Simulation of Collective Intelligence of a Multi-Species Artificial Ecosystem Based on Energy Flow

Aliakbar Asgari

Thesis submitted to the

Faculty of Graduate and Postdoctoral Studies

In partial fulfillment of the requirements for the degree

Master of Systems Science

School of Electrical Engineering and Computer Science

University of Ottawa

Ottawa, Ontario, Canada

September 2014

Abstract

Collective intelligence (CI) emerges from local coordination, collaboration and competition among the individuals within a social group. CI mainly results in a global intelligent behavior.

One of the fundamental interactional channels within a CI system is energy flow. Each agent within an artificial or physical ecosystem must absorb energy in order to survive, evolve, breed, and reshape its local environment. In addition because the energy resources are limited in the environment, each agent has to compete with other agents to reach the required level of energy. Understanding the internal energy flow can potentially provide a deep insight into internal activities and external emergent behaviors of a given complex system.

This study proposes a stochastic scheme for modeling a multi-species prey-predator artificial ecosystem with two levels of food chain. This will enable us to investigate the influence of energy flow on the ecosystem's lifetime. The proposed model consists of a stationary hosting environment with dynamic weather condition and fruit trees. The inhabitants of this ecosystem are herbivore and carnivore birds each consisting of species. In our model, the collective behavior emerges in terms of flocking with more added rules consist of breeding, competing, resting, hunting, escaping, seeking and foraging behaviors. Using multi-species scheme, we define the ecosystem as a combination of prey and predator species with inter-competition among species within same level of food chain and intra-competition among those belonging to different levels of food chain. Furthermore, in order to model the energy within the ecosystem, some energy variables as functions of behaviors are incorporated in to the model.

Finally, a simulation and visualization structure for implementing the proposed model is developed in this study. The experimental results of 11,000 simulations analyzed by Cox univariate analysis and hazard function suggest that only five out of eight behaviors can statistically significant influence the ecosystem's lifetime. Furthermore, the results of survival analysis show that out of all possible interactions among energy factors, only two of them, interaction between flocking and seeking energies, and interaction between flocking and hunting energies, have statistically significant impact on the system's lifetime. In addition, software implementation of the proposed framework validates the stability of simulation and visualization architecture. At last regression results using Nelson-Aalen cumulative hazard function and Cox-Snell variable and scaled Schoenfeld residuals test strongly validate our experimental results.

To the best of our knowledge, there are three contributions in this research:

First, the high level of complexity in the structure of the proposed model in comparison with the other systems which mostly contains only one species of prey, one species of predator and a kind of resource. While this study introduces two species of prey, capability of competition among species, dynamic weather condition with two element of wind and rain and dynamic resources, various behavioral rules such as escaping, breeding, hunting, resting, etc.

Energy flow analysis within an artificial ecosystem is the second contribution. To the best of author's knowledge there is no similar comprehensive model in the previous literature that investigates the life span of a stochastic multi-species predator-prey artificial ecosystem based on energy flow using Survival Analysis method.

Lastly, the simulation results show that the flocking and seeking energy and flocking and hunting energy interactions are the most significant interactions which match with the Thompson et al. [65] observations in the real life. Their findings indicate that in the real life, birds use flocking behavior for better movement, more efficient food searching and social learning. Flocking motion also decrease predation risk as much as the flock size increases.

Acknowledgments

First and foremost, I would like to express my deepest appreciation to my supervisor, Prof. Won-Sook Lee, for her patience and constant support during my thesis research. Her critical comments and insightful feedbacks were essential for any steps toward progress of my research.

My thanks also goes to Richard for participating in first part of the project, and my colleague at CG++ Research Laboratory, Kaveh, who has greatly helped me to complete my master studies through his valuable collaboration and guidance.

Last by not least, I thank almighty my darling Soodeh for her constant encouragement, without which this research would not be possible.

Dedication

To my

Mother and Father

For their endless love

Table of Contents

List of Figures

List of Tables

Abbreviations

ABMS Agent-based Modeling and Simulation

CI Collective Intelligence

AL Artificial Life

AI Artificial Intelligence

PSO Particle Swarm Optimization

GUI Graphical User Interface

Chapter 1. Introduction

1.1 Objectives

Complex system refers to a system that includes a large number of interacting, self-organizing and inter-linked components with decentralized control scheme that follows certain rules [3]. From a mathematical point of view, complex system is defined as a system whose total energy exceeds the threshold to operate based on classical mechanics but does not reach the threshold to exhibit chaotic properties. The most fundamental property of a complex system is emergence which is the outcome of collective behavior of interacting components. However, this property cannot be identified independently in each component`s behavior due to dynamics and nonlinearity of the system. [45,46] Biological complex systems refer to social groups of living organisms with collective behaviors whose emergent property results in collective intelligence (CI).

According to Levy and Bonomo [38], CI is defined as "a form of universally distributed intelligence, constantly enhanced, coordinated in real-time, and resulting in the effective mobilization of skills. The basis and goal of collective intelligence is mutual recognition and enrichment of individuals rather than the cult of fetishized or hypostatized communities." Another definition coined by center for CI in MIT expresses CI as a "Group of individuals doing things collectively that seems intelligent". [57] Essentially, CI emerges from local coordination and collaboration, or even competition among the individuals within a social group, and results in a global intelligent behavior. From a mathematical point of view, CI is a decentralized emergent property of a complex system with sufficient information sharing capabilities among its individuals that arises from some simple rules. Schut [58] mentioned adaptivity, emergence, global-local behavior, rules, interactions, redundancy, robustness, and randomness as characteristics and mechanisms that discriminate a CI system from others.

CI is inspired by evolution of biological societies such as bacteria groups, fish schools, bird flocks, ant colonies, and some human artifacts, and has been utilized in a vast variety of real world applications such as telecommunication networks and robotic [53]. Generally, CI systems are categorized into natural systems and artificial computational systems [15]. Natural CI models are investigated in biology [34, 13] and physics [24, 8] to address the emergence of self-organized collective behaviors in nature. An example of natural CI is flocking birds. According to Parrish et al., [50], flocking is not just an aggregated and choreographed motion, but the dynamics of its density and structure is essentially a defensive mechanism for confusing the predators and surviving the predation. On the other hand, artificial CI systems are bio-inspired computational models that are widely utilized in computational intelligence approaches such as particle swarm optimization (PSO), artificial bee colony (ABC), and ant colony optimization (ACO) artificial fish swarm algorithm (AFSA), and multi-agent systems such as self-organizing mobile multi-robot systems. In addition, CI is exploited in computer animation. Reynolds [54] introduced Boids (i.e. artificial bird imitating agents) to visualize the flocking behavior of birds using three simple rules including cohesion, alignment and separation.

The importance of complex system in physics, information theory and biology necessitates developing proper tools for modeling them. For this purpose, bottom-up design patterns are more suitable than top-down and analytical-reductionist approaches [56]. According to Gilbert and Terna [21], traditional modeling tools are not suitable for heterogeneous complex system with nonlinear and discontinued interactions among its components. Agent-based simulation of CI systems regards each individual within the ecosystem as an agent which in turn. Agent-based simulation of CI improves understanding of the dynamic interactions within aggregated phenomena occurring in the ecosystem [61]. Emergent property in agent-based model (ABM) can be originated in agent properties, environmental influence, inter-agent interactions and evolutionary processes in both agent and environment side. ABM can be classified into five categories. In first category, agents with stationary behavioral rules interact within a static environment, whereas in second category environment plays an active role and influences the agents. In third class, agents evolve through time within a static environment. In fourth category environment plays an active role in the evolution of agents. Ultimately, in fifth category, the agents and the environment evolve concurrently [33]. Rodriguez and Reggia [55] compared the performance of CI-based agents against independent agents in terms of their degree of adaptivity regarding the environmental changes and concluded that former group outperforms the independent agents.

1.2 Methodology

One of the fundamental interactional channels within a CI system is energy flow. We define energy flow as the process of generating, transforming and consuming energy which plays a critical role in the stability of the system at hand. Each agent within an artificial or physical ecosystem must absorb energy in order to survive, evolve, breed, and reshape its local environment. Furthermore, due to limited energy resources in the environment, each agent has to compete with other agents to gain the required energy. Optimizing the energy flow within the given system yields to more stability and enhances the life time of the ecosystem. Understanding internal energy flow provides a deep insight into internal activities and external emergent behaviors of a given complex system. [1]

One of the fundamental interactional channels within a CI system is energy flow which individuals needs to survive, evolve, breed, and reshape its local environment. Due to limited energy resources, each agent has to compete with other agents to gain the required energy, Understanding internal this internal energy flow provides a deep insight into internal activities and external emergent behaviors of a given complex system and will make a platform for analysing any similar systems.

This study develops a complex stochastic model to simulate a multi-species predator-prey ecosystem with proper degree of complexity that can provide sufficient information regarding energy flow. Software implementation for the proposed model validates stability of simulation and visualization architecture.

Analyzing the experimental results using Survival Analysis indicates corresponding energy variables and interactions significantly influencing the ecosystem's lifetime.

1.3 Contributions

To the best of our knowledge, there are three contributions in this research:

First, the high level of complexity in the structure of proposed model in comparison with the other systems which mostly contains only one species of prey, one species of predator and a kind of resource. While this study introduces two species of prey, capability of competition among species, dynamic weather condition with two element of wind and rain and dynamic resources, various behavioral rules such as escaping, breeding, hunting, resting, etc.

Energy flow analysis within an artificial ecosystem is second contribution. To the best of author's knowledge there is no similar comprehensive model in the previous literature that investigates the life span of a stochastic multi-species predator-prey artificial ecosystem based on energy flow using Survival Analysis method.

Third, our simulation results show the flocking and seeking energy and flocking and hunting energy interactions are the most significant interactions which matches the Thompson et al. 65] observations in the real life. Their findings indicates that in the real life birds use flocking behavior for better movement, more efficient food searching and social learning. Flocking motion also decrease predation risk as much as the flock size increases.

1.4 Overview of the Thesis

The rest of this thesis is organized in the following chapters:

- *Chapter 2* presents a comprehensive survey on related works. Moreover, this chapter investigates collective intelligence in complex systems, its properties and mechanisms. Agent-based modeling is explained in details. It explores the existing toolkits in agent based-modeling and simulation. Finally this chapter discusses Survival Analysis method in details.

- *Chapter 3* describes the methodology. This chapter presents our proposed model for simulating a multi-species predator-prey artificial ecosystem.

- *Chapter 4* presents experimental results. This chapter highlights the dynamics of a stable model with different sets of energy parameters and their impact on the system's lifetime.

- *Chapter 5* is the conclusion and discussion. This chapter outlines the potential and future path for this research.

Chapter 2. Literature Survey

2.1 An Overview

An agent-based artificial ecosystem is inhabited by abstract or embodied agents that interact autonomously within a dynamic environment. This section introduces a comprehensive overview of related studies. Later in section 2.2, 2.3, 2.4 and 2.5 and 2.7 main core aspects of this thesis which are complex systems and collective intelligence, CI simulation using agent-based modelling and predator-prey ecosystem modelling, and Survival Analysis are described in details. Moreover, section 2.6 investigates literature in terms of visualization in existing agent-based toolkits.

Heleno and Próspero dos Santos [26] simulated an early agent-based artificial ecosystem with virtual animals using a set of internal states including hunger, health and energy, and set of prioritized behavioral rules. Nishimura and Ikegami [47] studied the emergence of collective strategies in a predator-prey system using a rewarding system. In this system, species gain rewards by feeding and hunting and those species that have gathered more rewards have more chance to reproduce. The simulation results suggest that creating swarms allows both preys and predators to maximize their lifetime. Werner [70] investigated the collective behavior in artificial animals regarding predation risk, breeding efficiency, and optimality of food search. Probably, one the simplest and yet most studied models of artificial ecosystem is Reynolds' [54] Boids (i.e. artificial bird imitating agents) which was originally proposed to visualize the flocking birds for graphical applications. In this model, natural and choreographed motion of birds is simulated using three simple rules including separation, alignment, and cohesion. Reynolds' Boids model has been extended in a few studies. For example, Vicsek et al., [66] introduced phase transitions into Boids, Olfati-Saber [49] proposed a distributed version of Reynolds' Boids, Hartman and Benes [25] added a model of external force into alignment behavior to define leadership, Silva et al., [60] expanded the original model by embedding a self-occlusion model, Zhou et al., [73] introduced pseudo-leadership mechanism into flocking behavior, Ben-Shahar et al., [5] expanded Reynolds' model by defining direction election capability, and ultimately Tanner et al.,[63] studied the effects of topology on stable flocking. A comprehensive overview on MAS simulation of ecosystems can be found in Bousquet and Page [7].

As Thompson et al. [65] indicate, in reality of life, birds use flocking behavior for better movement, more efficient food searching and social learning. Flocking motion also decrease predation risk as much as the flock size increases and our simulation results show the flocking and seeking energy and flocking and hunting energy interaction are most significant interactions which matches the Thompson et al. observations in real life.

