**Modeling response thresholds for multiple simultaneous stimuli using a Boltzmann-sampling approach**

Lynch C.M.1, Pavlic, T.P.1, 2, 3, 4

1School of Life Science, Arizona State University, Tempe, AZ, USA 85287

2School of Computing and Augmented Intelligence, Arizona State University, Tempe, AZ, USA 85287

3School of Complex Adaptive Systems, Arizona State University, Tempe, AZ, USA 85287

4School of Sustainability, Arizona State University, Tempe, AZ, USA 85287

**Abstract**

Response thresholds are often cited as an important mechanism behind division of labor. According to this hypothesis, individuals with low thresholds to some tasks will be more likely to perform those tasks and over time become specialists in those tasks. Modeling such a mechanism requires an understanding of how social insects integrate multiple competing signals for work, however we do not yet have a comprehensive view of the consequences of different modeling decisions. Response thresholds are usually used to model the task allocation process, but standard models either ignore the problem of multiple integration by considering only a single task, they allow virtual insects to only encounter a single task at a time, or they use the relative strengths of stimuli across tasks to calculate the probability of starting any one task. The first case is unrealistic, the second case does not consider the possibility that an antenna may harbor multiple types of pheromones simultaneously, and the third case tends to homogenize the probability of performing different tasks, muting division of labor. Additionally, these methods are unable to use negative task-associated stimuli as an input, which could be useful in contexts where the stimulus can be modeled as a distance from an objective. This paper proposes a Bolzmann sampler that can resolve both issues simultaneously. This sampler resembles the softmax function used in multi-class neural networks, and in the single-task case, it degenerates into a form similar to the Hill function used for most response threshold models. This sampler is a flexible extension of traditional response threshold models, as it is scale-free, symmetric, it can incorporate negative stimuli, and it is always guaranteed to produce a set of probabilities whose sum is 1. It also successfully reproduces colony-level behaviors of the Hill function, although it tends to more strongly emphasize the effect of thresholds, leading to higher levels of division of labor. This Boltzmann sampler is therefore a natural successor to the Hill function, and can be used in a wider set of scenarios for the fields of theoretical biology and swarm intelligence.

**Introduction**

The response threshold concept is an extremely parsimonious explanation of social insect behavior, combining explanations of plasticity and specialization into one mechanistic framework (Beshers & Fewell, 2001). Under this paradigm, individual workers in a colony intrinsically differ in the stimulus level at which they start working, and by performing the task associated with that stimulus, they reduce the stimulus and therefore also reduce the probability that this task will be performed by others (Jeanson & Weidenmüller, 2014). Models have shown that response thresholds can increase the flexibility and robustness of a colony in response to environmental perturbations (Calabi, 1988; Camazine et al., 2001) and it can act as a proximate mechanism behind division of labor (Beshers et al., 1999). Empirically, response thresholds have also captured social behaviors related to division of labor (Fewell & Page, 1999; Pankiw & Page, 2000; Holbrook et al., 2011; Holbrook et al., 2013; Brahma et al., 2018). Despite these successes, response threshold models remain limited in their ability to capture many ecologically-relevant behaviors at the heart of division of labor and other phenomena (Leitner et al., 2019; Ulrich et al., 2021).

Response thresholds are traditionally modeled with a Hill function, which takes a task-associated stimulus as an input and outputs an s-shaped curve which gives the probability of starting a single task (Bonabeau et al., 1996). Models using this function successfully predicted an uptick in division of labor in larger harvester ant colonies (Jeanson et al., 2007; Holbrook et al., 2011) as well as increased homeostasis for small clonal raider ant groups (Ulrich et al., 2018). It has also been used to solve task allocation problems in swarm robotics (Wu et al., 2018; Jiang et al., 2020). However, this function is limited in two respects.

1) The Hill function breaks down for negative task-associated stimuli. Traditionally, task-associated stimuli are thought to be physical cues that can only take on positive real values (Bonabeau et al., 1996), or the probability of starting a task must be set to 0 with a piecewise function if the stimulus becomes negative (Yang et al., 2009). Modeling them this way works for cases where the social insect colony wants to either minimize the stimulus (such as with brood pheromone; Pankiw et al., 1998) or maximize it (such as the amount of food in the larder, Lynch et al., *in prep*). However, there are cases where an intermediate value of the stimulus is preferred, such as with nest temperature (Weidenmüller, 2004). In these cases, it may make sense to model the task-associated stimulus not as the raw cue, but rather as a distance from an objective, which could take on a negative value. Having a low temperature - or a negative objective value - may make the probability of fanning increasingly small while the probability of incubating brood increasingly large (O'Donnell & Foster, 2001).

2) The Hill function has difficulty incorporating multiple tasks. Naively applying this function to multiple tasks simultaneously without a normalization protocol can result in probabilities that sum to a value greater than one. Multiple approaches have been proposed to solve this issue while still using the Hill function. Some assume that there is only one task (Bonabeau et al., 1996; Arcuri & Lanchier, 2017; Feng et al., 2021), others allow ants to randomly encounter tasks one at a time (Jeanson et al., 2007; Ulrich et al., 2018; Lin, 2021; Ulrich et al., 2021), and finally some calculate relative stimuli and thresholds (Wu et al., 2018; Jiang et al., 2020; Lynch et al., *in prep*). The assumption that there is only one task is likely problematic in many cases, as real social insects perform multiple tasks and variable task numbers can influence model performance (Dornhaus et al., 2019). Random encounter models can be biologically reasonable within specific contexts, such as within nests where stimuli may be locally isolated (Ravary et al., 2007; Jeanson & Lachaud, 2015). However, this assumption is not always met, as social insects can have more open nests (Tschinkel, 2005) which can allow them to encounter multiple pheromones at the same time. They can then differentiate these signals with different types of sensilla (Dussutour et al., 2009). Reprocessing a task-associated stimulus as a relative value to other stimuli can avoid the assumptions of the previous two approaches, however as we will show, this approach tends to uniformly distribute the probabilities of performing different tasks, which may or may not be a desirable property.

A special form of a Boltzmann sampler can be used to resolve both these issues. Boltzmann samplers are designed to sample from different types of combinatorial structures, which will have a certain size depending on how many different combinations of elements there are within that structure (Duchon et al., 2004). In the field of statistical mechanics, physicists are interested in describing the distribution of particles in different states. Higher-energy states are less likely to occur than lower-energy states, as there are fewer possible combinations of microscopic configurations that result from that energy. When one equates energy with the size of combinatorial structure while also including the temperature of the system, the Boltzmann sampler returns the Boltzmann distribution, the maximal entropy probability distribution across these energy states. As this distribution includes a partition function which guarantees that the sum of the probabilities will be 1, this distribution has applications in many other fields. Most famously, it is known as the softmax function in data science, where it is used as the last activation function of a neural network. For neural networks, the softmax function normalizes the output of the algorithm so that the scores associated with various classes can be interpreted as probabilities. This function has also been used to find the probability of performing different actions in field of reinforcement learning (Sutton & Barto, 1998), and so it seems like a natural candidate for modeling response thresholds.

In the case of social insects, the difference between the task-associated and the response threshold stimulus is analogous to the energy of the system (or the scores of a neural network). If we further equate the inverse of the temperature with the steepness of the probability curve (the Hill coefficient), then the Boltzmann sampler gives a mathematically trackable way of normalizing the probabilities of performing different tasks that can allow for both positive and negative task-associated stimuli. This also offers an interpretation of how a colony integrates multiple task-associated cues in a way similar to that of physical particles, but it allows these particles some degree of volition by giving each their own threshold. The Boltzmann sampler also has a number of other desirable properties that make it a natural extension of the Hill function. When incorporated into models of division of labor, it produces qualitatively similar results to the Hill function in both single and multi-task scenarios. The function has point symmetry in the single task case, the steepness of the function only depends on a single parameter, the function is scale-free, and it emphasizes the effect of the thresholds on an individual social insect’s behavior. It can also be easily modified to incorporate other phenomena as well, including task completion cues and satisfaction thresholds (Lynch et al., *in prep*). Together, these properties ease the process of model construction, allowing for scenarios that would not have been possible to model using traditional Hill functions.

**1. Building Threshold Models with a Boltzmann Sampler**

*1.1 Single task response thresholds models can be derived from a softmax function*

Response thresholds are typically modeled probabilistically. That is, different values of the task-associated stimulus (x) yield different probabilities of a response P(x, 𝛉). This is usually written as a Hill function:

P(x, \theta) = \frac{x^k}{x^k+\theta^k}

which generates an s-shaped probability curve. Here, θ is the inflection point (the response threshold) of the curve while k (the Hill coefficient) determines the steepness of the curve. As k approaches infinity, the curve approaches a stepwise function, and can be interpreted as a measure of how deterministic the social insect’s choice is.

