Simulating the Evolution of Neural Pathways and Structures

Development Journal

Kári Hlynsson[1] and Young Jun Lee[2]

[1]University of Iceland, Department of Mathematics
[2]University of Oxford, Department of Biology

Contents

1 Theoretical basis **4**
1.1 Note 4

2 Model outline **5**
2.1 Model outline 5
2.1.1 Overview 5
2.1.2 Sensory mapping of organisms 6
2.2 Simulation phases 7
2.3 Runtime optimization 8
2.3.1 Chunk system 8
2.4 Adjacent chunk loading and critical boundary 10
2.5 Performance comparison 11
2.5.1 Amortized cost of non-CBS implementation 11
2.5.2 Amortized cost of CBS implementation 11
2.5.3 Comparative analysis 11

A Preliminaries **13**
A.1 Probability theory and statistics 13
A.2 Neural networks 13

Index of notation

ABBREVIATION	DEFINITION
$\mathscr{E} = [0; w] \times [0; h]$	Environment with *width* w and *height* h. The environment can also be expressed as the set of locations ℓ such that $\mathscr{E} = \{\ell := (x, y) \mid x \in [0; w] \wedge y \in [0; h]\}$
n	Size of the population.
$E = (e_x, e_y)$	An entity with x and y coordinates in the environment such that $\ell_E \in \mathscr{E}$ (ℓ_E is the location which corresponds to the entity's location)
$\mathbf{C} = \{C_1, \ldots, C_k\}$	The partition of $\mathscr{E}$ into k chunks $C_1, \ldots, C_k$ such that $\bigcap_{i=1}^{k} C_i = \emptyset$ and $\bigcup_{i=1}^{k} C_i = \mathscr{E}$, i.e. the *set of chunks* $\mathbf{C}$ forms a complete partition of $\mathscr{E}$.
$\mathbf{O} = \{O_1, \ldots, O_n\}$	The population, the set of all organisms present within the environment.
$\mathcal{P} : \mathbf{O} \rightarrow \mathscr{E}$	Positional mapping. Returns point representation of an entity's location within $\mathscr{E}$. Note that $\mathcal{P}$ is a random variable, which we will discuss in more detail in the runtime optimization section.

Common Acronyms

Acronym	Definition
CBS	Chunk-Based System
UGP	Undirected-Graph Partitioning
BFS	Breadth-First Search

Chapter 1

Theoretical basis

1.1 Note

Hello there Jun
This is an EXTREMELY primitive draft.
None of this final and subject to changes as we cooperate on this project.
I also want to apologize for the common abbreviatons section, its a load of cowdung but I feel we will need this to make our lives easier later on.

Chapter 2

Model outline

2.1 Model outline

2.1.1 Overview

The aim of the model is to study the natural evolution of neural pathways in a population of organism when exposed to survivalistic conditions. A rigid logical and syntactical foundation will make all succeeding articulation on the model parameters and attributes easier. We therefore dedicate this first section towards establishing a foundation of terms and definitions which we build on later.

The most critical aspects of the model we define here is the *environment* and the *entities* contained therein. Neglecting any elevation, we define the environment as the bounded subset of the Cartesian plane, which we symbolize $\mathscr{E}$ [1].

Contained within the environment are *entities* which we can think of as actors within the simulation. The two types that occur in this model are *organisms* and *food*. Again, a simple intuitive definition is that the organism is an individual of a species present within the environment and nutrition is the foodstuffs which it consumes to gain energy and thus survive.

Entities can be divided into two types: *organisms* and *food*. What follows is simple: organisms are motile, can sense their surroundings and consume food to

[1]Although elevation certainly plays a vital role in the foraging patterns of organisms in natural environments, we refrain from its implementation as it only adds a level of complexity to the model design while having no immediate benefit for the simulation.

