clifford & Sudbury (1973)		•				•											•
Dandekar et al. (2013)			•									•			•		•
Deffuant et al. (2000)				•			•	•		•							•
Deffuant et al. (2002)										•							•
DeGroot (1974)															•		
Duggins (2017)													•			•	
Flache & Mäs (2008a)					•								•				•
Flache & Mäs (2008b)					•								•				•

Publication	Cult. Dyn.	Disc. Op.	Cont. Op.	Vec. Op.	Fixed Attr.	CA	ABM	BC	Pers. Arg.	Soc. Comp.	Self-Cat.	Soc. Inf. Net	Neg. Inf.	Ind.	Sync.	Rand. Ord.	Rand. Ind.
Flache & Macy (2011)			•	•			•						•				•
Friedkin (1999)			•	•								•			•		
Friedkin (2001)			•	•								•			•		
Galam (1997)		•															
Hegselmann & Krause (2002)			•	•			•	•							•		
Holley & Liggett (1975)		•				•											•
Lorenz (2010)			•	•			•	•							•		•
Mark (1998)	•																•
Martins (2009)			•	•			•	•									•
Martins et al. (2009)			•	•			•	•									•
Mäs et al. (2010)			•	•			•				•			•			•
Mäs et al. (2013)		•		•	•		•		•								•
Mäs & Flache (2013)		•		•			•		•								•
Mäs et al. (2014)			•				•				•			•			•

Publication	Cult. Dyn.	Disc. Op.	Cont. Op.	Vec. Op.	Fixed Attr.	CA	ABM	BC	Pers. Arg.	Soc. Comp.	Self-Cat.	Soc. Inf. Net	Neg. Inf.	Ind.	Sync.	Rand. Ord.	Rand. Ind.
Nowak et al. (1990)		•				•									•		
Pires & Crooks (2017)			•									•					
Salzarulo (2006)			•				•	•			•		•				•
Stauffer et al. (2000)		•				•											•
Stauffer (2002)		•				•											•
Sun & Müller (2013)			•				•	•								•	
Sznajd-Weron & Sznajd (2000)		•				•											•
C. Weimer et al. (2013)		•		•			•						•			•	
Weisbuch et al. (2002)			•	•			•	•		•							•

Appendix B. NetLogo Code: Replication of Carley (1991)



```
extensions [ Rnd ]

globals [ total-facts ]

turtles-own [ facts partner-facts busy ]

to setup
  clear-all
  setup-turtles
  set total-facts (length remove-duplicates (reduce sentence [facts] of
    turtles))
  reset-ticks
end

to setup-turtles
  create-turtles pop-size [
    set facts n-of init-known-facts (range num-facts)
    set busy false
    hide-turtle
  ]
end

to go
  ask turtles [ pick-partner ]
  ask turtles [ update-facts ]
  tick
  if homogeneity = 1 [stop]
end

to pick-partner
  if not busy [
    let candidates turtles with [ not busy ]
```

```

    let partner rnd:weighted-one-of candidates [ num-shared-facts myself ]
    set partner-facts [facts] of partner
    set busy true
    ask partner [
      set partner-facts [facts] of myself
      set busy true
    ]
  ]
end

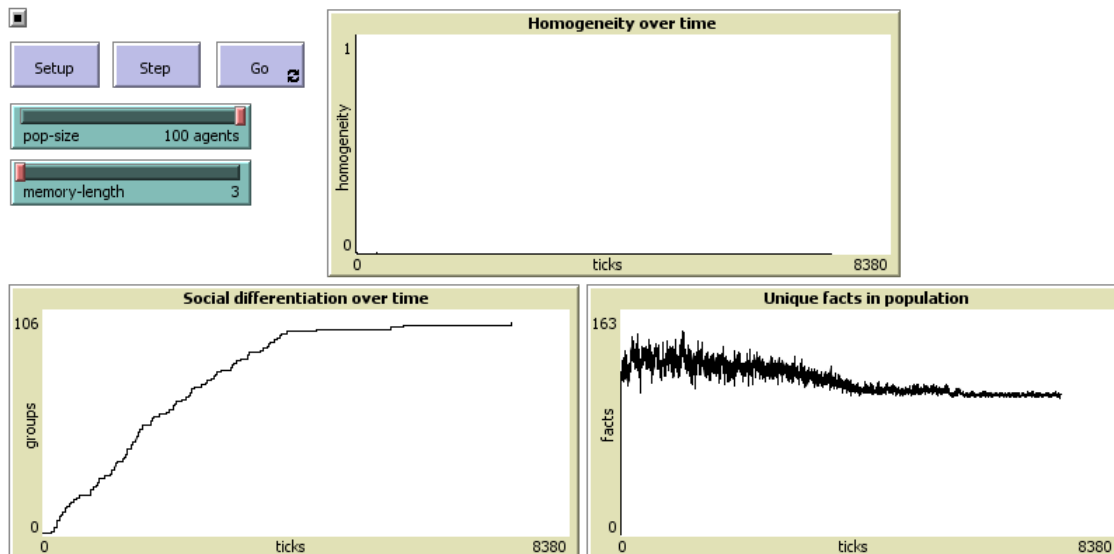
to-report num-shared-facts [ them ]
  ; called by turtle, reports number of shared facts
  report sum map [ [k] -> ifelse-value (member? k facts) [1][0] ] ([facts]
    of them)
end

to update-facts
  let k one-of partner-facts
  if (not member? k facts) [
    set facts lput k facts
  ]
  set busy false
end

to-report homogeneity
  let den (pop-size * (pop-size - 1)) / 2 * total-facts
  let num 0
  ask turtles [
    ask turtles with [who > [who] of myself] [
      set num (num + num-shared-facts myself)
    ]
  ]
  report num / den
end

```

Appendix C. NetLogo Code: Replication of Mark (1998)



```
extensions [ Rnd ]

globals [ total-facts max-fact ]

turtles-own [ facts fact-times partner ]

to setup
  clear-all
  setup-turtles
  setup-globals
  reset-ticks
end

to setup-turtles
  ;;; initially sets up turtles
  create-turtles pop-size [
    ; turtles all begin with 1 fact common to all turtles
    set facts (list 0)
    set fact-times (list 0)
    hide-turtle
  ]
end

to setup-globals
  ;;; initially sets up, or updates, global variables
  set total-facts remove-duplicates (reduce sentence [facts] of turtles)
```



```

; list of all facts in the system
set max-fact max total-facts
; used to ensure new facts are unique
end

to go
ask turtles [ pick-partner ]
ask turtles [ interact ]
ask turtles [ forget ]
setup-globals ; updates total-facts (list of all facts in the system)
tick
let h homogeneity
if h = 1 or h = 0 [stop]
; if ticks = 500 [stop]
end

to pick-partner
;;; turtle - implements partner selection rules
let candidates turtles
set partner rnd:weighted-one-of candidates [ num-shared-facts myself ]
end

to interact
;;; turtle - implements interaction rules
let all-facts remove-duplicates sentence facts [facts] of partner
if partner != self [ set all-facts lput (max-fact + 1) all-facts ]
let convo-topic one-of all-facts
if convo-topic > max-fact [ set max-fact convo-topic ]
add-fact convo-topic
if partner != self [ ask partner [ add-fact convo-topic ] ]
end

to forget
;;; turtle - implements forgetting rules
while [min fact-times < (ticks - memory-length)] [
let index position (min fact-times) fact-times
set facts remove-item index facts
set fact-times remove-item index fact-times
]
end

to-report num-shared-facts [ them ]
;;; turtle - reports number of shared facts with them
report sum map [ [k] -> ifelse-value (member? k facts) [1][0] ] ([facts]
of them)
end

```

```

to add-fact [ this-fact ]
  ;;; turtle - adds this-fact to its facts (if new) and updates the
  fact-times
  ifelse member? this-fact facts [
    let index position this-fact facts
    set fact-times replace-item index fact-times ticks
  ] [
    set facts lput this-fact facts
    set fact-times lput ticks fact-times
  ]
end

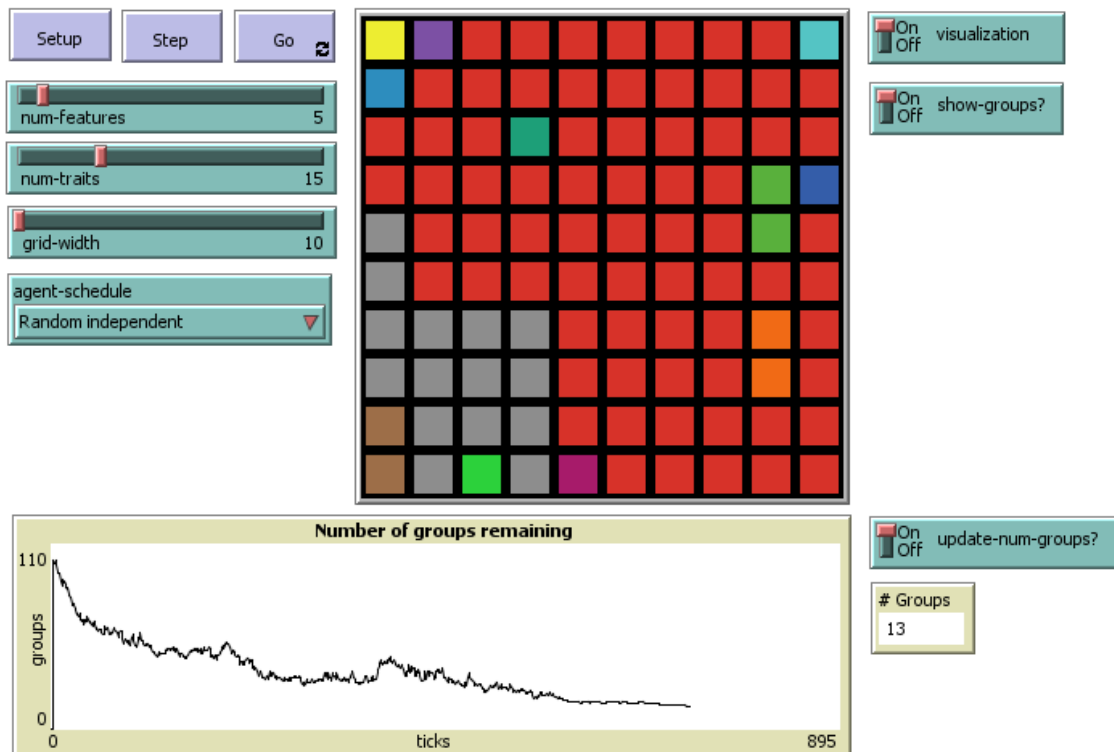
to-report homogeneity
  ;;; reports the cultural homogeneity of the system, values in range [0,1]
  let num-facts length total-facts
  let den (pop-size * (pop-size - 1)) / 2 * num-facts
  let num 0
  ask turtles [
    ask turtles with [who > [who] of myself] [
      set num (num + num-shared-facts myself)
    ]
  ]
  report num / den
end

to-report social-diff
  ;;; reports the number of disconnected groups
  let num-groups 0
  let ungrouped-agents turtles
  while [any? ungrouped-agents] [
    let this-group turtle-set one-of ungrouped-agents
    let group-facts reduce sentence [facts] of this-group
    let still-crawling? true
    while [still-crawling?] [
      let old-group count this-group
      foreach group-facts [ [this-fact] ->
        set this-group (turtle-set this-group (turtles with [member?
          this-fact facts]))
      ]
      set group-facts remove-duplicates reduce sentence [facts] of
        this-group
      set still-crawling? count this-group > old-group
    ]
    set ungrouped-agents ungrouped-agents with [not member? self
      this-group]
  ]

```

```
    set num-groups num-groups + 1  
  ]  
  report num-groups  
end
```

Appendix D. NetLogo Code: Replication of Axelrod (1997)



```

globals [ num-groups N last-change ]
turtles-own [ culture group next-culture ]

to setup
  clear-all
  setup-world
  setup-turtles
  if visualization [ ask turtles [ setup-links ] ]
  set num-groups length all-cultures
  set N count turtles
  reset-ticks
end

to setup-world
  resize-world 0 (grid-width - 1) 0 (grid-width - 1)
  set-patch-size floor (300 / grid-width)
end

to setup-turtles
  set-default-shape turtles "square"

```

```

ask patches [
  sprout 1 [
    set culture n-values num-features [ one-of range num-traits ]
    set next-culture culture
    set color white
  ]
]
end

to setup-links
  create-links-with turtles-on neighbors4 [
    set thickness 0.8
    set color 9.9 * similarity both-ends
  ]
end

to go
  if agent-schedule = "Synchronous" [
    ask turtles [ interact ]
    ask turtles [ set culture next-culture ]
  ]
  if agent-schedule = "Random_order" [ ask turtles [ interact ] ]
  if agent-schedule = "Random_independent" [ repeat N [ ask random-turtle
    [ interact ] ] ]
  ask links [ set color 9.9 * similarity both-ends ]
  tick
  if update-num-groups? [ set num-groups length all-cultures ]
  if num-groups < 15 [
    if show-groups? [
      show-groups
      ask links [ ifelse color = white [ hide-link ] [ show-link ] ]
    ]
  ]
  if last-change <= (ticks - 5) [ stop ]
end

to interact
  let them one-of turtles-on neighbors4
  let G-indices filter [ [index] -> item index culture != [item index
    culture] of them ] range num-features
  let f random num-features
  if item f culture = [item f culture] of them [
    if G-indices != [] [
      let g one-of G-indices
      ifelse agent-schedule = "Synchronous" [
        set next-culture replace-item g culture [item g culture] of them

```

```

    ] [
      set culture replace-item g culture [item g culture] of them
    ]
    set last-change ticks
  ]
]
end

to-report random-turtle
  report turtle (random N)
end

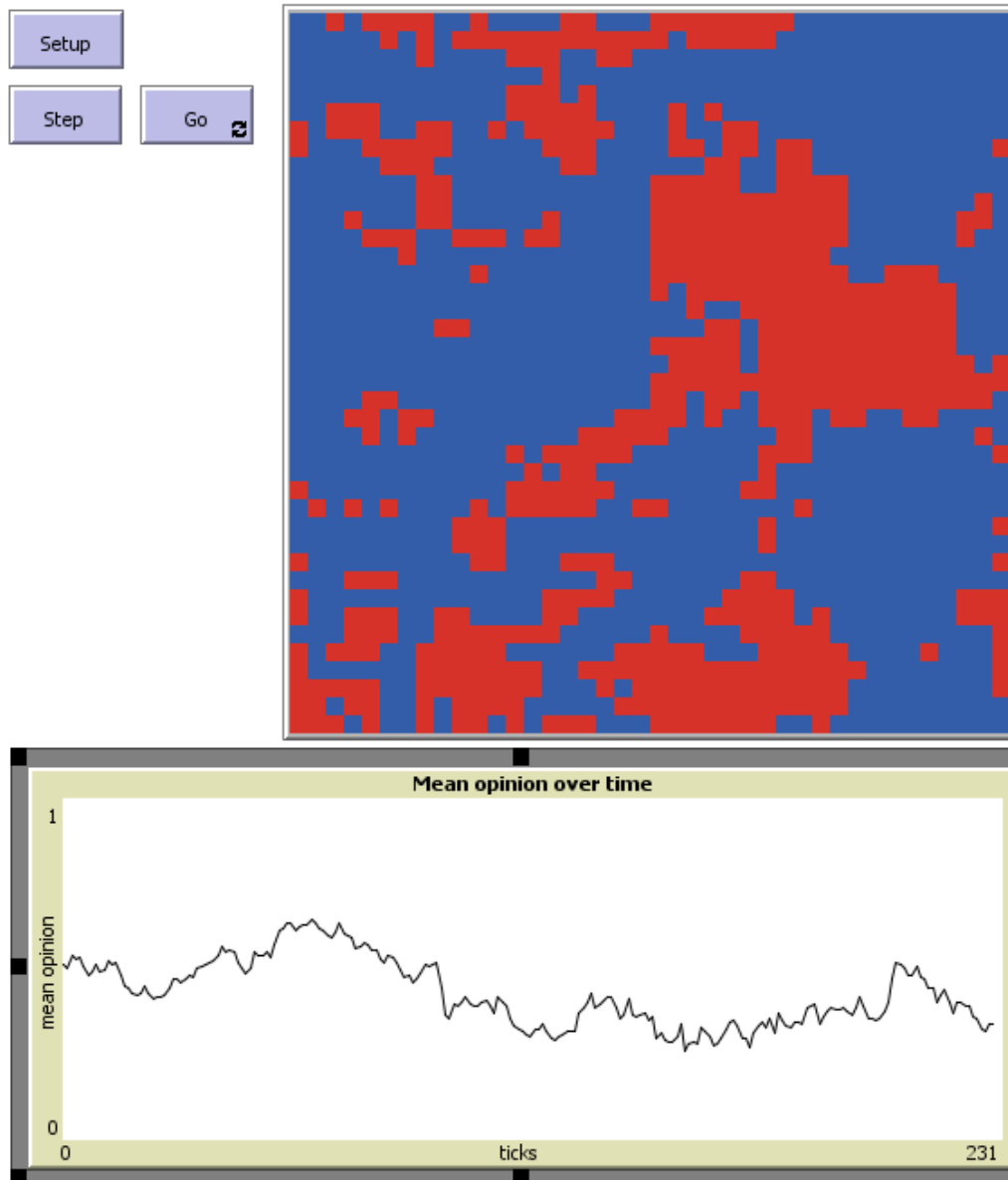
to-report similarity [ pair ]
  let cultures [ culture ] of pair
  let sim length filter [ [index] -> item index (item 0 cultures) = item
    index (item 1 cultures) ] range num-features
  report sim / num-features
end

to-report all-cultures
  report remove-duplicates n-values N [ [index] -> [culture] of turtle
    index ]
end

to show-groups
  let cultures all-cultures
  (foreach cultures (range length cultures) [ [this-culture grp] ->
    ask turtles with [culture = this-culture] [
      set group grp
      ifelse length cultures < 10 [
        set color item group (list red blue green yellow violet orange
          turquoise white black)
      ] [ set color (10 * group + 5)
    ]
  ]
])
;ask links [ hide-link ]
end

```

Appendix E. NetLogo Code: Voter Model



```
patches-own [ opinion ]

to setup
  clear-all
  ask patches [ set-opinion one-of [0 1] ]
  reset-ticks
end
```

```
to go
  let i one-of patches
  let o [opinion] of i
  ask one-of patches [ set-opinion [opinion] of one-of neighbors4 ]
  if o != [opinion] of i [ tick ]
  let op [opinion] of patch 0 0
  if all? patches [ opinion = op ] [stop]
end

to set-opinion [ val ]
  set opinion val
  set pcolor ifelse-value (val = 1) [red] [blue]
end
```


Appendix F. NetLogo Code: Replication of Nowak et al. (1990)

Setup

Step Go to End

Set percentage of minority (blue) cells before setup:

minority-percentage 30

In original model, range of impact was limited to 10. Include this?

