# Appendix: The Elections and Governments Program and Its Implementation

**Part A. The Program**

The Elections and Governments Simulations Program was designed by Paul Warwick and written on C# using Microsoft Visual Studio.NET (2008) by Professor Irina Zakharova of the Department of Software at Tyumen State University (Tyumen, Russia). The use of this software is based on a license obtained through the cooperation program between the MSDN Academic Alliance with the Institute of Mathematics and Computer Science, Tyumen State University. The program is intended for academic use only.

The program generates any specified number of political systems and, for each, conducts both an election and a government formation process. It includes a great many options to structure these processes. In what follows, only those options used in the paper are described; a full description of the program’s capabilities is available on request. The program itself is also available on request.

**Political System Generation**

The simulation sets presented in the paper each consist of 5000 independent political systems. Each political system consists of voters and parties located in a policy space of two dimensions of range [0, 1]. The numbers and positions of both are determined in the following ways:

***Voters***

1. **Number.** The number of voters is set at 10,000.
2. **Positions.**  The positions of voters on both dimensions were drawn from either uniform or truncated normal distributions. Positions on each dimension are drawn independently and hence are uncorrelated with one another. All distributions have range [0,1].

For truncated normal distributions, the means were drawn from uniform distributions with range [.45, .55] to allow for scenarios in which one side of the dimension center point (0.5) is more heavily populated than the other. This is done to enhance realism: there is no reason to expect that ‘left’ and ‘right’, however they are defined in a given political system, would be populated by equal numbers of entities. The standard deviations for these distributions have a mean value of 0.26, but they, too, vary across runs according to a standard deviation of 0.04. These values are reasonably consonant with real data, as shown in the paper. An error term drawn from a normal distribution with a mean of 0 and standard deviation of 0.03 is added to reach the final standard deviation for each distribution.

***Parties***

**Number.** The number of parties, *J*, in a political system ranges between 3 and 7 (*3 ≤ J ≤ 7*). In the simulations, 1000 governments were drawn for each value of *J*.

**Positions.**

In each system, party positions were drawn from the same distributions used for voters.

**The Election Stage**

With all positions specified for a given political system, the program moves to the election stage. A set of 10 elections is conducted for each system. All input parameters stay constant except for the voter positions and the error term in each utility function. The average results for each set of 10 elections is reported as a single election result.

Each of these elections involves calculating each voter’s utility for each party and casting votes accordingly. The program offers two basic models of voter utility, a standard model, based on policy distance and possibly valence, and a compensational voting model (Kedar 2009).

1. ***The Standard Model.*** In the standard model, voter utility is a function of policy distance, party valence, and random error.

**Policy distance**. The distance between the policy positions of voter *i* and party *j* in a given political system is measured in a city-block metric with the mixing parameter *δ* denoting the relative importance of the second dimension in the political system:

 (1)

In the simulations for Table 2, δ is set at .33 to give the first dimension twice the weight of the second; elsewhere it is set to 0.5 (SS 5 through SS 7) or drawn randomly from a uniform distribution with range [0,1].

**Valence**. In some simulations, voters in each system attribute valence assessments, *Vj*, to the various parties in their system. These assessments are drawn from either a uniform or a normal distribution, both with range [0, 1]. Following evidence from the CSES data (noted in the paper), the standard deviation in party valence scores is set to 0.1 on the 0-1 scale used here, with a standard deviation across political systems of 0.03.

For both operationalizations of valence, the parties’ total raw valences are transformed into proportions and multiplied by the product of the number of parties and the mean voter-party distance in the system. The first element in the product sets the average party valence in each system to 1; the second then rescales that value so that the average valence in each system equals its average voter-party distance. This is done to balance out the relative sizes of the valence and policy distance components in voter utility calculations. The relative weight of the two components can then be manipulated by means of the mixing parameter *βval*. In the paper, *βval* was drawn from a uniform distribution with range [0, 1].

Valence may not be constant across voters; instead, it may diminish with distance (according to the extent that distance matters). Hence the valence portion of utility in a political system is given by:

,

where *ω* is the rate at which valence declines with policy distance and *ej* is an error term. The value of *ω* for each political system was drawn from a uniform distribution with range [0,1]. The random error was drawn from a normal distribution with mean 0 and standard deviation 0.03.

**The utility function.** The total utility of Voter *i* for Party *j* is:



where *eij* = random error, selected from a normal distribution with mean 0 and standard deviation 0.03.

***The Compensational Voting Model*.** In this model, voter utility includes a distance component, compensational component, and random error; there is no valence component.

**Policy distance.** As in the standard model.

**Compensationalism.** In the compensational component, voters estimate the utility of voting for each party by calculating how much their utility would be improved if that party played its part in influencing government policy rather than not doing so; i.e., the utility of party *j* for voter *i* is defined as:

,

where *Diz* is the distance from *i* to expected government policy *z* and Di.*z(-j)*= distance from *i* to counterfactual policy *z(-j)* (government policy if party *j* is excluded from policy influence). Since we have no expectations about which parties will form the government, *z* on each dimension is assumed to be the seat-weighted mean position of all legislative parties on that dimension and *z(-j)* is assumed to be the seat-weighted mean position of all parties except *j*.

