Progetto dei Corsi di

Complex Systems and Network Science

Corso di Laurea Magistrale in Informatica

Analyzing Complexity

Corso di Laurea Magistrale in Relazioni Internazionali

A.A. 2019-2020

Dipartimento di Informatica — Scienza e Ingegneria
Università di Bologna

Ozalp Babaoglu Angelo Trotta

November 21, 2019

---

1 General Information about the Project

As part of your course requirement, you are to complete the project described below, which must be carried out individually. Submission of the project for evaluation must be done via email to the address: `angelo.trotta5@unibo.it`.

The deadline for submission is **23:59:59 hours on 13 January, 2020**. The email must have the subject field as *CSNS Project 2020* and must be sent from your University address (`name.surname@studio.unibo.it`).
You will receive a confirmation message within a few days of your submission. The email should contain an archive (in .zip or .tar.gz format) containing the following:

1. The source code that was developed (either in NetLogo or PeerSim);
2. A short paper, in PDF format, describing the model that was implemented, the experiments that were carried out using it, and a discussion explaining the results that were obtained.

Your full name, email address and student ID number (matricola) must be included in all of the source files, in the paper, and in the submission email that you send. The source code should be well documented and formatted, following good programming practices. The paper can be written in Italian or in English, and should be structured like a technical paper, thus containing a title, abstract and bibliography. It is strongly suggested that you limit the length to 16 pages and that you follow the Springer format for *Lecture Notes in Computer Science* (LNCS). Templates are available for both Word[1] and LaTeX[2]. You can use any text processing system you prefer (even though LaTeX is suggested) to write the paper as long as you submit the result as a PDF file.

The project must be done *individually*: no sharing of papers or source code is permitted. You are, of course, encouraged to discuss issues and solutions with fellow students or with the instructors.

2 Grading Policy

For your project to be satisfactory, it must satisfy the following requirements:

- The project must implement the specifications that follow. You are allowed (and encouraged) to apply modifications and extensions to the project, but they must be proposed to the instructors beforehand and approved by them.
- The model's implementation, and all of the related simulations and experiments, must be carried out using either the NetLogo or PeerSim software systems. If PeerSim is chosen, the cycle-driven simulation engine should

[1] Link to .doc template
[2] Link to .tex template

be used, and the simulator must be configurable by means of the standard PeerSim configuration file.

- Your paper must thoroughly describe the model that was implemented and justify all significant design decisions and extensions that were applied to it. You should also discuss the expected behavior of the model, by making previsions. Most importantly, you have to explain the experiments that you performed in terms of methodology and the results that you obtained. Significant implementation details can be inserted, if important in the context of the model, but should otherwise be kept as comments in the code itself.

You are encouraged to focus on a simple model and to apply extensions to it only if you completely understand the behavior of the base model. This can be achieved by working in modular fashion, thus incrementally (and carefully) adding new features, enriching your model. Ending up with a complex, unpredictable and difficult to understand model is very easy. On the contrary, you should prove through your experiments that you fully understand the behavior of your model and that you can interpret the results you obtained, and are able to relate them with real-world phenomena. Finally, you should try to find tipping points or interesting equilibrium states in your model.

If you are interested in these topics (e.g. you want to build better models or study other systems of this kind), do not hesitate to contact us when looking for a thesis topic.

3 Description of the Project

The purpose of this year's project is to create a model of *division of labor and task allocation* [1].

Many cases of social behaviours have a division of labor. The resilience of task allocation exhibited at the global level is connected to the elasticity of individual workers. The behavioral repertoire of workers can be stretched back and forth in response to perturbations. A model based on response thresholds connects individual-level plasticity with collectivity-level resiliency and can account for some important experimental results. Response thresholds refer to likelihood of reacting to task-associated stimuli. Low-threshold individuals perform tasks at a lower level of stimulus than high-threshold individuals.

An extension of this model includes a simple form of learning. Within individual workers, performing a given task induces a decrease of the corresponding threshold, and not performing the task induces an increase of the threshold. This double reinforcement process leads to the emergence of specialized workers, that is, workers that are more responsive to stimuli associated with particular task requirements, from a group of initially identical individuals. The fixed response threshold model can be used to allocate tasks in a multiagent system, in a way that is similar to market-based models, where agents bid to get resources or perform tasks. The response threshold model with learning can be used to generate differentiation in task performance in a multiagent system composed of initially identical entities. Task allocation in this case is emergent and more robust with respect to perturbations of the system than when response thresholds are fixed.

