Polya's Bees

Decentralized Decision Making With Quorum-Based Strategies

Russell Golman[1], David Hagmann*[1], and John Miller[1,2]

[1] Department of Social and Decision Sciences, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213, USA

[2] The Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM 87501, USA

Abstract. From ants [1] and bees [2] to neurons in the brain [3], decentralized systems of agents are capable of making critical choices. Here, we propose a simple model of a two-part process that decentralized systems use to make decisions. We use an urn scheme to capture the first part of the process, in which individual agents randomly search over the set of feasible choices, biased by the quality of the choices revealed during previous searches. We assume that the urn scheme runs only until a threshold is hit in order to capture the second part of the process, in which a final choice is triggered when the system senses a quorum of agents investigating any one particular choice. We find that the combination of these two elements results in a robust and effective means by which a decentralized system can make good choices. We further apply our model to sites with multiple attributes and show that the decision rule leads to risk-averse decision making. Finally, we model disruptions in the recruitment process, as are observed in nature, and show that this mitigates the risk of information cascades.

Keywords: decentralized decision-making, agent-based modeling, quorum rules, bees, ants

1 Introduction

There are a variety of natural systems that have demonstrated the ability to productively employ a decentralized decision mechanism. Ants and bees appear to rely on decentralized decision making for critical choices [4]. For example, in choosing a new nest site – a decision that has huge implications for the survival of the group – good decisions must be made without central control and with no single individual evaluating the total available information [5] or any one individual making direct comparisons of the available options [6]. Thus, even though individual agents follow simple rules that allow them to uncover very limited and local information, the colony as a whole is able to seamlessly and efficiently integrate the resulting flow of information into a high-quality, final

* Corresponding author: hagmann@cmu.edu

decision [7]. The process applies similarly to bees [8]. Indeed, the primate brain may function analogously to a colony of social insects [9].

Our model abstracts beyond any single one of these systems and aims to provide a deeper understanding of how a decentralized decision mechanism operates in order to generate insights that apply across many of these natural systems. Moreover, the fact that many natural systems independently evolved similar decentralized decision mechanisms suggests that such mechanisms may represent a robust solution to the general problem of making good, group-level decisions in the absence of centralized control. Our analysis is informed by previous simulations that have revealed that effective collective decision making can emerge from the aggregation of individual agents obeying simple rules of behavior [10]. Our focus is on the nature of the decision mechanism and its consequences for the speed and accuracy of the collective decision.

Any decision mechanism must trade off exploring new choices with exploiting the best choice known to date. While further exploration might reveal superior choices, it comes at the cost of not picking and acting on a known choice. Thus, too much exploration may lead to indecisiveness and thus harm fitness, while too little may imply the acceptance of sub-optimal choices. There is evidence that ants adapt their quorum threshold rule in response to the urgency of the need to abandon their old nest [11]. The tradeoff between speed and accuracy in decentralized decision making has been studied in the diffusion model of decision making [9]. The diffusion model shares much in common with our urn model, including a mechanism to integrate local information without centralized processing, but it does not incorporate positive feedbacks in the accumulation of evidence for a given choice.

2 The Model

2.1 Search and Recruitment

Agents (e.g., scouts) explore one of many possible choices (e.g. nectar sources for honey bees or nest sites for ants) and then return home to recruit additional agents to further explore the choice they have investigated. A Poisson process for each agent governs its return home, and the recruitment of additional scouts corresponds to rewards based on the value of the explored choice.

We assume that there are C potential choices, and that each of these choices, c, has some number of agents w_c^t investigating it at any time t. We refer to w_c^t as the weight on choice option c at time t. All weights are initially set to the same, positive value w^0. Each agent is equally likely to return home at any time, and when one does, it recruits additional agents to continue exploring the same choice it just investigated. The chance of an agent recruiting for choice c at time t is thus simply proportional to the weight w_c^t. Each choice has a set of immutable attributes that define its "quality," and the extent of recruitment for choice c depends on the quality of that choice. After an agent investigates choice c, it recruits v_c additional agents to continue exploring it, increasing the weight on this choice by v_c. We think of v_c as a measure of the quality of choice c.