On Off limit-range?

In original paper, an alternative formula is proposed (p. 373). Use it?

On Off alternative-formula?

Original model used synchronous updates.

update-schedule
Synchronous

Proportion of cells of each color

tick	red count	blue count
0	0.7	0.3
1	1.0	0.0
13.8	1.0	0.0

```
patches-own [
  p ; persuasiveness
  s ; social support
]
```

```
to setup
  clear-all
  ask patches [
    set pcolor red
    set p random 101
  ]
end
```

```

    set s random 101
  ]
  ask n-of (round ( (count patches) * minority-percentage / 100)) patches [
    set pcolor blue
  ]
  reset-ticks
end

to go
  let finished? false ; when it ends the turn true, stop the model at
    equilibrium

  if update-schedule = "Synchronous" [ ; original schedule
    let flippers patches with [flip?]
    ask flippers [ flip ]
    if count flippers = 0 [ set finished? true ]
  ]
  if update-schedule = "Random_order" [
    set finished? true
    ask patches [
      if flip? [
        flip
        set finished? false
      ]
    ]
  ]
  if update-schedule = "Random_independent" [
    set finished? true
    repeat count patches [
      ask one-of patches [
        if flip? [
          flip
          set finished? false
        ]
      ]
    ]
  ]
]

tick
if finished? [ stop ]
end

to-report flip?
  let all-cells ifelse-value (limit-range?) [ patches in-radius 10 ] [
    patches ]
  ; Calculate persuasive impact

```

```

let opposers all-cells with [pcolor != [pcolor] of myself]
let No count opposers
let ip 0
if No > 0 [ ; to keep from failing when none oppose
  if-else alternative-formula? [
    set ip sqrt (sum [(p / (((sqrt 2) + distance myself) ^ 2) ) ^ 2 ] of
      opposers)
  ] [
    set ip ( (mean [p / (((sqrt 2) + distance myself) ^ 2)] of opposers)
      * sqrt(No) ) ; persuasive impact
  ]
]

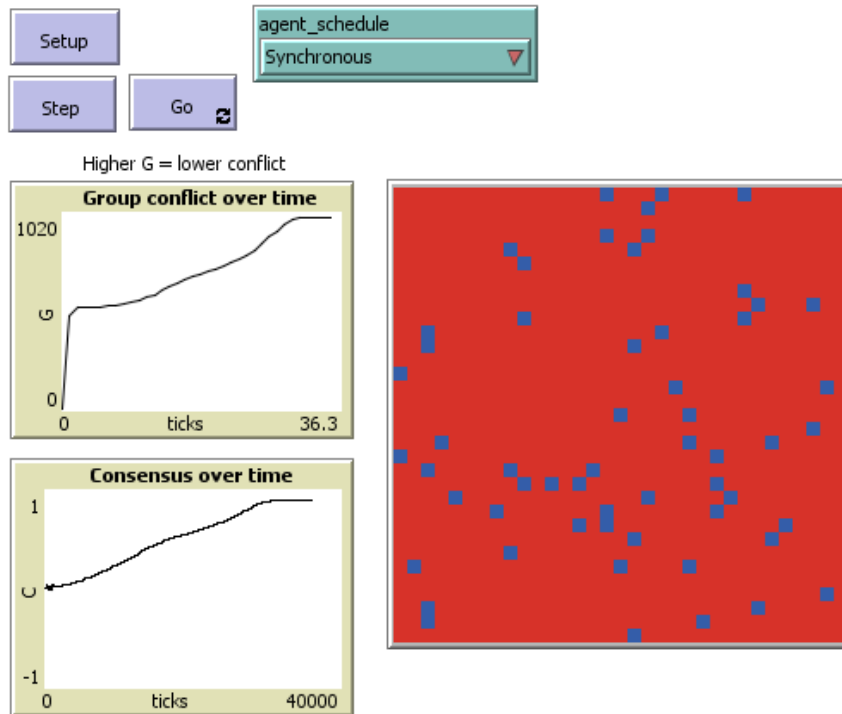
; Calculate social support impact
let supporters all-cells with [pcolor = [pcolor] of myself]
let Ns count supporters
let is 0
if-else alternative-formula? [
  set is sqrt (sum [(s / (((sqrt 2) + distance myself) ^ 2) ) ^ 2 ] of
    supporters)
] [
  set is ( (mean [s / (((sqrt 2) + distance myself) ^ 2)] of supporters)
    * sqrt(Ns) ) ; social support impact
]

report ip > is
end

to flip
  set pcolor ifelse-value (pcolor = red) [ blue ] [ red ]
  set p random 101
  set s random 101
end

```

Appendix G. NetLogo Code: Agent-based implementation of Galam (1997)



```

globals [
  N ; number of people (patches)
  I ; conflict amplitude
  gamma ; constant used in field computation
  smalln ; number of people with whom each individual interacts
  S ; external social field
  patch-list ; list of patches to allow ticks within runs through all
    patches
  updates ; number of updates this step (to check if model is finished)
]

patches-own [
  ci ; binary choice (-1 or 1) at the individual level
  Si ; internal social field
  nextci ; next value of ci (for use in synchronous updates)
]

to setup
  clear-all
  setup-patches

```

```

    setup-globals
    reset-ticks
end

to setup-globals
    set N count patches
    set I 1 / N
    set smalln N
    set S 0
    set gamma (smalln * I * N) / (2 * (N - 1))
    set patch-list [self] of patches
    set updates 0
end

to setup-patches
    ask patches [
        set ci one-of [-1 1]
        set nextci ci
        set Si (random-float 2) - 1
        set pcolor ifelse-value (ci = 1) [red] [blue]
    ]
end

to go
    set updates 0
    if agent_schedule = "Random_independent" [
        repeat N [
            ask one-of patches [ make-choice update-state]
            tick
        ]
    ]
    if agent_schedule = "Random_order" [
        foreach shuffle patch-list [ [this-patch] ->
            ask this-patch [make-choice update-state]
            tick
        ]
    ]
    if agent_schedule = "Synchronous" [
        ask patches [ make-choice ]
        foreach shuffle patch-list [ [this-patch] ->
            ask this-patch [ update-state ]
            tick
        ]
    ]
    if updates = 0 [stop]
end

```

```

to make-choice
  let deltaG (-2 * ci * (S + Si + I * sum [ci] of other patches))
  if deltaG > 0 [
    set nextci -1 * ci
    set updates updates + 1
  ]
end

to update-state
  set ci nextci
  set pcolor ifelse-value (ci = 1) [red] [blue]
end

to-report C ; group choice
  report sum [ci] of patches
end

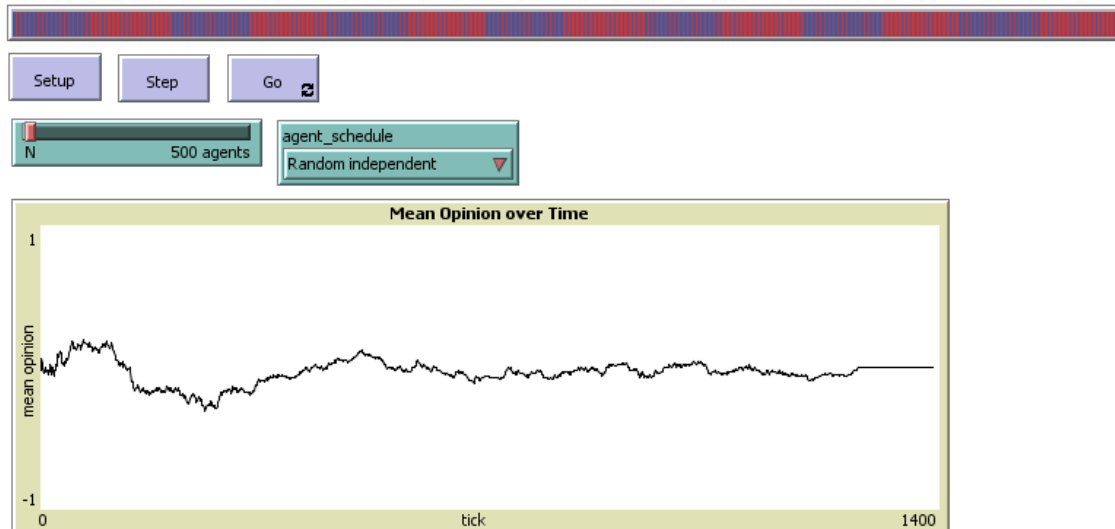
to-report GI ; group internal conflict function
  let gtemp 0
  foreach [ci] of patches [ [ci1] ->
    foreach [ci] of patches [ [ci2] ->
      set gtemp (gtemp + (ci1 * ci2))
    ]
  ]
  report I * gtemp
end

to-report Sg ; group social field
  report gamma * C / N
end

to-report G ; total group conflict (higher is less conflict)
  report GI + sum [(S + Si) * ci] of patches
end

```

Appendix H. NetLogo Code: Replication of Sznajd-Weron & Sznajd (2000)



```
patches-own [ influences opinion ]
```

```
to setup
```

```
  clear-all
  resize-world 0 (N - 1) 0 20
  set-patch-size 650 / N
  ask patches [ set opinion 0 ]
  foreach range N [ [x] ->
    set-opinion x (one-of [-1 1])
    ask patch x 0 [ set influences [] ]
  ]
  reset-ticks
```

```
end
```

```
to go
```

```
  if agent_schedule = "Random_independent" [
    repeat N [
      let r random-float (N - 1)
      ask patch (floor r) 0 [ flip-neighbors ]
    ]
  ]
  if agent_schedule = "Random_order" [
    let left-partners shuffle n-values (N - 1) [ [i] -> patch i 0 ]
    foreach left-partners [ [left-patch] ->
      ask left-patch [ flip-neighbors ]
    ]
  ]
```

```

]
]
if agent-schedule = "Synchronous" [
  let left-partners n-values (N - 1) [ [i] -> patch i 0 ]
  foreach left-partners [ [left-patch] ->
    ask left-patch [ influence-neighbors ]
  ]
  update-patches
]
tick
if all? patches [pcolor = red] [stop]
if all? patches [pcolor = blue] [stop]
end

to flip-neighbors
  let xi pxcor
  let xj (xi + 1)
  let Si opinion
  let Sj [opinion] of patch xj 0
  if xi > 0 [ set-opinion (xi - 1) Sj ]
  if xj < (N - 1) [ set-opinion (xj + 1) Si ]
end

to influence-neighbors
  let xi pxcor
  let xj (xi + 1)
  let Si opinion
  let Sj [opinion] of patch xj 0
  if xi > 0 [ add-influence (xi - 1) Sj ]
  if xj < (N - 1) [ add-influence (xj + 1) Si ]
end

to update-patches
  foreach range N [ [x] ->
    let consensus mean [influences] of patch x 0
    if consensus != 0 [ set-opinion x consensus ]
    ask patch x 0 [ set influences [] ]
  ]
end

to add-influence [ x val ]
  ask patch x 0 [ set influences lput val influences ]
end

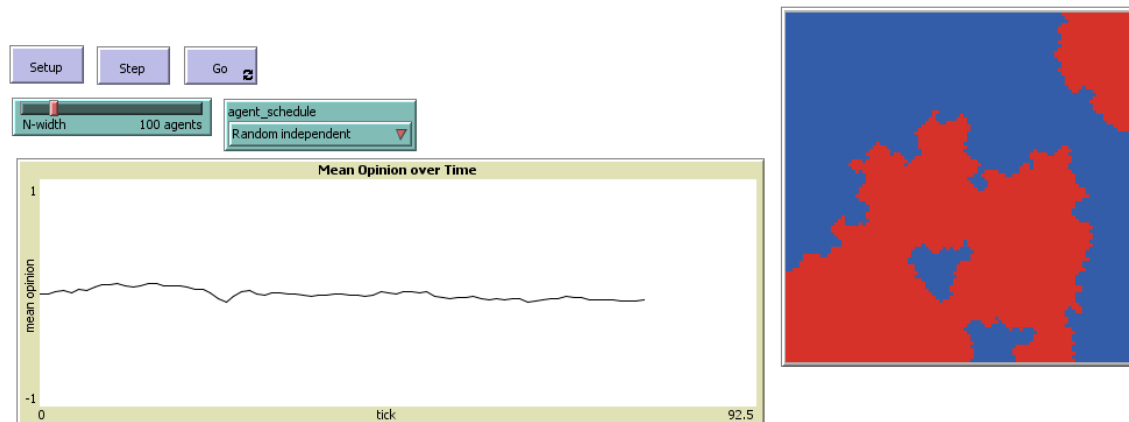
to set-opinion [ x val ]
  ask patch x 0 [ set opinion val ]
end

```



```
let col ifelse-value (val = -1) [blue] [red]
foreach range (max-pycor + 1) [ [i] ->
  ask patch x i [ set pcolor col ]
]
end
```

Appendix I. NetLogo Code: Replication of Stauffer (2002)



```
patches-own [ influences opinion ]

globals [ N pairs max-obs min-obs mo ]

to setup
  clear-all
  setup-world
  setup-patches
  reset-ticks
end

to setup-world
  set N (N-width ^ 2)
  resize-world 0 (N-width - 1) 0 (N-width - 1)
  set-patch-size 300 / N-width
end

to setup-patches
  ask patches [
    set-opinion one-of [-1 1]
    set pcolor ifelse-value (opinion = 1) [ red ] [ blue ]
    set influences []
  ]
  set pairs all-pairs
  set mo mean [opinion] of patches
  set max-obs mo
  set min-obs mo
end

to-report all-pairs
```

```

let patch-list []
ask patches with [pxcor < max-pxcor] [
  let x pxcor
  let y pycor
  set patch-list lput (list self (patch (x + 1) y)) patch-list
]
ask patches with [pycor < max-pycor] [
  let x pxcor
  let y pycor
  set patch-list lput (list self (patch x (y + 1))) patch-list
]
report patch-list
end

to go
if agent_schedule = "Random_independent" [
  repeat N [
    let actors one-of pairs
    flip-neighbors actors
  ]
]
if agent_schedule = "Random_order" [
  foreach shuffle pairs [ [actors] ->
    flip-neighbors actors
  ]
]
if agent_schedule = "Synchronous" [
  foreach pairs [ [actors] ->
    influence-neighbors actors
  ]
]
update-patches
]
tick
update-outputs
if abs mo = 1 [stop]
end

to set-opinion [ o ]
set opinion o
set pcolor ifelse-value (o = 1) [red] [blue]
end

to flip-neighbors [ actors ]
let op [opinion] of item 0 actors
if (op = [opinion] of item 1 actors) [
  let audience patch-set [neighbors4] of patch-set actors

```

```

    ask audience [ set-opinion op ]
  ]
end

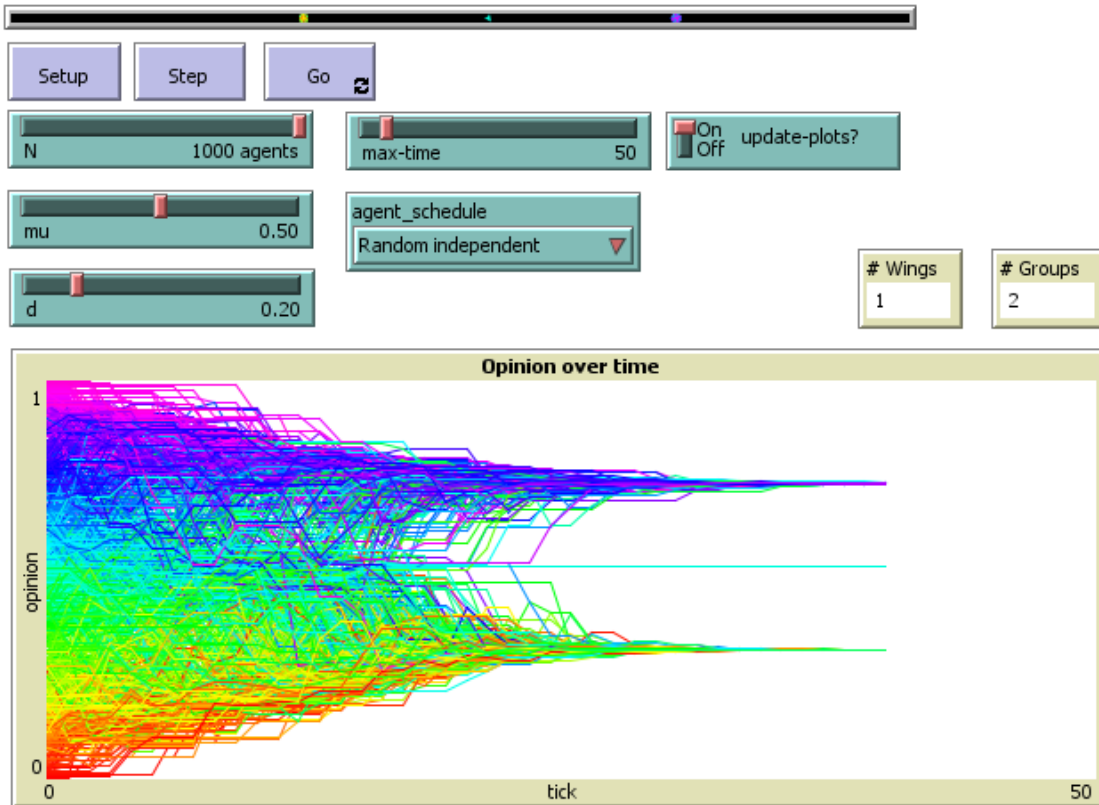
to influence-neighbors [ actors ]
  let op [opinion] of item 0 actors
  if (op = [opinion] of item 1 actors) [
    let audience patch-set [neighbors4] of patch-set actors
    ask item 0 actors [ set audience other audience ]
    ask item 1 actors [ set audience other audience ]
    ask audience [ set influences lput op influences ]
  ]
end

to update-patches
  ask patches [
    if not empty? influences [
      let consensus mean influences
      if consensus != 0 [ set consensus ifelse-value (consensus < 0) [-1]
        [1] ]
      set-opinion consensus
      set influences []
    ]
  ]
end

to update-outputs
  set mo mean [opinion] of patches
  set min-obs min list min-obs mo
  set max-obs max list max-obs mo
end