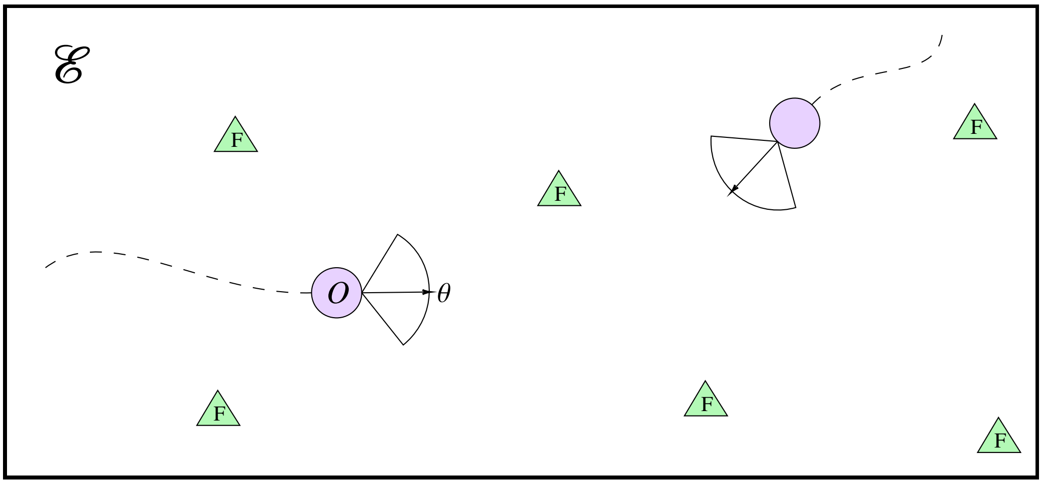

Figure 2.1: An illustration of the model

gain energy. On the other hand, food has none of these qualities. We represent an entity as the object E, while organisms are denoted O and food by F.

2.1.2 Sensory mapping of organisms

One of the key characteristics of organisms is that they are able to sense their proximal surroundings and base their succeeding actions on the information they have gathered on the environment. In this section we aim to establish a mathematical and syntactical foundation describing the sensory capabilities of organisms which allows passing environmental data to the organism's neural network.

A convenient and well established method of sensory mapping is obtained through the use of *raycasting* or *raylines*, where several line segments originating from the organism's point location are used as collision sensors which serve as sensors for distance. By calculating the distance of the intersection between some rayline emitted by an organism and an entity in the field, a metric describing the *sensory depth* from the organism to another entity is established.

Definition 1 (Ray set)**.** Let $\lambda \in \mathbb{R}^+$. Suppose that an organism O in an environment $\mathscr{E}$ with a present entity set $\mathbf{E}$ has the forward facing angle θ (see figure 2.1). We define the *ray set* of the organism as the linear space vector $\mathbf{R} = \{r_1, \ldots, r_{\nu_\mathbf{R}}\}$ from $[\theta - \Delta_\mathbf{R}; \theta + \Delta_\mathbf{R}]$ numbering $\nu_\mathbf{R}$ elements ($\nu_\mathbf{R}$ is called the *ray number*).

Furthermore, we define the quantity $\mathcal{S}_{\mathbf{R}} = 2\Delta_{\mathbf{R}}$ as the *span* of the ray set.

Definition 2 (Ray map)**.** The vector function R_λ is the *ray map* from the ray set $\mathbf{R}$ to the family of vectors bounded within the organism's sensory field. Furthermore, it is defined by

$$R_\lambda(r) = \left\langle R_\lambda^x, R_\lambda^y \right\rangle = \langle \lambda \cos r_i, \lambda \sin r_i \rangle$$

Where λ is the *maximum sensory depth*, i.e. the radius of the sensory field.

Remark. Note that

$$|R_\lambda(r)|^2 = \left(R_\lambda^x\right)^2 + \left(R_\lambda^y\right)^2 = \lambda^2$$

The vector function is thus the parametrization of a circle sector in the range $[\theta - \Delta_{\mathbf{R}}; \theta + \Delta_{\mathbf{R}}]$ and the ray set $\mathbf{R}$ returns a finite collection of vectors with their endpoints located on the sector.

Definition 3 (Sensory field)**.** The sensory field of an organism O is the set of vectors $\mathbf{S} = \{\vec{s}_1, \ldots, \vec{s}_{\nu_{\mathbf{R}}}\}$ which is returned by the ray map R_λ acting on the ray set $\mathbf{R}$. The *maximum sensory depth* is the parameter λ which describes the length of the vectors, i.e. $\forall \vec{s}_i \in \mathbf{S} : |\vec{s}_i| = \lambda$.