However, as Kanakia et al. (2016) showed, the response threshold can be reimagined as the logistic function S(x, 𝛉):

S(x, \theta) = \frac{1}{1+e^{-k(x-\theta)}}

Here, the numerator sets the maximum level of the logistic, which we set to 1 as we interpret this output as a probability. S(x, 𝛉), however, can only function if there are two possible outcomes, either performing the task or not performing it. If there are multiple potential outcomes, then a separate logistic function cannot be used for each task, as with the Hill function, the sums of the resulting probabilities could be greater than 1. The softmax function \sigma(\vec z) can be used here as well to resolve this issue:

\sigma(\vec z)_i = \frac{e^{z_i}}{\sum_{j=1}^Ke_i^{z_j}}

where the vector \vec zcontains the task associated stimuli for K-1 tasks (the remaining state is the resting state). In the case where there are only two possible outcomes, \vec z = \begin{bmatrix}x \\ 0\end{bmatrix} where x is the task associated stimulus for the task and 0 is the task associated stimulus for the resting state, which is to say that there is no stimulus for not performing a task. We can now solve for the first element of the vector, \sigma(\vec z) simplifies to:

\sigma(\vec z)_1 = \frac{e^{z_1}}{e^{z_1}+e^{z_2}} = \frac{e^x}{e^x+1} = \frac{1}{1+e^{-x}} 

which is the usual, less general version of the logistic function S(x, 𝛉). This process shows that the probability of responding to a stimulus in some sense is still modeled as a logistic, only it is now normalized by the task-associated stimuli of other tasks. In order to arrive at S(x, 𝛉) for the one task task, we must first redefine the softmax function to \phi(\vec z, \vec \theta):

\phi(\vec z, \vec \theta)_i = \frac{e^{k(z_i-\theta_i)}}{\sum_{j=1}^Ke^{k(z_j-\theta_j)}}

where \vec \thetais the vector which contains the response thresholds for each task. In this context, k is the inverse of temperature for both the Boltzmann distribution and the softmax function used in neural networks, which are special cases of the Boltzmann sampler. When \vec z= \begin{bmatrix}x \\ 0\end{bmatrix} and \vec \theta = \begin{bmatrix} \theta_1 \\ 0\end{bmatrix} where 0 indicates that there is a response threshold of 0 for not performing a task, and we solve for state 1, \phi(\vec z, \vec \theta) simplifies to:

\phi(x, \theta)_1 = \frac{e^{k(x-\theta_1)}}{e^{k(x-\theta_1)} + e^{k(0-0)}} = \frac{e^{k(x-\theta_1)}}{e^{k(x-\theta_1)}+1} = \frac{1}{1+e^{-k(x-\theta_1)}} = S(x, \theta)


The sum of \phi(\vec z, \vec \theta) across i is guaranteed to be 1 as we are normalizing by the sum, so this function can be used to model a set of K-1 response thresholds for an individual social insect.

*1.2 Incorporating satisfaction thresholds and task completion stimuli*

We can further generalize the softmax function by allowing it to incorporate satisfaction thresholds as well as task completion stimuli (Lynch et al., *in prep*). To review, response threshold models usually assume that the signals that determine a need for a task increase over time, and can be interpreted as an ‘on’ signal called a task demand stimulus. Satisfaction thresholds determine the probability of stopping a task rather than starting a task. Therefore, the probability of stopping tasks should decrease as the task demand stimulus increases. However, if the signal decreases over time rather than increases, then it can be interpreted as an off signal and is called a task completion stimulus. The probability of starting a task should decrease with higher task completion stimuli, and satisfaction thresholds should increase. We therefore need a function that can create s-shaped curves with a negative as well as a positive first derivative.

To do this, we can introduce two binary variables \alpha, \beta \in \{ -1,1\}. 𝛂 is encoded such that 1 represents a task demand stimulus and -1 represents task completion. 𝛃 is similar where 1 represents a response threshold model and -1 represents a satisfaction threshold model (Fig. 1). Multiplying these variables determines the sign of the exponent, which in turn determines the slope of the resulting probability curves. For response thresholds, these curves represent the probability of starting a task whereas for satisfaction thresholds these represent the probability of stopping a task. The general function can be written as:

\phi(\vec z, \vec \theta, \alpha, \beta)_i = \frac{e^{\alpha \beta k(z_i-\theta_i)}}{\sum_{j=1}^K e^{\alpha \beta k(z_j-\theta_j)}}

Note that if 𝛂 ⨉ 𝛃 = -1, and K = 2, then the first element of this function reduces to:

\phi(x, \theta, \alpha, \beta)_1 = \frac{1}{1+e^{k(x-\theta_1)}}

This function, like the probability function used in Lynch et al. (*in prep*) sums to unity for all x when added to the 𝛂 ⨉ 𝛃 = 1 version of the function, indicating that the two functions are complements of one another:

\\
1 = \frac{1}{1+e^{k(x-\theta_1)}}+\frac{1}{1+e^{-k(x-\theta_1)}} 
\\
1(e^{k(x-\theta_1)}+1)(e^{-k(x-\theta_1)}+1) = \frac{(e^{k(x-\theta_1)}+1)(e^{-k(x-\theta_1)}+1) }{1+e^{k(x-\theta_1)}} + \frac{(e^{k(x-\theta_1)}+1)(e^{-k(x-\theta_1)}+1) }{1+e^{-k(x-\theta_1)}}
\\
(e^{k(x-\theta_1)}+1)(e^{-k(x-\theta_1)}+1) = e^{k(x-\theta_1)} + e^{-k(x-\theta_1)} + 2
\\
0 = e^{k(x-\theta_1)} + e^{-k(x-\theta_1)} + 2 - (e^{k(x-\theta_1)}+1)(e^{-k(x-\theta_1)}+1) 
\\
0 = 0

Response and satisfaction thresholds can also be incorporated into the same model of an ant to produce a composite threshold (Lynch et al., *in prep*). Response threshold models usually assume that the probability of stopping and starting (respectively) are constant (Bonabeau et al., 1996; Jeanson et al., 2007; Jiang et al., 2020). That is, the mirror of the ɸ function, ɸ’, is a constant here denoted as ci:

\phi'(\vec z, \vec \theta, \alpha, \beta)_i = \phi'_i = c_i

where 0 ≤ ci ≤ 1. It is also possible to force both the starting and stopping probabilities to be dependent on the task-associated stimulus, which would be a composite threshold model. In this case, the mirror function would be:

\phi'(\vec z, \vec \zeta, \alpha, \beta)_i = \frac{e^{\alpha \beta k (z_i-\zeta_i)}}{\sum _{j=1}^Ke^{\alpha \beta k (z_j-\zeta_j)}}

where the vector 𝝵 contains the thresholds for the opposite operation, depending on how one defines the primary function. For example, if ɸ determines the probability of starting a task, 𝛉 would contain response thresholds while ɸ’ would determine the probability of stopping the task and 𝝵 would contain satisfaction thresholds (Fig. 1).

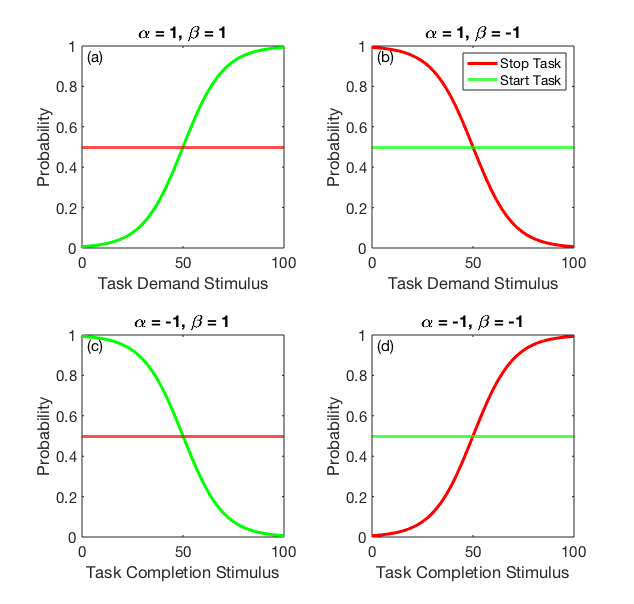


Figure 1: ɸ and ɸ’ functions when 𝛉 = 50, k = 0.1, c = 0.5, and K = 2. In each case, the logistic is generated from ɸ while the straight line is from ɸ’. Red lines show the probability of stopping a task, and the green lines show starting probabilities. Each panel shows the effect of different combinations of 𝛂 and 𝛃 on the logistic curve. (a) represents a response threshold model with a task demand stimulus, (b) satisfaction threshold model with task demand stimulus, (c) response threshold model with task completion stimulus, (d) satisfaction threshold with task completion stimulus.