2.2 Quorom Detection

The search process above generates a distribution of agents investigating each possible choice at any given time. Given the decentralized nature of these systems, there must be some feasible trigger that ends the search process and finalizes the choice. One possible solution to this problem would be to have the search probabilities converge on a consensus, that is, have all of the probability concentrated on a single choice. Forcing such a consensus on the system is problematic. First, such convergence may be extremely slow, in which case the system may not be able to make a decision in time to avoid a serious loss of fitness. Second, there would need to be some plausible, decentralized mechanism by which the system as a whole could recognize that such a convergence has occurred, and such a mechanism is not obvious. Finally, we have empirical evidence, at least in the case of honey bees [2], that consensus is not what triggers a choice. Instead, in the case of honey bees, an irrevocable choice seems to be made once the number of scout bees at a particular site reaches a quorum.

Based on the above arguments, we incorporate into our model a *quorum threshold*, τ, that triggers as the final decision any choice that is being investigated by at least that number of agents. The level of the quorum threshold has important implications for the decisions that arise in the system. If the threshold is set too high, then a quorum may not be reached for a long time, resulting in prolonged inaction. If the threshold is set too low, then a quorum might be achieved for a suboptimal choice. Thus, the optimal choice of the quorum threshold depends on a critical tradeoff between speed and accuracy in the decision making process. From a normative standpoint, a good threshold allows the system to withstand various transients in the probability distribution over time while still remaining responsive to the acquired information in a timely manner.

2.3 The Urn Scheme

We use a simple Polya urn process to model this decision mechanism. This process is easy to visualize. Assign to each of the C choices a color, and place w^0 balls of each color into an urn. The number of balls of a particular color in the urn corresponds to the number of agents investigating the associated choice option. Each ball has the same rate at which it may be randomly drawn from the urn. When a ball with color c is drawn, it is immediately placed back into the urn along with v_c identically-colored balls. This process continues until a threshold number of balls τ is reached. We want to understand how the mix of balls evolves over time.

3 Computational Results

To explore the effects of parameter variation and the introduction of noise into the process, we run computational simulations of the proposed mechanism using

Python.[3] While the process described above runs in continuous time, we can identify discrete time steps marked every time an agent returns home to recruit (i.e., every time a ball is drawn from the urn). Let the index μ count the number of agents that have returned home for a visit, and denote the time when the μ^{th} agent returns home as t_μ. When there are w agents exploring the set of possible choices, the expected time until the next agent returns home is $\frac{1}{w\lambda}$. Setting $\lambda = \frac{1}{Cw^0}$, which amounts to normalizing the units of time so that $\mathbf{E}[t_1] = 1$, we have

$$\mathbf{E}[t_{\mu+1} - t_\mu \mid \sum_c w_c^{t_\mu} = w] = \frac{Cw^0}{w}.$$

Each simulation reports the average time until decision $T(\tau)$ and probability of optimal choice $p_{c^*}(\tau)$ as a function of the quorom threshold τ, based on $100,000$ trials.

3.1 The Baseline Model

We first establish a baseline, in which all targets initially have one agent recruiting. The rate at which new agents join is 2 for the single high quality target and 1 for one or more low quality targets. Figure 1 shows the share of outcomes in which the high quality site is the first to reach the threshold as a function of the threshold and a function of continuous time.

We first observe in Figure 1 (top) that the process selects the high quality target at a significantly higher rate than pure chance even for a very low threshold of 5. The greatest marginal improvements occur early on, up to a threshold of 50 with two targets. Increasing the number of potential targets decreases the likelihood of picking the optimal site. With high thresholds, however, adding more targets has only a very small effect. Even for small thresholds, the difference is not nearly as big as we would expect by chance. This suggests that the decision algorithm converges quickly and that it effectively deals with multiple unattractive options. Plotting the baseline as a function of time (Figure 1, bottom) shows that adding more targets increases the time required to reach a given accuracy of picking the high quality target (vertical distance). The additional time required appears to increase linearly in the number of targets.