Furthermore, some research works have studied evolutionary modeling of artificial ecosystems. Husbands et al., [27] proposed artificial evolution that utilizes evolutionary paradigms to evolve the agents so that they can exhibit emergent behavior. Another evolutionary artificial ecology is introduced by Ward et al., [68]. In their proposed system, an initial amount of energy is assigned to each individual who is allowed to spend on a set of behaviors. Through the evolution and within generations, individuals learn how to optimize their energy consumption while increasing their survival chances. The results showed that preys learn to construct flocks and predators learn to seek for preys. Gras et al., [23] exploited a fuzzy cognitive map (FCM) as a functional approach to track and analyze agents' behaviors such as keeping distance from predators and moving toward resources, and their internal states such as hunger and fear. They suggested that diversity of emergent patterns within species is explicitly associated with environmental information and internal states. Other studies on artificial evolution in virtual ecosystems can be found in [16, 42, 71].

Weisberg and Reisman [69] conducted sensitivity analysis and structural robustness analysis to study the robustness in populations of dynamic predator-prey ecosystems. They concluded that both approaches can lead to precise models. Golestani and Gras [22] analyzed the long term behavior of a CI ecosystem using MAS simulations. This study evaluates the signals and patterns produced by ecosystem to estimate the amount of chaotic behaviors based on Lyapunov exponent. Elsadany et al., [20] investigated the influence of parameters such as prey growth rate and predator termination rate on the local stability and chaotic behaviors. Cooper and Ofria [12] studied the effect of competition among agents over resources in terms of stability and emergent behavior.

A few studies have addresses the energy flow within predator-prey artificial ecosystems. Kernbach and Kernbach [32] utilized nonlinear dynamic models to enhance the energy consumption within a swarm of micro-robots performing collective behaviors. Scogings and Hawick [59] studied the effects of incorporating energy conservation in a predator-prey agent-based model. They investigated the influence of reproducing offspring on the swarm by assuming that breeding process results in health risks and energy consumption. Counterintuitively, they suggested that omitting the assumption of health risk in breeding can lead to extinction of species. They concluded that implicit energy regulations such as health risk are equivalent to explicit control parameters such as maximum allowed population size. Dorin et al., [19] developed a model for agent-based artificial ecosystems that exploits a model of energy transition adopted from ecological data. They suggest that studying energy flow within a virtual ecosystem and comparing it with natural ecological data can enhance the modeling process. Polyworld pioneered by Yaeger [72], was an artificial ecosystem in which parents transfer their energies to newborns and preys transfer their stored energy to their predators. Ultimately, Dorin and Korb [18] studied the effects of energy flow on emergent properties of ecosystems and showed that energy flow has a deep impact on how collective behavior emerges.

As far as author' knowledge is considered, there is no comprehensive model in literature that investigates the life span of a stochastic multi-species predator-prey artificial ecosystem based on energy flow analysis.

2.2 Collective Intelligence in Complex Systems

After a quick overview of relevant works, this section surveys literature related to complex systems and collective intelligence. In fact, simulating a multi-species predator-prey artificial ecosystem needs a deep understanding of complex systems, emergent behavior within complex systems and collective intelligence. Complex systems are formed by interlinked components that follow certain rules with self-organizing and collective intelligence. Within complex systems, there are dynamic and aggregated behaviors which can be described by a well-known paradigm of "The whole is more than just sum of the parts". [4]

Moreover, a deep uncertainty associated with complex systems makes them unpredictable with emergent properties. [3] According to Gilbert and Terna [21], the traditional modeling tools do not work properly when there is nonlinearity and discontinued interaction within a complex system. Therefore there is an essential need to better understand mechanism of complex system.

According to Dorigo et al., [15] collective intelligence emerges from local collaboration of agents which produces a global intelligent behavior. From a mathematical point of view, CI motion is decentralized emergent behavior that originates from some simple rules. As Pitcher [52] argued, from a biological point of view CI emerges as birds shape in highly cohesive aggregation by continuously changing density and structure that help them to survive from predation. Kunz and Hemelrijk [34] and Couzin et al., [13] in biology as well as Gregoire and Chate [24] and Buhl et al., [8] in physics are some examples of previous works that try to explain how self-organized collective behavior emerges, but CI behavior was first modeled by Reynolds [54].

As mentioned in section 2.1, Reynolds [54] introduced a first computer graphical modelling of a CI complex system, named Boids that follow three simple rules to mimic flocking behavior of birds; separation, alignment and cohesion. CI also has been applied to complex patterns of self-organized robots.

Schut [58] listed some properties and mechanisms that help us to distinguish collective intelligence complex systems from other systems. These properties are adaptivity, emergence, global-local behavior, rules, interactions, redundancy, robustness and randomness. Following each property is described briefly according to Schut [58]:

Adaptivity: Adaptivity is having flexibility to change structure of a system based on environment's needs. This change could be either micro such as change in a system particle behavior or macro such as change in the entire system. Sometimes a change in a whole system might happen without any change in individual's behavior.

Emergence: In a complex system, emergence is where patterns dynamically arise from a multiplicity of simple interactions.

Global-Local: The local level of a system deals with individuals while global level considers system as a whole.

Rules: This key property is used to describe the behavior of an individual or the entire system from input to output.

Interaction: Individuals within an agent-based system interact with each other, which could be a kind of communication. As mentioned by Vicsek et al., [66] individual's behavior within a complex system could not be analyzed without considering the effect of interactions as a key property.

Redundancy: This property stands for sharing information among individuals using the same rules that all individuals work with.

Robustness: Redundancy helps a system to become robust and maintain itself in case of any perturbation or damage. In case of any loss, there are chances to have information and knowledge somewhere in the system. When the system is in balance only dramatic disturbances can change its state while any little disorder has no effect on system's functionality. A system can response to any action depends on the impact of that action on its performance.

Randomness: Some element of randomness leads to a self-organized behavior which is one of the properties of agent-based complex systems.

2.2.1 Modelling a Collective Intelligence Complex System

The core structure of this thesis is to model a multi-species predator-prey artificial ecosystem. Schut [58] proposed four different recipes to build a CI complex system. These recipes are mainly based on a similar basic platform which has been developed on different aspects.

Basic scenario: A basic scenario contains six steps as follows: First, an action should be defined for all agents. Second, all available information for agents is determined using a set of observations such as position *(x, y)* for coordinating *x* and *y*. The third step is defining a set of functions which models the environment of a given system. In this step all possible actions for each agent are determined in every time steps. These functions also define the state of each action such as fail or succeed. Fourth, the cost of performing a function in the model is assigned to an agent. For example, the cost of breeding for agent can be considered as a portion of energy that this agent should use for offspring.

The fifth step is to categorize the benefit of doing a successful action. The sixth step defines behaviors of the individuals. Likewise an if-then rules or decision tree model, a basic scenario of artificial ecosystem outlines how observations can come in and when each action should take place. In the simple basic recipe agents are not allowed to have internal states. Therefore, the only way to assign a behavior to an individual is the action assigned for observations.

Internal recipe: The second recipe is extended model of the simple recipe. However internal recipe allows individuals to have different behaviors. In this scenario, agents can have their own internal state such as having a memory for their different role descriptions. These internal states behave like intermediate variables between observations and actions.

Diversity recipe: In this scenario a set of different actions, observations and policy functions are assigned to the agents. In spite of basic recipe actions, observations and action’s functions for each individual are not defined similarly for all agents. It is worthwhile to mention that each agent should have its own set of functions.

Non-determined recipe: Real world phenomena are not usually certain, therefore modeling the real world needs to model uncertainty. This recipe allows including uncertainty in the model. Non-determined recipe assign probabilities to each individual's functions. It means that in the same situation agents do not always assume to have same actions.

There are some models and algorithms available to design a CI complex system without reinventing the wheel. According to Schut [58] collective intelligence models and algorithms can be categorized in Table 1 and Table 2. These methods can be used to design the desired models and then can be transformed to a computer model based on available software platforms for further simulation and experimentation. Next section surveys studies related to CI complex system's simulation platforms.

Table 1: Algorithms that can be used for the specific CI complex system (Source [58]).

Algorithm Name	**Description**
Evolutionary Methods	An evolutionary method (or algorithm) is ‘‘a generic population-based metaheuristic optimization algorithm”. The heuristics that it uses are inspired by biological evolution: reproduction, mutation, recombination, survival of the fittest, and natural selection.
Co-evolution	If a natural environment does not give any information about the quality of individuals, then evolution is impossible. However, if the environment is informative, then we can define fitness on the basis of coevolution. A typical setup of a basic co-evolutionary algorithm then consists of two populations with first and second player strategies, respectively.
Learning Classifier Systems (LCS)	LCS exploits the combination of evolutionary computation and reinforcement learning to develop sets of condition-action rules. They exploit ‘‘possibilities of general-purpose learning algorithms applied to parallel rule-based systems”.
Neuro-Evolution (NE)	NE is a technique that evolves artificial neural networks with evolutionary methods. The evolutionary operators of NE work on the weight space of the network: recombination combines weight vectors of complete networks; mutation works on single networks’ weights.
COllective INtelligence (COIN)	A COIN is a multi-agent system where: (1) the agents each run reinforcement learning (RL) algorithms; (2) there is little to no centralized communication or control; and (3) there is provided a world utility function that rates the possible histories of the system.
Particle Swarm Optimization (PSO)	In a PSO algorithm, particles form a swarm that flies through a (multi-dimensional) problem space, whereby the positions of the particles are updated each generation based on the particle’s experience and that of its neighbors.

Table 2: Models that can be used for the specific CI complex system (Source [58])

Model Name	**Description**
Cellular automata	A CA is ‘‘an array of identically programmed automata, or cells which interact with one another in a neighborhood and have definite state”. They can be complex systems themselves, and offer good ways of studying the behavior of such systems.
Multi-Agent Systems	The MAS paradigm involves the investigation and design of situated agents that exhibit autonomous and rational behavior with respect to their environment and to each other.
Boolean Networks	The basis of a BN is a set of logical propositions, which can be evaluated true or false and are combined by logical operators (OR, AND, XOR, etcetera) in some networked fashion.
NK-Model	A Random Boolean Network (RBN) is a Boolean network of which the configuration (defines how the propositions are connected to each other) is unknown. Furthermore, the gate types and their connections are randomly initialized. Name of this model (NK) comes from the number of gates denoted by N and the number of inputs denoted by K,
Particle-Based Models	Vicsek et al., (1995) have successfully modeled human panic behavior in case of a crowd stampede based on a physical force model for many-particle systems.
Game and Decision Theory	A game involves all those situations where ‘‘a number of players or decision makers interact, possibly threaten each other and form coalitions, take actions under uncertain conditions, and finally receive some benefit or reward or possibly some punishment or monetary loss”. The study of such games is the area of game theory.
Formal Logics	Many AI researchers have used formal logical models, e.g. agent societies have been formalized using logic; agent based social simulation has been compared with formal systems, and human organizations have been formally modeled.
Knowledge Systems	A knowledge system is a computer system that represents and uses knowledge to accomplish a task. The Physical Symbol System Hypothesis says that ‘‘physical symbol systems are necessary and sufficient for general intelligent action”. This hypothesis, although debatable, justifies the interest in knowledge systems.

2.3 Simulation of CI in Complex Systems using Agent-based Modeling

This section explains how previous studies built a model for a collective intelligence complex system. Agent-based modeling and simulation is a method to model and simulate a CI complex system. This section first presents the concepts of Agent-Based Modeling and Simulation (ABMS) in details. Second, existing toolkits for ABMS are introduced briefly. Moreover, I compare the ABMS with classic population-level models briefly and explain why population-based models are not able to describe the reality of collective behavior phenomena by investigating previous studies.

Agent-Based Modeling and Simulation (ABMS) which is also known as an Agent-based Modeling (ABM), Individual-Based Modeling (IBM) and Agent-Based Systems (ABS), is collaboration of distributed artificial intelligence and computational simulation. [56] ABMS gives an infrastructure for modeling and analyzing any kind of process related to agents' interaction within groups. In fact, ABMS is a new approach for simulating systems with interacting autonomous objects called agents [10].

The First step in understanding ABMS is to recognize the meaning of agent in this framework. Agents are independent entities, which are identifiable, discrete individuals with a set of dynamic attributes, behaviors and decision-making capability [56].

According to Casti [9] agents should contain two sets of rules: base level and a higher-level. The base level rules provide response to environment; in higher-level rules agents must have a set of rules to change the rules that provides adoption.

Macal and North [40] noted five characteristics of an agent: being identifiable, being goal-directed, having flexibility and capability to learn and adopt behavior based on experience and being autonomous and self-directed to work independently. Figure 1 illustrates a typical agent with certain attributes, methods and interactions.

