

# Simulazione di un ecosistema Preda-Predatore

Mattia Marchi 817587  
Lorenzo Tomasoni 829906

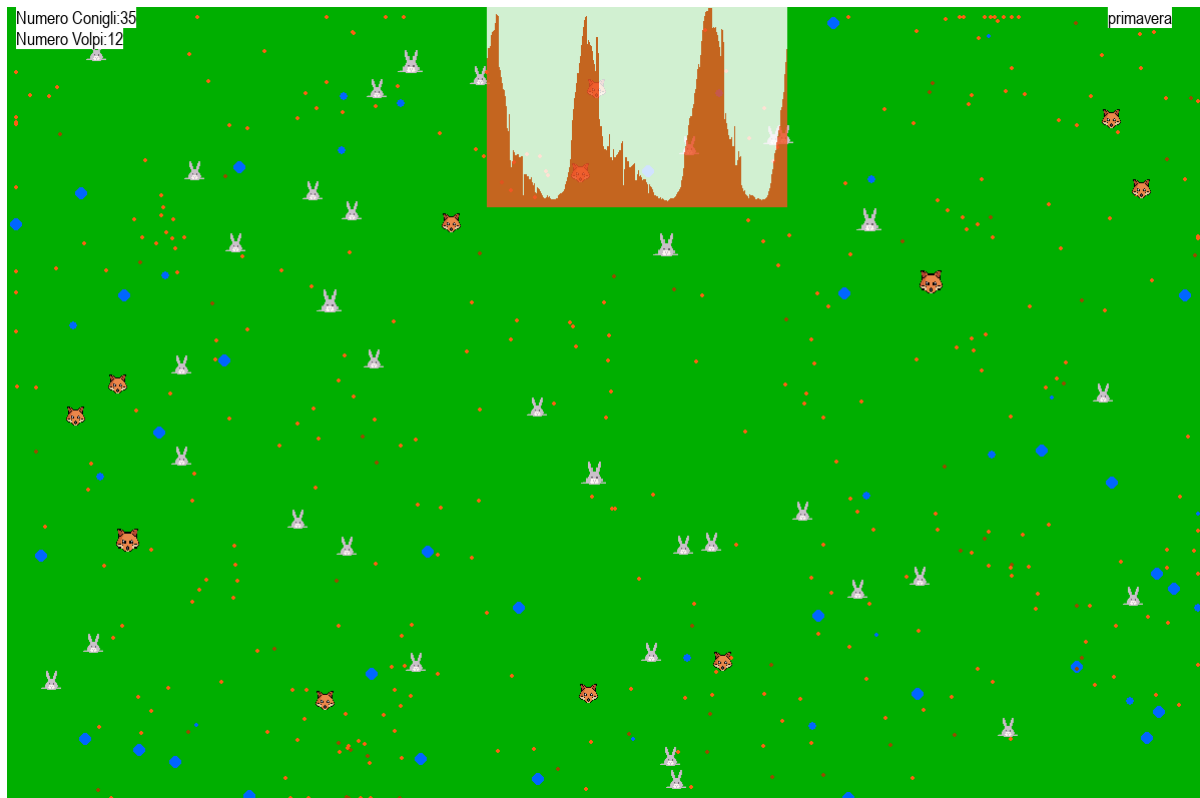
Sistemi Complessi: Modelli e Simulazione

Università degli Studi Milano- Bicocca  
Dipartimento di Informatica, Sistemistica e Comunicazione  
Anno Accademico 2020-2021

## Sommario

In questo documento viene illustrato un modello di simulazione multi-agente di un ecosistema naturale, caratterizzato dalla presenza di prede e predatori. Nella prima parte viene presentata una panoramica relativa allo stato dell'arte della simulazione dei sistemi complessi in generale e dei modelli presenti in letteratura dedicati alla simulazione di sistemi preda-predatore. In seguito, vengono illustrati il modello realizzato nelle sue diverse configurazioni, il processo di validazione e infine, l'analisi dei risultati ottenuti dalla simulazione.

La repository del progetto è consultabile al seguente link:  
[https://github.com/M-Marchi/Simulazione\\_Preda-Predatore](https://github.com/M-Marchi/Simulazione_Preda-Predatore)



# Indice

<b>1</b>	<b>Introduzione</b>	<b>4</b>
<b>2</b>	<b>Panoramica sugli ecosistemi naturali</b>	<b>4</b>
2.1	Cos'è un ecosistema naturale . . . . .	4
2.1.1	La volpe rossa . . . . .	4
2.1.2	Il coniglio selvatico . . . . .	6
2.2	La teoria della selezione naturale di Darwin . . . . .	7
<b>3</b>	<b>Stato dell'arte</b>	<b>8</b>
3.1	Sistemi complessi . . . . .	8
3.2	Simulazione di sistemi complessi . . . . .	8
3.3	Il modello Lotka-Volterra . . . . .	10
3.4	Modelli di simulazione di un ecosistema naturale multi-agente presenti in letteratura . . . . .	14
3.4.1	Modello preda-predatore agent based di Physics of Risk . .	14
3.4.2	Modello preda-predatore realizzati in NetLogo . . . . .	15
3.5	I limiti dei modelli presenti in letteratura . . . . .	17
<b>4</b>	<b>Obiettivo del progetto</b>	<b>18</b>
<b>5</b>	<b>Descrizione del modello</b>	<b>19</b>
5.1	Ambiente . . . . .	19
5.1.1	Implementazione . . . . .	20
5.2	Agenti . . . . .	23
5.2.1	Acqua . . . . .	23
5.2.2	Implementazione . . . . .	23
5.2.3	Carota . . . . .	24
5.2.4	Implementazione . . . . .	24
5.2.5	Coniglio e Volpe . . . . .	25
5.2.6	Implementazione . . . . .	25
<b>6</b>	<b>Ambiente di sviluppo e Codice</b>	<b>38</b>
<b>7</b>	<b>Analisi dei risultati</b>	<b>42</b>
7.1	Configurazione senza reintroduzione e senza evoluzione . . . . .	42
7.2	Configurazione con reintroduzione senza evoluzione . . . . .	46
7.3	Configurazione con reintroduzione ed evoluzione . . . . .	51
7.3.1	Validazione del modello . . . . .	59
7.4	Confronto tra configurazioni . . . . .	63
<b>8</b>	<b>Conclusioni e sviluppi futuri</b>	<b>65</b>

# 1 Introduzione

L'obiettivo del progetto è quello di realizzare una simulazione basata su agenti al fine di simulare il comportamento di un piccolo ecosistema naturale, costituito da una specie di predatori (volpi), una di prede (conigli), una vegetale (carote) e da alcuni specchi d'acqua (pozzanghere). Entrambe le specie animali presenti nell'ecosistema devono soddisfare alcuni bisogni fisiologici come il mangiare, il bere e il riprodursi. Per soddisfare la prima di queste necessità, le volpi si cibano dei conigli e i conigli a loro volta delle carote ed entrambi saziano la propria sete bevendo dalle pozzanghere che si formano nell'ambiente. Ogni singolo individuo è caratterizzato da alcune specifiche caratteristiche, quali soglia della fame, velocità di movimento, velocità di corsa, raggio di percezione, necessità di accoppiarsi e così via. Ogni individuo adulto è in grado di riprodursi e lo potrà fare solamente quando raggiunge il proprio periodo di fertilità ed incontra un altro individuo adulto della stessa specie. Ogni individuo che nasce in seguito all'accoppiamento erediterà i geni dai genitori ed inoltre il modello mette a disposizione la possibilità di simulare l'evoluzione genetica degli animali che popolano l'ambiente.

Il fine del lavoro è quello di valutare come le varie configurazioni sviluppate si comportino rispetto ai modelli di riferimento presenti in letteratura.

## 2 Panoramica sugli ecosistemi naturali

### 2.1 Cos'è un ecosistema naturale

Un ecosistema[1] può essere definito come una comunità dove esseri viventi di specie differenti coesistono nello stesso ambiente fisico e interagiscono vicendevolmente in modo da sviluppare un ciclo di vita e facilitare il flusso di energia e nutrienti. Nello specifico, l'**european environment agency** definisce un ecosistema naturale come una particolare categoria di ecosistema dove l'impatto umano non ha avuto un'influenza maggiore di quello di qualsiasi altra specie autoctona e non ha influenzato la struttura dell'ecosistema dalla rivoluzione industriale. L'impatto umano esclude cambiamenti di proporzioni globali, come il cambiamento climatico dovuto al riscaldamento globale [2]. Il termine ecosistema venne per la prima volta usato in una pubblicazione dall'ecologista inglese **Arthur Tansley** nel 1935[3].

Esistono due tipi di ecosistemi: quelli naturali e quelli artificiali. La prima categoria riguarda quelli che si sviluppano naturalmente e che possono svilupparsi e sopravvivere senza l'intervento umano. Ne sono esempi le foreste, le montagne, i fiumi, i prati e così via. Gli ecosistemi artificiali [4], invece, hanno caratteristiche in comune con quelli naturali, ma sono creati e mantenuti dagli esseri umani. Essi sono più semplici rispetto a quelli naturali e sono quelli che più comunemente circondano l'esistenza umana.

#### 2.1.1 La volpe rossa

La volpe rossa (*vulpes vulpes*) rappresenta il predatore scelto per la simulazione. Essa è la più grande delle volpi e il carnivoro più largamente diffuso, essendo pre-

sente in tutto l'emisfero boreale dal circolo polare artico all'Africa settentrionale, il nord America e l'Eurasia [5]. Questi animali possono misurare dai 75 ai 140 cm, per un peso che varia tra i 3 e gli 11 Kg. Il colore varia tra il giallo e il marrone e spesso è rossiccio.

La volpe rossa vive generalmente in coppia con i cuccioli, ma talvolta è possibile osservare degli esemplari che vivono solitari oppure in gruppi di 4-6 adulti. Generalmente le volpi rosse cacciano in modo solitario e difendono il proprio territorio da sole durante l'estate e in coppia durante l'inverno.

L'alimentazione della volpe rossa, nonostante sia classificato come animale carnivoro, comprende sia animali che vegetali. La dieta di questo animale è basata su una grande quantità di specie, dagli invertebrati ai piccoli mammiferi, uova rettili e piccoli anfibi. Tra i vegetali, la volpe rossa si ciba di vari tipi di frutti di bosco. In particolare, essa caccia i conigli appostandosi in modo furtivo e silenzioso per poi balzare tramite un rapido scatto sulla preda.

Il periodo degli amori è molto variabile e cambia secondo la latitudine: in Italia ha luogo in inverno, tra dicembre e febbraio. I parti avvengono generalmente tra marzo e aprile. La femmina, dopo una gestazione di 7 settimane, partorisce in media da 3 a 5 piccoli, che vengono allattati per un mese. Le volpi si riproducono una sola volta all'anno.

In natura, le volpi rosse possono raggiungere l'età di 12 anni.

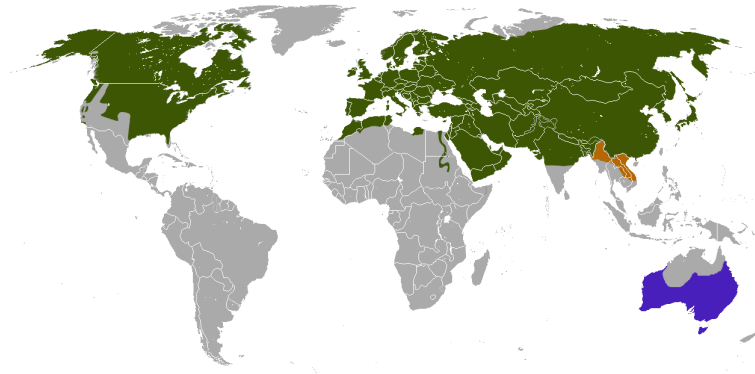


Figura 1: Areale della volpe. Le zone in verde indicano le aree dove la volpe è nativa, quelle blu dove è introdotta e quelle in arancione dove la presenza della volpe è dubbia.

Osservazioni sul campo[6] suggeriscono che le volpi siano miopi; possono attraversare la vegetazione senza incidenti, ma si avvicineranno agli oggetti fermi entro pochi metri a meno che un altro senso (come l'udito o l'olfatto) non le avvertano del pericolo o l'oggetto si muova.

Le volpi hanno due orecchie altamente mobili, che possono essere mosse indipendentemente l'una dall'altra. Esse possono ruotare ciascun orecchio di circa 150 gradi in un'unica direzione, l'orecchio destro ruota in senso orario e l'orecchio sinistro in senso antiorario, per captare i suoni che provengono dai lati o dallo spa-

zio retrostante. L'udito della volpe è molto sensibile ai suoni a bassa frequenza, come i fruscii emessi dalle prede.

Per quanto concerne l'olfatto, la volpe rossa sfrutta questo senso non solo per identificare le prede, ma anche per comunicare con i propri simili e delimitare il proprio territorio.

### 2.1.2 Il coniglio selvatico

Il coniglio selvatico europeo (*Oryctolagus cuniculus*) rappresenta la preda scelta per la simulazione. Esso [7] è un mammifero della famiglia dei leporidi, diffuso in gran parte dell'Europa (dal Portogallo alla Polonia, in Gran Bretagna e in alcune parti della Norvegia, Svezia e Ucraina), oltre al Nord Africa. Inoltre, i conigli selvatici sono stati introdotti in Australia, Nuova Zelanda, Cile ed in numerose isole.



Figura 2: Areale del coniglio selvatico.

Il coniglio selvatico predilige ambienti aperti, con clima secco e mite, ad altitudine non troppo elevata: il suolo dev'essere soffice o sabbioso, in modo da permettere all'animale di scavarsi la tana. Misura fino a 45 cm di lunghezza, per un peso che raggiunge i 2,5 kg: i maschi sono generalmente più grossi e robusti delle femmine. Si tratta di animali principalmente notturni e fortemente gregari, che possono vivere in colonie di grandezza direttamente proporzionale alla disponibilità di cibo. Una colonia tipo è composta da una decina di individui, senza distinzione di sesso.

I conigli selvatici sono animali erbivori che si nutrono di diversi tipi di vegetali, dall'erba alle foglie e le carote.

I conigli sono famosi per la propria capacità riproduttiva: la femmina va infatti in estro ogni 21 giorni e tende a riprodursi durante i primi mesi dell'anno, anche se in condizioni favorevoli può allevare i piccoli durante tutto l'anno al ritmo di una cucciolata al mese.

La gestazione dura un mese, al termine del quale vengono dati alla luce dai 3 ai 14 cuccioli. La femmina del coniglio può tornare a riprodursi appena dopo il parto[8]. La speranza di vita di questi animali è di circa 9 anni, anche se in generale il 90% degli esemplari muore prima di raggiungere l'anno d'età, a causa della forte pressione dei predatori.

## 2.2 La teoria della selezione naturale di Darwin

Uno degli obiettivi del progetto è stato quello di simulare come l'evoluzione genetica e l'adattamento rispetto alle condizioni dell'ambiente influenzino la competizione tra le prede e i predatori.

Il concetto di selezione naturale è stato introdotto per la prima volta dal biologo, naturalista e antropologo britannico **Charles Darwin** nel 1859 nella famosa opera *L'origine delle specie*[9]. Si tratta di un meccanismo alla base dell'evoluzione, secondo il quale il numero di individui che dimostrano di avere caratteristiche che si adattano in modo ottimale all'ambiente in cui vivono tende a aumentare in modo progressivo nel lungo periodo. Gli individui di una specie si distinguono per alcune caratteristiche genetiche. La teoria della selezione naturale afferma che tale variabilità è causata da mutazioni genetiche casuali, ovvero non causate dall'adattamento degli individui all'ambiente. Con il passare del tempo, vengono selezionate (ovvero favorite) quelle mutazioni che permettono agli individui di adattarsi meglio all'ambiente in cui vivono.

Gli individui che si adattano meglio a un certo habitat hanno la possibilità di procacciarsi più facilmente il cibo e si accoppieranno più facilmente rispetto agli altri individui della stessa specie che non presentano tali caratteristiche. I geni che fanno sì che l'individuo si adatti bene all'ambiente verranno trasmessi di generazione in generazione, mentre quelli inutili o dannosi vengono scartati. Questo porta all'evoluzione della specie che svilupperà caratteristiche che la rendono adatta a vivere e sopravvivere nell'ambiente.

Uno dei più famosi esempi relativo alla selezione naturale è quello riguardante il collo delle giraffe. Nel corso di centinaia di migliaia di anni le mutazioni genetiche fecero in modo che alcuni individui di giraffe avessero un collo più lungo. Questa caratteristica si rivelò molto vantaggiosa perché permise a questi individui di raggiungere con agilità le foglie degli alberi più alti. In situazioni di scarsità di cibo questa caratteristica si rivelò determinante: gli individui con il collo lungo avevano maggiore probabilità di sopravvivere, di raggiungere l'età della riproduzione e di conseguenza di riprodursi e di trasmettere i propri geni (e di conseguenza anche la peculiarità del collo lungo) alle generazioni successive.

I principi generali su cui si basa la teoria della selezione naturale, che verranno utilizzati per la creazione del modello utilizzato nella simulazione, sono i seguenti:

- **Principio della variazione:** tra gli individui di una stessa popolazione esiste una variabilità di caratteri;
- **Principio dell'adattamento:** gli individui che meglio si adattano all'ambiente hanno dei caratteri che offrono un vantaggio in termini di sopravvivenza e di riproduzione;
- **Principio dell'ereditarietà.** Si tratta del principio secondo il quale le caratteristiche trasmissibili ai discendenti per mezzo della riproduzione sono localizzate all'interno dei geni degli individui.

## 3 Stato dell'arte

### 3.1 Sistemi complessi

In letteratura esistono svariate definizioni, più o meno formali, di sistema complesso. In via del tutto generale, possiamo definire un sistema complesso come un sistema altamente strutturato, dinamicamente in evoluzione nel corso del tempo, il cui funzionamento è spesso complesso da spiegare o comprendere e in cui si verificano frequentemente interazioni multiple tra differenti componenti del sistema stesso. Alcune tipiche caratteristiche dei sistemi complessi sono le seguenti:

- Essi sono formati da elementi in continua interazione. Questi elementi prendono il nome di **agenti**;
- La **non linearità**. Spesso emerge che nei sistemi complessi non esiste un rapporto lineare tra l'input e l'output.
- Possibilità di evolvere ed adattarsi nel tempo.
- Robustezza e plasticità.

Oltre ai sistemi complessi, possiamo definire altre due categorie di sistemi: i sistemi semplici e quelli complicati. La categoria dei sistemi semplici comprende quelli le cui relazioni causa-effetto sono ben note e documentate, oltre che prevedibili e ripetitive. Si può affermare che i sistemi semplici sono sistemi noti e conosciuti. Ne è un esempio la bicicletta. Un sistema complicato invece, è un sistema molto simile dal punto di vista delle relazioni causa-effetto rispetto ai sistemi semplici; la differenza risiede nel fatto che i sistemi complicati sono a più ampia scala e richiedono expertise specializzate. Sono sistemi che potremmo definire conosciuti e ne sono esempi le macchine, i razzi spaziali, i treni e così via. Infine, i sistemi complessi si differenziano dalle altre due categorie a causa del fatto che le relazioni causa-effetto possono essere apprese in retrospettiva, ma difficilmente risultano essere riproducibili o predicibili. Questa categoria è analizzabile seguendo l'approccio della simulazione. Ne sono esempi gli organismi viventi, il sistema nervoso umano e, ovviamente, gli ecosistemi naturali che sono oggetto del progetto descritto in questo documento.

### 3.2 Simulazione di sistemi complessi

Come già affermato precedentemente, i sistemi complessi sono difficili da studiare e analizzare con un approccio teorico, è quindi opportuno utilizzare un approccio basato sulla simulazione.

Gli scopi per cui si fa simulazione sono vari e possono essere riassunti nei seguenti punti:

- Valutare piani e progetti operativi senza la necessità di metterli in atto nel mondo reale. Questo punto è molto importante perché attraverso la simulazione è possibile evitare di mettere in pratica un piano operativo per



poi accorgersi che non può essere portato a termine in modo corretto. Questo permette di risparmiare risorse e fatica.

- Valutare teorie e modelli. Questo significa che la simulazione permette di comprendere meglio il sistema che stiamo studiando, valutandone le relazioni di causa-effetto che lo fanno evolvere in modo dinamico.
- Osservare un sistema che non è ancora stato realizzato nel mondo reale e che pertanto non è osservabile. Questa situazione si può verificare nel momento in cui il sistema è in via di sviluppo.

Come per tutti gli strumenti scientifici, anche per fare simulazione non esiste un unico approccio. Il primo approccio è quello macroscopico. Utilizzando questa tecnica, vengono specificati dei vincoli globali, rappresentati sotto forma di equazioni. Un esempio è il vincolo della conservazione della massa che impone che, qualunque cosa succeda nel sistema, la quantità di massa totale dovrà mantenersi costante nel tempo. Nei modelli macroscopici vengono stilate delle equazioni differenziali che indicano come variano le quantità aggregate nel corso del tempo. Si assume che gli individui siano tutti uguali e abbiano lo stesso comportamento e lo stesso atteggiamento. Lo svantaggio di questo approccio è che rende difficile catturare tutti gli aspetti della dinamica del gruppo di agenti che compongono il sistema. L'approccio macroscopico è particolarmente indicato per problemi di ottimizzazione in contesti specifici. Per esempio, questo metodo potrebbe rivelarsi utile per simulare l'evacuazione di un edificio e per verificare se tutte le persone all'interno di esso siano in grado di evacuare in meno di 10 minuti. Uno degli esempi più famosi di modelli di simulazione macroscopici è il modello di Lotka-Volterra, descritto nella sezione 3.3.

L'approccio alternativo prende il nome di simulazione basata sugli agenti o modellazione microscopica. Con questa metodologia l'unità analitica è costituita dal singolo individuo (che prende appunto il nome di agente) e non più dalle variabili aggregate come nell'approccio macroscopico. Il vantaggio sta nel fatto che i risultati dovrebbero evidenziare esattamente le stesse dinamiche descritte dai modelli macroscopici, oltre ad altre informazioni aggiuntive relative per esempio all'aspetto spaziale. Per esempio, nel contesto di un ecosistema naturale, alcuni modelli microscopici permettono di comprendere con un buon livello di confidenza dove si collocano i predatori e quali percorsi tendono a seguire. L'aspetto negativo riguarda il fatto che per realizzare un modello di questo tipo è necessario conoscere alcuni aspetti aggiuntivi del sistema e degli agenti che lo compongono. Per esempio, sempre prendendo in considerazione il modello dell'ecosistema naturale, è necessario conoscere come sono fatti i predatori, quali sono le loro caratteristiche, come cacciano, quali bisogni hanno e per le prede conoscere di cosa si cibano, come scappano o si nascondono dai predatori e così via.

A prescindere dall'approccio che si usa, il ciclo di vita di un modello di simulazione è ben delineato. Infatti, per effettuare la corretta simulazione di un sistema complesso in generale si segue il ciclo di sviluppo riportato in figura 3.