In social systems, different activities are often performed simultaneously by specialized individuals. This phenomenon is called division of labor. Simultaneous task performance by specialized workers is believed to be more efficient than sequential task performance by unspecialized workers. Parallelism avoids task switching, which costs energy and time. Specialization allows greater efficiency of individuals in task performance because they "know" the task or are better equipped for it.

A key feature of division of labor is its plasticity. Division of labor is rarely rigid. The ratios of agents performing the different tasks can vary in response to internal perturbations or external challenges. The agents force must be allocated to tasks so as to adjust to changing conditions.

3.1 Formal Specification

In order to explain these observations, we will use a simple model which relies on response thresholds. In this model, every individual has a response threshold for every task. Individuals engage in task performance when the level of the task-associated stimuli exceeds their thresholds.
When individuals performing a given task are withdrawn from the population (they have low response thresholds with respect to stimuli related to the task) the associated demand increases and so does the intensity of the stimulus, until it eventually reaches the higher response thresholds of the remaining individuals. The increase of stimulus intensity beyond threshold has the effect of stimulating these individuals into performing the task.

The nature of task-related stimuli may vary greatly from one task to another and so can information sampling techniques, which may involve direct interactions among agents, or more or less random exposure to task-related stimuli.

The first question that we have to answer is: how do we formally define a response threshold? Let s be the intensity of a stimulus associated with a particular task. s can be a number of encounters, a chemical concentration, or any quantitative cue sensed by individuals. A response threshold θ, expressed in units of stimulus intensity, is an internal variable that determines the tendency of an individual/agent to respond to the stimulus s and perform the associated task. More precisely, θ is such that the probability of response is low for $s \ll \theta$ and high for $s \gg \theta$. One family of response functions $T_\theta(s)$ (the probability of performing the task as a function of stimulus intensity s) that satisfy this requirement is given by Equation 1:

$$T_\theta(s) = \frac{s^n}{s^n + \theta^n} \tag{1}$$

where $n > 1$ determines the steepness of the threshold. A value of $n = 2$ is a good starting point for general scenarios, but extended results can be obtained with other values of $n > 1$. Figure 1 shows several such response curves, with $n = 2$,

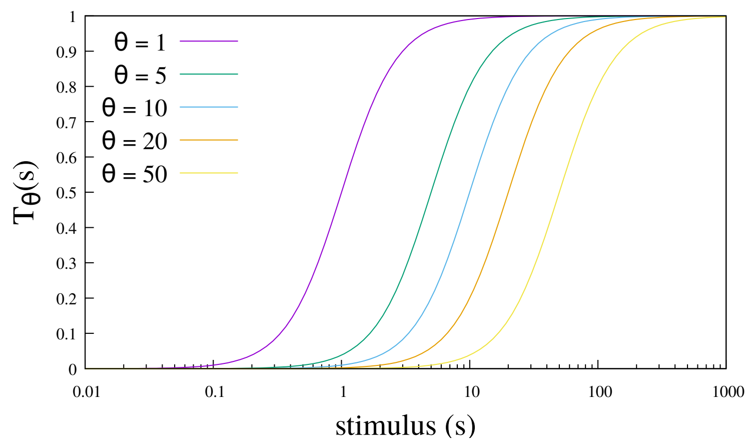

Figure 1: Semi-log plot of threshold response curves ($n = 2$) with different thresholds.

for different values of θ. The meaning of θ is clear: for $s \ll \theta$, the probability of engaging task performance is close to 0, and for $s \gg \theta$, this probability is close to 1. At $s = \theta$, this probability is exactly $1/2$. Therefore, individuals with a lower value of θ are likely to respond to a lower level of stimulus.

It is important to emphasize that threshold models encompass exponential response functions: the main ingredient of a threshold model is the existence, within each individual, of a value θ of the stimulus intensity s such that task performance is likely for $s \gg \theta$ and unlikely for $s \ll \theta$. Exponential response functions are particularly important because they are likely to be common. For example, an exponential response functions may be observed when there are waiting times involved, although it may not always be the case. Let us assume that tasks A and B are causally related in the sense that a worker performing task A has to wait for a worker performing task B. If a task-A worker has a fixed probability p per unit waiting time of giving up task-A performance, the probability that this worker will still be waiting after t time units is given by $P(t) = 1-(1-p)^t = 1-e^{t\ln(1-p)}$. In summary, response functions which exhibit an explicit threshold-like shape, such as the ones given by Equation 1 can be encountered in various situations and yield similar results. Viewed from the perspective of response thresholds, castes may correspond to possible physical differences, but also to innate differences in response thresholds without any visible physical difference.

