A Tournament of Party Decision Rules

James H. Fowler and Michael Laver

Journal of Conflict Resolution 2008; 52; 68
DOI: 10.1177/0022002707308598

Journal of Conflict Resolution
Volume 52 Number 1
February 2008 68-92
© Sage Publications
10.1177/0022002707308598
http://jcr.sagepub.com
hosted at
http://online.sagepub.com

A Tournament of Party Decision Rules

James H. Fowler
Department of Political Science
University of California, San Diego
Michael Laver
Department of Politics
New York University

Following Axelrod's tournaments for strategies in the repeat-play prisoner's dilemma, we ran a "tournament of party decision rules" in a dynamic agent-based model of party competition. We asked researchers to submit rules for selecting party positions in a two-dimensional policy space, pitting each rule against all others in a suite of long-running simulations. The most successful rule combined a number of striking features: satisficing rather than maximizing in the short run, being "parasitic" on choices made by successful rules, and being hardwired not to attack other agents using the same rule. In a second suite of simulations in a more evolutionary setting in which the selection probability of a rule was a function of the previous success of agents using the same rule, the rule winning the original tournament pulled even further ahead of the competition.

Keywords: *agent-based model; computer tournament; party competition; parties and elections*

1. Introduction

Consider a dynamic system of multiparty competition in a multidimensional policy space. Party leaders compete for votes at time t by selecting a party policy position intended to appeal to as many voters as possible, given an observed history of the system up to the previous point in time, $t-1$. The decision problem for a leader is to select an optimal party position, conditional on all available information. This problem is intractable analytically for the general case, which implies, as

Authors' Note: We thank Robert Axelrod and Scott deMarchi for comments on the design of this tournament as well as more than twenty-four professional colleagues for considerable time and effort spent designing tournament entries. Drafts of this article were presented at the 2006 annual meeting of the American Political Science Association and at departmental seminars in Columbia University and New York University. Comments from seminar participants were very helpful. The authors will be conducting a second tournament in 2008. To enter, please visit http://jhfowler.ucsd.edu. An online appendix is available at http://jcr.sagepub.com/supplemental.

a behavioral assumption, that leaders use decision heuristics rather than provable best-response strategies when choosing party positions.[1] One very effective way to investigate the deployment of different decision heuristics in multidimensional multiparty competition (MDMPC) is to build agent based models (ABMs) of party competition and interrogate these systematically (De Marchi 1999, 2003; Fowler and Smirnov 2007, 2005; Kollman, Miller, and Page 2003a, 2003b; Kollman, Miller, and Page 1992, 1998; Laver 2005; Laver and Schilperoord 2007; Smirnov and Fowler 2007). However, existing ABMs of MDMPC explore interactions between a limited set of analyst-specified decision rules; in no sense has the full set of *potential* rules been investigated.

Responding to the similar problem of investigating potential decision rules for the more tractable setting of the iterated prisoner's dilemma (PD), Robert Axelrod ran a famous series of computer tournaments (Axelrod 1980a, 1980b, 1997). Scholars were asked to submit strategies, which were pitted against each other in a series of computer simulations. The first tournament (Axelrod 1980a) had fourteen entries and the winner was *Tit-for-Tat*, submitted by Anatol Rapoport. In a second tournament, new entries were invited in light of the results of the first (Axelrod 1980b); there were sixty-two entries and *Tit-for-Tat* won again. In these two tournaments, all decision rules investigated were predefined and immutable. In a more recent tournament, Axelrod explored the *evolution of new strategies* for playing the iterated PD. He started with a set of random rules, as opposed to a set predefined by others, and applied the standard genetic operators of crossover and mutation to these over successive iterations, rewarding successful decision rules with higher fitness scores and hence higher reproduction probabilities (Axelrod 1997). Rules resembling *Tit-for-Tat* often emerged from this evolutionary process. However, completely new types of successful rules also evolved that beat *Tit-for-Tat*. These were strategies no game theorist had submitted to earlier tournaments, and all involved at least one initial defection, serving to probe an opponent's "type" in a more informative way than initial cooperation.

We adapted this fertile "computer tournament" design to the problem of evaluating potential decision rules for choosing party positions in dynamic MDMPC. We took a model of MDMPC that implements the canonical static spatial model of party competition as a dynamic ABM (Laver 2005), programmed a version of this as a tournament "test-bed," and published the test-bed code (in R). We invited submissions of position-selection rules, advertising our tournament widely and offering a $1,000 prize for the rule most successful in winning votes in very long-running simulations. The four rules investigated by Laver (2005) were declared preentered in the tournament but ineligible to win the prize. A full description of the tournament and copy of the test-bed code can be found in Appendix A (online).

We describe the simulation test-bed, together with the decision rules preentered in the tournament, in sections 2 and 3. In section 4, we describe and discuss the submissions we received. In section 5, we report and discuss the results of the tournament and describe the results of extending the tournament into a much more evolutionary setting, where the probability that a decision rule will reproduce is a function of the past success of agents using the same rule. Finally, we conclude and lay out a program for future work.

2. A Tournament Test-Bed for Policy-Selection Rules

Baseline ABMs of MDMPC

There are two main reasons to move from the classical formal analytical underpinning of the canonical static spatial model of party competition to systematic interrogation of computer simulations using ABMs. The first concerns fundamental *analytical intractability* of complex dynamic models of MDMPC, which are tractable using computer simulations. The second, and much deeper, reason involves a reassessment of behavioral assumptions about agents. Confronted with fundamental intractability in their decision making environment, and in the resulting absence of formally provable best strategies, agents are constrained to use decision heuristics. ABMs, quintessentially, are computational models that elaborate emergent features of interactions between agents who deploy decision heuristics rather than analytically provable best strategies.

Building on earlier work by Kollman, Miller, and Page (2003b, 1992, 1998) and by De Marchi (1999, 2003), Laver (2005) developed an ABM of MDMPC. In the Downsian tradition, this assumes nonstrategic "proximity" voters with ideal points randomly drawn from a bivariate normal distribution, each voter supporting the closest party.[2] Party leaders select policy positions without knowing the ideal point of any voter, basing their choices on the inferences they can draw from past positions and support levels of each party in the system. Laver defined four policy-selection rules suggested by traditional empirical literatures on intraparty decision making:

- STICKER: never change position (an "ideological" leader)
- AGGREGATOR: set party policy on each dimension as the mean position of all party supporters (a "democratic" leader responding to supporter preferences)
- HUNTER: if the last policy move increased support, make the same move; else, reverse heading and make a unit move on a heading chosen randomly from the arc $\pm 90^\circ$ from the direction now being faced (a Pavlovian vote-forager)
- PREDATOR: identify largest party; if this is you, stand still; else, make a unit move toward largest party (an autocratic leader seeking votes by attacking larger parties)

Laver's most striking finding was that party leaders using the *Hunter* rule, despite its simplicity, are systematically more successful at finding popular policy positions than party leaders using any of the other rules investigated. Furthermore, *Hunters* tend to seek votes toward the center of the distribution of voter ideal points but very systematically avoid the dead center of this distribution. Laver and Schilperoord (2007) build on the arguments of "citizen candidate" models (Besley and Coate

1997; Osborne and Slivinski 1996) and extend the Laver ABM to endogenize the number of parties, while using the same predefined set of decision rules. They model the ‘‘birth’’ of new parties at points in the policy space where citizens have relatively high levels of cumulative dissatisfaction, measured in terms of their distance, aggregated over time, from their closest party. The ‘‘death’’ of existing parties is modeled in terms of parties’ inability to maintain their support over some survival threshold. In a birth-evolved system such as this, even unresponsive *Sticker* parties tend not to occupy unpopular locations over the long run. *Stickers* at unpopular locations die; new *Stickers* tend to be born at more popular locations.