Come si vede dall'immagine, le fasi dello sviluppo di una simulazione di un sistema complesso sono le seguenti:

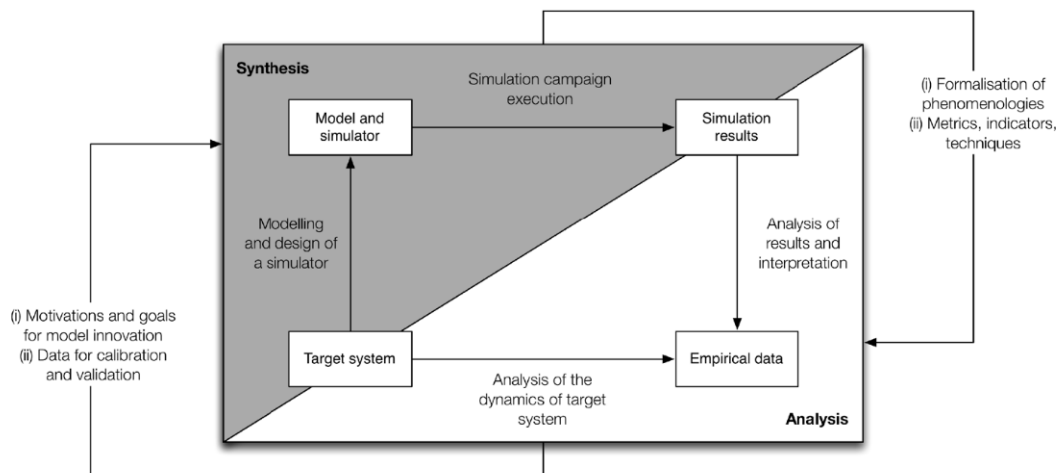


Figura 3: Schema che rappresenta il ciclo di sviluppo di una simulazione di un sistema complesso

- Definizione di un sistema target di riferimento, derivante dal processo interdisciplinare di analisi del fenomeno che si vuole studiare attraverso la simulazione.
- Operazione di modellazione e progettazione di un modello concettuale e di un simulatore del sistema in esame.
- Esecuzione di simulazioni. Questo passaggio permette di acquisire una grande quantità di dati che risultano essere particolarmente utili nella fase successiva.
- Analisi dei risultati e validazione. In quest'ultima fase si sfruttano i dati raccolti dalle simulazioni effettuate e li si confrontano con i dati reali collezionati all'inizio del ciclo. Se i risultati della simulazione coincidono con le dinamiche osservate nel sistema reale (ovvero se il ciclo commuta), è possibile sfruttare il modello realizzato per fini esplicativi oppure predittivi; viceversa se dal modello sviluppato si ottengono dati che non coincidono con le osservazioni raccolte è necessario ripetere il ciclo e adattare il modello, sistemandone alcuni aspetti.

### 3.3 Il modello Lotka-Volterra

Il modello di Lotka-Volterra, noto anche come equazioni o modello preda-predatore, è il modello matematico più noto di analisi dei sistemi di tipo preda-predatore. Questo modello è stato proposto in modo indipendente dal matematico americano **Alfred J. Lotka** nel 1925 e dal fisico e matematico italiano **Vito Volterra** nel 1926.

Il modello venne inizialmente proposto da **Alfred J. Lotka** nella teoria delle reazioni chimiche autocatalitiche pubblicata nel 1910 [10]. Successivamente, dieci anni più tardi, Lotka estese il proprio modello, grazie anche al contributo del matematico sovietico **Andrey Kolmogorov**, a sistemi organici, utilizzando come esempio piante e specie di animali erbivori [11]. Infine, nel 1925 il matematico sfruttò il modello per analizzare l'interazione tra prede e predatori in un ambiente naturale[12]. L'anno successivo Vito Volterra si interessò allo stesso argomento poiché desiderava rispondere alla questione sollevata dal biologo **Umberto d'Ancona**[13]. Egli aveva osservato che le percentuali di pesci cacciatori catturati dai pescatori erano aumentate durante la prima guerra mondiale. Questa osservazione lo stupì dato che durante gli anni di guerra la pesca venne praticata in misura molto minore rispetto agli anni precedenti. Volterra sviluppò il proprio modello in modo del tutto indipendente da Lotka e lo utilizzò per spiegare questo apparente paradosso.

Il modello di Lotka-Volterra[14] descrive la dinamica di interazione fra due specie che convivono in uno stesso ambiente e sono in interazione tra di loro, definendo le seguenti ipotesi, che non sono necessariamente realizzabili in natura:

1. Le prede hanno sempre a disposizione una quantità illimitata di cibo. In altre parole, questo significa che le prede non potranno mai morire a causa della scarsità di cibo.
2. I predatori si cibano esclusivamente delle prede.
3. Il tasso di crescita delle popolazioni di entrambe le specie è proporzionale rispetto al numero di individui che compongono popolazioni stesse.
4. L'ambiente, con il trascorrere del tempo, non evolve a favore di nessuna delle sue specie e i cambiamenti genetici sono irrilevanti.
5. La necessità di cibarsi dei predatori è limitata.

Sotto queste ipotesi, il modello di Lotka-Volterra prende la forma di un sistema di equazioni differenziali non lineari del primo ordine:

$$\begin{cases} \frac{dx}{dt} = \alpha x(t) - \beta x(t)y(t) \\ \frac{dy}{dt} = \delta x(t)y(t) - \gamma y(t) \end{cases}$$

Dove:

- $x(t)$  e  $y(t)$  rappresentano rispettivamente il numero di prede e predatori che popolano il sistema al tempo  $t$ .
- I parametri  $\alpha$  e  $\gamma$  indicano il tasso di crescita rispettivamente delle prede e dei predatori. In altre parole, indicano la differenza tra il tasso di nascita e il tasso di morte delle due specie.
- Il parametro negativo  $-\beta$  indica come diminuiscono le prede in funzione dei predatori.

- Il parametro positivo  $\delta$  indica come crescono i predatori in funzione delle prede.

Nel caso specifico di un sistema che presenti solo prede, ricordando che esse hanno la possibilità di riprodursi senza limiti, l'unica cosa che può farne diminuire il numero sono i predatori. In questa situazione il sistema si può tradurre in un'unica equazione, che assume la seguente forma:

$$\frac{dx}{dt} = \alpha x(t)$$

Anche in questo caso il parametro positivo  $\alpha$  indica il tasso di crescita delle prede. La soluzione a tale equazione è

$$x(t) = x(0)e^{\alpha t} \quad (1)$$

Dove  $x(0)$  rappresenta il numero di prede presenti nel sistema al tempo 0. Analizzando il grafico della funzione (1) riportato in **figura 4**, è possibile notare che la funzione è monotona crescente. Questo implica che le prede tendono a crescere in numero, dato che l'unico fattore che ne potrebbe limitare la crescita è rappresentato dai predatori.

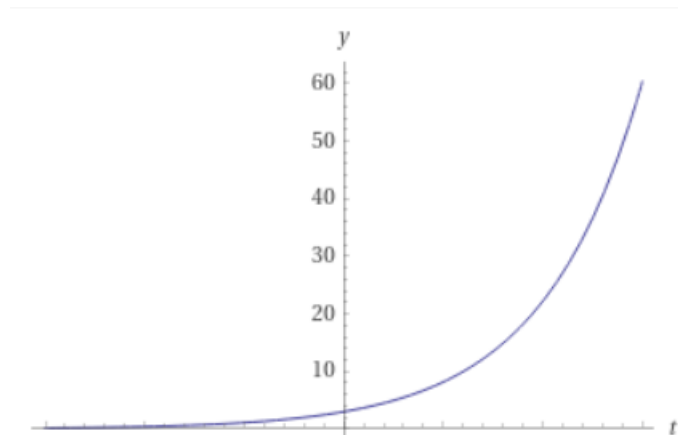


Figura 4: Grafico della funzione (1) fissati i parametri  $x(0)$  e  $\alpha$  rispettivamente a 3 e 2

Studiando invece, il caso specifico in cui il sistema sia abitato esclusivamente da predatori, il sistema di equazioni differenziali si riduce, anche in questo caso, ad una sola equazione che assume la seguente forma:

$$\frac{dy}{dt} = -\gamma y(t)$$

Ancora una volta, il parametro negativo  $-\gamma$  indica il tasso di morte dei predatori. Risolvendo l'equazione, otteniamo il seguente risultato:

$$y(t) = y(0)e^{-\gamma t} \quad (2)$$

Dove  $y(0)$  rappresenta il numero di predatori che popolano il sistema al tempo 0. Analizzando il grafico della funzione (2) riportato in **figura 5**, è possibile notare che la funzione è monotona decrescente. Questo implica che i predatori continueranno a diminuire in numero fino a estinguersi completamente.

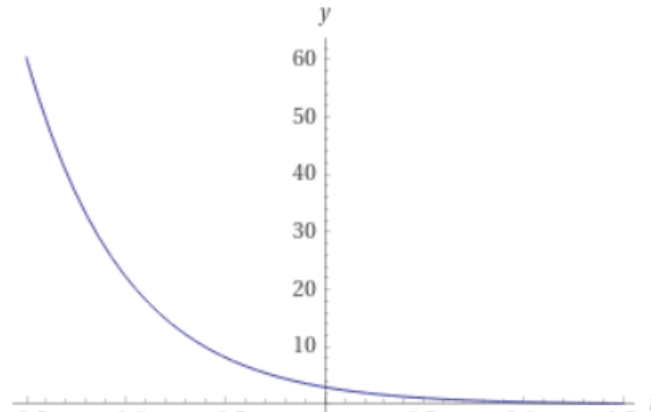


Figura 5: Grafico della funzione (2) fissati i parametri  $y(0)$  e  $\gamma$  rispettivamente a 3 e 2

Il modello Lotka-Volterra generale, descritto dal sistema 3.3 presenta il seguente andamento:

- Il numero di prede tende a crescere fintantoché il numero di predatori rimane basso;
- Al crescere delle prede, cresce anche il numero di predatori;
- Quando il numero di predatori è elevato e superiore rispetto a una certa soglia, il numero di prede tende a calare;
- Il diminuire delle prede viene accompagnato dalla contestuale diminuzione del numero di predatori;
- Nel momento in cui il numero di predatori è troppo basso, il numero di prede tende a risalire;
- tutto il ciclo si ripete;

Le soluzioni al sistema di equazioni non possiedono una espressione semplice in termini delle funzioni trigonometriche. Se si approssimano le soluzioni che si trovano in un intorno del punto di equilibrio stabile linearizzando il sistema, si ottiene un moto armonico semplice in cui la popolazione delle prede precede quella dei predatori con uno sfasamento di  $\frac{\pi}{2}$ , come visibile nella **figura 6**[15].

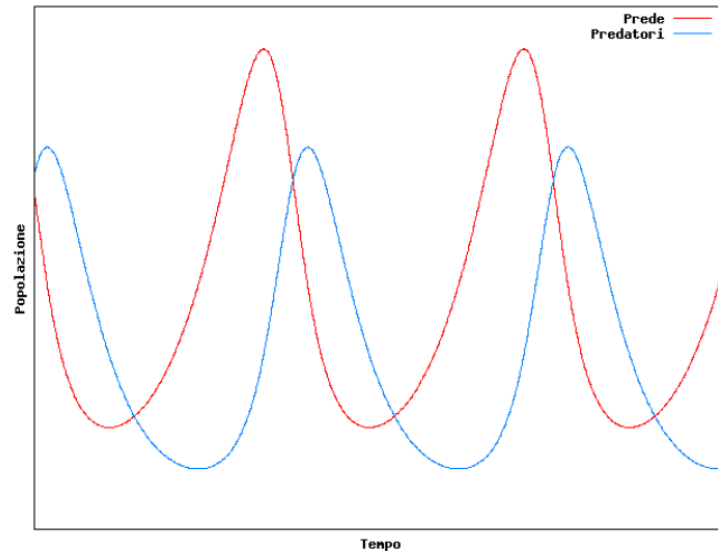


Figura 6: Andamento del sistema Lotka-Volterra

### 3.4 Modelli di simulazione di un ecosistema naturale multi-agente presenti in letteratura

In questa sezione viene presentata una panoramica di alcuni dei modelli di simulazione multi-agente di un ecosistema naturale presenti in letteratura.

Al termine delle loro descrizioni, vengono messi in evidenza i punti di carenza di questi modelli. L'obiettivo del progetto è quello di superarli e quindi realizzare un modello più realistico rispetto a quelli esaminati.

#### 3.4.1 Modello preda-predatore agent based di Physics of Risk

Il primo modello di simulazione di un ecosistema naturale che verrà presentato è quello disponibile al seguente link Physics of Risk<sup>1</sup>.

Si tratta di un semplice modello realizzato sotto forma di applet HTML. In questo sistema l'ambiente è modellato in modo discreto, ogni preda è rappresentata da una cella colorata di bianco e ogni predatore è rappresentato come una cella colorata di nero. Il movimento è gestito in modo tale che l'agente che si muove verso il limite sinistro dell'ambiente, una volta oltrepassato, si troverà nella parte destra dell'ambiente. In modo del tutto analogo viene gestito il movimento verso l'alto e verso il basso. Ad ogni istante di tempo viene prelevata una cella in modo casuale dalla griglia. Se essa risulta essere vuota, allora non succede nulla, viceversa se è occupata da un agente allora viene selezionata una cella scegliendola in modo casuale tra quelle adiacenti ad essa. Prendendo in considerazione le due celle selezionate, all'inizio di ogni istante temporale vengono applicate le seguenti regole[16]:

<sup>1</sup><https://rf.mokslasplus.lt/agent-based-prey-predator-model/>

- Se una cella è occupata da un predatore e l'altra da una preda, allora la preda viene mangiata, dopodiché il predatore dà alla luce un figlio con una certa probabilità. Nel caso in cui sia nato un nuovo predatore, esso viene posizionato nella cella dove era posizionata la preda prima di essere stata mangiata dal predatore.
- Se le due celle sono occupate da due prede oppure da due predatori, allora non accade nulla.
- Se una cella è occupata da una preda e l'altra è libera, allora la preda dà vita ad un cucciolo con una certa probabilità che occuperà la cella libera.
- Se una cella è vuota e l'altra è occupata da un predatore, esso muore con una certa probabilità.
- Se dopo l'applicazione delle regole sopra descritte non cambia nulla e la configurazione rende possibile il movimento dell'agente, allora esso si muoverà verso una cella non occupata.

Provando ad eseguire diverse simulazioni con questo modello, si è constatato che esso in linea di massima risulta essere instabile. Infatti una delle due popolazioni (molto più frequentemente le prede) si estingue. Nonostante questo la dinamica del modello sembra essere attendibile, considerando che l'evoluzione del sistema nel corso del tempo è simile a quella descritta dal modello Lotka-Volterra.

### 3.4.2 Modello preda-predatore realizzati in NetLogo

NetLogo è un ambiente programmabile di modellazione multi-agente, usato da migliaia di studenti, professori e ricercatori in tutto il mondo [17]. NetLogo presenta un insieme piuttosto ricco di semplici modelli relativi a svariate discipline. Uno di questi è il modello denominato **Wolf Sheep Predation**. Esso si pone l'obiettivo di esplorare la stabilità dell'ecosistema preda-predatore. Per stabilità si intende il fatto che il sistema tende ad evolvere con fluttuazioni in termini di numerosità delle popolazioni delle specie che lo compongono, senza però che una abbia il sopravvento sull'altra, causandone la completa estinzione. Il sistema si dice quindi instabile nel caso in cui una o entrambe le popolazioni tendano ad estinguersi dopo un certo periodo di tempo.

NetLogo propone due varianti dello stesso modello. La prima, più semplice, viene chiamata **sheep-wolves** e prevede la presenza di soli due tipi di agenti: i lupi (predatori) e le pecore (prede). La seconda versione, quella più completa, oltre ai lupi e alle pecore include anche l'erba, attraverso la quale le pecore si cibano per mantenere alto il proprio livello di energia. In entrambi i modelli, sia le prede che i predatori hanno un certo livello di energia. In ogni istante temporale quello dei lupi diminuisce di un valore prestabilito. Essi per sfamarsi devono predare le pecore, così facendo il loro livello di energia diventerà massimo. Ogni qualvolta il livello di energia di un lupo dovesse raggiungere il valore zero, allora esso muore. Ogni lupo e pecora si riproduce con una probabilità prefissata ad

ogni istante di tempo. Nella versione semplificata del modello la quantità di erba presente nell'ambiente è infinita, questo significa che ogni pecora non perde mai energia e muore solo nel caso in cui sia cacciata da un lupo. Questo implica anche che non viene modellata la crescita dell'erba. Questa versione, come affermato dalla documentazione di NetLogo, pur producendo risultati interessanti è in via definitiva instabile. La versione più completa presenta le seguenti differenze: viene aggiunta la modellazione dell'erba, le pecore (così come i lupi) devono cibarsi per poter acquisire energia e nel caso in cui dovessero esaurirla morirebbero. Una volta che l'erba viene mangiata da una pecora, si rigenera dopo un certo periodo di tempo prefissato.

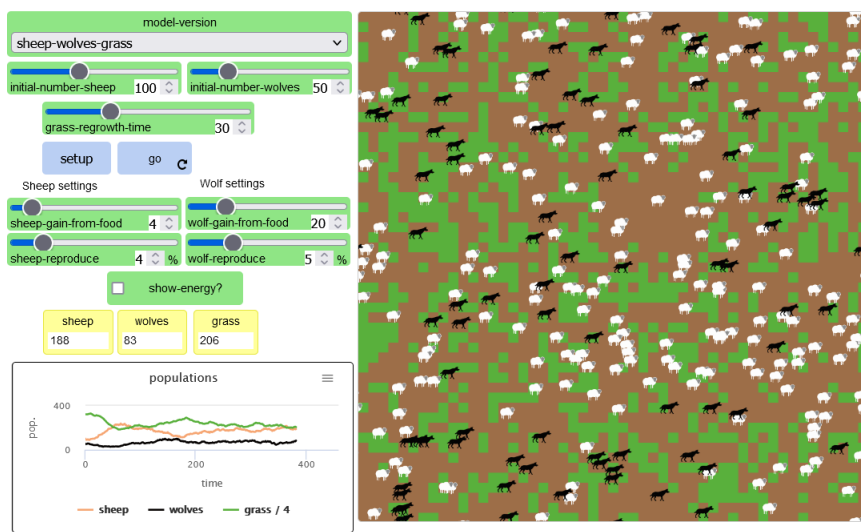


Figura 7: La figura mostra il modello "Wolf Sheep predation" della libreria NetLogo

Oltre alle varianti appena descritte, in letteratura esistono svariati modelli multi-agente di simulazione di un ecosistema naturale realizzati in NetLogo. Gran parte di essi non si differenziano molto dai modelli standard della libreria di NetLogo. Di seguito, viene presentato quello proposto dai due ricercatori **Wilensky e Rand**[18], le cui principali caratteristiche sono le seguenti:

- *Tipi di agenti*: pecore, lupi, erba;
- *Proprietà degli agenti*: energia, posizione, direzione (solo per le pecore e i lupi), quantità (solo per l'erba);
- *Comportamento degli agenti*: movimento, morte, riproduzione (solo per lupi e pecore), mangia-pecora (solo per i lupi), mangia-erba (solo per le pecore), ricrescita (solo per l'erba);
- *Parametri*: Numero di pecore, numero di lupi, costo del movimento, energia acquisita dal cibarsi dell'erba, energia acquisita dal cibarsi delle pecore, tasso di ricrescita dell'erba.



- *Andamento nel tempo*: Le pecore e i lupi si muovono  $\rightarrow$  le pecore e i lupi muoiono  $\rightarrow$  Le pecore e i lupi mangiano  $\rightarrow$  Le pecore e i lupi si riproducono  $\rightarrow$  l'erba ricresce
- *Misure mostrate nei grafici*: Popolazione delle pecore in funzione del tempo, popolazione dei lupi in funzione del tempo e quantità di erba presente nel sistema sempre in funzione del tempo.

Quest'ultimo modello appena descritto risulta essere stabile, ma, come descritto nella sezione successiva risulta essere troppo semplificato rispetto al contesto reale che intende simulare.

### 3.5 I limiti dei modelli presenti in letteratura

I modelli descritti nella sezione precedente sono certamente validi, tuttavia risultano essere troppo semplificati rispetto al fenomeno reale che intendono simulare. In particolare, i limiti di questi modelli sono riassumibili nei seguenti punti:

- Tutti i modelli presentati semplificano in modo estremo la modellazione dei fenomeni di accoppiamento e di riproduzione sia delle prede che dei predatori. Infatti, nessuno di essi simula il fenomeno naturale dell'accoppiamento degli individui. La riproduzione in tutti i casi è modellata in modo tale che ogni individuo in ogni istante di tempo abbia la possibilità di dare vita ad un cucciolo con una probabilità prefissata. Ovviamente, questa è una semplificazione estrema rispetto a ciò che accade nella realtà. Infatti in un contesto reale, come ampiamente descritto nelle sezioni 2.1.2 e 2.1.1, gli animali quando sentono la necessità di riprodursi, cercano un partner ideale col quale accoppiarsi. Solo dopo il processo di accoppiamento può avere luogo la riproduzione.
- Una seconda importante limitazione di questi modelli riguarda il fatto che in natura ogni animale (che sia una volpe, una pecora o un lupo) non necessita solo di cibo per poter sopravvivere, ma ha bisogno anche di dissetarsi, bevendo quotidianamente una certa dose di acqua che può essere localizzata nell'ambiente in una pozzanghera, in un fiume, in un lago e così via.
- Tutti i modelli descritti non considerano il concetto di percezione degli animali. Come è noto e come descritto nelle sezioni precedenti, le volpi e i conigli in particolare, ma questo vale anche per i lupi e le pecore e più in generale per qualunque tipo di animale, hanno una certa percezione che deriva da quanto i suoi sensi (soprattutto udito, vista e olfatto) risultano essere acuti. Nessuno dei sistemi analizzati considera questo aspetto. Essi infatti sono stati sviluppati in modo tale che quando un predatore si trova sulla stessa cella di una preda, esso se ne ciba uccidendola. Anche questa è una semplificazione eccessiva della realtà: in natura i predatori spesso attuano delle strategie di caccia sfruttando i propri sensi. Questo significa che, in un contesto reale, i predatori non si muovono in modo casuale e quando

si trovano esattamente a ridosso di una preda la cacciano, bensì tendono a seguire le tracce lasciate dalle prede e a inseguirle nel caso ne rilevino la loro presenza nelle vicinanze, grazie alla propria percezione. Vale lo stesso discorso per le prede; esse sfruttano i propri sensi per capire se c'è un pericolo nelle vicinanze e in caso affermativo tentano di fuggire, eventualmente cercando di nascondersi alla vista del predatore.

