Supplementary Information for

Dynamic Behavioral Model Uncovers Conditions for Administrative Bloat

Contents

1 Methods

1.1 Determining equilibria and stability

Recall that the dynamics of our model (under general resource allocation) are characterized by the following system of equations:

$$\frac{dR_u}{dt} = \frac{(w - w_A)c}{k_c} - r_d R_u \tag{1}$$

$$\frac{dR_o}{dt} = r_d R_u - \frac{(w - w_A)p}{k_p} \frac{R_o}{R_u + R_o} \tag{2}$$

where $w_A = k_A(R_u + R_o)$. Further recall that c and p are later adjusted to be functions of R_u and R_o under the behavioral allocation scheme.

Determining the equilibrium positions under the general allocation scheme is fully tractable. We set the derivatives equal to zero and solving provides a solution corresponding to administrative bloat and solution corresponding to a functional administration level. The administrative bloat solution is the pair $R_u = 0$ and $R_o = w/k_A$. The functional administrative solution has the form:

$$R_u = \frac{wc(k_c p - c k_p)}{k_c(c k_A p - c k_p r_d + k_c p r_d)} \tag{3}$$

$$R_o = \frac{w k_p c^2}{k_c(c k_A p - c k_p r_d + k_c p r_d)} \tag{4}$$

In the behavioral allocation scheme there is no apparent closed-form solution, so we determine the equilibria using a numerical solver (in our case, we use SciPy's fsolve). Regarding the stability of these solutions, we analytically determine the relevant second derivatives, evaluate the Jacobian matrix of the system at the equilibrium positions, and then numerically solve for the eigenvalues of the system's Jacobian.

Interestingly, we find that the stability of the administrative bloat solution has a simple form which allows us to understand the critical transition at $k_C = k_P$. Accordingly, we address that in Section 1.3.

1.2 Simulation

Given the system of differential equations governing the system's dynamics, we use Euler's method to simulate in discrete time steps. We use the values of R_u and R_o at each time step to determine the firm's performance. We then consider what choice of decision parameters, which we will refer to as the "management strategy", results in the highest performance.

To consider the optimal management strategies, we assume that the utility value being optimized ($\mathcal{U}^O$) is equal to the utility of the firm ($\mathcal{U}$) exponentially discounted over time (i.e., $\mathcal{U}_t^O = \mathcal{U}e^{-rt}$) at some constant rate r. We then use a maximization algorithm (in this case, SciPy's minimize function) to determine the management strategy which maximizes the simulated utility over a finite time horizon (given by T). (We minimize the utility multiplied by negative one to obtain the maximum.)

We can consider optimization under a static management strategy and a dynamic management strategy. As we have discussed, we determine the optimal static management strategy by optimizing the decision parameters to maximize discounted utility over a fixed window. To consider the optimal dynamic management strategy, we introduce an update time (given by τ) which is much smaller than the time horizon. We then update the management strategy every τ time steps. To do this, we begin with the optimal static management strategy and simulate the system under that strategy for τ time steps. We then reevaluate the optimal strategy (given the new state of the system) and update the management strategy. We simulate this new strategy for another τ time steps and repeat the evaluation process again. We repeat this process of optimization, simulation, and re-optimization as long as desired.

1.3 Derivation for critical transition in $k_c = k_p$

In our behavioral model, we observe that the number and character of equilibrium positions in the system change at $k_C = k_P$. Here we consider why this occurs.

For the administrative bloat solution (which is a valid solution in both allocation schemes), we find that the stability has a simple analytical solution. For our nonlinear system of equations, we find the Jacobian, evaluate at the bloat position ($R_u = 0, R_o = w/k_A$), and calculate the eigenvalues of the system. These two eigenvalues are analytically tractable; however, we will not go through the derivation here. We first present the resultant eigenvalues for the general allocation scheme:

$$\lambda_1 = -r_d \qquad \lambda_2 = k_A \left(\frac{p}{k_P} - \frac{c}{k_C} \right) \tag{5}$$

If we substitute c and p for the behavioral version and evaluate $\tilde{c}$ and $\tilde{p}$ at the bloat position, then we obtain the eigenvalues of the behavioral version:

$$\lambda_1 = -r_d \qquad \lambda_2 = \frac{k_A}{2 + \tilde{d}} \left(\frac{1}{k_P} - \frac{1}{k_C} \right) \tag{6}$$

This is why we have a critical transition at $k_P = k_C$. The equilibrium is stable if both eigenvalues are negative, and because r_d is a positive constant, λ_1 is always negative. Therefore, the administrative bloat position only changes stability when λ_2 switches signs. Since k_A and $\tilde{d}$ are both positive, we see that λ_2 can only change signs when the term in parenthesis changes signs. This only happens when $k_P = k_C$. If $k_P < k_C$, then λ_2 is positive, and the administrative bloat position is unstable. If $k_P > k_C$, then λ_2 is negative, and the administrative bloat position is stable.