The ABM Tournament Test-Bed

Our task was not to create the most realistic possible model of party competition but to create a ‘‘level playing field’’ on the ABM test-bed used to evaluate different decision rules. The crucial requirement was that the tournament not be biased against any particular rule. We augmented the ABM developed by Laver (2005), adding provision for the ‘‘birth’’ of new parties and the ‘‘death’’ of existing parties. Unlike the Laver and Schilperoord (2007) ABM, however, our test-bed exogenously *forces* party births at random locations, rather than enabling endogenous births at fertile locations. In our tournament, one new party is born every twenty periods, regardless of the state of the party system. The new party is given a random spatial location and assigned a decision rule randomly drawn, with equal probability and with replacement, from the set of rules in the tournament. To simulate party death, we introduced a 10 percent survival threshold; parties die if they fall below 10 percent of the vote for two successive elections. Self-evidently, if any party system systematically generates party births without at the same time having a de facto survival threshold, the number of parties in competition will grow relentlessly toward a *reductio ad absurdum* with an infinite number of parties.[3]

In an important departure from Laver’s ABM, our test-bed distinguished between *interelectoral* periods of party competition, in which parties set policy positions in response to published polling information about levels of party support; and *election* periods, in which real votes are cast by citizens, party vote totals are rewarded and punished, and parties die and are born. There is an election every twenty periods of our model. At the end of each of the nineteen interelectoral periods, all party positions and support levels are made public, but these are not taken into account as part of the evaluation of different decision rules. The interelectoral periods give parties time to adapt their positions prior to each election period, while only in electoral periods do outcomes have consequences for the success or failure of particular parties.

There were five simulation runs for each rule in the tournament, each run lasting 220,000 periods. The first 20,000 periods were discarded as ‘‘burn-in’’ to remove any effects arising purely from arbitrary initial conditions. Thus, the scoring phase of each run lasted for 200,000 periods and 10,000 elections. Since twenty-five decision rules were submitted in addition to the four presubmitted rules, the tournament involved 145 burnt-in simulation runs, 29,000,000 periods, and 1,450,000 elections. Appendix B (online) discusses methodological issues relating to model burn-in and the long-run convergence of results.

Tournament rules specified that policy-selection rules could only make use of the following information: any previously announced policy position and level of voter support for any party; and the mean or median position on each dimension of the ideal points of the party's own current supporters. Valid policy selection rules did *not* have access to the ideal point of any individual voter or to the decision rule being used by any other party, and they were *not* allowed to ask voters what they would do in counterfactual situations.

3. Features of ABM Test-Bed Differing from Earlier ABMs of MDMPC

Knowledge of Preentered Decision Rules and Anticipation of Others

A vast number of different decision rules could be designed for the setting under investigation; the success of any given rule depends crucially on the other rules against which it must compete. In the iterated PD for example, *Tit-for-Tat* is far from optimal if every other agent uses *Cooperate Unconditionally*, to which *Defect Unconditionally* is a much better response. Computer tournaments identify decision rules that are optimal *within a given set*, and it is thus significant we explicitly declared the four decision rules from Laver (2005) as preentered in the tournament but ineligible to win. The designer of any new rule thus knew that this would at the very least be tested against *Sticker*, *Aggregator*, *Hunter*, and *Predator*. In addition, rule designers knew that many other rules, of unknown (but somewhat foreseeable) content, would be entered by others. A successful decision rule would thus have to be "robust" to other types of rule it might encounter. Strikingly in this context, one submitted rule (*Genety*) used a series of simulations running a genetic algorithm to optimize its parameter settings against not only the four presubmitted rules but also against nine other hypothetical rules taken to represent types of rule it might be expected to encounter.

Diverse Rule Set/Rules Competing against Themselves

ABMs of MDMPC typically investigate competition between agents all using the same decision rule, although Laver (2005) investigated limited examples of competition between pairs of rules, including *Hunter–Predator*, *Hunter–Sticker*, and *Aggregator–Predator* competition. Our computer tournament, in contrast, is designed to explore, in a systematic way, two quite distinct aspects of the effectiveness of decision rules: their effectiveness in competition with *other rules*; their effectiveness in competition with *themselves*. Given twenty-nine rules in the tournament, random selection of rules for newborn parties, an average of nine parties in competition and an average of one party dying each election (given the 10 percent survival threshold), the probability a randomly drawn decision rule differs from every surviving rule is $(28/29)^8 \approx 0.75$. In other words, given the number of different rules entered into the tournament and the random selection mechanism, any given decision rule would typically not be competing with itself, reducing a relative disadvantage of rules, such as *Predator*, that do badly in competition against themselves. However, this situation is quite different for more successful rules, which are more likely to survive and thus more likely to find themselves competing against other agents using the same rule.

The fact that decision rules must compete both against other rules and against themselves raises a very important issue for the computer tournament methodology, thrown into sharp relief by the Jennings "master–slave" suite of strategies which swept the board in the twentieth-anniversary rerun of the Axelrod PD tournament.[4] Rules may be programmed to recognize each other using a "secret handshake"—an obscure sequence of moves known only to themselves[5]—after which the rules can collude in some way. An interesting and deep question arises as to whether the act of collusion is "cheating," at least in terms of the spirit if not the letter of the tournament method. Thinking generally about political competition, it does not seem to us that collusion is unrealistic. In a general political setting in which agents do not know everything, it seems perfectly realistic for an agent to have to consider the possibility that another agent with which he or she is competing is not in fact an autonomous decision maker but under the control of some unknown third agent.

A version of this issue did arise in our own tournament, since two of the rules submitted were programmed (a) to use a secret handshake to recognize other agents using the same rule and (b) never to attack other agents using the same rule. Since we had not explicitly prohibited rules using secret handshakes, we accepted these rules as valid tournament entries. However, we regard it as substantively very unrealistic to model a world of MDMPC in which agents collude merely because they use the same decision rule—it is not clear, for example, why two parties deploying the *Hunter* rule would help each other *simply because they both use the same decision rule*. We thus reran the entire tournament, having edited the code of two rules using a secret handshake, disabling this so that any attempt to infer the rule type of other agents had to be based on public information. As it turns out, the handshakes did not alter our results, but in future tournaments, we will seriously consider prohibiting on substantive grounds the use of decision rules that rely in some way on a secret handshake.

Survival Threshold

The 10 percent survival threshold was a declared exogenous feature of our tournament environment. When the success of decision rules is measured in terms of their performance in long-running simulations such as ours, one important feature of a successful decision rule will be its ability to "stay alive" over the survival threshold. The existence of a survival threshold thus introduces a completely new feature of the policy-setting environment for party leaders. Decision rules are rewarded for finding policy positions that keep their agent alive over the long term, as opposed to (more risky) positions that might increase short-term support but also increase the probability of death. It seems to us to be entirely reasonable substantively, and not at all a tournament artifact, to assume that successful party decision rules will be those that keep their agents above some de facto survival threshold over the long term, as opposed to those that maximize support in the short term, regardless of long-term survival.