- Un'ulteriore semplificazione della realtà adottata da questi modelli risiede nel fatto che le prede possono morire per due sole cause. Una preda infatti, può morire cacciata da un predatore oppure può morire di fame, a causa della scarsità di cibo presente nell'ambiente. Per quanto concerne i predatori invece, tutti e tre i modelli analizzati prevedono che essi possano morire solo a causa della scarsità di prede di cui cibarsi. Come facilmente immaginabile, in realtà queste non sono le uniche cause che possono portare alla morte in natura degli animali. Le prede e i predatori possono anche morire di sete, a causa della scarsità di acqua per esempio causata dal verificarsi di un lungo periodo di siccità, oppure possono morire di vecchiaia.
- Ulteriore semplificazione adottata da questi modelli riguarda il fatto che in natura l'alternarsi delle quattro stagioni può portare ad effetti molto importanti nel contesto dell'ecosistema naturale simulato. La crescita dei vegetali non segue un andamento costante nel tempo. In primavera e in autunno, quando le piogge sono più frequenti, essi tenderanno a proliferare nell'ambiente, viceversa d'estate quando le piogge sono più rare, essi tenderanno a scarseggiare. Le stesse considerazioni valgono per le masse d'acqua.
- Nessuno dei tre modelli considerati simula il passaggio dei geni dai genitori ai figli e la conseguente possibilità del verificarsi di un fenomeno evolutivo che può portare, col passare del tempo, allo svilupparsi di nuove caratteristiche nelle nuove generazioni di animali che gli permettono di adattarsi meglio all'ambiente in cui vivono.

## 4 Obiettivo del progetto

L'obiettivo del progetto è stato quello di realizzare un modello di simulazione multi-agente di un ecosistema naturale. Come illustrato nella sezione precedente, gran parte dei modelli di questo tipo presenti in letteratura sono piuttosto limitati, in quanto semplificano eccessivamente la realtà che intendono simulare. L'obiettivo quindi è stato quello di eliminare queste limitazioni. Dopo il processo di calibrazione dei parametri del modello si è optato per lo sviluppo di tre configurazioni dello stesso. La prima, la più semplice, ha portato a risultati non del tutto soddisfacenti, per questo motivo si è deciso di affinare il modello facendo sì che un certo numero di animali (sia prede che predatori) venissero reintrodotti nell'ambiente in modo periodico con una certa frequenza, così da simulare il fatto che altri individui entrano nell'ambiente, migrando dalle zone circostanti. I risultati

ottenuti con questa seconda configurazione sono decisamente migliori e il modello è risultato essere notevolmente più realistico. Infine, si è deciso di estenderlo ulteriormente, introducendo la simulazione dell'evoluzione genetica degli animali che popolano l'ambiente. Il fine è stato quello di valutare tramite simulazione come l'evoluzione genetica permetta sia ai conigli che alle volpi di adattarsi alle condizioni ambientali. Per ciascuna delle tre varianti è stata eseguita un'analisi dei risultati e una comparazione con quelli relativi alle altre configurazioni.

## 5 Descrizione del modello

Come già detto precedentemente, il modello sviluppato è un sistema multi-agente, composto da quattro agenti: i *Conigli* (ovvero le prede), le *Volpi* (ovvero i predatori), l'Acqua (ovvero le pozzanghere d'acqua attraverso le quali le prede e i predatori si dissetano) e le Carote (ovvero il vegetale con cui i conigli si sfamano). In questa sezione vengono illustrati l'ambiente, le quattro tipologie di agenti e i parametri del modello. Inoltre, per ognuno di essi, viene illustrato il valore finale scelto in seguito alla fase di tuning, con la quale si è cercato di rendere il modello stabile.

### 5.1 Ambiente

L'ambiente è stato modellato nel seguente modo:

- L'ambiente è **continuo**. Si è adottata questa scelta implementativa per rendere la simulazione più realistica. Infatti, l'unità di misura utilizzata per descrivere i movimenti e le percezioni degli agenti nello spazio è il singolo pixel, raggiungendo così un indice di granularità sufficientemente alto da considerare l'ambiente continuo.
- L'ambiente è **inaccessibile**. Questo significa che ogni agente non può, in ogni istante di tempo, ottenere informazioni complete, precise e aggiornate relative allo stato dell'ambiente stesso. Infatti ogni volpe è in grado di conoscere esclusivamente le informazioni che la sua percezione le permette di ottenere. Questo significa che una volpe può sapere se c'è un coniglio, uno specchio d'acqua oppure un'altra volpe con cui accoppiarsi nelle vicinanze, ma non può conoscere (e non è nemmeno necessario) il fatto che nella posizione diametralmente opposta alla sua si trova un altro individuo della sua specie, per esempio.
- L'ambiente è, inoltre, **dinamico**: nella finestra temporale nella quale ogni agente compie la decisione relativa al comportamento che deve tenere, l'ambiente può evolvere. In altre parole questo significa che esistono fattori esterni che influenzano e modificano l'ambiente mentre ogni agente valuta l'opportuna azione da effettuare. Per esempio, d'estate alcune delle pozze d'acqua vengono rimosse a causa del caldo e quindi l'ambiente evolve grazie ad un fattore esterno non gestibile dagli agenti.

- L'ambiente è **non deterministico**, ovvero ogni qualvolta un agente compie un'azione, non è possibile considerare come certo l'avvenimento dell'azione stessa. Infatti, alcune azioni sono state gestite tramite un processo regolato in modo stocastico e quindi non è certo che i loro effetti si realizzino effettivamente. Oltre a ciò, anche eventi esterni possono impedire la riuscita dell'azione che un agente ha prestabilito. Per esempio, se una volpe decide di predare un coniglio nelle vicinanze, non è garantito che possa riuscirci poiché possono intervenire svariati fattori esterni; per esempio il coniglio potrebbe nel frattempo morire di fame e quindi essere rimosso dall'ambiente prima che la volpe possa cibarsene.
- Ulteriore caratteristica dell'ambiente è che esso risulta essere **episodico**. Episodico significa che ad ogni step temporale il comportamento di ogni agente è determinato considerando esclusivamente lo stato attuale del sistema. Gli agenti non sono dotati di memoria e non mantengono traccia delle azioni precedenti.

### 5.1.1 Implementazione

Ambiente
+ agenti + background + colore.erba + lista_agenti_aggiunti + lista_agenti_rimossi + numero_volpi_nate + numero_conigli_nati + numero_conigli_morti_fame + numero_volpi_morti_fame + numero_conigli_morti sete + numero_volpi_morti sete + numero_conigli_morti_vecchiaia + numero_volpi_morti_vecchiaia + numero_conigli_morti_cacciati + stagione + giorno
+ aggiungi_agente() + rimuovi_agente() + get() + process() + render() + agente_vicino()

Figura 8: Diagramma della classe Ambiente

L'ambiente è stato implementato attraverso un'apposita classe, le cui specifiche sono mostrate in figura 8. Gli attributi di questa classe includono: la lista di agenti presenti nell'ambiente, la lista di agenti rimossi dall'ambiente dopo la loro morte, la lista di agenti aggiunti in seguito alla loro nascita, una variabile di tipo

stringa denominata *stagione* che indica in quale stagione si trova attualmente la simulazione, una variabile di tipo floating point denominata *giorno* che permette di tenere traccia del passare del tempo durante la simulazione, una variabile denominata *background* utile in fase di render dell'ambiente, una tupla denominata *colore\_erba* che permette di specificare il colore con cui l'ambiente deve essere colorato in ciascuna stagione e infine, una serie di variabili intere che permettono di raccogliere i dati della simulazione in corso.

L'ambiente può eseguire le seguenti operazioni: aggiungere un agente quando esso nasce, rimuovere un agente quando muore, restituire l'oggetto che rappresenta uno specifico agente dato il suo identificativo, eseguire il render dell'ambiente, portare avanti la simulazione in ogni istante temporale (metodo *process()*) e ricercare l'agente che si trova più vicino ad un altro agente passato come parametro al metodo *agente\_vicino*. Più nello specifico, il metodo *process* si occupa di mettere in atto il comportamento di ogni agente in ciascun istante temporale, di aggiungere alla simulazione gli agenti contenuti nella lista denominata *lista\_agenti\_aggiunti*, di rimuovere dalla simulazione gli agenti contenuti nella lista chiamata *lista\_agenti\_rimossi* e infine, di rimuovere gli oggetti contenuti nelle due liste. Il metodo *agente\_vicino* invece, viene richiamato dagli agenti di tipo Volpe o Coniglio nel momento in cui essi devono cercare una preda da cacciare oppure un compagno con cui accoppiarsi. Il metodo prende in input le coordinate attuali dell'agente, il raggio in cui cercare (ovvero la loro percezione) e il tipo di agente da cercare. Il metodo quindi restituisce l'oggetto che fa riferimento all'agente più vicino del tipo specificato se ne è presente uno nel raggio specificato, altrimenti non restituisce nulla.

**Parametri** I parametri relativi all'ambiente sono i seguenti:

- **Dimensione dell'ambiente.** Questo parametro indica la dimensione in altezza e in lunghezza espressa in pixel della finestra nel quale viene eseguito il render grafico dell'ambiente. In seguito a diverse prove, si è constatato che la dimensione dell'ambiente in rapporto al numero di agenti che lo popolano non deve essere troppo elevata. Infatti se l'ambiente fosse troppo grande, allora gli agenti si disperderebbero al suo interno. D'altro canto è anche essenziale che la dimensione dell'ambiente non sia eccessivamente ridotta in proporzione al numero di agenti al suo interno e alla loro velocità di movimento.
- **Probabilità pioggia** in primavera, estate, autunno e inverno. Questo parametro indica per ciascuna delle quattro stagioni con quale probabilità in ogni istante di tempo pioverà.
- **Quantità nuove pozzanghere** in primavera, estate, autunno e inverno. Indica, per ogni stagione in caso dovesse piovere in un istante temporale, quante pozzanghere verranno aggiunte all'ambiente. Rappresenta l'intensità del fenomeno atmosferico.

- **Probabilità aggiunta carote** in primavera, estate, autunno e inverno. Indica, per ogni stagione, la probabilità che in ogni istante di tempo venga aggiunta una carota in una posizione casuale nell'ambiente.
- **Numero iniziale di agenti.** Indica il numero di agenti di tipo Coniglio, di tipo Volpe, di tipo Acqua e di tipo Carota inseriti inizialmente nel sistema.
- **Durata stagioni.** Indica la durata, espressa in istanti temporali, di ognuna delle quattro stagioni.
- **Entità riduzione pozzanghere.** Questo parametro indica di quante unità in ogni istante temporale d'estate le pozzanghere vengono ridimensionate a causa del caldo che fa evaporare l'acqua.

**Tuning dei parametri** Di seguito viene esposto il valore con cui sono stati settati i parametri descritti nel paragrafo precedente.

	Larghezza (pixel)	Altezza (pixel)
Dimensione dell'ambiente	1200	800

	Volpe	Coniglio	Acqua	Carota
Numero iniziale di agenti	25	75	200	250

	Stagione			
Parametro	Primavera	Estate	Autunno	Inverno
Probabilità pioggia	0.285%	0.1%	0.5%	0.4%
Quantità nuove pozzanghere	rand(0,10)	rand(0,50)	rand(0,10)	rand(0,10)
Probabilità aggiunta carote	33%	20%	10%	20%
Durata stagioni	500	500	500	500
Entità riduzione pozzanghere	-	1	-	-

## 5.2 Agenti

Il modello sviluppato è composto da un totale di quattro tipi di agenti: volpi, conigli, carote e pozzanghere d'acqua. Di seguito vengono descritte le caratteristiche generali i parametri e il comportamento di ognuno di essi.

### 5.2.1 Acqua

Questo agente è il più semplice insieme all'agente *Carota* e rappresenta gli specchi d'acqua (pozzanghere) attraverso i quali le volpi e i conigli hanno la possibilità di soddisfare il proprio bisogno di dissetarsi. Sono distribuite in modo casuale nell'ambiente e possono essere sfruttate dalle prede e dai predatori fino ad un massimo di tre volte prima di scomparire. Ne vengono aggiunte di nuove nell'ambiente con frequenza differente nelle quattro stagioni, come mostrato nella sezione 5.1.1.

### 5.2.2 Implementazione

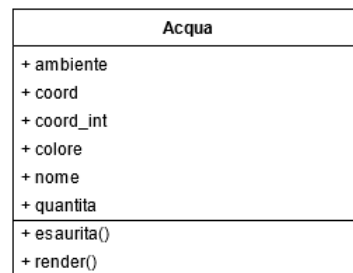


Figura 9: Diagramma della classe Acqua

Gli attributi della classe Acqua sono i seguenti:

- **ambiente**: è un riferimento all'oggetto Ambiente.
- **coord**. Si tratta di un vettore in due dimensioni che rappresenta le coordinate dove si trova la rispettiva pozzanghera d'acqua nell'ambiente.
- **coord\_int**. Analogo a coord, solo che *coord\_int* è espresso tramite numeri interi e non decimali.
- **colore**. Indica il colore con cui dovrà essere eseguito il render degli agenti di questo tipo.
- **nome**. Si tratta di una stringa con la quale vengono identificati gli agenti appartenenti a questa categoria.
- **quantità**. Indica la dimensione della pozzanghera. Può assumere valori interi compresi tra 0 e 3. 3 significa che la pozzanghera può essere sfruttata

ancora da 3 conigli e/o volpi prima di essere eliminata dall'ambiente, 0 significa che la pozzanghera deve essere rimossa, in quanto è stata già sfruttata 3 volte dalle volpi e dai conigli e quindi risulta essere esaurita.

I metodi *esaurita* e *render* permettono rispettivamente di chiedere all'ambiente l'eliminazione dell'agente e di eseguire il render grafico degli agenti di questo tipo.

### 5.2.3 Carota

Questa tipologia di agente rappresenta ciò di cui i conigli si cibano per potersi sostenere. Anche le carote, così come le pozzanghere d'acqua, sono inizialmente distribuite in modo casuale nell'ambiente. Le carote però, a differenza degli specchi d'acqua, una volta che sono state mangiate dai conigli scompaiono dall'ambiente e non possono essere ulteriormente mangiate da altri agenti. Così come per l'acqua, anche le carote vengono aggiunte sul territorio con frequenza che varia in base alla stagione, come illustrato nella sezione 5.1.1.

### 5.2.4 Implementazione

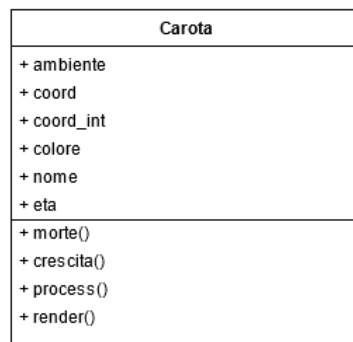


Figura 10: Diagramma della classe Carota

Gli attributi degli agenti di tipo Carota sono i seguenti:

- **ambiente**. Si tratta di un riferimento all'oggetto della classe Ambiente.
- **coord**. Si tratta di un vettore in due dimensioni, che rappresenta le coordinate che identificano la posizione della carota all'interno dell'ambiente.
- **coord\_int**. Analogo a coord, la differenza risiede nel fatto che *coord\_int* è espresso tramite numeri interi e non decimali.
- **colore**. Indica il colore con cui dovrà essere eseguito il render degli agenti di questo tipo.
- **nome**. Si tratta di una stringa, utile per identificare gli agenti che appartengono a questa categoria.



- **eta.** Indica da quanto tempo la carota è presente nell'ambiente. Una volta che la carota ha superato una soglia di vecchiaia verrà rimossa, in quanto marcita.

I metodi della classe Carota sono i seguenti:

- **morte.** Questo metodo viene invocato ogni qualvolta una carota viene mangiata da un coniglio, oppure marcisce. Esso chiede all'ambiente di rimuovere il rispettivo agente.
- **crescita.** Questo metodo permette con una probabilità pari a 0.05%, di far nascere una nuova carota nelle vicinanze dell'agente che lo richiama.
- **process.** Questo metodo, richiamato in ogni istante temporale della simulazione, esegue le seguenti operazioni:
  - richiama il metodo *crescita*, descritto nel punto precedente;
  - incrementa l'attributo *eta* dell'agente di 0.03;
  - nel caso in cui l'età della carota dovesse aver raggiunto un valore superiore a 2.5 cambia il colore con cui essa viene mostrata in modo da evidenziare che essa sta per marcire e quindi per sta per essere rimossa dalla simulazione;
  - nel caso in cui la carota dovesse avere un'età superiore a 3, chiede all'ambiente di rimuovere l'agente.
- **render.** Questo metodo permette di eseguire il render grafico dell'agente nella corretta posizione dell'ambiente.

### 5.2.5 Coniglio e Volpe

Gli agenti di tipo *Coniglio* rappresentano le prede dell'ecosistema naturale che si intende simulare, mentre gli agenti di tipo *Volpe* rappresentano i predatori. I due agenti sono stati implementati tramite le classi Coniglio e Volpe, che, come si può vedere dalla figura 11, condividono la stessa struttura.

### 5.2.6 Implementazione

Gli attributi relativi agli agenti Coniglio e Volpe sono i seguenti:

- **età.** Indica l'età dell'agente. In ogni istante temporale questo valore viene incrementato di 0.03. Conigli e Volpi di età superiore a 4 o inferiore a 0.5 hanno percezione e velocità di corsa dimezzata rispetto a quella degli animali adulti. Inoltre, Conigli e Volpi con età inferiore a 0.5 non possono accoppiarsi.
- **percezione.** Questo attributo indica il raggio di percezione dell'agente. Un valore alto indica che l'agente può vedere e sentire predatori da grande distanza, viceversa un valore basso sta a significare che l'agente può vedere e percepire la presenza di altri agenti solo nel momento in cui si trovano a distanze ridotte rispetto al punto in cui esso si trova.



Figura 11: Diagrammi delle classi Coniglio e Volpe

- **ambiente**. Si tratta di un riferimento all'oggetto della classe Ambiente, precedentemente descritta.
- **nome**. Si tratta di una stringa che indica il tipo di agente.
- **coord**. Esso è un vettore in due dimensioni che identifica la posizione dell'agente in ogni istante temporale.
- **coord\_int**. Indica la posizione attuale dell'agente tramite coordinate intere.
- **destinazione**. Indica la destinazione dell'agente. Inizialmente ogni coniglio e ogni volpe hanno una destinazione che viene fissata in modo casuale. Successivamente, nel momento in cui l'agente percepisce la necessità di accoppiarsi, cibarsi o dissetarsi e trova un agente che soddisfa il suo bisogno nelle vicinanze, questo parametro viene impostato in modo tale che la destinazione sia pari alle coordinate della posizione dell'altro agente. Invece nel caso in cui l'agente non ha necessità allora la sua destinazione verrà

impostata in modo casuale, in modo tale che esso si muova liberamente nell'ambiente.

- **fame.** Si tratta di un valore booleano che indica se l'agente ha necessità di sfamarsi o meno. Inizialmente il valore è pari a False e nel momento in cui il numero di step temporali passati senza che abbia mangiato diventa superiore al parametro *soglia\_fame* il suo valore viene impostato a True.
- **sete.** Anche in questo caso si tratta di un valore booleano. Esso indica se l'agente necessita o meno di bere. Anche in questo caso, esso inizialmente è impostato a False e appena il tempo passato dall'agente senza bere diventa superiore al valore del parametro *soglia\_disidratazione* il suo valore viene impostato a True.
- **fertilità.** Si tratta di un valore booleano che indica se l'agente si trova in un periodo fertile o meno.
- **gravidanza.** Anch'esso è un parametro di tipo booleano e indica se l'agente risulta essere in attesa di una cucciolata o meno. Nel primo caso, dopo un certo periodo di gestazione, esso darà alla luce i cuccioli.
- **preda.** Fa riferimento ad un agente di tipo *Carota* nel caso dei conigli e un agente di tipo *Coniglio* nel caso delle volpi e rappresenta la preda identificata dall'agente per potersi cibare.
- **preda\_nome.** Si tratta di una stringa che identifica il tipo di preda: coniglio per le volpi e carota per i conigli.
- **compagno.** Esso rappresenta un agente con cui l'animale ha intenzione di accoppiarsi.
- **tempo\_digiuno.** Si tratta di un parametro che indica il tempo passato dall'ultimo istante in cui l'agente si è cibato.
- **tempo\_disidratazione.** Indica quanti istanti temporali sono trascorsi dall'ultima volta che l'agente si è abbeverato in una pozzanghera.
- **tempo\_periodo\_fertile.** Indica quanto tempo è passato dall'ultimo accoppiamento dell'animale.
- **tempo\_gestazione.** Indica il tempo che deve passare dal momento in cui l'agente si è accoppiato prima che esso possa dare alla luce i cuccioli.
- **pozzanghera.** Indica un agente di tipo Acqua che l'agente ha avvistato per potersi dissetare.
- **color.** Indica il colore con cui eseguire il render dell'agente. Questo attributo viene considerato solo nel momento in cui la simulazione viene lanciata con l'attributo GRAFICA impostato a False. Vedere la sezione 6 per maggiori dettagli in tal senso.

- **img.** Indica l'immagine con cui eseguire il render dell'agente. Questo attributo viene considerato solo per le simulazioni lanciate con l'attributo **GRAFICA** settato a **True**. Fare riferimento anche in questo caso alla sezione 6 per maggiori dettagli.
- **soglia\_fame.** Si tratta di una soglia superata la quale l'agente inizia ad avere necessità di cibarsi (di carote nel caso dei conigli e di conigli nel caso delle volpi).
- **soglia\_morte\_di\_fame.** Si tratta della soglia superata la quale l'agente morirà di fame. In altre parole, significa che se un coniglio o una volpe rimangono senza mangiare per un numero di istanti temporali superiore a tale soglia, allora esso morirà di fame.
- **soglia\_sete.** Indica la soglia oltre la quale l'agente inizia a percepire la necessità di dissetarsi.
- **soglia\_morte\_di\_sete.** Così come per la fame, anche questa soglia è rappresentativa del fatto che se trascorre un numero di time step senza che l'agente abbia bevuto che è superiore rispetto a questa soglia, esso morirà di sete e verrà rimosso dall'ambiente.
- **soglia\_fertilità.** Indica la soglia oltre la quale l'agente entra in un periodo di fertilità. Anche in questo caso, una volta passati un numero di time step senza che il coniglio si sia accoppiato, superiore a questa soglia, l'agente entra nel periodo di fertilità e quindi potrà cercare un compagno.
- **velocità\_camminata.** Questo parametro indica la velocità con cui si sposta l'agente in condizioni normali, ovvero quando non è inseguito da un predatore, non è alla ricerca di un compagno e non è alla ricerca di cibo con cui sfamarsi.
- **velocità\_corsa.** Questo parametro invece, indica la velocità con cui l'agente si muove nelle seguenti condizioni: l'agente ha trovato un compagno con cui accoppiarsi, l'agente ha trovato una pozzanghera da cui abbeverarsi oppure l'animale ha trovato una preda di cui cibarsi nelle vicinanze.
- **eta\_morte\_vecchiaia.** Questo parametro indica a quale età le volpi e i conigli muoiono di vecchiaia.

