Campaign Funding Under Uncertainty

Sawyer Birnbaum, Lucy Li, Vivian Hsu
Department of Computer Science
Stanford University
Stanford, CA, USA
{sawyerb, lucy3, vhsu23}@stanford.edu

***Abstract*—Billions of dollars are spent yearly on political campaigns at the local, state, and national levels in the United States. However, money is not always spent efficiently, and the problem of campaign donation allocation has been relatively unexplored. We apply POMDP solving methods to discrete and continuous formulations of this problem. We find that for basic problem descriptions, these solvers achieve comparable scores to near-optimal baselines. We also find that in the continuous space, these solvers outperform baselines when we increase the complexity of the problem description by introducing model uncertainty and by considering multiple races. Our results indicate that POMDP-based methods for allocating donations could provide significant cost-savings.**

***Index Terms*—partially observable Markov decision process, resource allocation, campaign funding, QMDP, POMCPOW, PFT-DPW**

I. Introduction

We develop a model for efficiently allocating donations among political campaigns. To optimize their return on investment, campaign donors must balance between overspending to support campaigns that will win without extra funds or will lose despite them and underspending in races where more money would make a crucial difference. At the simplest level, our problem involves a single donor (with a budget B) and a single campaign. At each of N timesteps, the donor can contribute funds from B to the campaign, and these contributions will affect the state of the race at the next timestep. At the end of the race, the donor receives a large positive reward if the campaign wins and a negative reward proportional to the amount she spent on the race. At the outset of the race, the donor does not know the true level of support for the campaign and must estimate this quantity based on poll results. [1]

II. Background

In the 2018 U.S. midterm election cycle, individuals, parties, and interest groups spent over $5 billion [7]. These expenditures reflect the extent to which political activists believe that money can decide election outcomes. Across all of the House election cycles from 2000 to 2016 (with the exception of the 2010 cycle), over 90% of the candidates who outspent their opponents won. In Senate races over the same period, the top spender won over 80% of the time [1]. Individual contributions, in particular, correlate closely with candidates' final vote shares [2]. Accordingly, those interested in influencing public policy have an incentive to contribute to close races to influence their outcomes.

However, a large portion of campaign contributions are "wasted"; in 2018, in Ohio alone, donor donated over $5 million to House candidates who lost the general election by a margin of greater than 10% [11] [12].[2] Donors to the losing candidates in not-close elections receive no return on their investment in these races, and donors to winning candidates could achieve the same outcomes at a lesser expense. Moreover, both groups could have greater influence on election outcomes by channeled their funds only into close races.

Donors should contribute to the races in which their contributions will be most likely to change outcomes. However, identifying these races can be challenging because 1) donors do not know the true level of support for their preferred candidates, and 2) the effect of a donation on a race varies over time and between races. Moreover, while donations are most impactful early in the election cycle (i.e., before campaign narratives have hardened), at this point, donors have only a vague understanding of which races will be competitive [1]. Thus, an optimal donation strategy for a single race requires balancing the need to collect information about the state of the race against the need to act quickly to set the race's trajectory. The problem becomes even more complex when the donor must consider multiple races.

Allocating campaign donations is an unexplored problem space, but researchers have heavily studied the more general space of resource allocation under uncertainty. For example, the allocation of different resources needed to manage conservation efforts has been modeled as a Markov decision process (MDP) [3] [4] [5] and a partially observable Markov

[1]For project code (including a demo of our continuous model), visit https://github.com/Sawyerb/CS238.

[2]There are, of course, reasons to contribute to politicians other than helping them win elections. Indeed, because campaign contributions can "buy" access to elected official, donors seeking to cultivate government connections may wish to contribute only to safe races [8]. Nonetheless, many donors are interested in shaping the ideological bent of the government and therefore in influencing close elections.

decision process (POMDP) [6].[3] As our problem deals with state uncertainty–i.e., about underlying levels of support–we adopt the latter approach.

Although academics and journalists have built sophisticated models for predicting elections (e.g., at *FiveThirtyEight* or *The Upshot at The New York Times*), we could find no quantitative models advising on a donation strategy [13] [14]. The Princeton Election Consortium occasionally writes articles providing donation advice, but these are based loosely on judgments about which races are close rather than a formal donation model [15]. This project demonstrates that a quantitative, computational approach to donation allocation is feasible and introduces several basic models.

III. Approach

Efficiently allocating donations is a challenging problem. The state space (percentage of vote share that a candidate has), action space (amount of money the donor contributes), and transition model (the effect that a donation has on a candidate's vote share) are all continuous. The true state and transition model are unknown. To assess the former, the donor should consider a range of factors, including polling data, the incumbency status of the preferred candidates, local and national economic trends, and fundraising information [13]. The transition model is even harder to estimate as it depends on such factors as the quality of campaigns' advertising staff and because it changes throughout the race. Furthermore, as the number of races the donor considers increases, the dimensionality of the problem increases, exponentially increasing in complexity.

