

# A Computational Model for Ant Sorting

Matteo Lugli

283122@studenti.unimore.it

July 17, 2025

**Abstract**—This mini-paper presents a minimal yet powerful computational model for ant sorting, inspired by naturally occurring behaviors in ant colonies. The model simulates how ants, through simple probabilistic rules and limited local memory, can organize scattered items—such as food or debris—into spatially coherent clusters. Relying solely on decentralized interactions, the system exhibits emergent self-organization without requiring any form of global control or communication. The algorithm is implemented in NetLogo and evaluated through a series of experiments, which demonstrate that even such a simple setup can lead to the spontaneous formation of distinct clusters. These results underscore the potential of minimal agent-based models for understanding biological organization and for applications in swarm intelligence.

## I. INTRODUCTION

Ant colonies represent a fascinating example of small-scale distributed systems found in nature. They display complex emergent behaviors, including organized foraging [1] and spatial sorting of resources. While ant foraging has inspired numerous applications in computer science [2], the mechanisms behind ant-sorting remains less understood and continue to be the subject of active research.

Despite their complex collective behaviors, individual ants are remarkably simple organisms. They operate based on a limited set of rules and interact with their environment through basic, local actions. These minimal capabilities, when combined across many individuals, give rise to sophisticated global patterns without the need for centralized control. This phenomenon is commonly known as *swarm intelligence* [3]. Figure 1 illustrates various ant nest topologies observed in nature, highlighting the remarkable ability of colonies to construct complex and intricate underground structures.

Ants not only construct complex nests composed of numerous galleries but also spatially allocate distinct areas for specific types of resources [4] [5]. In this work, I propose and implement a minimal computational model that emulates the sorting capabilities observed in ant colonies, without relying on global coordination and by allocating only a fixed-size memory to each individual ant.

The model is implemented in NetLogo, and it is publicly available on Github [6].

## II. METHODS

Algorithm 1 outlines the behavior of a single ant in the simulation. At each step, the ant decides whether to pick up a resource—if one is present on its current patch—or to drop the resource it is carrying, if it holds one. The decision to

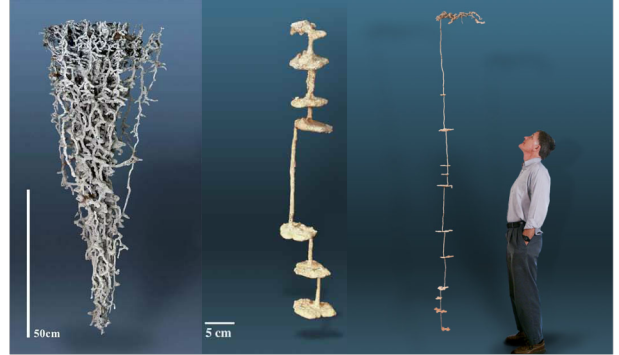


Fig. 1: Three different ant nests, exhibiting peculiar topologies. Usually, each room stores a specific type of resource.

pick up a resource depends on how frequently the resource type has been observed in the past. If the resource type is rare (i.e., appears infrequently in memory), the ant has a higher probability of picking it up. Conversely, the decision to drop a carried item depends on how familiar it is—frequently encountered resources are more likely to be dropped, since they are “boring”. Both decisions are probabilistic and rely on simple local rules and limited memory, making the overall system decentralized. Memory is updated every time a new resource is encountered, allowing ants to adjust their behavior dynamically based on their environment. Movement is stochastic, with ants performing a random walk to explore the space.

The hyper-parameters of the model are the following:

- $G$  (greed): used to compute the interest of the ant in picking up a resource; A higher  $G$  leads the ants in picking up resources more frequently;
- $C$  (conservation): used to compute the probability of the ant in dropping its resource. Higher  $C$  leads the ant in dropping things less frequently;
- $M$  (memory size): size of the memory array of each ant;
- $D$  (density):  $\frac{\text{n. of resource patches}}{\text{n. of total patches}}$ . Density does not change over time as ants can only move resources;
- $N$  (number of ants): total number of ants in the simulation;

To assess the quality of the clusters that form over time, I use a metric that I call *Mean Neighbor Rate* (MNR), defined in Equation 1. Here,  $N(i, j)$  denotes the set of neighboring cells (the 8 cells adjacent to the one at row  $i$  and column  $j$ ), and  $G[i, j]$  represents the color of the cell at position  $(i, j)$  in

the grid. Let  $n$  be the color of a neighboring cell. The MNR quantifies the average number of neighbors whose color  $n$  matches that of the central cell.