**2. Useful Properties of the Boltzmann Sampler**

*2.1 Softmax probabilities automatically sum to one*

The softmax function allows for flexible model building. For instance, transitions between states can be effectively disallowed by making the threshold for a transition extremely high when 𝛂 ⨉ 𝛃 = 1, or extremely negative when 𝛂 ⨉ 𝛃 = -1. These transitions can also be used for continuous-time models or discrete-time models. Additionally, the partition function in the denominator will allow any distribution of thresholds can be used and the function will still sum to 1, regardless of how thresholds are distributed or change over time:

\sum_{i=1}^K\phi(\vec z, \vec \theta, \alpha, \beta)_i = \sum_{i=1}^K \frac{e^{\alpha \beta k(z_i-\theta_i)}}{\sum_{j=1}^K e^{\alpha \beta k(z_j-\theta_j)}} = \frac{\sum_{i=1}^K e^{\alpha \beta k(z_i-\theta_i)}}{\sum_{j=1}^K e^{\alpha \beta k(z_j-\theta_j)}} = 1

*2.2 Task associated stimuli can be negative for the softmax function*

P(x, 𝛉) breaks down when x < 0, as it is either undefined on the real number line for fractional values of k or tends to increase for whole numbers of k. This latter property can be problematic in cases where there is a negative task-associated stimulus.

The softmax function can easily incorporate negative distances without any of the issues that arise with P(x, 𝛉). To see this, we can evaluate the limit of the softmax function 0 as zi → -∞ for different signs of the exponent given different signs of the exponent:

\lim_{z_i \to -\infty} \frac{e^{\alpha \beta k(z_i-\theta_i)}}{\sum_{j=1}^K e^{\alpha \beta k(z_j-\theta_j)}} = 
\begin{cases}
        1, & \text{if } \alpha \times \beta = -1\\
        0, & \text{if } \alpha \times \beta = 1
\end{cases}


We can do the same for zi → ∞:

\lim_{z_i \to \infty} \frac{e^{\alpha \beta k(z_i-\theta_i)}}{\sum_{j=1}^K e^{\alpha \beta k(z_j-\theta_j)}} = 
\begin{cases}
        0, & \text{if } \alpha \times \beta = -1\\
        1, & \text{if } \alpha \times \beta = 1
\end{cases}


Thus, we can see that the function will always be constrained between 0 and 1 regardless of the sign of the exponent and whether zi is positive or negative.

*2.3 Probabilities curves around softmax threshold are symmetric*

The function S(x, 𝛉) has point symmetry around (𝛉, ½) such that S(x, 𝛉) for x > 𝛉 is a 180° rotation of S(x, 𝛉) for x ≤ 𝛉. This happens because S(x, 𝛉) satisfies the condition 1-S(x, 𝛉) = S(-x, 𝛉) where 𝛉 translates the inflection point (the point of symmetry) along the x-axis. The function P(x, 𝛉) does not have this property. While P(0, 𝛉) = 0, the function asymptotes in the positive x direction as

\lim_{x \to +\infty} \frac{x^k}{x^k+\theta ^k} = \lim_{x \to +\infty} \frac{1}{1+\frac{\theta^k}{x^k}} = \frac{\lim_{x \to +\infty} 1}{\lim_{x \to +\infty}  1+\frac{\theta^k}{x^k}} = \frac{1}{1} = 1

If P(x, 𝛉) represents a response threshold, this means that a virtual ant is guaranteed to not perform a task when the task demand stimulus is 0, but is not guaranteed to start the task even if the stimulus is extremely high. Therefore, this function does not have point symmetry.

*2.4 Steepness property dependent on only a single parameter for the softmax function*

The steepness of the function Hill function P(x, 𝛉) is purported to be controlled k, however this is decidedly not the case as the steepness of the function also depends on 𝛉 (Fig. 2). This is problematic for biological models, as this makes it more difficult to interpret the effects of each parameter in isolation of one another. For swarm roboticists, this is problematic as it makes it more difficult to control the exact shapes of these curves as neither parameter is truly free of the other. On the other hand, these parameters are largely independent for S(x, 𝛉).

We can show this by first taking the derivative of this function:

\frac{d}{dx}[S(x, \theta)] = \frac{d}{dx}(\frac{1}{1+e^{-k(x-\theta)}}) = \frac{ke^{k(x-\theta)}}{(1+e^{k(x-\theta)})^2}

At the inflection point, x and 𝛉 cancel out each other and this expression simplifies to:

\frac{ke^{0}}{(1+e^{0})^2} = \frac{k}{4}

Thus, the steepness of S(x, 𝛉) only depends on k. This is not the case for P(x, 𝛉). Taking the derivative of this function yields:

\frac{d}{dx} [P(x, \theta)] = \frac{d}{dx} (\frac{x^k}{x^k + \theta^k}) = \frac{\theta^kkx^{k-1}}{(x^k+\theta^k)^2}

We can now evaluate the slope of P(x, 𝛉) at the inflection point where x = 𝛉 = 𝝊:

\frac{\upsilon^kk\upsilon^{k-1}}{(\upsilon^k+\upsilon^k)^2} = \frac{k}{4\upsilon}

So we can see that as 𝛉 increases, the slope of P(x, 𝛉) at the inflection point decreases, so slope does not purely depend on k.

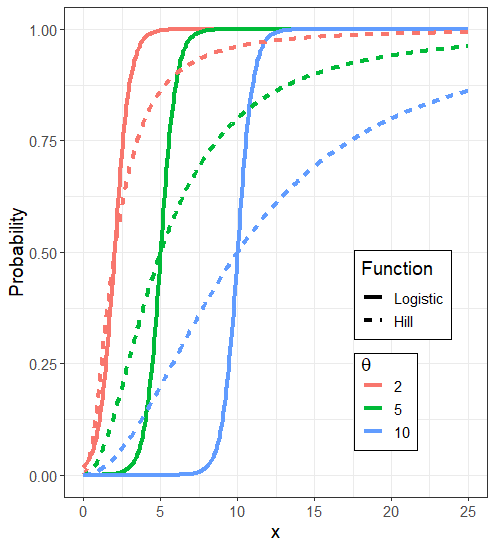
**

Figure 2: Response probability curves for Hill (dashed line) vs softmax functions (solid line) across different values of 𝛉 (color). In these plots, the steepness parameter k for both functions is set to 2. Note that the probability curves for the logistic tend to be parallel to one another despite variation in 𝛉. The same is not true for the Hill function.

We can perform the same derivation for the general, multi-task case for softmax (SI 1). If S_i = \phi(\vec z, \vec \theta, \alpha, \beta)_i  and a = \alpha \beta k , then the partial derivative for the softmax function is:

\frac{\partial S_i}{\partial z_j} = \begin{cases} 
      aS_i(1-S_j) & \text{if} \ i = j \\
      -a^2S_iS_j&  \text{if} \ i \neq j
   \end{cases}


In the i = j case:

aS_i(1-S_i) = a(\frac{e^{a(z_i-\theta_i)}}{\sum_{j=1}^Ke^{a(z_j-\theta_j)}})(1-\frac{e^{a(z_i-\theta_i)}}{\sum_{j=1}^Ke^{a(z_j-\theta_j)}})

At the inflection point of all tasks, z_i = \theta_i \ \forall \ i = 1, 2, ..., K, so:

\\ a(\frac{e^{a(z_i-\theta_i)}}{\sum_{j=1}^Ke^{a(z_j-\theta_j)}})(1-\frac{e^{a(z_i-\theta_i)}}{\sum_{j=1}^Ke^{a(z_j-\theta_j)}}) = \\
 a(\frac{e^{a(0)}}{\sum_{j=1}^Ke^{a(0)}})(1-\frac{e^{a(0)}}{\sum_{j=1}^Ke^{a(0)}}) \\
 = a(\frac{1}{K})(1-\frac{1}{K}) = \frac{a(K-1)}{K^2}

Performing this same procedure for i ≠ j yields \frac{-a^2}{K^2}. In the response threshold model, 𝛂 ⨉ 𝛃 = 1, so here a = k, and therefore the steepness of the response surface does not depend on 𝛉, only on k and the number of states.

*2.5 Scale-free nature of softmax function*

The debate on how to integrate multiple stimuli is similar to the debate on how to convert the output of a neural network into a probability. The naive approach would be to use normalization, which would conserve the relative proportions of each score. Conversely, the softmax function down-weighs non-maximal values without eliminating them, increasing the probability that the highest score will be chosen in a process perhaps not dissimilar to that of a real neural system (find biological example, edge detection?).

One method of setting the probabilities to do tasks with response thresholds is similar in spirit to normalization. In Lynch et al. (*in prep*), the probability of performing a task for a response threshold task demand model is the response threshold multiplied by the bias b for that particular task:

P(x, \theta)_i = \frac{b_ix_i^k}{x_i^k + \theta_i^k}

where b is the relative strength of a focal task-associated stimulus compared to all other stimuli:

b_i = \frac{x_i}{\sum_{j=1}^Kx_j}

which guarantees that the sum of probabilities across all tasks will be less than 1. Similar to normalization by sum, it will assign a probability of 0 to a task which has a task-associated stimulus of 0 (which may or may not be biologically feasible), it has trouble handling negative values, and when the thresholds for all tasks are equal, then the relative proportions of stimuli are correlated in the relative probability values. None of these properties are shared by the softmax function.