```

Appendix J. NetLogo Code: Replication of Deffuant et al. (2000)



```
turtles-own [  
  opinion  
  next-opinion  
]  
  
to setup  
  clear-all  
  create-turtles N [  
    set opinion random-float 1  
    set next-opinion opinion  
    set color hsb (310 * opinion) 100 100  
    setxy (opinion * 100) 0  
  ]  
  reset-ticks  
  setup-plot  
end
```

```

to setup-plot
  set-current-plot "Opinion-over-time"
  set-plot-x-range 0 max-time
  ask turtles [
    create-temporary-plot-pen word "Turtle_" who
    set-plot-pen-color color
    plot-pen-up
  ]
end

to go
  if agent-schedule = "Random-independent" [
    repeat count turtles [
      ask one-of turtles [
        influence-another
      ]
    ]
  ]
  if agent-schedule = "Random-order" [
    ask turtles [
      influence-another
    ]
  ]
  if agent-schedule = "Synchronous" [
    ask turtles [ survey-nearby ]
    ask turtles [ update-opinion ]
  ]
  if update-plots? [ ask turtles [ update-plot ] ]
  tick

  let finished? true
  let opinions group-opinions
  if (length opinions > 1) [
    let opnow item 0 opinions
    set opinions but-first opinions
    foreach opinions [ [op] ->
      if ((op - opnow) < d) [ set finished? false ]
      set opnow op
    ]
  ]
  if finished? [ stop ]
  ;if ticks > max-time [ stop ]
end

to influence-another
  let j one-of other turtles

```

```

let xi opinion
let xj [opinion] of j
if abs (xi - xj) < d [
  set opinion xi + mu * (xj - xi)
  set xcor (opinion * 100)
  ask j [
    set opinion xj + mu * (xi - xj)
    set xcor (opinion * 100)
  ]
]
end

to survey-nearby
  let j one-of other turtles
  let xi opinion
  let xj [opinion] of j
  if abs (xi - xj) < d [
    set next-opinion xi + mu * (xj - xi)
  ]
end

to update-opinion
  set opinion next-opinion
  set xcor (opinion * 100)
end

to update-plot
  set-current-plot "Opinion over time"
  set-current-plot-pen word "Turtle_" who
  plot-pen-down
  plotxy ticks opinion
  plot-pen-up
end

to-report group-opinions
  let opinions remove-duplicates [precision opinion 2] of turtles
  report sort opinions
end

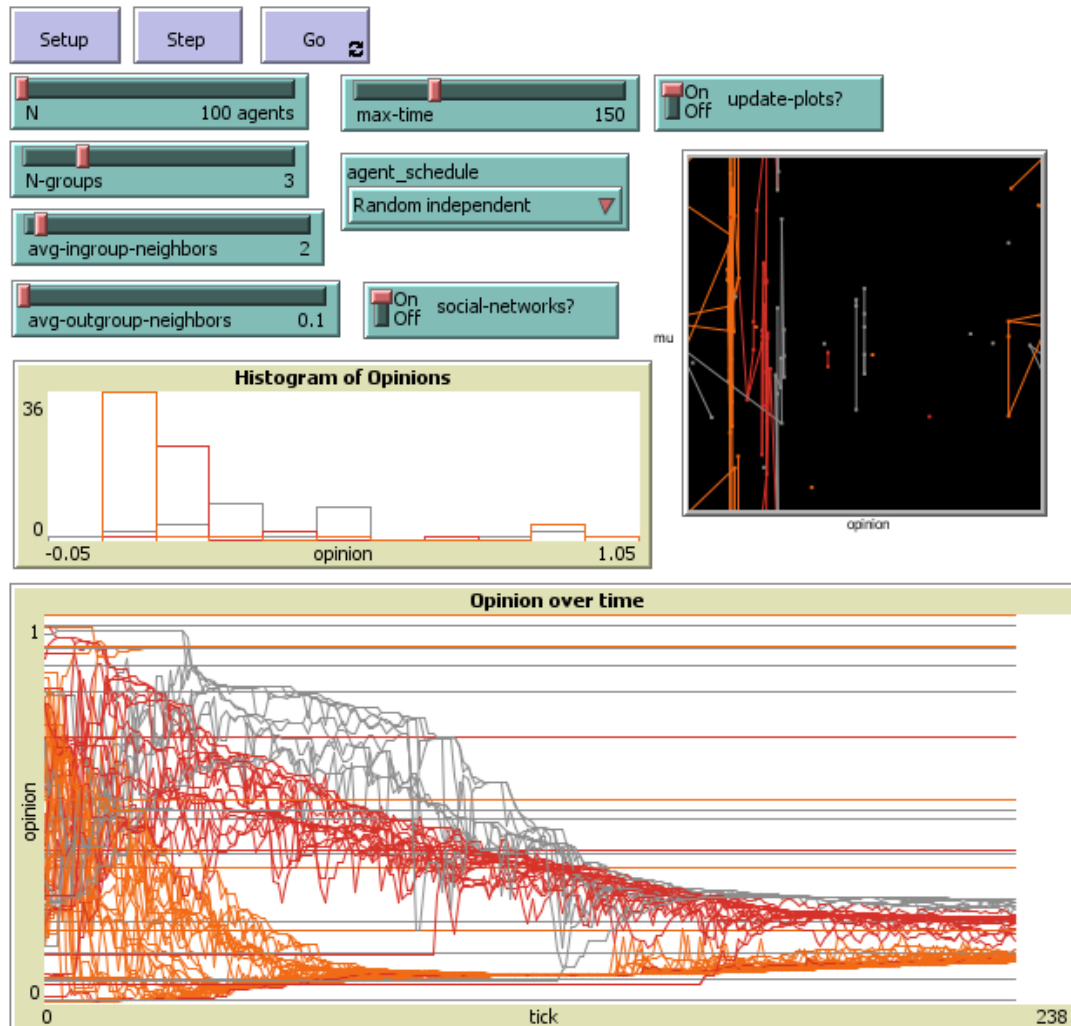
to-report group-membership
  let membership []
  foreach group-opinions [ [op] ->
    set membership lput (count turtles with [(precision opinion 2) = op])
    membership
  ]
  report membership
end

```

```
end

to-report wingless-group-opinions
  let opinions group-opinions
  let gm group-membership
  (foreach opinions gm [ [op mem] ->
    if (mem < 0.03 * N) [
      set opinions remove-item op opinions
    ]
  ])
  report opinions
end
```


Appendix K. NetLogo Code: Replication of opinion dynamics portion of Sun & Müller (2013)



```
globals [
  ingroup-density
  outgroup-density
]
```

```
turtles-own [
  opinion
  next-influence
  mu
  d
  group
]
```

```

]

to setup
  clear-all
  set ingroup-density (avg-ingroup-neighbors / (N / N-groups) )
  set outgroup-density ifelse-value (N-groups > 1) [
    (avg-outgroup-neighbors / (N - N / N-groups))] [ 0 ]
  setup-turtles
  reset-ticks
  setup-plot
end

to setup-turtles
  create-turtles N [
    set opinion random-float 1
    set mu random-normal 0.5 0.2
    while [(mu < 0) or (mu > 1)] [
      set mu random-normal 0.5 0.2
    ]
    setxy (opinion * 100) (mu * 100)
    update-d
    set group one-of range N-groups
    set color ifelse-value social-networks? [(10 * group + 5)] [hsb (310 *
      opinion) 100 100]
  ]
  if social-networks? [
    ask turtles [
      foreach (range (who + 1) N) [ [their-who] ->
        let them turtle their-who
        let r random-float 1
        if r < (ifelse-value ([group] of them = group) [ingroup-density]
          [outgroup-density]) [
          create-link-with them [ set color (10 * [group] of myself + 5) ]
        ]
      ]
    ]
  ]
end

to setup-plot
  set-current-plot "Opinion over time"
  set-plot-x-range 0 max-time
  ask turtles [
    create-temporary-plot-pen word "Turtle_" who
    set-plot-pen-color color
    plot-pen-up
  ]
end

```

```

]
end

to go
  if agent_schedule = "Random_independent" [
    repeat count turtles [
      ask one-of turtles [
        influence-another
      ]
    ]
  ]
  if agent_schedule = "Random_order" [
    ask turtles [
      influence-another
    ]
  ]
  if agent_schedule = "Synchronous" [
    ask turtles [ survey-nearby ]
    ask turtles [ update-opinion ]
  ]
  if update-plots? [ ask turtles [ update-plot ] ]
  tick

  if is-finished [ stop ]

end

to influence-another
  let my-neighbors ifelse-value social-networks? [ other
    out-link-neighbors ] [ other turtles ]
  if any? my-neighbors [
    let j one-of my-neighbors
    let xi opinion
    let xj [opinion] of j
    if abs (xi - xj) < d [
      set opinion xi + mu * (xj - xi)
      update-d
      set xcor (opinion * 100)
      if abs (xi - xj) < [d] of j [
        ask j [
          set opinion xj + mu * (xi - xj)
          update-d
          set xcor (opinion * 100)
        ]
      ]
    ]
  ]
]

```

```

]
end

to survey-nearby
  let my-neighbors ifelse-value social-networks? [ other
    out-link-neighbors ] [ other turtles ]
  let eligible-others my-neighbors with [ abs (opinion - [opinion] of
    myself) < [d] of myself ]
  if-else any? eligible-others [
    set next-influence mean [opinion] of eligible-others
  ] [
    set next-influence opinion
  ]
end

to update-opinion
  set opinion opinion + mu * (next-influence - opinion)
  set xcor (opinion * 100)
  update-d
end

to update-d
  set d (1 - 2 * abs (opinion - 0.5))
end

to update-plot
  set-current-plot "Opinion over time"
  set-current-plot-pen word "Turtle" who
  plot-pen-down
  plotxy ticks opinion
  plot-pen-up
end

to-report group-opinions
  let opinions remove-duplicates [precision opinion 2] of turtles
  report sort opinions
end

to-report group-membership
  let membership []
  foreach group-opinions [ [op] ->
    set membership lput (count turtles with [(precision opinion 2) = op])
    membership
  ]
  report membership
end

```

```

to-report wingless-group-opinions
  let opinions group-opinions
  let gm group-membership
  (foreach opinions gm [ [op mem] ->
    if (mem < 0.03 * N) [
      set opinions remove-item op opinions
    ]
  ])
  report opinions
end

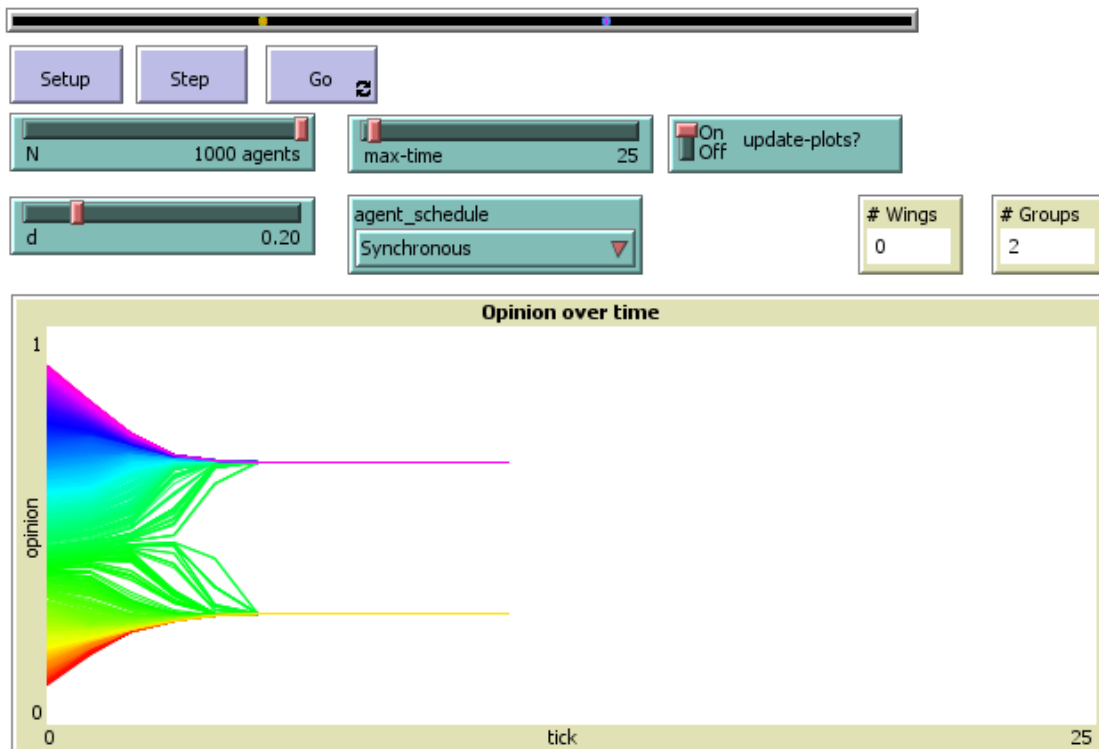
```

```

to-report is-finished
  let finished? true
  let opinions group-opinions
  if (length opinions > 1) [
    let opnow item 0 opinions
    set opinions but-first opinions
    foreach opinions [ [op] ->
      if ((op - opnow) < (max [d] of turtles)) [ set finished? false ]
      set opnow op
    ]
  ]
  report finished?
  ;if ticks > max-time [ stop ]
end

```

Appendix L. NetLogo Code: Replication of Hegselmann & Krause (2002)



```

turtles-own [
  opinion
  next-opinion
]

to setup
  clear-all
  create-turtles N [
    set opinion random-float 1
    set color hsb (310 * opinion) 100 100
    setxy (opinion * 100) 0
  ]
  reset-ticks
  setup-plot
end

to setup-plot
  set-current-plot "Opinion over time"
  set-plot-x-range 0 max-time

```

```

ask turtles [
  create-temporary-plot-pen word "Turtle_" who
  set-plot-pen-color color
  plot-pen-up
]
end

to go
  if agent_schedule = "Random_independent" [
    repeat count turtles [
      ask one-of turtles [
        survey-nearby
        update-opinion
      ]
    ]
  ]
  if agent_schedule = "Random_order" [
    ask turtles [
      survey-nearby
      update-opinion
    ]
  ]
  if agent_schedule = "Synchronous" [
    ask turtles [ survey-nearby ]
    ask turtles [ update-opinion ]
  ]
  if update-plots? [ ask turtles [ update-plot ] ]
  tick

  let finished? true
  let opinions group-opinions
  if (length opinions > 1) [
    let opnow item 0 opinions
    set opinions but-first opinions
    foreach opinions [ [op] ->
      if ((op - opnow) < d ) [ set finished? false ]
      set opnow op
    ]
  ]
  if finished? [ stop ]
  ;if ticks > max-time [ stop ]
end

to survey-nearby
  let eligible-others turtles with [ abs (opinion - [opinion] of myself) <
    d ]

```

```

    set next-opinion mean [opinion] of eligible-others
end

to update-opinion
    set opinion next-opinion
    set xcor (opinion * 100)
end

to update-plot
    set-current-plot "Opinion over time"
    set-current-plot-pen word "Turtle_" who
    plot-pen-down
    plotxy ticks opinion
    plot-pen-up
end

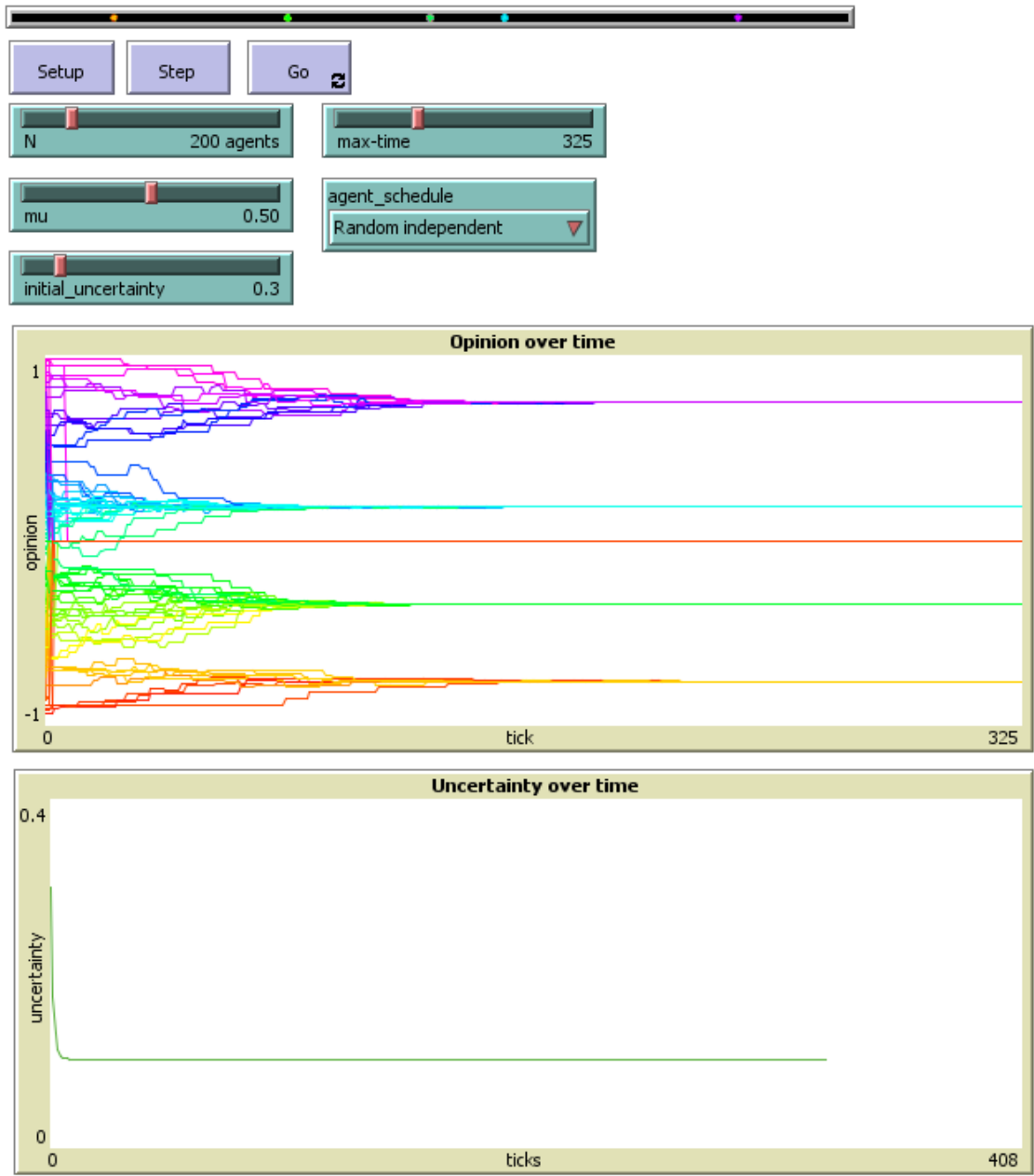
to-report group-opinions
    let opinions remove-duplicates [precision opinion 2] of turtles
    report sort opinions
end

to-report group-membership
    let membership []
    foreach group-opinions [ [op] ->
        set membership lput (count turtles with [(precision opinion 2) = op])
        membership
    ]
    report membership
end

to-report wingless-group-opinions
    let opinions group-opinions
    let gm group-membership
    (foreach opinions gm [ [op mem] ->
        if (mem < 0.03 * N) [
            set opinions remove-item op opinions
        ]
    ])
    report opinions
end