Thus, we consider a simplified problem. For a race R, we divide the race into N timesteps. At each timestep, the donor sees a poll of the race and can contribute an amount C (constrained by her budget B) that will affect the current actual vote share, S, of the race. The remaining budget is the initial budget with every action subtracted. To assess the effect of a contribution on the race, we evaluate the ratio between the preferred candidate's funds F and the opposing candidate's funds O (i.e., the candidate's money share). The preferred candidate's post-contribution support is given by

$$S_{\text{new}} = T * \frac{F + C}{F + O + C}$$

where T is a parameter governing the relationship between money share and vote share. Note that this formulation means that the preferred candidate's vote share becomes progressively harder to change as it increases above 0.5, in keeping with an intuitive understanding that money is more likely to sway the opinions of swing voters than of committed partisans. To further simplify the problem, we begin by considering only one election and provide the donor with T. When the relationship between support and money is direct and deterministic, $T = 1$ and once the preferred candidate has a vote share above 50%, it continues winning in all future time steps.

We explored simplifying the problem into a discrete model (i.e., where the state, action, and observation spaces are discrete) and considered a continuous model. Details of these models are described in the Models section.

We use historical data to estimate the observation function O (i.e., the polling process) and the transition function $\mathcal{T}$. The observation function tells the donor how the true state of the race (what percentage of the votes a candidate actually has) affects the polling (what percentage of the votes a candidate has as reflected in the polls), and the transition function governs the effect that a contribution has on the true state of the race at the next timestep. After estimating O and $\mathcal{T}$, we model the problem as a POMDP and experiment with various algorithms to find the optimal donation allocation policy.

In general, we run each model for ten timesteps. The donor is not allowed to spend more than she has in her remaining budget. Although in theory the only reward is achieved at end of the election, to encourage learning (especially in the discrete space) we opt to provide a reward at each timestep based on whether the preferred candidate leads the race. Still, to reflect the importance of the election itself, rewards are weighted higher as the number of remaining timesteps decreased. Thus, our reward function is the following, where w_r is the maximum reward for winning, l_r is maximum penalty for losing, and n is the number remaining steps:

$$r = \begin{cases} -a + w_r(\frac{N-n+1}{N+1}) & \text{if } S > 50\% \\ -a - l_r(\frac{N-n+1}{N+1}) & \text{if } S \leq 50\% \end{cases}$$

Each race has two candidates; each candidate's initial funds are calculated from the preferred candidate's initial support S_{10} and from the donor's budget B. The preferred candidate's funds are $F = S_{10} * B$, and the opponent's funds are $O = (1 - S_{10}) * B$. Our donor can only affect her preferred candidate's funds. The opponent's funds remain stable.

After extensive investigation of this simplified problem, we expanded the model to more accurately reflect the problem of campaign donation allocation in two ways: by modeling transition uncertainty and by considering multiple elections.

A. Data

To estimate the observation and transition functions O and $\mathcal{T}$, we gathered voting outcomes and total individual contributions to campaigns for the 2014 U.S. House elections from the Federal Election Commission (FEC) and collected data from New York Times, CBS News, and YouGov polls conducted between October 16 and October 23, 2014. This resulted in polling, voting, and funding percentages for 754 candidates. We calculated the ratio of polls to votes and the ratio of votes to money, as plotted in Figures 1 and 2,

[3]The problem of resource allocation in conservation strategies requires accounting for trade-offs between costs and return on investment: the optimal policy is to allocate funds among actions and areas. Furthermore, the best action to implement depends on uncertain information about the presence of species in the areas of interest. [6] This parallels our problem in that a donor does not know the true level of support for her preferred candidate and must find the optimal amount to donate to maximize her return on investment. In other words, she must use polling results to make a decision on what is the minimum contribution amount that will shift the race in favor of her preferred candidate.

removing outliers where the ratio exceeded 2.0.[4] The observation function closely follows the Tukey-Lambda probability density function, and the transition function closely follows the Johnson SU probability density function; accordingly, we used these functions in our continuous model. These functions have the lowest sum of square error when fitted to a 200-binned histogram of the original data, compared to all other SciPy statistical distributions.

B. Models

*1) **Discrete**:* Under our discrete problem definition, each state includes the number of time steps left, the amount of remaining budget, and the current support of the donor's preferred candidate. We found that it was difficult to use offline solvers efficiently on a large state space. To keep our overall state space manageable, we chose to limit our budget at time n, B_n, to integer values from 0 to 10 and the support S_n to integer values from 0 to 10 as well, where S_n relates to $10 * S_n\%$ of votes. Our win reward parameter, w_r, is also on the same scale.

The distribution of observations is determined by multiplying the actual vote percentage by the ratios of poll percent to vote percent depicted in Figure 1, and then flooring the resulting value. Similarly, if transitions are not deterministic, the distribution of transitions to the next state is determined by multiplying the actual money percent by a sample from the distribution of ratios of vote percent to money percent depicted in Figure 2 and flooring that value. When transitions are deterministic, $T = 1$.

Baseline. The baseline solution for our discrete model is a random policy, where the action at timestep n, C_n, is sampled from all possible donations. Another baseline is always choosing the smallest possible action, which is 0.