Consider the following numeric example. Here, we find the output of both the Hill and softmax functions given the vector [2 4 3]’. To isolate the effect of the functions, we set all 𝛉i = 1, and we set k to its typical value of 2. The first function is the Hill, and the latter the softmax (note that the output vector for the Hill does not sum to 1, as the remaining probability is allocated to the rest state. None of the task-associated stimuli were made to represent the rest state for the softmax function, so this output sums to 1. This difference is irrelevant for this demonstration):

\begin{bmatrix}
           2 \\
           4 \\
           3
         \end{bmatrix}  \rightarrow

\begin{bmatrix}
           0.178 \\
           0.418 \\
           0.3
         \end{bmatrix} , \quad \begin{bmatrix}
           2 \\
           4 \\
           3
         \end{bmatrix}  \rightarrow

\begin{bmatrix}
           0.015 \\
           0.867 \\
           0.117
         \end{bmatrix} 


For the Hill, we can see that the output probabilities are close to the bias for its corresponding stimulus (for example, 2/[2+4+3] ~ 0.222 ~ 0.178). In fact, as k increases, these probabilities will converge on the biases when xi > 𝛉i or will go to 0 when xi < 𝛉i or bi /2 when xi = 𝛉i. Conversely, the softmax function will disproportionately promote the highest stimulus, although lowering the response thresholds (or increasing satisfaction thresholds) of the other stimuli will mute this effect. Notably, the softmax function will retain these probabilities even if a constant is added to all the stimuli. Here, we add 100 to the input vector:

\begin{bmatrix}
           102 \\
           104 \\
           103
         \end{bmatrix}  \rightarrow

\begin{bmatrix}
           0.33 \\
           0.336 \\
           0.333
         \end{bmatrix} , \quad \begin{bmatrix}
           102 \\
           104 \\
          103
         \end{bmatrix}  \rightarrow

\begin{bmatrix}
           0.015 \\
           0.867 \\
           0.117
         \end{bmatrix} 


This occurs because (given a = \alpha \beta k ):

\\\frac{e^{a(z_i+c-\theta_i)}}{\sum_{j=1}^Ke^{a(z_j+c-\theta_j)}} = \\

\frac{e^{az_i+ac-a\theta_i}}{\sum_{j=1}^Ke^{az_j+ac-a\theta_j }}= \\

\frac{e^{az_i-a\theta_i}e^{ac}}{\sum_{j=1}^Ke^{az_j-a\theta_j} e^{ac}} = \\

\frac{e^{az_i-a\theta_i}}{\sum_{j=1}^Ke^{az_j-a\theta_j}} = \\

\frac{e^{a(z_i-\theta_i)}}{\sum_{j=1}^Ke^{a(z_j-\theta_j)}}  = 

\phi(\vec z, \vec \theta, \alpha, \beta )


whereas

\frac{\frac{z_i}{\sum_{v=1}^Kz_v}z_i^k}{z_i^k + \theta_i^k} \neq \frac{\frac{z_i+c}{\sum_{v=1}^Kz_v+c}(z_i+c)^k}{(z_i+c)^k + \theta_i^k} 

Thus the softmax function is less sensitive to sudden increases in all of the stimuli.

**3. Colony-level Dynamics of Softmax vs Hill Functions**

*3.1 Hill and logistic functions produce qualitatively similar dynamics in a single task model*

The logistic function S(x, 𝛉) is a good extension of the Hill function P(x, 𝛉), as it can produce nearly the same dynamics at a colony level. Consider the following toy discrete time Markovian model as a test case. Here, there are N ants that can be in K = 2 states S where S \in \{0,1\}. 0 is the rest state, 1 is the active state. The variable xt is the value of the task demand stimulus at time t and can take on any real positive number while t = 1, 2, …, T. Each ant has an identical response threshold 𝛉 which is the point where xt results in a 50% chance that the ant will start performing the task. Once the ant has started to perform the task, she has a fixed probability c of stopping the task. The probability of starting is either determined by the Hill or logistic functions. In the case of the Hill, this probability is:

P(x_t,\theta) = \frac{x_t^{k_s}}{x_t^{k_s}+\theta^{k_s}}

where ks is the steepness of the probability curve for the Hill function. The probability for the logistic function is:

S(x_t,\theta) = \frac{1}{1+e^{-k_l(x_t-\theta)}}

where kl is the steepness of the logistic. The same values ks and kl can produce drastically differently shaped probability curves, so to ensure equivalence we set kl = 5 and then find the value of ks that minimizes the sum of squared differences between the two types of curves (Fig. 3; Fig. 4a).

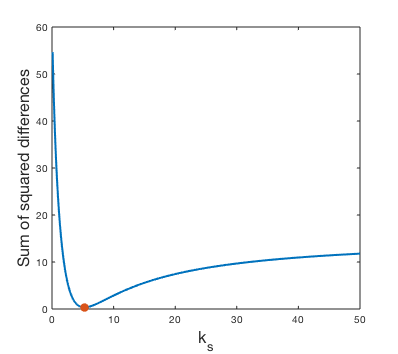


Figure 3: Sum of squared differences between P(x, 𝛉) and S(x, 𝛉) across different values of ks when kl = 5 and 𝛉 = 1 across the vector **x** = [0, 0.005, 0.01, …, 2]. The minimum of this difference is indicated by the red point where ks = 5.3.

Every timestep the task demand stimulus updates according to the following function:

x_t = x_{t-1} + \lambda - \frac{\upsilon n_t}{N}

where nt is the number of ants in state 1 at time t and 𝝊 is the amount of stimulus an ant can remove from the environment in a single timestep. To ensure linear scaling of work across colony sizes, we divide 𝝊nt by N. When P is the general probability of starting a task (so it is either P(x, 𝛉) or S(x, 𝛉)), nt is given by:

n_t = (N-n_{t-1})P + n_{t-1}(1-c)

We are interested in exploring stability points on either side of the response threshold to see how the two probability functions differ. We can find the x-nullcline by setting xt = xt-1 = x:

\\x = x + \lambda - \frac{\upsilon n_t}{N} 
\\ \lambda = \frac{\upsilon n_t}{N} 

Similarly, the n-nullcline is given by:

\\n = (N-n)P + n(1-c)
\\n = \frac{PN}{c+P}

The system will reach an equilibrium where these two nullclines intersect. By definition, then, at the point of stability nt will be constant, and since N is only a scaling factor, the final x value will only be determined by the relationship between 𝝊 and λ. We arbitrarily decide to fix 𝝊 = ⅕ and only vary λ between λmin ≤ λ ≤ λmax. As we are not interested in exploring other properties of the model, we set other parameters to c = ½, 𝛉 = 1, kl = 5, ks = 5.3, and N = 100 as well as n1 = 0 and x1 = 1. For simulations, we set tmax = 1000.

Next, we solve for λmin and λmax so that x will neither dip below 0 and will not increase at a rate higher than the colony can manage. λmax is the growth rate where equilibrium will be reached when the maximum number of ants are active (nmax) . This occurs when P = 1, so:

\\n_{max} = \frac{1N}{c+1} = \frac{2}{3}N
\\ \lambda_{max} = \frac{\upsilon n_{max}}{N} = \frac{\frac{1}{5}( \frac{2}{3}N)}{N} \approx 0.1333

Conversely, λmin is the growth rate that balances with the minimum number of active ants (nmin). nmin  = 0 for P(x, 𝛉) but for S(x, 𝛉) the probability of starting a task at x = 0 is approximately 0.0067 = Pmin, so we must calculate nmin  for this case:

\\n_{min} = \frac{P_{min}N}{c+1} = \frac{2}{3}NP_{min}
\\\lambda_{min} = \frac{\upsilon n_{min}}{N} = \frac{\frac{1}{5}NP_{min}}{N} \approx 0.0013

We have verified through simulations that this system will eventually reach equilibrium when λ is within these bounds before t = tmax (Fig. 4b).

Fixed parameters were chosen such that half the workers would be active at λ = 𝝊/2 and xt would stabilize at the 𝛉 for both models. The equilibrium value of x is xe. By adjusting λ, we can move the x-nullcline on either side of the threshold and see the differences between the two models. To derive a closed-form expression for how λ affects xe for both P(x, 𝛉) but for S(x, 𝛉) , we must find where the nullclines intersect. We will start by finding the n-nullcline for P(xe, 𝛉):

n = \frac{P(x_e, \theta)N}{c+P(x_e, \theta)} =
\frac{\frac{x_e^{k_s}}{x_e^{k_s}+\theta^{k_s}}N}{c+\frac{x_e^{k_s}}{x_e^{k_s}+\theta^{k_s}}}

Which we plug into the x-nullcline:

\\ \lambda = \frac{\upsilon n}{N} = 

\frac{\upsilon \frac{\frac{x_e^{k_s}}{x_e^{k_s}+\theta^{k_s}}N}{c+\frac{x_e^{k_s}}{x_e^{k_s}+\theta^{k_s}}}}{N} =
 \frac{\upsilon \frac{x_e^{k_s}}{x_e^{k_s}+\theta^{k_s}}}{c+\frac{x_e^{k_s}}{x_e^{k_s}+\theta^{k_s}}}




Solving for xe yields:

x_e = e^{\frac{\ln(\frac{\frac{\lambda c \theta^{k_s}}{\upsilon-\lambda}}{1-\frac{\lambda c}{\upsilon-\lambda}})}{k_s}}


The same process for the logistic function gives:

x_e = \frac{-\ln(\frac{\upsilon - \lambda - \lambda c}{\lambda c})+k_l\theta}{k_l}


The number of active workers at equilibrium is identical for both models, as it does not depend on the probability of starting a task. Rearranging the x-nullcline so that we solve for n yields:

n = \frac{\lambda N}{\upsilon }


The 3 equations above have each been validated via simulations.