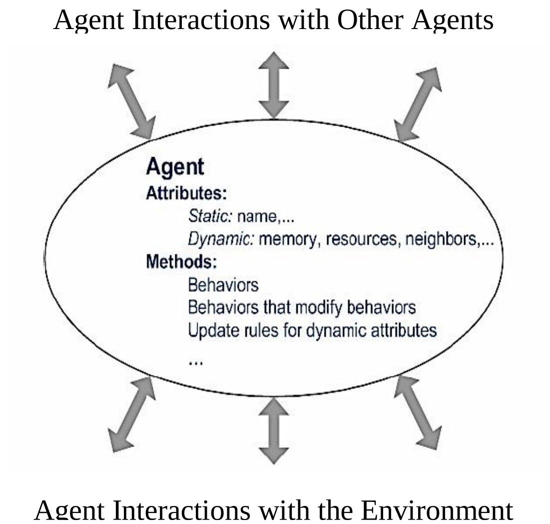


Figure 1: A typical agent [40]

Based on Macal and North [40], agents have four distinguished properties, which are autonomy, modularity, sociality and conditionality. An agent’s autonomy refers to independency and self-organizing. An agent’s modularity defines the boundary of an agent, which easily could determine whether or not an element of a model’s state is part of an agent. Sociality of an agent refers to interaction among agents. This interaction involves information exchange, recognition, collision avoidance and any other types of specific mechanisms.

Any agent has a specific state that represents its condition. The state of each agent might change over time. A collective state of all agents along with the state of environment determines the state of an agent-based model. Furthermore, an agent might have its own goals that drive its behavior and ability to learn. Moreover, each agent‘s behavior can adapt in environment based on the agent’s experiences. [40]

Agents' relationships and interaction are the main concern in modeling agent-based systems. The notion is that agents at any given time interact with a limited number of other agents. This interaction may follow some common topologies such as Network, Geographic Information Systems (GIS), Grid (cellular automata) and Soup (agents have no locational attribute in a non-spatial model).

ABMS provides a proper tool for understanding how an internal agent's attribution affects their local and global behavior. The impact of agent's attribution is then can be used to identify, explain, generate and design emergent behaviors. Because of complexity in the real world, traditional modeling tools are not applicable in defining agents' behavior. Therefore, agent based models became more and more popular. Computational power to deal with large-scale datasets and simulation tools are the other reason for popularity of ABMS. ABMS has strong roots in robotic and multi-agent systems in the field of Artificial Intelligence (AI). [6]

Agent-based models are capable of producing collective intelligence (CI) systems using well-defined simple rules. CI algorithms can be implemented in agent-based frameworks using optimization techniques. This kind of system not only can survive but also can adapt better to its environment.

To build an agent-based model to simulate a collective intelligence system, the agents, agent interactions, relationships and their related data must be identified. Agent's behavior should validate. Then model can run and results can be analyzed.

There are dozens of topologies for modeling social interactions of agents. According to Schut [58], Table 2 presents models that can be used for the specific CI complex system. For instance, the cellular automata is one of the methods which is used for simulating the CI complex system. This method uses a grid for available local information.

2.3.1 Agent-based Modeling and Simulation Toolkits

There are several toolkits which are developed for simulation of agent-based models. These platforms can be used to design and implementation CI complex systems and also artificial ecosystems.

Allan [2] categorized agent-based applications into eight main groups: business and organizations, economics, infrastructure, crowds, society and culture, biology, terrorism and military. Allan [2] investigated the most common uses of ABMS tools in the various areas such as crowd behavior, engineering and systems biology etc. He listed more than thirty software platforms and multi-agent applications.

Schut [58] categorizes ABMS platforms into four groups; programmable modeling environments such as StarLogo and NetLogo, Libraries such as Ascape, 3D/Physical/A-Life and Specialized platforms such as AgentSheets. Macal and North [40] put ABMS toolkits in three major group including desktop computing for ABMS application development such as NetLogo, MATLAB and Mathematica, large-scale development environment like Repast, Swarm and MASON and general programing language such as C++, Java and Python.

This part introduces the toolkits, which are used more frequently in scientific research. Moreover, some advantages and disadvantages of each toolkit are described. There is also an Internet link available for documentation of each framework.

- **Swarm**

The Swarm tool was introduced by Santa Fe Institute in 1994 as the first software tool specifically for studying complex adaptive systems and artificial life applications. In this model swarms have hierarchy and one swarm can contain a sub-swarm with an integrated schedule. Swarm was designed in Objective-C and has a set of libraries for building and analyzing models for developers. Java based Swarm is now available.

A swarm as a key component in the Swarm model is a collection of individuals with a schedule of events and hierarchical modeling approach. Agents inside a swarm hierarchically make the emergent behavior of the higher-level agent.

Although Swarm is still one of the most powerful simulation tools, it needs knowledge and experience programming in Objective-C, Java and Swarm code. Also, there are not many developers available for this software anymore. Swarm is open source and free under the GNU license.

See http://www.swarm.org

- **NetLogo**

NetLogo originally started with the name StarLogo. It is a simple and powerful high level programming language structure and widely used to study micro-level behavior and macro-level pattern emergence in natural and social sciences, computer sciences, biology, epidemiology, physics etc. NetLogo has a built in graphical interface with capability for model deployment and use online over the Internet.

NetLogo has many updated resources and documentations, libraries, tools, extensions and models. Based on the information on the official website of the project some of the most viewed and frequently downloaded models are Collective Behavior in the Stock Market, Building Evacuation, GIS General Examples, Traffic Scheduler and Predator Prey Game. The current version of software is NetLogo 5.0.5 released in December 2013.

See http://www.ccl.northwestern.edu/netlogo

- **Breve**

Breve is open-source free software to simulate 3D multi-agent systems and artificial life. The package uses Python object-oriented language and lets user apply OpenGL libraries to simulate and visualize realistic creatures. This software allows users to define behavior of agents in a 3D environment and see how they interact with each other. There is available support documentation for users to walk through the environment and write new simulations. Breve 2.7.2 is the last available version, released in February 2008 and available for Mac OS X, Linux and Windows.

See http://www.spiderland.org

- **MASON**

MASON stands for Multi-Agent Simulator Of Networks. It is a Java based simulation toolkit from a joint project between George Mason University Center for Social Complexity and George Mason University's Computer Science Department. It has similar features with Swarm and is fast and small compared to other competitors such as Repast. Mason has a capability of running model and visualization separately. In long simulation the system gives the possibility for simulation stop and transferring simulation and graphical interface between computers. MASON provides 2D and 3D visualization with access to modeled data, play, pause and stop, and steps of the simulations, model recovery and task performance. Users can select any object in the visualization part to track data model changes within the object. Mason version 17, released on May 3rd, 2013 is the last update available for this software.

See http://cs.gmu.edu/~eclab/projects/mason

- **Repast**

Recursive Porous Agent Simulation Toolkit, “Repast” was designed for agent-based simulation in social sciences. The University of Chicago project is like a Java-based modeling version of Swarm toolkit and developing a simulation on it necessitates knowledge of Java programming. There are some implemented models available such as Area Coverage, Boids (Java and Relogo), Diffusion, Game of Life, and Small World for users with a good source of documentation.

The last version of Repast called Symphony 2.1, released on 12 August 2013, is designed for use on small computing clusters. Repast HPC, the high performance computing C++-based version is also available for use on supercomputers and large computing clusters.

See http://repast.sourceforge.net/

- **Ascape**

Ascape is a java-based exploring general-purpose agent-based modeling tool that uses far less code than some other software like Swarm. Models in this platform can be modified easier and with available end-user tools, so that even non-programmers can explore many aspects of model dynamics. It has some features for statistical analysis and visualization tools for graph illustration. Ascape has the ability to publish over the Internet and also make video of running models. There are just few user manuals and other documentation for user support. The majority of available models are related to market modeling and economics. The current version of Ascape is 5.6.1, updated on April 28th, 2011.

See http://ascape.sourceforge.net

- **FLAME – The Flexible Large-scale Agent Modeling Environment**

FLAME agent-based modeling tools on parallel architectures was designed and developed at the University of Sheffield. It was first used in biological domains, and then extended into other fields of research such as behavioral sciences, computational economics and finance. FLAME models are designed in XML syntax that could be compiled to C codes. FLAME has an efficient inter-agent communication and birth and death algorithm that makes well-done simulation performance. The last update of FLAME was released on 27th July 2012.

See http://www.flame.ac.uk

- **AgentSheets**

AgentSheets is an agent-based simulation and games tool that uses a spreadsheet approach to build a mesh of agents' allocation in the simulation environment with drag and drop interface and publishing the model over the Internet. This platform uses a visual programming approach to build an easy to use tool for non-programmers.

Some of the features in the current version are agent animation, control web browsers, keeping track of agents for data analysis by counting each type of agent over time and real time data interaction with a slide-controllable interface. Because of simplicity, AgentSheets is widely used to learn simulation basics.

Developers have two purposes for this product, firstly for interactive simulations in computational science applications and secondly building the games for motivational and educational purposes to teach programming skills through game and simulation design.

AgentSheets in its new project named AgentCubes is going to introduce a new programming platform that makes 3D game design possible even for users without programming skills. In this new tool users are able to design behaviors in 3D, making and sharing sophisticated simulation environments. AgentSheets is funded by National Science Foundation (NFS). Its current version 4 was released in 2014 and is available for purchase on Windows and Mac compatible versions.

See http://www.agentsheets.com

- **LSD: Laboratory for Simulation Development**

LSD is a C++ based district-time platform to implement an agent-based model from Sant'Anna School of Advanced Studies in Italy. It is intended mostly for economic and social sciences modeling. LSD is easy to use even for modelers without programing experience; it automatically constructs and runs the simulation program based on user defined simulation model. There is just one paper as documentation and a tutorial available on the official website. The last version of this toolkit is LSD6.2 that was released in October 2013.

See http://www.labsimdev.org

- **MASS: Multi-Agent Simulation Suite**

MASS suite offering solutions for modeling and simulation, includes four major components: FABLES, MEME, CP and PET. MASS uses its own language, FABLES (Functional Based Language for Simulation), which is an easy-to-use java based language and needs minimum skills in programming. CP (charting package) is an output visualization, graph illustration and editor tool.

MEME (model exploration module) designs simulation experiments by generating, management and analyzing simulation results with an interactive graphical editor. It has a parameter sweeper wizard that allows running a simulation on more than one computer and on cloud computing clusters. There are some statistical analysis features and pre-processed models available for the modeler on MEME. This module also is integrated with the other parts like CP. PET (participatory extension tool) is a web-based multi-user environment for controlling agents over the Internet. PET module give educational lab tools to prepare an agent based simulation with pre-configured available models, where teachers can fix some parameters and students can use graphical interface to take control over some agent's behavior and see the resulting actions. Users also can record all simulation process as a video.

Each module in the MASS package has different versions available; the most recent update is for MEME that was released in May 2013.

See http://www.mass.aitia.ai

- **StarLogo TNG**

StarLogo TNG is a toolkit developed by MIT's STEP (Scheller teacher education program) team. This toolkit continues the past projects, StarLogo and OpenStarLogo, and introduces a graphical programming language for modeling and simulation. StarLogo TNG, like the other agent-based modeling and simulation tools, can model emergent behavior of many real-life decentralized phenomena such as market behavior, flocking birds and urban traffic.

StarLogo TNG is the next generation of StarLogo tools with 3D modeling features that make it a proper tool for educational programming as well. This version uses a colorful block-based interface that facilitates agent-based simulation programming such as making a puzzle. The project team offers a good source of documentation, tutorial and workshops within a powerful user support. StarLogo Nova is a new online platform of this project that let users build up, organize and share agent-based simulations over the online public browser without software installation. The current available version is StarLogo TNG 1.5.

See http://education.mit.edu/projects/starlogo-tng

2.4 Agent construction tools

There are also some other multi-agent systems that are not ABMS tools and intended for constructing agents. They are used to build smart resilient networks rather than just in modeling and simulation. These agent construction tools should not be confused with ABMS applications. Cougaar (cognitive agent architecture), JADE (Java agent development framework) and SeSAm (shell for simulated agent systems) are some of these agent construction tools.

2.5 Predator-prey Models

An agent-based model of predator-prey ecosystems is simulated in this thesis. Predator-prey models are quite well-documented subject in literature. Basically there are two strands of studies relevant to predator-prey models. First, population-level models which are widely studied in computational ecology and biology. In population-level approach, collective behaviors such as predator-prey are modeled using differential equations. Second, agent-based models which is the main subject of this thesis. Previous sections survey studies relevant to agent-based models. This part describes population-level models. Moreover, the main differences between population and agent based models are investigated.

Population-based model is pioneered by Volterra [67] and developed by Lotka [39]. However, the population-level predator-prey model with collective behaviors interactions is independently proposed by Lotka [39]. Therefore, predator-prey model is commonly called Lotka-Voltera model in the literature.