```


Appendix M. NetLogo Code: Replication of Deffuant et al. (2002)



```
turtles-own [  
  opinion  
  uncertainty  
  next-opinion  
  next-uncertainty
```

```

]

to setup
  clear-all
  create-turtles N [
    set opinion (random-float 2) - 1
    set uncertainty initial_uncertainty
    set color hsb (155 * opinion + 155) 100 100
    setxy (opinion * 50 + 50) 0
  ]
  reset-ticks
  setup-plot
end

to setup-plot
  set-current-plot "Uncertainty_over_time"
  set-plot-x-range 0 max-time
  set-plot-y-range 0 initial_uncertainty + 0.1

  set-current-plot "Opinion_over_time"
  set-plot-x-range 0 max-time
  ask turtles [
    create-temporary-plot-pen word "Turtle_" who
    set-plot-pen-color color
    plot-pen-up
  ]
end

to go
  if agent_schedule = "Random_independent" [
    repeat count turtles [
      ask one-of turtles [
        influence-another
      ]
    ]
  ]
  if agent_schedule = "Random_order" [
    ask turtles [
      influence-another
    ]
  ]
  if agent_schedule = "Synchronous" [
    ask turtles [ survey-nearby ]
    ask turtles [ update-opinion ]
  ]
  ask turtles [ update-plot ]

```

```

    tick
    if ticks > max-time [ stop ]
end

to influence-another
  survey-nearby
  update-opinion
end

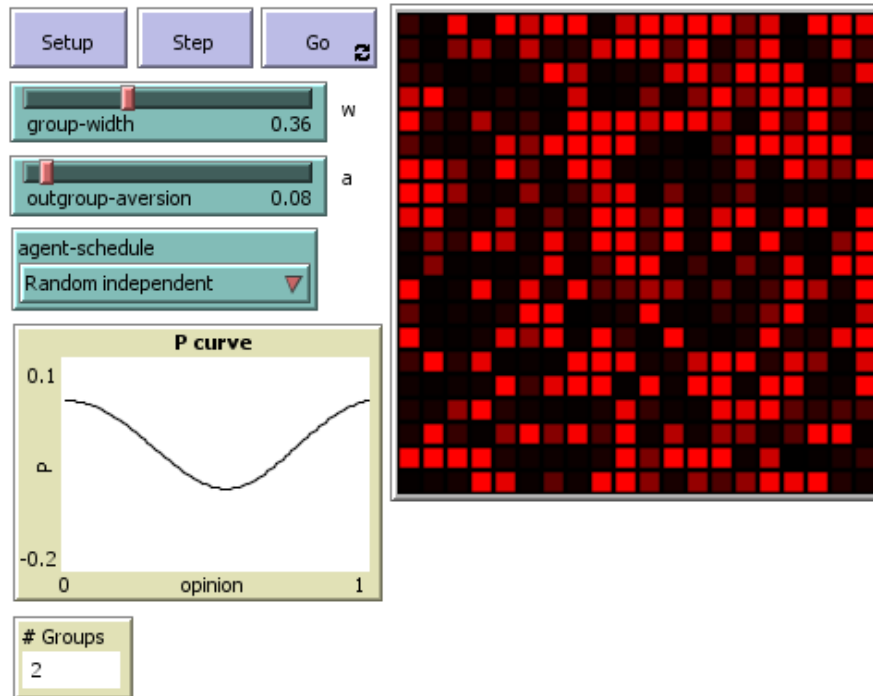
to survey-nearby
  let j one-of other turtles
  let xi opinion
  let ui uncertainty
  let xj [opinion] of j
  let uj [uncertainty] of j
  let hij ( (min list (xi + ui) (xj + uj)) - (max list (xi - ui) (xj -
    uj)) )
  if hij > uj [
    set next-opinion (xi + mu * (hij / uj - 1) * (xj - xi))
    set next-uncertainty (ui + mu * (hij / uj - 1) * (uj - ui))
  ]
  if hij > ui [
    ask j [
      set next-opinion (xj + mu * (hij / ui - 1) * (xi - xj))
      set next-uncertainty (uj + mu * (hij / ui - 1) * (ui - uj))
    ]
  ]
end

to update-opinion
  set opinion next-opinion
  set uncertainty next-uncertainty
  setxy (opinion * 50 + 50) 0
end

to update-plot
  set-current-plot "Opinion over time"
  set-current-plot-pen word "Turtle_" who
  plot-pen-down
  plotxy ticks opinion
  plot-pen-up
end

```

Appendix N. NetLogo Code: Replication of base model of Salzarulo (2006)



```

globals [ proto-opinions N ]

turtles-own [ opinion next-opinion ]

to test-P-curve
  clear-all
  ask patch 0 0 [ sprout 1 [
    set opinion 0.3
    set color hsb (310 * opinion) 100 100
  ] ]
  ask patch 0 1 [ sprout 1 [
    set opinion 0.7
    set color hsb (310 * opinion) 100 100
  ] ]
  set group-width 0.36
  set outgroup-aversion 0.08
  setup-plot
end

to setup

```

```

clear-all
setup-turtles
set N count turtles
setup-plot
reset-ticks
end

to setup-turtles
  set-default-shape turtles "square"
  ask patches [ sprout 1 [
    set opinion random-float 1
    update-color
  ] ]
end

to setup-plot
  set-current-plot "Pcurve"
  create-temporary-plot-pen "Pplot"
  update-proto-opinions
end

to go
  if agent-schedule = "Randomindependent" [ repeat N [ ask one-of turtles
    [ be-influenced ] ] ]
  if agent-schedule = "Randomorder" [ ask turtles [ be-influenced ] ]
  if agent-schedule = "Synchronous" [
    ask turtles [ get-opinion ]
    ask turtles [ update-opinion ]
    update-proto-opinions
  ]
  tick
  if length remove-duplicates [opinion] of turtles = length proto-opinions
    [ stop ]
end

to be-influenced
  get-opinion
  update-opinion
  update-proto-opinions
end

to get-opinion
  let distances map [ [xi] -> abs (opinion - xi) ] proto-opinions
  let ideal-index position (min distances) distances
  let ideal-opinion item ideal-index proto-opinions
  let group-leader min-one-of turtles [ abs (opinion - ideal-opinion) ]

```

```

    set next-opinion [opinion] of group-leader
end

to update-opinion
    set opinion next-opinion
    update-color
end

to update-color
    set color hsb 0 100 (100 * opinion)
end

to update-proto-opinions
    plot-pen-reset
    let index 0
    let protos [0]
    let Plist ( list prototypicality 0 )
    let increasing? true
    foreach n-values 100 [ [i] -> (i + 1) / 100 ] [ [xi] ->
        let P prototypicality xi
        plotxy xi P
        if-else increasing? [
            if-else P > last Plist [
                set Plist replace-item index Plist P
                set protos replace-item index protos xi
            ] [
                set increasing? false
                set index index + 1
                set protos lput xi protos
                set Plist lput P Plist
                if xi = 1 [ set protos but-last protos ]
            ]
        ] [
            if P > last Plist [ set increasing? true ]
            set Plist replace-item index Plist P
            set protos replace-item index protos xi
            if xi = 1 [ set protos but-last protos ]
        ]
    ]
    set proto-opinions protos
end

to-report fuz-mem [ x xi ]
    ;;; fuzzy membership function
    report exp ( - ((x - xi) ^ 2) / group-width ^ 2)
end

```

```

to-report d_intra [x]
  ;;; intra-category distance
  let dnum 0 ; numerator
  let dden 0 ; denominator
  foreach [opinion] of turtles [ [xi] ->
    let mu fuz-mem x xi
    set dnum (dnum + (mu * (x - xi) ^ 2))
    set dden (dden + mu)
  ]
  report dnum / dden
end

```

```

to-report d_inter [x]
  ;;; inter-category distance
  let dnum 0 ; numerator
  let dden 0 ; denominator
  foreach [opinion] of turtles [ [xi] ->
    let mu fuz-mem x xi
    set dnum (dnum + ((1 - mu) * (x - xi) ^ 2))
    set dden (dden + (1 - mu))
  ]
  report dnum / dden
end

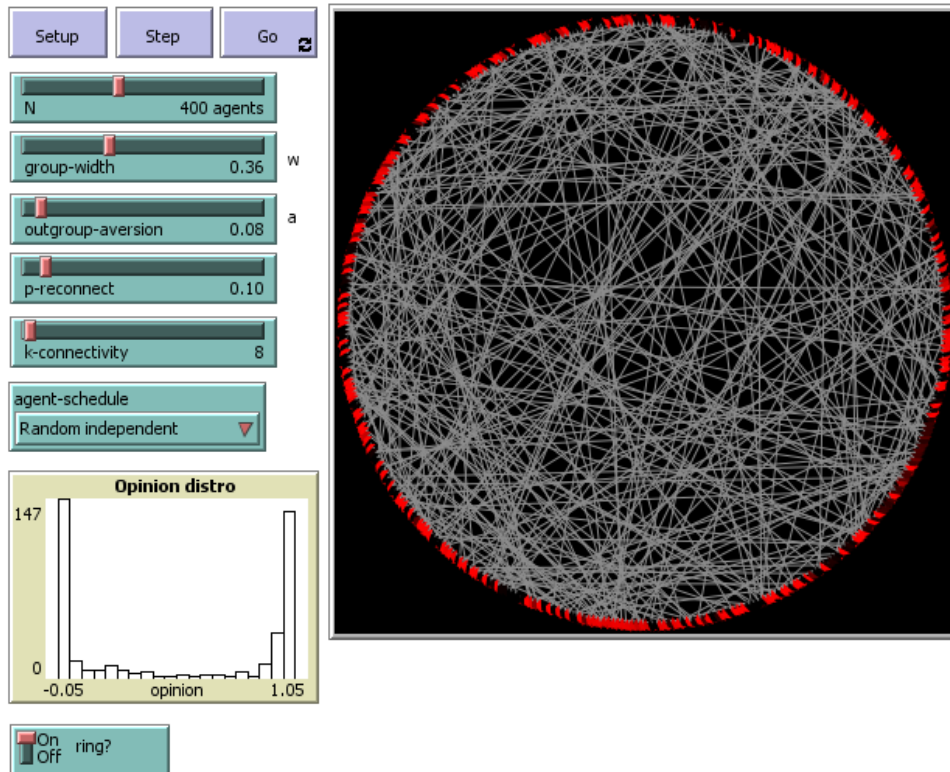
```

```

to-report prototypicality [x]
  report (outgroup-aversion * d_inter x) - ((1 - outgroup-aversion) *
    d_intra x)
end

```

Appendix O. NetLogo Code: Replication of small-world network model of Salzarulo (2006)



```
extensions [ nw ]
```

```
turtles-own [ opinion next-opinion proto-opinions ]
```

```
to setup
  clear-all
  setup-turtles
  reset-ticks
end
```

```
to setup-turtles
  ifelse ring? [
    nw:generate-watts-strogatz turtles links N k-connectivity p-reconnect [
      set opinion random-float 1
      update-color
      setxy (max-pxcor / 2) (max-pycor / 2)
      fd max-pxcor / 2
      face patch (max-pxcor / 2) (max-pycor / 2)
    ]
  ] [ ]
end
```



```

]
] [
  ask patches [ sprout 1 [
    set shape "square"
    set opinion random-float 1
    update-color
  ] ]
ask turtles [ create-links-with turtles-on neighbors ]
ask turtles [
  let r n-values (count my-out-links) [ifelse-value (random-float 1 <
    p-reconnect) [1] [0]]
  ask n-of (sum r) my-links [ die ]
  create-links-with n-of (sum r) (other turtles with [ not
    in-link-neighbor? myself ])
]
ask links [ hide-link ]
]
end

to go
  if agent-schedule = "Random_independent" [ repeat N [ ask one-of turtles
    [ be-influenced ] ] ]
  if agent-schedule = "Random_order" [ ask turtles [ be-influenced ] ]
  if agent-schedule = "Synchronous" [
    ask turtles [ get-opinion ]
    ask turtles [ update-opinion ]
  ]
  let protos remove-duplicates reduce se [proto-opinions] of turtles
  if length remove-duplicates [opinion] of turtles = length protos [stop]
  tick
end

to update-proto-opinions
  let my_neighbors link-neighbors
  set my_neighbors (turtle-set my_neighbors self)
  let index 0
  let protos [0]
  let Plist ( list prototypicality 0 my_neighbors )
  let increasing? true
  foreach n-values 100 [ [i] -> (i + 1) / 100 ] [ [xi] ->
    let P prototypicality xi my_neighbors
    if-else increasing? [
      if-else P > last Plist [
        set Plist replace-item index Plist P
        set protos replace-item index protos xi
      ] [

```

```

    set increasing? false
    set index index + 1
    set protos lput xi protos
    set Plist lput P Plist
    if xi = 1 [ set protos but-last protos ]
  ]
] [
  if P > last Plist [ set increasing? true ]
  set Plist replace-item index Plist P
  set protos replace-item index protos xi
  if xi = 1 [ set protos but-last protos ]
]
]
set proto-opinions protos
end

to be-influenced
  update-proto-opinions
  get-opinion
  update-opinion
end

to get-opinion
  let distances map [ [xi] -> abs (opinion - xi) ] proto-opinions
  let ideal-index position (min distances) distances
  let ideal-opinion item ideal-index proto-opinions
  let group-leader min-one-of turtles [ abs (opinion - ideal-opinion) ]
  set next-opinion [opinion] of group-leader
end

to update-opinion
  set opinion next-opinion
  update-color
end

to update-color
  set color hsb 0 100 (100 * opinion)
end

to-report fuz-mem [ x xi ]
  ;;; fuzzy membership function
  report exp (- ((x - xi) ^ 2) / group-width ^ 2)
end

to-report d_intra [x my_neighbors ]
  ;;; intra-category distance

```

```

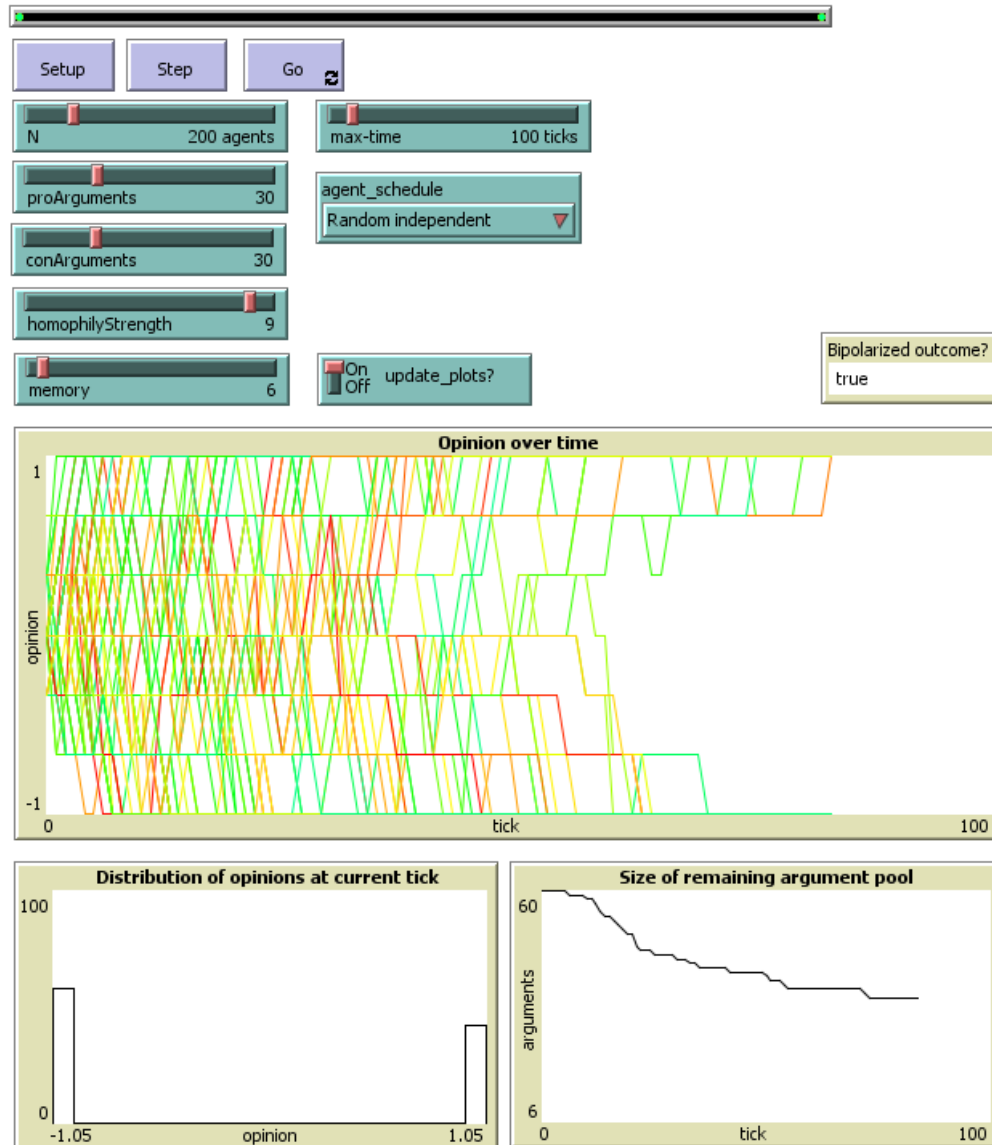
let dnum 0 ; numerator
let dden 0 ; denominator
foreach [opinion] of my_neighbors [ [xi] ->
  let mu fuz-mem x xi
  set dnum (dnum + (mu * (x - xi) ^ 2))
  set dden (dden + mu)
]
report dnum / dden
end

to-report d_inter [x my_neighbors ]
  ;; inter-category distance
  let dnum 0 ; numerator
  let dden 0 ; denominator
  foreach [opinion] of my_neighbors [ [xi] ->
    let mu fuz-mem x xi
    set dnum (dnum + ((1 - mu) * (x - xi) ^ 2))
    set dden (dden + (1 - mu))
  ]
  report dnum / dden
end

to-report prototypicality [x my_neighbors]
  report (outgroup-aversion * d_inter x my_neighbors) - ((1 -
    outgroup-aversion) * d_intra x my_neighbors)
end