3.2 Disrupting Recruitment

We now introduce some probability (p_D) with which recruitment for all sites is disrupted at discrete time intervals. In Figure 2, we show simulation results for $p_D = \{0.05, 0.10, 0.15, 0.20\}$. Adding some chance of disruption makes it less likely that the mechanism settles on the optimal target. For $p_D = 0.20$, higher thresholds lower the likelihood of choosing the optimal target, because a growing number of simulations fails to achieve the quorum and no decision is made. These results suggest that there is no benefit to disrupting agents' recruitment efforts given the baseline assumptions.

[3] The source code – along with the simulation data and the R code to produce all figures in this paper – is available at `http://www.dhagmann.com/presentations/`

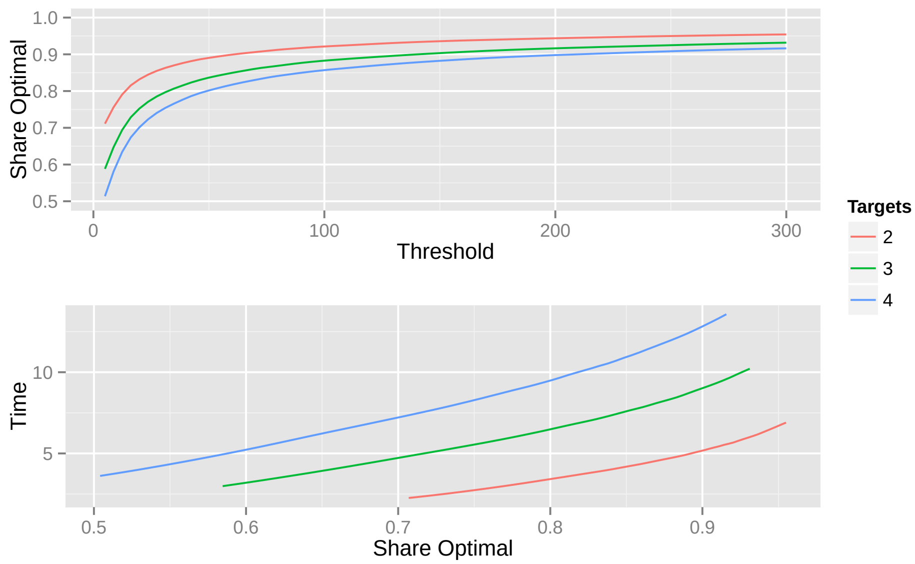

Fig. 1. Share of optimal decisions with 2, 3, and 4 targets. The high quality site recruits 2 new agents, the low quality sites recruit 1.

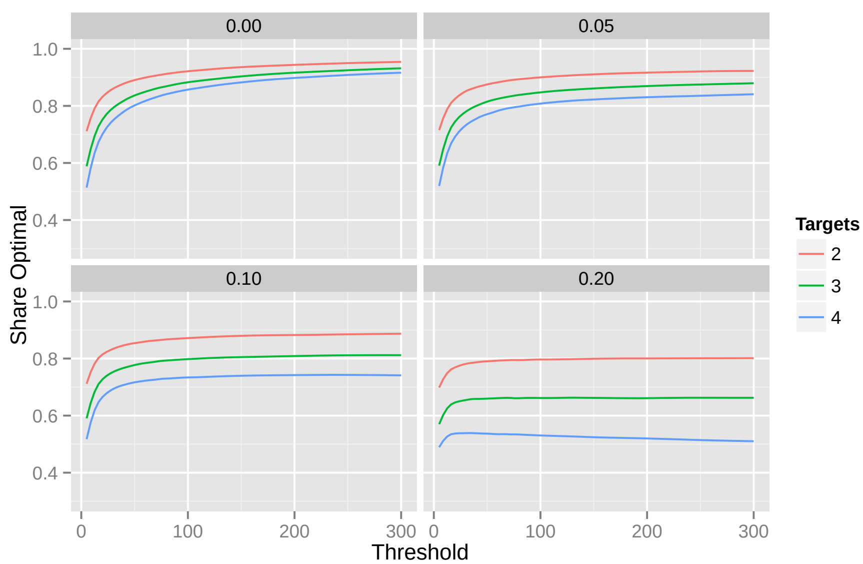

Fig. 2. Share of optimal decisions with 2, 3, and 4 targets, and a probability of disruption of 0, 0.05, 0.10, 0.20

3.3 Changing Quality Differentials

The high quality target has so far been twice as attractive as the low quality targets. We now introduce a medium quality target and set the number of agents recruited for the three qualities as follows: an agent recruiting for a high quality target adds 3 new agents upon its return, an agent recruiting for a medium quality site adds 2 new agents, and an agent recruiting for a low quality site adds 1 new agent. Figure 3 shows the likelihood of picking the high quality target if there is one high quality target and two low quality targets, vs. one high, one medium, and one low quality target.