The Twenty-Period Electoral Process

The ABMs of Kollman, Miller, and Page (1992, 1998, 2003a, 2003b) involve defined electoral campaign periods while that of Laver (2005) does not, modeling a continuous underlying process of party competition. In our tournament, elections are held and payoffs distributed every twenty periods. In the nineteen intervening periods, parties adapt their positions and get feedback on support levels, but no rewards and punishments are meted out by the system. One reason for this is a pure tournament artifact. It allows new parties time to adapt away from potentially unfavorable random birth locations. The second reason has what we consider to be a very sound substantive basis. It allows all parties to make use of information gathered cheaply during an interelectoral period (for example, public opinion poll feedback) with the intention of setting an optimal position on election day, when, as any walking, talking politician will tell you with great relish, the "real" election is held (and "real" rewards or punishments distributed). Our test-bed models an extreme situation in which party leaders can choose any action at all, without cost, at an interelectoral period; the moment of reckoning comes only at an election period. Interelectoral maneuvering can thus be designed to explore the system with the intention of optimizing payoffs at an election. In any real process of party competition, it is clear that the election-day cycle is indeed critical in terms of party payoffs; but it is also clear that interelectoral maneuvering is not costless.

4. A Portfolio of Policy-Selection Rules for Political Parties

As noted, twenty-five decision rules were entered into the tournament. Each entry is described, in its authors' own words, in Appendix C (online). Here we present a broad-brush discussion of some general features of submitted rules.

Center-Seekers

While explicit use of the center of the distribution of supporter ideal points was not permitted under tournament rules, several submitted rules were designed to find ways to move toward this point. One way of approximating this, for example, is to find the dimension-by-dimension median of party positions at the previous period, or the vote-weighted centroid of these, information for calculating which is available from previous published locations and support levels. There is, however, no reason to believe from investigations of previous ABMs that parties locating at the voter centroid, or indeed any other fixed point in the space, will maximize party support. Indeed, a characteristic finding of previous work is that they do not. This tendency is considerably exacerbated if several competing decisions rules all deploy center-seeking behaviors at the same time. Party support at *any* fixed point in the space depends crucially on the locations of other parties.

Tweaks of Preentered Rules

Several submissions set out to refine *Hunter*, the most successful rule investigated by Laver (2005). One tweak involved moving, if punished with static or declining vote share, toward the most recently successful adjacent party. Another gave *Hunter* a very distinctive "secret handshake" step size to ensure that two agents using the same rule never attack each other, while otherwise behaving like *Hunters*. (As noted, we disabled this feature when we reran the tournament with no secret handshakes.) One tweak of *Predator* moving toward the largest adjacent party rather than the largest party overall, potentially alleviating the problem that unreconstructed *Predators* always converge on precisely the same point in the policy space. Since *Sticker*, do-nothing, is a completely unresponsive "baseline" decision rule, it might seem hard to refine. However, one tweak that was in the event surprisingly successful visits nineteen random locations during the very first interelectoral period and then "sticks" at the location that yielded most votes when visited.

Interelectoral Explorers

One rule made very explicit use of each interelectoral period, systematically exploring the adjacent policy space in the first half of the period, revisiting and refining promising locations in the second half of the period before selecting a final policy position for the election in period twenty. As already noted, interelectoral moves were (unrealistically) cost-free in our tournament. We will see below, however, that a striking feature of our findings is the emergence of an *endogenous* penalty for position-selection rules that change their parties' positions "too much."

Parasites

Not unrelated to the *Predator* rule but with a far less flattering public image, a number of submitted decision rules were effectively "parasites"—indeed, one was explicitly named *Parasite*. This rule, which was characteristic of the genre, stays put for the nineteen interelectoral cycles and then goes directly to a location very close to that of the party with the highest support in the period immediately preceding the election. All of the parasite rules submitted were ‘‘multihost’’ parasites, not designed for one particular species of host. While the distinction between multihost and single-host parasites is unlikely to be significant in the context of the present tournament, it is likely to become much more relevant in the evolutionary environment to which we will be turning (see below), where different species of host might develop different types of defense against the same parasite, with obvious potential effects on the evolutionary stability of parasite rules (Gandon 2004).

The impact of parasites on party policy positions, as with real parasites, is likely to be complex. Parasite parties are unlikely ever to win more votes than any other party, since even if there is only one parasite in the system, it is likely to attract about half of the support of the largest host party, on average splitting the host’s support down the middle. With more than one parasite party in contention, parasites will be forced to share the support levels of their hosts. However, parasites do systematically punish other parties for being successful, and as we shall see, this did happen in our tournament. This could have a significant impact on the outcome of party competition in a system where parasites are common. Imagine a position-selection rule that is perfect in every other respect but takes no account of parasites; other things equal, it always wins every election. In a competition with a set of parties including parasites, however, ‘‘success’’ paradoxically almost guarantees failure. None of the submitted nonparasite rules anticipated parasites. None of the submitted parasite rules anticipated the possibility of different species of parasite; indeed, most were not even designed to deal with other agents using the same parasite rule.

Satisficers and Survivors

A number of submitted decision rules, rather that setting out to maximize party support, were either *survivors* (aiming to keep party support levels above the public 10 percent survival threshold) or *satisficers* (aiming to keep support over some private comfort threshold that was higher than the survival threshold). One submission, for example, made tiny random moves when above the survival threshold and only explored the space for a better location after falling below the threshold for three consecutive periods.

There are several different rationales for satisficing in the world of dynamic MDMPC. The first involves a substantive behavioral assumption, prefaced by an obligatory citation of the work of Herbert Simon, that real people satisfice rather than maximize. We find two different subrationales within this tradition. One is a psychological assumption about the motivations of real humans: that what they actually want is ‘‘enough,’’ rather than ‘‘as much as possible,’’ of some payoff. The other has to do with decision making in a complex and/or low-information environment, even by agents who seek to maximize payoffs. Such (Simonesque) satisficing involves selecting from the set of actions that can feasibly be evaluated, in the knowledge that this is not the universe of all possible actions. ‘‘Win–stay/lose–shift’’ decision rules have been shown to be effective in such settings (Nowack and Sigmund 1993). A win–stay/lose–shift rule maintains the same behavior if satisfied and changes behavior if dissatisfied. In the context of party competition, Bendor, Mookherjee, and Ray (forthcoming) model a dynamic two-party incumbent–challenger setting in which the winning party (incumbent) uses an ‘‘ain't broke, don't fix it’’ rule and sticks at its current policy position, while the loser (challenger) explores ways to increase its support.

Dynamic, and especially evolutionary, models generate yet another rationale for satisficing. In a long-run dynamic setting, a decision rule designed to maximize payoffs in any single period may not maximize these over the long run, whereas a rule designed to satisfice at any given period may indeed maximize payoffs over the long run. Our tournament highlights at least two ways in which this can happen. First, maximizing rules may well be ‘‘riskier’’ than satisficing rules, so that when they are good they are very, very good, but when they are bad they are horrid. In long-run dynamic systems with a survival threshold, rules designed to maximize at every period may well generate higher variance payoffs than more ‘‘conservative’’ rules that satisfice in some way over the short run. A short-run maximizer may have higher long-run expectations than other rules if the survival threshold is ignored. Given a survival threshold, however, higher variance payoffs may imply lower survival probabilities for a short-run maximizer and thus lower long-run expectations (McDermott, Fowler, and Smirnov forthcoming). Thus, the need to stay above a survival threshold can mean that long-run maximizing implies short-run satisficing. A second reason to satisfice rather than maximize in a dynamic setting has to do with competitive multirule environments that include parasites. Since parasites systematically punish decision rules that would otherwise have been the most successful, a rule designed to keep its agent above a certain performance threshold, rather than to maximize in the short run (even in a world with perfect information and no survival threshold), might well yield greater long-run payoffs if it avoids the attention of parasites. Putting all of this together, the distinction between satisficing and maximizing blurs considerably in a long run dynamic setting

5. Success of Submitted Decision Rules

Beating a Static Benchmark

One benchmark for the success of any decision rule in our computer tournament is its success at winning votes relative to the unresponsive *Sticker* rule. A basic criterion for evaluating any *dynamic* policy-selection rule is that it should perform better than a rule that randomly picks a policy position and then never moves.