**Comportamento degli agenti Volpe e Coniglio** Il metodo *bisogni\_primari*, come si può intuire dal nome stesso, si occupa della gestione dei bisogni primari degli agenti di tipo Volpe e di tipo Coniglio. Entrambe queste categorie di agenti hanno tre necessità: cacciare (le volpi cacciano i conigli e i conigli si cibano delle carote) per non morire di fame, dissetarsi per non morire di sete e infine, accoppiarsi per potersi riprodurre.

La gestione dei bisogni primari avviene come mostrato in figura 12. Come si vede in tale immagine, per prima cosa si valuta se l'agente è morto di fame o di

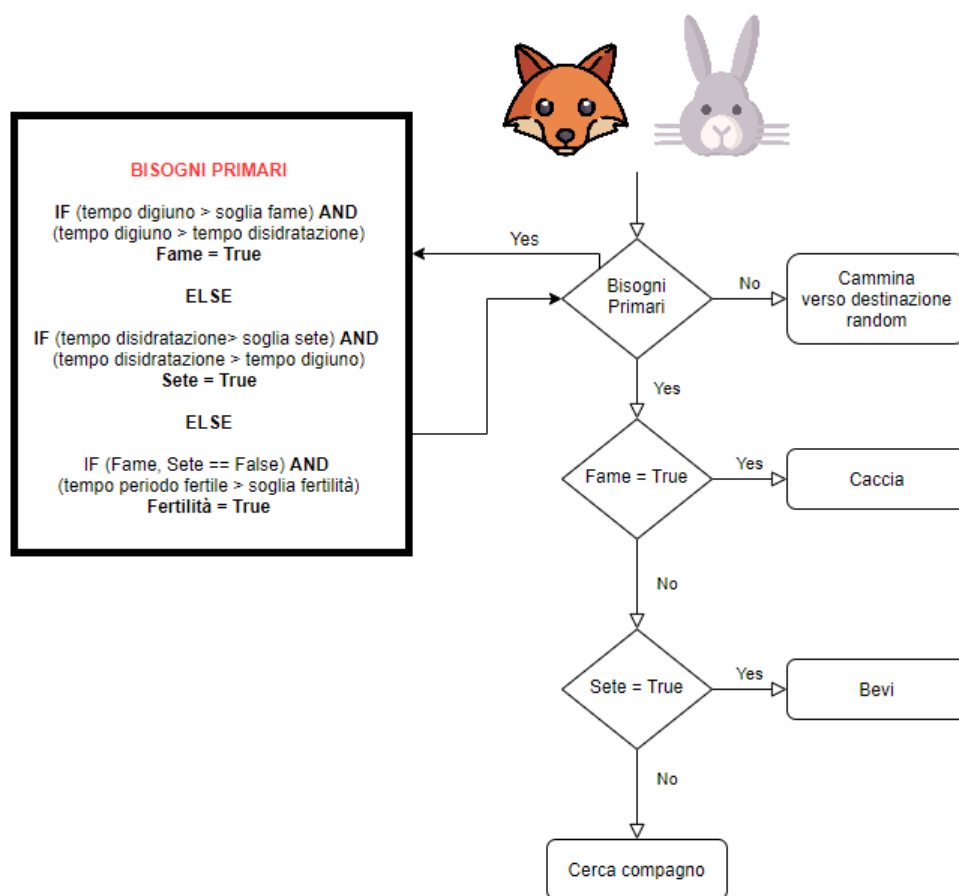


Figura 12: Il diagramma in figura mostra il comportamento degli agenti di tipo Volpe e di tipo Coniglio ad ogni iterazione.

sete e nel caso viene richiamato il metodo *morte* descritto in seguito. Successivamente si valuta se l'agente ha una delle tre necessità. Per far ciò, per prima cosa, viene considerato il tempo passato dall'ultima volta che l'animale si è cibato (valore memorizzato nella variabile *tempo\_digiuno*) e il valore del tempo trascorso dall'ultimo istante in cui l'animale si è dissetato (memorizzato nella variabile *tempo\_disidratazione*). Se il primo dei due valori è maggiore del valore di *soglia\_fame* e del valore della variabile *tempo\_disidratazione*, allora il valore della variabile booleana *fame* viene settato a True. In seguito, si valuta se il tempo trascorso dall'ultima volta che l'animale si è dissetato è maggiore del valore di *soglia\_sete* e del valore che indica il tempo trascorso dall'ultimo momento in cui l'animale si è sfamato. In caso affermativo, allora il valore della variabile booleana *sete* viene settato a True. Infine, se il tempo trascorso dall'ultima volta che l'animale si è cibato è minore del valore *soglia\_fame* e il tempo trascorso dall'ultima volta che l'animale si è dissetato è minore di *soglia\_sete* e il tempo trascorso dall'ultima volta che l'animale si è accoppiato (memorizzato nella variabile *tempo\_periodo\_fertile*) è maggiore del valore assunto dal parametro *soglia\_fertilità*, allora il valore della variabile booleana *fertilita* viene settato a True.

In seguito a tali valutazioni, se nessuna delle tre variabili *fame*, *sete* e *fertilita* ha assunto un valore pari a True, allora l'agente cammina verso una destinazione fissata in modo casuale all'interno dell'ambiente. Viceversa, se il valore della variabile *fame* è diventato pari a True, allora l'agente inizia la procedura denominata *Caccia*, descritta successivamente. Nel caso in cui la variabile *fame* sia False e la variabile *sete* sia True, allora l'agente inizia la procedura denominata *Bevi*, descritta in seguito. Infine, nel caso in cui sia la variabile *fame* che la variabile *sete* sono pari a False e la variabile *fertilita* ha assunto valore pari a True, allora l'agente inizia la procedura *Cerca compagno*, descritta anch'essa successivamente in questa sezione.

Il metodo *caccia* permette di attivare la procedura denominata in modo analogo, attraverso la quale i Conigli possono cercare carote nelle vicinanze per potersi sfamare e le volpi possono cercare dei conigli nelle vicinanze in modo tale da poterli cacciare per poter soddisfare la propria necessità di cibarsi. Come visibile nella figura 13, l'agente per prima cosa cerca se c'è una preda nell'area circolare avente come centro la posizione attuale dell'agente e come raggio la sua percezione. Se l'agente non trova una preda (una carota nel caso dei conigli, un coniglio nel caso delle volpi), allora esso camminerà verso un punto posizionato in modo casuale nell'ambiente e al prossimo istante temporale la procedura verrà ripetuta dall'inizio. Viceversa, nel caso in cui l'agente dovesse trovare una preda nei dintorni, allora l'ambiente comunica all'agente se la preda è ancora viva oppure no. Quest'ultima valutazione si rende utile per evitare il presentarsi di alcuni conflitti che occorrerebbero nel momento in cui un agente ha trovato una preda, ma prima che esso possa raggiungerla, un altro agente se ne ciba rendendola non più disponibile per il primo agente. Nel caso in cui la preda non dovesse essere più presente nell'ambiente, allora l'agente si muove verso una destinazione random e ripete dall'inizio la procedura. Invece, nel caso in cui la preda dovesse essere ancora viva, l'agente inizia a correre verso di essa. Nel momento in cui la preda risulterà

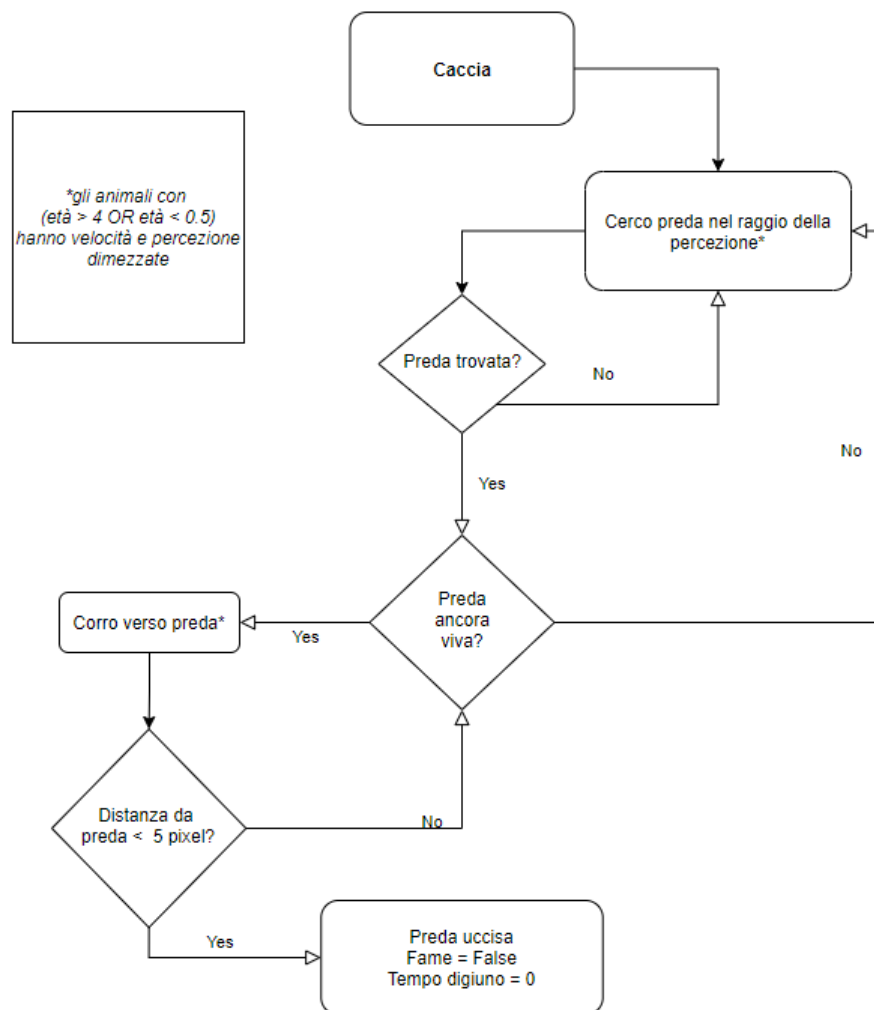


Figura 13: Il diagramma in figura mostra come si comportano i conigli e le volpi quando hanno la necessità di cibarsi

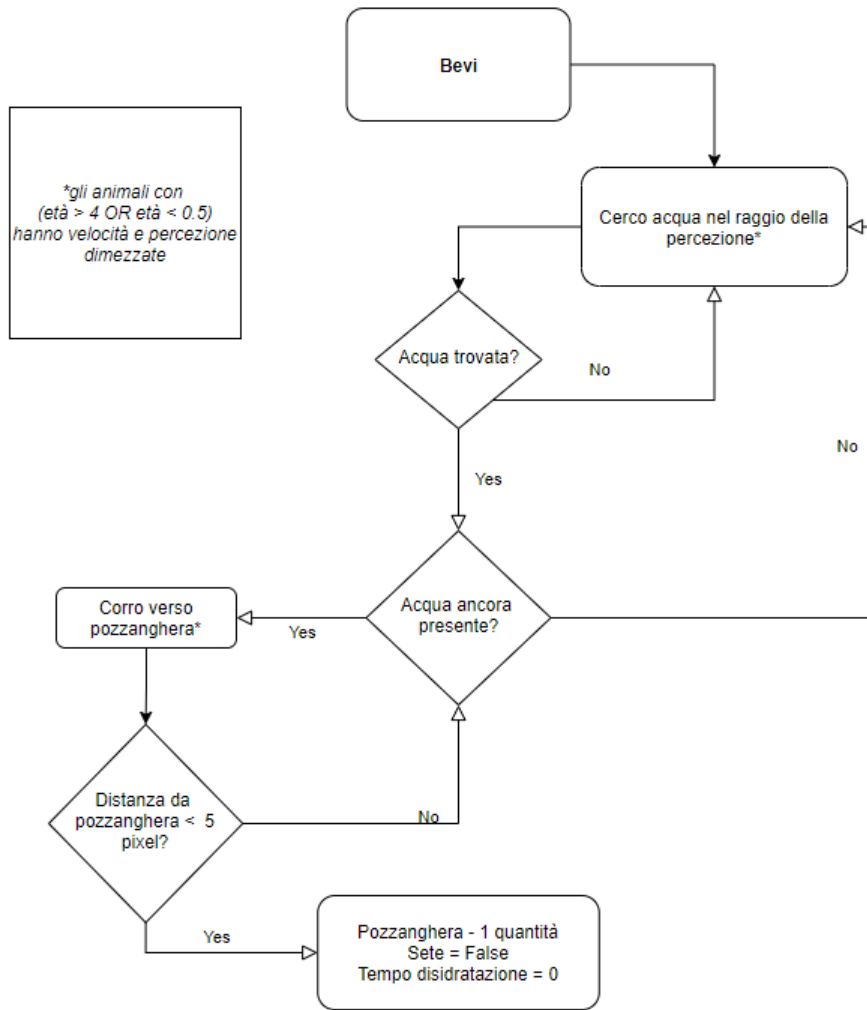


Figura 14: Il diagramma in figura mostra come si comportano i conigli e le volpi quando hanno la necessità di dissetarsi

essere ad una distanza ridotta (inferiore a 5 pixel), allora la preda verrà uccisa, il parametro *fame* dell'agente viene posto a False e l'attributo *tempo\_digiuno* viene settato con un valore pari a zero. Se invece la distanza dovesse essere superiore a 5 pixel, l'agente chiede nuovamente all'ambiente se la preda è ancora viva e la procedura si ripete.

Il metodo *dissetarsi* permette di dare inizio alla procedura denominata *bevi*, mostrata nel diagramma visualizzato nella figura 14. Questa procedura permette sia ai conigli che alle volpi di cercare una pozzanghera d'acqua per soddisfare la propria sete. Essa prevede in primo luogo che l'agente cerchi una pozzanghera nell'area circolare avente come centro la posizione attuale dell'agente e come raggio la percezione dell'agente stesso. Se l'ambiente comunica all'agente che non esiste una pozzanghera nei dintorni, allora l'agente si muove con destinazione casua-



le nell'ambiente e dal prossimo istante temporale la procedura viene riapplicata dall'inizio. Viceversa, nel caso in cui l'ambiente comunica all'agente che esiste una pozzanghera nelle vicinanze, allora esso inizia a correre verso la posizione della pozzanghera. In ogni istante temporale l'ambiente comunica all'agente se la pozzanghera è ancora presente nell'ambiente o meno (potrebbe accadere che altri agenti nel frattempo si sono diretti verso la stessa direzione e abbiano esaurito la pozzanghera). In caso negativo l'agente si dirige camminando verso una destinazione casuale e nel successivo istante temporale ripete la procedura dall'inizio. Se invece l'ambiente comunica all'agente che la pozzanghera non è stata esaurita da altri agenti, allora esso continua la corsa verso di essa. Questo ciclo si ripete fino a quando la distanza tra la pozzanghera e l'agente risulta inferiore a 5 pixel. In questa situazione l'agente si disseta, quindi il valore dell'attributo *sete* diventa False, viene diminuita di una unità la dimensione della pozzanghera e l'attributo *tempo\_disidratazione* dell'agente viene impostato con un valore pari a zero.

Il metodo *cerca\_compagno* permette di dare inizio alla procedura denominata in modo analogo, che permette alle volpi e ai conigli di cercare un animale della propria specie con cui accoppiarsi ed è illustrata con un diagramma visibile nella figura 15. Per prima cosa ogni agente che ha la necessità di accoppiarsi chiede all'ambiente se ci sono altri agenti dello stesso tipo e con età maggiore di 0.5 (i cuccioli non si possono accoppiare) nell'area circolare centrata nella posizione attuale dell'agente e di raggio pari alla percezione dello stesso. In caso negativo l'agente si muove verso una direzione casuale nell'ambiente e dall'istante di tempo successivo riprende dall'inizio l'esecuzione della procedura. Viceversa in caso affermativo, l'ambiente comunica la posizione del candidato compagno più vicino. L'agente inizia a correre verso la posizione comunicatagli e in ogni istante temporale chiede all'ambiente se il compagno è ancora vivo. In caso negativo esso si sposta verso una destinazione casuale per poi ripetere l'intera procedura. Se invece l'ambiente comunica che il compagno è ancora vivo, allora l'agente continua la corsa verso di esso. Questa operazione viene continuamente ripetuta in ogni istante temporale fino a quando la distanza tra i due agenti risulta essere inferiore a 5 pixel. In questo caso l'accoppiamento vero e proprio può avere luogo, richiamando il metodo *accoppiamento*.

L'accoppiamento viene gestito in questo modo: gli attributi *fertilita* dei due agenti coinvolti vengono impostati a False, gli attributi *tempo\_periodo\_fertile*, che indicano quanto tempo è passato dall'ultimo accoppiamento dell'agente, vengono posti a 0 e inoltre l'attributo *gravidanza* di uno dei due agenti viene impostato a True. Dopo di questo, viene richiamato il metodo *nascita* e dopo un periodo di gestazione, avviene la nascita dei cuccioli (vedere il paragrafo 5.1.1 per informazioni relative al numero di cuccioli che nascono ad ogni parto e ai tempi di gestazione delle due tipologie di agenti).

Nelle versioni del modello che non prevedono la gestione dell'evoluzione genetica i parametri *soglia\_fame*, *soglia\_morte\_di\_fame*, *soglia\_sete*, *soglia\_morte\_di\_sete*, *soglia\_fertilità*, *percezione*, *velocita\_camminata* e *velocita\_corsa* dei cuccioli vengono

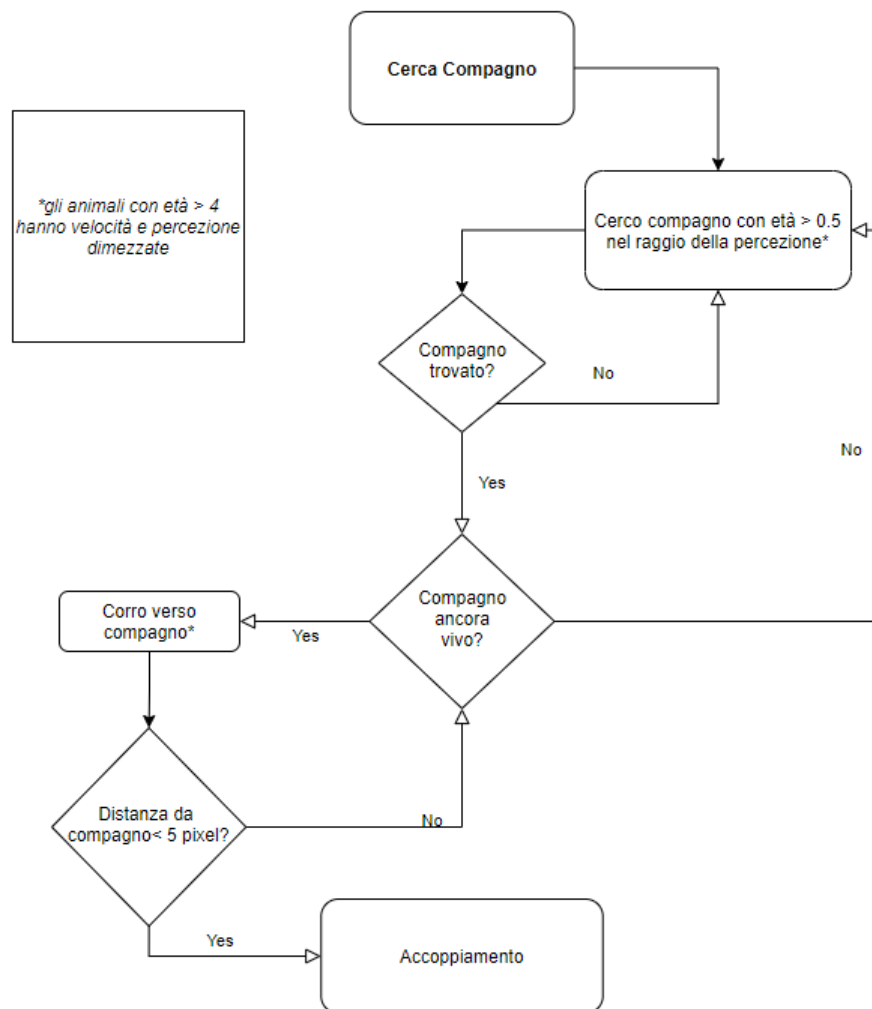


Figura 15: Il diagramma in figura mostra come si comportano i conigli e le volpi quando hanno la necessità di accoppiarsi

impostati seguendo il seguente principio:

$$y = rand(min(x_1, x_2), max(x_1, x_2))$$

Dove  $y$  è il valore ereditato dal cucciolo e  $x_1$  e  $x_2$  sono i rispettivi parametri dei due genitori.

Viceversa, nella versione del modello in cui è stata implementata l'evoluzione genetica, ogni qualvolta nasce un cucciolo i parametri *soglia\_fame*, *soglia\_morte\_di\_fame*, *soglia\_sete*, *soglia\_morte\_di\_sete*, *soglia\_fertilità*, *percezione*, *velocita\_camminata* e *velocita\_corsa* dei cuccioli vengono ereditati dai genitori seguendo questo approccio:

$$y = rand(min(x_1, x_2), max(x_1, x_2)) + rand(-y_1, y_1)$$

Dove  $y$  è il parametro ereditato dal cucciolo e  $x_1, x_2$  sono i rispettivi parametri dei due genitori. Invece,  $y_1$  è un offset appositamente scelto per ognuno degli otto parametri. Un valore grande implica che i cambiamenti genetici, anche nel breve periodo, siano più ampi, mentre se viene scelto un valore basso i cambiamenti genetici, almeno nel breve periodo, risultano essere piuttosto contenuti. Un esempio di questo comportamento è mostrato in figura 16.

Invece, il metodo *morte* viene richiamato quando un animale muore di vecchiaia, muore di sete, muore di fame oppure muore cacciato da un predatore (ovviamente quest'ultimo caso è valevole solo per i conigli e non per le volpi). La prima situazione si verifica quando il valore *eta* dell'agente supera il valore 4.5 nel caso dei conigli e il valore 5 nel caso delle volpi. Una volpe o un coniglio, invece muore di sete quando il valore *tempo\_disidratazione* diventa maggiore del parametro *soglia\_morte\_di\_sete*. Analogamente, un animale muore di fame quando il valore *tempo\_digiuno* diventa superiore al parametro *soglia\_morte\_di\_fame*. In tutti questi casi viene richiamato il metodo *morte* che si occupa di aggiungere l'agente alla lista di agenti da rimuovere. Sarà poi compito dell'ambiente occuparsi della rimozione vera e propria dell'agente dall'ambiente.

Il metodo *cammina*, con l'ausilio del metodo *movimento*, permette di spostare l'agente verso la propria destinazione con velocità pari a quella indicata nel parametro *velocita\_camminata*.