Lotka-Voltera model has been developed in many aspects by studies such as MacArthur and Connell [41] and May [44, 43]. Lotka [39] used population size and species abundance as a main element of a system. The model keeps track of the predator and prey population growth and dead rates. It is also worthwhile to mention that as indicated by Weisberg and Reisman [69] the effect of predation on the population of prey and the effect of prey capture on the population of predators is incorporated into Lotka-Voltera model. By merging growth and dead rate into a just growth rate actually there are four factors that these models are dealing with, (i) the predator and prey population growth and (ii) the dead rates, (iii) the effect of predation on the population of prey and (iv) the effect of prey capture on the population of predators. As argued by Levins [37], Lotka-Voltera model represents population level in predator-prey model with set of nonlinear, differential equations as follows:

Predators Population:

$$\frac{dp}{dt} = -qP \qquad (1)$$

$$\frac{dp}{dt} = ca^{'}PN - qP \qquad (2)$$

Preys Population:

$$\frac{dN}{dt} = rN \quad (3)$$

$$\frac{dN}{dt} = rN - a' PN \quad (4)$$

where P is predators population, N denotes prey population, t is time. Also, r shows prey growth rate, a' denotes searching efficiency/attack rate, q is predator's mortality rate and c indicates predators' efficiency at turning food into offspring (conversion efficiency).

Lotka-Volterra model uses population density as a state variable. According to Weisberg and Reisman [69], in the classic population-based predator-prey models there is just an aggregation of many organisms' properties without any explicit image of individuals or their properties, while in the agent-based models (ABMs) each agent has its own set of properties.

As discussed agent-based models also include set of rules and assumption about how agents interact over time. In contrast, because of structure of real life systems, classic population level models are not able to describe the reality of collective behavior phenomena. Population-level models have very limitations because of their simplicity and mathematical analysis. While in ABMs each agent has its own set of state variables and their dynamics are explored by computational simulation instead of mathematical analysis. [69]

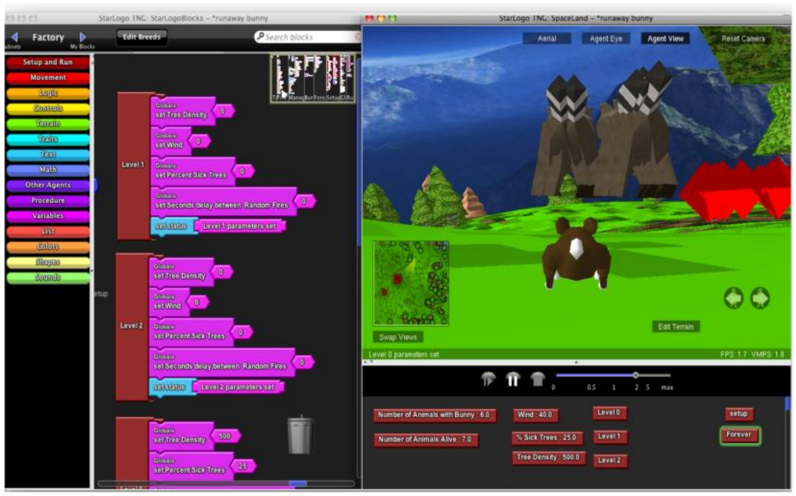
The population level models are based on the assumption that community is silent and obscure. Growth and death rate and interaction in population will be assumed as individual agents, which are discrete. In recently developed IBM models for each unit of time all agents are doing set of rules for their move, predate and death. To convert a population-based model into individual-based model, each agent needs to have explicit set of intrinsic and environmental rules and attributes, which are not included in population-based models.

2.6 Visualization in Existing Simulation Toolkits

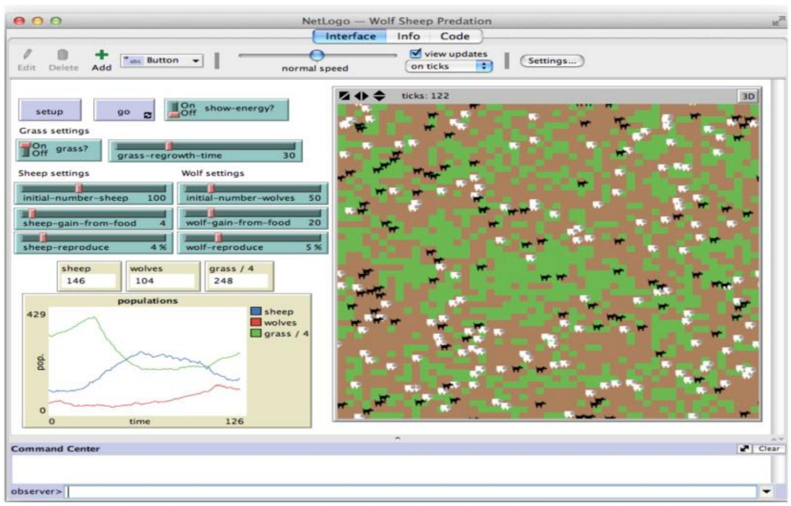
ABMS has its own requirements for visualization. In fact, according to Dorin and Geard [17], all phases of ABMS can benefit from visualization. Using ABMS toolkits, visualizer animates the environment, its inhabitants and motions in a 2D real-time manner according to the data receiving from visualization engine. Visualization is one of the most important features of vision-based simulation of CI complex system which gives a sense of space, interactions, monitoring parameters and emergent behavior.

Graphical User Interface (GUI) is responsible for a graphical animated visualization of the simulation for toolkits that are introduced in section 2.4. In particular, real-time charts provide users with real-time numeric data diagrams which represent the statistical information of the ecosystem. Regarding our purpose of study, NetLogo and Repast S. has more related interface. For instance, in Repast S., GUI includes different types of charts such as sequence graph and histogram for data visualization with ability of taking snapshots and recording movies over time. [11]

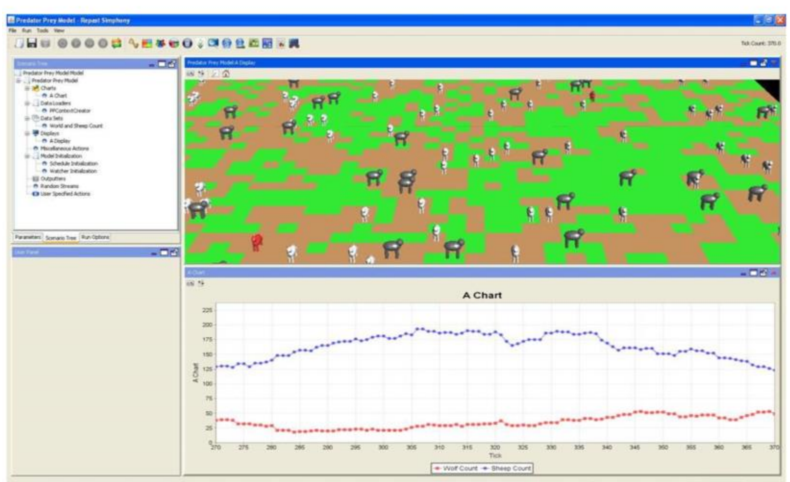
Figure 2 shows simulation and visualization of an artificial ecosystem in the latest version of StarLogo TNG, NetLogo and Repast S., respectively.



(a)



(b)



(c)

Figure 2: Simulation and Visualization of Artificial Ecosystem using (a):StarLogo TNG, (b):NetLogo, (c):Repast S.

Tatara et al., [64] argued that in Repast S., the simulation parameters could set in GUI, to adjust the initial number of predators and preys. Also the resource which is grass could be added to each location on the grid in run environment. Moreover, according to North et al., [48], the visual capabilities of Repast S. includes a model operation and configuration by point and click; an integrated 2D and 3D view; automated connections to data sources; and connection with external statistical programs for conducting analysis and visualizing model results.

It is important to mention that there are some limitations for visualization of predator-prey models. For instance, according to Sklar [62], running intense computations and visualization over many iterations in NetLogo and similar Java-based simulator may be too slow and systems might breakdown. The designed framework for this thesis, compare to the similar available models has a proper level of complexity and features, with a stationary environment (i.e. not static). Moreover, control tools and the GUI which is typically responsible for setting up, controlling simulation and visualization make it possible to define as many species as required in any preferred configuration. Furthermore, the proposed framework is developed from zero point so comparing to the existing developed models, we may face less constraint for future structure enhancement such as increasing the level of complexity, changing the internal and environmental rules and customizing the model based on given specifications.

The presented toolkits in section 2.3.1, has been mostly developed by C++, Java and Python while the user interface layer of my proposed work is implemented using C#.Net programming language and utilizes graphical components embedded within .Net framework.

The next chapter explains the whole process of design, implementation and visualization of our proposed simulation structure in details.

2.7 Survival Analysis

One Survival analysis (i.e. also known as reliability analysis and failure time analysis) is a statistical approach that aims to model time-to-death in a given system such as modeling failure in mechanical systems, lifetimes of products in marketplace, and time-to-death of biological entities.[35] Using survival analysis, one can determine the percentage of population that will survive up to given time or investigate the effects of particular parameters on system survival. In the other words, survival analysis provides techniques for estimating expected lifetime and important features of the system that influence the probability of death or failure at a specific time in a system.[51] Survival analysis exploits probability density function (pdf) which represents the rate of death in each time step. Survival function is defined as Eq. (5).

$$\mathrm{S(t) = P(T > t)} \quad , \mathrm{S(0) = 1} \quad (5)$$

Where $S(t)$ denotes survival function and T is a random variable that denotes time of death. The lifetime distribution function is defined as the complement of survival function as shown in Eq. (6).

$$F(t) = P(T \le t) = 1 - S(t) \quad (6)$$

Where $F(t)$ denotes lifetime distribution function. The derivative of lifetime distribution function is the density function of the lifetime distribution (i.e. event density) and is calculated using Eq. (7).

$$f(t) = \frac{d}{dt} F(t) \quad (7)$$

Survival analysis is frequently represented in terms of hazard function, denoted as $h(t)$, which is instantaneous likelihood of a system breaking down as function of time. [14] Essentially, the hazard function measures the instant risk of death at time t conditioned on the survival to that time as shown in Eq. (8).

$$h(t) = \lim_{\Delta t \to 0} \frac{P[(t \leq T < t + \Delta t) \mid T \geq t]}{\Delta t} = \frac{f(t)}{S(t)} \quad (8)$$

According to Lawless [36], the Cox-Mixed proportional hazard model is the most frequently applied approach for incorporating the covariate effect on lifetime of a system. This semi-parametric model introduced by Cox [14] is depicted in Eq. (9).

$$h(t) = e^{x_t \times \beta} \times h_0(t) \quad (9)$$

In this equation, $h(t)$ is the hazard function, x_t is a vector of time-varying covariates, β is the parameter of interest which is estimated using cox regression and indicates the effect of each covariate on the probability of death in each time step, $h_0(t)$ is an initial non-parametric hazard function.

The Cox regression models the effect of covariates on the hazard rate without considering any specific initial hazard rate or knowledge of absolute risk of death. Moreover, it estimates the relative risk of death rather than the absolute risk. Assuming two different observations, i and j, the corresponding Cox-regression is calculated by Eq. (10).

$$h_i(t) = h_0(t) \times e^{\sum_{k=1}^{K} \beta_k \times x_{ik}} \quad , h_j(t) = h_0(t) \times e^{\sum_{k=1}^{K} \beta_k \times x_{jk}} \quad (10)$$

Then, the hazard ratio for these two observations is acquired using Eq. (11).

$$\frac{h_i(t)}{h_i(t)} = \frac{h_0(t) \times e^{\eta i}}{h_0(t) \times e^{\eta j}} = \frac{e^{\eta i}}{e^{\eta j}} \quad (11)$$

η is elasticity of death with respect to energy levels and is computed by Eq. (12).

$$\eta_i = \sum_{k=1}^{K} (\beta_k \times x_{ik}) \quad , \eta_j = \sum_{k=1}^{K} (\beta_k \times x_{jk}) \quad (12)$$

3 Methodology

3.1 Artificial Agent-Based Ecosystem Modeling

In order to develop an artificial ecosystem with proper degree of complexity that can provide sufficient information regarding energy flow, we develop a complex stochastic model to simulate a multi-species predator-prey ecosystem. To do so, we expand the Reynolds' [54] Boids in three directions. First, we extend the original version from a single species ecosystem to a multi-species predator-prey system. In our system, it is possible to define as many species as required in any preferred configuration. It is possible to define only one species which is equivalent to the original version of Boids, also we can define a few prey species with same food diet without predators which will simulate a competition among species within same level of food chain due to the limited food sources. Ultimately, it is possible to define the ecosystem as combination of prey and predator species where there are inter-competition among species within same level of food chain and intra-competition among those belonging to different levels of food chain. In our ecosystem, fruit trees are in the bottom of the food chain. These trees have a fixed growth rate and vanish as soon as their fruits are consumed by birds. The second extension is regarding the rules governing the flocks. In Reynolds' original system, only three rules named separation, alignment, and cohesion govern the flocks' collective behavior whereas we extended these rules to ten which will be entailed in following sections. Finally, we introduce the energy parameters into the system to model the energy flow within an ecosystem. We will discuss the details of environmental parameters and collective behavior in our proposed mode in following subsections.