Perhaps the most remarkable feature of our tournament was that *sixteen of the twenty-five submitted rules were less effective at winning votes than the unresponsive* Sticker *rule*. Many of the scholars who submitted unsuccessful rules were experienced and well-published specialists in static spatial models of party competition. Their failure to devise effective decision rules for setting policy positions in dynamic MDMPC is striking testimony to the complexity and intellectual challenge posed by such a setting. This also reminds us that, given the fundamental analytical intractability of MDMPC, *any* rule for setting party policy positions is a decision heuristic—a decision-making rule of thumb as opposed to a formally provable best-response strategy. This means that rule designers are as well if not better served by intuition and empirical experience as they are by formal analytical firepower. Nonetheless, seven of the twenty-five submitted rules did indeed win significantly more votes than *Hunter*, the most successful preannounced rule. These rules are listed, in order of success, in Table 1, together with a summary sketch of their logic. We return below to characterize features of successful rules.

The Tournament Winner

The tournament winner, the rule whose agents won most votes over the very long run, was *KQ-Strat*, submitted by Kevin Quinn. Furthermore, had we decided to use votes per party as the criterion for tournament success (as opposed to votes per rule) or number of elections survived, the results would have been essentially the same. (Online Appendix D shows box plots of these additional results.) Under any of these criteria, the set of decision rules beating *Hunter* and the set of rules losing to *Sticker* were the same.

The winning rule, *KQ-Strat*, was in essence a satisficing parasite with a secret handshake, hardwired not to attack itself. *KQ*-parties jittered when over the threshold, using tiny random moves with a very distinctive step size recognizable to other *KQ*-parties. When under the threshold, a *KQ*-party moved very (very) close to the position of a randomly selected other party over the threshold, provided this was not another *KQ*-party. Since *KQ-Strat*'s secret handshake programming is controversial and could be seen as substantively unrealistic, we reran the entire tournament, making the sole change of editing the *KQ-Strat* code (and that of another rule, *Raptor*) to disable secret handshakes. Figure 1 reports full results of this "corrected" tournament in box plot format and shows that *KQ-Strat* won convincingly, even with its secret handshake disabled.

Given the large number of unsuccessful submitted decision rules, measuring success against the static *Sticker* benchmark, it is possible that rules were successful because they were typically pitted against ineffective rules. We investigated this possibility by rerunning the tournament using only the top set of seven rules listed in Table 1.[6] Vote shares in this "runoff" tournament are reported in Table 2 and

Table 1
Sketch of Decision Rules Significantly More Effective Than Hunter, Ranked in Order of Tournament Success

Rule	Description
KQ-Strat	If alive for 3 iterations and below survival threshold for 3 iterations and have not moved more than 1.46e-4 units in 3 iterations then set new position at p + e, where p is the position of a randomly chosen party that did not previously move exactly 1.46e-4 units and whose current support is greater than threshold and e is a N(0, .0625^2) random variable. Otherwise move 1.46e-4 units in random direction.
Shuffle	Do *Aggregator* if over 11.5% or more votes in previous round; do *Hunter* if at 11.5%-8.0%. Else divide space into quadrants around weighted party centroid; pick a random location in the quadrant in which votes/party is highest.
Genety	Weighs three vectors to choose where to move: (a) direction of estimated voter centroid; (b) direction of current supporter centroid; (c) direction indicated by *Hunter* rule. Static weights of these vectors determined in simulations by applying a genetic algorithm to optimize in competition with the four pre-submitted strategies and nine other posited alternatives.
Fisher	(a) For the first 10 inter-election periods, random walk through the space using large steps, recording support levels at visited locations. In remaining periods, refine search around best location. (b) If support in an election above survival threshold, reduce step size to 0.2 of original and do (a), exploring close to successful position. (c) If party support below threshold, repeat (a) with original step size until finding a position with support above threshold, then do (b).
Pragmatist	Combination of the vote-weighted mean location of all parties and the party's own median voter. A normally distributed noise term prevents parties with the same strategy from overlapping with each other
Sticky-Hunter–Median-Finder	If over survival threshold in last election, do *Sticky Hunter*, else do *Median Finder*. *Sticky Hunter:* If under survival threshold in last two periods: if last step increased vote share step forward 0.135; else pick a direction from the first and last third of the opposite 180 degrees and make a 0.135 step. Else do nothing. *Median Finder:* Move to median position of current supporters.
Pick-and-Stick	In the 19 periods before the first election, locate party at random points in the space. Then return to the point at which it received most votes and stay there for subsequent elections.

are compared with vote shares for the same rules in the full tournament with twenty-nine different rules.

KQ-Strat pulled ahead of the competition in the runoff. Of the other rules, *Shuffle* and *Fisher* declined strikingly in relative success, suggesting that their success

Figure 1
Total Votes Received, by Rule (*KQ-Strat* Secret Handshake Disabled)

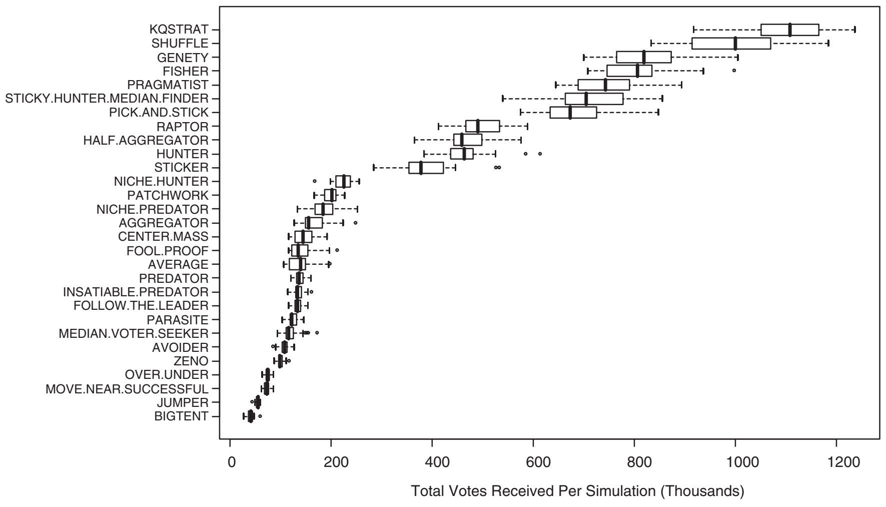


Table 2
Vote Percentages, by Rule for Runoff Simulations Using Only the Top Set of Seven Submitted Rules

	Runoff		Tournament	
Rule	Mean Vote	Median Rank	Mean Vote	Median Rank
KQ-Strat	19.6	1	11.2	1
Pick-and-Stick	15.4	2	6.8	6
Sticky Hunter	15.0	3	7.3	5
Genety	14.0	4	8.4	4
Pragmatist	13.6	5	7.4	5
Shuffle	11.6	6	9.7	2
Fisher	10.9	7	7.9	4

in the full tournament was partly the result of being pitted against ineffective rules. Perhaps most surprising is the increased success of the very simple *Pick-and-Stick* rule, which came in second in the runoff. As we show in the next section, a general feature of our tournament was that, despite the fact that changing policy position was cost-free, successful decision rules tended *not* to make big changes of policy position.