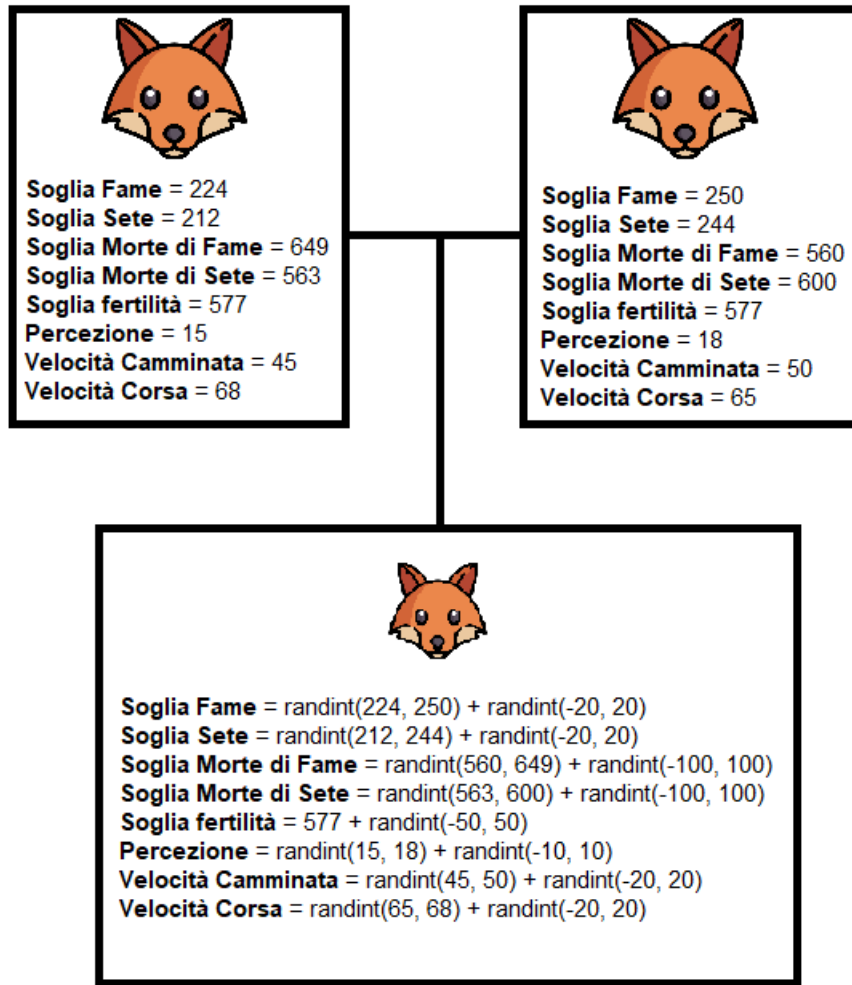


Figura 16: Il diagramma in figura mostra un esempio concreto di come viene gestita l'ereditarietà dei caratteri nel modello.

Infine, il metodo *process* si occupa del coordinamento di tutte le operazioni descritte fino ad ora. Esso viene richiamato in ciascun istante temporale e effettua le seguenti operazioni:

- incrementare di una unità il valore delle variabili *tempo\_digiuno* e *tempo\_disidratazione*;
- incrementare di 0.003 il valore dell'attributo *eta* dell'agente;
- nel caso in cui l'età dell'agente sia maggiore di 0.5, aumentare il valore dell'attributo *tempo\_periodo\_fertile* di una unità;

- valutare se l'agente ha raggiunto un'età tale per cui esso deve morire di vecchiaia e nel caso richiamare il metodo *morte*;
- richiamare il metodo *bisogni\_primari* per la gestione delle necessità dell'agente;
- far muovere l'agente

In ultimo luogo, il metodo *render* viene richiamato per eseguire il render grafico dell'agente all'interno dell'ambiente.

**Tuning dei parametri** Nella tabella sottostante viene esposto il valore con cui sono stati settati i parametri descritti nel paragrafo precedente.

Parametro	Volpe	Coniglio
soglia_fame	rand(200; 250)	rand(200;250)
soglia_morte_di_fame	rand(550;650)	rand(500;600)
soglia sete	rand(200;250)	rand(200;250)
soglia_morte_di_sete	rand(550;650)	rand(500;600)
soglia_fertilita	rand(500;600)	rand(250;300)
percezione	rand(15;20)	rand(15;20)
velocita_camminata	rand(40;60)	rand(40;60)
velocita_corsa	rand(65;75)	rand(65;75)
eta_morte_vecchiaia	5	4.5

## 6 Ambiente di sviluppo e Codice

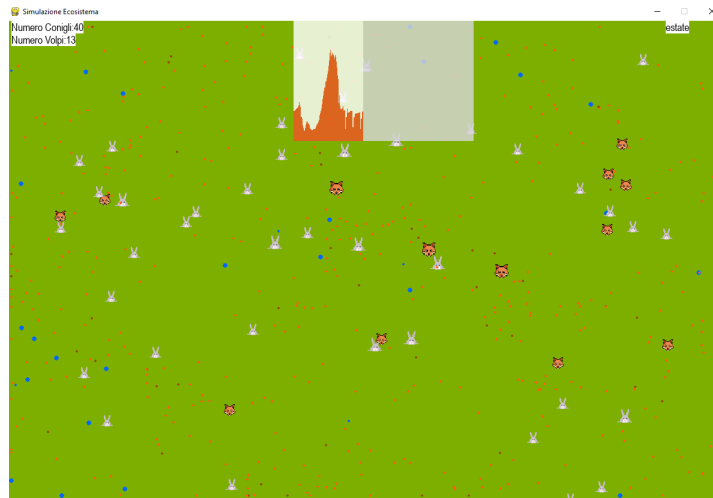
Al fine di realizzare il progetto di simulazione, si è deciso di utilizzare il linguaggio di programmazione python, sfruttando la libreria pygame per realizzare l'interfaccia grafica. Pygame, come descritto nel sito ufficiale[19], è un modulo python pensato per lo sviluppo di videogiochi che aggiunge funzionalità alla libreria multi-piattaforma SDL. Il suo utilizzo non è limitato esclusivamente allo sviluppo di videogame, ma è anche adatto per realizzare simulazioni multi-agente come nel caso del progetto corrente. Si è deciso di suddividere il codice in diversi moduli per garantirne una migliore chiarezza. Inoltre sono state usate diverse librerie aggiuntive, tra le quali:

- **random** per la gestione della simulazione è stato fatto un ampio utilizzo della generazione di valori pseudocasuali, attraverso questa libreria ed in particolare del metodo `randint(int1, int2)`, per ricreare al meglio la variabilità e l'entropia di un sistema complesso quale un ecosistema. Inoltre, si è deciso di memorizzare dopo ogni esecuzione il **seed**, in modo tale da poter, in caso di necessità, lanciare una nuova esecuzione ottenendo esattamente i medesimi risultati.
- **pandas** per la gestione dei dati attraverso l'utilizzo di dataframe e per la memorizzazione in file `.csv` per effettuarne lo studio.
- **matplotlib** per la creazione dei grafici, utili nella fase di analisi dei risultati.
- **os** e **sys** per l'interfacciamento con il sistema operativo: gestione dell'evento di chiusura della finestra, creazione delle cartelle e memorizzazione dei file (grafici e `.csv`).

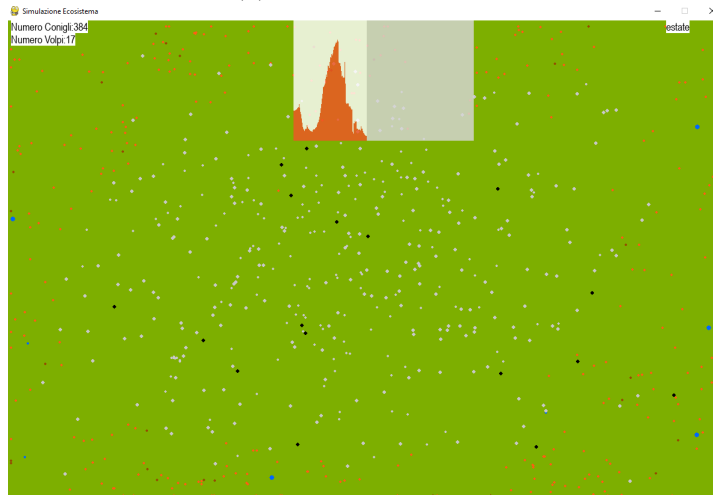
Il codice, separato in vari moduli, è stato così suddiviso:

- **Simulazione:** è il modulo principale in cui si effettua la simulazione vera e propria. Qui sono implementati i vari agenti sotto forma di classi. Viene sfruttato il concetto dell'ereditarietà per rappresentare gli agenti Volpe e Coniglio, che ereditano alcuni parametri comuni dalla super-classe Animale. Viene inoltre gestita la raccolta dei dati, memorizzando delle "fotografie" dello stato della simulazione ad ogni 250 iterazioni, e tutto ciò che riguarda il "game loop", ovvero la logica con cui pygame gestisce le iterazioni. Anche il tempo di esecuzione è gestito in questo modulo e viene calcolato sulla base del lasso temporale che passa da un'iterazione e l'altra e serve per regolare la quantità di pixel che ogni agente percorre in movimento. Ad ogni 500 iterazioni un controllo va a gestire il cambio delle stagioni e ciò che ne comporta, come il cambio di render del terreno e di diversi parametri di gestione della pioggia e delle carote. Una volta chiusa la finestra di simulazione questo modulo salva i dati in un file `.csv` che indirizzerà al modulo *Plot*. Prima di ogni esecuzione è possibile regolare alcuni parametri, che servono per regolare il funzionamento e la visualizzazione della simulazione. Essi sono:

- **GRAFICA:** se settato a *True* gli agenti animali verranno visualizzati con sprite 2D, se settato a *False* gli agenti animali saranno visualizzati come dei semplici pixel colorati. Quest’ultima configurazione è particolarmente adeguata per garantire una migliore fluidità nell’esecuzione della simulazione, soprattutto nel momento in cui il numero di agenti da visualizzare a schermo risulta essere particolarmente elevato.
  - **EVOLUZIONE:** se settato a *True* verrà lanciata la simulazione nella configurazione con evoluzione. Questo implica, come descritto nel paragrafo precedente, che ogni qualvolta nasce un cucciolo, esso erediterà i caratteri dei genitori ed eventualmente verrà applicata una mutazione, in modo tale da studiare il meccanismo di selezione evolutiva dei geni. Viceversa, se settato a *False* la simulazione verrà lanciata in modalità senza evoluzione. In questo caso il fenomeno della riproduzione degli animali porterà alla nascita di cuccioli i cui geni sono ereditati dai genitori senza che si presenti alcun tipo di mutazione.
  - **REINSERIMENTO:** se settato a *True* la simulazione viene lanciata in modalità con reinserimento. Questo significa, come già descritto nei paragrafi precedenti, che ogni qualvolta si manifesta un cambio di stagione verranno inseriti nell’ambiente alcuni nuovi animali. Viceversa, se settato a *False* non ci sarà alcun reinserimento.
  - **NUMERO CONIGLI/VOLPI/POZZANGHERE/CAROTE:** rappresenta il numero di agenti che all’inizio della simulazione popoleranno l’ambiente.
  - **PROBABILITA\_CRESCITA\_CAROTE:** rappresenta la probabilità che ad ogni iterazione una carota generi un’altra carota nelle vicinanze.
  - **QUANTITA\_ACQUA:** rappresenta la dimensione della pozzanghera, ovvero il numero di volte che essa può essere sfruttata da un animale prima che si esaurisca e quindi venga rimossa.
  - **WIDTH/HEIGHT:** rappresentano la dimensione della finestra di visualizzazione della simulazione, cambiarli non modifica solo l’aspetto visivo, ma va ad alterare anche l’esito della simulazione stessa a causa del fatto che vengono modificate le distanze tra gli agenti e questo può avere un impatto importante sull’esito della simulazione.
- **Ambiente:** modulo inizializzato all’interno di *Simulazione* che gestisce e regola l’ambiente della stessa. Al suo interno sono dichiarati i metodi per aggiungere, rimuovere, processare ed effettuare il render degli agenti. Inoltre vi sono inizializzate alcune variabili, descritte nel paragrafo 5.1.1.
  - **Vector2D:** collezione di metodi per la gestione delle coordinate e dello spazio vettoriale. Vengono utilizzati metodi per ottenere ed elaborare la posizione, la distanza, il modulo (magnitudine), la norma e la direzione dei punti sul piano 2D.



(a) GRAFICA = True

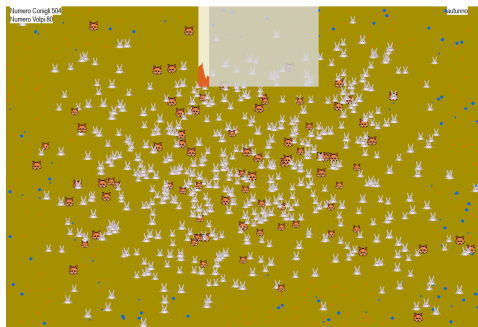


(b) GRAFICA = False

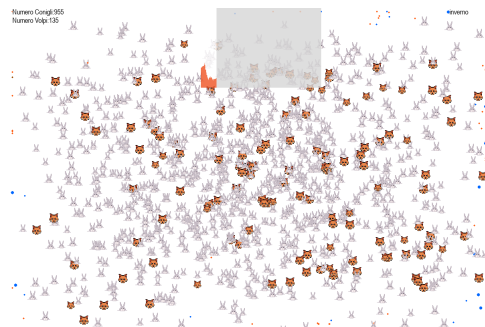
Figura 17: Visualizzazione della simulazione

- **Plot:** modulo che utilizza come input il file *.csv* generato dalla simulazione e ne visualizza i valori creando diversi grafici e memorizzandoli in cartelle apposite, nominate in modo tale da riportare il seed della simulazione e un tag che identifica la configurazione con cui è stata lanciata la simulazione (tag E significa evoluzione, tag R significa reintroduzione, nessun tag significa che la simulazione è stata lanciata senza reintroduzione e senza evoluzione). I grafici generati per ogni simulazione sono i seguenti:
  - *conigli-carote.png*: mostra la differenza tra le popolazioni di conigli e carote;
  - *età.png*: mostra le differenze dell'età media tra le due specie;
  - *fame.png*: mostra le differenze di fame media tra le due specie;

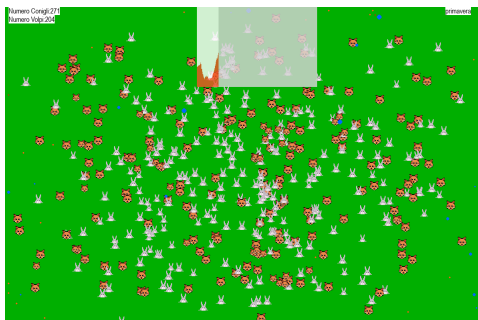




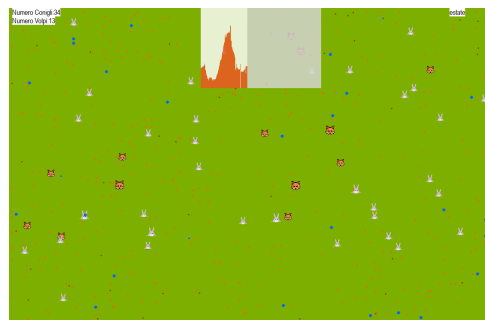
(a) Autunno



(b) Inverno



(c) Primavera



(d) Estate

Figura 18: Cambi di stagione nella simulazione

- *sete.png*: mostra le differenze di sete media tra le due specie;
- *fertilita.png*: mostra le differenze di fertilità media tra le due specie;
- *morti\_conigli.png*: mostra le cause di morte dei conigli;
- *morti\_volpi.png*: mostra le cause di morte delle volpi;
- *nascite.png*: mostra le nascite delle due specie a confronto;
- *popolazioni.png*: mostra le popolazioni degli animali a confronto;
- *risorse.png*: mostra le risorse a confronto;
- *percezione.png*: mostra la percezione media degli animali;
- *soglia\_fame\_media.png*: mostra la soglia fame media degli animali;
- *soglia\_sete\_media.png*: mostra la soglia sete media degli animali;
- *soglia\_morte\_di\_fame\_media.png*: mostra la soglia di morte di fame media degli animali;
- *soglia\_morte\_di\_sete\_media.png*: mostra la soglia di morte di sete media degli animali;
- *velocita\_camminata\_media.png*: mostra la velocità di camminata media degli animali;
- *velocita\_corsa\_media.png*: mostra la velocità di corsa media degli animali;

- *stackedBarchart\_cause\_morti\_volpi.png*: è uno stacked barchart che mostra come sono suddivise il numero e le cause di morte delle volpi nelle quattro stagioni;
- *stackedBarchart\_cause\_morti\_conigli.png*: analogo a quello descritto nel punto precedente, ma relativo ai conigli;
- *conigliVsVolpi\_NoTimeSeries.png*: è un grafico che mostra sull'asse delle ascisse il numero di prede e sulle ordinate il numero di predatori. Questo grafico è risultato utile per eseguire la validazione del modello;
- *caroteVsConigli\_NoTimeSeries.png*: analogamente a quello del punto precedente, mostra sull'asse delle ascisse il numero di carote e su quello delle ordinate il numero di conigli. Anche questo è risultato utile in fase di validazione del modello.

## 7 Analisi dei risultati

Per collezionare i dati ed analizzare i risultati ottenuti, sono state lanciate svariate simulazioni relative a diverse configurazioni del modello. Tra le varie configurazioni testate, le tre principali per le quali è stata condotta un'analisi dei risultati sono le seguenti:

- Senza reintroduzione e senza evoluzione;
- Con reintroduzione e senza evoluzione;
- Con reintroduzione ed evoluzione;

### 7.1 Configurazione senza reintroduzione e senza evoluzione

La prima configurazione è stata implementata in modo tale che non vi fosse una periodica reintroduzione degli agenti animale nell'ambiente e che non venisse simulata l'evoluzione genetica degli individui, per tale ragione è anche la configurazione più semplice delle tre e quella i cui risultati si sono rivelati essere poco soddisfacenti. Per questo motivo, essa è stata un punto di partenza dalla quale sono state realizzate le configurazioni successive che hanno dato risultati decisamente più attendibili. La differenza tra un sistema con reintroduzione e senza reintroduzione riguarda il fatto che nel secondo caso alcuni animali (sia prede che predatori) vengono inseriti all'interno dell'ambiente con una certa frequenza, mentre nel primo caso questa operazione non viene mai eseguita, lasciando quindi l'ambiente totalmente chiuso.

Questa prima configurazione, nonostante gli esiti non siano stati del tutto soddisfacenti, è stata particolarmente utile per effettuare una prima operazione di

tuning ottimale dei vari parametri coinvolti dalla simulazione e descritti nei paragrafi precedenti. Per esempio, si è tentato di trovare una soluzione che permettesse di ottenere un trend verosimile delle cause di morte degli animali, facendo in modo che la frequenza dei decessi causati dalla vecchiaia risultassero inferiori rispetto a quelli dovuti ad altre cause (fame, sete e caccia), che invece sarebbero dovute risultare bilanciate tra loro.

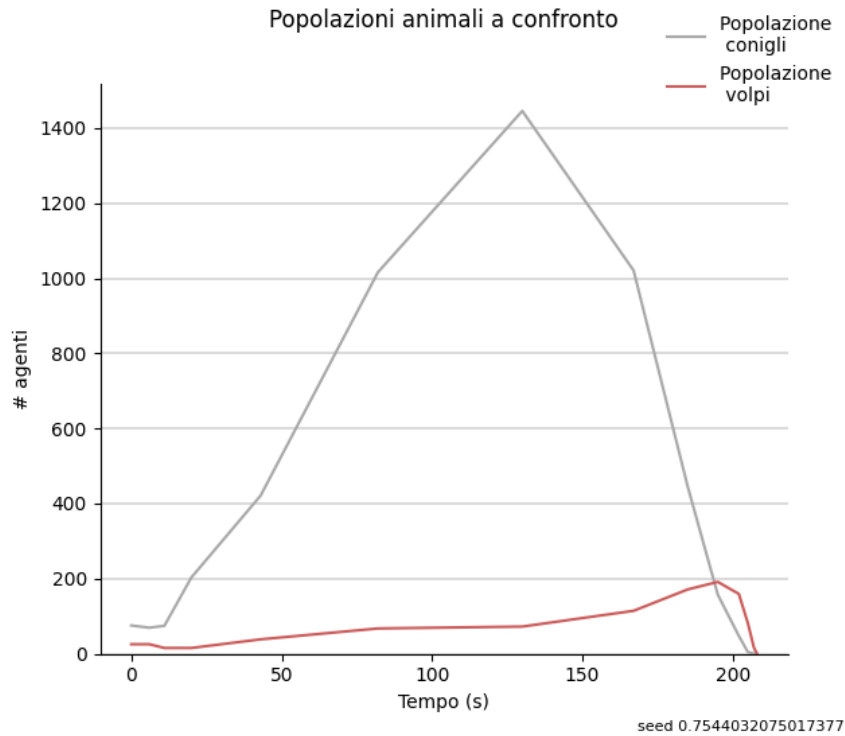
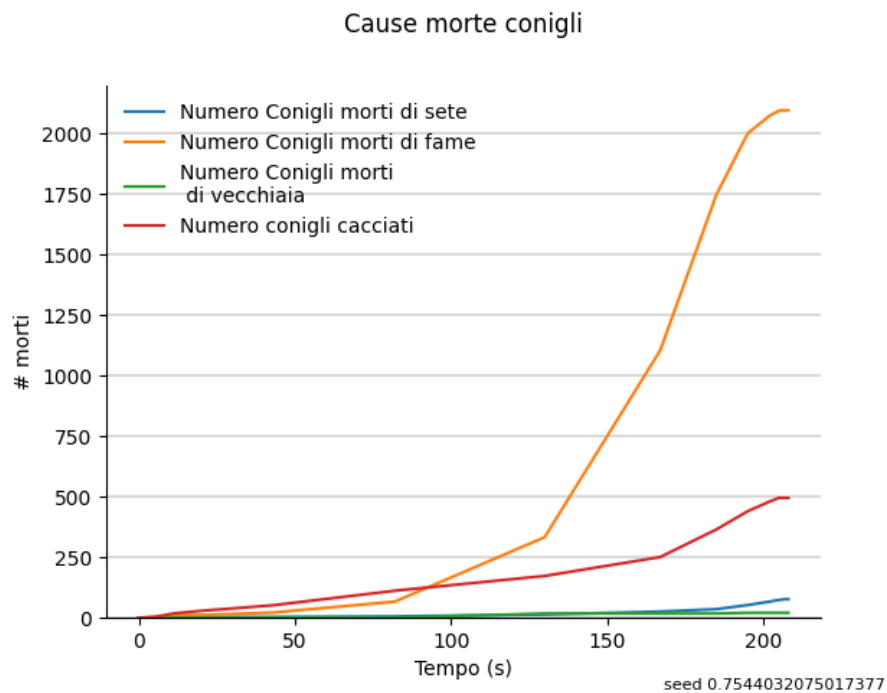
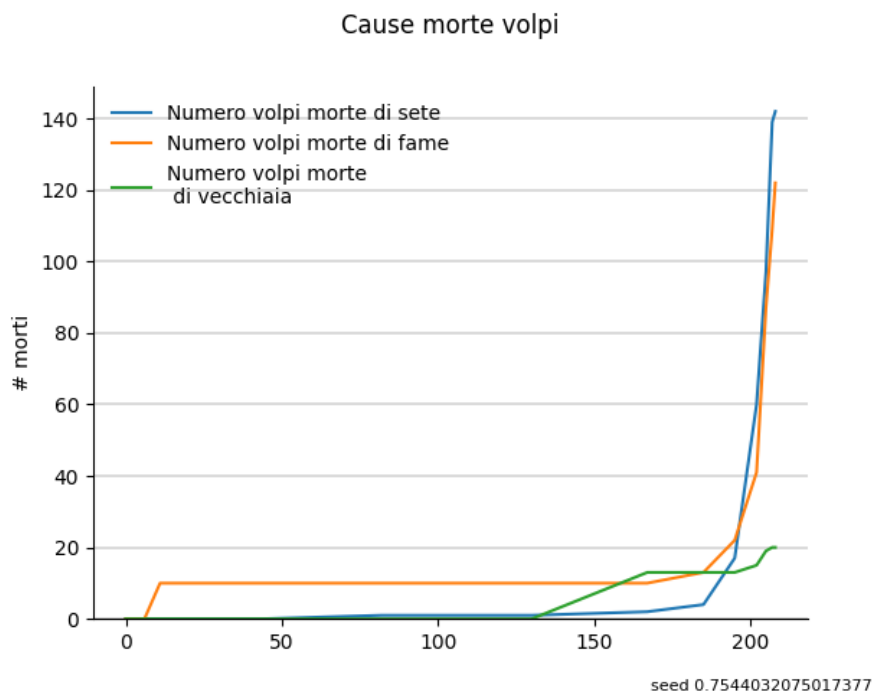


Figura 19: Grafico popolazione volpi vs popolazione conigli

Il grafico in **figura 19** mostra l'andamento nel tempo delle popolazioni delle prede e dei predatori ottenuto con questa prima configurazione del modello. Si può osservare che i conigli inizialmente crescono notevolmente in numero, ma quando saturano le risorse disponibili nell'ambiente iniziano a morire di fame, e anche a causa della predazione condotta dai predatori, la popolazione si estingue e di conseguenza anche il numero di volpi tende a decrescere fino a quando non si estinguono anch'esse, a causa della mancanza di prede di cui cibarsi. Possiamo quindi concludere che il modello è risultato essere instabile, in quanto, in quasi tutte le esecuzioni, si è verificata l'estinzione di entrambe le specie di animali.