```

Appendix P. NetLogo Code: Replication of Mäs & Flache (2013)



```
globals [
  arguments
  groupSplit
]
```

```
turtles-own [
  opinion
  args
```

```

CDF
next-arg
]

to setup
  clear-all
  setup-globals
  setup-turtles
  reset-ticks
  setup-plot
end

to setup-globals
  set arguments sentence (n-values conArguments [-1]) (n-values
    proArguments [1])
end

to setup-turtles
  create-turtles N / 2 [
    let conargs n-of (memory / 2) (n-values conArguments [ [i] -> i ])
    let proargs n-of (memory / 2) (n-values proArguments [ [i] -> i +
      conArguments ])
    set args shuffle sentence conargs proargs
    set opinion mean map [ [i] -> item i arguments ] args
    set color hsb (310 * who / N) 100 100
    setxy (opinion * 50 + 50) 0
  ]
end

to setup-plot
  set-current-plot "Size_of_remaining_argument_pool"
  set-plot-x-range 0 max-time
  set-plot-y-range memory (conArguments + proArguments)

  set-current-plot "Opinion_over_time"
  set-plot-x-range 0 max-time
  ask turtles [
    create-temporary-plot-pen word "Turtle_" who
    set-plot-pen-color color
    plot-pen-up
  ]
end

to go
  if agent_schedule = "Random_independent" [
    repeat count turtles [ ask one-of turtles [ get-influenced ] ]
  ]

```

```

]
if agent_schedule = "Random_order" [
  ask turtles [ get-influenced ]
]
if agent_schedule = "Synchronous" [
  ask turtles [ survey-nearby ]
  ask turtles [ update-opinion ]
]
if update_plots? [ ask turtles [ update-plot ] ]
tick

if stopping-conditions-met? [ stop ]
end

to get-influenced
  survey-nearby
  update-opinion
end

to survey-nearby
  let i random-neighbor
  set next-arg one-of [args] of i
end

to update-opinion
  set args ifelse-value (member? next-arg args) [remove next-arg args]
  [but-first args]
  set args lput next-arg args
  set opinion mean map [ [index] -> item index arguments ] args
  setxy (opinion * 50 + 50) 0
end

to-report remainingArgs [ agents ]
  let remArgs []
  foreach [args] of agents [ [this-args] ->
    set remArgs remove-duplicates sentence remArgs this-args
  ]
  report remArgs
end

to-report random-neighbor
  set CDF []
  let sumS sum [(similarity myself) ^ homophilyStrength] of other turtles
  let pCum 0
  foreach (n-values (count turtles) turtle) [ [other_agent] ->
    if other_agent != self [

```

```

        let p ((similarity other_agent) ^ homophilyStrength) / sumS
        set pCum pCum + p
    ]
    set CDF lput pCum CDF
]
let r random-float 1
report turtle (position (min filter [ [p] -> p >= r ] CDF) CDF)
end

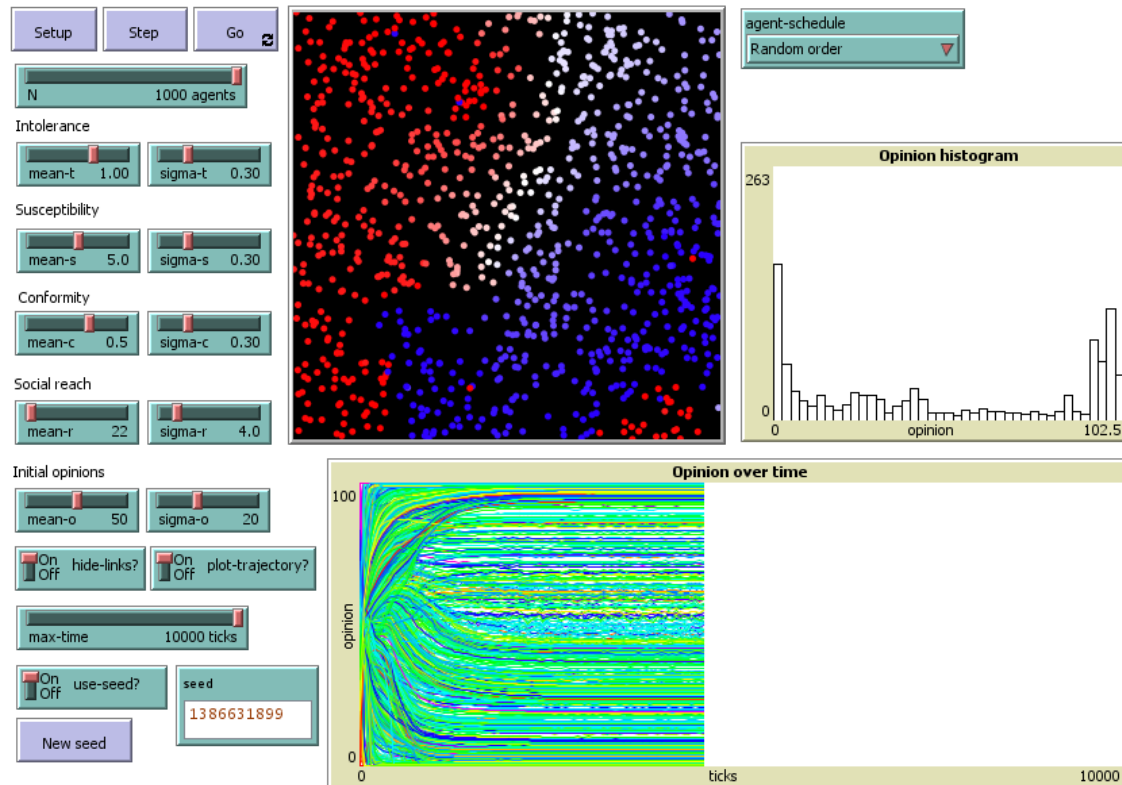
to-report similarity [ other_agent ]
    report 0.5 * (2 - abs (opinion - [opinion] of other_agent) )
end

to update-plot
    set-current-plot "Opinion over time"
    set-current-plot-pen word "Turtle_" who
    plot-pen-down
    plotxy ticks opinion
    plot-pen-up
end

to-report stopping-conditions-met?
    let allOpinions remove-duplicates [opinion] of turtles
    if length allOpinions = 1 [
        if (abs item 0 allOpinions = 1) [
            set groupSplit false
            report true
        ]
        if length (remainingArgs turtles) = memory [
            set groupSplit false
            report true
        ]
    ]
    if (sort allOpinions = [-1 1]) [
        set groupSplit true
        report true
    ]
    report false
end

```

Appendix Q. NetLogo Code: Replication of Duggins (2017)



```
turtles-own [
  true-opinion
  expr-opinion ; expressed opinion
  intolerance ; t_i
  conformity ; c_j
  susceptibility ; s_j
  commitment ; k_j
  reach
  next-opinion ; used to enable Synchronous version
]
```

```
to setup
  clear-all
  random-seed seed
  setup-turtles
  setup-links
  reset-ticks
  setup-plot
end
```



```

to setup-turtles
  create-turtles N [
    set-opinion random-normal mean-o sigma-o
    if agent-schedule = "Synchronous" [ update-opinion ]
    set conformity random-normal mean-c sigma-c
    set intolerance random-normal mean-t sigma-t
    if intolerance < 0 [ set intolerance 0 ]
    set susceptibility random-normal mean-s sigma-s
    if susceptibility < 0 [ set susceptibility 0 ]
    ;set susceptibility conformity ; this implements a bug I found in the
      original code
    set reach random-normal mean-r sigma-r
    if reach < 0 [ set reach 0 ]
    setxy random-xcor random-ycor
    set size 5 ; to make it easier to see
    set shape "circle"
  ]
end

to setup-links
; ask turtles [
; create-links-from other turtles in-radius reach [ set hidden?
  hide-links? ]
; ] ; removed because it's not how he did it
ask turtles [
  let network []
  ask other turtles in-radius reach [
    if distance myself < reach [ create-link-with myself [ set hidden?
      hide-links? ] ]
  ]
]
end

to setup-plot
set-current-plot "Opinion over time"
ask turtles [
  create-temporary-plot-pen word "Turtle_" who
  set-plot-pen-color hsb (310 * (true-opinion / 100 )) 100 100
  plot-pen-down
  plotxy ticks true-opinion
  plot-pen-up
]
end

;;; END SETUP

```

```

to go
  if agent-schedule = "Random_order" [ ask turtles [ initiate-dialogue ] ]
  if agent-schedule = "Random_independent" [ repeat N [ ask one-of turtles
    [ initiate-dialogue ] ] ]
  if agent-schedule = "Synchronous" [
    ask turtles [ initiate-dialogue ]
    ask turtles [ update-opinion ]
  ]
  tick
  if plot-trajectory? [ if ticks mod 50 = 0 [ ask turtles [ update-plot ]
    ] ]
  let all-opinions [ true-opinion ] of turtles
  if ( (max all-opinions) - (min all-opinions) < 1 ) [ stop ]
  if ticks >= max-time [ stop ]
end

to initiate-dialogue
  if any? in-link-neighbors [
    set expr-opinion true-opinion
    let D (list expr-opinion) ; list of opinions expressed so far in this
      dialogue
    let w [] ; list of weights for each expressed opinion other than own
    ask in-link-neighbors [
      set expr-opinion true-opinion + (conformity / commitment) * (mean D -
        true-opinion)
      if expr-opinion < 0 [ set expr-opinion 0 ]
      if expr-opinion > 100 [ set expr-opinion 100 ]
      set D lput expr-opinion D
      let wij (1 - [intolerance] of myself * abs (expr-opinion -
        [true-opinion] of myself) / 50)
      if wij < -1 [ set wij -1 ]
      if wij > 1 [ set wij 1 ]
      set w lput wij w
    ]
    set D but-first D ; remove own opinion from list of influences
    let influence (sum (map [[ ej wij ] -> wij * (ej - true-opinion)] D
      w)) / (sum map abs w)
    set-opinion (true-opinion + influence / commitment)
  ]
end

to set-opinion [ new-op ]
  set next-opinion new-op
  if next-opinion < 0 [ set next-opinion 0 ]
  if next-opinion > 100 [ set next-opinion 100 ]
  if agent-schedule != "Synchronous" [ update-opinion ]

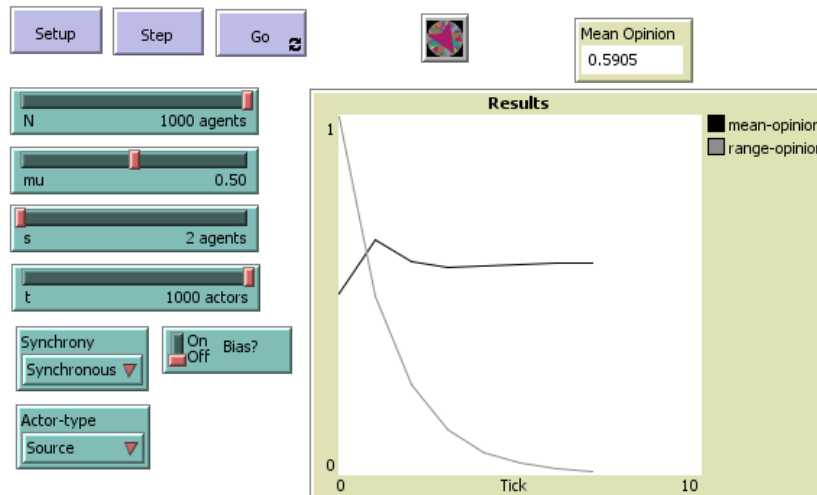
```

```
end

to update-opinion
  set true-opinion next-opinion
  set commitment 1 + susceptibility * (abs (50 - true-opinion)) / 50
  let hue ifelse-value (true-opinion < 50) [ 250 ] [ 0 ]
  let saturation 2 * abs (50 - true-opinion)
  set color hsb hue saturation 100
end

to update-plot
  set-current-plot "Opinion over time"
  set-current-plot-pen word "Turtle_" who
  plot-pen-down
  plotxy ticks true-opinion
  plot-pen-up
end
```

Appendix R. NetLogo Code: Generalized Repeated Averaging Model



```

globals [ actors opinions range-op all-pairs num-prim-actors
          num-sec-actors]

turtles-own [ opinion sources targets next-opinion coeff ]

to setup
  clear-all
  create-turtles N [
    set opinion random-float 1
  ]
  set opinions [opinion] of turtles
  set range-op ( max opinions - min opinions )
  if Actor-type = "Source" [
    set num-prim-actors (min (list s N))
    set num-sec-actors (min (list t (N - 1)))
  ]
  if Actor-type = "Target" [
    set num-prim-actors (min (list t N))
    set num-sec-actors (min (list s (N - 1)))
  ]
  if Actor-type = "Group" [
    set num-prim-actors (min (list t (N * (N - 1) / 2)))
    set all-pairs []
    foreach range N [ [who1] ->
      foreach (range (who1 + 1) N) [ [who2] ->
        set all-pairs lput (turtle-set (turtle who1) (turtle who2))
          all-pairs
      ]
    ]
  ]

```

```

    ]
  ]
]
reset-ticks
end

to go
  ask turtles [
    set sources no-turtles
    set targets no-turtles
  ]
  ; Assign relevant sets S, T, S_j, T_i as needed by actor type
  ; Note that if-values exist only to speed up calculations when
  ; parameters are infinite
  if Actor-type = "Source" [
    set actors ifelse-value (num-prim-actors = N)
      [turtles] [n-of num-prim-actors turtles] ; Set S
    ask actors [
      set targets ifelse-value (num-sec-actors = N - 1)
        [other turtles] [n-of num-sec-actors other turtles] ; Set T_i
      if (num-prim-actors < N) [ask targets [
        set sources (turtle-set sources myself) ; Set S_j
      ] ]
    ]
  ]
  if (num-prim-actors = N) [ask turtles [
    set sources other turtles ; Set S_j
  ] ]
]
  if Actor-type = "Target" [
    set actors ifelse-value (num-prim-actors = N)
      [turtles] [n-of num-prim-actors turtles] ; Set S
    ask actors [
      set sources ifelse-value (num-sec-actors = N - 1)
        [other turtles] [n-of num-sec-actors other turtles] ; Set S_j
    ]
  ]
]
  if Actor-type = "Group" [
    set actors ifelse-value (num-prim-actors = length all-pairs)
      [all-pairs] [n-of num-prim-actors all-pairs] ; Set A
    foreach actors [ [this-pair] ->
      ask this-pair [
        set sources (turtle-set sources other this-pair) ; Set S_j
      ]
    ]
  ]
]
if-else Synchrony = "Synchronous" [ go-sync ] [ go-async ]