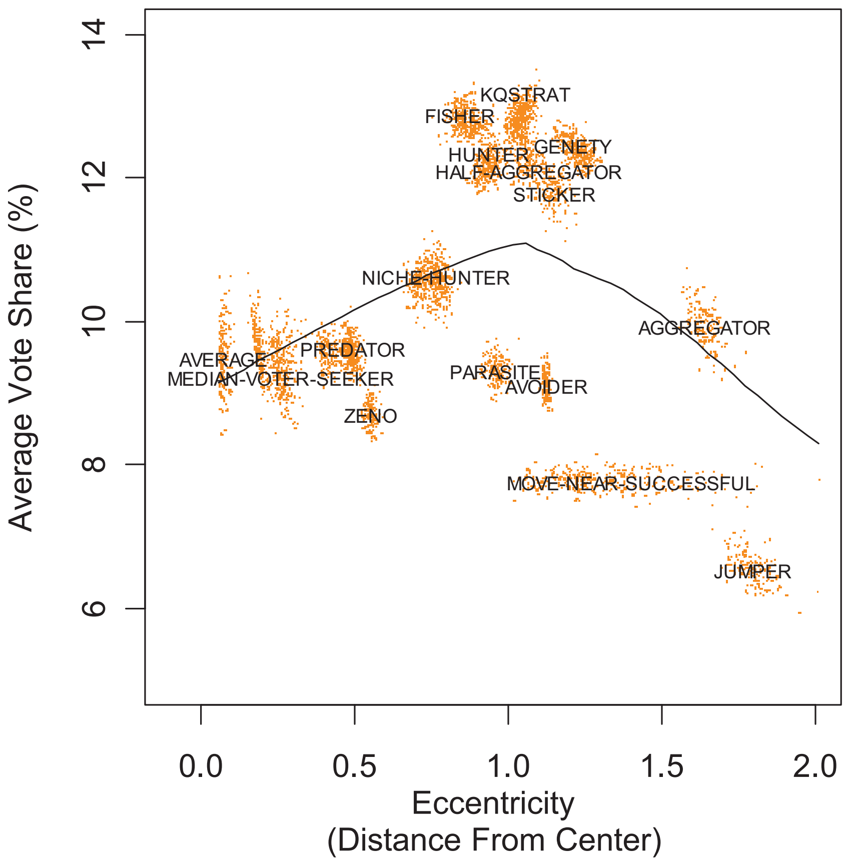
Characteristics of Successful Rules

As Table 1 shows, the decision rules that beat *Hunter*, the most successful preannounced rule, are a mixed bag. Three rules condition on the public survival threshold *(KQ-Strat*, *Fisher*, *Sticky-Hunter/Median-Finder*) and one on private thresholds (*Shuffle*). Two rules exploit the nineteen-period interelection phase (*Fisher*, *Pick-and-Stick*). One rule (*Genety*) set out to optimize against predeclared rules. As noted, *KQ-Strat* is parasitic when not satisficing, capitalizing on positions set by other successful rules. However, rules that were *purely* parasitic did quite badly in the tournament. Thus, most of the rules that beat *Hunter* exploited a specific feature of the tournament environment not available to *Hunter*.

This focuses our attention on the need, if we want to learn useful lessons from exercises such as this, to *avoid tournament artifacts*. In the present context, we believe that the features of our tournament environment that were exploited by successful rules are, with one exception, substantively realistic. For example, in the real world, the survival threshold is generated by various features of the competitive environment, including the electoral formula, districting regime, party and campaign finance regimes, and many other local institutional details. We also see it as realistic to distinguish between interelectoral periods, during which moves are made and information gathered, and electoral payoffs, where the big political payoffs are distributed. However, as we have already noted, we regard rules using secret handshakes as exploiting a tournament artifact. In the real world, parties do collude, but they probably do so on the basis of shared interests rather than shared cognition (i.e., using the same heuristic). Thus, we disabled secret handshakes and reran the tournament without them.

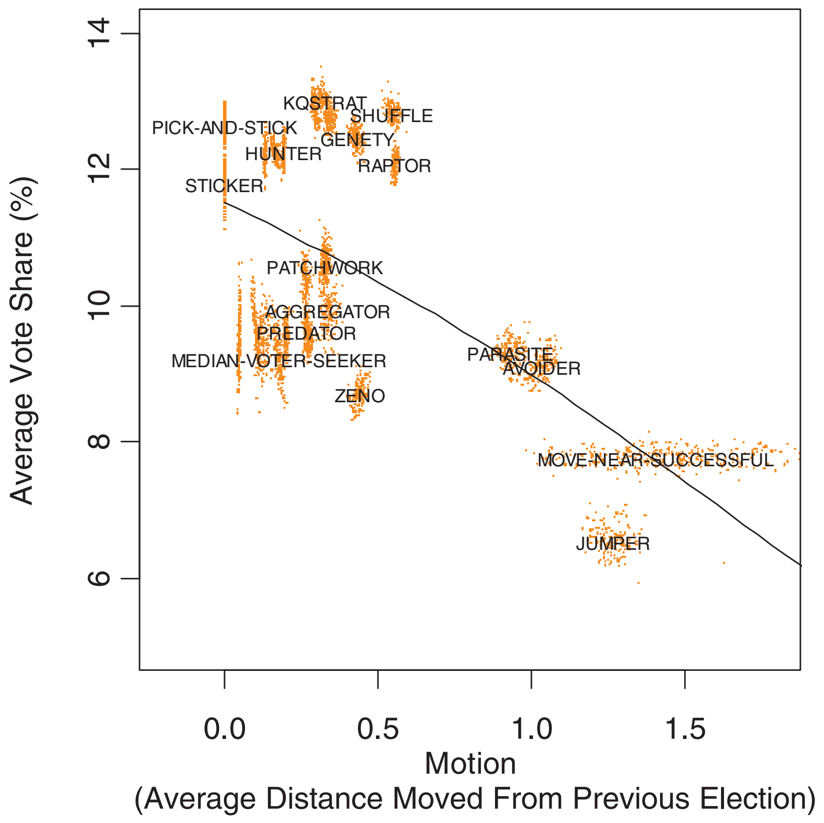
In spite of the diversity of approaches used by successful rules, there were some striking regularities. First, there were systematic trends in how far from the center of the voter distribution successful rules tended to locate. Figure 2 plots mean vote share of each party type against its mean Euclidean *eccentricity*—distance from the centroid of the voter distribution, measured in standard deviations of the voter distribution. This yields 145 observations for each rule entered,[7] and we use a full-bandwidth loess procedure to draw a line summarizing this relationship across all observations. It is clear that successful position-selection rules tend to locate their parties about one standard deviation away from the center of the space. This echoes findings by Laver (2005) that *Hunters* tend to hunt for votes at somewhat less that one standard deviation of the voter distribution away from the center, and from benchmarking simulations by Laver and Shilperoord (2007), showing that the mean distance of voters from their closest party tends to be minimized when randomly scattered parties locate about one standard deviation from the center. Faring much worse were center-seeking rules, as well as rules such as *Aggregator* that tend to set policy positions at more extreme spatial locations.