3.1.1 Environmental Parameters

The hosting environment of the ecosystem is a stationary environment (i.e. not static) with two features defined as *E=<W,T>* in which *W* denotes the weather condition and *T* presents the set of fruit trees within the environment. It is noteworthy that weather condition has an important influence on the flocking behavior whereas fruit trees provide the herbivore birds (i.e. preys) within the ecosystem with required energy. Weather condition is defined as a set of state-probability pairs which present the probability of environment being in a particular weather condition. We consider three possible weather conditions in simulations including rainy, windy and sunny states. Thus, we can define *W* as a triple *W=<(r,pr),(w,pw),(S,ps)>* representing rainy, windy and sunny weather conditions along with their associated probabilities respectively.

We derived the following mathematic formulas based on fundamental physics and biological rules.

Fruit trees, represented by *T*, occupy the bottom level of food chain and thus can be considered as energy resources of the ecosystem. Each tree is defined as a triple *Ti=<P,Tc,Fc>* in which *P* denotes the position of the tree within the environment, *Tc* represents the tree's age and *Fc* denotes the current amount of fruit left on the tree. The amount of fruit on the tree is governed by a dynamic decay model as depicted in Eq. (13).

$$F_c(t+1) = F_c(t) - w_b \times |B(t)| - e^{w_d \times t} \quad (13)$$

$F_c(t)$ denotes amount of fruit in time t, $|B(t)|$ is the number of birds feeding from tree in time t, and w_b and w_d are weighting parameters for food consumption by birds and natural decay, respectively. Essentially, the model presented in Eq. (13) states that the amount of fruit on a tree is decreased proportional to the weighted combination of number of birds feeding on that tree and its age (i.e. older trees produce less fruit). The initial value is generated by a uniform random distribution *U* as *Fc(0)~U[Fmin,Fmax]*, where *Fmin* and *Fmax* are minimum and maximum allowed fruit amount on a tree, respectively. Furthermore, we remove a tree from ecosystem with probability P_{tr} computed by Eq. (14).

$$P_{tr} = e^{-w_{tr} \times F_c(t)/t} \quad (14)$$

Where P_{tr} denotes the probability of tree removal and W_{tr} presents a normalizing weight. This probability increases as the tree's fruit decrease and its age increases until it reaches to one and tree is removed. As soon as a tree is removed from the environment, it is replaced by a tree with a specific probability.

3.2 Collective Behavior

The Collective behavior within the ecosystem emerges from interactions among birds in terms of flocking, and hard-wired innate behavior in terms of foraging, breeding, and competing. We define a generic bird as a decuple *B=<T,S,M,P,V,A,G,E,SR,C* > where *T* denotes the bird's type and determines whether it is a carnivore (predator) or herbivore (prey) whereas *S* determines the species that the bird belongs to. *M* denotes the bird's behavioral mode and can be set to one of the following behaviors: flocking, escaping, feeding, seeking, breeding, hunting, competing and resting. *P* denotes the bird's position whereas *V* denotes its velocity. It is noteworthy that velocity vector determines the direction of the motion and its magnitude. *A* defines the bird's age, *G* presents its gender, *E* denotes its energy level, *SR* is the bird's sight radius and ultimately *C* denotes the average position (i.e. center of the mass) of the closest flock to the bird which is in its sight radius and belongs to the same species as the bird's.

3.2.1 Choreographed Motion

In order to simulate the choreographed motion of herbivore birds, we utilize three flocking rules proposed by Reynolds' [54] including cohesion, alignment and separation. (Figure 3)

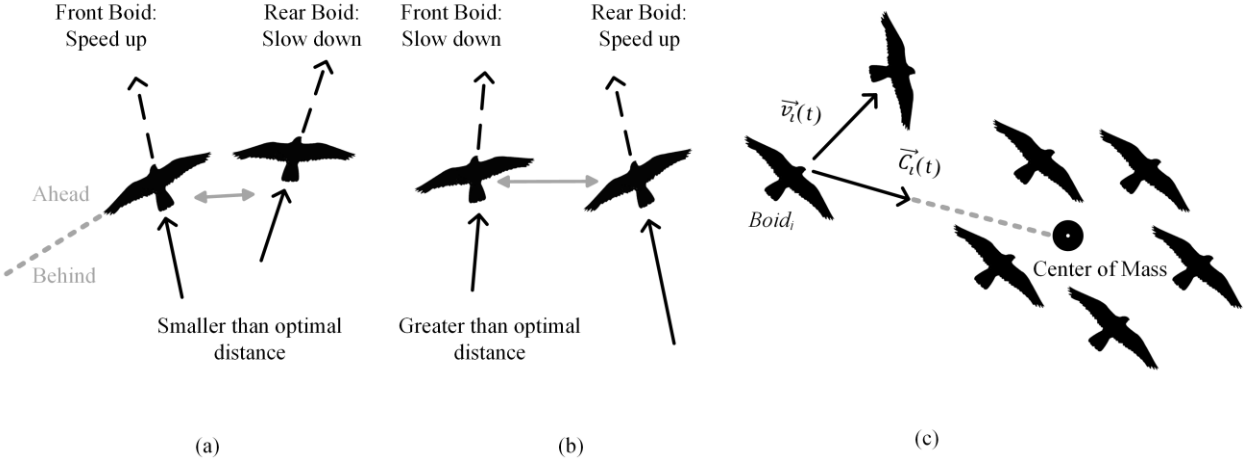


Figure 3: The conceptual presentation of Reynolds' Boids. (a) Separation rule. (b) Alignment rule. (c) Cohesion rule. [29]

Cohesion is an act of steering to move toward the center of mass of local flock mates. Performing cohesion demands each bird in the flock to compute its local perceived center of mass which is essentially the local average position of the bird's flock mates that are in its sight radius.

We derived the following mathematic formulas based on Reynolds model and motion rules in physics.

The perceived center of mass is computed by Eq. (15).

$$\vec{C}_i^p(t+1) = \Big(\sum_{\substack{j \in S_i \\ j \neq i}} \vec{P}_j^h(t)\Big)/N \quad , \forall i : |\vec{P}_i^h(t) - \vec{P}_j^h(t)| \leq SR_i^h \quad (15)$$

$C^p i(t)$ is the perceived center of mass of bird i, $Pi(t)$ is the position of herbivore bird i, $SR^h{}_i$ is the sight radius of herbivore bird i, S_i is the species that bird i belongs to and N is the number of flock mates of bird i that are in its sight radius and belong to same species as bird i. Using the local center of mass, the velocity vector of bird i influenced by cohesion is updated by Eq. (16).

$$\vec{V}_i^h(t+1) = \vec{V}_i^h(t) + w_{coh} \times (\vec{C}_i^p(t) - \vec{P}_i^h(t)) \quad (16)$$

In this equation $V^h{}_i(t)$ is the velocity vector of herbivore bird i in time t and w_{coh} is the weighting parameter of cohesion rule.

Alignment action refers to steering towards the average heading of local flock mates. Contrary to cohesion, in alignment each bird calculates the perceived local velocity in order to maintain a coordinated direction and pace with its local flock. Perceived velocity can be computed similar to center of mass by substituting the position with velocity as shown in Eq. (17).

$$\vec{C}_i^v(t+1) = (\sum_{\substack{j \in S_i \\ j \neq i}} \vec{V}_j^h(t)) / N \quad , \forall i : | \vec{P}_i^h(t) - \vec{P}_j^h(t) | \leq SR_i^h \quad (17)$$

$C^v{}_i(t)$ is the perceived center of velocity of bird i and $V_i(t)$ is the velocity vector of bird i in time t. The velocity vector of bird i is tweaked due to alignment as depicted in Eq. (18).

$$\vec{V}_i^h(t+1) = \vec{V}_i^h(t) + w_{al} \times \vec{C}_i^v(t) \quad (18)$$

Contrary to cohesion and alignment, separation rule diverge the flock to avoid crowding local flock and colliding into each other. In order to perform separation, a minimum allowed distance among the birds, d_{min}, is defined to detect the birds that are closer than the allowed distance to bird i. Then, the velocity of bird i is updated in opposite direction of those birds that violate the minimum space in order to provide the appropriate spacing. The update equation for separation is shown in Eq. (19).

$$\vec{V}_i^h(t+1) = \vec{V}_i^h(t) + w_{sep} \times \sum_{\substack{j \in S_i \\ j \neq i}} (\vec{P}_i^h(t) - \vec{P}_j^h(t)) \quad , \forall i : |\vec{P}_i^h(t) - \vec{P}_j^h(t)| \leq d_{\min}) \qquad (19)$$

Another aspect of flocking behavior that plays a critical rule in predator-prey ecosystems is escaping from the predators. Escaping mechanism is based on moving in opposite direction of the predators within sight radius with maximum possible speed. The updating rule of velocity vector while escaping from predator is shown in Eq. (20).

$$\vec{V}_i^h(t+1) = \vec{V}_i^h(t) + w_{esc} \times \sum_{j \in T_{pred}} (\vec{P}_i^h(t) - \vec{P}_j^c(t)) \quad , \forall i : |\vec{P}_i^h(t) - \vec{P}_j^c(t)| \leq SR_i^h) \qquad (20)$$

Where T_{pred} denotes all predators that are within the sight radius of bird i, $P^c_j(t)$ is the position vector of carnivore bird j, and w_{esc} presents the weight of escaping rule. Obviously, surviving an attack is the top priority and thus the weight of escaping must be much higher than those of flocking behaviors.

Another important behavior that emerges from flocking birds is to careen toward the food source. According to this behavior, as soon as fruit tree appears within the sight radius of a bird, it tweaks its current direction to reach the food. This individual act along with cohesion and alignment leads the flock to the food source. The velocity of bird i is updated using Eq. (21) in order to lead it to food.

$$\vec{V}_i^h(t+1) = \vec{V}_i^h(t) + w_{food} \times (\arg\min_{\vec{P}_j^T}(|\vec{P}_j^T - \vec{P}_i^h(t)|) - \vec{P}_i(t)) \quad , \forall i : |\vec{P}_j^T - \vec{P}_i^h(t)| \leq SR_i^h) \qquad (21)$$

w_{food} denotes the importance weight of foraging and $P^T{}_j$ denotes the position of tree *j* within the environment that is also within the sight radius of bird *i*. It is noteworthy that as shown in Eq. (19), bird *i* steers toward the closest food source in terms of Euclidian distance.

The velocity update rules mention in Eq. (15) up to Eq. (21) only apply to herbivore birds which are on top of fruit trees in food chain. On the other hand, predator species do not illustrate social behaviors. The only motivation in their motion is seeking herbivore birds and prey them. The predator attacks its closest prey by changing its direction toward it as shown in Eq. (22).

$$\vec{V}_i^c(t+1) = \vec{V}_i^c(t) + (\arg\min_{\vec{P}_j^h}(|\vec{P}_j^h(t) - \vec{P}_i^c(t)|) - \vec{P}_i^c(t)) \quad , \forall i : |\vec{P}_j^h(t) - \vec{P}_i^c(t)| \leq SR_i^c) \quad (22)$$

Where $V^c{}_i(t)$ and $P^c{}_i(t)$ are the velocity and position vectors of predator *i* in time *t*, respectively. $P^h{}_j$ denotes the position of prey *j* and $SR^c{}_i$ is the sight radius of predator *i*. Furthermore, in order to confine the inhabitants within the geographical borders of the ecosystem, it is required to softly reverse the direction of those birds that are close to the borders. This constraint can be satisfied using Eq. (23).

$$\vec{V}_i(t+1) = \vec{V}_i(t) + (\vec{P}^B - \vec{P}_i(t)) \quad , \forall i : |\vec{P}^B - \vec{P}_i(t)| \leq d_{\min}) \quad (23)$$

In this equation, P^B denotes the universal coordinates of the geographical borders of the ecosystem and d_{min} denotes the minimum allowed distance that a given bird can get close to the borders.

As aforementioned in previous section, three different weather conditions are defined including sunny, windy and rainy. In case of sunny weather, the velocity vectors remain unchanged. However in case of windy or rainy weather, velocity vectors are modified using Eq. (24) and Eq. (25), respectively.

$$\vec{V}_i(t+1) = \vec{V}_i(t) + w_W \times \vec{W} \quad (24)$$

$$\vec{V}_i(t+1) = w_R \times \vec{V}_i(t) \quad , w_R < 1 \quad (25)$$

W denotes the velocity vector of wind. In case wind blows in the same direction as bird's direction, it will increase its speed; otherwise it will damp its speed. It is noteworthy that the velocity of wind is a random uniform variable within range of [W_{min},W_{max}]. W_W is the weight of wind and determines the influence of wind on the bird's motion. We assume that in rainy weather the flight speed reduces. The control parameter W_R determines this speed reduction.

Moreover, in order to guarantee a smooth motion, maximum allowed speed is defined as a physical constraint on birds' velocity. In order to satisfy this constraint, Eq. (26) is utilized to limit the velocity of those birds that have higher velocity than the allowed one.

$$\vec{V}_i(t+1) = (\vec{V}_i(t) / |\vec{V}_i(t)|) \times S_i^{\max}(t) \quad , \forall i : |\vec{V}_i(t)| > S_i^{\max}(t) \quad (26)$$

S_{maxi} denotes the maximum allowed speed and is a scalar entity whereas $|V_i(t)|$ is the magnitude of the velocity vector of bird *i* (i.e. its speed) and is scalar as well. As shown in this equation, maximum allowed speed for each bird changes through time. The dynamic of this element is modeled as Eq. (27).

$$S_i^{\max}(t) = (w_E \times E_i(t)) + (w_A / A_i(t)) \quad (27)$$

In this equation, $E_i(t)$ denotes the energy level of bird *i* in time *t* and $A_i(t)$ denotes the age of bird *i* in time *t*. W_E and W_A are energy and age influence weights, respectively. Intuitively, the equation states that those young birds with higher energy can move faster than old birds with low energy. It is noteworthy that by setting W_E and W_A higher for predators, we give them higher maximum allowed speed than herbivore birds.