```

```

set opinions [opinion] of turtles
set range-op ( max opinions - min opinions )
tick
if range-op < 0.01 [ stop ]
end

to go-sync
let target-set no-turtles ; set of agents with |S_j|>0
if-else Actor-type = "Source" [
  set target-set ifelse-value (num-sec-actors = N - 1)
    [turtles] [turtle-set [targets] of actors]
] [
  set target-set ifelse-value (num-prim-actors = ifelse-value
    (Actor-type = "Target") [N] [length all-pairs])
    [turtles] [turtle-set actors]
]
ask target-set [
  set next-opinion ( (1 - mu) * opinion + mu * mean [opinion] of sources
  )
]
ask target-set [
  set opinion next-opinion
]
end

to go-async
ask turtles with [ count sources > 0 ] [ set coeff ((1 - mu) ^ (1 /
  count sources)) ] ; coeff = 1 - mu^*
if Actor-type = "Source" [
  let sourcelist ifelse-value Bias? [
    ; sort agents in ascending order of opinion
    sort-on [opinion] actors
  ] [
    ; keep agents in random order
    [self] of actors
  ]
  foreach sourcelist [ [this-source] ->
    ask this-source [ ask targets [ ; Set T_i
      set opinion coeff * opinion + (1 - coeff) * [opinion] of this-source
    ] ]
  ]
]
if Actor-type = "Target" [
  ask actors [
    let sourcelist ifelse-value Bias? [
      ; sort agents in descending order of opinion

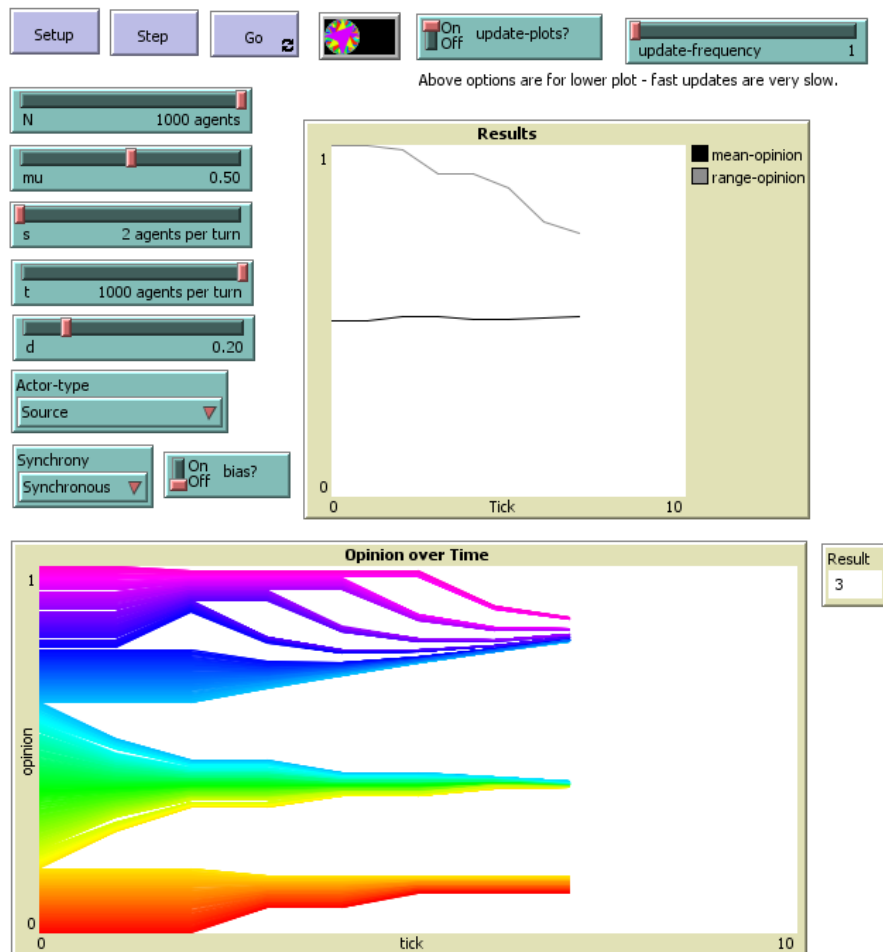
```

```

    reverse (sort-on [opinion] sources)
  ] [
    ; keep agents in random order
    [self] of sources
  ]
  foreach sourcelist [ [this-source] ->
    set opinion coeff * opinion + (1 - coeff) * [opinion] of this-source
  ]
]
]
if Actor-type = "Group" [ ; Note, this only works for s=2
; if Bias, sort in ascending order by mean opinion
if Bias? [ set actors sort-by [ [set1 set2] -> mean [opinion] of set1
  < mean [opinion] of set2 ] actors ]
foreach actors [ [this-pair] ->
; Synchronously update both members' opinions
ask this-pair [
  let source-op (item 0 [opinion] of other this-pair)
  set next-opinion coeff * opinion + (1 - coeff) * source-op
]
ask this-pair [
  set opinion next-opinion
]
]
]
end

```

Appendix S. NetLogo Code: Generalized Bounded Confidence Model



```
globals [ actors opinions range-op all-pairs num-prim-actors
          num-sec-actors final-clusters final-opinions cluster-count]
```

```
turtles-own [ opinion sources targets next-opinion coeff my-cluster ]
```

```
to setup
  clear-all
  create-turtles N [
    set opinion random-float 1
  ]
  setxy opinion 0
  set color hsb (310 * opinion) 100 100
  set shape "dot"
]
set opinions [opinion] of turtles
```



```

set range-op ( max opinions - min opinions )
if Actor-type = "Source" [
  set num-prim-actors (min (list s N))
  set num-sec-actors (min (list t (N - 1)))
]
if Actor-type = "Target" [
  set num-prim-actors (min (list t N))
  set num-sec-actors (min (list s (N - 1)))
]
if Actor-type = "Group" [
  set num-prim-actors (min (list t (N * (N - 1) / 2)))
  set all-pairs []
  foreach range N [ [who1] ->
    foreach (range (who1 + 1) N) [ [who2] ->
      set all-pairs lput (turtle-set (turtle who1) (turtle who2))
      all-pairs
    ]
  ]
]
reset-ticks
if update-plots? [ setup-plot ]
end

to setup-plot
set-current-plot "Opinion over time"
set-plot-x-range 0 (10 * update-frequency)
ask turtles [
  create-temporary-plot-pen word "Turtle_" who
  set-plot-pen-color color
  plot-pen-up
]
if update-plots? [ ask turtles [ update-plot ] ]
end

to go
ask turtles [
  set sources no-turtles
  set targets no-turtles
]
; Assign relevant sets S, T, S_j, T_i as needed by actor type
; Note that if-values exist only to speed up calculations when
; parameters are infinite
if Actor-type = "Source" [
  set actors ifelse-value (num-prim-actors = N)
    [turtles] [n-of num-prim-actors turtles] ; Set S
  ask actors [

```

```

set targets ifelse-value (num-sec-actors = N - 1)
  [other turtles] [n-of num-sec-actors other turtles] ; Set T_i
set targets targets with [abs (opinion - [opinion] of myself) < d]
ifelse (num-prim-actors = N and num-sec-actors = N - 1) [
  ; unlimited actors means sources = targets (runs much faster)
  set sources targets ; Set S_j
] [
  ask targets [
    set sources (turtle-set sources myself) ; Set S_j
  ]
]
]
]
if Actor-type = "Target" [
  set actors ifelse-value (num-prim-actors = N)
    [turtles] [n-of num-prim-actors turtles] ; Set T
  ask actors [
    set sources n-of num-sec-actors other turtles ; Set S_j
    set sources sources with [abs (opinion - [opinion] of myself) < d]
  ]
]
if Actor-type = "Group" [
  ; filter pairs based on opinion similarity
  set actors ifelse-value (num-prim-actors = length all-pairs)
    [all-pairs] [n-of num-prim-actors all-pairs] ; Set A
  set actors filter [ [pair] -> max [opinion] of pair - min [opinion] of
    pair < d ] actors
  foreach actors [ [this-pair] ->
    ask this-pair [
      set sources (turtle-set sources other this-pair) ; Set S_j
    ]
  ]
]
if-else Synchrony = "Synchronous" [ go-sync ] [ go-async ]

; Update plots and plotted values
set opinions [opinion] of turtles
set range-op ( max opinions - min opinions )
tick
if update-plots? [ if (ticks mod update-frequency = 0) [ ask turtles [
  update-plot ] ] ]

if check-clusters [
  if update-plots? [ ask turtles [ update-plot ] ]
  stop
]

```



```

        reverse (sort-on [opinion] sources)
    ] [
        ; keep agents in random order
        [self] of sources
    ]
    foreach sourcelist [ [this-source] ->
        set opinion coeff * opinion + (1 - coeff) * [opinion] of this-source
; setxy opinion 0
    ]
]
]
if Actor-type = "Group" [
    ; if Bias, sort in ascending order by mean opinion
    if Bias? [ set actors sort-by [ [set1 set2] -> mean [opinion] of set1
        < mean [opinion] of set2 ] actors ]
    foreach actors [ [this-pair] ->
        ; Synchronously update both members' opinions
        ask this-pair [
            let source-op (item 0 [opinion] of other this-pair)
            set next-opinion coeff * opinion + (1 - coeff) * source-op
        ]
        ask this-pair [
            set opinion next-opinion
; setxy opinion 0
        ]
    ]
]
end

```

```

to update-plot
    set-current-plot "Opinion over time"
    set-current-plot-pen word "Turtle" who
    plot-pen-down
    plotxy ticks opinion
    plot-pen-up
end

```

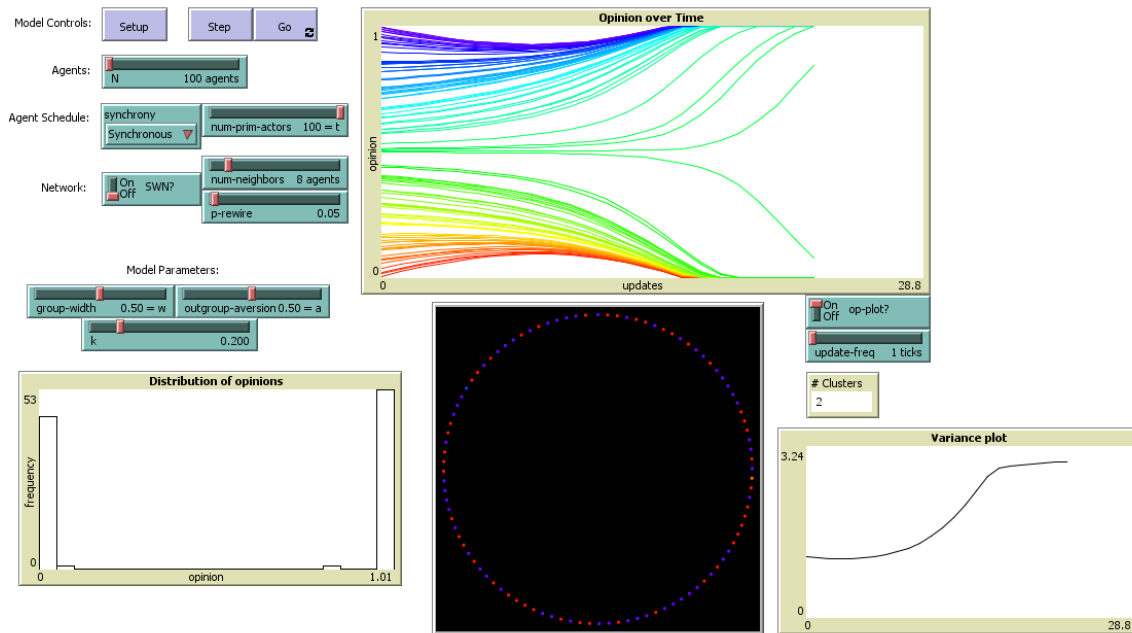
```

to-report check-clusters
    ; Stop at convergence within clusters
    let cluster-width 0
    let clusters []
    ask turtles [
        let cluster turtles with [abs (opinion - [opinion] of myself) < d]
        set clusters sentence clusters cluster
        set cluster-width max list cluster-width (max [opinion] of cluster -
            min [opinion] of cluster)
    ]

```

```
]
if cluster-width < d / 2 [
  set clusters remove-duplicates clusters
  set final-clusters length clusters
  set final-opinions sort map [ [clus] -> mean [opinion] of clus ]
    clusters
  set cluster-count map [ [op] -> count turtles with [abs (opinion - op)
    < d] ] final-opinions
  report true
]
report false
end
```

Appendix T. NetLogo Code: Meta-contrast Influence Field Model



```

extensions [ nw ]

globals [ clusters cluster-ops ]

turtles-own [ opinion next-opinion ]

;;;;; SETUP ;;;;;

to setup
  clear-all
  setup-turtles
  reset-ticks
  setup-plot
end

to setup-turtles
  set-default-shape turtles "dot"
  ifelse SWN? [ ; generate small-world network using Watts-Strogatz method
    nw:generate-watts-strogatz turtles links N num-neighbors p-rewire [
      set size 0.5
      fd min list max-pxcor max-pycor
      initialize-turtle
    ]
  ]

```

```

] [ ; else, don't bother with links
  create-turtles N [
    set size 0.5
    set heading (who / N) * 359
    fd min list max-pxcor max-pycor
    initialize-turtle
  ]
]
end

to initialize-turtle
  set opinion random-float 1
  set next-opinion opinion
  set color hsb (260 * opinion + 5) 100 100
end

to setup-plot
  set-current-plot "Opinion over time"
  ask turtles [
    create-temporary-plot-pen word "Turtle_" who
    set-plot-pen-color color
  ]
  ask turtles [ update-plot ]
end

;;;;; GO ;;;;;

to go
  let prim-actors ifelse-value (num-prim-actors >= N) [ turtles ] [ n-of
    num-prim-actors turtles ]
  ask prim-actors [
    be-influenced
  ]
  if synchrony = "Synchronous" [ ask prim-actors [ update-opinion ] ]
  tick
  if op-plot? [ if ticks mod update-freq = 0 [ ask turtles [ update-plot ] ] ]
  if not SWN? [ update-clusters ]
  if not SWN? [ if max map [ [y] -> max [opinion] of y - min [opinion] of
    y ] clusters <= (group-width / 2) [stop] ]
  if count turtles with [opinion >= 0.0001 and opinion <= 0.9999] = 0 [
    stop ]
  if max [opinion] of turtles - min [opinion] of turtles < 0.01 [ stop ]
  if SWN? and ticks > 0 and ticks mod 50 = 0 [ if max [ abs (get-influence
    get-ops)] of turtles with [opinion > 0 and opinion < 1] <=
    (group-width / 100) [ stop ] ]

```

```

end

;;;;; PROCEDURES ;;;;;

;; OBSERVER PROCEDURES ;;

to update-clusters
  let old-clusters clusters
  set clusters remove-duplicates [ turtles with [abs (opinion - [opinion]
    of myself) < group-width] ] of turtles
  set cluster-ops map [ [y] -> mean [opinion] of y ] clusters
end

;; TURTLE PROCEDURES ;;

to update-plot
  set-current-plot "Opinion over time"
  set-current-plot-pen word "Turtle_" who
  plotxy ticks opinion
end

to update-opinion
  if next-opinion > 1 [ set next-opinion 1 ]
  if next-opinion < 0 [ set next-opinion 0 ]
  set opinion next-opinion
  set color hsb (260 * opinion + 5) 100 100
end

to be-influenced
  let ops get-ops
  let influence get-influence ops
  set next-opinion next-opinion + influence
  if synchrony = "Asynchronous" [ update-opinion ]
end

;;;;; FUNCTIONS ;;;;;

;; OBSERVER FUNCTIONS ;;

to-report membership [x xi] ; mu(x,x_i) in Salzarulo (2006)
  ;; Note: this assumes agents are unaware of group membership (may not be
  appropriate in all cases)
  let w group-width
  report exp (- ((x - xi) ^ 2 / (w ^ 2)))
end

```



```

to-report prototypicality [ x ops ] ; P(x,X) in Salzarulo (2006), modified
  with lambda multiplier
  let a outgroup-aversion
  let w group-width
  let mus map [[xi] -> membership x xi] ops
  let diffs2 map [[y] -> (x - item y ops) ^ 2 ] range length ops
  let summus sum mus ; sum(mu(x,xi))
  let diff2mus sum map [[y] -> (item y mus) * (item y diffs2)] range
    length ops; sum((x-xi)^2 mu(x,xi))
  let summus2 (length ops) - summus ; sum(1-mu(x,xi))
  let diff2mus2 sum map [[y] -> (1 - item y mus) * (item y diffs2)] range
    length ops; sum((x-xi)^2 (1-mu(x,xi)))
  let dintra ifelse-value (summus = 0) [0] [diff2mus / summus]
  let dinter ifelse-value (summus2 = 0) [0] [diff2mus2 / summus2]
  let lambda (w ^ 2) / ((exp 1) - (exp (1 - (1 / (w ^ 2)))))
  let P (a * lambda * dinter - (1 - a) * dintra)
  report P
end

```

```

to-report P-deriv [ x ops ] ; (d/dx)P(x,X) as specified above, with lambda
  multiplier
  let a outgroup-aversion
  let w group-width
  let mus map [[xi] -> membership x xi] ops
  let diffs map [[xi] -> x - xi] ops
  let diffs2 map [[y] -> (item y diffs) ^ 2 ] range length ops
  let diffs3 map [[y] -> (item y diffs) ^ 3 ] range length ops
  let summus sum mus ; sum(mu(x,xi))
  let diffmus sum map [[y] -> (item y mus) * (item y diffs)] range length
    ops ; sum((x-xi)mu(x,xi))
  let diff2mus sum map [[y] -> (item y mus) * (item y diffs2)] range
    length ops; sum((x-xi)^2 mu(x,xi))
  let diff3mus sum map [[y] -> (item y mus) * (item y diffs3)] range
    length ops; sum((x-xi)^3 mu(x,xi))
  let summus2 (length ops) - summus ; sum(1-mu(x,xi)) = n - sum(mu(x,xi))
  let diffmus2 sum map [[y] -> (1 - item y mus) * (item y diffs)] range
    length ops ; sum((x-xi)(1-mu(x,xi)))
  let diff2mus2 sum map [[y] -> (1 - item y mus) * (item y diffs2)] range
    length ops; sum((x-xi)^2 (1-mu(x,xi)))
  let ddintradx ifelse-value (summus = 0) [0] [2 * (((((w ^ 2) * diffmus)
    - diff3mus) / ((w ^ 2) * summus)) + ((diff2mus * diffmus) / ((w *
    summus) ^ 2)))]
  let ddinterdx ifelse-value (summus2 = 0) [0] [2 * (((((w ^ 2) *
    diffmus2) + diff3mus) / ((w ^ 2) * summus2)) - ((diff2mus2 * diffmus)
    / ((w * summus2) ^ 2)))]
  let lambda (w ^ 2) / ((exp 1) - (exp (1 - (1 / (w ^ 2)))))

