Replication Data for "Improving the Organization of Knowledge in Production by Screening Problems"

Guilherme Carmona* Krittanai Laohakunakorn†

University of Surrey University of Surrey

July 23, 2023

Contents

*Address: University of Surrey, School of Economics, Guildford, GU2 7XH, UK; email: g.carmona@surrey.ac.uk.

†Address: University of Surrey, School of Economics, Guildford, GU2 7XH, UK; email: k.laohakunakorn@surrey.ac.uk.

1 List of files

We have provided the following files to allow replication of our numerical results in Sections 5.2, 5.3 and 5.5:

1. org.py
2. hier.py
3. diff.py
4. orders.py
5. orgL2.py
6. orgL3.py
7. orgL4.py
8. orgLaux.py
9. orgL.py
10. orgLoptimal.py
11. orgLh.py

2 Codes for the small ξ case

The following three codes are used for the computations in Section 5.2: First, org.py computes the optimal organization as a function of the parameters and hier.py computes the best hierarchy, also as a function of the parameters.[1] Then diff.py makes all the reported computations for the chosen parameter values.

[1] Both of these codes use the built-in function `minimize`. We have tried replacing it in org.py with `basinhopping` and the results are virtually the same but the program takes far longer to run.

3 Codes for intermediate values of ξ

In this section, we describe the computational approach used in Sections 5.3 and 5.5.

3.1 Computations

We assume that $\pi = h$ and that $h < 1$. The former simplifies the expression for α_i for each $i \in L \setminus \{1\}$ since, by Lemma A.11, $\alpha_i = h(1 - F((\cup_{j<i} B_j) \setminus A_i))$. The latter then implies that $\alpha_i < 1$ and Lemma A.16 implies that $\alpha_i > 0$.

We use the approach described in Section A.10 to compute optimal organizations. In what follows, we describe the candidates for optimal organizations when the number of layers is L and $L \in \{1, 2, 3, 4\}$.

3.1.1 $L = 1$

The best organization with one layer does not depend on ξ, i.e. $B_1 \setminus A_1 = \emptyset$ always. In this case, $\mu_1 = \min\left\{\max\left\{\frac{a-c}{b}, 0\right\}, 1\right\}$ and $y_1 = F(\mu_1) - c\mu_1$.

3.1.2 $L = 2$

In this case, $\mathcal{C} = \{A_1, A_2, B_1 \setminus A_1\}$. Since $A_1 < C$ for each $C \in \mathcal{C}$, there are two possible orders:

1. $A_1 < A_2 < B_1 \setminus A_1$, and
2. $A_1 < B_1 \setminus A_1 < A_2$.

We let $\mu_0 = \mu(A_1)$, $\mu_1 = \mu(A_2)$ and $\mu_2 = \mu(B_1 \setminus A_1)$. In order 1, $A_1 = [0, \mu_0)$, $A_2 = [\mu_0, \mu_0 + \mu_1)$ and $B_1 \setminus A_1 = [\mu_0 + \mu_1, \mu_0 + \mu_1 + \mu_2)$. Hence, $\alpha_2 = h(1 - F(\mu_0) - F(\sum_{i=0}^{2} \mu_i) + F(\mu_0 + \mu_1))$ and $y = (F(\mu_0 + \mu_1) - c\mu_0 - c\alpha_2\mu_1 - \xi\mu_2)/(1 + \alpha_2)$.

In order 2, $A_1 = [0, \mu_0)$, $B_1 \setminus A_1 = [\mu_0, \mu_0 + \mu_2)$ and $A_2 = [\mu_0 + \mu_2, \mu_0 + \mu_1 + \mu_2)$. Hence, $\alpha_2 = h(1 - F(\mu_0 + \mu_2))$ and $y = (F(\mu_0) + F(\sum_{i=0}^{2} \mu_i) - F(\mu_0 + \mu_2) - c\mu_0 - c\alpha_2\mu_1 - \xi\mu_2)/(1 + \alpha_2)$.

3.1.3 $L = 3$

In this case, $\mathcal{C} = \{A_3 \cap (B_1 \setminus A_1), A_1, A_2, A_3 \cap (B_1 \setminus A_1)^c, (B_1 \setminus A_1) \cap A_3^c, B_2 \setminus A_2\}$. We have that $A_1 < C$ for each $C \in \mathcal{C} \setminus \{A_1\}$ and $A_2 < A_3 \cap (B_1 \setminus A_1)^c$ by Corollary 2. Moreover, Corollary 1 implies that $A_3 \cap (B_1 \setminus A_1) < A_3 \cap (B_1 \setminus A_1)^c$, $A_3 \cap (B_1 \setminus A_1) < (B_1 \setminus A_1) \cap A_3^c$, $(B_1 \setminus A_1) \cap A_3^c < B_2 \setminus A_2$ and $A_2 < B_2 \setminus A_2$.

