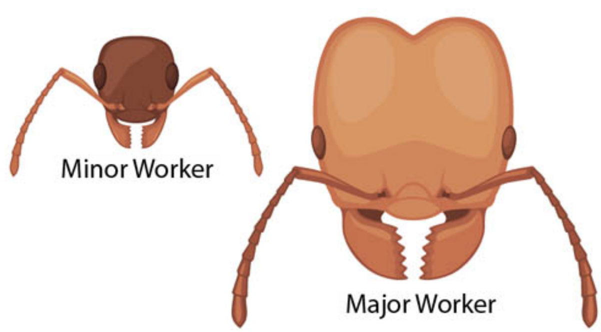
Chapter 3.4-3.6: Specialization and Adaptive Task Allocation

Tristan Scheiner and Julian Fiorito

1

Introduction and Review of 3.1-3.3

- Wanted to simulate division of labor which took inspiration from major and minor ants
- Created a model based on stimulus thresholds
- Response threshold that agents had to stimuli were static



2

Specialization: Probability of Tasks

$$T_{\theta_{ij}}(s_j) = \frac{s_j^2}{s_j^2 + \theta_{ij}^2}$$

$T_{\theta ij}$:	probability to do task at each time step
s_j:	demand for task j
θ_{ij}:	response threshold of individual i to task j

3

Specialization: Response Threshold

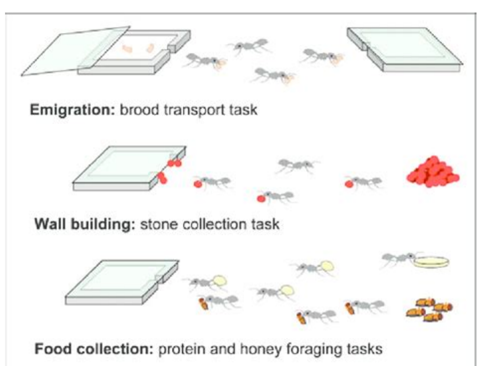
$$\theta_{ij} \leftarrow \theta_{ij} - \xi \Delta t$$

$$\theta_{ij} \leftarrow \theta_{ij} + \varphi \Delta t$$

$$\theta_{ij} \leftarrow \theta_{ij} - x_{ij}\xi \Delta t + (1 - x_{ij})\varphi \Delta t$$

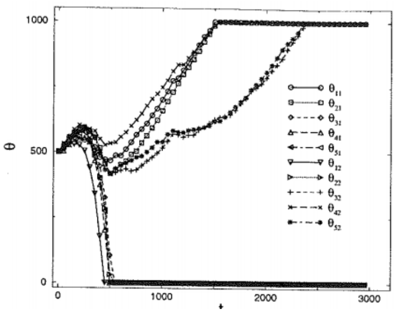
$$\partial_t \theta_{ij} = [(1 - x_{ij})\varphi - x_{ij}\xi]\Theta(\theta_{ij} - \theta_{\min})\Theta(\theta_{\max} - \theta_{ij})$$

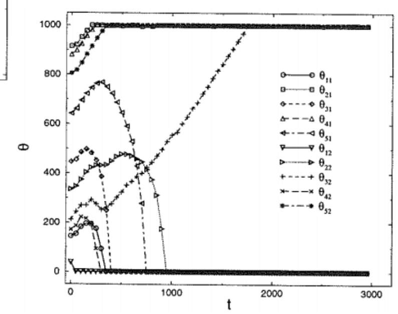
ξ:	learning rate
φ:	forgetting rate
Θ:	stepwise function to maintain bounds
x_{ij}:	fraction of time spent doing j