QMDP. We used a QMDP solver for our discretely defined problem. The QMDP technique first creates an alpha vector for each action based on the $Q(s, a)$ function under full observability, then computes the alpha vectors using alpha iteration [10]. By multiplying each resulting alpha vector with the belief vector, we get an approximation of the value function at $\max_a \alpha_a^T b$. An important note about QMDP is that it assumes there is no state uncertainty at the next step.

*2) **Continuous**:* For a more realistic and flexible design, we modeled the problem with continuous states, actions, and observations. This formulation of the problem uses the same reward model as the discrete formulation but without binning. The transition model is largely similar, but the T parameter is fixed beforehand rather than being sampled at each timestep. As with the discrete problem, we set $T = 1$ for our basic experiments. For the multi-race experiment, we sample T from $\mathcal{T}$. To improve readability, we increased the amount of money in the system from 10 to 1000; actions and scores are scaled accordingly. We developed three solvers for the continuous model.

Baseline. Our baseline model accepts as input a poll value P, remaining donor funds F, candidate funds C, opponent funds O, and money-to-votes parameter T. At each timestep, it chooses a contribution according to Algorithm 1.

Algorithm 1 Baseline Donation Generation

```
function GENERATECONTRIBUTION(P, F, C, O, T)
    if P < 0.5 then
        return 0
    else
        needed = getNeededContribution(P, C, O, T)
        if needed > F then
            return 0
        else
            return needed

function GETNEEDEDCONTRIBUTION(P, C, O, T)
    currentPct = C/(C + O)
    neededPct = (0.5 − C)/T
    R = currentPct + neededPct
    return (C − CR − OR)/(C − 1)
```

Essentially, the donor assumes that the current poll reflects the true level of support and donates the minimum amount required to swing the election in the preferred candidate's favor, if doing so is necessary and possible given the donor's remaining funds. Note that while this is a simple algorithm, it is also nearly optimal for this simplified version of the donation problem. The algorithm will quickly contribute enough to swing the election and will on average not waste funds on a race that is already won or unwinnable.

POMCPOW. We implemented the POMCPOW algorithm described in Sunberg and Kochenderfer, 2018 [9].

Implementation details:

1) When choosing an action (i.e., in ACTIONPROGWIDEN), we compute $X =$ GETNEEDEDCONTRIBUTION (assuming that our estimated support is the true support) and restrict the range of possible donations to $[0, 1.5 * X]$. This heuristic reduces the number of actions we must consider to hone in on a good strategy.
2) Contributing 0 is always considered as a possible action.
3) We found that tuning the c parameter (which controls exploration/exploitation) was difficult because the appropriate value varied depending on the number of rounds remaining. Therefore, we select actions with probability proportional to their values:
$$P(\text{contribute C}) = \frac{X + Q(C)}{\sum_{a \in A} X + Q(a)}$$
Where X is a parameter that flattens the probability distribution. We set X to 0 for the experiments with one race and to 0.01 for the multi-race experiment.
4) Our ROLLOUT policy is the same as the baseline policy.

PFT-DPW. We also implemented the PFT-DPW algorithm described in Sunberg and Kochenderfer, 2018 [9]. For our particle filter belief update algorithm, we used the weighted particle approach described in Algorithm 6.3 of the textbook [10].

[4]Figures are located in the Appendix.

TABLE I: Hyperparameters for Continuous Experiments

	POMCPOW	PFT-DPW
N	10000	1000
k_a	30	20
α_a	1/30	1/25
k_o	5	8
α_o	1/100	1/85
m	N/A	500

TABLE II: Default Parameters for Experiments

Parameter	***Discrete Model***	***Continuous Model***
support	0.5	0.5
win reward	10	500
lose penalty	0	0
time steps	10	10
initial budget	10	1000
transitions	deterministic	deterministic

As hyperparameter tuning was not a priority, we adopted the parameters used by Sunberg and Kochenderfer on their VDP Tag experiment, which–like our problem–involves continuous state, action, and observation spaces. The only exceptions are the c parameter, which we removed (see above), and the m parameter, which we increased to improve the accuracy of belief updates. See Table I for details.

IV. Experiments

Table II shows the default parameters used for our experiments. We tried to make them similar for the discrete and continuous definitions of our problem. Since the discrete model is on a smaller scale than the larger ones, there are some differences. We made sure the reward function balanced the consequences of spending money and winning races so that our agent is incentivized to produce some actions.

We investigated how initial support impacts our donor's actions, to see if our agent learns to avoid spending too much in cases where its money would not affect the outcome of the race, that is, when initial support is very low and when it is very high. We also examined model behavior when varying the maximum reward for winning, w_r and the penalty for losing, l_r.

In our default case, the transition from one state to the next is direct and deterministic. Accordingly, we experiment with modifications to our model to accommodate uncertainty in the transition function.

Finally, for the continuous problem, we investigate the inclusion of multiple races to see if the donor can effectively allocate resources among them: i.e., whether our models can generalize to a more complex and realistic state space.