2.2 Simulation phases

For your contemplation (Jun, if you're reading this): I've thought of dividing the simulation into a *foraging phase*, where organisms roam around and collect food. If they don't get any or deplete their energy, they die. Once the foraging phase is over, the *reproductive phase* starts, where remaining energy is a measure of how likely organisms are to find a partner and reproduce (this is of course a simplication, there are many other ways to go about this I'm sure). This way, we don't have to make the reproduction itself an extreme pain (organisms having to find each other, etc.) This would mean that the reproductive phase is not carried out in the "plane" where the simulation occurs but rather "off screen" where its just a bunch of calculations really.

On the other hand it might make for some really interesting data if we were to assign individuals genders and they would map their current energy level and the gender of individuals in their sensory field and allow for them to reproduce "in the field" lol. Let me know what you think!

2.3 Runtime optimization

One of the run-ins we've had so far is determining how to design the sensory mapping capabilities of organisms within the environment. By sensory mapping, I am referring to the organism's ability to sense its proximal surroundings, sensing the proximity and types of the various entities they may encounter. This will be fed into their neural network, which outputs some response which instructs the organism how to behave given its current surroundings.

The first attempt I made was in the days where the environment was grid-based instead of a float-based environment. There, sensory mapping was quite easy as all that had to be done was inspect the proximal tiles and check for the entity type present in the tile. This is not possible in the float-based environment, so we propose another solution.

An excellent idea you came up with was the idea of partitioning the environment into separate chunks, which organisms restrict their sensory mapping to unless there sensory fields intersect another adjacent chunk (more on that later). We will start by discussing this idea, which as you will see, will be of great use.

2.3.1 Chunk system

In this section, we will be doing a mathematical analysis of the chunk system to see how it will benefit the simulation. To start off, we inspect what fundamental laws apply to this system.

Proposition 1. *Let $\mathscr{E}$ be an environment paritioned into k chunks such that $\mathbf{C} = \{C_1, \dots, C_k\}$. The probability of an entity being present in a generic chunk C_i equals $1/k$, i.e.*

$$\Pr\{E \in C_i\} = \frac{1}{k}$$

Proof. Let $\mathscr{E}$ be the space $[0; w] \times [0; h]$ with $\text{area}(\mathscr{E}) = wh$ and the partition $\mathbf{C}$.

Under the assumption that the chunks are of uniform size, we assume

$$\text{area}(C_i) = \frac{\text{area}(\mathscr{E})}{k} \tag{$*$}$$

for all $C_i \in \mathbf{C}$ where $i \in [1; k]$. Under conventional probability theory, we can express the probability of an entity being in a generic chunk as the area of that particular chunk over the area of the environment, i.e.

$$\begin{aligned} \Pr\{E \in C_i\} &= \frac{|C_i|}{|\mathscr{E}|} \\ &= \frac{\text{area}(C_i)}{\text{area}(\mathscr{E})} \\ &= \frac{1}{k} \end{aligned}$$

The result of the calculations above are immediate of the definition of the area of the chunks, which is derived in ($*$). □

Definition 4 (Chunk load)**.** The random variable $\mathcal{L}$, or the *chunk load* of some generic chunk C_i, denotes the number of entities contained within the chunk. Immediate of proposition 1, we have that $\mathcal{L} \sim \text{Bin}(n, 1/k)$, where n is the total number of entities in the environment. [2]

The random variables $\mathcal{L}_{C_1}, \ldots, \mathcal{L}_{C_k}$ are dependent, which by inference leads to $\sum_{i=1}^{k} \mathcal{L}_{C_i} = N$, where $|\mathbf{E}| = N$. Algorithm 1 shows a method with which a amortized cost model can be simulated.