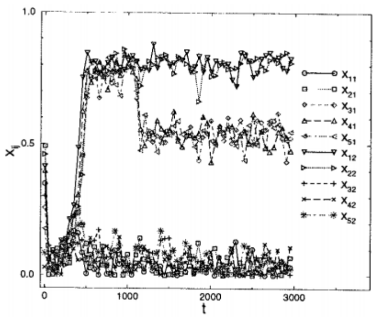


4

Specialization: Specialization of Role

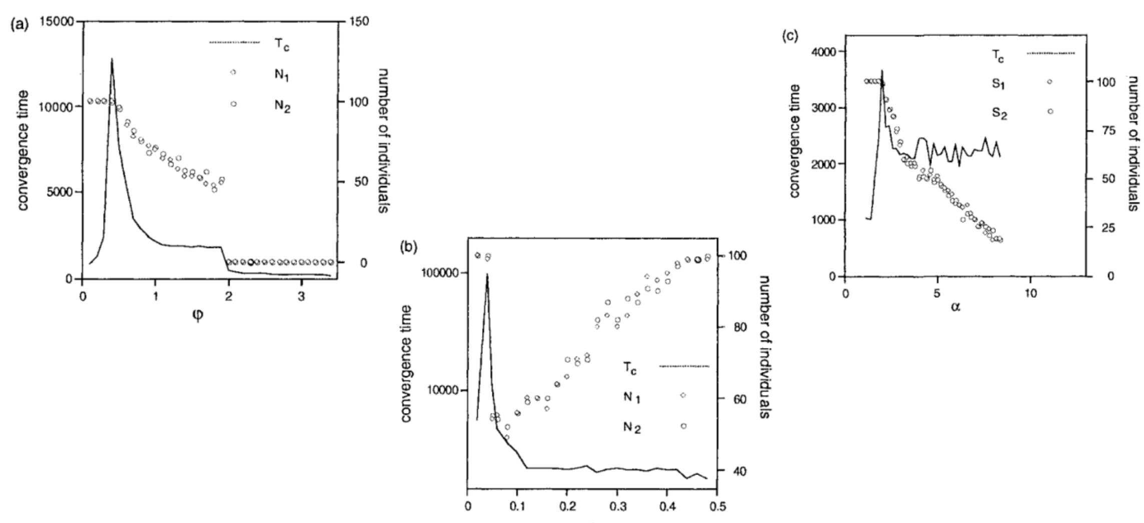






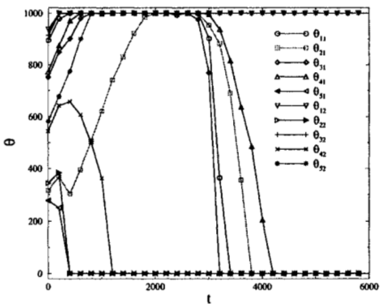
5

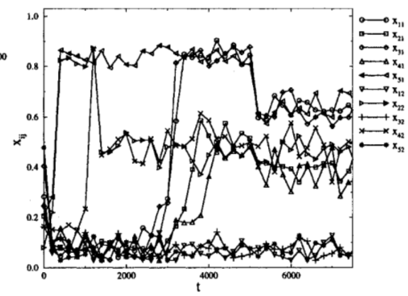
Specialization: Varying Hyperparameters

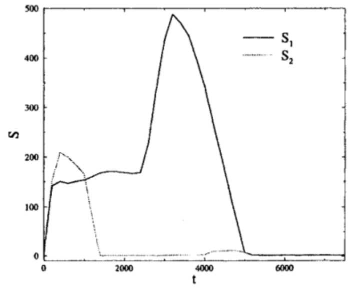


6

Specialization: Perturbation at t=2500







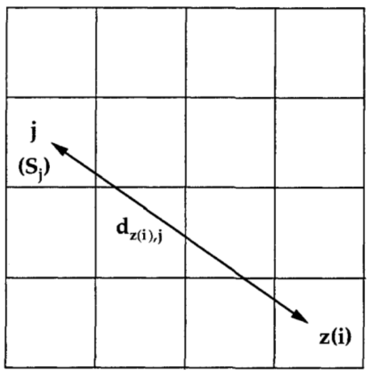
7

Differentiation In A Multiagent System or A Group of Robots

- Simple, homogeneous robots are easy to design and produce
- Response threshold model and reinforcement procedure used to differentiate identical agents
- Even fixed-threshold models can organize groups of robots
  - Puck-Foraging Task (Krieger)

8

Adaptive Task Allocation



Zone grid for mailman problem. Individual i is located in zone z(i) and responds to stimulus S_j from zone j at a distance $d_{z(i),j}$