When $L = 3$, we have that

$$B_1 \setminus A_2 = A_1 \cup (A_3 \cap (B_1 \setminus A_1)) \cup ((B_1 \setminus A_1) \cap A_3^c) \text{ and}$$
$$(B_1 \cup B_2) \setminus A_3 = A_1 \cup ((B_1 \setminus A_1) \cap A_3^c) \cup A_2 \cup (B_2 \setminus A_2).$$

Then:

$$\begin{aligned}
\alpha_2 = &\quad h(1 - F(A_1) - F(A_3 \cap (B_1 \setminus A_1)) - F((B_1 \setminus A_1) \cap A_3^c)), \\
\alpha_3 = &\quad h(1 - F(A_1) - F((B_1 \setminus A_1) \cap A_3^c) - F(A_2) - F(B_2 \setminus A_2)), \\
\gamma = &\quad 1 + \alpha_2 + \alpha_3 \text{ and} \\
\theta = &\quad F(A_1) + F(A_2) + F(A_3 \cap (B_1 \setminus A_1)) + F(A_3 \cap (B_1 \setminus A_1)^c) \\
&\quad - c\mu(A_1) - \xi\mu((B_1 \setminus A_1) \cap A_3^c) - c\alpha_2\mu(A_2) - \xi\alpha_2\mu(B_2 \setminus A_2) \\
&\quad - c\alpha_3\mu(A_3 \cap (B_1 \setminus A_1)^c) - (c\alpha_3 + \xi)\mu(A_3 \cap (B_1 \setminus A_1)).
\end{aligned}$$

There are eight possible orders consistent with Corollaries 1 and 2:

1. $A_1 < A_3 \cap (B_1 \setminus A_1) < A_2 < A_3 \cap (B_1 \setminus A_1)^c < (B_1 \setminus A_1) \cap A_3^c < B_2 \setminus A_2$.
2. $A_1 < A_3 \cap (B_1 \setminus A_1) < A_2 < (B_1 \setminus A_1) \cap A_3^c < A_3 \cap (B_1 \setminus A_1)^c < B_2 \setminus A_2$.
3. $A_1 < A_3 \cap (B_1 \setminus A_1) < (B_1 \setminus A_1) \cap A_3^c < A_2 < A_3 \cap (B_1 \setminus A_1)^c < B_2 \setminus A_2$.
4. $A_1 < A_2 < A_3 \cap (B_1 \setminus A_1) < A_3 \cap (B_1 \setminus A_1)^c < (B_1 \setminus A_1) \cap A_3^c < B_2 \setminus A_2$.
5. $A_1 < A_2 < A_3 \cap (B_1 \setminus A_1) < (B_1 \setminus A_1) \cap A_3^c < A_3 \cap (B_1 \setminus A_1)^c < B_2 \setminus A_2$.
6. $A_1 < A_3 \cap (B_1 \setminus A_1) < A_2 < (B_1 \setminus A_1) \cap A_3^c < B_2 \setminus A_2 < A_3 \cap (B_1 \setminus A_1)^c$.
7. $A_1 < A_3 \cap (B_1 \setminus A_1) < (B_1 \setminus A_1) \cap A_3^c < A_2 < B_2 \setminus A_2 < A_3 \cap (B_1 \setminus A_1)^c$.
8. $A_1 < A_2 < A_3 \cap (B_1 \setminus A_1) < (B_1 \setminus A_1) \cap A_3^c < B_2 \setminus A_2 < A_3 \cap (B_1 \setminus A_1)^c$.

It turns out that, in fact, order 1 is the optimal one in all of our simulations.

3.1.4 $L = 4$

In this case, $\mathcal{C} = \{A_1, A_3 \cap (B_1 \setminus A_1), A_4 \cap (B_1 \setminus A_1), A_2, A_4 \cap (B_2 \setminus A_2), A_3 \cap (B_1 \setminus A_1)^c, A_4 \cap (B_1 \setminus A_1)^c \cap (B_2 \setminus A_2)^c, (B_1 \setminus A_1) \cap A_3^c \cap A_4^c, (B_2 \setminus A_2) \cap A_4^c, B_3 \setminus A_3\}$.