The algorithm demands the assignment of $\mathcal{L}_{C_i}$ for chunks $C_i \in \mathbf{C}$ by a random process. However, given the nature of probabilistic distributions of dependent variables,

[2]Note that this assumes the uniform distribution of entities within the environment. While it not entirely safe to say that the distribution of entities is always uniform, we do so in order to create some upper bound for simulation time

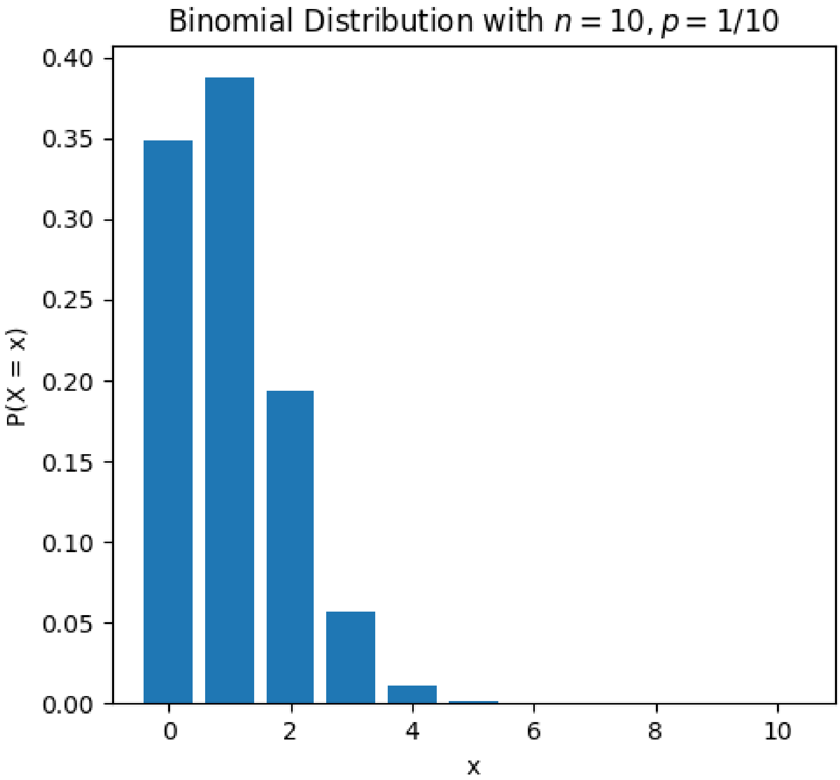

Figure 2.2: An example binomial distribution

2.4 Adjacent chunk loading and critical boundary

Although the chunk system minimizes the calculations needed to check for collisions with a ray, it introduces the risk of an entity escaping an organisms sensory field despite being contained within it. This is due to the fact that the CBS only performs calculations concerning entities contained within the chunk itself without paying attention to the contents of the sensory field.

A way to ensure that all entities within the sensory field are recognized is by introducing adjacent chunk loading (henceforth ACL) which loads the adjacent chunk given that an organisms field of view intersects an adjoining chunk.

Expanding on this concept, we loosely define the *critical boundary* as the subset of the environment, denoted $C_{\mathscr{E}}$, which suffices the condition that the sensory field of any organism contained within it intersect an adjoining chunk, regardless of the