3.1.1 One-Task Model

Assume that only one task needs to be performed. This task is associated with a stimulus or demand, the intensity s of which increases if it is not satisfied because either the task is not performed by enough individuals or it is not performed with enough efficiency. Let X_i be the state of an individual/agent

i: $X_i = 0$ corresponds to inactivity and $X_i = 1$ corresponds to performing the task (in this case ther is only task 1). Let θ_i be the response threshold of i. An inactive individual starts performing the task with a probability P per unit time:

$$P(X_i = 0 \to X_i = 1) = T_{\theta_i}(s) = \frac{s^n}{s^n + \theta_i^n} \tag{2}$$

The probability that individual i will perform the task depends on s, the magnitude of the task-associated stimulus (which also affects the probability of being exposed to it) and on θ_i, individual i's response threshold.

An active individual gives up task performance and becomes inactive with probability p per unit time:

$$P(X_i = 1 \to X_i = 0) = p \tag{3}$$

Parameter p is assumed to be identical for all individuals, $1/p$ is the average time spent by an individual in task performance before giving up the task. It is assumed that p is fixed and independent of stimulus. Individuals give up task performance after $1/p$ time units, but may become engaged again immediately if stimulus is still large.

Variations in stimulus intensity result from task performance, which reduces stimulus intensity, and from the natural increase of demand irrespective of whether or not the task is performed. The resulting equation for the discrete-time dynamics of stimulus intensity s is given by:

$$s(t+1) = s(t) + \delta - \frac{\alpha \cdot N_{act}}{N} \tag{4}$$

where N_{act} is the number of active individuals, N is the total number of potentially active individuals in the colony, δ is the increase in stimulus intensity per unit time, and α is a scale factor measuring the efficiency of task performance. The amount of work performed by active individuals is scaled by N to reflect the intuitive idea that the demand is an increasing function of N, that we take linear here.

3.1.2 Mulit-Task Model

Let us now proceed to the case of m tasks. By analogy with the previous case, generalizing to a milti-task scenario, we have s_j as the stimulus of performing task j by the agent i having the specific threshold $\theta_{i,j}$.
Then, we modify the Equation 2 as follow:

$$T_{\theta_{i,j}}(s_j) = \frac{s_j^n}{s_j^n + \theta_{i,j}^n} \tag{5}$$

$$P(X_i = 0 \to X_i = j) = \left(\dot{\sum}_{1 \le k \le m} T_{\theta_{i,k}}(s_k)\right) \cdot \frac{T_{\theta_{i,j}}(s_j)}{\sum_{1 \le k \le m} T_{\theta_{i,k}}(s_k)} \tag{6}$$

where $\dot{\sum}$ is the summation using the algebraic sum defined as $a\dot{+}b = a + b - ab$. Here, Equation 5 is similar to Equation 2 but customized for each task-stimulus (s_j) and depending on the threshold ($\theta_{i,j}$) that agent i has to perform task j.

In Equation 6 the first factor defines the probability to respond to at least one stimulus, while the second one defines the probability of executing the specific task k. We use a uniform value of probability p to leave the actual task (similar to Equation 3):

$$P(X_i = j \to X_i = 0) = p \tag{7}$$

And, finally, the stimulus evolution is:

$$s_j(t+1) = s_j(t) + \delta - \frac{\alpha \cdot (\sum_i N_{i,j})}{N} \tag{8}$$

where $N_{i,j}$, the number of workers of type i engaged in task j performance.

3.1.3 Specialization

The simple response threshold model introduced in the previous section, which assumes that each worker responds to a given stimulus when stimulus intensity exceeds the worker's threshold, can explain how flexibility at the colony level results from the workers' behavioral flexibility. Unfortunately, this model is of limited applicability because it assumes that workers' thresholds are fixed over the studied time scale.

In order to overcome these limitations, we use an extended version of the fixed-threshold model by allowing thresholds to vary in time, following a simple reinforcement process: a threshold decreases when the corresponding task is performed and increases when the corresponding task is not performed.