First, we rule out as many orders as we can. Corollaries 1 and 2 imply that:

1. $A_1 < C$ for each $C \in \mathcal{C} \setminus \{A_1\}$ (Corollary 2).
2. $A_3 \cap (B_1 \setminus A_1) < A_3 \cap (B_1 \setminus A_1)^c$ (Corollary 1).
3. $A_3 \cap (B_1 \setminus A_1) < (B_1 \setminus A_1) \cap A_3^c \cap A_4^c$ (Corollary 1).
4. $A_4 \cap (B_1 \setminus A_1) < A_4 \cap (B_2 \setminus A_2)$ (Corollary 2).
5. $A_4 \cap (B_1 \setminus A_1) < A_4 \cap (B_1 \setminus A_1)^c \cap (B_2 \setminus A_2)^c$ (Corollary 1).
6. $A_4 \cap (B_1 \setminus A_1) < (B_1 \setminus A_1) \cap A_3^c \cap A_4^c$ (Corollary 1).
7. $A_4 \cap (B_2 \setminus A_2) < A_4 \cap (B_1 \setminus A_1)^c \cap (B_2 \setminus A_2)^c$ (Corollary 1).
8. $A_4 \cap (B_2 \setminus A_2) < (B_2 \setminus A_2) \cap A_4^c$ (Corollary 1).
9. $A_2 < (B_2 \setminus A_2) \cap A_4^c$ (Corollary 1).
10. $A_3 \cap (B_1 \setminus A_1)^c < B_3 \setminus A_3$ (Corollary 1).
11. $(B_1 \setminus A_1) \cap A_3^c \cap A_4^c < (B_2 \setminus A_2) \cap A_4^c$ (Corollary 2).
12. $(B_1 \setminus A_1) \cap A_3^c \cap A_4^c < B_3 \setminus A_3$ (Corollary 2).
13. $A_2 < A_3 \cap (B_1 \setminus A_1)^c$ (Corollary 2).
14. $A_3 \cap (B_1 \setminus A_1)^c < A_4 \cap (B_1 \setminus A_1)^c \cap (B_2 \setminus A_2)^c$ (Corollary 2).
15. $(B_2 \setminus A_2) \cap A_4^c < B_3 \setminus A_3$ (Corollary 2).
16. $A_2 < A_4 \cap (B_2 \setminus A_2)$ (Corollary 2).

When $L = 4$, we have that

$$\begin{aligned}
B_1 \setminus A_2 \quad &= (A_3 \cap (B_1 \setminus A_1)) \cup (A_4 \cap (B_1 \setminus A_1)) \\
&\cup((B_1 \setminus A_1) \cap (A_3^c \cap A_4^c)) \cup A_1, \\
(B_1 \cup B_2) \setminus A_3 \quad &= ((B_1 \setminus A_1) \cap A_4) \cup ((B_1 \setminus A_1) \cap (A_3^c \cap A_4^c)) \cup A_1 \\
&\cup(A_4 \cap (B_2 \setminus A_2)) \cup ((B_2 \setminus A_2) \cap A_4^c) \cup A_2, \text{ and} \\
(B_1 \cup B_2 \cup B_3) \setminus A_4 \quad &= (A_3 \cap (B_1 \setminus A_1)) \cup ((B_1 \setminus A_1) \cap (A_3^c \cap A_4^c)) \cup A_1 \\
&\cup((B_2 \setminus A_2) \cap A_4^c) \cup A_2 \cup (B_3 \setminus A_3) \cup (A_3 \cap (B_1 \setminus A_1)^c).
\end{aligned}$$

Hence,

$$\begin{aligned}
\alpha_2 \quad &= h(1 - F(A_3 \cap (B_1 \setminus A_1)) - F(A_4 \cap (B_1 \setminus A_1)) \\
&-F((B_1 \setminus A_1) \cap (A_3^c \cap A_4^c)) - F(A_1)), \\
\alpha_3 \quad &= h(1 - F((B_1 \setminus A_1) \cap A_4) - F((B_1 \setminus A_1) \cap (A_3^c \cap A_4^c)) - F(A_1) \\
&-F(A_4 \cap (B_2 \setminus A_2)) - F((B_2 \setminus A_2) \cap A_4^c) - F(A_2)), \text{ and} \\
\alpha_4 \quad &= h(1 - F(A_3 \cap (B_1 \setminus A_1)) - F((B_1 \setminus A_1) \cap (A_3^c \cap A_4^c)) - F(A_1) \\
&-F((B_2 \setminus A_2) \cap A_4^c) - F(A_2) - F(B_3 \setminus A_3) - F(A_3 \cap (B_1 \setminus A_1)^c)).
\end{aligned}$$