Ultimately, by consecutive updates of velocity vectors using Eq. (15) up to Eq. (27), the position vectors of birds are updated using Eq. (28) in next simulation step.

$$\vec{P}_i(t+1) = \vec{P}_i(t) + \vec{V}_i(t+1) \quad (28)$$

This equation is equivalent to position update equation utilized in particle swarm optimization algorithm introduced by Kennedy and Eberhart [31]. It is noteworthy that the initial velocity and position vectors of birds are generated randomly following a uniform distribution.

3.2.2 Foraging and Competing for Resources

As mentioned in previous sections, each bird regardless of its species and type seeks food to survive. In case of herbivore birds, as soon as they find a fruit tree, they fly toward it following the trajectory determined by Eq. (21) and when they reach it, they land on the tree branches and feed until they reach the maximum allowed energy. On the other hand, predators careen toward their closest prey following the trajectory defined by Eq. (24), and as soon as their reach to a minimum distance with the prey, they consume them. However, when different species approach the same resource, competition starts over that particular resource. We define competing groups as groups with same types (i.e. within same level of food chain) that belong to different species. When these groups are approaching a common resource and the distance among their center of masses reaches a predefined threshold, the competition begins. In order to model the competition, we exploit a probabilistic model to compute the domination probability of each species within the competition as shown in Eq. (29).

We derived the following mathematic formulas based on probability fundamentals.

$$p_i(dom) = \frac{|B_i| \times \mu_i^E \times \sum_{j \neq i} \mu_j^A}{\mu_i^A \times \sum_{j \neq i} |B_j| \times \mu_j^E} \qquad (29)$$

In this equation $p_i(dom)$ denotes the domination probability of group i, $|B_i|$ is the number of birds within that group, μ_{Ei} presents the average of energy within group i and μ_{Ai} denotes the average age of the birds within group i. Basically, the model states that younger groups with larger population and higher energy have more chance to win the competition. We utilize a roulette wheel selection strategy with probabilities computed by Eq. (29) to determine which group dominates other. Contrary to deterministic approach, in this approach all groups have the chance to win the competition, however those groups with higher domination probability have higher chance to dominate others. Using this stochastic approach, we can approximate the actual competition within biological systems in which stronger groups frequently dominate others, however it is observed that weaker groups conquer others infrequently. Ultimately, the dominant group marks the resource as its territory and exploits it until the resource is consumed or the group has exploited sufficiently from it. Also, the dominant group may rest in its territory even after having absorbed sufficient energy from that resource.

3.2.3 Breeding and Termination

Stability of an ecosystem can be expressed in terms of breeding and termination of its inhabitants. If the rate of birth and death are fine-tuned, the dynamics of ecosystem converges to a stable equilibrium. We model breeding process as an informed stochastic process similar to natural selection based on survival of fittest. However, in current version of our model, we ignore the evolutionary process.

$$p_{ij(bre)} = \begin{cases} 1/[1+e^{-\log\left(\frac{E_i(t)\times E_j(t)}{A_i(t)\times A_j(t)}\right)}] & \forall i,j:(G_i \neq G_j)\wedge(S_i = S_j)\wedge(|P_i - P_j| \leq SR_i, SR_j) \\ 0 & Otherwise \end{cases} \quad (30)$$

In this model $P_{ij}(bre)$ denotes the mating probability between birds i and j. G_i denotes the gender of bird i whereas S_i denotes the species it belongs to. According to Eq. (30), if two birds have same gender, or belong to different species, or are not in a visible distance, their mating probability is zero. On the other hand, if two birds satisfy the mentioned constraints their mating chance depends on their energy level and age. Younger and more energetic pairs have more chance to produce offspring. In other words, fitness of a pair is determined based on age and energy features. Using Eq. (30), each pair of birds that satisfy the mentioned conditions, produce an offspring with randomly assigned gender and with same species as parents with probability p(b).

On the other hand a bird may get terminated due to several reasons. A bird is removed from ecosystem if its age exceeds the maximum allowed age $(A_i>A_{max})$ or its energy level goes beneath the minimum allowed energy level $(E_i<E_{min})$. These two rules simulate the natural dead and starvation, respectively. Moreover, a bird may die because of disease or being separated from its flock. We utilize a sigmoid function to model the disease as shown in Eq. (31).

$$p_i(dis) = 1/(1+e^{\frac{-w_{dis}\times A_i(t)}{E_i(t)}}) \quad (31)$$

$p_i(dis)$ denotes the disease probability for bird i and W_{dis} denotes a normalization weight. As the age increases and the energy level decreases, probability of getting disease increases proportionally. Moreover, we exploit a similar approach to determine the probability of being deceased because of getting lost. A bird is considered as lost when there is no bird within its sight radius that belongs to the same species as it belongs to. The probability of removing a lost bird is computed using Eq. (32).

$$p_i(los) = 1/(1 + e^{-w_l \times t_l}) \quad (32)$$

Where *pi(los)* is the probability of terminating lost bird i, w_l is a normalization weight and t_l is the total time of being lost. As this time increases, the chance of death increases as well. Additionally, an herbivore bird may die by getting preyed by a predator as discussed in section 3.2.2. Ultimately, competition among groups over resources may have some casualties. In order to address this, we utilize the domination probability introduced in Eq. (25) to model the probability of getting deceased during the competition. This probability is computed using Eq. (33).

$$p_i(cas) = w_{cas} \times (1 - p_i(dom)) \quad (33)$$

Where $p_i(cas)$ denotes the probability of death of bird i due to competition, W_{cas} is a normalizing weight and $p_i(dom)$ is the domination probability. According to this equation, dominant birds have more chance to survive a competition than dominated birds.

3.2.4 Energy Flow

Energy flow plays a crucial role in the life span of the ecosystem. Energy flow is associated with the behavioral mode of birds. Regarding this behavioral mode, a bird consumes or gains energy in each time step. The initial energy of a bird is assigned randomly using a uniform distribution within the range of $[E_{min}, E_{max}]$. We model the influence of each behavioral mode on bird's energy level using Eq. (34). All these energy values are abstract values.

$$E_i(t+1)=\begin{cases} \mathrm{w}_{\mathrm{flo}}\times E_i(t) & ,\mathrm{M_i}(t): \mathit{Flocking} \\ \mathrm{wesc}\times E_i(t) & ,\mathrm{M_i}(t): \mathit{Escaping} \\ \mathrm{w}_{\mathrm{fee}}\times E_i(t) & ,\mathrm{M_i}(t): \mathit{Feeding} \\ \mathrm{w}_{\mathrm{see}}\times E_i(t) & ,\mathrm{M_i}(t): \mathit{Seeking} \\ \mathrm{w}_{\mathrm{bre}}\times E_i(t) & ,\mathrm{M_i}(t): \mathit{Breeding} \\ \mathrm{w}_{\mathrm{hun}}\times E_i(t) & ,\mathrm{M_i}(t): \mathit{Hunting} \\ \mathrm{w}_{\mathrm{com}}\times E_i(t) & ,\mathrm{M_i}(t): \mathit{Competing} \\ \mathrm{w}_{\mathrm{res}}\times E_i(t) & ,\mathrm{M_i}(t): \mathrm{Re}\mathit{sting} \end{cases} \qquad (34)$$

where $E_i(t)$ is the energy level of bird i in time t and $M_i(t)$ is its behavioral model in time t. w_{flo} is the energy consumption rate while flocking, w_{esc} is the energy consumption rate while escaping from predator, w_{fee} is the energy absorption rate while feeding, w_{see} is the energy consumption rate while seeking for food, w_{bree} is the energy consumption rate of breeding process, w_{hun} is the energy consumption rate for predators while hunting herbivore birds, w_{com} is the energy consumption rate while competing over resources and territory, and finally w_{res} is the energy consumption rate while resting on a tree. These rates are different for various activities, for example resting energy consumption rate is much less than escaping energy consumption rate. The goal of this research is to investigate the relation among these rates to maximize the stability of the artificial ecosystem.

3.3 Simulation and Visualization Structure

In order to implement the proposed model, we designed an object-oriented simulation and visualization architecture. The proposed structure consists of two layers and a data repository. It is noteworthy that due to layered characteristics of the architecture, it can efficiently support distributed processing for massive simulations by distributing each layer to an independent machine. The block diagram of our simulation and visualization structure is shown in Figure 4.

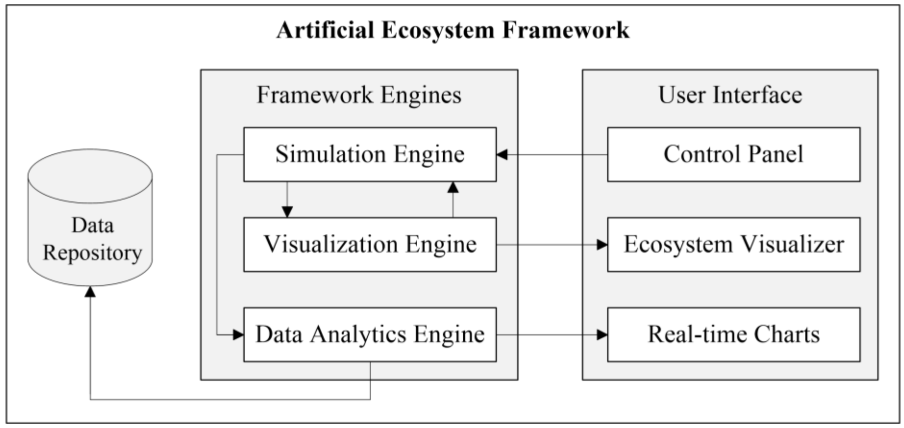


Figure 4: Block diagram of proposed simulation structure for artificial ecosystem.

The first layer is user interface and consists of three main components including control panel, ecosystem visualizer and real-time charts. Control panel provides the user with an interface to tweak the values of parameters applied in simulation including environmental parameters such as weather conditions and trees, parameters associated with species such as number of species, population size for each species, maximum velocities, sight radius etc. and control parameters such as simulation steps. Using this control panel, researchers can tune the initial values to their preferred values and observe their influence on the dynamics of system. Ecosystem visualizer animates the environment and its inhabitants and their motions in a 2D real-time manner according to the data receiving from visualization engine. It is noteworthy that visualizer is able to animate in 3D as well. Real-time charts provide the user with real-time numeric data diagrams representing the statistical information of the ecosystem. The user interface layer is implemented using C#.Net programming language and utilizes graphical components embedded within .Net framework. A snapshot of user interface is illustrated in Figure 5.

The second layer provides the architecture with required engines including simulation, visualization and data analytics engines. The simulation engine is the core of structure and is responsible for behavioral modeling of the species and simulating the environment. Essentially, it implements the proposed stochastic model of ecosystem introduced in section 3. This engine utilizes the behavioral rules and information acquired from visualization engine to determine the individuals' motion trajectory. The initial condition of this engine is provided by control panel in user interface layer. Visualization engine is responsible for processing geometric information acquired from simulation engine and mapping it to graphical information which, in turn is utilized by ecosystem visualizer to render the data and animate it. These two engines are implemented using C#.Net programming language as well. Furthermore, the visualization engine can be implemented by any available game engine. Ultimately, data analytics engine analyzes the dynamics of simulation in real-time and displays it on the charts. This engine utilizes STATA data analysis toolkit. Ultimately, data and its analytical results are stored in the data repository in Excel spread sheet format for further studies.