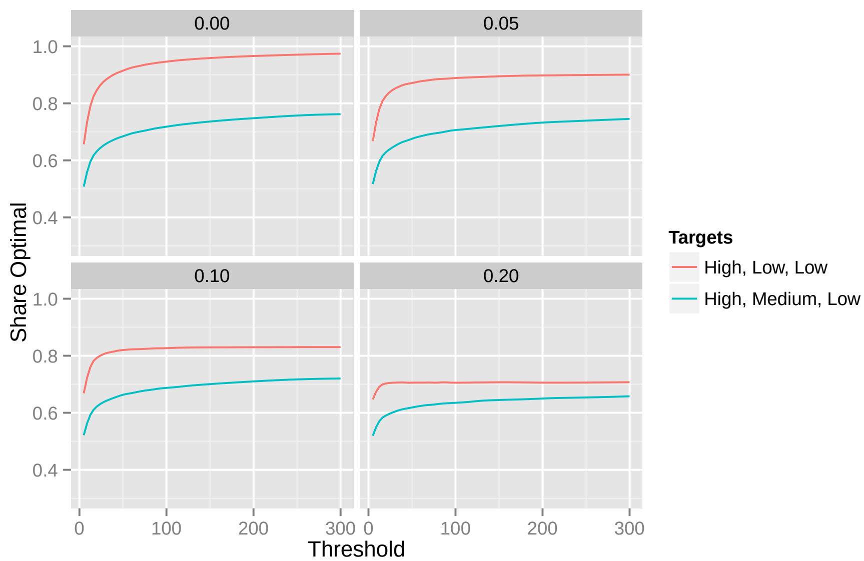

Fig. 3. Likelihood of selecting a high quality site when the choice set consists of a high quality and two low quality sites vs. one high, one medium, and one low quality site.

Introducing a medium quality site significantly reduces the likelihood that the high quality target is selected. Surprisingly, increasing the threshold does not diminish this difference and does not allow the high quality site to catch up. We again observe that disruption reduces the likelihood of picking the high quality target. However, the magnitude of the impact is considerably smaller if there is a medium quality site.

3.4 Noisy Information

Until now, each target increased the number of agents recruiting by a fixed amount. We now introduce some noise into the process and compare the algorithm's performance with a condition without noise. The simulation is initialized with one high quality target and two low quality targets. In the condition without noise, the rates of increase are 2 and 1, respectively, for the high and low quality targets. In the condition with noise, the high quality target has a 25% chance of adding 3 agents, a 50% chance of adding 2 agents, and a 25% chance of adding 1 agent. The low quality targets have a 25% chance of adding 2 agents, a 50% chance of adding 1 agent, and a 25% chance of adding no agent. Figure 4 compares the likelihood of picking a high quality in the two conditions. Introducting noise decreases slightly the likelihood of choosing the high quality target, consistently across all thresholds. As before, introducing disruption decreases the rate with which the desirable site is chosen. However, disruption has no effect on the relative performance of noise vs. no noise.