```

```

    let dPdx (a * lambda * ddinterdx - (1 - a) * ddintradx)
    report dPdx
end

to-report influence-MIF [ x ops ] ; converts P-deriv to actual influence
    value
    let influence P-deriv x ops
    set influence k * influence
    report influence
end

to-report norm-var ; report normalized variance of opinions
    let var variance [opinion] of turtles ; NetLogo calculates the sample
        variance (not population)
    report 12 * var * (N - 1) / N ; Convert to pop variance, divide by 1/12
end

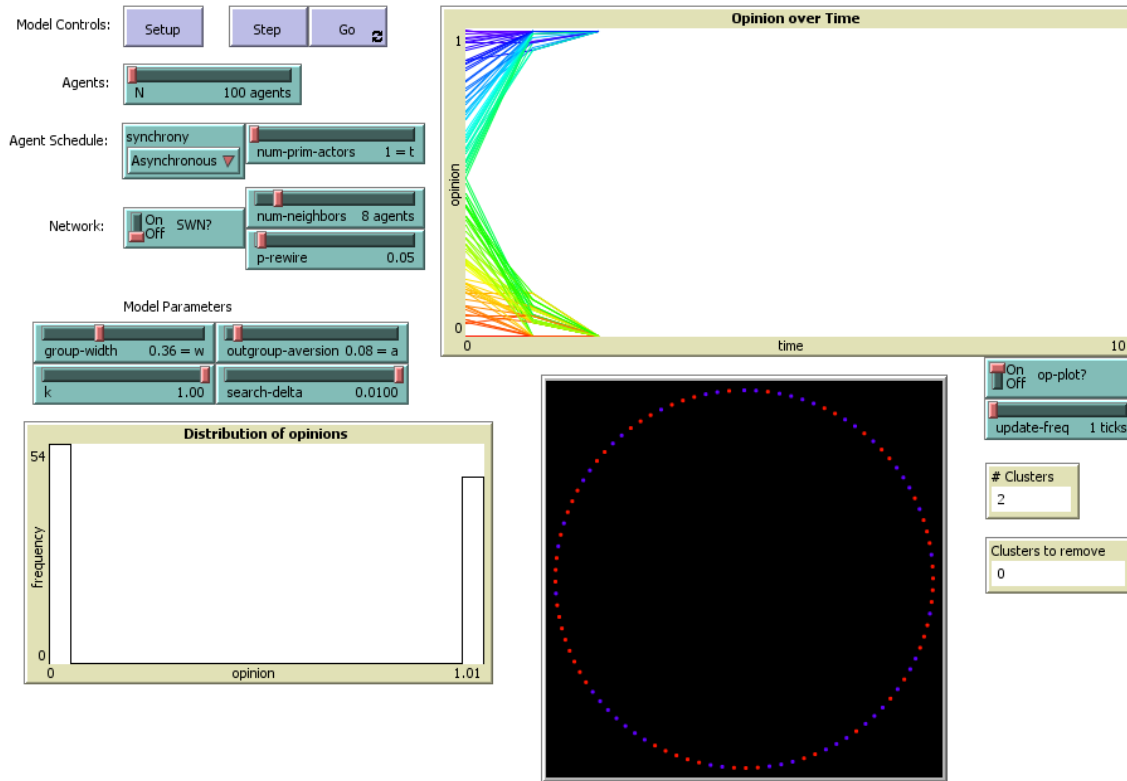
;; TURTLE FUNCTIONS ;;

to-report get-ops ; gets appropriate opinions, based on setting of SWN?
    let ops []
    ifelse SWN? [ set ops [opinion] of (turtle-set link-neighbors self) ] [
        set ops [opinion] of turtles ]
    report ops
end

to-report get-influence [ ops ] ; gets appropriate influence value. Added
    to increase modularity.
    let influence influence-MIF opinion ops
    report influence
end

```

Appendix U. NetLogo Code: Meta-contrast Model



```

extensions [ nw ]

globals [ MC-list clusters cluster-ops ]

turtles-own [ opinion next-opinion ]

;;;;; SETUP ;;;;;;

to setup
  clear-all
  setup-turtles
  reset-ticks
  setup-plot
  set search-delta min list search-delta (1 / round (2 / group-width))
end

to setup-turtles
  set-default-shape turtles "dot"
  ifelse SWN? [ ; generate small-world network using Watts-Strogatz method
    nw:generate-watts-strogatz turtles links N num-neighbors p-rewire [

```

```

        set size 0.5
        fd min list max-pxcor max-pycor
        initialize-turtle
    ]
] [ ; else, don't bother with links
create-turtles N [
    set size 0.5
    set heading (who / N) * 359
    fd min list max-pxcor max-pycor
    initialize-turtle
]
]
end

to initialize-turtle
    set opinion random-float 1
    set next-opinion opinion
    set color hsb (260 * opinion + 5) 100 100
end

to setup-plot
    set-current-plot "Opinion over time"
    ask turtles [
        create-temporary-plot-pen word "Turtle_" who
        set-plot-pen-color color
    ; plot-pen-up
    ]
    ask turtles [ update-plot ]
end

;;;;; GO ;;;;;

to go
    let prim-actors ifelse-value (num-prim-actors >= N) [ turtles ] [ n-of
        num-prim-actors turtles ]
    if (not SWN?) and synchrony = "Synchronous" [set-noSWN-source]
    ask prim-actors [
        if (not SWN?) and synchrony = "Asynchronous" [set-noSWN-source]
        be-influenced
    ]
    if synchrony = "Synchronous" [ ask prim-actors [ update-opinion ] ]
    tick
    if op-plot? [ if ticks mod update-freq = 0 [ ask turtles [ update-plot ] ] ]
    update-clusters
    let max-op max [opinion] of turtles

```

```

let min-op min [opinion] of turtles
if count turtles with [opinion = min-op or opinion = max-op] = N [ stop ]
if max-op - min-op < 0.01 [ stop ]
if not SWN? and k < 1 [ if max map [ [y] -> max [opinion] of y - min
  [opinion] of y ] clusters <= (group-width / 2) [stop] ]
if not SWN? [ if length remove-duplicates [opinion] of turtles <= length
  MC-list [ stop ] ]
end

;;;;; PROCEDURES ;;;;;

;; OBSERVER PROCEDURES ;;

to set-noSWN-source ; pre-calculates values for fully connected values
set MC-list []
let ops [opinion] of turtles
; create list of prototypicality values
let inputs map [ [y] -> y * search-delta ] range ((1 / search-delta) + 1)
let x-list []
let P-list []
foreach inputs [ [x] ->
  set x-list lput x x-list
  set P-list lput (prototypicality x ops) P-list
]
; eliminate values that are not local maxima
let decreasing? false
foreach range (length x-list) [ [index] ->
  ifelse decreasing? [
    if index < length x-list - 1 [
      if (item index P-list) < (item (index + 1) P-list) [
        set decreasing? false
      ]
    ]
  ] [ ; not decreasing?
  ifelse index = length x-list - 1 [
    set MC-list lput last x-list MC-list
  ] [
    if (item index P-list) > (item (index + 1) P-list) [
      set MC-list lput (item index x-list) MC-list
      set decreasing? true
    ]
  ]
]
]
end

```

```

to update-clusters
  let old-clusters clusters
  set clusters remove-duplicates [ turtles with [abs (opinion - [opinion]
    of myself) < group-width] ] of turtles
  set cluster-ops map [ [y] -> mean [opinion] of y ] clusters
end

;; TURTLE PROCEDURES ;;

to update-plot
  set-current-plot "Opinion over time"
  set-current-plot-pen word "Turtle" who
  plotxy ticks opinion
end

to update-opinion
  set opinion next-opinion
  if opinion > 1 [ set opinion 1 ]
  if opinion < 0 [ set opinion 0 ]
  set color hsb (260 * opinion + 5) 100 100
end

to be-influenced
  let ops get-ops
  let influence get-influence ops
  set next-opinion next-opinion + influence
  if synchrony = "Asynchronous" [ update-opinion ]
end

;;;;; FUNCTIONS ;;;;;;

;; OBSERVER FUNCTIONS ;;

to-report membership [x xi] ; mu(x,x_i) in Salzarulo (2006)
  ;; Note: this assumes agents are unaware of group membership (may not be
  appropriate in all cases)
  let w group-width
  report exp (- ((x - xi) ^ 2 / (w ^ 2)))
end

to-report prototypicality [ x ops ] ; P(x,X) in Salzarulo (2006)
  let a outgroup-aversion
  let w group-width
  let mus map [[xi] -> membership x xi] ops
  let diffs2 map [[y] -> (x - item y ops) ^ 2 ] range length ops
  let summus sum mus ; sum(mu(x,xi))

```

```

let diff2mus sum map [[y] -> (item y mus) * (item y diffs2)] range
    length ops; sum((x-xi)^2 mu(x,xi))
let summus2 (length ops) - summus ; sum(1-mu(x,xi))
let diff2mus2 sum map [[y] -> (1 - item y mus) * (item y diffs2)] range
    length ops; sum((x-xi)^2 (1-mu(x,xi)))
let dintra ifelse-value (summus = 0) [0] [diff2mus / summus]
let dinter ifelse-value (summus2 = 0) [0] [diff2mus2 / summus2]
let P (a * dinter - (1 - a) * dintra)
report P
end

to-report influence-MC [ x ops ]
let proto-ops []
ifelse SWN? [
    foreach list (- search-delta) search-delta [ [eps] -> ; find local
        maxima in each direction
    let proto-op x
    let P prototypicality proto-op ops
    let decreasing? true ; becomes false once first local minimum is found
    let increasing? true ; becomes false once first local maximum is found
    while [increasing? and (proto-op >= 0) and (proto-op <= 1)] [
        set proto-op (proto-op + eps)
        let Pnew prototypicality proto-op ops
        ifelse decreasing? [
            if Pnew > P [ set decreasing? false ]
        ] [ ; not decreasing
            if Pnew < P [ ; last value was the local maximum
                set increasing? false
                set proto-op (proto-op - eps)
            ]
        ]
        set P Pnew
    ]
    if not decreasing? [ set proto-ops lput proto-op proto-ops ]
] [ ; fully connected model
set proto-ops MC-list
]
let proto-distance (map [ [y] -> abs (x - y) ] proto-ops)
let my-proto-op item (position (min proto-distance) proto-distance)
proto-ops
let source-op 0
let source-distance min map [ [y] -> abs (y - my-proto-op) ] ops
set source-op item 0 (filter [ [y] -> abs (y - my-proto-op) =
source-distance] ops)

```

```

    if source-op > 1 [ set source-op 1 ]
    if source-op < 0 [ set source-op 0 ]
    let influence k * (source-op - x)
    report influence
end

to-report skewness [ xlist ]
  let len length xlist
  let xbar mean xlist
  let skew (sum map [ [x] -> (x - xbar) ^ 3 ] xlist) / ( len * (((len - 1)
    / len) * variance xlist) ^ 1.5)
  report skew
end

to-report kurtosis [ xlist ]
  let len length xlist
  let xbar mean xlist
  let kurt (sum map [ [x] -> (x - xbar) ^ 4 ] xlist) / ( len * (((len - 1)
    / len) * variance xlist) ^ 2)
  report kurt
end

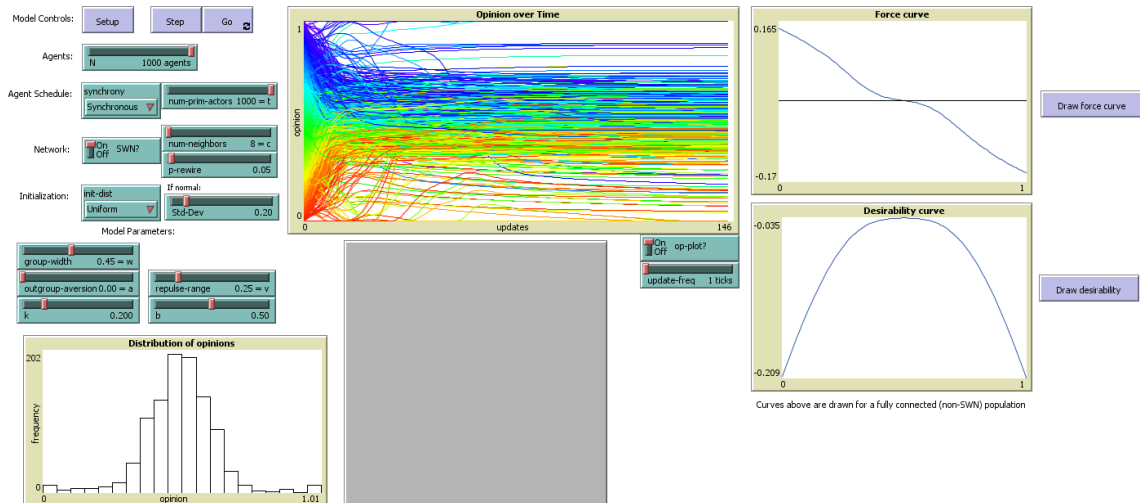
;; TURTLE FUNCTIONS ;;

to-report get-ops
  let ops []
  ifelse SWN? [ set ops [opinion] of (turtle-set link-neighbors self) ] [
    set ops [opinion] of turtles ]
  report ops
end

to-report get-influence [ ops ]
  let influence influence-MC opinion ops
  report influence
end

```


Appendix V. NetLogo Code: Meta-contrast Influence Field - Local Repulsion Model



```
extensions [ nw ]
```

```
globals [ clusters cluster-ops ]
```

```
turtles-own [ opinion next-opinion ]
```

```
;;;;; SETUP ;;;;;;
```

```
to setup
  clear-all
  setup-turtles
  reset-ticks
  setup-plot
end
```

```
to setup-turtles
  set-default-shape turtles "dot"
  ifelse SWN? [ ; generate small-world network using Watts-Strogatz method
    nw:generate-watts-strogatz turtles links N num-neighbors p-rewire [
      set size 0.5
      fd min list max-pxcor max-pycor
      initialize-turtle
    ]
  ] [ ; else, don't bother with links
    create-turtles N [
      set size 0.5

```

```

        set heading (who / N) * 359
        fd min list max-pxcor max-pycor
        initialize-turtle
    ]
]
end

to initialize-turtle
    if init-dist = "Uniform" [set next-opinion random-float 1]
    if init-dist = "Normal" [set next-opinion random-normal 0.5 Std-Dev]
    update-opinion
end

to setup-plot
    set-current-plot "Opinion over time"
    ask turtles [
        create-temporary-plot-pen word "Turtle_" who
        set-plot-pen-color color
    ]
    ask turtles [ update-plot ]
end

;;;;; GO ;;;;;

to go
    let prim-actors ifelse-value (num-prim-actors >= N) [ turtles ] [ n-of
        num-prim-actors turtles ]
    ask prim-actors [
        be-influenced
    ]
    if synchrony = "Synchronous" [ ask prim-actors [ update-opinion ] ]
    tick
    if op-plot? [ if ticks mod update-freq = 0 [ ask turtles [ update-plot ] ] ]
end

;;;;; PROCEDURES ;;;;;

;; OBSERVER PROCEDURES ;;

to draw-force-curve
    set-current-plot "Force curve"
    clear-plot
    foreach range 101 [ [x] ->
        let ops [opinion] of turtles
        let y (influence-MIF (x / 100) ops)
    ]
end

```

```

    set-current-plot-pen "default"
    plotxy (x / 100) y
    set-current-plot-pen "zero"
    plotxy (x / 100) 0
  ]
end

to draw-D-curve
  set-current-plot "Desirability_curve"
  clear-plot
  let ops [opinion] of turtles
  let P map [ [x] -> desirability (x / 100) ops ] range 101
  let min-P (floor (1000 * min P)) / 1000
  let max-P (ceiling (1000 * max P)) / 1000
  set-plot-y-range min-P max-P
  let x 0
  set-current-plot-pen "default"
  foreach P [ [y] ->
    plotxy x y
    set x (x + 0.01)
  ]
  set-plot-x-range 0 1
end

;; TURTLE PROCEDURES ;;

to update-plot
  set-current-plot "Opinion_over_time"
  set-current-plot-pen word "Turtle_" who
  plotxy ticks opinion
end

to update-opinion
  if next-opinion > 1 [ set next-opinion 1 ]
  if next-opinion < 0 [ set next-opinion 0 ]
  set opinion next-opinion
  set color hsb (260 * opinion + 5) 100 100
end

to be-influenced
  let ops get-ops
  let influence get-influence ops
  set next-opinion next-opinion + influence
  if synchrony = "Asynchronous" [ update-opinion ]
end

```

```

;;;;; FUNCTIONS ;;;;;

;; OBSERVER FUNCTIONS ;;

to-report membership [x xi] ; mu(x,x_i) in Salzarulo (2006)
  ;; Note: this assumes agents are unaware of group membership (may not be
  appropriate in all cases)
  let w group-width
  report exp (- ((x - xi) ^ 2 / (w ^ 2)))
end

to-report saturation [x xi] ; same as membership but different value for w
  let v repulse-range * group-width
  report exp (- ((x - xi) ^ 2 / (v ^ 2)))
end

to-report desirability [ x ops ]
  let a outgroup-aversion
  let w group-width
  let v repulse-range * w

  let mus map [[xi] -> membership x xi] ops

  let diffs2 map [[y] -> (x - item y ops) ^ 2 ] range length ops
  let summus sum mus ; sum(mu(x,xi))
  let diff2mus sum map [[y] -> (item y mus) * (item y diffs2)] range
    length ops; sum((x-xi)^2 mu(x,xi))
  let dintra ifelse-value (summus = 0) [0] [-1 * diff2mus / summus]

  let lambda (w ^ 2) / ((exp 1) - (exp (1 - (1 / (w ^ 2)))))
  let diff2mus2 sum map [[y] -> (1 - item y mus) * (item y diffs2)] range
    length ops; sum((x-xi)^2 (1-mu(x,xi)))
  let summus2 (length ops) - summus ; sum(1-mu(x,xi))
  let dinter lambda * ifelse-value (summus2 = 0) [0] [diff2mus2 / summus2]

  let gamma (w ^ 2) / exp(1)
  let sats map [[xi] -> saturation x xi] ops
  let sumsats sum sats
  let dindiv (1 - a) * b * (w ^ 2) / (exp 1) * ifelse-value (summus = 0)
    [0] [-1 * sumsats / summus]

  let P (a * dinter) + ((1 - a) * (1 - b) * dintra) + ((1 - a) * b *
    dindiv)
  report P
end