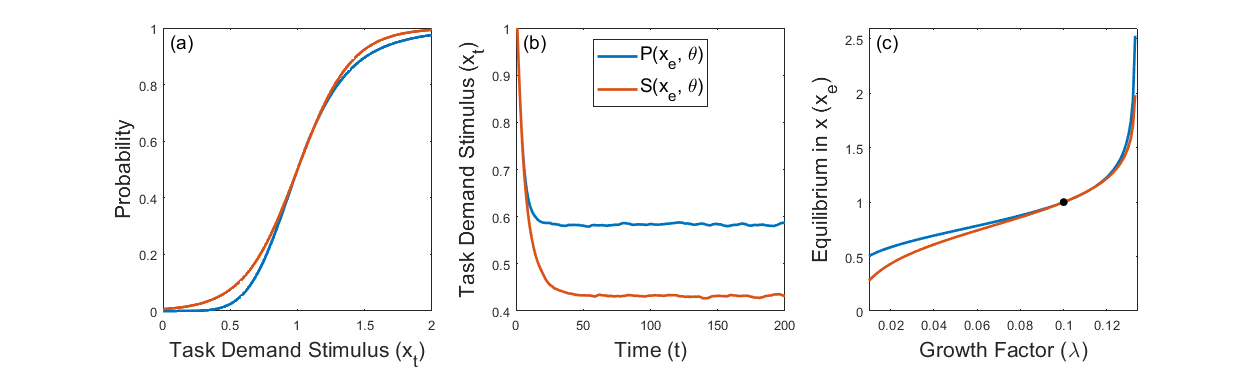


Figure 4: a) Response threshold curves for both the Hill function (blue) and the logistic function (red) when 𝛉 = 1, kl = 5, and ks = 5.3. b) Average value of xt over time for 100 simulations where λ = 0.02, N = 100, c = ½, 𝝊 = ⅕, n1 = 0 and x1 = 1. While tmax = 1,000, we only show the first 200 timesteps to show more detail at the beginning of the simulation. c) xe across λmin ≤ λ ≤ λmax. The two functions cross at the black point (λ = 𝝊/2, xe = 1). To the left of this point, simulations tend to hover below the response threshold 𝛉 while to the right of this point simulations hover above it.

In Fig. 4c, we can see that in almost all cases xe is higher for the Hill function than it is for the logistic function (it should be noted, though, that there is a thin strip between λ = 0.1 and λ ≈ 0.113 where the logistic is slightly higher. This corresponds to the region of the probability curves (Fig. 4a) between x = 1 and x ≈ 1.13 where the Hill probability is slightly higher than the logistic probability). This occurs because the probability of starting the task for the logistic is generally higher than the probability for the Hill, and thus ants in with that model tend to keep the stimulus level lower. Despite this, the shapes of the curves in Fig. 4c are nearly identical, and so both models produce qualitatively similar dynamics.

*3.2 Hill functions and Boltzmann samplers produce similar colony-level dynamics, but the behaviors of individuals differ*

While the softmax function produces similar results to the Hill function when there is only one task, it produces a different set of predictions when there is more than one task. Quantitatively, colony-level behaviors are more predictable in softmax than they are with alternative models. If the average threshold is viewed as the target stimulus level, then the stimulus values produced by the softmax function for each task are close to this target, and the variance of this stimulus is also fairly low over time. Qualitatively, these colony-level metrics are controlled by different sets of parameters. The variance of thresholds has a stronger effect on the softmax model than alternative models, while the opposite is true for the average value of these thresholds. Thus, distributions of thresholds will have different effects on each type of model and can therefore alter downstream predictions of these models.

To demonstrate this, we can compare the dynamics of a softmax model to a random encounter model as well as a normalization model of task allocation. As the name implies, random encounter models do not assume that social insects actually integrate multiple stimuli at the same time, rather agents randomly encounter stimuli in the environment and only calculate their response to that single stimulus. These models are sometime formulated so that the agents are guaranteed to respond to a stimulus when the threshold is crossed (Jeanson et al., 2007), however this seems unrealistic given that social insects are rarely so deterministic (Leitner et al., 2019) even after training (Giurfa & Sandoz, 2012). We therefore consider a probabilistic version of the random encounter model, where an ant has a set probability of performing a task as determined by the Hill function. We also consider the normalization model presented in section 2.5.

These models are treated as discrete time Markovian models, which is largely derived from Jeanson et al., 2007. Here, there are K-1 tasks, and thus the state space S \in \{ 0, 1, 2, ..., K-1\}where 0 represents the rest state. xti is the value of the task demand stimulus at time t for task i and can take on any real positive number while t = 1, 2, …, T. As there are K-1 tasks, the vector **x**t which contains the task-associated stimuli for all tasks has K-1 elements. However, the softmax model requires K elements so that the final stimulus can correspond to the rest state. The value of this final element is set to 0. Each ant i has a response threshold 𝛉ij for task j. Each threshold for each task and ant is drawn independently from a normal distribution whose mean is μ and whose standard deviation is 𝛔. The threshold matrix for both the random encounter and normalization models has dimensions K-1 × N, however for the softmax model we need to include an additional row for the rest state and thus has dimensions K × N. The threshold corresponding to the rest state is set to 0.

In the random encounter model, a resting ant is exposed to each task in random order every timestep. The probability that the ant would then respond to that stimulus is given by the Hill function:

P(x_{ti}, \theta_{ij})_{ij} = \frac{x^{k_{RE}}_{ti}}{x^{k_{RE}}_{ti} + \theta^{k_{RE}}_{ij}}

where d kRE gives the steepness of the random encounter model. The probability that ant j will perform task i at time t in the normalization model is:

P(x_{ti}, \theta_{ij})_{ij} = \frac{

\frac{x_{ti}^{k_N}}{\sum_{m=1}^{K-1}x_{tm}^{k_N}}
x^{k_{N}}_{ti}}{x^{k_{N}}_{ti} + \theta^{k_{N}}_{ij}}

where kN gives the steepness of the normalization model. In the softmax model, this probability is given by:

\phi (x_{ti}, \theta_{ij})_{ij} = \frac{e^{k_{SM}(x_{it}-\theta_{ij})}}{\sum_{m=1}^K e^{k_{SM}(x_{mt}-\theta_{mj})}}

where kSM gives the steepness of the softmax model. Once an ant is engaged in a task, she must at some point return to the rest state before starting another task. On average, an ant will spend 𝝉 = 5 timesteps on the task, or have a ⅕ probability of stopping the task.

Next, the task demand stimulus for each task is updated with the following function:

x_{ti} = x_{t-1i} + \frac{\frac{\alpha N 
\delta}{K-1}-\alpha n_{ti}}{N}

where N is the colony size, njt is the number of ants in state j at time t, 𝛂 is the amount of stimulus an ant can remove from the environment in a single timestep, and 𝛅 is the amount that each task increases every time step.

When comparing the performances of these models, we want to ensure that the stimulus does not increase at a rate faster than the rate at which the colony can remove this stimulus. Thus we want to find the maximum 𝛅 such that the stimulus stays constant throughout time (SI.2). This occurs when:

\delta = \frac{\tau}{\tau+1}

We do not vary 𝝉, so the upper limit for 𝛅 is 5/(5+1) = ⅚.

As was the case with the single-task model, we also need to ensure that the steepness of the probability curves are as equivalent as possible. That is, we matched a given value k of the softmax model, kSM, to the k of the normalization model (kN) and that of the random encounter model (kRE).The procedure for calibrating kSM to kRE is identical to that of the 1 task model. Here, we set kSM and 𝛉 to constant values and generated the corresponding single task logistic curve. We then iterated over values of kRE, measuring the sum of squared errors between the logistic curve and the corresponding single-task hill function. We then found the kRE that minimizes this error. We performed a similar procedure to find kN, except in that case the probability of performing one task depends on multiple simultaneous stimuli rather than just one. We therefore subtracted the probability of doing one task of the softmax function over all tasks from the probability of doing that task of the normalization function over all tasks.

We finally perform a sensitivity analysis where we vary 7 free parameters of the model. This is a full factorial 27 design where each free parameter has two levels, we run simulations on all combinations of free parameters, and we run 5 repeated measurements for every combination, meaning we ran a total of 27 𝗑 5 = 640 simulations. k_{sm} \in \{0.1, 0.5\}, \sigma \in \{0, 10\}, \mu \in \{40, 60\}, K-1 \in \{2, 5\}, N \in \{50, 100\}, \delta \in \{0.6, 0.8\}, and the initial value of x can either be 0 or 100.