Figure 2
Relationship between Average Vote Share per Party and Eccentricity



Second, our simulated system of party competition systematically generates an *endogenous* penalty for parties that tend to change their positions by large amounts from one election to the next, although no such penalty was programmed into the system. Figure 3 plots the average vote share of each party type in each simulation against *motion*—the mean Euclidean distance each party moves from election t to election $t+1$. The relationship is unambiguous. Decision rules that generate a lot of movement from one election to the next (for example, *Parasite*, *Avoider*, and *Move-Near-Successful*) are much less successful than rules that tend to stay put. Recall that *Pick-and-Stick* performs remarkably well despite the fact that it never

Figure 3
Relationship between Average Vote Share per Party and Motion



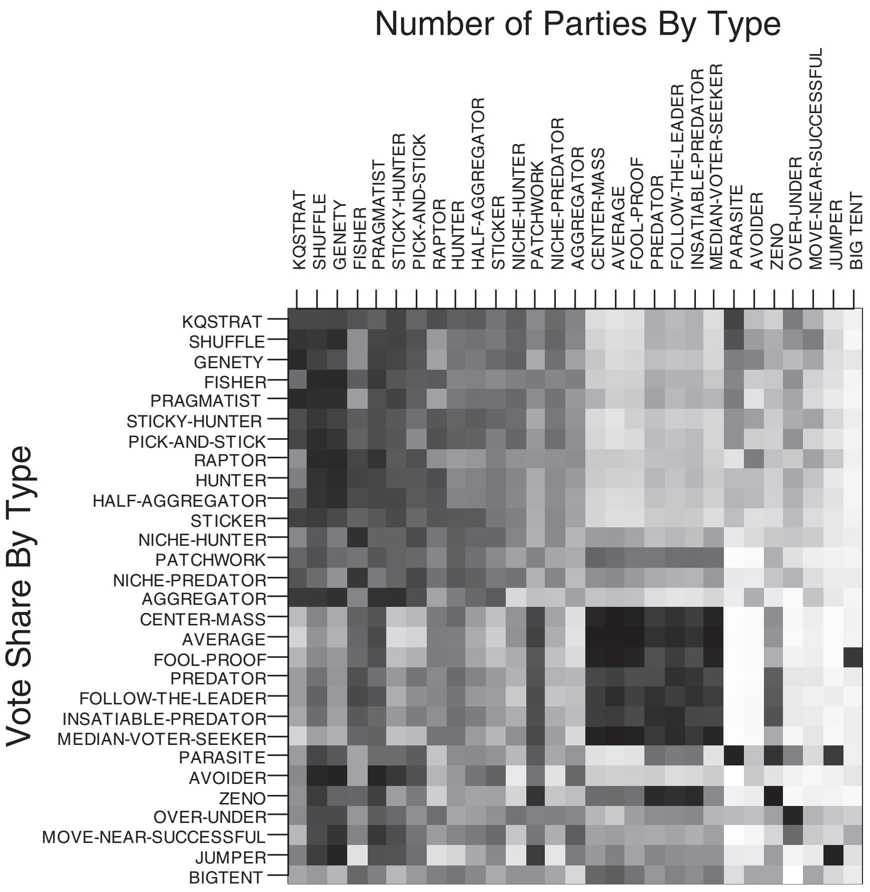
adapts at all after the first election. All position-setting rules for political parties face the classic exploration–exploitation trade-off, between exploiting a position that has previously been successful and exploring for a new position that might improve on this, but might also be worse. What is striking and somewhat unexpected is how little there is to gain from exploration in this setting. Although a few parties fared better than *Pick-and-Stick* in the full tournament, only *KQ-Strat* performed better in the runoff. And parties using *KQ-Strat* typically make near-zero adjustments to their position, changing policy position only when facing a shift in electoral conditions that puts them under the survival threshold for more than three periods.

Analyzing how well pairs of decision rules performed when pitted against each other, we see this exhibits strong clustering by type. Figure 4 shows a visual representation of the set of pairwise correlations between vote shares received by a given party type (vertical axis) and the number of parties of a given type (horizontal axis).[8] Lighter shades indicate increasingly positive correlation and darker shades indicate increasingly negative correlation. Parties are ordered by their final performance in the tournament, best to worst. Thus, going from left to right of the top row, we see that *KQ-Strat* performed relatively worse against the more effective rules and relatively better against the less effective rules. We also notice a vertical ‘‘spike’’ for *Parasite* near the top right-hand corner of the plot. As we anticipated earlier, while not doing particularly well in the tournament overall, *Parasite* inflicted systematically more damage on the more successful decision rules when pitted against these. Indeed *KQ-Strat,* the most successful of all the rules, was particularly vulnerable to the presence of *Parasites*. However, *Parasite* was also vulnerable to other rules, and most notably to itself. Nonetheless we can see that if parasites do exist in a party system, they are likely to have a considerable effect on the evolution of party competition—in particular by systematically punishing otherwise successful decision rules.

Figure 4 highlights two distinctive blocks of party types. The darker block in the top left-hand corner, extending from *KQ-Strat* to *Aggregator* shows, unsurprisingly, that successful rules had most difficulty in competition with other successful rules. The very dark square toward the bottom right, running from *Center-Mass* to *Median-Voter-Seeker*, is generated by rules designed to move either toward the center of the policy space or toward other parties. This was thus a set of rules that effectively competed either with themselves or with other similar rules, and not with the top set of position-selection rules, which, as we have seen, tended to locate about one standard deviation away from the center. This emphasizes the point that while prospecting the center of the voter distribution can be rewarding for a party acting alone or in competition with just one other party, as the number of interacting parties increases, the more center-seeking parties there are, the worse things get for each of them.

Turning briefly from success to failure, in addition to the relentlessly poor performance, already noted, of rules that seek out a fixed point such as the voter centroid, it is interesting to note the signal failure of the decision rule that was by far the most computationally intensive, *Bigtent*. This was the least successful rule in the entire tournament; it used a sophisticated statistical method to estimate locations of ‘‘unknown’’ voter ideal points by analyzing past party positions and vote shares. Its catastrophic failing was that it was always killed off before it came close to achieving this. Note in this context that tournament rules imposed *no penalty on decision rules that were computationally expensive*. A more realistic dynamic model might well penalize rules that take far longer to decide what to do than their rivals. However, a striking feature of our results was that even without a penalty

Figure 4
Effect of Other Party Types on Average Vote Share per Party



for slow performance, successful rules were neither computationally intensive nor convoluted in their logic.

6. Evolutionary Competition between Rules

The simulations reported thus far were designed to provide a level playing field for all decision rules in the tournament rather than to be substantively plausible. Of much more substantive interest, however, is an evolutionary environment in which "reproduction" probabilities of different rules are not constant, but evolve dynamically as functions of each rule's past success. This type of "replicator dynamics" models a process in which new parties entering the system are more likely to adopt decision rules that have been successful in the past (Weibull 1995). Replicator dynamics set a rule's reproduction rate proportional to its *fitness*, measured in terms of past success. In what follows, we measure a rule's instantaneous success at any single point in time as the *mean percentage vote share won by all parties using that rule at that time*. The crucial issue concerns the manner in which success at each previous instant in history is aggregated into some memory of past success—which is then used to condition rule fitness. If the fitness regime has "too long" a memory, then rule fitness may be overly influenced by long-gone outcomes that no longer have relevance to the current state of the system. If it has "too short" a memory, then it may overreact to quite possibly random short-term fluctuations. This is another manifestation of the "exploitation–exploration" trade-off discussed in the reinforcement learning literature (Sutton and Barto 1998). One solution is to discount past success exponentially when computing current rule fitness. This can be computed rather elegantly using a recursive updating algorithm. Let F_{rt} be rule r's fitness at the beginning of period t. This is updated by $V_{r(t-1)}$—vote share/party-using-r at the end of period $t-1$:

$$F_{rt} = \alpha F_{r(t-1)} + (1-\alpha)Vr_{(t-1)} : 0 \leq \alpha \leq 1.$$

Thus, α is the memory parameter of the regime under which reproductive fitness evolves. If $\alpha = 0$, we have a "goldfish memory" regime that conditions rule fitness only on success in the immediately preceding period. When $\alpha = 1$, we have a regime in which fitness never updates. Set in this more general context, our original tournament is a special case with a fitness regime for which $\alpha = 1$, and with priors, never updated, that all rules have equal reproduction rates. A fitness regime for which $\alpha = .5$ is highly reactive, with an agent's fitness level eight periods previously contributing only about 1 percent of the information in its current fitness. In the evolutionary simulations we report below, we model a fitness regime for which $\alpha = .9$ under which, to get a sense of things, an agent's fitness level thirty periods previously contributes about 5 percent of the information in its current fitness.