V. Results

A. Discrete Model

When varying initial support, the total score, (i.e., the sum of rewards at each time step) for the QMDP model is consistently the same or higher than for the constant zero policy and consistently higher than for the random policy (Figure 3a). As seen in Figure 4a, the QMDP policy learns to take no actions at any time step when it is unable to influence the outcome of the race, such as when the initial support for the preferred candidate is too low (0-2) and winning is impossible, or when it is too high (6-10) and winning is guaranteed. For initial supports ranging from 3 to 5, the donor contributes a non-zero amount as early as doing so provides a positive reward. The donor usually does not contribute as early as possible because the rewards for winning in earlier timesteps usually do not outweigh the cost of donating. (As explained above, our reward function assigns a greater reward to winning during later timesteps.) Thus, although contributing early is always superior to contributing the same amount later (i.e., because the donor pays the same amount but accumulates more rewards), QMDP's short-term planning prevents us from learning this behavior.

When we increase the reward for winning, w_r, our model better achieves close to the maximum total score possible over all time steps (Figure 5a). This is because a higher w_r means that the reward for winning outweighs the cost of spending sooner, so our QMDP donor is more willing to donate earlier (Figure 4b). Likewise, increasing the losing penalty l_r impedes our model from achieving the maximum total score, because the donor incurs higher penalties when it is losing but (because the win reward is unchanged), still does not contribute until relatively late in the race. In Figure 3b, Nonetheless, QMDP performs better or comparable to the two baselines when the losing penalty is increased to 20.

For the discrete model, with transition uncertainty included, this uncertainty overpowered any incentive to donate with the default winning reward of 10, so our agent never donates. Even when the winning reward is inflated up to 50, only on occasion would the model make small contributions. When this POMDP with uncertain transitions was simulated for 10 time steps, sometimes the QMDP policy performed better than random and sometimes it did not. Thus, QMDP does not do very well when transitions are not deterministic. We achieved greater success at handling transition uncertainty in the continuous problem (see below).

B. Continuous Model

1) Basic Model: Figure 6a shows the scores achieved by each model as a function of the initial support level. The values for the baseline and POMCPOW models are averaged over 50 trials, and the values for the PFT-DWP model are averaged over 5 trials. The maximum possible scores is $\approx$ 2900 (assuming that no contribution is required); all three models preform very well. (Note that when the starting support is 0, the race is unwinable, so the appropriate strategy is to contribute nothing.) The POMCPOW model essentially learns to behave like the baseline model, while the PFT-DWP model is a more aggressive donor at low support levels and a more conservative donor at high support levels. This means that the PFT-DWP model under-preforms when support is slightly less than 0.5 and over-preforms when support is slightly above

0.5. Figures 6b and 6c shed more light on the behavior of the models. (The baseline model behaves in almost the exact same way as the POMCPOW model.) The POMCPOW model seems to recognize the value of contributing early (and locking in rewards) while the PFT-DPW model is slightly more hesitant to contribute. Overall, these results indicate that both continuous models can learn near-optimal strategies for the basic problem. Given the comparable performances of the two models, because the PFT-DWP model took about 100 times longer to train compared to the POMCPOW model, we restricted our other experiments to the POMCPOW and baseline models.

2) Varying Win Reward and Loss Penalty: Figures 7a-7c present the results of varying the winning reward w_r and the losing penalty l_r. Note that these scores were generated by averaging 5 trials rather than 50 and, accordingly, are noisier than those from the previous experiment. In general, both models preform well regardless of the reward or penalty. However, the POMCPOW model preforms increasingly well compared to the baseline as the win reward shrinks and as the lose penalty grows; this reflects the fact that because the POMCPOW model takes rewards into account (unlike the baseline model), it better recognizes when a potentially race-swinging contribution is too expensive (i.e., because the value from winning is less than the amount of the contribution). This illustrates one of the key advantages of modeling the problem as a POMDP.

3) Transition Uncertainty: Our basic model assumes that the donor knows T, the parameter that controls the effect of a contribution on future support. In a more realistic model, however, this parameter is unknown at the start of a race and must be estimated over the course of the election. Because T is unknown, the solver does not know the transition function; becasue POMDP models assume knowledge of the transition function, we adopted a 2-step process for making donations with this type of transition uncertainty. [5] On each round of the election (except the first), we update our belief about T based on our most recent contribution and the new polling data we received and run the POMCPOW model to generate a contribution based on our new belief about T. To avoid over-contributing in the first few rounds (when T is still uncertain), we modify our reward by multiplying it with the conf parameter, which ranges from 0 to 1 and reflects our confidence in our current belief about T. Algorithm 2 presents an outline of the new contribution process. H refers to the history of contributions and polls, T is initialized to 0.75 and conf is initialized to 0.01.

To update our belief about T, we simply calculate the average transition effect over the past timesteps in the race:

$$T = \left(\sum_{i=0}^{n-1} \frac{P_{i+1}}{M_i} \right) / (n-1)$$

where P_i is the poll at timestep i, M_i is the money share for the candidate at t_i, and n is number of time steps that have occurred so far. (Note that we cannot update our belief on the first timestep because we have not yet contributed.) Our confidence increases linearly with each round, beginning at $1/n$ and ending at 1. We also experimented with using a weighted particle filter to update our belief about T but found that this approach converged slower while taking longer to run.