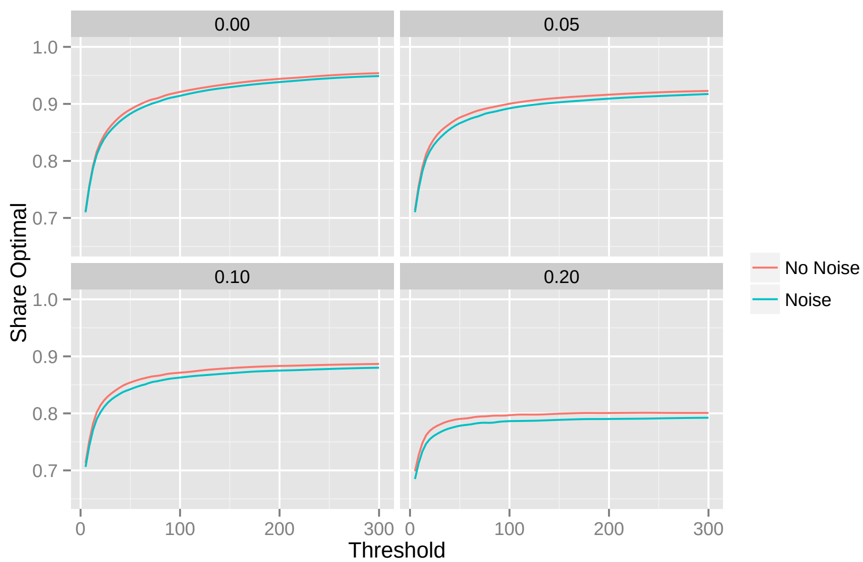

Fig. 4. Likelihood of selecting a high quality site with noise vs. no noise.

3.5 Multiple Attributes

We now consider target attributes along two dimensions. A particular site may be of high quality in one attribute and of low quality in another or it may be of average quality in both. Of interest is whether the decision algorithm is more likely to select a site with a high quality and a low quality attribute (the "risky" target) with the same expected value as a site with two attributes of the same quality (the "safe" target). An agent recruiting for the risky target will add either 3 agents or 1 agent with equal probability. An agent recruiting for the safe target always adds 2 agents. We also add a low quality target that always adds 1 agent. The likelihood of picking the risky and the safe target are shown in figure 5. The results show that the target with average attributes is more likely to reach any given threshold before does the target with a high and a low quality attribute. Moreover, adding disruption does not favor either type of choice.

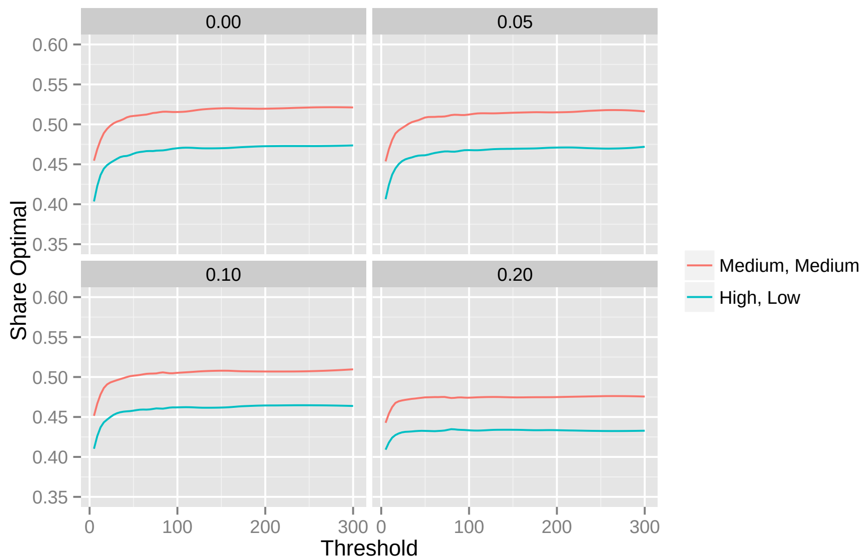

Fig. 5. The decision algorithm is more likely to select a target with average attributes than one in which one attribute is high and another is low.

3.6 Information Cascades

Finally, we test the robustness of the decision algorithm to making mistakes early on. In particular, we can imagine an information cascade in which a low quality site is selected early on and ends up gaining a lead that the high quality site cannot catch up to. We simulate this by setting the number of agents initially recruiting for a high and low quality site to 1 and 3, respectively. The high quality site again recruits 2 new agents upon selection and the low quality site recruits 1 new agent. Figure 6 shows the likelihood that the high quality target is selected using various rates of disruption.