```

```

to-report D-deriv [ x ops ]
  let a outgroup-aversion
  let w group-width
  let v repulse-range * w

  let mus map [[xi] -> membership x xi] ops
  let summus sum mus ; sum(mu(x,xi))
  let diffs map [[xi] -> x - xi] ops
  let diffmus sum map [[y] -> (item y mus) * (item y diffs)] range length
    ops ; sum((x-xi)mu(x,xi))
  let diffs2 map [[y] -> (item y diffs) ^ 2 ] range length ops
  let diff2mus sum map [[y] -> (item y mus) * (item y diffs2)] range
    length ops; sum((x-xi)^2 mu(x,xi))
  let diffs3 map [[y] -> (item y diffs) ^ 3 ] range length ops
  let diff3mus sum map [[y] -> (item y mus) * (item y diffs3)] range
    length ops; sum((x-xi)^3 mu(x,xi))
  let ddintradx ifelse-value ((w * summus) ^ 2 = 0) [0] [(-2 * (((((w ^ 2)
    * diffmus) - diff3mus) / ((w ^ 2) * summus)) + ((diff2mus * diffmus)
    / ((w * summus) ^ 2)))))]

  let summus2 (length ops) - summus ; sum(1-mu(x,xi)) = n - sum(mu(x,xi))
  let diffmus2 sum map [[y] -> (1 - item y mus) * (item y diffs)] range
    length ops ; sum((x-xi)(1-mu(x,xi)))
  let diff2mus2 sum map [[y] -> (1 - item y mus) * (item y diffs2)] range
    length ops; sum((x-xi)^2 (1-mu(x,xi)))
  let lambda (w ^ 2) / ((exp 1) - (exp (1 - (1 / (w ^ 2))))))
  let ddinterdx lambda * ifelse-value ((w * summus2) ^ 2 = 0) [0] [2 *
    (((((w ^ 2) * diffmus2) + diff3mus) / ((w ^ 2) * summus2)) -
    ((diff2mus2 * diffmus) / ((w * summus2) ^ 2)))]

  let sats map [[xi] -> (saturation x xi)] ops
  let sumsats sum sats
  let diffstats sum map [[y] -> (item y sats) * (item y diffs)] range
    length ops ; sum ((x-xi)mu*(x,xi))
  let ddindivdx (w ^ 2) / exp(1) * ifelse-value ((w * summus) ^ 2 = 0) [0]
    [-2 * ((diffmus * sumsats / ((w ^ 2) * (summus ^ 2))) - (diffstats /
    ((v ^ 2) * summus)))] ; THIS WORKS

  let dDdx (a * ddinterdx) + ((1 - a) * (1 - b) * ddintradx) + ((1 - a) *
    b * ddindivdx)
  report dDdx
end

to-report influence-MIF [ x ops ] ; converts P-deriv to actual influence
  value
  let influence (D-deriv x ops)

```

```

    set influence k * influence
    report influence
end

to-report skewness [ xlist ] ; reports skewness of a distribution
  let len length xlist
  let xbar mean xlist
  let skew (sum map [ [x] -> (x - xbar) ^ 3 ] xlist) / ( len * (((len - 1)
    / len) * variance xlist) ^ 1.5)
  report skew
end

to-report kurtosis [ xlist ] ; reports kurtosis of a distribution
  let len length xlist
  let xbar mean xlist
  let kurt (sum map [ [x] -> (x - xbar) ^ 4 ] xlist) / ( len * (((len - 1)
    / len) * variance xlist) ^ 2)
  report kurt
end

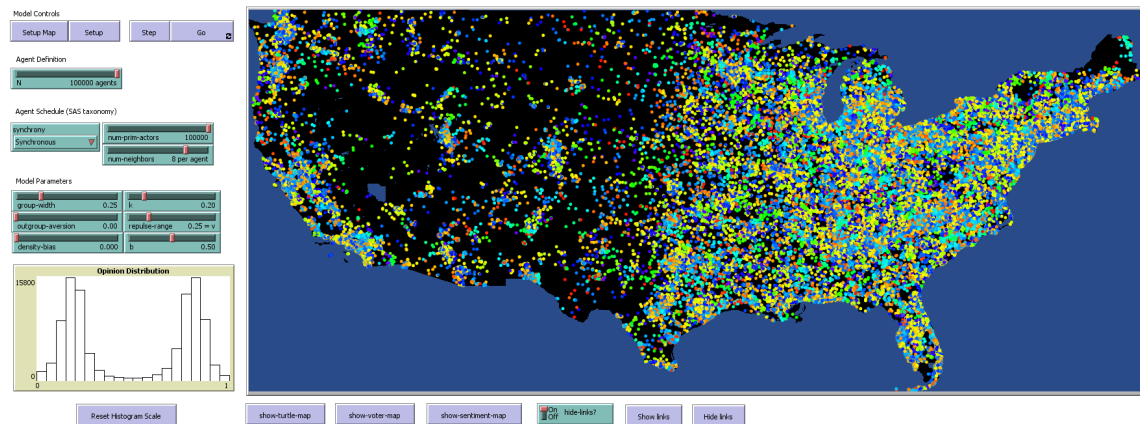
;; TURTLE FUNCTIONS ;;

to-report get-ops ; gets appropriate opinions, based on setting of SWN?
  let ops []
  ifelse SWN? [ set ops [opinion] of (turtle-set link-neighbors self) ] [
    set ops [opinion] of turtles ]
  report ops
end

to-report get-influence [ ops ] ; gets appropriate influence value. Added
  to increase modularity.
  let influence (influence-MIF opinion ops)
  report influence
end

```

Appendix W. NetLogo Code: Meta-contrast Influence Field - Local Repulsion Model on U.S. Map



```

extensions [ rnd ]

globals [ clusters cluster-ops land-patches ]

patches-own [ density influence-density ]

turtles-own [opinion next-opinion]

;;;;; SETUP ;;;;;

to setup-map
  clear-all
  file-open "uspopdensity.txt"
  foreach range world-height [ [y] ->
    foreach range world-width [ [x] ->
      ask patch x (max-pycor - y) [
        set density file-read
        if density = 0 [ set pcolor blue - 1 ]
        ;set pcolor ifelse-value (density < 1) [ 0 ] [ ln density ] ; this
        shows a density map of U.S.
      ]
    ]
  ]
  file-close
  set land-patches patches with [density > 0]
  let mean-density mean [density] of land-patches
  ask land-patches [ set influence-density (1 - exp (- density /
    mean-density)) ]
  let mean-inf mean [influence-density] of land-patches

```

```

ask land-patches [ set influence-density influence-density - mean-inf ]
; causes mean influence to be zero
end

to setup
clear-turtles
setup-turtles
setup-network
reset-ticks
end

to setup-turtles
set-default-shape turtles "circle"
let sprouters rnd:weighted-n-of-with-repeats N patches [ density ]
foreach sprouters [ [here] ->
ask here [ sprout 1 [
set size 5
jitter-pos
initialize-turtle
] ]
]
end

to setup-network
ask turtles [
let my-neighbors other turtles in-radius 5
if count my-neighbors < num-neighbors [
set my-neighbors (turtle-set my-neighbors min-n-of (num-neighbors -
count my-neighbors) (other turtles with [not member? self
my-neighbors]) [distance myself])
]
if count my-neighbors < num-neighbors [ show count my-neighbors ]
create-links-with my-neighbors [
set hidden? hide-links?
]
]
end

to initialize-turtle
set opinion random-float 1
set next-opinion opinion
set color hsb (260 * opinion + 5) 100 100
end

;;;;; STEP ;;;;;;

```



```

to go
  let prim-actors ifelse-value (num-prim-actors >= N) [ turtles ] [ n-of
    num-prim-actors turtles ]
  ask prim-actors [
    be-influenced
  ]
  if synchrony = "Synchronous" [ ask prim-actors [ update-opinion ] ]
  tick
  if count turtles with [opinion = 0 or opinion = 1] = N [ stop ]
; if max [opinion] of turtles - min [opinion] of turtles < 0.01 [ stop ]
; if max [abs (get-influence get-ops)] of turtles <= (group-width / 100) [
  stop ]
end

;;;;; PROCEDURES ;;;;;

; OBSERVER PROCEDURES

to show-turtle-map
  ask land-patches [ set pcolor black ]
  ask turtles [ show-turtle ]
end

to show-sentiment-map
  ask turtles [ hide-turtle ]
  ask land-patches [
    set pcolor black
    let nearby-ops [opinion] of turtles in-radius 5
    if length nearby-ops > 0 [
      set pcolor hsb (260 * mean nearby-ops + 5) 100 100
    ]
  ]
  spread-color
end

to show-voter-map
  ask turtles [ hide-turtle ]
  ask land-patches [
    set pcolor black
    let nearby-ops [opinion] of turtles in-radius 5
    if length nearby-ops > 0 [
      ifelse mean nearby-ops < 0.5 [
        set pcolor red
      ] [
        set pcolor blue
      ]
    ]
  ]

```

```

    ]
  ]
]
spread-color
end

to spread-color
  let done? false
  while [not done?] [
    set done? true
    ask land-patches with [pcolor != black] [
      ask neighbors4 with [pcolor = black] [
        set pcolor [pcolor] of myself
        set done? false
      ]
    ]
  ]
end

; TURTLE PROCEDURES

to jitter-pos
  setxy (pxcor - 0.5 + random-float 1) (pycor - 0.5 + random-float 1)
end

to update-opinion
  set opinion next-opinion
  if opinion > 1 [ set opinion 1 ]
  if opinion < 0 [ set opinion 0 ]
  set color hsb (260 * opinion + 5) 100 100
end

to be-influenced
  let ops get-ops
  let influence get-influence ops
  set next-opinion next-opinion + influence
  if synchrony = "Asynchronous" [ update-opinion ]
end

;;;;; FUNCTIONS ;;;;;;

;; OBSERVER FUNCTIONS ;;

to-report membership [x xi] ; mu(x,x_i) in Salzarulo (2006)
  ;; Note: this assumes agents are unaware of group membership (may not be
  appropriate in all cases)

```

```

let w group-width
report exp (- ((x - xi) ^ 2 / (w ^ 2)))
end

to-report saturation [x xi] ; same as membership but different value for w
let v repulse-range * group-width
report exp (- ((x - xi) ^ 2 / (v ^ 2)))
end

to-report desirability [ x ops ]
let a outgroup-aversion
let w group-width
let v repulse-range * w

let mus map [[xi] -> membership x xi] ops

let diffs2 map [[y] -> (x - item y ops) ^ 2 ] range length ops
let summus sum mus ; sum(mu(x,xi))
let diff2mus sum map [[y] -> (item y mus) * (item y diffs2)] range
length ops; sum((x-xi)^2 mu(x,xi))
let dintra ifelse-value (summus = 0) [0] [-1 * diff2mus / summus]

let lambda (w ^ 2) / ((exp 1) - (exp (1 - (1 / (w ^ 2)))))
let diff2mus2 sum map [[y] -> (1 - item y mus) * (item y diffs2)] range
length ops; sum((x-xi)^2 (1-mu(x,xi)))
let summus2 (length ops) - summus ; sum(1-mu(x,xi))
let dinter lambda * ifelse-value (summus2 = 0) [0] [diff2mus2 / summus2]

let gamma (w ^ 2) / exp(1)
let sats map [[xi] -> saturation x xi] ops
let sumsats sum sats
let dindiv (1 - a) * b * (w ^ 2) / (exp 1) * ifelse-value (summus = 0)
[0] [-1 * sumsats / summus]

let P (a * dinter) + ((1 - a) * dintra) + dindiv
report P
end

to-report P-deriv [ x ops ]
let a outgroup-aversion
let w group-width
let v repulse-range * w

let mus map [[xi] -> membership x xi] ops
let summus sum mus ; sum(mu(x,xi))
let diffs map [[xi] -> x - xi] ops

```

```

let diffmus sum map [[y] -> (item y mus) * (item y diffs)] range length
  ops ; sum((x-xi)mu(x,xi))
let diffs2 map [[y] -> (item y diffs) ^ 2 ] range length ops
let diff2mus sum map [[y] -> (item y mus) * (item y diffs2)] range
  length ops ; sum((x-xi)^2 mu(x,xi))
let diffs3 map [[y] -> (item y diffs) ^ 3 ] range length ops
let diff3mus sum map [[y] -> (item y mus) * (item y diffs3)] range
  length ops ; sum((x-xi)^3 mu(x,xi))
let ddintradx ifelse-value ((w * summus) ^ 2 = 0) [0] [(-2 * (((((w ^ 2)
  * diffmus) - diff3mus) / ((w ^ 2) * summus)) + ((diff2mus * diffmus)
  / ((w * summus) ^ 2)))]

let summus2 (length ops) - summus ; sum(1-mu(x,xi)) = n - sum(mu(x,xi))
let diffmus2 sum map [[y] -> (1 - item y mus) * (item y diffs)] range
  length ops ; sum((x-xi)(1-mu(x,xi)))
let diff2mus2 sum map [[y] -> (1 - item y mus) * (item y diffs2)] range
  length ops ; sum((x-xi)^2 (1-mu(x,xi)))
let lambda (w ^ 2) / ((exp 1) - (exp (1 - (1 / (w ^ 2))))))
let ddinterdx lambda * ifelse-value ((w * summus2) ^ 2 = 0) [0] [2 *
  (((((w ^ 2) * diffmus2) + diff3mus) / ((w ^ 2) * summus2)) -
  ((diff2mus2 * diffmus) / ((w * summus2) ^ 2)))]

let sats map [[xi] -> (saturation x xi)] ops
let sumsats sum sats
let diffstats sum map [[y] -> (item y sats) * (item y diffs)] range
  length ops ; sum ((x-xi)mu*(x,xi))
let ddindivdx (w ^ 2) / exp(1) * ifelse-value ((w * summus) ^ 2 = 0) [0]
  [-2 * ((diffmus * sumsats / ((w ^ 2) * (summus ^ 2))) - (diffstats /
  ((v ^ 2) * summus)))] ; THIS WORKS

let dPdx (a * (1 - b) * ddinterdx) + ((1 - a) * ddintradx) + ((1 - a) *
  b * ddindivdx)
report dPdx
end

to-report influence-MIF [ x ops ] ; converts P-deriv to actual influence
  value
  let influence (P-deriv x ops)
  set influence k * influence
  report influence
end

to-report skewness [ xlist ] ; reports skewness of a distribution
  let len length xlist
  let xbar mean xlist
  let skew (sum map [ [x] -> (x - xbar) ^ 3 ] xlist) / ( len * ((len - 1)

```

```

    / len) * variance xlist) ^ 1.5)
  report skew
end

to-report kurtosis [ xlist ] ; reports kurtosis of a distribution
  let len length xlist
  let xbar mean xlist
  let kurt (sum map [ [x] -> (x - xbar) ^ 4 ] xlist) / ( len * ((len - 1)
    / len) * variance xlist) ^ 2)
  report kurt
end

;; TURTLE FUNCTIONS ;;

to-report get-ops ; gets appropriate opinions, based on setting of SWN?
  let ops []
  set ops [opinion] of (turtle-set link-neighbors self)
  report ops
end

to-report get-influence [ ops ] ; gets appropriate influence value. Added
  to increase modularity.
  let influence (influence-MIF opinion ops) + (k * density-bias *
    influence-density)
  report influence
end

```

Vita

Maj Chris Weimer graduated from Niceville High School in Niceville, Florida in 2002. He graduated from the United States Air Force Academy with a Bachelor of Science in Behavioral Science and was commissioned in May 2006.

Maj Weimer served his first assignment as a Research Scientist for Adversarial Modeling for the 711th Human Performance Wing, Cognitive Systems Branch. From August 2010 to March 2012, he attended the Graduate School of Engineering and Management, Air Force Institute of Technology, at Wright-Patterson AFB, Ohio. He graduated in March 2012 with a Masters of Science degree in Operations Research. He was then assigned to Headquarters, Air Force Operational Test and Evaluation Center, first as Senior Test Analyst and subsequently as the Chief, Tools and Methods Branch. In August 2015 he returned to the Air Force Institute of Technology, Wright-Patterson AFB, Ohio. After completion of his PhD in Operations Research, he will be assigned to the Air Force Installation and Mission Support Center, Lackland AFB, Texas.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE (DD-MM-YYYY) 13-09-2018		2. REPORT TYPE Doctoral Dissertation		3. DATES COVERED (From — To) Oct 2015 — Sep 2018	
4. TITLE AND SUBTITLE Generating Strong Diversity of Opinions: Agent Models of Continuous Opinion Dynamics				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
6. AUTHOR(S) Weimer, Christopher W., Maj, USAF				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765				8. PERFORMING ORGANIZATION REPORT NUMBER AFIT-ENS-DS-18-S-044	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) 711th Human Performance Wing Attn: Robert Thomas 4141 Petroleum Road, Bldg 3260 JBSA Fort Sam Houston, TX 78234-2644 (210) 539-8150 (DSN 389); Email: robert.thomas.47@us.af.mil				10. SPONSOR/MONITOR'S ACRONYM(S) 711 HPW/RHDO	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION / AVAILABILITY STATEMENT DISTRIBUTION STATEMENT A: APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.					
13. SUPPLEMENTARY NOTES This work is declared a work of the U.S. Government and is not subject to copyright protection in the United States.					
14. ABSTRACT Opinion dynamics is the study of how opinions in a group of individuals change over time. A goal of opinion dynamics modelers has long been to find a social science-based model that generates strong diversity — smooth, stable, possibly multi-modal distributions of opinions. This research lays the foundations for and develops such a model. First, a taxonomy is developed to precisely describe agent schedules in an opinion dynamics model. The importance of scheduling is shown with applications to generalized forms of two models. Next, the meta-contrast influence field (MIF) model is defined. It is rooted in self-categorization theory and improves on the existing meta-contrast model by providing a properly scaled, continuous influence basis. Finally, the MIF-Local Repulsion (MIF-LR) model is developed and presented. This augments the MIF model with a formulation of uniqueness theory. The MIF-LR model generates strong diversity. An application of the model shows that partisan polarization can be explained by increased non-local social ties enabled by communications technology.					
15. SUBJECT TERMS Opinion dynamics, self-categorization theory, uniqueness theory, agent-based modeling, psychosocial simulation					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. J. O. Miller, AFIT/ENS
U	U	U	U	266	19b. TELEPHONE NUMBER (include area code) (937) 255-3636, x4326; john.miller@afit.edu