The final issue to consider in this context is the extent to which there are shocks in the evolutionary system, in the sense that random births occur that are not in accord with the evolving fitness regime. Even though a rule might have been very successful and have a consequently high reproduction rate, there may remain some small probability that an unsuccessful rule will nonetheless be born into the system. Successful decision rules must be able to prosper against random invaders, even after they have become very fit. We model this by defining a probability, π, that a decision rule for newborn party at period t is selected with a probability proportional to current evolved rule fitness, F_{rt}. With probability $(1-\pi)$, the decision rule

Table 3
Vote Percentages for Evolutionary Simulations Using Success-Updated Rule Fitness

Rule	Success-Updated Fitness ($\pi = .9$; $\alpha = .9$)	Original Tournament ($\pi = .0$; $\alpha = 1$)
KQ-Strat	17.0	11.2
Genety	13.9	8.4
Sticky-Hunter	11.7	7.3
Shuffle	11.2	9.7
Pick-and-Stick	10.4	6.8
Pragmatist	9.6	7.4
Fisher	8.5	7.9
Raptor	4.1	4.9
Hunter	3.7	4.7
Half-Aggregator	2.6	4.7
Sticker	1.2	3.9

at birth is selected by setting probabilities equal for all rules in the tournament set. In results we report below, we set $\pi = .9$. Thus, 90 percent of the time rules for newborn parties are selected on the basis of evolved rule fitness; 10 percent of the time every rule in the tournament has an equal chance of being chosen by a newborn party. Key results of this new tournament, which included all twenty-nine rules from the original tournament and run in an evolutionary setting in which $\pi = .9$ and $\alpha = .9$, are reported in Table 3.[9] Each rule for which results are *not* reported in Table 3 won on average less than 1 percent of the vote.

Table 3 shows that, as might be expected, the gap between successful and unsuccessful rules widens when rule reproduction probabilities evolve as a function of past success. By far the most important finding in Table 3, however, is that the tournament-winning *KQ-Strat* did not even come close to "driving out" all other decision rules against which it competed. Table 3 shows that *each of the top set of seven rules prospered in this evolutionary setting*. Our evolutionary simulations converged on a limiting distribution in which elements in the diverse top set of rules coexisted with one another. For us, the evolutionary stability of this diverse set of rules for setting party policy positions is perhaps the most substantively important result of our work to date. It suggests that evolutionarily stable party systems may include parties deploying a diverse set of different decision rules, rather than all using the same rule which has in some sense evolved as "the best."

7. Conclusions and Future Work

The evolutionary simulations we have just described implement simple replicator dynamics, making reproductive fitness a function of past success and assuming that all ‘‘children’’ are perfect clones of their ‘‘parent’’ rules. Obviously, this setting can never allow completely new decision rules to evolve—results are determined by the identity of a portfolio of rules that is exogenously set at the start of the process. The purpose of these simulations is thus to investigate the evolved relative fitness of each of the rules in the starting portfolio, and in particular to look for situations in which one decision rule drives out all others, or in which there are limiting states involving combinations of successful rules within the starting portfolio. The next step is clearly to move to an evolutionary environment in which the *content of decision rules is not fixed exogenously*, but instead evolves endogenously. We can model such an environment using the *genetic algorithm*, although implementing the genetic algorithm in the substantive context of position-selection rules for political parties presents robust challenges.

Consider the classic genetic operators of *mutation* and *crossover* and begin by thinking about an evolutionary setting in which party decision rules reproduce *asexually*, with mutation but no crossover. This in effect involves a small but vital modification to the replicator dynamics we described in the previous section. Reproductive fitness remains a function of past success. However, ‘‘child’’ strategies are no longer bound to be perfect clones of their ‘‘parents,’’ but may be subject to random mutations at the point of reproduction. Substantively, this models a situation in which newborn parties select a decision rule based on their observations of the past success of decision rules used by other parties, but that random mutations (which we might think of in a political context as innovations) may be made to particular rules. Many mutations will be abject failures, but some may be unexpected successes in an evolutionary setting driven by replicator dynamics. The intellectual challenge in doing this is to produce a full parameterization of each of the rules in the portfolio, which can serve as the rule’s ‘‘genetic material’’ that becomes subject to random mutation. The result will be the evolution of more effective ‘‘breeds’’ within predefined rule species—the evolution of ‘‘*Super-Hunters*,’’ for example—whose parameter settings are particularly well adapted to the competitive environment being modeled.

Now consider the genetic operator of *crossover*, the essence of *sexual* reproduction, under which the genetic material of the child is some combination of the genetic material of each parent. We might first want to think about whether there is a substantive story about politics, whereby party leaders have interactions that are the political equivalent of sex, with the result of this interaction being a new set of decision rules, each of which is some combination of features of decision rules used by its parents. It seems to us that such a characterization is neither impossible nor implausible. Once achieved and subjected to the genetic algorithm, it would enable us to model situations where, for example, next-generation party decision rules would blend heuristics from successful party decision rules that had been among their ancestors.

We return, in conclusion, to the outcome of our existing tournament—with its predefined rule set submitted by a heterogeneous collection of scholars. The first and most important point to make is to emphasize the lessons learnt from the Axelrod tournaments. The tournament we report here is but the first phase of a larger tournament method: we advertised a tournament; we received some submissions; we computed and published the results; some lessons have been learned. As the Axelrod tournaments show us, the real payoffs will come, and the robustness of conclusions will develop, as this process is iterated. Rule designers entering our next tournament will know the results of this one. Above all, the conclusions we can draw on the basis of the present tournament depend critically upon the rules that were submitted to it. And substantive inferences we might draw from the patterns we observe depend critically on the assumption that these patterns are not tournament artifacts.

Notwithstanding these caveats, we believe that the results of this tournament do offer striking intuitions for scholars interested in modeling MDMPC in a dynamic setting. First, recall that several of the most successful rules focused on satisficing, keeping their party above some de facto survival threshold that must inevitably be an intrinsic feature of any dynamic system that enables the birth and death of political parties. This is something new that arises when we move from a static to a dynamic setting. On many substantively plausible criteria, a party can be more successful if it satisfices with low variance rewards over the long term rather than maximizes with high variance rewards, and thus a higher risk of extinction, over the shorter term.

The top set of successful rules also included some that made effective use of the interelection period to look for policy positions that win votes at election time. Our simulation test-bed was almost certainly unrealistic by making such exploration costless. However, while our tournament imposes no exogenous penalty for changing policy positions, a strong emergent pattern was that decision rules making large changes in policy positions tended strongly to fare worse than rules making smaller changes in positions. Without explicitly punishing excessive policy movement in our simulated party system, excessive policy movement was punished endogenously as part of the emergent pattern of party competition.