(a) Cause morte dei conigli



(b) Cause morte delle volpi

Figura 20: Grafici cause morte degli animali

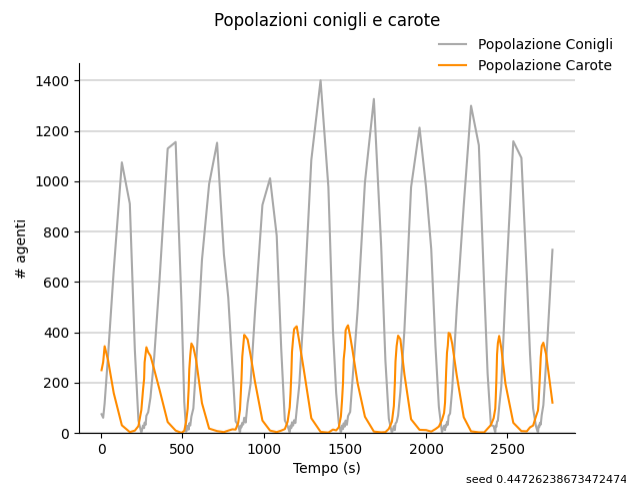
I grafici **20a** e **20b** mostrano lo sviluppo nel tempo delle cause di morte, rispettivamente delle prede e dei predatori. Si può notare che la causa di morte prevalente nella popolazione dei conigli è la fame, seguita dalla caccia da parte delle volpi. A causa della brevità della simulazione, un numero molto basso di conigli tende a morire di sete. Per lo stesso motivo solamente 21 conigli sono risultati morti di vecchiaia nell'esecuzione mostrata nei grafici riportati.

Per quanto riguarda le volpi, le cause di morte più frequenti sono risultate essere la sete e la fame. Il numero di volpi morte di vecchiaia è risultato essere uguale a quello dei conigli morti per lo stesso motivo.

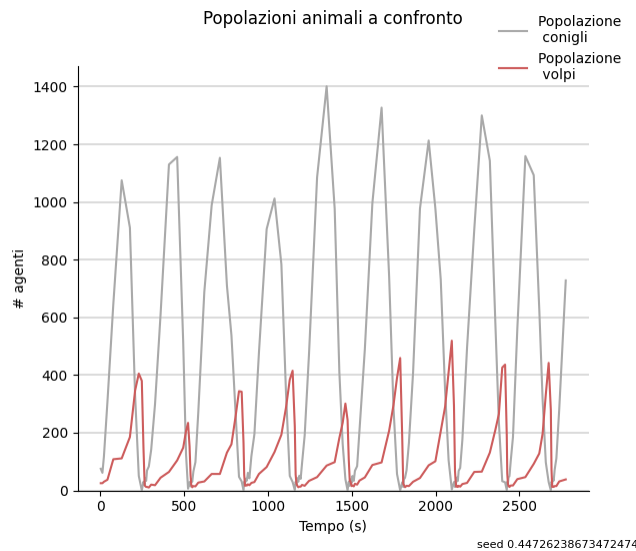
Come è possibile notare dalle **figure 19, 20a e 20b**, il tempo di esecuzione della simulazione è molto basso, infatti si conclude in poco più di 200 secondi. Il sistema infatti, risulta essere piuttosto instabile in quanto gli animali tendono ad estinguersi completamente dopo poco tempo dall'inizio della simulazione. Per andare oltre questo problema, si è deciso di estendere ulteriormente il modello realizzando un approccio alternativo con reinserimento, i cui risultati sono esposti nella sezione successiva.

## 7.2 Configurazione con reintroduzione senza evoluzione

L'idea di questa configurazione è quella di non pensare più all'ambiente come una scatola chiusa, ma far sì che ci sia la possibilità che nuovi animali (sia volpi che conigli) possano entrare nell'ambiente dall'esterno, in modo tale da simulare le migrazioni di animali dalle aree circostanti, come se lo spazio che viene visualizzato sia una piccola porzione posizionata in un ambiente più grande. Seguendo questa logica, è stato fatto in modo che, ogni qualvolta si verifica un cambio di stagione durante la simulazione, venissero inseriti volpi e conigli nell'ambiente in numero proporzionale rispetto alla quantità di animali presenti all'inizio della simulazione stessa.



(a) Grafico popolazione conigli vs popolazione carote



(b) Grafico popolazione conigli vs popolazione volpi

Dal **grafico 21a** è possibile notare innanzitutto che il sistema così esteso risulta essere stabile, a differenza del modello nella configurazione presentata in precedenza, in quanto non si verifica mai durante la simulazione l'estinzione di tutti gli animali. Il numero delle carote e dei conigli presenti nel sistema segue un andamento di questo tipo: inizialmente i conigli crescono in numero, fino a diventare circa 1100 e contemporaneamente le carote vengono completamente mangiate fino a quando non ne resta praticamente nessuna; a questo punto il numero di conigli inizia a diminuire drasticamente a causa del fatto che nell'ambiente non è presente abbastanza cibo per poter sostenere tutta la popolazione. Nel contempo, il numero di carote inizia a crescere e all'incirca quando il numero di conigli ha raggiunto il minimo locale, esso raggiunge il massimo. Dopodiché, il numero di conigli può tornare a crescere, dato che ci sono abbastanza carote nell'ambiente. Tutto ciò si ripete ciclicamente fino al termine della simulazione.

La **figura 21b** mostra invece l'andamento delle popolazioni dei conigli e delle volpi nel corso della simulazione. Come si vede dall'immagine, il trend è esattamente analogo a quello delle popolazioni dei conigli e delle carote discusso in precedenza. In entrambi i casi infatti, l'andamento segue all'incirca quello di due sinusoidi sfasate che si ripetono ciclicamente.

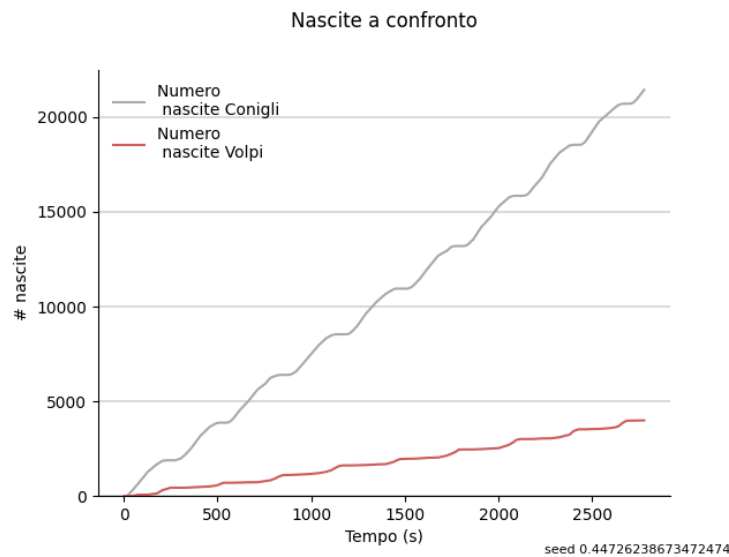
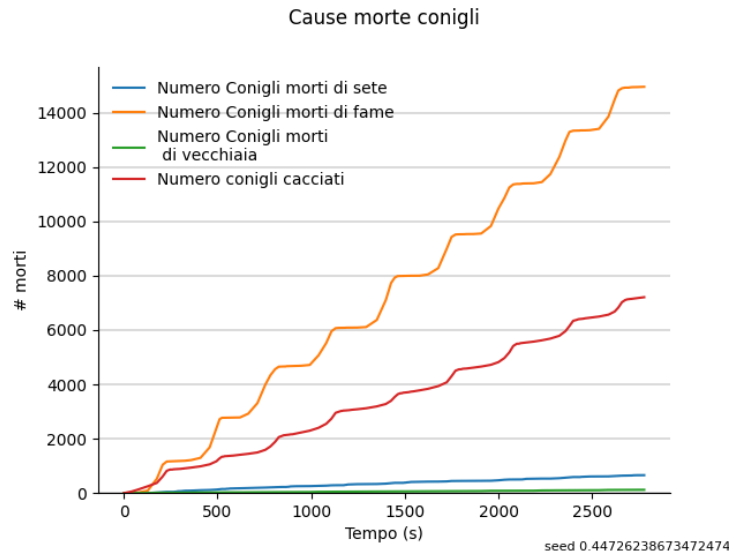
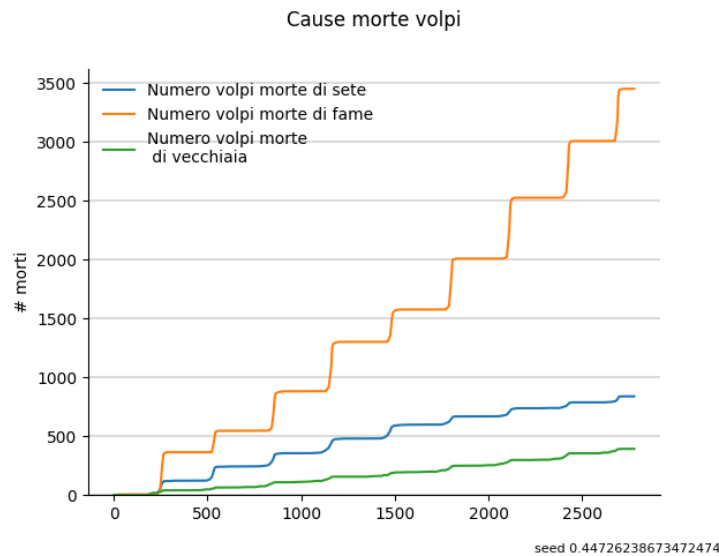


Figura 22: Grafico delle nascite dei conigli e delle volpi

La **figura 22** mostra come evolvono le nascite all'interno delle due popolazioni. Come, prevedibile, è risultato che i conigli si riproducono con una frequenza significativamente superiore rispetto alle volpi.



(a) Cause di morte conigli



(b) Cause di morte volpi

Figura 23: Grafici cause di morte animali

Il **grafico 23a** mostra l'andamento del numero di decessi dei conigli suddivisi in base alla causa. Si può notare che anche in questo caso i conigli muoiono perlopiù di fame a causa del fatto che, come appurato dal grafico mostrato in **figura 22**, essi tendono a riprodursi frequentemente e quindi questo porta alla competizione tra gli individui per ottenere le carote disponibili. Molto rilevante è anche il numero di morti causate dalla caccia da parte dei predatori. Invece, il numero di conigli morti di sete e di vecchiaia risulta essere piuttosto contenuto.

Dalla **figura 23b**, che rappresenta il grafico relativo all'andamento delle numero di decessi delle volpi e le loro cause, si può notare che anche in questo caso



la causa di morte prevalente tra le volpi è la fame. Questo perché nei momenti in cui sono presenti pochi conigli nell'ambiente le volpi non trovano prede con cui sfamarsi e di conseguenza muoiono. La seconda causa più rilevante è la sete e infine vecchiaia. La ragione per la quale sia per le volpi che per i conigli la vecchiaia sia solo l'ultima delle cause di morte è che gli individui di entrambe le popolazioni difficilmente riescono a raggiungere l'età della vecchiaia perché tendenzialmente muoiono prima a causa della scarsità delle risorse (acqua per entrambe le specie, carote per i conigli e conigli per le volpi) che si verifica ciclicamente nella simulazione.

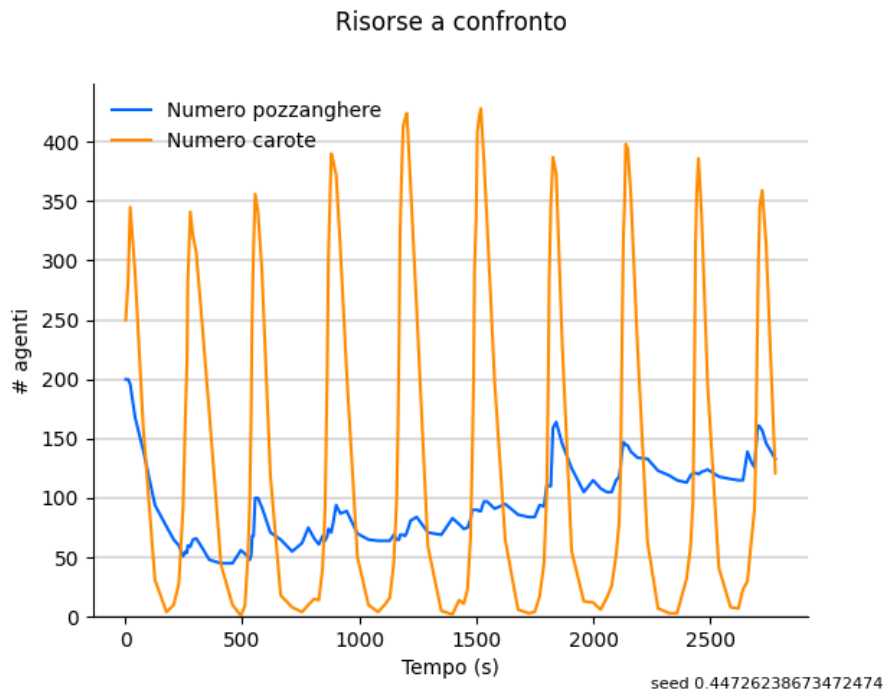


Figura 24: Grafico delle risorse

Oltre a ciò, dal grafico in **figura 24** si può notare che il numero di pozzanghere presenti nell'ambiente rimane sempre abbastanza equilibrato a differenza delle carote che tendono a esaurirsi in modo ciclico. É questa la ragione per cui gli individui di entrambe le popolazioni raramente muoiono di sete.

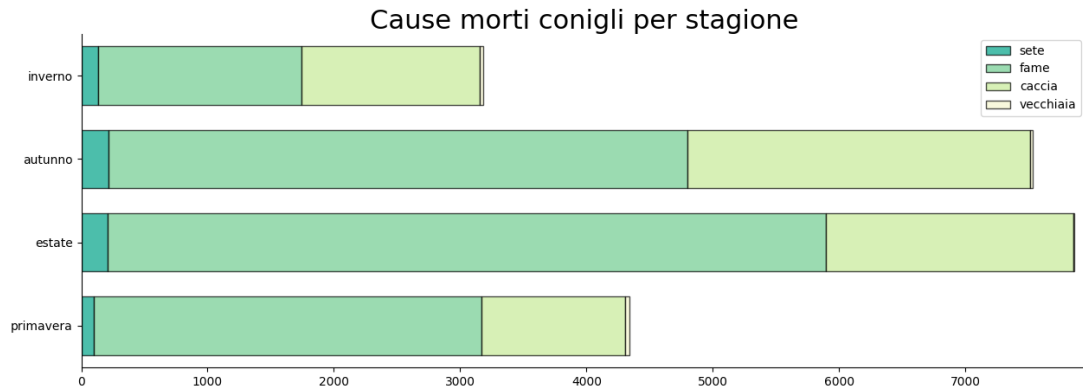


Figura 25: Barchart delle morti dei conigli

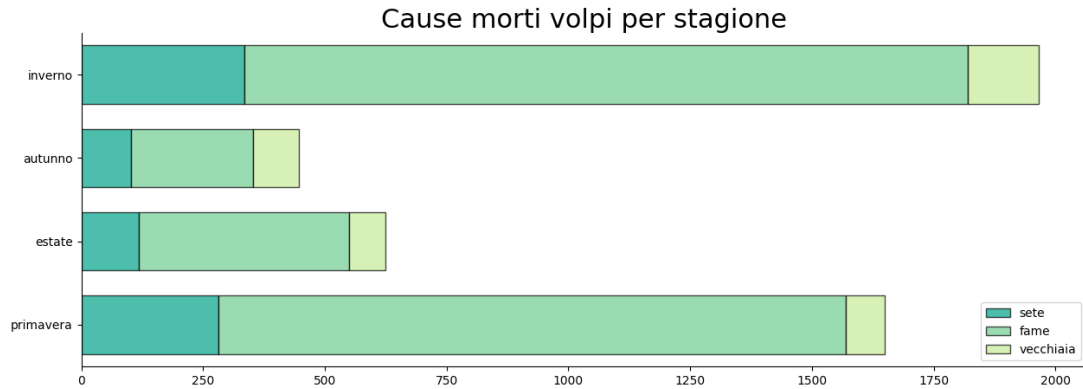
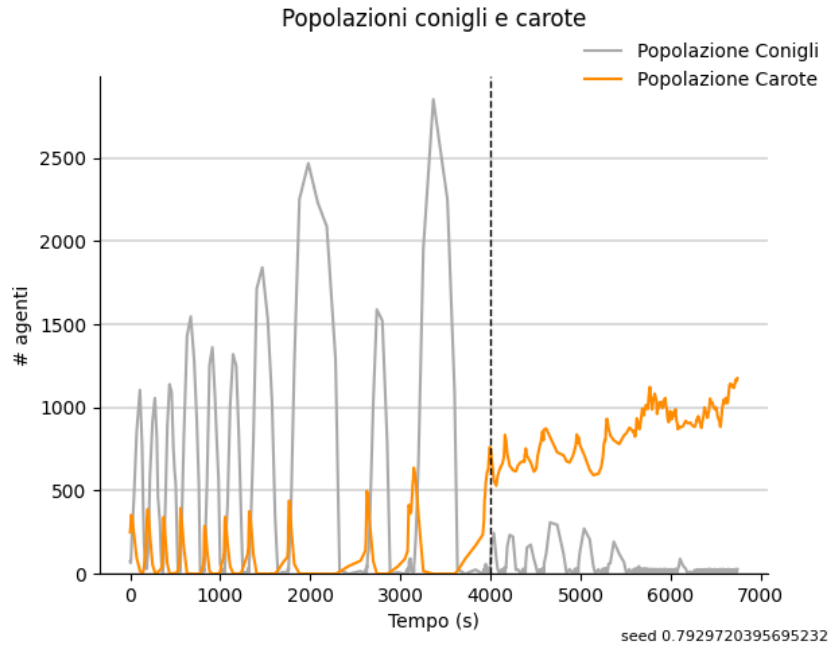


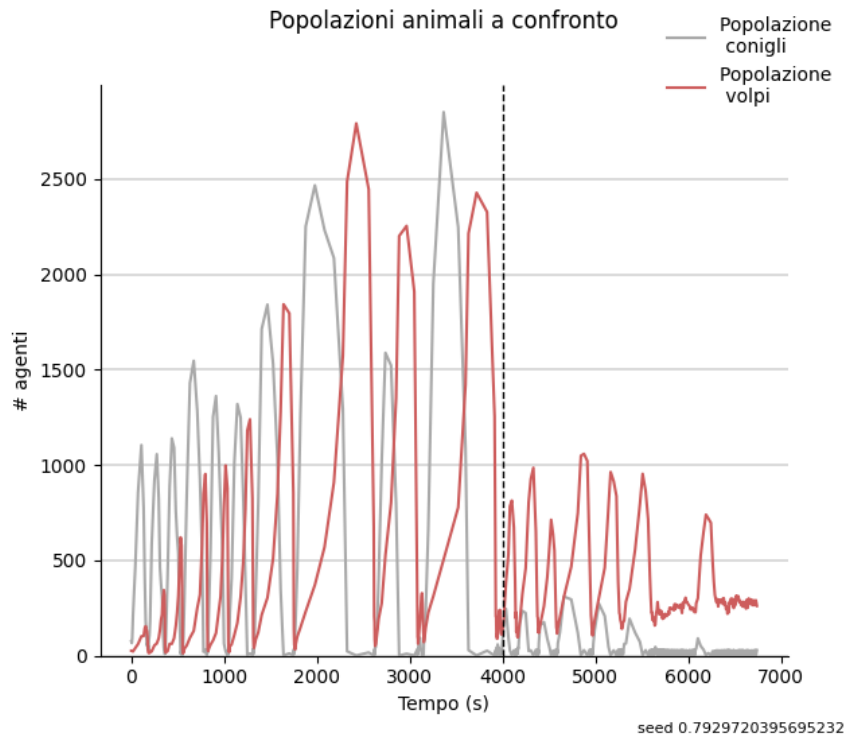
Figura 26: Barchart delle morti delle volpi

Infine, i grafici mostrati nelle **figure 25 e 26** indicano come sono ripartite le morti, rispettivamente dei conigli e delle volpi, rispetto alle stagioni. I decessi dei conigli avvengono prevalentemente in autunno e in estate e la causa principale è la mancanza di cibo. Le volpi invece tendono a morire prevalentemente in inverno e in primavera e anche in questo caso la causa principale riguarda la mancanza di prede di cui cibarsi. Una possibile spiegazione riguarda il fatto che, come analizzato, i conigli tendono a morire perlopiù in autunno e in estate, quindi il numero di prede presenti nell'ambiente tende a calare in queste stagioni. L'effetto sui predatori però non è immediato, ma inizia a presentarsi solo nelle stagioni successive in inverno e in primavera quando il numero di morti delle volpi a causa della mancanza di prede cresce in modo notevole rispetto alle altre due stagioni.