When running experiments with transition uncertainty, we also adjusted the basic model in two ways. First, to reduce variance in the transition function, we sample values of T from a normal distribution with mean 0.75 and standard deviation 0.3. Second, to provide the model long enough to build up confidence in T, we increased the number of timesteps to 100.

Algorithm 2 Behavior with Transition Uncertainty

```
election = initializeElection() ▷ chooses a true value of T
for i ∈ N do
    poll = generatePoll(election)
    if i ≠ 0 then
        T, conf = updateTransitionBelief(H)
    contribution = POMCPOW(poll, H, T, conf)
```

Figure 8 presents the scores of the baseline and POMCPOW models with the introduction of transition uncertainty. The baseline values are averaged over 50 trials, and the POMCPOW values are averaged over 5. The POMCPOW model consistently achieves higher average scores than the baseline model (which operates with $T = 0.75$). This demonstrates that the POMCPOW model is withholding donations until it is has sufficiently high confidence in the transition function and illustrates that our 2-step method for addressing transition uncertainty is promising.

TABLE III: Scores from Multiple Races

Model	*Average Score*
Baseline	1476.4
POMCPOW	1599.8

4) Multiple Elections: As a final evaluation of the POMCPOW model, we expand the model to cover multiple elections, reflecting a more realistic scenario in which a donor must choose the races in which to allocate funds. Each race is initialized with a different T parameter and initial support level; however, unlike in the prior experiment, the donor knows the T values. The score for a mutli-race is the sum of the scores in its component races.

The baseline model for a multi-race iterates through the individual races (in a random order) and contributes to each according to its standard policy.

The multi-race POMCPOW model follows naturally from the single-race version. States, actions, and observations are

[5]Note that our representations of transition uncertainty vary between the discreet and continuous problems. In the former, the solver has complete knowledge of a non-deterministic version of the transition function $\mathcal{T}$, In the latter, the solver has limited knowledge of the deterministic version of $\mathcal{T}$. These representations correspond to variation in the effect of donations on support within and among races, respectively.

now n-tuples of their respective single-race selves. Accordingly, the action space expands exponentially with the number of races. To compensate for this, we increased the k_α and k_o parameters by a factor of 10.

Table III shows the average score over 10 iterations with each model where $n = 5$. The POMCPOW model achieves on average an ≈ 130 higher score than the baseline.

Figures 9a and 9b illustrate the different behaviors of the two models and help explain the POMCPOW model's higher average score. The baseline model contributes heavily to first races it considers and, accordingly, has few funds remaining to spend on other races. The POMCPOW model does a better job distributing its donations among the races and withholding them until later timesteps when it has more information about the state of each race.

This experiment demonstrates that, compare to the baseline mode, the POMCPOW model can learn to solve more complex problems.

VI. Conclusion

We could expand our model to more closely reflect real campaign funding dynamics in many ways. First, we could make greater use of historical data, increasing the robustness of our models' parameters. Second, we could expand our observations to include more factors that predict a candidate's success, such as incumbency. Third, we could experiment with removing intermediate rewards to better reflect the all-or-nothing outcome of an election. Fourth, we could experiment with better methods of learning the transition function, such as a BAMDP. Fifth, we could explore integrating deep-learning into our model, perhaps as the ROLLOUT function in the POMCPOW model. Sixth, we could factor in the contributions of another donor on the same or opposite side of the race. Seventh, we could consider some of the interrelations between races, such as the potential for donations to one race to affect downballot races in overlapping media markets. And finally, we could investigate reducing the impact of donations over the course of the race, so that the agent must balance earlier, more impactful donations in a relatively uncertain environment against later, less impactful donations when the agent has a clearer understanding of the state of the race.

We demonstrated that POMDP methods can outperform baselines in both discrete and continuous formulations of the donation allocation problem. We found that POMCPOW and PFT-DPW were more forward thinking than QMDP and therefore tended to donate earlier in each race.

Overall, our results demonstrate the effectiveness of a computational, quantitative approach to campaign donation allocation and suggest that the benefits of such an approach, compared to simpler allocation strategies, increase with the complexity–and therefore the realism–of the problem space. Given the enormous sums spent on political campaigns, even minor improvements to allocation efficiency provide significant cost savings. Accordingly, we expect that just as data driven, model-based methods have become increasingly popular as means for targeting voters and for forecasting election outcomes, these approaches will increasingly shape the contribution strategies of parties, interest groups, and large donors.

VII. Author Contributions

We met frequently as a group to discuss how to model our problem and plan work and to make sure that everyone was on the same page. During these work sessions, we would code, debug, and experiment together as a team. Separately, Vivian researched related work, Sawyer implemented continuous POMDP solvers in Python, and Lucy gathered historical data and ran discrete POMDP solvers in Julia.