From each simulation, we calculate the average xti across tasks and time for the last 500 timesteps, well after the colony has reached equilibrium. To see how close this equilibrium point is compared to the target point, we also calculate the squared difference between this mean and μ. We also measure the standard deviation of xti across time for these 500 timesteps, averaged across tasks. We calculate the division of labor index (Gorelick et al., 2004) as well as the percentage of individuals in a colony whose threshold rank order matches the reverse rank order of time the ant spent on each task. In other words, we count the number of timesteps an ant performed on each task and determine whether or not tasks that were performed often correspond to low thresholds (% agreement).

For each of these metrics, we perform a non-parametric Kruskal-Wallis test, grouping by model. If the test is significant (p-value < 0.05), then we perform a post hoc test to compare the medians of groups. Here, we do a pairwise Wilcoxon test with a Bonferonni p-value correction (Fig. 5).

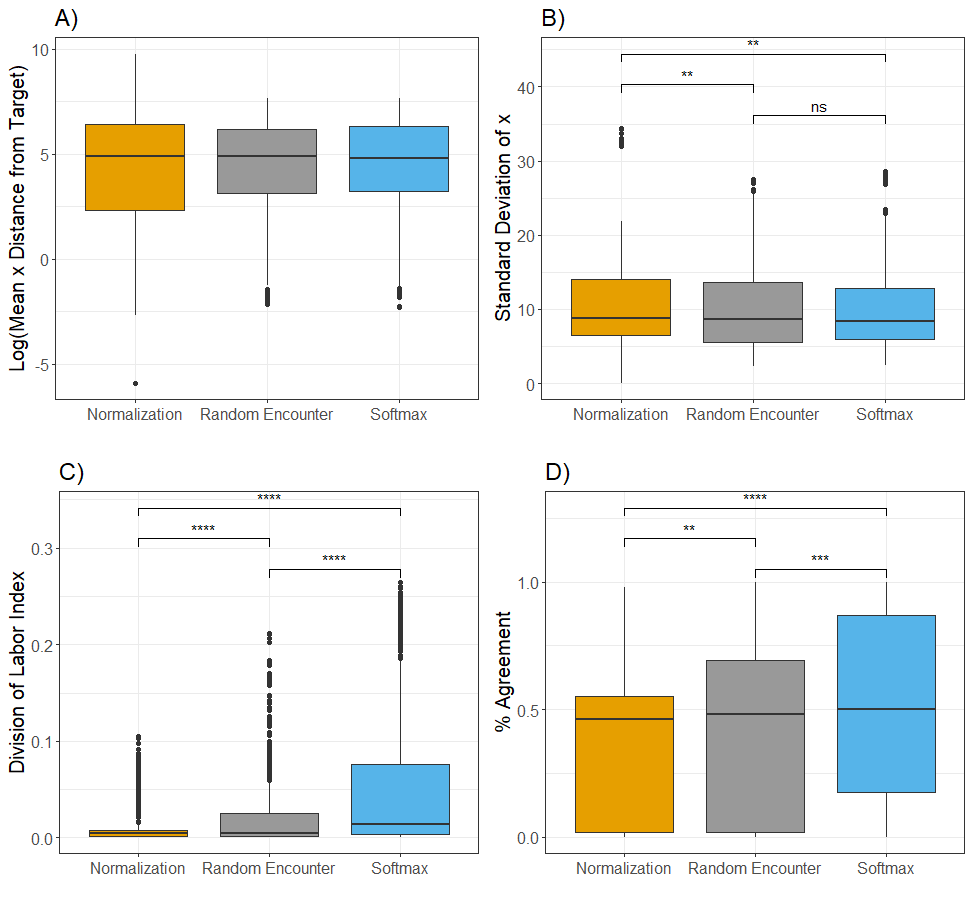


Figure 5: Colony level metrics of performance for each threshold model (A, B) and descriptions of individual worker performance (C, D) across simulations. A) Squared difference between equilibrium point and mean threshold μ, averaged across tasks. Kruskal-Wallis test was insignificant, so pairwise comparisons are not shown. Axis is log-transformed for visibility, but statistics were performed on raw data. B) Standard deviation of x over the last 500 timesteps of the simulation, averaged over tasks. Bars show significance of a pairwise comparison. C) Division of labor for each model. D) The percentage of individuals within a colony whose task preference aligned with their threshold order. ns = p-value > 0.05, \* = p-value < 0.05, \*\* = p-value < 0.01, \*\*\* = p-value < 0.001, \*\*\*\* = p-value < 0.0001.

We found that colony-level performance did not differ strongly among the models.The equilibrium point of x for each model was the same distance from the target μ (Fig. 5A; Kruskal-Wallis p-value < 0.05), and while the variation of x in the normalization model was significantly different from the other two, the median difference between all the models is minute (Fig. 5B). Despite this, individual workers behaved differently in each of these models. Division of labor is higher in the Softmax model (Fig. 5C), and thresholds are more likely to determine which tasks an individual is likely to perform (Fig. 5D). This means that individuals are more specialized in the softmax model, and that this specialization is driven by their response thresholds. This means that while the softmax function may not generate different predictions than the other models at a colony level, it may better represent the philosophy underlying the response threshold mechanism when there are multiple tasks.

**Conclusions**

Multiple task integration is a necessary component of any response threshold model which looks to explain the emergence of division of labor in social insect colonies. Tasks can sometimes be encountered sequentially if a task stimulus is locally bounded (such as wall building, Pinter-Wollman et al., 2012). However, these insects can also be exposed to these tasks concurrently. For instance, both temperature and barometric pressure are felt throughout the nest (Abou-Shaara et al., 2017; Sujimoto et al., 2020) and alarm pheromone can interrupt the performance of a task (Guo et al., 2022). These coincident signals can have nonlinear effects on behavior. For instance, bees which perform a waggle dance on empty cells recruit more foragers than those that perform the dance on capped brood cells (Tautz, 1996). Additionally, the nature of these stimuli are also not always known (Beshers & Fewell, 2001), and modeling them as solely positive numbers may not always be appropriate. For example, the distance from the nest entrance on a return journey can be negative if an ant walks past the entrance (Wehner, 2020). Any comprehensive model of division of labor should account for the integration of simultaneous task stimuli and should also be able to handle negative stimuli. Response thresholds, as modeled with a Boltzmann sampler, is a strong candidate for such a model.

Consider Smith et al. (*in prep*) as an example for the importance of incorporating both negative stimuli and multiple tasks. Here, authors used the softmax function to model macronutrient regulation experiment in leafcutter ants. In this experiment, a colony could forage from one of two food sources with varying amounts of protein and carbohydrates. One food item had a higher carbohydrate to protein ratio than the colony’s intake target, while the other had a lower ratio, so the colony would need to collect both to properly regulate the growth of its fungal gardens. Authors expected that specialists would arise in this system, where the number of foraging trips per ant would be correlated with the proportion of trips to one food item or the other. No such correlation arose. Instead, there seemed to be no relationship between the number of trips a forger took and the proportion of trips to one food item or the other. In the softmax model of this experiment, the distance between the colony’s current nutritional status and the intake target is the task-associated stimulus, and this distance can be either positive or negative. The probability of foraging from one food source depends just on this stimulus, while the probability of foraging from the other depends on the negative stimulus. Each ant also has two fixed, independently drawn response thresholds, one for each food item. Simulations recreated seemingly random experimental distributions of behavior, showing that the number of foraging trips was a result of the sum of the thresholds, whereas the propensity to choose one food item over the other was determined by the difference in thresholds. Thus, a threshold model can potentially describe a biological phenomenon, even in the absence of apparent specialists. This model, like others before it (Ulrich et al., 2021) shows the importance of considering multiple thresholds simultaneously, as well as the need to sometimes consider negative stimuli.

This model also demonstrates that, under the softmax framework, it is possible for a single stimulus to trigger different responses at different levels of the stimulus. For instance, temperature can trigger both fanning and incubating when it is either high or low (Weidenmüller, 2004). Task allocation in this context can therefore be modeled in much the same way as the macronutrient example where one behavior is triggered by the stimulus directly and the other is triggered by the negative stimulus. This is the equivalent of saying that temperature is a task demand cue for one task, but is a task completion cue for the other (Lynch et al., *in prep*).

This highlights another advantage of the Boltzmann sampler. A single, closed form vector function can be used to model a plethora of scenarios, as the partition function guarantees that the sum of the resulting probabilities will be 1. For instance, a modeler can explore how threshold distributions may affect colony performance (Beshers & Fewell, 2001), and any real-valued threshold landscape will be possible using this sampler. Thresholds can change over time with reinforcement (response thresholds) or descenitation (satisfaction thresholds). They can even be correlated with one another via copulas regardless of one’s choice of threshold distribution (Bouyé et al., 2000), although some caution must be taken with this approach as there are non-intuitive constraints on multiple correlations. For instance, correlations between multiple random variables are transitive, so independently setting the correlation between any pairwise set of variables is impossible. Additionally, there is a lower limit on how negatively correlated these random variables can be. If we define a set of n identical random variables with variation 𝛔2, then the correlation between those variables cannot be lower than -\frac{\sigma^2}{n-1}. Finally, there is the issue of overfitting. Including too many higher-order correlational structures may allow a model to fit any dataset. The most parsimonious assumption then - that thresholds are independent of one another - may be the most advantageous in many situations. This partition function can also allow the steepness parameter to be a vector if tuning the stochasticity of different tasks is biologically meaningful or has some adaptive benefit. Finally, thresholds also change according to the behavioral context (Sazaki et al., 2014; Sadeh & Clopath, 2022).