Probability for agent i located in zone z(i) to respond to stimulus Sj

$$P_{ij} = \frac{S_j^2}{S_j^2 + \alpha\theta_{i,j}^2 + \beta d_{z(i),j}^2}$$ where $\theta_{i,j}(\in [\theta_{\min}, \theta_{\max}])$

S_j: Demand for task j

$\theta_{i,j}$: Response threshold for agent i on task j

$d_{z(i),j}$: Distance between z(i) and j

α, β: Tuning parameters to modulate $\theta_{i,j}$ and $d_{z(i),j}$ respectively

9

Adaptive Task Allocation

$$\theta_{i,j} \leftarrow \theta_{i,j} - \xi_0$$

$$\theta_{i,n(j)} \leftarrow \theta_{i,n(j)} - \xi_1, \ \forall n(j)$$

$$\theta_{i,k} \leftarrow \theta_{i,k} + \varphi \text{ for } k \neq j, k \notin \{n(j)\}$$

ξ_0, ξ_1: Learning coefficients associated with zone j and its neighbors

φ: Forgetting rate applied to all other zones

{n(j)}: Set of zones surrounding j

10

Adaptive Task Allocation

/* Values of parameters used in simulations */
$\alpha = 0.5$, $\beta = 500$, $\theta_{\text{min}} = 0$, $\theta_{\text{max}} = 1000$, $\xi_0 = 150$, $\xi_1 = 70$, $\delta = 50$, $\varphi = 10$, $m = 5$, $RS = 5$, $SW = 5$, $L = 5$

11

Adaptive Task Allocation

Algorithm 3.1 High-level description of zone allocation algorithm

```
/* Initialization */
For j = 1 to L × L do
  S_j = 0 /* initial demand in zone j is equal to 0 */
End For
For i = 1 to m do
  For j = 1 to L × L do
    θ_ij = 500 /* initialization of thresholds at neutral values */
  End For
  Place agent i in randomly selected zone z(i)
End For
```

12

Adaptive Task Allocation

```
/* Main loop */
For t = 1 to t_max do
  /* increase demand in randomly selected zone and assign agent if possible */
  For k = 1 to RS do
    Draw random integer number n between 1 and L × L /* select zone */
    If (zone n not covered) then
      S_n ← S_n + δ /* increase demand by δ in selected zone */
      Reponse = 0 /* no agent has responded to demand from zone n */
      Sweep = 1 /* first sweeping of all agents */
      Repeat
      For i = 1 to m do
        If (agent i available) /* agent i is not traveling */ then
          Draw real number r between 0 and 1
          Compute P_in /* Eq. (3.32) */
          If (r < P_in) then
            Zone z(i) is no longer covered
            Zone n is covered
            S_n ← 0
            Agent i unavailable for d_z(i)n time units
            z(i) ← n
          End If
        End If
      End For
      Sweep ← Sweep +1
      Until ((Response=1) or (Sweep = SW)
    End If
  End For
```

