Pseudocode:

Repeat steps 1-8 over multiple effort updates (we’ve checked that it works similarly for 100, 200 or 1000):

1. Initialize effort (e\_0) (randomly from normal distribution; always convergence when constant starting effort for everyone)

Repeat steps 2-5 for multiple (checked that doesn’t vary if 200 or 2000) noise draws:

1. Draw noise realization around z\_0 = y\_0 + e\_0
2. For each person, calculate dPr/de of being in any rank using the pdf centered around 0 & 0.18 (standard deviation of predicted performance)

Φ(z\_0k-1 + εk-1 – z\_0i) - Φ(z\_0k + εk – z\_0i)

1. For each person, find which of their preferences they would receive from being in different ranks
2. Calculate dEu/de = sum over ranks ((dPr(rank=j)/de)\*u(pref received in rank j))
3. Average dEu/de over all noise iterations
4. Update effort to 0.8\*old effort + 0.2\*(dEu/de)/(2\*alpha): note that we do this regardless of difference between 2\*alpha\*old effort and dEu/de, because we should (and do) observe convergence if we keep updating.
5. Make all negative efforts = 0 (effectively useful for the 1st update)