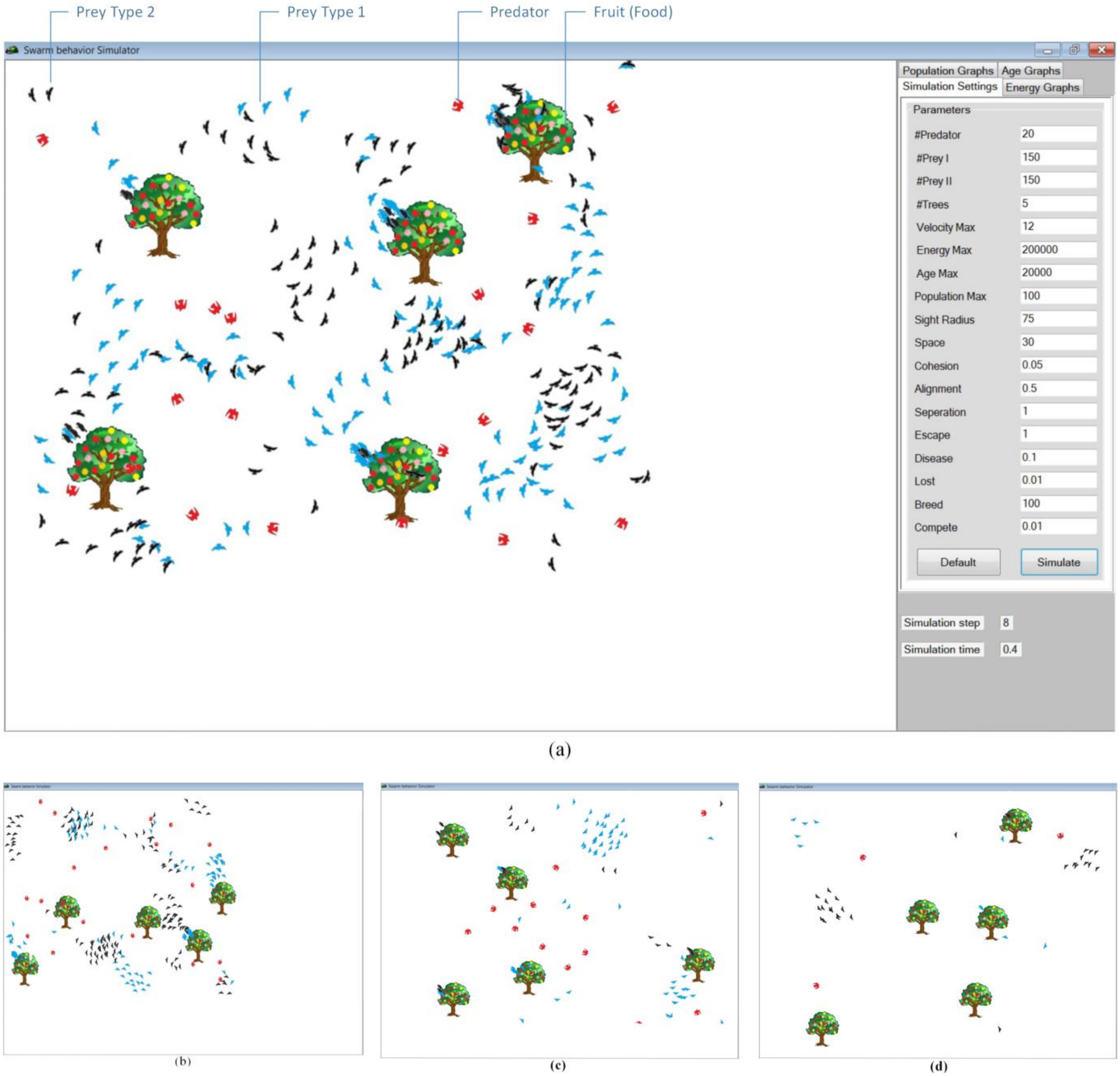


Figure 5: Snapshots of developed structure (a) system at the initial stage of simulation. (b) After 50 time steps, (c) after 248 time step (d) after 592 time steps, decreasing prey population results reduction in predators population.

4 Experimental Results

In order to evaluate the proposed model of artificial ecosystem and validate the introduced structure, we implemented the integrated architecture. Then, we repeated the simulations for 11,000 times with the frequency of 50Hz to investigate the dynamics of the model with different sets of energy parameters and their impact on the systems lifetime using the survival analysis. The simulations perform both data analytics and visualization, concurrently. In order to find the effective energy factors that influence the lifetime, all parameters except the energy factors are clamped to constant values through the simulations.

The initial values of parameters are set as follows: number of species is set to two from which one species is set as predators and the rest are herbivores. Maximum allowed age for individuals within all species is set to 20,000 time steps. The default prey population is set to 200 whereas predator population is set to 15. The carrying capacity (i.e. maximum allowed population size) of all species is set to 100 birds. Minimum allowed energy level for all species is set to zero and maximum allowed energy level that each individual can gain is set to 20,000. The maturity age (i.e. age threshold for mating) is set to $Age_{max}/100$. The maximum velocities for predators and preys are set to 6 and 12, respectively. Moreover, the sight radius for preys and predators is set to 75 and 120, respectively. Also, the minimum allowed space among the birds of a flock is set to 30. Furthermore, normalization weights for disease, separation, escaping, cohesion, alignment and getting lost are set to 0.1, 1.0, 1.0, 0.05, 0.5, and 0.01, respectively.

The initial parameters are adjustable from interactive user interface. This user interface also illustrates real time graph of population level, energy level and age level of each group within the artificial ecosystem in each run of simulation (Figure 6).

(a) (b) (c) (d)

Figure 6: A snapshot of user interface of developed structure; user control panel and real-time charts are shown. (a) Initial values of parameters are adjustable for each simulation. (b) Real time graph indicates population of each group. (c) Energy level of each species and (d) Age level of each group.

Feeding, flocking, hunting, resting and seeking energies are randomly generated using a uniform distribution within the range of [1, 20×Log(*EnergyMax*)]. Using the aforementioned initial conditions for clamped parameters and running the simulations for 11,000 times, the statistical data regarding all observations can be summarized as Table 3.

Table 3: Statistical summary of observed simulations.

Variable	# SIMULATION	Mean	SD	Min	Max
Lifetime of each simulation	11000	1998.726	2912.157	166.9965	51040.36

Table 4: Statistical summary of random assigned initial energy parameters

Variable	Mean	SD
Breeding energy	32.43381	17.29179
Competing energy	32.57229	17.09719
Escaping energy	32.51542	17.3328
Feeding energy	32.65927	17.32272
Flocking energy	32.56392	17.32749
Hunting energy	32.45901	17.34618
Resting energy	32.37376	17.36702
Seeking energy	32.38359	17.18797

As shown in Table 3, within the simulations the minimum lifetime of the ecosystem was 166.9965 time steps whereas its maximum lifetime was 51040.36. Also, the average lifetime of simulations was 1998.726 with standard deviation of 2912.157.

4.1 Survival Analysis Results

Since energy factors are continuous, in order to find those energy factors that have considerable influence on the sys-tem lifetime (i.e. predictor factors in final model) we exploit Survival Analysis using Cox proportional hazard regression. As aforementioned, the Cox regression is a semi-parametric model that extends survival analysis without having any specific assumption about the initial hazard rate. In Cox univariate regression, a predictor is embedded into the final model if its $\chi 2$ test has a p-value of 0.2 or less. Using this scheme, the results suggest that competing, resting and feeding energies contribute much less than others on the system's lifetime. In next step, we investigate the interactions among energy parameters. Since there is no structural approach to determine the interactions among energy factors, we utilize a brute force technique which tests all possible interactions. It is noteworthy that due to small variable space, it is feasible to find the significant interactions by testing each interaction, independently. Since there is no structural approach to determine these interactions, this study utilizes a brute force technique which tests all possible interactions (Table 6). Using this technique, we observed that there are two interactions among energy parameters including interaction between flocking and hunting energies, and interaction between flocking and seeking energies.

The Cox regression is valid if and only if the proportionality assumption is satisfied for all covariates. One of the methods to test this assumption is Schoenfeld and scaled Schoenfeld residuals plots. [36] In Schoenfeld residuals test, a proportionality assumption is satisfied if each predictor residual plot is a horizontal line. As shown in Fig. 7, all energy factors and interactions satisfy this hypothesis.

The results of applying Cox regression on effective energy parameters and their interactions is indicated in Table 5 and Table 7.

Table 5: Energy factors that have significant impact on system's lifetime.

Variable	**Coefficient**	**Standard Error**	**P- Value**
Breeding energy	-1.58787	1.335942	0.079
Escaping energy	-21.0795	1.3335	0
Flocking energy	-145.4	3.795943	0
Hunting energy	-31.2682	2.856803	0
Seeking energy	-58.5227	2.885732	0

Table 6: All possible interactions among significant energy parameters.

	Variable	Interaction 1	Interaction 2	Interaction 3	Interaction 4
1	Breeding energy	Escaping energy	Flocking energy	Hunting energy	Seeking energy
2	Escaping energy	Flocking energy	Hunting energy	Seeking energy	
3	Flocking energy	Hunting energy	Seeking energy		
4	Hunting energy	Seeking energy			
5	Seeking energy	-			

Table 7: Interactions with significant impact on system's lifetime.

Variable	Coefficient	Standard Error	P- Value
Interaction between flocking and seeking energies	1.147137	0.077527	0.001
Interaction between flocking and hunting energies	0.682809	0.077631	0

As shown in Table 5 and Table 7, breeding, escaping, flocking, hunting and seeking energies have significantly negative effect on the system's lifetime whereas the interactions have significantly positive effects on it. This implies that as breeding, escaping, flocking, hunting and seeking energies increase, the lifetime decreases. On the other hand, increasing interactions leads to enhancement in the ecosystem's lifetime.

As Thompson et al. [65] indicate, in reality of life, birds use flocking behavior for better movement, more efficient food searching and social learning. Flocking motion also decrease predation risk as much as the flock size increases and our simulation results show that flocking and seeking energy and flocking and hunting energy interactions are most significant interactions which matches the Thompson et al. observations in the real life.

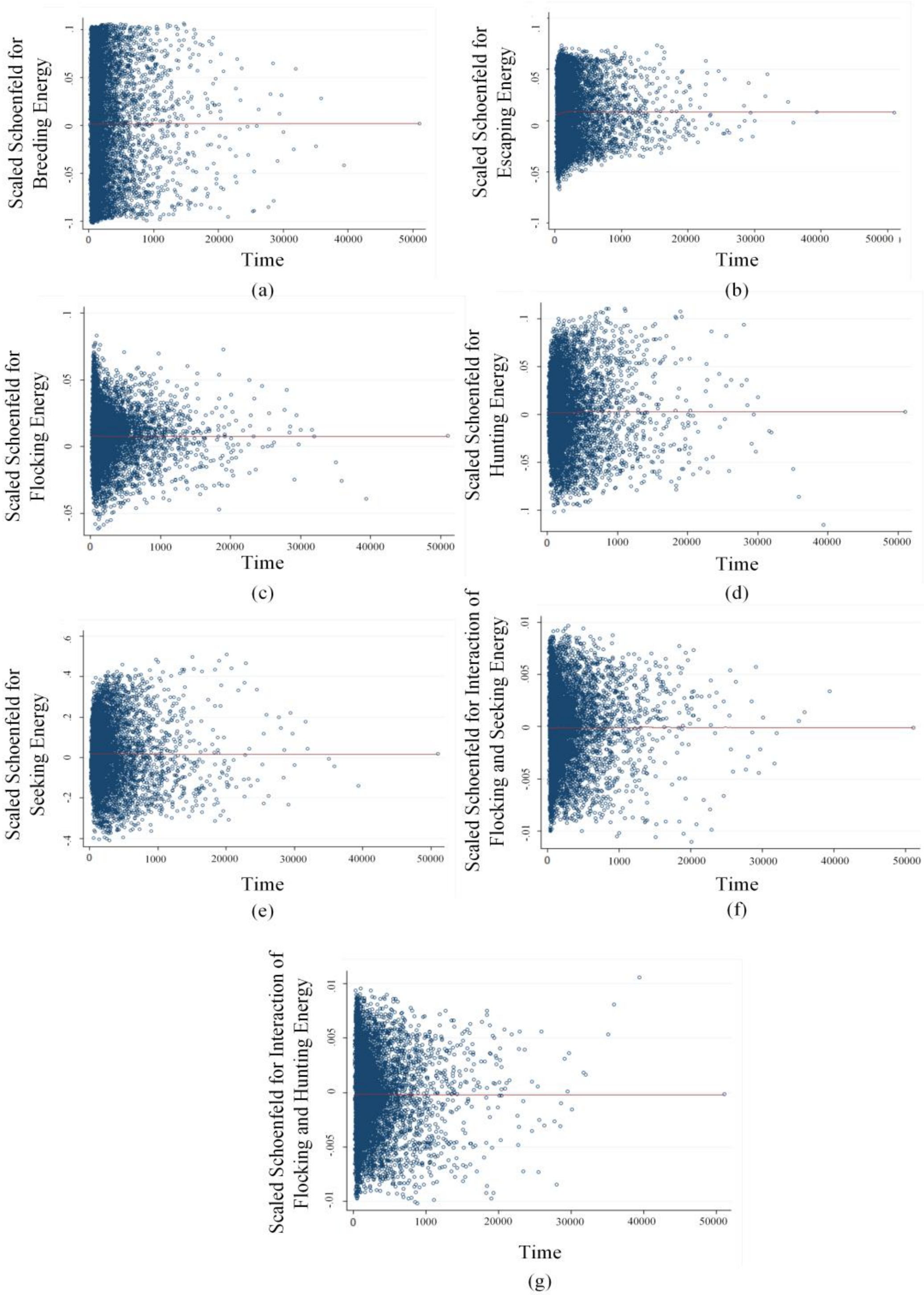


Figure 7: The proportionality assumption test is satisfied for all energy parameters since all residual plots are horizontal line. Here are the results for: (a) breeding energy (b) escaping energy (c) flocking energy (d) Hunting energy (e) predator's seeking energy (f) interaction of flocking energy with predator's seeking energy and (g) interaction of flocking energy with predator's hunting energy.