A third headline result from our tournament, rerun in its more evolutionary setting, is that no single decision rule in the starting portfolio drove out all others, with the result that a stable top set of different rules emerged to coexist with each other, each successful rule tending to exploit a different feature of the competitive environment. The results of this tournament have thus strengthened our conviction that further investigations of position selection rules in models of party competition should be set in the context of how each rule performs against a heterogeneous rule set, rather than looking at the dynamics of single-rule systems.

Given these general intuitions from the present tournament, we feel there are two main ways forward. One is, as we have already suggested, to conduct further "empirical" experiments that involve successive tournaments like the one we describe here. Such tournaments are empirical in the sense that they are a way of synthesizing and evaluating what we might take to be the received wisdom about party decision making, within a particular group of scholars, at a particular point in time. (Thus, the current intense intellectual interest in secret handshake protocols, for example, is driven by concerns about the security of Internet commerce—concerns effectively unheard-of at the time of the original Axelrod tournament.) They are of particular significance when the underlying problem under investigation is analytically intractable, since there is no formally provable best-response strategy and solutions to the problem must depend upon the deployment of a set of decision heuristics that has an inevitably empirical character. The other way forward is to develop more general parameterizations of party decision rules. Given such parameterizations, comprehensive and systematic suites of simulations using the genetic algorithm, over the very long run, offer the prospect of being able to explore the space of potential party decision rules to find combinations that are evolutionarily "fit," in the sense of being better than other rules at attracting the votes of ordinary decent citizens.

Notes

1. We do not prove the analytical intractability of this problem here. Interested readers can consult formal intractability proofs for the closely analogous Voronoi Game when played in more than one dimension (Teramoto, Demaine, and Uehara 2006).

2. To model strategic voting in multidimensional multiparty competition (MDMPC) when a party wins a majority of votes requires modeling the following: how voters forecast the election result without knowing the ideal points of other voters, how voters forecast which government will form following postelectoral coalition bargaining, how voters forecast the real-world policy outputs arising from this government, and how voters resolve the calculus of turnout problem.

3. The 10 percent survival threshold was selected because it tends, on average, to result in a system with what we take to be an empirically realistic number of seven to ten parties at any given time. Laver and Schilperoord (2007) ran simulations systematically investigating the "carrying capacity" of dynamic party systems with different survival thresholds.

4. In a twentieth-anniversary rerun of the Axelrod tournament with 223 entries, the competition was won by a 60-entry portfolio of "master–slave" strategies submitted by a team from the University of Southampton led by computer scientist Nick Jennings. Southampton strategies signed in with a distinctive and unusual sequence of early moves and were thus able to recognize each other during play, during which Southampton slaves offered themselves up for exploitation by a Southampton master whenever they encountered one. For details, see www.prisoners-dilemma.com. The Southampton portfolio of strategies was submitted as part of a research program investigating collusion and competition between software agents.

5. Readers interested in secret handshakes can consult an extensive computer science literature on public key cryptography—for example, Balfanz et al. (2003).

6. Seven more 220,000-period simulation runs were performed, with each of the seven rules in the top set the initial party in one simulation.

7. We label the mean location in the figure for several rules, omitting some labels for clarity.

8. Each election permits an observation of vote share for each party type that competes in that election and an observation of the number of parties of each type for all parties. The correlation matrix looks very similar if we convert number of parties to a dichotomous variable coded 1 for any positive number of parties and 0 otherwise.

9. This evolutionary tournament involved twenty-nine separate runs, each of 200,000 periods after discarding a 20,000-period burn-in, each run forcing in one of the submitted strategies as the one to be selected in the very first election. The total number of recorded periods involved in this simulation is thus 14,500,000. Brooks-Gelman tests confirmed statistical convergence of the twenty-nine chains.

References

Axelrod, Robert. 1980a. Effective choice in the prisoner's dilemma. *Journal of Conflict Resolution* 24:3-25.

———. 1980b. More effective choice in the prisoner's dilemma. *Journal of Conflict Resolution* 24:379-403.

———. 1997. The evolution of strategies in the iterated prisoner's dilemma. In *The complexity of cooperation: Agent-based models of competition and collaboration*, ed. R. Axelrod. Princeton, NJ: Princeton University Press.

Balfanz, Dirk, Glenn Durfee, Narendar Shankar, Diana Smetters, Jessica Staddon, and Hao-Chi Wong. 2003. Secret handshakes from pairing-based key agreements. Pp. 180-96 in *Proceedings of the 2003 IEEE Symposium on Security and Privacy (SP '03).* http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber = 1199336.

Bendor, Jonathan, Dilip Mookherjee, and Debraj Ray. Forthcoming. Satisficing and selection in electoral competition. *Quarterly Journal of Political Science*.

Besley, Timothy, and Stephen Coate. 1997. An economic model of representative democracy. *Quarterly Journal of Economics* 112 (1): 85-106.

De Marchi, Scott. 1999. Adaptive models and electoral instability. *Journal of Theoretical Politics* 11 (July): 393-419.

———. 2003. A computational model of voter sophistication, ideology and candidate position-taking. In *Computational models in political economy*, ed. K. Kollman, J. H. Miller, and S. E. Page. Cambridge, MA: MIT Press.

Fowler, James H., and Oleg Smirnov. 2005. Dynamic parties and social turnout: An agent-based model. *American Journal of Sociology* 110 (4): 1070–94.

———. 2007. *Mandates, parties, and voters: How elections shape the future*. Philadelphia: Temple University Press.

Gandon, Sylvain. 2004. Evolution of multihost parasites. *Evolution* 58 (3): 455-69.

Kollman, Ken, John Miller, and Scott Page. 1992. Adaptive parties in spatial elections. *American Political Science Review* 86 (December): 929-37.

———. 1998. Political parties and electoral landscapes. *British Journal of Political Science* 28 (January): 139-58.

———. 2003a. *Computational models in political economy*. Cambridge, MA: MIT Press.

———. 2003b. Political institutions and sorting in a Tiebout model. In *Computational models in political economy*, ed. K. Kollman, J. H. Miller, and S. E. Page. Cambridge, MA: MIT Press.

Laver, Michael. 2005. Policy and the dynamics of political competition. *American Political Science Review* 99 (2): 263-81.

Laver, Michael, and Michel Schilperoord. 2007. Spatial models of political competition with endogenous political parties. *Philosophical Transactions of the Royal Society B: Biology* 362:1711-21.

McDermott, Rose, James H. Fowler, and Oleg Smirnov. Forthcoming. On the evolutionary origin of prospect theory preferences. *Journal of Politics*.

Nowack, M., and K. Sigmund. 1993. A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoners' dilemma game. *Nature* 364:56–58.

Osborne, Martin J., and Al Slivinski. 1996. A model of competition with citizen candidates. *Quarterly Journal of Economics* 111 (1): 65-96.

Smirnov, Oleg, and James H. Fowler. 2007. Policy-motivated parties in dynamic political competition. *Journal of Theoretical Politics* 19 (1): 9-31.

Sutton, R. S., and A. G. Barto. 1998. *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press.

Teramoto, Sachio, Erik D. Demaine, and Ryuhei Uehara. 2006. Voronoi game on graphs and its complexity. Pp. 265-71 in *2006 IEEE Symposium on Computational Intelligence and Games*. http://ieee.org.

Weibull, Jörgen W. 1995. *Evolutionary game theory*. Cambridge, MA: MIT Press.