References

[1] Koerth-Baker, Maggie. How Money Affects Elections. FiveThirtyEight, 10 Sept. 2018, fivethirtyeight.com/features/money-and-elections-a-complicated-love-story/?fbclid=IwAR3q-J8WMuBmBIP7k-oW_XJNfPqjywd_Z2Od1_GXanue-psCfwUqL66c1Ws.

[2] Rakich, Nathaniel. Election Update: How The Latest Fundraising Numbers Shifted Our House Forecasts. FiveThirtyEight, 18 Oct. 2018, fivethirtyeight.com/features/election-update-how-the-latest-fundraising-numbers-shifted-our-house-forecasts/?fbclid=IwAR1hTK49yFfqn7WK8YSDQD_H2BVkM3iHwHcJ6noXysKY4XjDkFjsJELapHc.

[3] McCarthy, M. A., H. P. Possingham, and A. M. Gill. 2001. Using stochastic dynamic programming to determine optimal fire management for Banksia ornata. Journal of Applied Ecology 38:585592

[4] McDonald-Madden, E., P. W. J. Baxter, and H. P. Possingham. 2008. Subpopulation triage: How to allocate conservation effort among populations. Conservation Biology 22:656665.

[5] Shea, K., and H. P. Possingham. 2000. Optimal release strategies for biological control agents: an application of stochastic dynamic programming to population management. Journal of Applied Ecology 37:7786.

[6] McDonald-Madden E, Chads I, McCarthy MA, Linkie M, Possingham HP. Allocating conservation resources between areas where persistence of a species is uncertain. Ecological Applications. 2011;21(3):84458

[7] "Cost of 2018 election to surpass $5 billion, CRP projects." Center for Responsive Politics, 17 Oct. 2018, https://www.opensecrets.org/news/2018/10/cost-of-2018-election/

[8] Kalla, J. and Broockman, D. 2015. Campaign Contributions Facilitate Access to Congressional Officials: A Randomized Field Experiment. American Journal of Political Science, 60: 545-558.

[9] Sunberg, Z.N. and Kochenderfer, M.J., 2018. Online Algorithms for POMDPs with Continuous State, Action, and Observation Spaces. In ICAPS (pp. 259-263).

[10] Mykel J. Kochenderfer, Christopher Amato, Girish Chowdhary, Jonathan P. How, Hayley J. Davison Reynolds, Jason R. Thornton, Pedro A. Torres-Carrasquillo, N. Kemal re, and John Vian. 2015. Decision Making Under Uncertainty: Theory and Application (1st ed.). The MIT Press.

[11] Election Overview. FollowTheMoney.org, www.followthemoney.org/tools/election-overview?s=OH&y=2018.

[12] "Ohio Election Results 2018." Politico, https://www.politico.com/election-results/2018/ohio/

[13] https://fivethirtyeight.com/

[14] https://www.nytimes.com/section/upshot

[15] Wang Sam. "Optimal 2018 donations in the home stretch: Senate, House, Governor." Princeton Election Consortium., 26 Oct. 2018, http://election.princeton.edu/2018/10/26/optimal-donations-in-the-home-stretch-2018/

APPENDIX

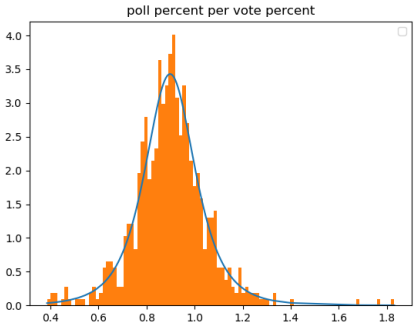


Fig. 1: A histogram of the ratio of polled vote percentage versus actual vote percentage in 2014 U.S. House elections. This plot shows that 1% in the polls typically translates to less than 1% of actual votes.

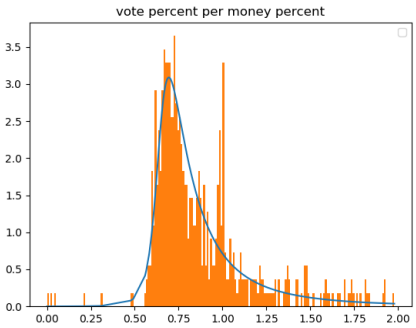
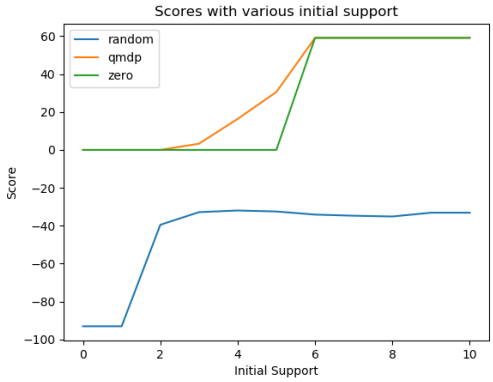
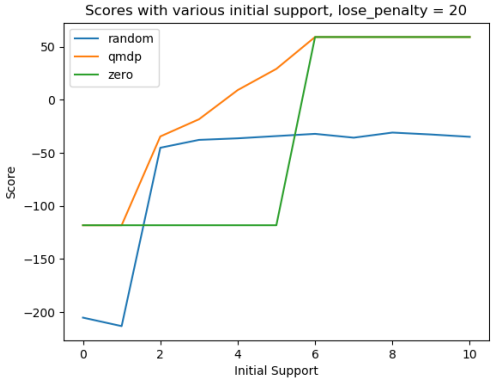


Fig. 2: A histogram of the ratio of actual vote percentage versus a candidate's share of total money raised during the race. This plot shows that 1% of votes in a race typically translates to about 0.75% of money raised by a candidate.