### 7.3 Configurazione con reintroduzione ed evoluzione



(a) Popolazione conigli vs popolazione carote



(b) Popolazione conigli vs popolazione volpi

Figura 27: Grafici delle popolazioni

Questa configurazione consente di effettuare un ulteriore passo avanti, in quanto permette di simulare anche l'evoluzione genetica negli animali. L'obiettivo è stato quello di verificare se l'aggiunta di un fenomeno di mutazione delle caratteristiche degli animali potesse in qualche modo cambiare l'esito della simulazione, e quindi l'equilibrio dell'ecosistema. Per rendere compatibile il processo evolutivo con il reinserimento, è stato assunto che gli animali al di fuori dell'ambiente visualizzato subiscano la stessa pressione evolutiva di quelli visualizzati nell'ambiente simulato. Di conseguenza, ogni volta che vi è reintroduzione, i parametri dei nuovi animali inseriti nell'ambiente saranno uguali alla media di quelli degli animali già presenti nello spazio simulato, portando così avanti il fenomeno di selezione naturale. Questa configurazione rappresenta il modello finale e più complesso delle tre analizzate e quello che ha portato ai risultati più interessanti e significativi.

L'andamento della simulazione nel tempo può essere suddiviso in due fasi, che nei grafici di seguito sono state evidenziate attraverso una linea verticale tratteggiata. Il primo stadio, che nei grafici va dall'inizio a circa l'istante 4000 della simulazione, vede i conigli sfruttare l'unica "arma" a loro disposizione per contrastare il dominio delle volpi, ovvero la loro elevata capacità riproduttiva, come mostrato chiaramente nella **figura 29b**. I conigli tendono ad evolvere abbassando drasticamente la loro soglia di fertilità, in modo tale che si verifichi un'esplosione di nascite che permette alla popolazione delle prede di contrastare la capacità predatoria delle volpi. Come possibile notare nella **figura 27b**, questo porta ad una situazione nella quale i conigli risultano essere in netta maggioranza rispetto alle volpi, raggiungendo picchi di quasi 3000 individui. Ciò però comporta anche diversi malus per i conigli stessi. Infatti, questa situazione porta ad una rapida saturazione delle risorse (**figura 27a**) e ad un aumento del numero di volpi proporzionale, se pur più lento, a quello dei conigli. In questo modo, viene garantita alle volpi l'energia necessaria per poter continuare a riprodursi.

La seconda fase, che inizia dall'istante 4000 e si prolunga fino al termine della simulazione, invece è caratterizzata da un dominio delle volpi nell'ambiente. Esse infatti, acquisiscono un controllo territoriale totale. Questo è dovuto al fatto che i predatori, col passare del tempo, migliorano costantemente le loro caratteristiche, fino ad ottenere un vantaggio rispetto alle prede tale che molte di esse vengono cacciate prima che si possano riprodurre. Questo, come mostra la **figura 28c**, produce un rallentamento nelle nascite dei conigli. Tuttavia, nel lungo periodo, a causa del ridotto numero di prede presenti nell'ambiente, la crescita del numero di volpi subirà una frenata, producendo come effetto un nuovo equilibrio tra le due specie.

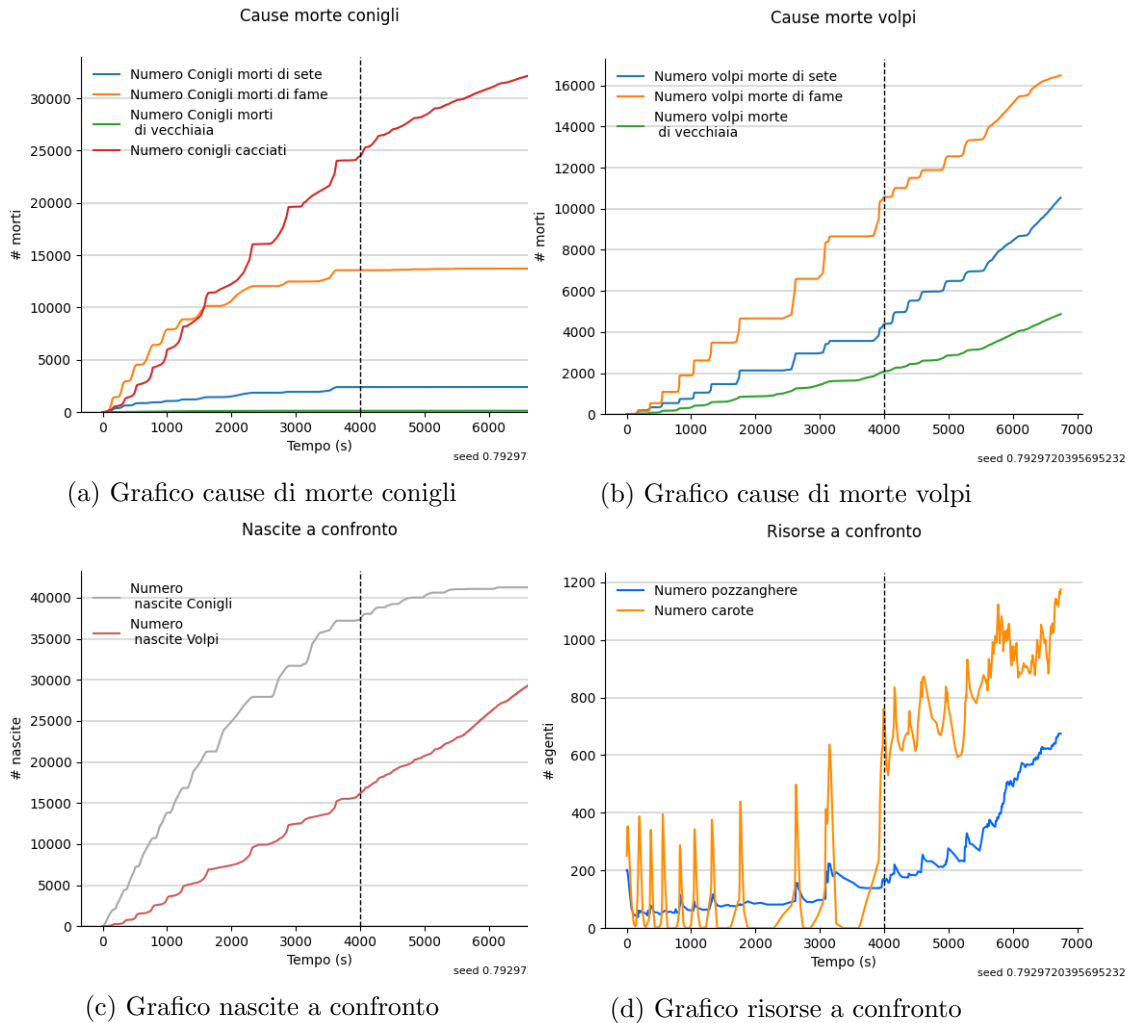
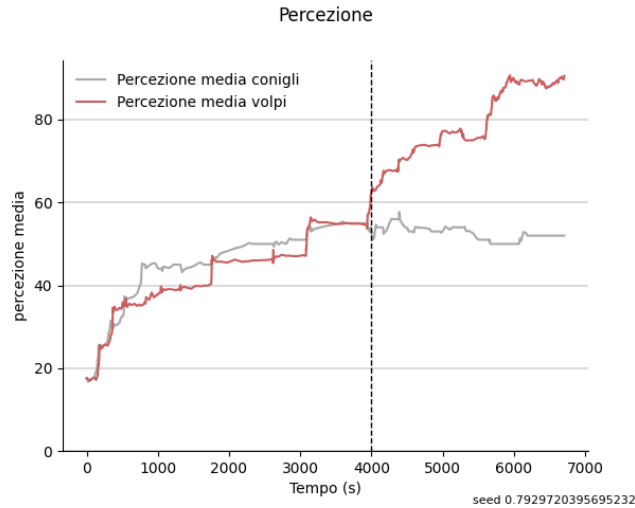


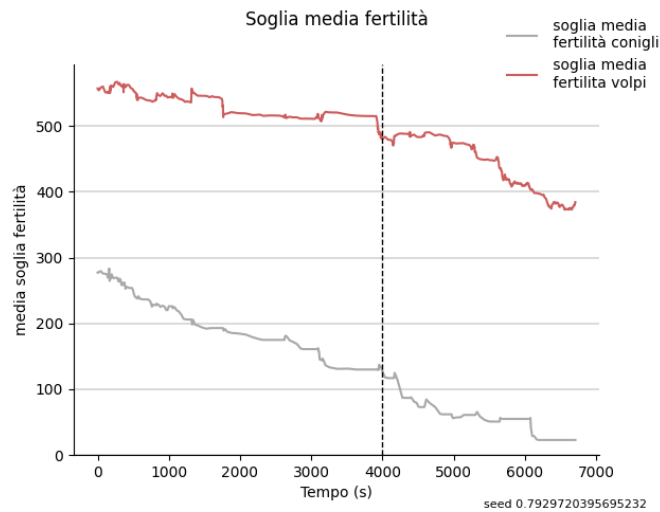
Figura 28

- La **figura 28a** mostra come le cause di morte dei conigli inizialmente siano bilanciate tra fame e caccia. Però quest'ultima diventa predominante nella seconda fase della simulazione nella quale le volpi prendono il sopravvento sui conigli. La **figura 28b** mostra invece come le cause di morte relative alle volpi siano bilanciate tra loro. Nella fase iniziale è possibile notare dei "salti" relativi ai momenti in cui i conigli proliferano e momenti in cui la mancanza di prede e acqua, porta a un elevato numero di decessi nella popolazione delle volpi.
- La **figura 28c** mostra chiaramente la dicotomia tra le due fasi; la prima in cui i conigli si diffondono vertiginosamente e la seconda in cui si arriva ad un punto asintotico in cui il numero delle nascite degli stessi subisce un forte rallentamento dovuto all'evoluzione delle caratteristiche delle volpi.

- La **figura 28d** mostra una prima fase di carenza di carote e pozzanghere presenti nell'ambiente e una seconda in cui l'assenza dei conigli comporta una crescita costante delle stesse.

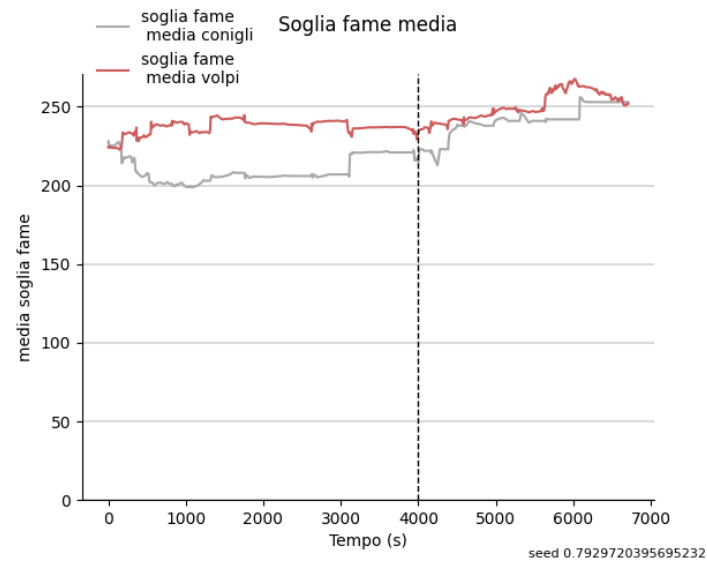


(a) Grafico andamento della caratteristica percezione media

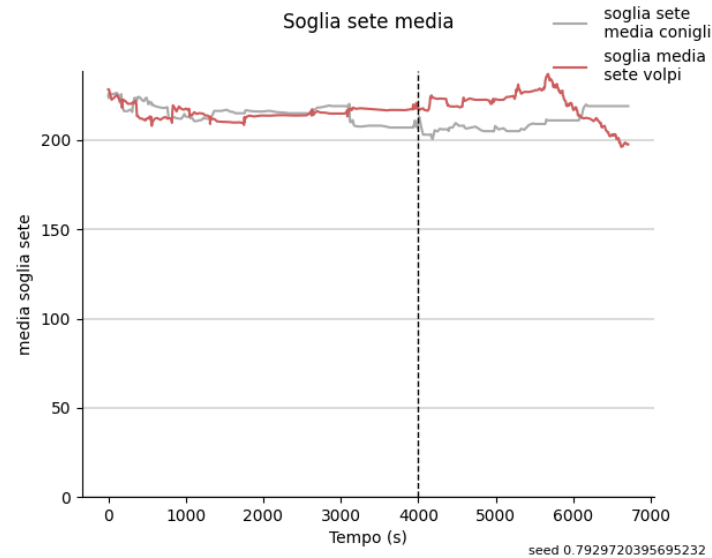


(b) Grafico andamento della caratteristica soglia fertilità media

- La **figura 29a** mostra quanto l'evoluzione giochi un ruolo fondamentale in questo modello. Le volpi arrivano a quadruplicare la loro percezione base, andando ad ottenere un controllo sul territorio molto più esteso.
- La **figura 29b** mostra come tra le due popolazioni sia quella dei conigli a trarre un enorme vantaggio dall'abbassamento della soglia di fertilità. Il valore medio, negli ultimi istanti della simulazione, addirittura sfiora lo zero, confermando quanto sia importante per i conigli questa caratteristica per contrastare il diffondersi dei predatori nel territorio.



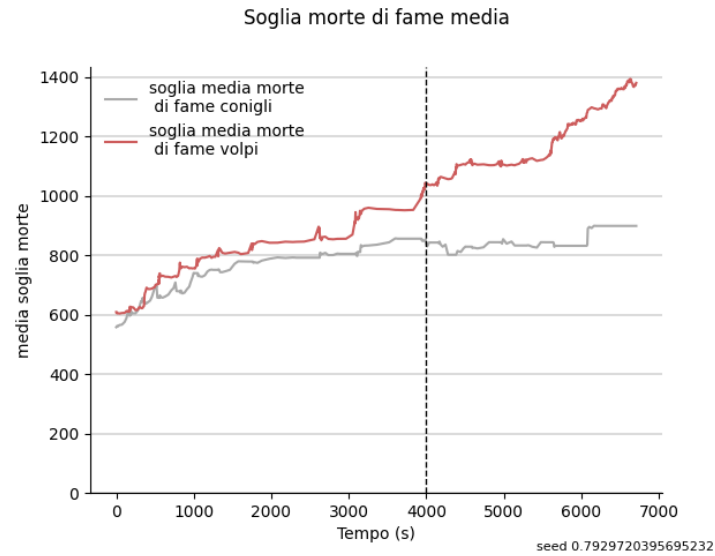
(a) Grafico andamento della caratteristica soglia fame media



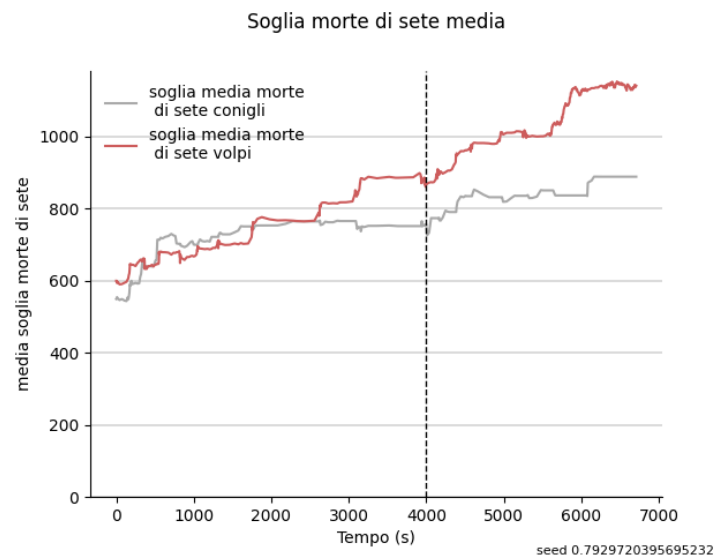
(b) Grafico andamento della caratteristica soglia sete media

Figura 30

- Le **figure 30a e 30b** sono simili e mostrano un dato interessante: le soglie di fame e sete non hanno un grande peso nell'evoluzione degli animali. Questo probabilmente è dovuto al fatto che all'inizio della simulazione le due soglie degli individui di entrambe le popolazioni assumono valori piuttosto bassi e quindi un'ulteriore diminuzione delle stesse non porta un vantaggio evolutivo significativo.



(a) Grafico andamento della caratteristica soglia morte di fame media

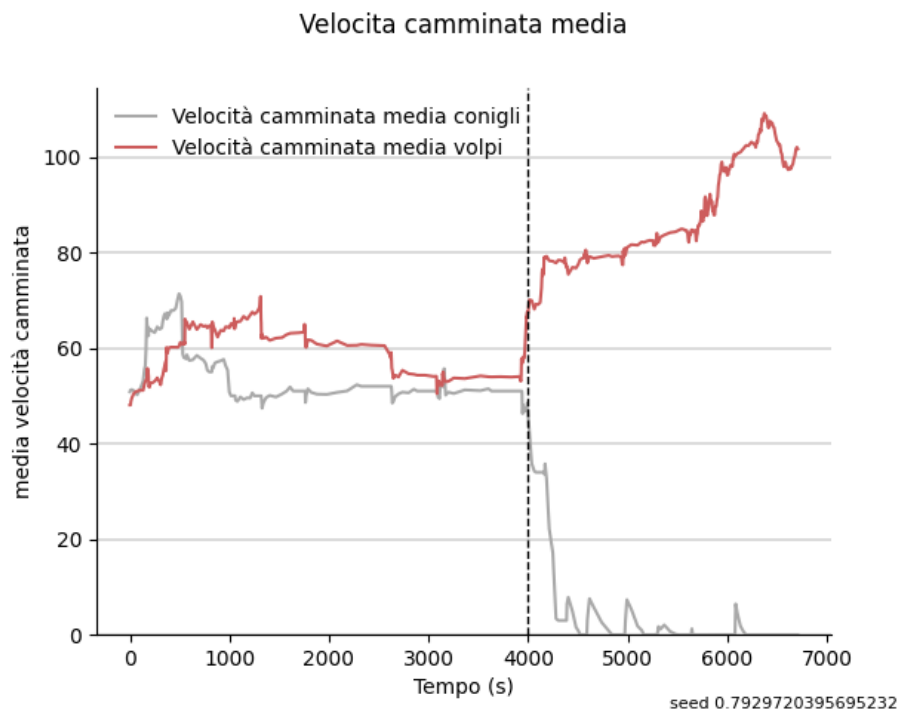


(b) Grafico andamento della caratteristica soglia morte di sete media

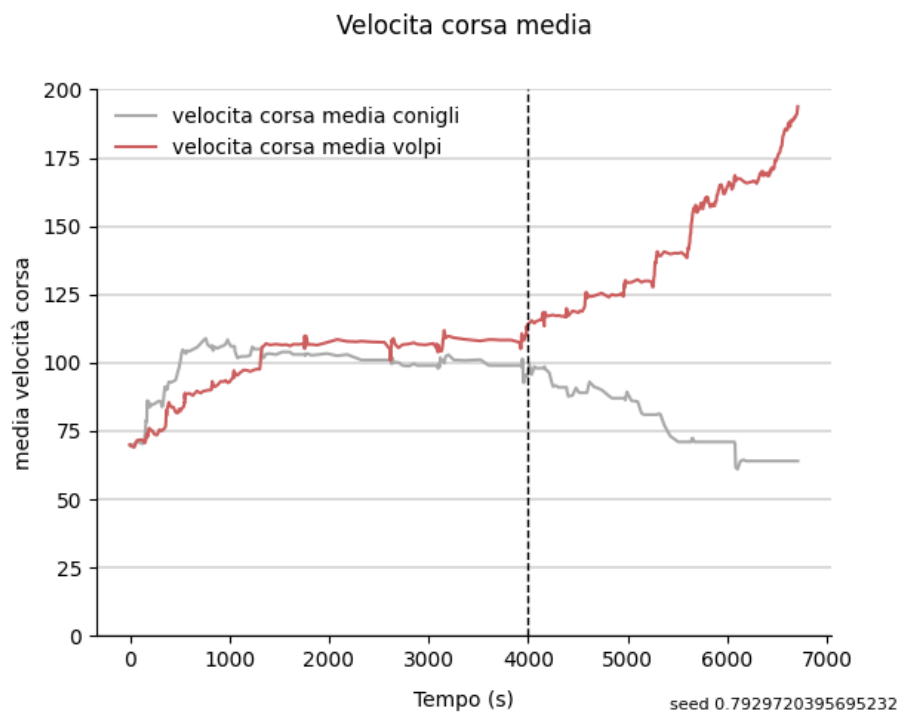
Figura 31

- Le **figure 31a** e **31b** sono altrettanto equiparabili e mostrano come le soglie di morte di fame e di sete invece, siano caratteristiche che, tramite l'evoluzione, possono permettere agli individui di adattarsi meglio all'ambiente permettendogli di sopravvivere più a lungo. Saranno gli agenti che hanno una maggiore resistenza alle carenze di risorse, quelli che si adatteranno meglio e che quindi avranno più possibilità di sopravvivere e trasmettere alla prole questa caratteristica.





(a) Grafico andamento della caratteristica velocità camminata media



(b) Grafico andamento della caratteristica velocità corsa media

Figura 32

- Le **figure 32a** e **32b** rappresentano i risultati più sorprendenti. E' possibile notare come, se nella prima parte della simulazione i valori non subiscano cambiamenti significativi rispetto alle velocità di corsa e di camminata, nella seconda invece, si crei una vera e propria divergenza tra lo stesso parametro nelle popolazioni delle due specie. Una possibile interpretazione relativa a questa differenza è la seguente:
  - Le volpi, dopo aver sovrastato in numero la popolazione delle prede, hanno a disposizione pochi conigli di cui cibarsi e dunque questo determina il fatto che saranno gli esemplari più veloci quelli che riusciranno a raggiungere con più facilità le poche prede rimaste, evitando così di morire di fame. In altre parole, il principio di selezione naturale favorirà gli individui più rapidi, che saranno facilitati nel catturare le prede.
  - I conigli invece, in situazione di netto svantaggio rispetto ai predatori, comprendono che la soluzione migliore sia quella di muoversi il meno possibile, evitando di esporsi alle minacce. Infatti, si nota come la velocità di corsa rimanga stabile, ma la loro velocità di camminata tende a decrescere sempre più. Questo significa che i conigli a lungo termine tendono a evitare di effettuare movimenti, a meno che non ci sia un obiettivo nelle immediate vicinanze. Questo può essere interpretato come un modo per i conigli di nascondersi dai predatori, proprio come nella realtà fanno attraverso l'utilizzo di tane. È proprio questo comportamento a dare il nome a questi mammiferi.<sup>2</sup>.