In addition to the fixed-threshold model, $\theta_{i,j}$ is now updated in a self-reinforcing way. The more individual i performs task j, the lower $\theta_{i,j}$, and vice versa. Let ξ and φ be the coefficients that describe learning and forgetting, respectively. In this time-incremental model, individual i becomes more (respectively less) sensitive by an amount $\xi\Delta t$ (respectively $\varphi\Delta t$) to task-j-associated stimuli when performing (respectively not performing) task j during a time period of duration Δt. The evolution of the threshold $\theta_{i,j}$ is:

$$\theta_{i,j} = \begin{cases} \theta_{i,j} - \xi\Delta t & \text{if agent } i \text{ executes task } j \\ \theta_{i,j} + \varphi\Delta t & \text{otherwise} \end{cases} \tag{9}$$

ξ and φ are assumed to be identical for all tasks, and the dynamics of $\theta_{i,j}$ is restricted to an interval $[\theta_{min}, \theta_{max}]$. Agent i is considered a specialist of task j if $\theta_{i,j}$ is small.

One important advantage of this strategy over the fixed-threshold strategy is that in the case where individuals with low thresholds are removed, the demand is quickly taken back to a much lower level because individuals with previously high thresholds lower their thresholds and become more responsive to task-associated stimuli. Flexibility and robustness can also be observed when task performance becomes less efficient (lower value of α). A lower value of α may be due to competition, changing environmental conditions, exhaustion of resources, or many other events.

3.2 The Project

Your task is to to model a complex system of your choice using the division of labor and task allocation described in the previous Sections.

More specifically, you have to choose a complex phenomenon studied in a scientific field of your choice (computer science, physics, biology, social sciences etc) and show how to model such phenomenon using the stimulus-response model. Some suggestions on possible topics are given below.

In your report you are supposed to give the reader the necessary background knowledge on the problem/phenomenon you have chosen, as well as motivations for that choice (e.g. why this is an interesting problem, which applications it has etc). The simulation/modeling of the phenomenon has to be justified as clearly as possible. For instance, you should give the full details of your encoding/simulation of the phenomenon as well as argue why this encoding/simulation is a good one. This is not a formal work and you are not supposed to formally proved the above statement: it could be enough to show that your encoding/simulation makes correct prediction on the behavior of the phenomenon.

If you have doubts about that please do not hesitate to contact us. Please keep in mind that this is not a research project. You are not supposed to come up with new applications of the stimulus-response model. Your task is to study in detail a (possibly already existing) application.

3.3 Projects Proposal

Here we list some possible topics for your final project. During the study of the complex system, analyze the evolution of the system (studyng all the used parameters, like the thresholds and the learning factors, and their impact on the system) and the behaviour during "stressing" events.
These examples can be also used as a cue for different types of problems.

3.3.1 A Bug's Life

In an ants' colony, the worker population is divided into two morphological subcastes: the minor and major sub-castes. Minors, which take care of most of the quotidian tasks of the colony, are smaller than the large-headed majors (often called soldiers), which are specialized either for seed milling, abdominal food storage, defense, or some combination of these functions. In the foraging task the ants search for food around the nest area and, once they found it, the food is transported in the nest. After, the food is elaborated and stocked where finally the larvae can be feeded. Sporadically, another species of ant comes close to the nest fighting for the territory.

3.3.2 City surveillance and maintainance

In a future smart city, a group of autonomous robots will patrol the city to accomplish multiple tasks. Let's assume that these robots use computer vision to accompish its task. Let's define two tasks for these robost: surveillance and cleaning. Each robot, while patrolling the city, uses its camera to recognize criminals or pollution, depending on the active task. The accomplishment of the task deends on the accuracy of its computer vision function. Here, we assume a *continual learning* algorithm that is able to improve its performance while using it on a specific task, but it *forgets* its ability to accomplish the other task.

3.3.3 IoT network

A wireless sensor network is deployed in a target scenario toaquire information data (temperaure, humidity, etc.). The sensors are battery powered and are connected in a wireless ad-hoc network. The acquired data must be uploaded to a service on the cloud. Each sensor is equipped with a LTE sim and hence, each sensors would be able to send its data directly to the cloud. However, there are different energy coss depending if the node is sensing the environment, taking part of the data routing and if becoming a gateway (activating the LTE sim).

References

[1] Bonabeau, E., Dorigo, M., Théraulaz, G. (1999). *Swarm intelligence: from natural to artificial systems* (No. 1). Oxford university press.