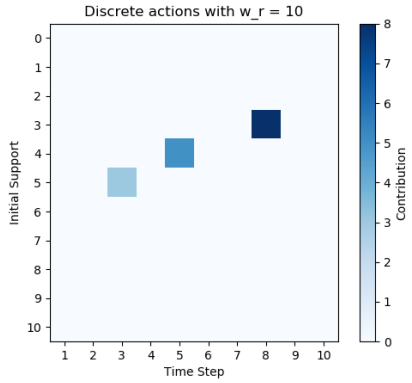


(a) Results with default parameters.

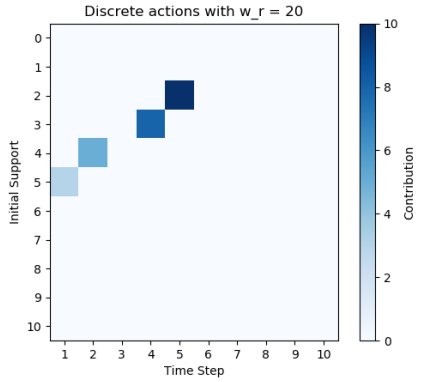


(b) Results with a non-zero lose penalty.

Fig. 3: The total score for the discrete model averaged over 5 trials for three different policies: QMDP, random, and constant zero. The starting support for the preferred candidate of each round is along the horizontal axis.

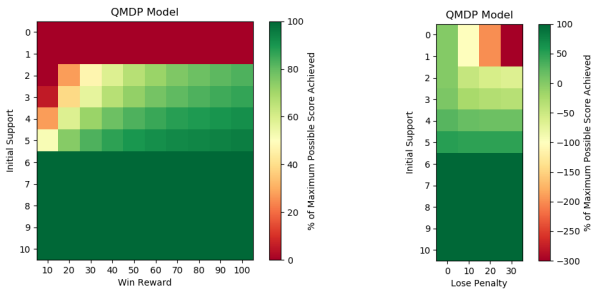


(a) Policy with default parameters.



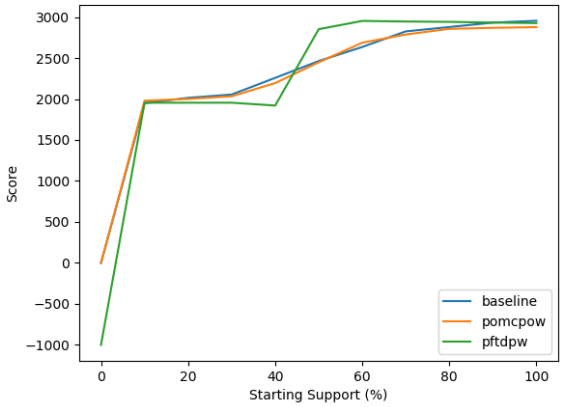
(b) Policy with higher win reward.

Fig. 4: The actions taken by the donor agent operating with the QMDP policy at each time step (x axis), with varying initial candidate support (y axis).

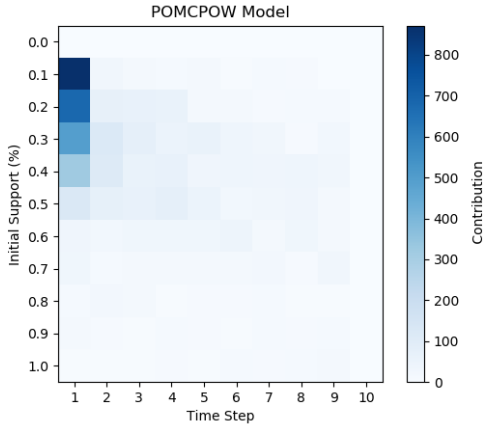


(a) (b)

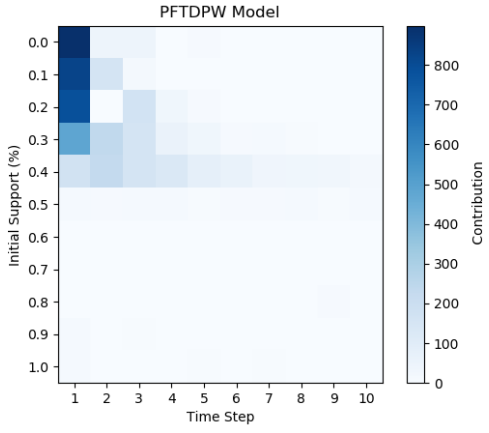
Fig. 5: The percent of maximum award achieved with various initial support values and reward functions. The left heatmap varies the win reward, w_r, while the right rightmap varies the losing penalty, l_r.