---

<sup>2</sup>La parola coniglio deriva da dal latino "*cuniculus*", cioè buca sotterranea.[7]

### 7.3.1 Validazione del modello

Al fine di verificare che il modello realizzato corrispondesse a ciò che ci si attendeva, è stata eseguita una operazione di validazione dello stesso, descritta in questo paragrafo. Il modello preso in considerazione per questa fase è quello con reintroduzione senza evoluzione, descritto nella sezione 7.2. Le ragioni per le quali si è deciso di effettuare la validazione solo per questa configurazione sono le seguenti:

- Una assunzione di gran parte dei modelli presenti in letteratura, e in particolare del modello Lotka-Volterra, è che gli individui che popolano l'ambiente non subiscano alcun tipo di evoluzione. Per questa ragione si è deciso di effettuare questa operazione solo per il modello più completo tra i tre realizzati che non prevede la simulazione dell'evoluzione;
- Gli effetti dell'evoluzione genetica in natura sono riscontrabili solo dopo un lunghissimo periodo di tempo. La creazione di un modello di simulazione che tiene conto dell'evoluzione è quindi una semplificazione di un modello a lunghissimo termine. In letteratura non esistono dati di osservazioni di evoluzioni genetiche a lungo termine con cui poter eseguire un confronto rispetto ai dati ottenuti con la simulazione.

Per effettuare questo lavoro sono stati considerati i risultati del modello lotka-volterra, descritto nella sezione 3.3. Oltre a ciò, è stato preso in considerazione anche un set di dati relativi all'interazione della popolazione delle linci canadesi e quella delle lepri scarpa di neve (*Lepus americanus*), raccolti grazie alle osservazioni durate quasi un secolo condotte dalla Hudson Bay Company[20].

Per prima cosa, è possibile affermare che il modello nella seconda configurazione risulta essere stabile e la sua esecuzione viene portata avanti per un tempo indefinito senza interrompersi, a differenza del modello realizzato nella prima configurazione.

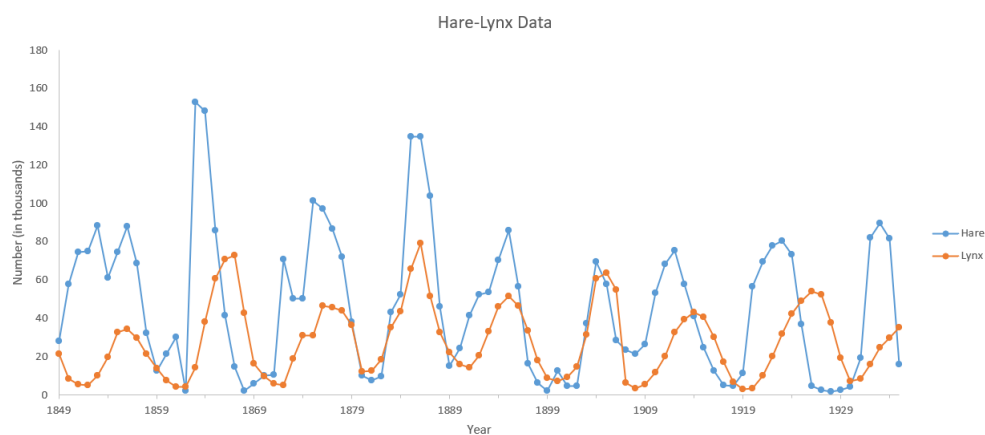


Figura 33: Grafico dell'andamento del numero di individui delle popolazioni di linci canadesi e di lepri scarpa di neve

La **figura 33** mostra l'evoluzione delle popolazioni delle linci canadesi e delle lepri scarpa di neve, osservate dalla Hudson Bay Company[21]. Confrontando tale grafico con quelli presentati in precedenza nelle **figure 21a** e **21b**, si nota che l'andamento è molto simile. Infatti, inizialmente il numero di prede è superiore al numero di predatori, successivamente dopo un momento in cui il numero di predatori cala ulteriormente, esso tende a risalire e contemporaneamente il numero di prede cala. In seguito il numero di predatori torna a calare e le prede aumentano in numero. Questo pattern si ripete in modo abbastanza sistematico per tutta la durata delle osservazioni.

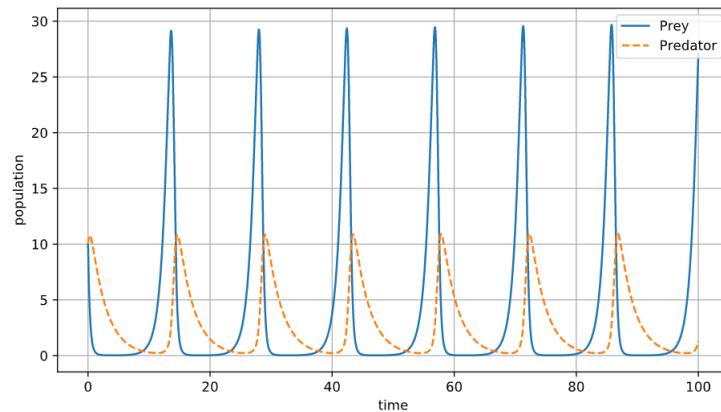
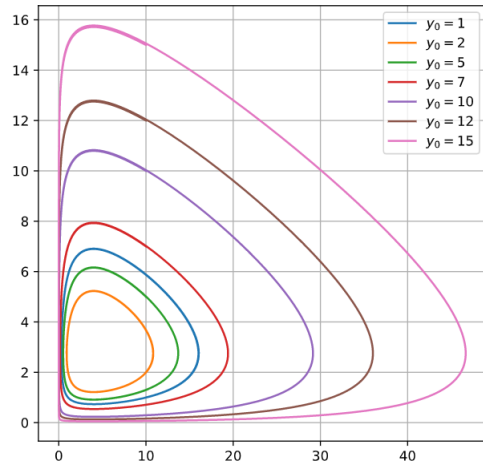


Figura 34: Grafico che mostra la dinamica delle popolazioni preda e predatore simulata mediante il modello Lotka-Volterra

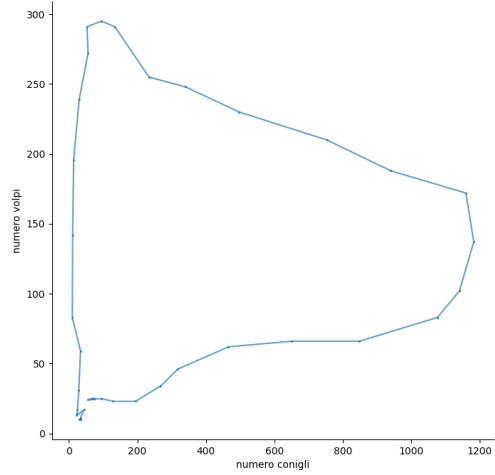
La **figura 34** mostra la dinamica delle popolazioni di specie di predatori e di una di prede che seguono le equazioni di Lotka-Volterra, dati i seguenti parametri:

- numero di prede all'inizio della simulazione: 10;
- numero di predatori all'inizio della simulazione: 10;
- tasso di crescita delle prede: 1.1;
- tasso di morte delle prede: 0.4;
- tasso di crescita dei predatori: 0.1;
- tasso di morte dei predatori: 0.4

Confrontando i **grafici 21a** e **21b** si può notare che l'andamento delle popolazioni del modello da validare e di quelle del modello di Lotka-Volterra sono del tutto speculari.



(a) Grafico Lotka-Volterra



(b) Grafico modello reinserimento

Figura 35: Confronto tra il phase space plot del modello Lotka-Volterra in condizioni diverse in base al numero iniziale di predatori e il grafico popolazione conigli vs popolazione volpi del modello in configurazione con reintroduzione

La **figura 35a** mostra sull'asse delle ascisse il numero di prede e sull'asse delle ordinate il numero di predatori per il modello di Lotka-Volterra in diverse condizioni in cui varia il numero di predatori che popolano inizialmente il sistema. Invece, la **figura 35b** riproduce lo stesso grafico, realizzato considerando i dati delle simulazioni condotte sul modello implementato. Confrontando i due grafici, è possibile notare che l'andamento è tendenzialmente il medesimo, pur essendoci delle piccole differenze.

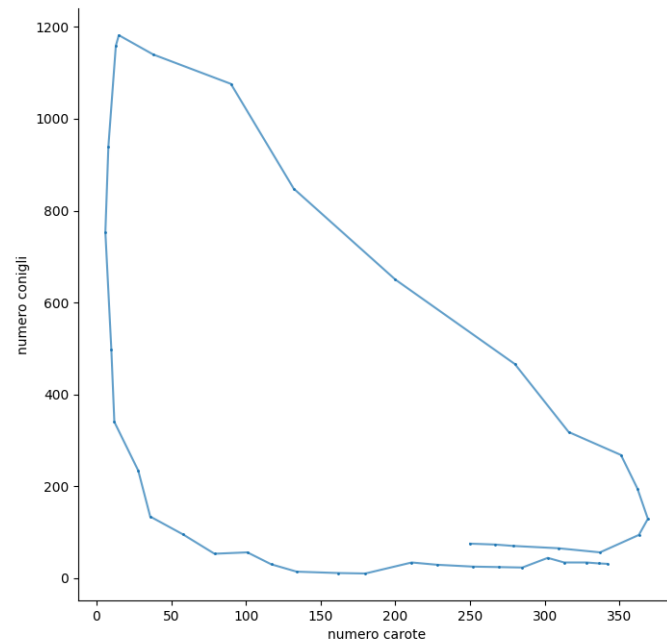


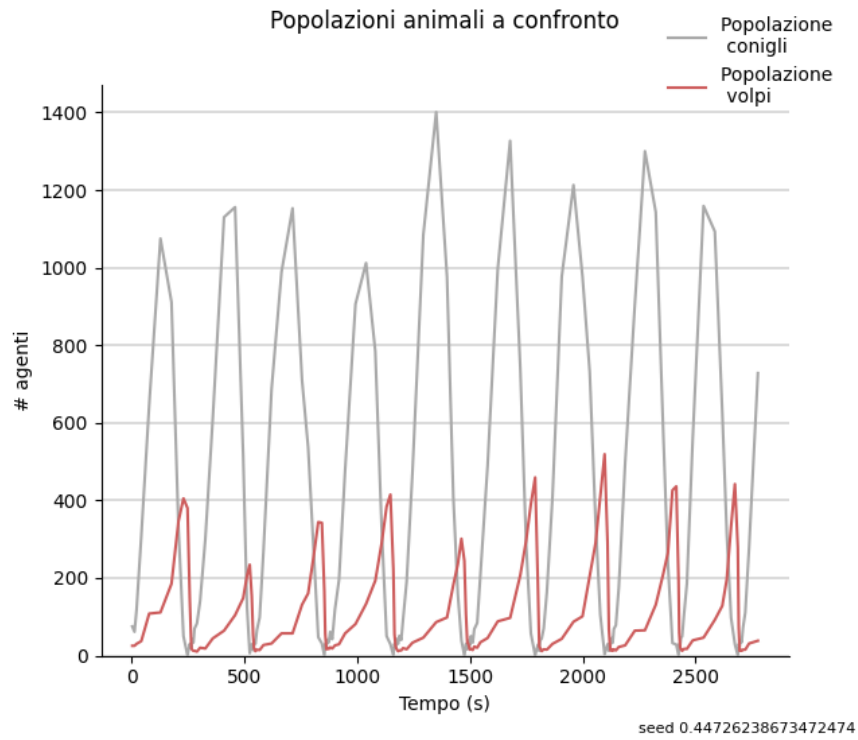
Figura 36: Grafico che mostra il confronto tra il numero di conigli e il numero di carote del modello realizzato

Lo stesso grafico è stato realizzato prendendo in considerazione il numero di carote invece che il numero di volpi, in quanto nel modello sviluppato possiamo considerare due coppie di popolazione preda-predatore: le volpi e i conigli da una parte e i conigli e le carote dall'altra. Il risultato è visibile in **figura 36**. In questo caso si è ottenuto un risultato ancora più conforme rispetto ai risultati del modello Lotka-Volterra mostrati in **figura 35a**.

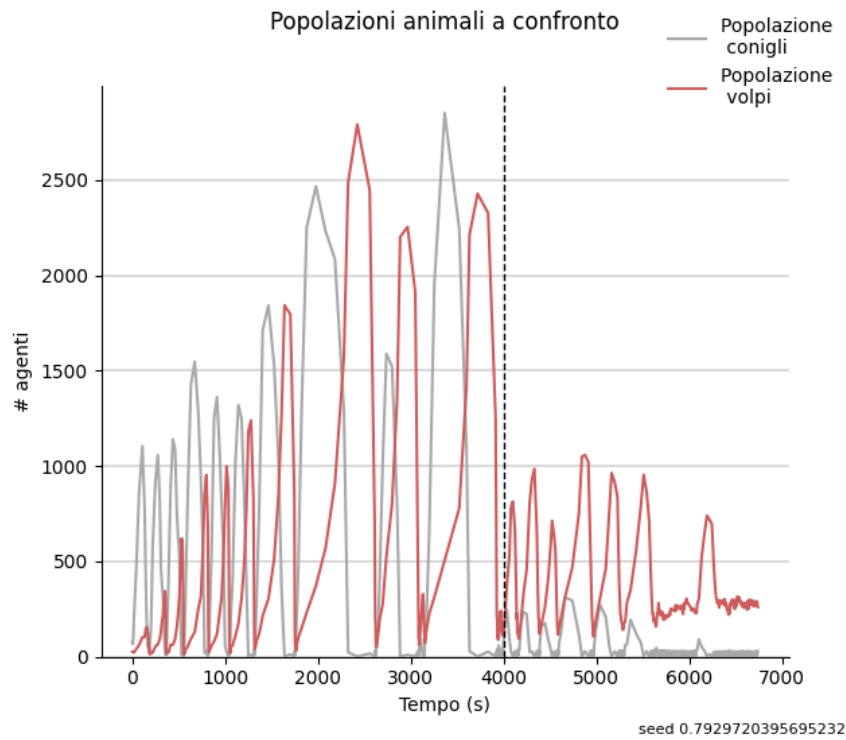
## 7.4 Confronto tra configurazioni

Una volta completata la fase di raccolta ed analisi dei dati si è ritenuto opportuno soffermarsi sulle differenze tra le due configurazioni principali, ovvero quella senza e quella con evoluzione, per studiarne il significato. Si può notare come nei **grafici 37a e 37b** l'andamento delle popolazioni tra le due specie animali all'inizio risulti essere sovrapponibile, seppur nella configurazione con evoluzione si verifica un aumento considerevole della quantità di agenti che popolano il sistema. Le differenze sostanziali nell'andamento delle popolazioni degli animali si presentano nel lungo periodo, infatti nel primo caso il sistema risulta omogeneo, senza cambiamenti nel trend, invece nel secondo vi è un cambiamento radicale che va a stravolgere il modello originale di Lotka-Volterra. Questa divergenza tra le configurazioni è dovuta al fatto che nel primo modello le caratteristiche degli animali rimangono immutate nel tempo e quindi non si verifica mai un cambiamento repentino nell'andamento delle due popolazioni. E' ipotizzabile che eseguendo la simulazione per periodi di tempo molto lunghi il trend si ripeterà ciclicamente in modo del tutto simile per ogni ciclo fino a quando non si decide di arrestare la simulazione. Nel secondo modello invece, l'andamento delle popolazioni si può dividere in due fasi. Inizialmente esso presenta delle fluttuazioni piuttosto evidenti, che portano al crescere sia del numero di conigli che del numero di volpi. Successivamente, nella seconda fase, grazie al fatto che l'evoluzione ha portato le volpi ad ottenere un vantaggio sostanziale sui conigli, si verifica un nuovo equilibrio, anche se estremamente diverso da quello ottenuto nel primo modello, con le volpi che prendono il dominio nei confronti dei conigli. Infatti, in questa seconda fase, a differenza della prima, il numero delle volpi sovrasta costantemente quello dei conigli.

In estrema sintesi, è possibile affermare che l'andamento delle popolazioni delle volpi e dei conigli nella prima configurazione segue quasi pedissequamente lo sviluppo descritto dal modello Lotka-Volterra. Viceversa, nella seconda configurazione l'andamento risulta essere leggermente differente e inizialmente vede il crescere delle due popolazioni fino a quando l'evoluzione favorisce le volpi che riescono a prendere il controllo dell'ambiente, discostandosi dall'andamento originale e creando un nuovo equilibrio.



(a) Configurazione senza evoluzione



(b) Configurazione con evoluzione

Figura 37: Confronto andamento popolazioni animali delle due configurazioni principali



## 8 Conclusioni e sviluppi futuri

I risultati ottenuti rispondono ai requisiti posti all'inizio del progetto. Come ampiamente descritto in precedenza, sono state prodotte tre configurazioni differenti del modello seguendo un approccio incrementale. Infatti, per prima cosa, si è partiti con una configurazione semplice, ottenendo dei risultati poco soddisfacenti. Per questo motivo, si è deciso di estendere il modello introducendo la dinamica della reintroduzione. Questo ha permesso di ottenere risultati decisamente più realistici. Si è deciso quindi di eseguire una validazione di questa configurazione per verificare la bontà del modello implementato. Infine, si è deciso di sperimentare un'ulteriore estensione, provando ad introdurre la dinamica dell'evoluzione genetica negli individui delle due popolazioni. Quest'ultima operazione ha permesso di ottenere risultati nuovi in questo ambito. Infatti in letteratura non sono presenti molti studi relativi alle dinamiche preda-predatore che tengono in considerazione anche della possibilità che le caratteristiche degli agenti cambino nel tempo. Inoltre, l'approccio multi-agente si è rivelato particolarmente adatto a modellare un sistema di questo tipo a causa del fatto che ogni agente possiede caratteristiche individuali, uniche.

Le principali criticità riscontrate riguardano la complessità computazionale del modello, in tutte e tre le configurazioni, che aumenta al crescere degli agenti presenti nel sistema. Infatti, quando il numero di individui che popolano l'ambiente diventa particolarmente elevato la simulazione viene rallentata a causa del numero di valutazioni e azioni che ciascun agente deve effettuare. Una possibile soluzione a questa problematica è quella di sfruttare tecnologie di sviluppo più complesse come accelerazione grafica tramite GPU e programmazione multi-thread. Un altro aspetto negativo è che non è stato possibile compiere una validazione della configurazione con evoluzione, a causa della scarsità di dati derivanti da osservazioni sperimentali presenti in letteratura relativi a questa tematica.

In conclusione, ulteriori sviluppi futuri possono essere i seguenti:

- Sviluppo più approfondito dell'evoluzione genetica, introducendo i concetti di fenotipo e genotipo;
- Studio delle dinamiche di gruppo, attraverso le quali i conigli possono sfuggire ai predatori e le volpi possono collaborare per cacciare in branco;
- Rendere il modello ancora più realistico, introducendo nuove dinamiche ambientali, per esempio: creazione di tane dove gli animali possono rifugiarsi, temperatura ambientale variabile, aggiunta di nuove specie di prede e predatori, oltre che di malattie;
- introduzione degli effetti del comportamento dell'uomo sull'ecosistema.

## Riferimenti bibliografici

- [1] “What is a Natural Ecosystem?” In: *VEDANTU* (lug. 2020). URL: <https://www.vedantu.com/biology/what-is-a-natural-ecosystem>.
- [2] *natural ecosystem*. Feb. 2017. URL: <https://www.eea.europa.eu/help/glossary/eea-glossary/natural-ecosystem>.
- [3] Contributors to Wikimedia projects. *Ecosystem - Wikipedia*. Mag. 2021. URL: <https://en.wikipedia.org/w/index.php?title=Ecosystem&oldid=1024959291>.
- [4] Tee L. Guidotti. “Artificial Ecosystems”. In: *Health and Sustainability*. Oxford, England, UK: Oxford University Press. ISBN: 978-0-19023880-3. DOI: 10.1093/acprof:oso/9780199325337.003.0009.
- [5] Contributor ai progetti Wikimedia. *Vulpes vulpes - Wikipedia*. Mag. 2021. URL: [https://it.wikipedia.org/w/index.php?title=Vulpes\\_vulpes&oldid=120916332](https://it.wikipedia.org/w/index.php?title=Vulpes_vulpes&oldid=120916332).
- [6] *Red Fox Senses | Wildlife Online*. Giu. 2021. URL: <https://www.wildlifeonline.me.uk/animals/article/red-fox-senses>.
- [7] Contributor ai progetti Wikimedia. *Coniglio - Wikipedia*. Giu. 2021. URL: <https://it.wikipedia.org/w/index.php?title=Coniglio&oldid=121136936>.
- [8] Giorgio Gattoni. “La riproduzione dei conigli”. In: *Animalpedia* (apr. 2019). URL: <https://www.animalpedia.it/la-riproduzione-dei-conigli-2545.html>.
- [9] Charles Darwin. *On the Origin of Species by Means of Natural Selection*. Murray, 1859.
- [10] Narendra S. Goel, Samareesh C. Maitra e Elliott W. Montroll. “On the Volterra and Other Nonlinear Models of Interacting Populations”. In: *Rev. Mod. Phys.* (apr. 1971). ISSN: 1539-0756. DOI: 10.1103/RevModPhys.43.231.
- [11] Alfred J. Lotka. “Analytical Note on Certain Rhythmic Relations in Organic Systems”. In: *Proc. Natl. Acad. Sci. U.S.A.* (lug. 1920). DOI: 10.1073/pnas.6.7.410.
- [12] *Elements of Physical Biology*. Vol. 116. Nature Publishing Group, set. 1925. DOI: 10.1038/116461b0.
- [13] Nicolas Bacaër. “Lotka, Volterra and the predator–prey system (1920–1926)”. In: *A Short History of Mathematical Population Dynamics*. London, England, UK: Springer, 2011. ISBN: 978-0-85729-114-1. DOI: 10.1007/978-0-85729-115-8\_13.
- [14] Simone Artioli. “Dinamica delle popolazioni: modelli deterministici di Lotka-Volterra”. Tesi di dott. 2020.
- [15] Contributor ai progetti Wikimedia. *Equazioni di Lotka-Volterra - Wikipedia*. Apr. 2021. URL: [https://it.wikipedia.org/w/index.php?title=Equazioni\\_di\\_Lotka-Volterra&oldid=119944996](https://it.wikipedia.org/w/index.php?title=Equazioni_di_Lotka-Volterra&oldid=119944996).

- [16] Physics of Risk. *Agent-based prey-predator model – Physics of Risk*. URL: <https://rf.mokslasplius.lt/agent-based-prey-predator-model>.
- [17] *Netlogo*. 1999. URL: <https://ccl.northwestern.edu/netlogo>.
- [18] Wilensky Uri e William Rand. “An introduction to agent-based modeling modeling natural, social, and engineered complex systems with NetLogo: a review”. In: *Complex Adaptive Systems Modeling* (2016).
- [19] PyGame Development team. *About - pygame wiki*. Giu. 2021. URL: <https://www.pygame.org/wiki/about>.
- [20] Nov. 2003. URL: <http://people.whitman.edu/~hundredr/courses/M250F03/LynxHare.txt>.
- [21] *Predator-Prey Interaction*. Apr. 2020. URL: [https://www2.nau.edu/lrm22/lessons/predator\\_pre/predator\\_pre.html](https://www2.nau.edu/lrm22/lessons/predator_pre/predator_pre.html).