The Boltzmann sampler also allows for more control of the model compared to the random encounter or normalization alternatives. First, the Boltzmann sampler is scale-free, meaning that a simulation can be ‘zeroed’ at any value of the stimulus and the workers will behave in the same way. The same is not true of the normalization model, where the probabilities of performing tasks becomes more uniform if the stimulus is much higher than the thresholds. A way to interpret this result for the normalization case is to say that when several cues are saturating the antenna, then the ant loses the ability to discriminate between the tasks and chooses to do one task randomly. This could be biologically reasonable, but this situation can also be modeled in the softmax case, where the steepness parameter could be turned into a function of the stimuli. Lowering the steepness parameter can make these probabilities more uniform for the softmax function, so one can imagine devising a function that lowers the steepness parameter when the ant is distracted by many stimuli. However, the Hill is not so easily controlled. Even with high values of the steepness parameter and low values of the threshold, the probabilities of performing each task stay fairly uniform if the stimuli themselves are at similar levels. Thus, the softmax is the more flexible alternative to integrating multiple stimuli, as it can maintain a responsive system even with similar stimuli values while it can also be made to model scenarios where the social insect cannot distinguish between stimuli.

The Boltzmann sampler also allows for more control in the sense that the steepness of the probability curves is only determined by a single parameter, whereas the steepness of Hill functions is determined by both the steepness parameter and the threshold. This conflation can make it difficult to interpret the effect of each of these parameters on a model, and it limits the number of potential configurations. The effects of the steepness and threshold parameters are inversely proportional, one cannot have a steeply-curved probability distribution that also has a high threshold. As these two parameters are independent of each other in the softmax function, though, then any combination of parameters is possible there. Again though, the softmax function can be forced to operate like the Hill function if one chooses by setting the two to be inversely proportional, but the reverse is not true.

The independence between parameters and scale-free properties are also important because the variables used to represent stimuli in social insect response threshold models are unitless (Bonabeau et al., 1996; Jeanson et al., 2007; Ulrich et al., 2021; Lynch et al., *in prep*). To our knowledge, no study has used measurements on real task-associated stimuli to parameterize the model, and so levels of the stimulus are chosen arbitrarily. These levels could artificially influence the performance of models using a Hill function, as high levels can homogenize probabilities and decrease the steepness of the probability curve. However, these issues are not present for the softmax model, and since it can also take on negative values, the stimulus can be set to any level without influencing performance.

Ultimately, these differences in properties did not strongly affect the outcome of the simulations tested here. Regardless of which model we used, colonies achieved about the same level of stimulus and the same variation in that stimulus over time. However, as individuals were not randomly exposed to different tasks, and probabilities were not homogenized when the stimulus was high, workers using the softmax mechanism tended to perform the task that corresponded to their lowest response threshold, resulting in higher levels of division of labor despite attempts to place the models on equal footing. This makes the Boltzmann sampler both a good extension of the Hill function while also improving on it, as workers operate more like one would expect given the verbal description of the response threshold hypothesis. Additionally, these themes of control should also make the softmax function an attractive option for a task allocation algorithm for swarm robotics, which already rely heavily on social insect inspired Hill functions (Kanakia et al., 2016; Wu et al., 2018; Jiang et al., 2020).

**Supplemental Information**

*SI.1: Partial derivatives of softmax and hill functions*

Let S_i = \phi(\vec z, \vec \theta, \alpha, \beta)_i , a = \alpha \beta k , and \Sigma = \sum_j^Ke^{\alpha \beta k(z_j - \theta_j)}. The partial derivative for the probability of performing task i with respect to the stimulus of an arbitrary j task is:

\frac{\partial S_i}{\partial z_j} = \frac{\partial}{\partial z_j} (\frac{e^{a(z_i-\theta_i)}}{\Sigma})

In the case where i = j:

\frac{\partial}{\partial z_j} (\frac{e^{a(z_i-\theta_i)}}{\Sigma}) = \frac{ae^{a(z_i-\theta_i)}\Sigma - ae^{a(z_j-\theta_j)ae^{a(z_i-\theta_i)}}}{\Sigma^2} \\

=\frac{ae^{a(z_i-\theta_i)}\Sigma - a^2e^{a(z_j-\theta_j)e^{a(z_i-\theta_i)}}}{\Sigma^2} \\

= \frac{ae^{a(z_i-\theta_i)}}{\Sigma}\frac{\Sigma - ae^{a(z_j-\theta_j)}}{\Sigma} \\

=aS_i(1-\frac{ae^{a(z_j-\theta_j)}}{\Sigma}) = aS_i(1-S_j)


When i ≠ j:

\\ \frac{\partial}{\partial z_j} (\frac{e^{a(z_i-\theta_i)}}{\Sigma}) = \frac{0-ae^{a(z_j-\theta_j)}ae^{a(z_i-\theta_i)}}{\Sigma^2}

\\ = -a^2\frac{e^{a(z_j-\theta_j)}e^{a(z_i-\theta_i)}}{\Sigma^2}

\\ = -a^2\frac{e^{a(z_j-\theta_j)}}{\Sigma} \frac{e^{a(z_i-\theta_i)}}{\Sigma} 

\\ = -a^2S_jS_i



So in summary:

\frac{\partial S_i}{\partial z_j} = \begin{cases} 
      aS_i(1-S_j) & \text{if} \ i = j \\
      -a^2S_iS_j&  \text{if} \ i \neq j
   \end{cases}


*SI.2: Maximum growth rate for multi-task response threshold models*

To find the maximum growth rate for each of the response threshold models, we must find the value of 𝛿 where the stimulus for the current timestep is equal to that of the previous timestep, so:

x_{ti} = x_{t-1i} + \frac{\frac{\alpha N 
\delta}{K-1}-\alpha n_{ti}}{N}

becomes:

x_{ti} = x_{ti} + \frac{\frac{\alpha N 
\delta}{K-1}-\alpha n_{ti}}{N} = \alpha(\frac{\delta}{K-1}-\frac{n_{ti}}{N})

we can then solve for 𝛅:

\\ 0 = \alpha (\frac{\delta}{K-1}- \frac{n_{ti}}{N}) \\
\frac{\delta}{K-1} = \frac{n_{ti}}{N}
\\ \delta = \frac{n_{ti}(K-1)}{N}

K and N are both constants, but we need an expression for nti. We can find the maximum possible value of nti by assuming that the thresholds for all ants are always crossed, and so will disperse into all tasks uniformly. In a simplified mode where all ants are guaranteed to start a task given they are in the rest state, the initial stimulus is set to 0, and so all ants start in the inactive state at t = 0. At t = 1, N/T ants switch into task j. At t = 2, most ants stay in the state, but N/T\*1/𝝉 stop performing the task while the others continue performing the task. At t = 3, the N/T\*1/𝝉 ants that had stopped performing the task return to task j, however a subset of the ants that remained also stopped performing the task at the same time ([N/T - 1/𝝉 \* N/T]\*1/𝝉). This process can be described with the recursive formula:

n_{ti} = n_{t-1j} - \frac{1}{\tau} n_{t-1j} + \frac{1}{\tau} n_{t-2j}

where n_{0i} = 0  and n_{1i} = \frac{N}{K-1} . This recursion can be expanded into the finite series:

n_{ti}=\frac{N}{K-1}(1-\frac{1}{\tau}+\frac{1}{\tau^2}-\frac{1}{\tau^3}+\frac{1}{\tau^4} - ... \mp\frac{1}{\tau^{t-1}})

which in turn can be expressed as a summation:

n_{ti}=\frac{N}{K-1}\sum_{m=1}^t (-1)^{m+1}\frac{1}{\tau^{m-1}}

which has the closed form:

n_{ti}=\frac{N}{K-1}\frac{(-1)^{t+1} \tau ^{1-t}+\tau}{\tau+1}

The limit of this expression as t → ∞ will give us the long-run equilibrium value of nti:

n_{ti}=\frac{N}{K-1} \lim_{t\to\infty} \frac{(-1)^{t+1} \tau ^{1-t}+\tau}{\tau+1} = \frac{N}{K-1}\frac{\tau}{\tau +1}

We can finally plug this expression into the equation for 𝛅 to get our final equation:

\delta = \frac{\frac{N}{K-1}\frac{\tau}{\tau+1}(K-1)}{N} = \frac{\tau}{\tau+1}

Thus, in this idealized model, only the value of 𝝉 determines whether or not a colony can keep up with the demand of a task. We do not vary 𝝉, so the upper limit for 𝛅 is 5/(5+1) = ⅚.