Figure 2 illustrates the state of the environment before and after a sample simulation. Over time, resources become clearly clustered by type.

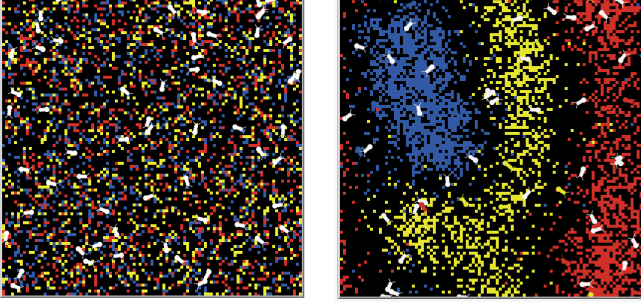


Fig. 2: NetLogo interface displaying the environment with resources and ants. The left image shows the scene immediately after the random setup phase, while the right image illustrates the color-clustered resources after 100,000 simulation steps.

$$MNR = \frac{1}{8 \times HWD} \sum_{i=1}^H \sum_{j=1}^W \sum_{n \in N(i,j)} [n = G[i, j]] \quad (1)$$

### III. EXPERIMENTS

In the proposed set of experiments, the goal is to evaluate the impact of  $G$ ,  $C$  and  $M$ . A comprehensive grid search falls outside the scope of this preliminary study and is therefore left for future work. The hyper-parameters used in the simulations are listed below, with corresponding results shown in Figure 3. The fixed constants were determined during an initial, informal evaluation phase, where a value of 4 was found to provide a good balance for both  $C$  and  $G$ . In the first two studies,  $M$  is fixed at 5, as allocating too much memory seemed akin to giving the ants excessive cognitive capacity.

- 1)  $C \in [1, 10]$ , while  $M = 5$ ,  $G = 4$ ;
- 2)  $G \in [1, 10]$  while  $M = 5$ ,  $C = 4$ ;
- 3)  $M \in [1, 10]$  while  $G = 4$ ,  $C = 4$ ;

In all experiments, the values of  $N$  and  $D$  were fixed to 60 and 30, respectively. These two parameters may be considered as hyper-parameters and could be subject to further investigation or optimization in future work.

It's not surprising that more memory leads to a better clustering factor, since the ants can yield more information about past experience. With  $M = 10$ , I was able to obtain a MNR of 55%. When  $M$  and  $C$  are fixed, setting  $G$  to high values negatively impacts performance. In this case, ants tend to pick up items even when they are not genuinely interested, leading to movements that do not contribute to cluster formation.

---

### Algorithm 1 Ant-sorting turtle procedure

---

```

1: procedure GO
2:   UpdateMemory;
3:   if not carrying object then
4:     TryPickUp;
5:   else
6:     TryDrop;
7:   end if
8:   Move;
9: end procedure
10: procedure TRY PICKUP
11:   if on a resource patch then
12:      $k \leftarrow \text{patch\_color}$   $\triangleright$  Resource type;
13:      $f \leftarrow \text{count of elements of type } k \text{ in memory}$ ;
14:      $I \leftarrow \frac{G}{G + f}$ ;
15:     With probability  $I$ , pick up the object;
16:   end if
17: end procedure
18: procedure TRY DROP
19:   if on an empty patch then
20:      $k \leftarrow \text{turtle's current color}$   $\triangleright$  Held object;
21:      $f \leftarrow \text{count of elements of type } k \text{ in memory}$ ;
22:      $D \leftarrow \frac{f}{C + f}$ ;
23:     With probability  $D$ , drop the object;
24:   end if
25: end procedure
26: procedure UPDATE MEMORY
27:   if on a resource patch then
28:     Replace a random memory slot with patch_color
29:   end if
30: end procedure
31: procedure MOVE
32:   Move forward
33:   Apply small random right and left turns
34: end procedure

```

---

Conversely, when  $M$  and  $G$  are held constant, the optimal value for the conservation rate  $C$  is approximately 5. Further increasing  $C$  does not degrade performance, as the improvement curve plateaus beyond this point. More details on how  $G$  and  $C$  influence the pickup-rate and the drop-rate can be found directly in the netlogo model.

### IV. CONCLUSIONS

In this work, I introduced a simple yet effective agent-based model that enables ants to cluster items through purely local interactions. Despite its simplicity, the model gives rise to emergent collective behavior, resulting in well-formed clusters without the need for global coordination or complex rules. The experiments demonstrate that such minimal mechanisms—based on probabilistic picking and dropping, along with memory and movement dynamics—are sufficient to reproduce the self-organizing principles observed in real-world ant behavior. This highlights the potential of lightweight

