

1 Model Description

1.1 Model Overview

In this section, we present a model of policy formation based on simulated annealing. Conceptually, a bill is sponsored by a single legislator, reviewed and revised (by simulated annealing) by a network of the sponsor’s friends, then referred to a committee of interested members who are allowed to alter the legislation through an additional round of annealing — reflecting committee agenda-setting powers (**cm93**; **cm05**) — and finally voted on by the entire legislative body.

Legislators are initialized with priorities and positions on a common list of issues. Model parameters determine the extent to which legislator priorities and positions conform to ideological party agenda. Legislators are connected to each other in a social network, which is generated using both homophily (**msc01**; **br11**) and preferential attachment (**Barabasi1999**).

We implement the model in Python, with an object-oriented design. There are three high-level classes that interact in the model: **State**, **Legislator**, and **Bill**. We describe these further in sections below. The fixed and free parameters of the model are listed and described in the Technical Appendix.

1.2 Model Initialization

The three main functions accomplished during model initialization are setting up the model environment with default issue priorities and positions via the **State** object, generating legislators with heterogeneous priorities and positions, and generating a social network among the legislators.

Initializing the Model Environment (the State object) The **State** object generates seed vectors of priorities and positions to be used during the stochastic generation of individual legislator preferences. Issue priorities are seeded with initial values according to the model parameters **Ideology_Issues** and **State_Priorities**. For our model, we assume that several core issues represent powerful, crystallizing factors that differentiate our simulated parties. Thus, for party affiliations, individual party ”platforms” consist of vectors of positions and priorities on a set of issues that includes both **State_Priorities** and a random sample of high-priority **Ideology_Issues**.

Legislator Initialization Each legislator’s issue priorities are assigned with stochastic preferential attachment method to the seed values provided by the **State** object. (See the Technical Appendix for details.) This generates a power law distribution of priorities, so that a legislator places very high priority on a small number of issues but low priority on most issues; we consider this a reasonable assumption. It also results in legislators having some correlation in priorities to the extent that their seed priorities are the same, but heterogeneity is stochastically introduced.

The final component of preference generation assumes that affiliated legislators adopt the positions of their party. For all other issues (and all issues for unaffiliated legislators), positions are assigned randomly from the range $[0, 2^{\text{Solution_Bit_Length}} - 1]$, inclusive (*i.e.*, a random bit string of a predefined length, set to the variable `solution.bit_length`).

The end-result of this process is a set of legislator agents with unique policy preferences. These policy preferences are conditioned by party ideologies and state priorities. Thus there is strong issue-position and -priority correlation between party-affiliated legislators but some heterogeneity, and some affiliation among unaffiliated members as well.

Network Generation Having generated a population of legislator agents that have heterogeneous preferences on some issues and shared preferences on others, we complete model initialization by generating a social network among legislators using those preferences. This network is designed to capture the social dynamics of legislating, in that a Congressmen will naturally have a set of friends and close colleagues that he interacts with more than others.

Drawing on legislator positions and priorities, we generate a social network using preferential-attachment and homophily mechanisms.

A legislator’s potential friends are defined as the set of legislator agents who have a minimum similarity of *friend-threshold* total homophily on all issues with the legislator. Total similarity is the issue priority-weighted and -normalized sum of homophilies on issue positions. Homophily (similarity of position between two legislators) in turn relies on a modified Hamming distance calculation between the two `issue` bit strings encoding their respective positions, wherein the most significant bits of a position carry higher weight than lower significant bits (*i.e.*, in increasing powers of $\frac{1}{2}$), resulting in a bi-modal (rather than uniform) distribution of all legislator homophilies on any given issue. Edge assignment is as described in Barabasi and Albert (1999), with target nodes selected randomly from *pdf* of degree distribution in the sub-network of potential friends. The typical outcome of this procedure is a network among legislator agents with *small-world* properties, consistent with existing research on social networks in Congress (**Granovetter1978**).

1.3 Simulation Algorithm Overview

Having defined a population of legislators and their relationship to each-other, we next establish a procedure for legislators to engage in the business of law-making.¹ In our model of law-making, the simulation sequentially repeats the following process for 200 proposals (or halts if all issues are passed into law):

1. Proposal:
 - (a) A random legislator is chosen to sponsor a bill.
 - (b) The sponsor proposes a draft bill on any issue that has not already been addressed by law.

¹ One might argue that this is a departure from realism, as the current Congress does not appear to do this. However, we are attempting to generate a broadly generalizable model of law-making in legislatures. Some legislatures do legislate, periodically.

2. Draft circulation among cosponsors:
 - (a) All legislators connected to the sponsor in the social network are selected as cosponsors.
 - (b) The cosponsors revise the draft using simulated annealing; new issues may be added to the bill during the revision process and solutions on existing issues may change.
3. Committee review:
 - (a) The draft is referred to a committee; committees are chosen according to legislators for whom the main issue of the bill is a high priority.
 - (b) The committee revises the bill by simulated annealing; again, new issues may be added and existing solutions changed as a result.
4. Floor vote:
 - (a) The bill is referred to the floor for a vote.
 - (b) A legislator votes ‘yes’ to a bill when her satisfaction with it is greater than the model parameter `satisfaction_threshold`.
 - (c) If the bill passes by simple majority (>50% votes), the bill is made into law; *i.e.*, the solutions addressed by the bill are recorded and the issues may not be revisited for the remainder of the realization.

Simulated Annealing The simulated annealer implements the Metropolis algorithm for simulated annealing (`mrrt53`; `kgv`): a state with lower energy than the current state is accepted, while a state with higher energy is accepted with probability $P(state) = e^{\Delta E/kT}$.

Our energy function is the cumulative dissatisfaction of all reviewers over all dimensions of the bill, where each reviewer’s dissatisfaction is a priority-weighted and -normalized sum of the modified Hamming distances between the reviewer’s preferred solutions and the current solutions proposed by the bill. For our model, we selected k such that, at a temperature of 1.0, an increase of 0.1 in dissatisfaction is accepted with probability $\frac{1}{2}$. The temperature annealing schedule is configured linearly.

1.4 Model Verification, Validation, and Calibration

We verified the model through code review and incremental testing. We tested sub-units of functionality by verifying expected intermediate outputs before implementing more complex functionality. We made little effort to validate outputs, as this is an exploratory model. Where applicable, we have stated our assumptions and validated them against either literature or reasonable expectations. For example, we cap the maximum number of committees a legislator agent may serve on by an approximation of the actual limitation imposed on real-world Congressmen.

Similarly, we calibrate legislators’ *satisfaction thresholds*—the point at which they agree with legislation—to a level that generates a pass-rate of approximate 4%, which is comparable to passage rates in the actual US Congress, which have varied between 2% and 7% in recent history (see Figure ??).²

² Pass rates are equal to the total number of bills passed in a given Congress divided by the total counts of introduced legislation for that Congress. Other metrics of legislative productivity as a proportion of items considered generate similar results. Data to calculate pass rates was collected from Civic Impulse LLC (<http://www.govtrack.us>).