Also,

$$\begin{aligned}
\gamma \quad &= 1 + \alpha_2 + \alpha_3 + \alpha_4 \text{ and} \\
\theta \quad &= F(A_1) + F(A_3 \cap (B_1 \setminus A_1)) + F(A_4 \cap (B_1 \setminus A_1)) + F(A_2) \\
&+F(A_4 \cap (B_2 \setminus A_2)) + F(A_3 \cap (B_1 \setminus A_1)^c) + F(A_4 \cap (B_1 \setminus A_1)^c \cap (B_2 \setminus A_2)^c) \\
&-c\mu(A_1) - (c\alpha_3 + \xi)\mu(A_3 \cap (B_1 \setminus A_1)) - (c\alpha_4 + \xi)\mu(A_4 \cap (B_1 \setminus A_1)) \\
&-c\alpha_2\mu(A_2) - (c\alpha_4 + \xi\alpha_2)\mu(A_4 \cap (B_2 \setminus A_2)) - c\alpha_3\mu(A_3 \cap (B_1 \setminus A_1)^c) \\
&-c\alpha_4\mu(A_4 \cap (B_1 \setminus A_1)^c \cap (B_2 \setminus A_2)^c) - \xi\mu((B_1 \setminus A_1) \cap A_3^c \cap A_4^c) \\
&-\xi\alpha_2\mu((B_2 \setminus A_2) \cap A_4^c) - \xi\alpha_3\mu(B_3 \setminus A_3).
\end{aligned}$$

We use orders.py to find all orders consistent with the above results; there are 192 in total which are listed in the code.

We note that as it was the case where $L = 3$, order 1, which is now

$$A_1 < A_4 \cap (B_1 \setminus A_1) < A_3 \cap (B_1 \setminus A_1) < A_2 < A_3 \cap (B_1 \setminus A_1)^c < A_4 \cap (B_2 \setminus A_2)$$
$$< A_4 \cap (B_1 \setminus A_1)^c \cap (B_2 \setminus A_2)^c < (B_1 \setminus A_1) \cap A_3^c \cap A_4^c < (B_2 \setminus A_2) \cap A_4^c < B_3 \setminus A_3,$$

is optimal in all our simulations.

3.2 Code description

The starting point are the codes orgL2.py, orgL3.py and orgL4.py, each of which computes the optimal organization for the corresponding number of layers. In each of these codes, each possible optimal ordering of $\mathcal{C}$ is considered and the built-in function `minimize` is used to find the size of each element of $\mathcal{C}$ and corresponding output.[2] Then the order that leads to the highest such output is selected; the code returns the order of $\mathcal{C}$, the size of each element of $\mathcal{C}$, the output of the optimal organization, the size β_i of each layer and the costs of learning ν_i of each layer.

One aspect of the above codes which is worth discussing concerns the choice of the ordering of $\mathcal{C}$, which we illustrate in the case where $L = 2$. In this case there are two possible orderings: $\psi_1 = (A_1 < A_2 < B_1 \setminus A_1)$ and $\psi_2 = (A_1 < B_1 \setminus A_1 < A_2)$. These two orders are the same if $B_1 \setminus A_1 = \emptyset$, namely $A_1 < A_2$. Hence, in orgL2.py, the order $A_1 < B_1 \setminus A_1 < A_2$ is the optimal one only if $y_{L,\psi_2} > y_{L,\psi_1}$ and $\mu(B_1 \setminus A_1) > 1/100000$, i.e. $\mu(B_1 \setminus A_1)$ is significantly above 0.[3]

The next step is performed by orgLaux.py, which solves $\max_{L\in\{1,2,3,4\}} y_L$. One issue with this maximization problem is that often $y_{L+1} \geq y_L$ (and then possibly $y_{L+1} > y_L$ due to approximation errors) by simply taking the organization that yields y_L and adding layer $L+1$ with $B_{L+1} = A_{L+1} = \emptyset$. To avoid this, for e.g. $L = 3$

[2] We use the solution to the optimal organization with $L = 1$ as the initial guess except when it features $\mu(A_1) = 0$. In this case, we use `basinhopping` instead of `minimize`.

[3] This approach requires checking that the relevant sets that distinguish between certain orders are (significantly) nonempty. An alternative approach is to require that $y_{L,\psi_2} > y_{L,\psi_1} + 1/1000000$ for ψ_2 to be considered better than ψ_1, which also ensures that differences between the orders are not just the result of the approximate nature of the `minimize` algorithm. We take the latter approach in orgL4.py, where it is more convenient because of the large number of possible orders.

to be better than $L = 2$, we require not only that $y_3 > y_2$ but also that $\mu(A_3) = \mu(A_3 \cap (B_1 \setminus A_1)) + \mu(A_3 \cap (B_1 \setminus A_1)^c) > 1/100000$.

Finally, the computations and graphs reported in Section 5.3 and 5.5 are produced using orgL.py, orgLoptimal.py and orgLh.py.