* 1. **The utility function**. The utility of Voter *i* for Party *j* is:



where:

*Diz* = distance from *xi* to expected government policy

Di.*z(-j)*= distance from *xi* to counterfactual policy (government policy if party *j* is excluded)

*θ* = weight given to proximity voting relative to compensational voting, drawn from a uniform distribution with range [0,1]

*eij* = random error selected from a normal distribution with mean 0 and standard deviation 0.03.

**The Government formation stage**

The government formation model is an adaptation of the Golder, Golder and Siegel (2012) ‘zero-intelligence’ model of government formation.

1. After an election, the status quo government position is set at (.5, .5) and portfolio shares are set at 0 for all parties.
2. Each party simultaneously proposes a government, always including itself. Each proposal consists of a distribution of portfolios and a policy position. Subject to the constraints listed below, all proposals are equally likely.

There are two sets of constraints that govern proposals:

* + 1. **Zero-intelligence**. To be permissible, a proposal must strictly beat the status quo for all its potential members *and* for a legislative majority.

**Proportional Allocation Weighted Mean (PAWM).** To be permissible, a proposal must (a) satisfy 2.1.1, (b) adopt a policy position on each dimension that is the seat-weighted mean position of its proposed members on that dimension, and (c) adopt a portfolio allocation that is proportional to the seat shares of its proposed members.

Each party then compares the utility associated with each of the proposed governments with that associated with the status quo.

**Party utility** is calculated in the following way:

Policy distance between party *j* and government *z* is calculated as:

,

where *xj.* is party *j*’s position on a given dimension and *xz.* is government *z*’s position on a given dimension. The *α* parameter determines the significance of the second dimension. It was either set to a fixed value, such as .33 (SS1 through SS 4) or 0.5 (SS 5 through SS 7), or drawn from a uniform distribution with range [0,1]. In the latter case, the saliences for both voters and parties in any given system are constrained to be equal.

* + 1. Party *j*’s utility for government *z* in a given political system is a linear combination of its portfolio shares in *z* and the policy costs that result from the policy divergence between its position and that of *z*. Thus, party *j*’s utility for *z* is:

,

where:

*csjz* = party *j*’s proportion of portfolios in the government *z* (‘0’ when not in government)

*Djz* = the policy distance between party *j* and government z

*Βport* = the weight placed on policy concerns relative to cabinet shares, drawn from a uniform distribution with range [0,1].

1. Any permissible proposal (as defined in 2) draws the attention of the head of state. If only one proposal draws the head of state’s attention, it is installed as the next government; if more than one does, the head of state chooses one at random and installs it as the next government; if none does, all proposals are discarded and the process re-starts at step 2 above.
2. The process continues until a new government is installed or 100 periods pass (in which case, the status quo is re-installed).
3. A run consists of an election followed by 2500 government formations. Only the results from the 2500th formation are used in analyses (this represents a ‘burn-in’ period long enough to get to the limiting distribution).

**Part B. The Implementation**

A set of simulations is launched by selecting from a wide variety of options on the program’s graphic user interface and then selecting Election > Start. Apart from the various specifications indicated above, the 12 simulation sets presented in the article followed the specifications listed in Table A.1.

[Table A.1 about here]

**Table A.1. Program Specifications Selected for the Simulation Sets.**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Simulation Set | | | | | | | | | | | |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| Voter/Party Distributions | N | N | U | U | N | N | N | N | N | N | N | N |
| *βport* (1 = port. only) | U | U | 1 | U | U | U | U | U | U | U | U | U |
| Dimensional saliences (δ and α) | .33 | .33 | 1\* | .5 | .5 | U | U | U | U | U | U | U |
| Valence distribution | – | – | – | – | – | – | – | U | N | N | – | – |
| *βval* | – | – | – | – | – | – | – | U | U | U | – | – |
| *ω* | – | – | – | – | – | – | – | 0 | 0 | U | – | – |
| *θ* | – | – | – | – | – | – | – | – | – | – | 0 | U |
| Party system weighting | Off | Off | Off | Off | Off | Off | Off, then On | Off | Off | Off | Off | Off |
| Voting Model | Std. | Std. | Std. | Std. | Std. | Std. | Std. | Std. | Std. | Std. | Comp. | Comp. + Std. |
| Government Model | 0-int. | PAWM | 0-int. | 0-int. | 0-int. | 0-int. | PAWM | PAWM | PAWM | PAWM | PAWM | PAWM |

*Note:* U = uniform distribution with range [0, 1]; N = normal or (truncated) normal distribution with parameters as specified above.

\* Voters decide based only on the second dimension, so choices are random with respect to first dimension.