We now see a benefit of disrupting recruitment efforts. Given a low threshold and an initial advantage to a low quality site, introducing the possibility of disrupting agents and dropping them from the decision process allows the system to self-correct. As the threshold used increases, the optimal rate of disruption decreases. With large thresholds ($T > 100$), the process again does best without

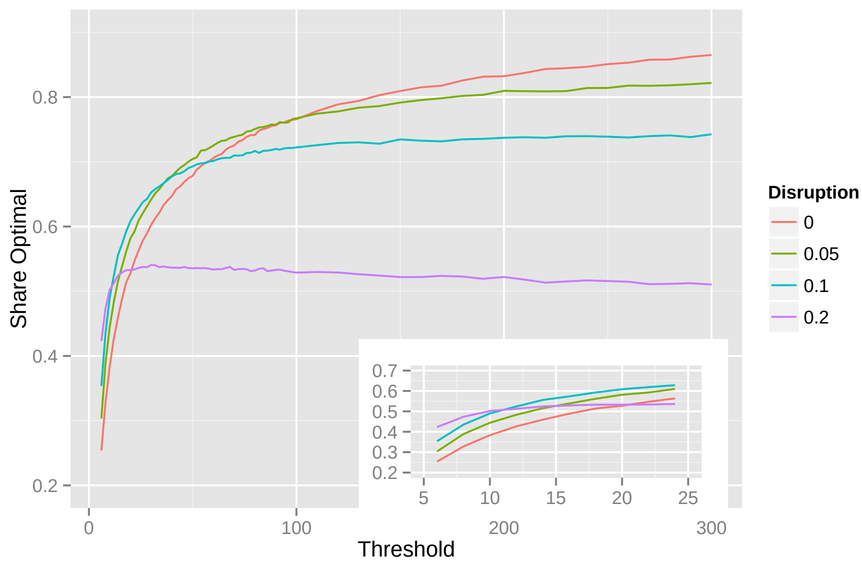

Fig. 6. Likelihood of picking high quality target when there is one high quality and one low quality target, starting off with 1 and 3 agents, respectively, recruiting for it.

disruptions. This suggests that for small groups, where high thresholds are not achievable, may do best by introducing a process to reset some agents.

4 Conclusion

The natural systems we have outlined appear to have found a low-cost, efficient, and robust mechanism by which a decentralized organization can make key decisions. Modeling the decision process using a Polya urn process allows us to replicate the quorum rule and incorporate positive feedbacks. This decentralized decision mechanism may have useful applications in the design of new social and artificial systems. Novel applications for such a mechanism range from improving human organizations to applying such techniques to artificial systems like algorithmic search and the control of swarms of robots or networked computers.

References

1. Gordon, D.M.: Ant Encounters: Interaction Networks and Colony Behavior. Princeton University Press (March 2010)
2. Seeley, T.D.: Honeybee Democracy. Princeton University Press (September 2010)
3. Gold, J.I., Shadlen, M.N.: The neural basis of decision making. Annual Review of Neuroscience **30**(1) (July 2007) 535–574

4. Conradt, L., Roper, T.J.: Consensus decision making in animals. Trends in Ecology & Evolution **20**(8) (August 2005) 449–456 WOS:000231232900013.
5. Mallon, E., Pratt, S., Franks, N.: Individual and collective decision-making during nest site selection by the ant leptothorax albipennis. Behavioral Ecology and Sociobiology **50**(4) (September 2001) 352–359
6. Jeanson, R., Deneubourg, J.L., Grimal, A., Theraulaz, G.: Modulation of individual behavior and collective decision-making during aggregation site selection by the ant messor barbarus. Behavioral Ecology and Sociobiology **55**(4) (February 2004) 388–394
7. Pratt, S.C., Mallon, E.B., Sumpter, D.J., Franks, N.R.: Quorum sensing, recruitment, and collective decision-making during colony emigration by the ant leptothorax albipennis. Behavioral Ecology and Sociobiology **52**(2) (July 2002) 117–127
8. Seeley, T.D., Kirk Visscher, P.: Group decision making in nest-site selection by honey bees. Apidologie **35**(2) (March 2004) 101–116
9. Marshall, J.A.R., Bogacz, R., Dornhaus, A., Planqu, R., Kovacs, T., Franks, N.R.: On optimal decision-making in brains and social insect colonies. Journal of The Royal Society Interface **6**(40) (November 2009) 1065–1074 PMID: 19324679.
10. Couzin, I.D., Krause, J., Franks, N.R., Levin, S.A.: Effective leadership and decision-making in animal groups on the move. Nature **433**(7025) (February 2005) 513–516
11. Pratt, S.C., Sumpter, D.J.T.: A tunable algorithm for collective decision-making. Proceedings of the National Academy of Sciences **103**(43) (October 2006) 15906–15910 PMID: 17038502.
12. Balaji, S., Mahmoud, H.M.: Exact and limiting distributions in diagonal plya processes. Annals of the Institute of Statistical Mathematics **58**(1) (March 2006) 171–185