**Works cited**

Abou-Shaara, H. F., Owayss, A. A., Ibrahim, Y. Y., & Basuny, N. K. (2017). A review of impacts of temperature and relative humidity on various activities of honey bees. Insectes sociaux, 64(4), 455-463.

Arcuri, A., & Lanchier, N. (2017). Stochastic spatial model for the division of labor in social insects. Mathematical Models and Methods in Applied Sciences, 27(01), 45-73.

Beshers, S. N., & Fewell, J. H. (2001). Models of division of labor in social insects. Annual review of entomology, 46, 413.

Beshers, S. N., Robinson, G. E., & Mittenthal, J. E. (1999). Response thresholds and division of labor in insect colonies. In Information processing in social insects (pp. 115-139). Birkhäuser, Basel.

Bonabeau, E., Theraulaz, G., & Deneubourg, J. L. (1996). Quantitative study of the fixed threshold model for the regulation of division of labour in insect societies. Proceedings of the Royal Society of London. Series B: Biological Sciences, 263(1376), 1565-1569.

Bouyé, E., Durrleman, V., Nikeghbali, A., Riboulet, G., & Roncalli, T. (2000). Copulas for finance-a reading guide and some applications. Available at SSRN 1032533.

Calabi, P. (1988). Behavioral flexibility in Hymenoptera: a re-examination of the

concept of caste. In Advances in Myrmecology (ed. J. C. Trager), pp. 237–258. E.J.

Brill, New York.

Camazine, S., Deneubourg, J. L., Franks, N. R., Sneyd, J., Theraulaz, G.

& Bonabeau, E. (2001). Self-Organization in Biological Systems. Princeton University

Press, Princeton.

Dornhaus, A., Leitner, N., Lynch, N., Mallmann-Trenn, F., & Pajak, D. (2019, January). Remember the Past and Forget Thresholds. In 7th Workshop on Biological Distributed Algorithms (BDA).

Ducatelle, F., Di Caro, G. A., & Gambardella, L. M. (2010). Cooperative self-organization in a heterogeneous swarm robotic system. In Proceedings of the 12th annual conference on Genetic and evolutionary computation (pp. 87-94).

Duchon, P., Flajolet, P., Louchard, G., & Schaeffer, G. (2004). Boltzmann samplers for the random generation of combinatorial structures. Combinatorics, Probability and Computing, 13(4-5), 577-625.

Dussutour, A., Nicolis, S. C., Shephard, G., Beekman, M., & Sumpter, D. J. (2009). The role of multiple pheromones in food recruitment by ants. Journal of Experimental Biology, 212(15), 2337-2348.

Giurfa, M., & Sandoz, J. C. (2012). Invertebrate learning and memory: fifty years of olfactory conditioning of the proboscis extension response in honeybees. Learning & memory, 19(2), 54-66.

Gordon DM. Dynamics of task switching in harvester ants. Anim Behav. 1989;38:194–204.

Guo, X., Lin, M. R., Azizi, A., Saldyt, L. P., Kang, Y., Pavlic, T. P., & Fewell, J. H. (2022). Decoding alarm signal propagation of seed-harvester ants using automated movement tracking and supervised machine learning. Proceedings of the Royal Society B, 289(1967), 20212176.

Feng, T., Charbonneau, D., Qiu, Z., & Kang, Y. (2021). Dynamics of task allocation in social insect colonies: scaling effects of colony size versus work activities. Journal of Mathematical Biology, 82(5), 1-53.

Fewell JH, Page RE. The emergence of division of labour in forced associations of normally solitary ant queens. Evol Ecol Res. 1999;1:537–48.

Holbrook, C. T., Barden, P. M., & Fewell, J. H. (2011). Division of labor increases with colony size in the harvester ant Pogonomyrmex californicus. Behavioral Ecology, 22(5), 960-966.

Holbrook CT, Kukuk PF, Fewell JH. Increased group size promotes task specialization in a normally solitary halictine bee. Behaviour. 2013;150:1449–66.

Jeanson, R., & Lachaud, J. P. (2015). Influence of task switching costs on colony homeostasis. The Science of Nature, 102(5), 1-4.

Jeanson, R., & Weidenmüller, A. (2014). Interindividual variability in social insects–proximate causes and ultimate consequences. Biological Reviews, 89(3), 671-687.

Jeanson, R., Fewell, J. H., Gorelick, R., & Bertram, S. M. (2007). Emergence of increased division of labor as a function of group size. Behavioral Ecology and Sociobiology, 62(2), 289-298.

Jiang, C., Chen, T., Li, R., Li, L., Li, G., Xu, C., & Li, S. (2020). Construction of extended ant colony labor division model for traffic signal timing and its application in mixed traffic flow model of single intersection. Concurrency and Computation: Practice and Experience, 32(7), e5592.

Kanakia, A., Klingner, J., & Correll, N. (2016). A response threshold sigmoid function model for swarm robot collaboration. In Distributed Autonomous Robotic Systems (pp. 193-206). Springer, Tokyo.

Leitner, N., Lynch, C., & Dornhaus, A. (2019). Ants in isolation: obstacles to testing worker responses to task stimuli outside of the colony context. Insectes Sociaux, 66(3), 343-354.

Lin, M. R. (2021). Energy Use Scaling and Alarm Spread in Social Ants: An Investigation Using Multi-agent Simulation and Object Tracking (Doctoral dissertation, Arizona State University).

Lynch, C., Dornhaus, A, Wilson, R. New version of an old mechanism for task allocation in social insects. In preparation.

Obute, S. O., Kilby, P., Dogar, M. R., & Boyle, J. H. (2022). Swarm Foraging Under Communication and Vision Uncertainties. IEEE Transactions on Automation Science and Engineering.

O'Donnell, S., & Foster, R. L. (2001). Thresholds of response in nest thermoregulation by worker bumble bees, Bombus bifarius nearcticus (Hymenoptera: Apidae). Ethology, 107(5), 387-399.

Pankiw, T., Page Jr, R. E., & Kim Fondrk, M. (1998). Brood pheromone stimulates pollen foraging in honey bees (Apis mellifera). Behavioral ecology and sociobiology, 44(3), 193-198.

Pankiw T, Page RE. Response thresholds to sucrose predict foraging division of labor in honeybees. Behav Ecol Sociobiol. 2000;47:265–7.

Pinter-Wollman, N., Hubler, J., Holley, J. A., Franks, N. R., & Dornhaus, A. (2012). How is activity distributed among and within tasks in Temnothorax ants?. Behavioral Ecology and Sociobiology, 66(10), 1407-1420.

Ravary, F., Lecoutey, E., Kaminski, G., Châline, N., & Jaisson, P. (2007). Individual experience alone can generate lasting division of labor in ants. Current Biology, 17(15), 1308-1312.

Sadeh, S., & Clopath, C. (2022). Contribution of behavioural variability to representational drift. bioRxiv.

Sasaki, T., Hölldobler, B., Millar, J. G., & Pratt, S. C. (2014). A context-dependent alarm signal in the ant Temnothorax rugatulus. Journal of Experimental Biology, 217(18), 3229-3236.

Sujimoto, F. R., Costa, C. M., Zitelli, C. H., & Bento, J. M. S. (2020). Foraging activity of leaf‐cutter ants is affected by barometric pressure. Ethology, 126(3), 290-296.

Sutton, R. S. and Barto A. G. Reinforcement Learning: An Introduction. The MIT Press, Cambridge, MA, 1998. Softmax Action Selection

Tautz, J. (1996). Honeybee waggle dance: recruitment success depends on the dance floor. The Journal of experimental biology, 199(6), 1375-1381.

Tschinkel, W. R. (2005). The nest architecture of the ant, Camponotus socius. Journal of Insect Science, 5(1), 9.

Ulrich Y, Saragosti J, Tokita CK, Tarnita CE, Kronauer DJC. Fitness benefits and emergent division of labor at the onset of group-living. Nature. 2018;560:635–8. pmid:30135576

Ulrich, Y., Kawakatsu, M., Tokita, C. K., Saragosti, J., Chandra, V., Tarnita, C. E., & Kronauer, D. J. (2021). Response thresholds alone cannot explain empirical patterns of division of labor in social insects. PLoS biology, 19(6), e3001269.

Wehner, R. (2020). Desert navigator. In Desert Navigator. Harvard University Press.

Weidenmüller, A. (2004). The control of nest climate in bumblebee (Bombus terrestris) colonies: interindividual variability and self reinforcement in fanning response. Behavioral Ecology, 15(1), 120-128.

Wu, H., Li, H., Xiao, R., & Liu, J. (2018). Modeling and simulation of dynamic ant colony’s labor division for task allocation of UAV swarm. Physica A: Statistical Mechanics and its Applications, 491, 127-141.

Yang, Y., Zhou, C., & Tian, Y. (2009, February). Swarm robots task allocation based on response threshold model. In 2009 4th International Conference on Autonomous Robots and Agents (pp. 171-176). IEEE.