(a)

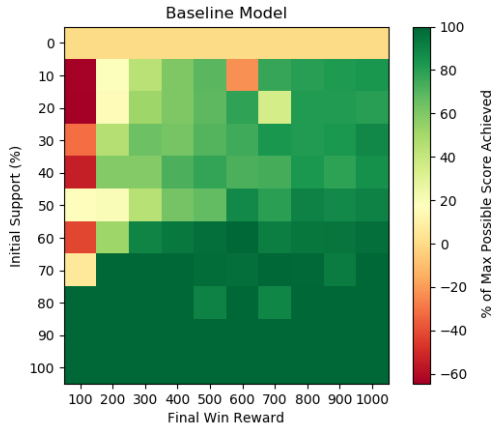


(b)

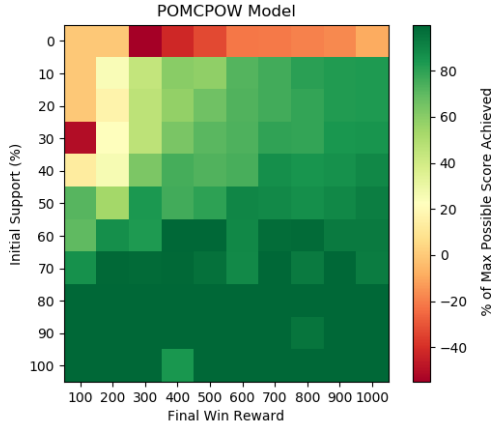


(c)

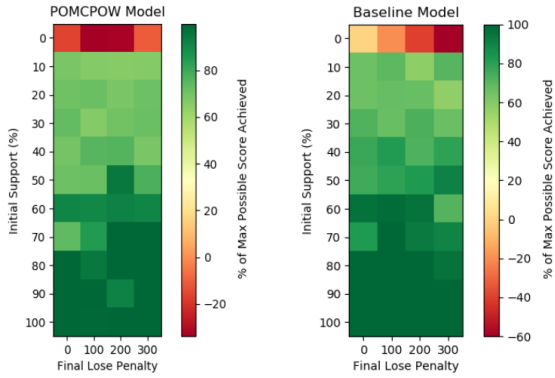
Fig. 6: Basic continuous model – note that the POMCPOW and PFT-DWP models achieve comparable scores to the baseline model. The POMCPOW model contributes sightly more aggressively.



(a)



(b)



(c)

Fig. 7: Varying the reward function – note that the POMCPOW model performs better (relative to the baseline model) when the win reward is small and when the lose reward is large.

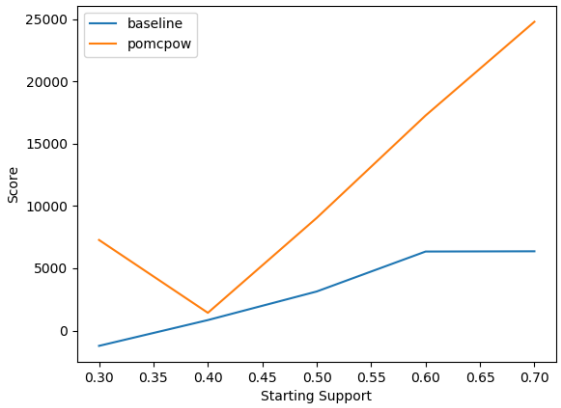
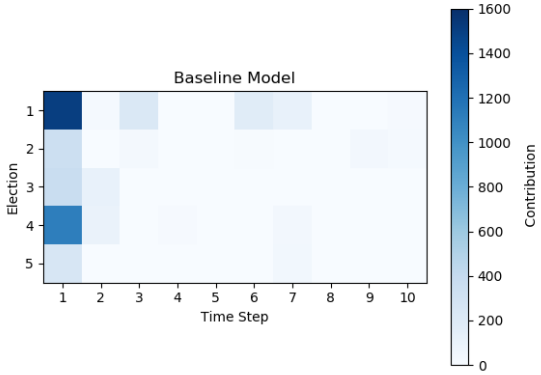
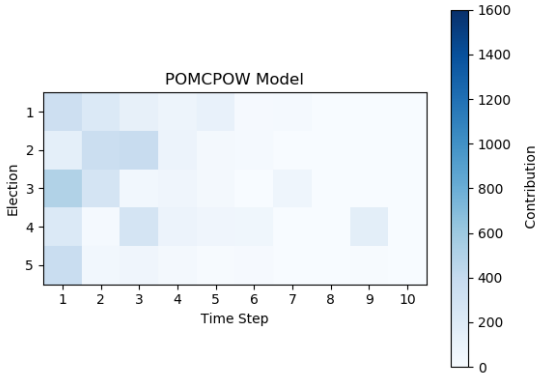


Fig. 8: With transition model uncertainty, the POMCPOW model constituency outperforms the baseline model.



(a)



(b)

Fig. 9: Contributions when contributing to multiple elections. Note that the baseline model contributes heavily to the first race it considers, while the POMCPOW model distributes its funds more evenly.