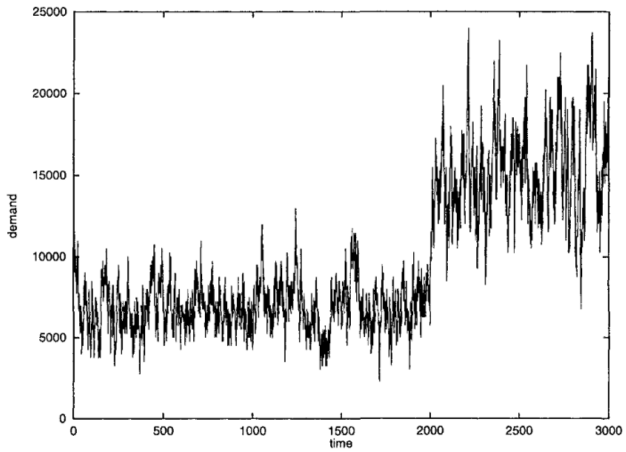
13

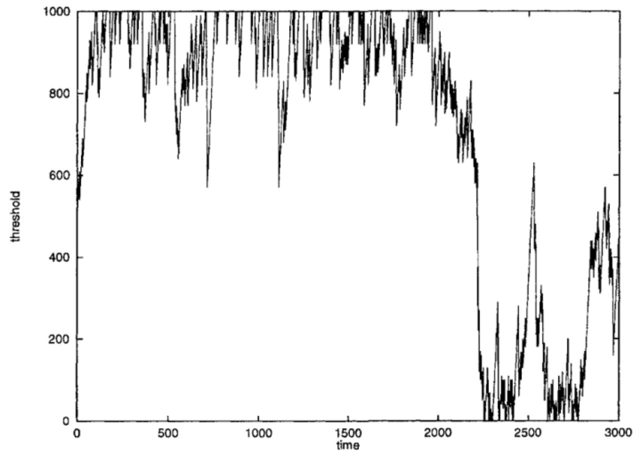
Adaptive Task Allocation

```
/* threshold updates */
For i = 1 to m do
  For j = 1 to L × L do
    If (j = z(i)) then
      θ_ij ← θ_ij − ξ_0 /* agent i "learns" zone j */
    Else If (j is in the neighborhood of z(i)) then
      θ_ij ← θ_ij − ξ_1 /* agent i "learns" zone j */
    Else
      θ_ij ← θ_ij + φ /* agent i "forgets" zone j */
    End If
  End For
End For
End For
```

14

Adaptive Task Allocation





15

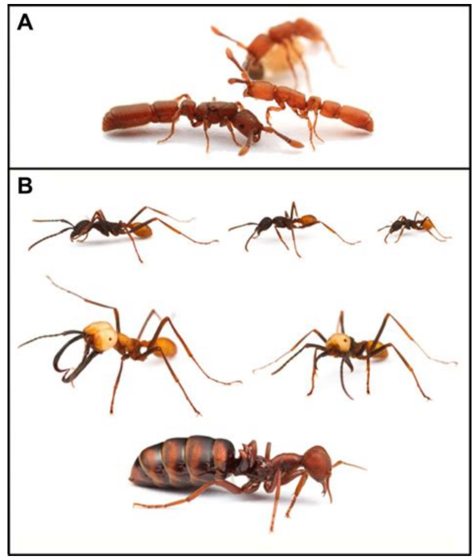
Adaptive Task Allocation: Connection to Bidding Algorithms

- Division of Labor algorithms inspired by nature share fundamental features with “market-based” algorithms
- Thermal Resource Distribution in a building can be solved with market-based control
- High bid in “market-based” algorithms is similar to a low response threshold in Division of Labor algorithms

16

3.6: Points to Remember

- Task allocation in swarms is dynamic
- Simple Models with response thresholds allow connection between individual and colony
- Changing response thresholds allows for differentiation of individuals
- Threshold models are useful for resource allocation in multi agent systems



17

Questions?

18

Specialization: Equations

$$T_{\theta_{ij}}(s_j) = \frac{s_j^2}{s_j^2 + \theta_{ij}^2}$$

Probability of individual i performing task j given the demand for j

Dynamics of the response threshold

$$\theta_{ij} \leftarrow \theta_{ij} - \xi \Delta t \quad \theta_{ij} \leftarrow \theta_{ij} + \varphi \Delta t$$

$$\theta_{ij} \leftarrow \theta_{ij} - x_{ij}\xi\Delta t + (1 - x_{ij})\varphi\Delta t$$

$$\partial_t \theta_{ij} = [(1 - x_{ij})\varphi - x_{ij}\xi]\Theta(\theta_{ij} - \theta_{\min})\Theta(\theta_{\max} - \theta_{ij})$$

Dynamics of task allocation percentages

$$\partial_t x_{ij} = T_{\theta_{ij}}(s_j)\left(1 - \sum_{k=1}^{m} x_{ik}\right) - p x_{ij} + \psi(i,j,t)$$

Stochastic dynamics to simulate local difference in conditions

$$\forall i,j,t \quad \langle \psi(i,j,t) \rangle = 0$$

$$\forall i,j,h,k,t,t' \quad \langle \psi(i,j,t)\psi(h,k,t') \rangle = \sigma^2 \delta_0(i-h)\delta_0(j-k)\delta_0(t-t')$$