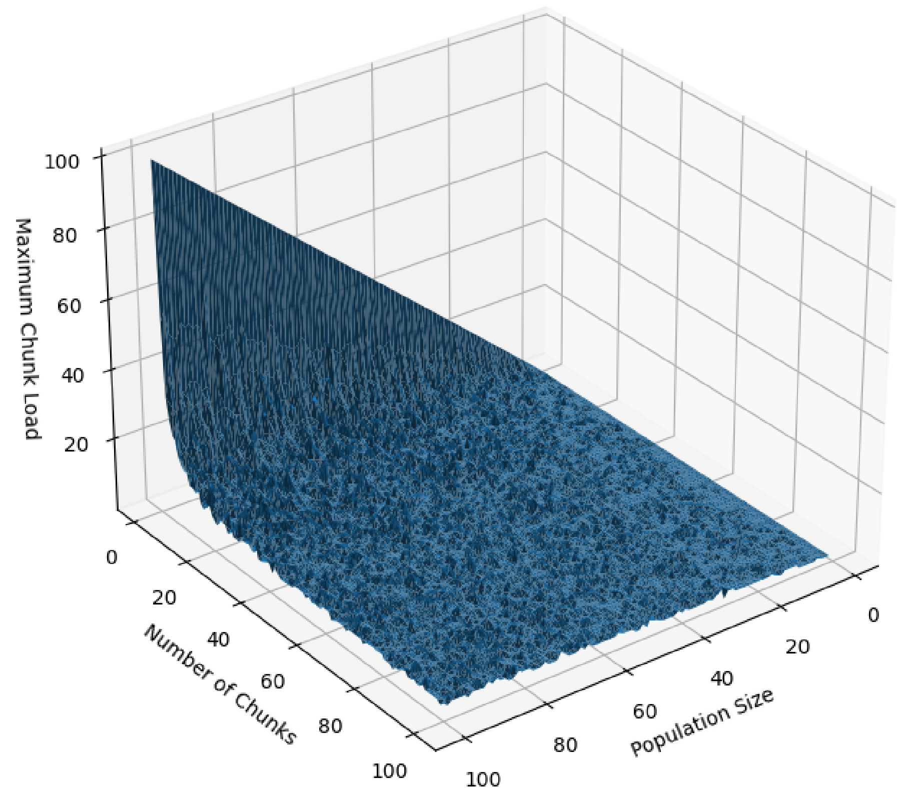

Figure 2.3: $\mathcal{L}_{\max}$ by population size and number of chunks

organism's position and orientation.

2.5 Performance comparison

In this section we compare the CBS versus non-CBS runtime performance to obtain a metric description of performance improvements as a result of the CBS implementation.

2.5.1 Amortized cost of non-CBS implementation

2.5.2 Amortized cost of CBS implementation

2.5.3 Comparative analysis

Algorithm 1 Algorithm for estimating amortized cost of CBS method

Require: The chunk set $\mathbf{C}$ which partitions $\mathscr{E}$ into k disjoint subsets $C_1, \ldots, C_k$ where $|\mathbf{C}| = k$. Entity set $\mathbf{E}$ within $\mathscr{E}$ where $|\mathbf{E}| = N$ with organism subset $\mathbf{O}$ such that $|\mathbf{O}| = n$.

procedure CbsCostModel($\mathbf{C}$, $\mathbf{E}$)
  $\text{cost}_{\text{CBS}} \leftarrow 0$ ▷ *Amortized CBS cost*
  for $C_i \in \mathbf{C}$ **do**
    Assign C_i chunk load $\mathcal{L}_{C_i} \sim \text{Bin}(N, 1/k)$ by random process
    $N \mathrel{-}= \mathcal{L}_{C_i}$ ▷ *Since $\mathcal{L}_{C_1}, \ldots, \mathcal{L}_{C_k}$ are dependent r.v.*
    $n_{O \in C_i} := |\{O \in \mathbf{O} \mid O \in C_i\}|$ ▷ $n_{O \in C_i} \leq n$
    $\text{cost}_{\text{CBS}} \leftarrow \text{cost}_{\text{CBS}} \mathrel{+}= n_{O \in C_i}(\mathcal{L}_{C_i} - 1)$
  $\text{cost}_{\text{CBS}} \leftarrow \text{cost}_{\text{CBS}} \mathrel{+}= k$

Appendix A

Preliminaries

A.1 Probability theory and statistics

Definition 5 (Probability)**.** Let ω be some event from the probability space Ω. We represent the probability of the event ω occuring using the notation $\Pr\{\omega\} = x$ where $x \in [0; 1]$.

Definition 6 (Random variable)**.** A random variable X is the mapping from the probability space Ω to the real number line, i.e. $X : \Omega \to \mathbb{R}$.

Remark. An example of a random variable is the varying height of a population, where Ω is the space of all possible outcomes and the random variable H (height) is for example 180 cm, or 157 cm.

Definition 7 (Expected value)**.** Let X be a random variable. The expected value is denoted $\mathbb{E}[X]$.

Definition 8 (Probability distribution)**.** Let X be a random variable. When it follows for example the Poisson distribution, we write $X \sim \text{Poisson}(\lambda)$.

A.2 Neural networks