Appendix: Analytical Results

We can describe this Polya process with a diagonal $C \times C$ matrix with the v_c values along the diagonal. The evolution of the number of balls of a given color is independent of the evolution of other colors (until the threshold is hit). Thus, we can use existing results to obtain the distribution of balls of each color at any time t (in the absence of a threshold for stopping the process). The moment generating function $\phi_c(t,s) = \mathbf{E}\left[e^{s\,w_c^t}\right]$ is given by $\phi_c(t,s) = \left(\frac{e^{v_c(s-t)}}{e^{v_c(s-t)} - e^{v_c s} + 1}\right)^{\frac{w^0}{v_c}}$ (see Lemma 3.1 of [12]). In principle, this moment generating function fully characterizes the distribution of weights w_c^t. In practice, however, calculating the likelihood of hitting a threshold τ at a given time t is complicated.

An asymptotic result is simple enough to obtain. Suppose the threshold τ is infinite so that the Polya process can run forever. Eventually, almost all of the weight converges on the choice with the highest quality. As $t \to \infty$, $\frac{w_c^t}{e^{v_c t}} \xrightarrow{\mathcal{D}} \text{Gamma}\left(\frac{w^0}{v_c}, v_c\right)$ (see Theorem 3.1 of [12]). Thus, if there is a unique optimal choice $c^* = \arg\max_c v_c$, then

$$\lim_{t\to\infty} \frac{w_c^t}{\sum_j w_j^t} = \begin{cases} 1 & \text{if } c = c^* \\ 0 & \text{otherwise.} \end{cases}$$

While the asymptotic properties of the urn process are informative, as discussed above, feasible decentralized systems must make decisions in a finite amount of time. Let λ denote the intensity of the Poisson process for each agent's return home. We have an Exponential(λ) distribution for the time until a given agent returns home, and thus at any time t we have an Exponential($w_c^t\lambda$) distribution for the time until additional agents are recruited to explore choice c. Thus, the time until the number of agents exploring choice c hits the threshold τ is the sum of independent exponentially distributed variables with arithmetically increasing parameters. That is, this time $T_c(\tau)$ has the Hypoexponential($\lambda_0, \lambda_1, \ldots, \lambda_n$) distribution with

$$\lambda_i = \left(w^0 + iv_c\right)\lambda \text{ for all } i \tag{1}$$

and

$$w^0 + nv_c < \tau \le w^0 + (n+1)v_c. \tag{2}$$

Equation (2) implies

$$n = \text{ceiling}\left(\frac{\tau - w^0}{v_c}\right) - 1. \tag{3}$$

The hypoexponential density (pdf) is $f(t) = \sum_{i=0}^{n} C_{i,n}\lambda_i e^{-\lambda_i t}$ with $C_{i,n} = \prod_{j\neq i}\frac{\lambda_j}{\lambda_j - \lambda_i}$. Taking n and $\lambda_1, \ldots, \lambda_n$ to be functions of c and τ given by Equations (1) and (3), this gives us the density $f_{c,\tau}(t)$ for each $T_c(\tau)$.

We can use the density functions for the $T_c(\tau)$ variables to get at the quantities of interest in the system. The time until a decision is made by the decentralized system is $T(\tau) = \min_c T_c(\tau)$. The probability that the eventual decision is for choice c is $p_c(\tau) = \Pr\left[T_c(\tau) < \min_{c'\neq c} T_{c'}(\tau)\right]$.