Dynamics of task demand

$$\partial_t s_j = \delta - \frac{\alpha_j}{N}\left(\sum_{i=1}^{N} x_{ij}\right)$$

19

Specialization: Probability of Tasks

Demand for task j

Probability that agent i does task j →

$$T_{\theta_{ij}}(s_j) = \frac{s_j^2}{s_j^2 + \theta_{ij}^2}$$

Individual Response Thresholds

20

Specialization: Response Threshold

Learning Rate

Forgetting Rate

$$\theta_{ij} \leftarrow \theta_{ij} - \xi \Delta t \qquad \theta_{ij} \leftarrow \theta_{ij} + \varphi \Delta t$$

Fraction of time spent doing task j

$$\theta_{ij} \leftarrow \theta_{ij} - x_{ij}\xi\Delta t + (1 - x_{ij})\varphi\Delta t$$

Stepwise function to maintain range

$$\partial_t \theta_{ij} = [(1 - x_{ij})\varphi - x_{ij}\xi]\Theta(\theta_{ij} - \theta_{\min})\Theta(\theta_{\max} - \theta_{ij})$$

21

Specialization: Task Allocation Fraction

Number of tasks

Downtime

Random Variation by gaussian

$$\partial_t x_{ij} = T_{\theta_{ij}}(s_j)\left(1 - \sum_{k=1}^{m} x_{ik}\right) - p x_{ij} + \psi(i,j,t)\,.$$

$$\forall i,j,t \quad \langle \psi(i,j,t) \rangle = 0$$

$$\forall i,j,h,k,t,t' \quad \langle \psi(i,j,t)\psi(h,k,t') \rangle = \sigma^2 \delta_0(i-h)\delta_0(j-k)\delta_0(t-t')$$

22

Specialization: Task Demand

Efficiency

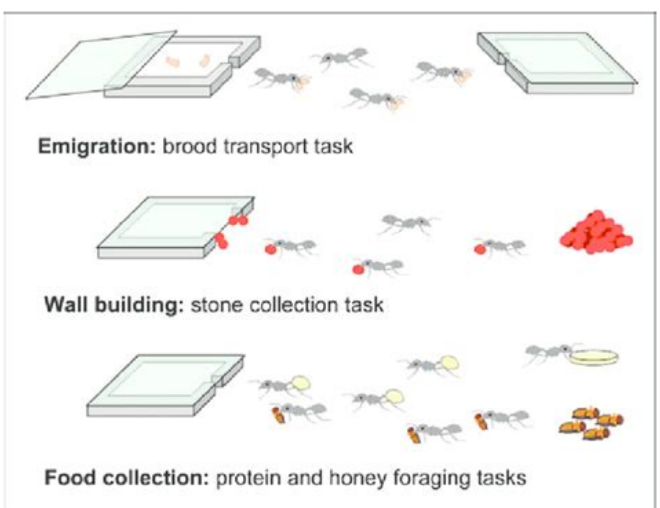
$$\partial_t s_j = \delta - \frac{\alpha_j}{N}\left(\sum_{i=1}^{N} x_{ij}\right)$$

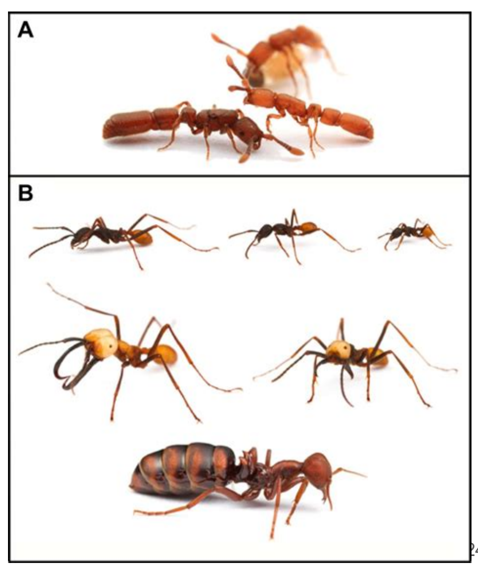
Increase Per Unit Time

Number of Agents

23

Introduction and Review of 3.1-3.3





24