As shown in Table 5, breeding, escaping, flocking, hunting and seeking energies have significantly negative effects on the system lifetime while the interactions have significantly positive effects on the system's lifetime. This implies that as breeding, escaping, flocking, hunting and seeking energies increase, the system lifetime decreases. On the other hand, increasing interactions leads to enhancement in the system lifetime.

The graphical regression plots indicating the impact of five effective energy factors on system lifetime is illustrated in Figure 8. As shown in Table 5 all parameters have negative effect on systems' lifetime which implies that increasing energy consumption rate results in shorter lifetime. Also, the results suggest that flocking energy has the highest impact on sys-tem.

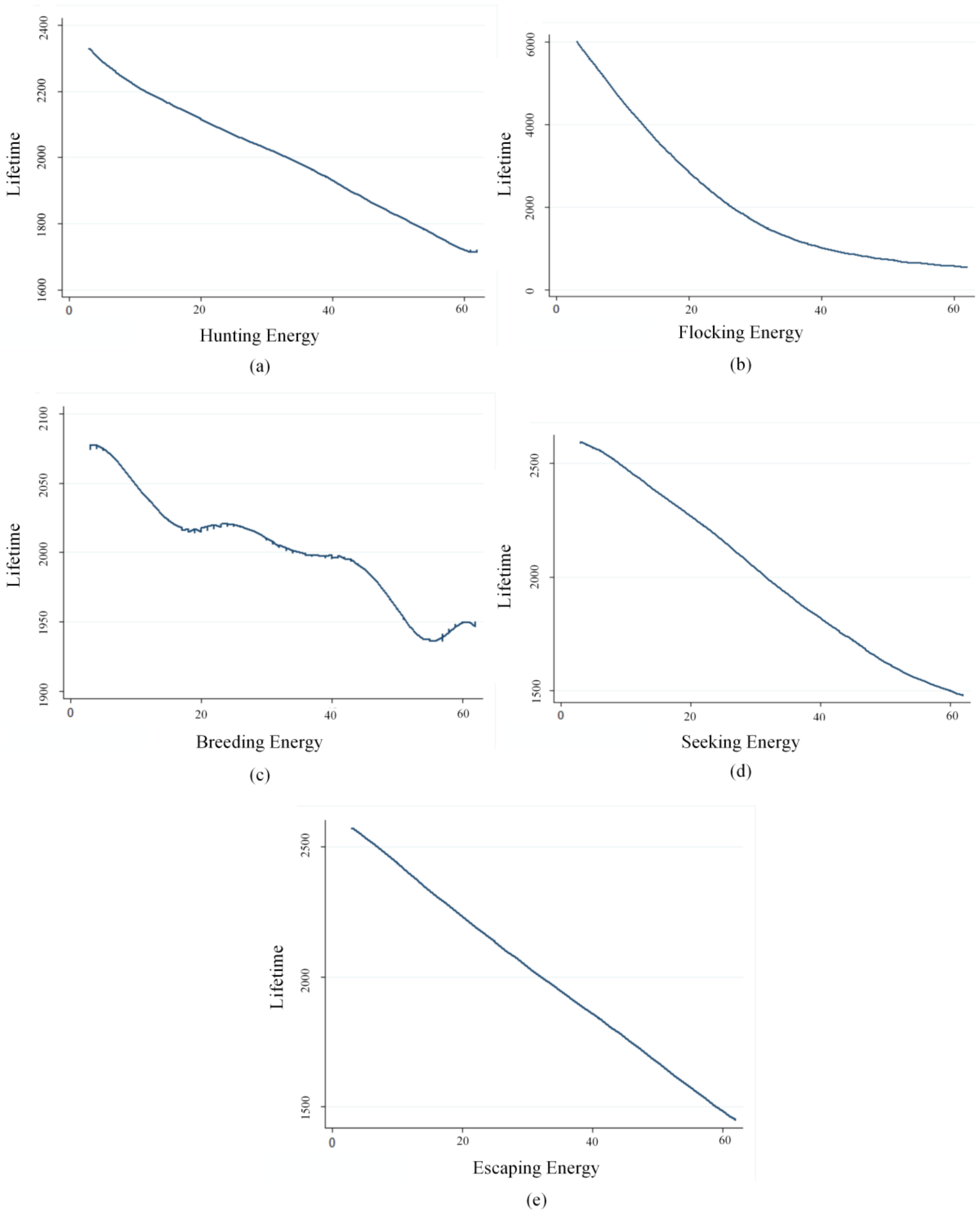


Figure 8: Impacts of energy factors on the ecosystem's life time. (a) Impact of hunting energy, (b) Impact of flocking energy,(c) Impact of breeding energy, (d) Impact of seeking energy, (e) Impact of escaping energy.

Utilizing the plot of cumulative probability of survival gives a better understanding of the effect of each energy parameter on the system. For instance, Figure 9 shows the cumulative probability of survival for two different levels of flocking energy given all other energy parameters are clamped. The red and blue curves illustrate survival function of two different flocking energies set as 50 and 10, respectively. As shown, higher level of flocking energy accelerates the extinction of ecosystem. The similar plots are for hunting and breeding energies are illustrated in Figure 10 and Figure 11 .

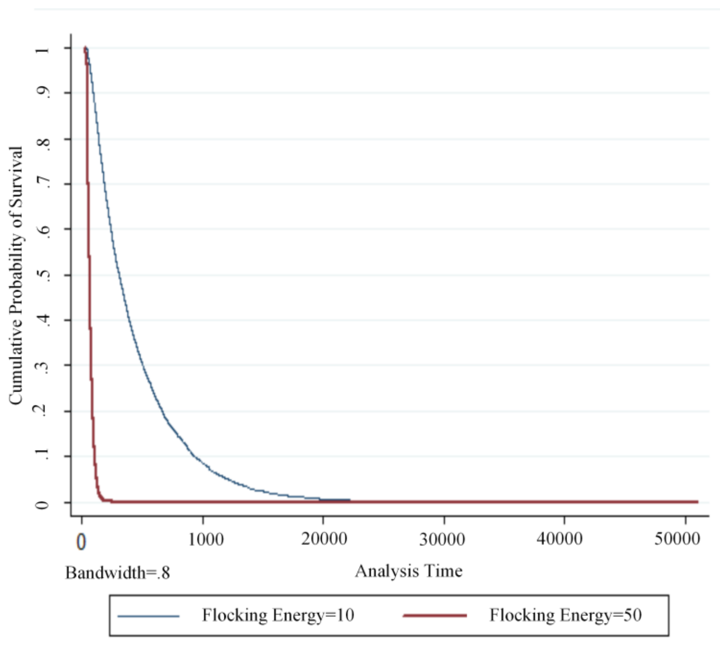


Figure 9: Cumulative probability plot of survival for flocking energy

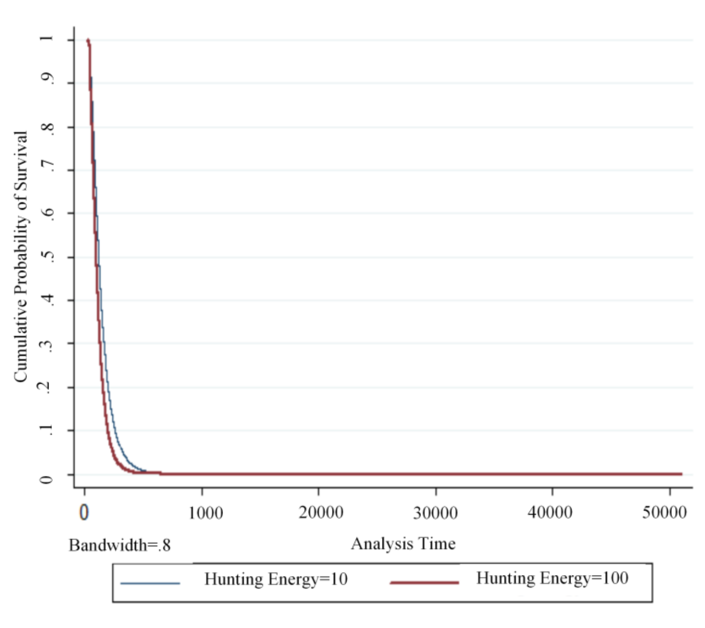


Figure 10: Cumulative probability plot of survival for hunting energy

In Figure 11 the survival function for two different levels of breeding energy is shown, when all other levels of energy the parameters are constant. The red and blue curves are survival function for breeding energy, respectively equal to 10 and 100. Graph shows that the higher level of breeding energy moves the system to extinction sooner. As before, Figure 11 also indicates that the breeding energy and system lifetime have an inverse relationship.

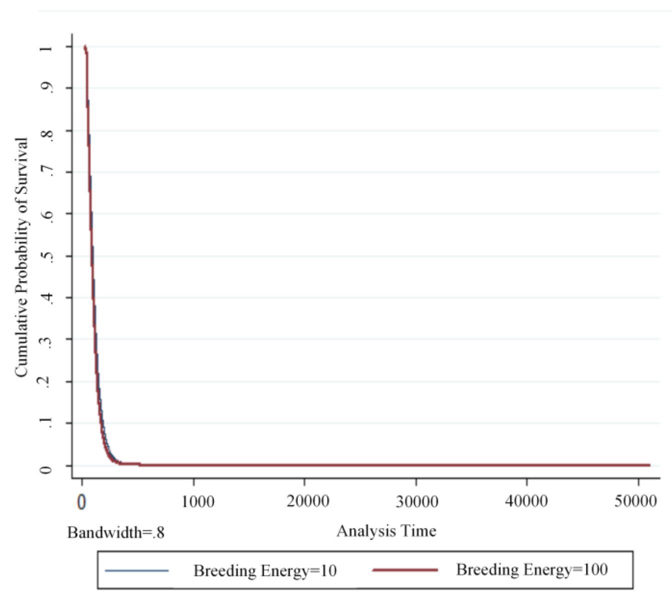


Figure 11: Cumulative probability plot of survival for breeding energy

Ultimately, in order to evaluate the goodness-of-fit (i.e. how well the model fits a set of observations) of the model, we employ the graph of Nelson-Aalen cumulative hazard function and the Cox- Snell. In this scheme, the Nelson-Aalen cumulative hazard function with slope of 45 degrees (i.e. equivalent to hazard rate of one) indicates that hazard function follows an exponential distribution whose hazard rate equals to one, which, in turn implies that the model fits the data well. As shown in Figure 12, the Nelson-Aalen residual for our model has an approximately 45-dgree slope which suggests that our final model fits the data very well.

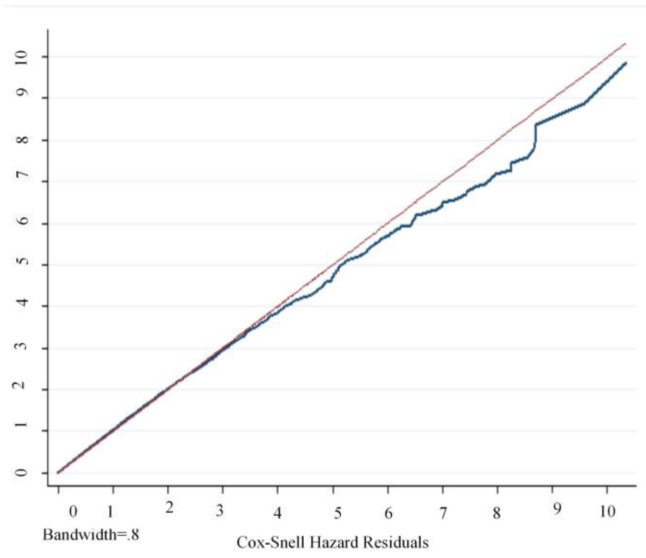


Figure 12: Graph of Nelson-Aalen cumulative hazard function and the Cox-Snell variable depicting Goodness-of-Fit for the model.

5 Summary and Conclusion

This study introduces a stochastic scheme for modeling a multi-species prey-predator artificial ecosystem with two levels of food chain to investigate the influence of energy flow on the ecosystem's lifetime. The proposed model consists of a stationary hosting environment with dynamic weather condition and fruit trees. The inhabitants of this ecosystem are herbivore and carnivore birds each consisting of a few species. In our model, the collective behavior emerges in terms of flocking, breeding, competing, resting, hunting, escaping, seeking and foraging behaviors. Using multi-species scheme, we define the ecosystem as a combination of prey and predator species with inter-competition among species within same level of food chain and intra-competition among those belonging to different levels of food chain. Further-more, we introduce some energy variables as functions of behaviors to model the energy within the ecosystem. Ultimately, we developed a simulation and visualization framework for implementing the proposed model. The experimental results of 11,000 simulations analyzed by Cox univariate analysis and hazard function suggest that out of eight aforementioned behaviors, only five corresponding energy variable influence the ecosystem's lifetime, significantly. Furthermore, the results of survival analysis show that out of all possible interactions among energy factors, only two of them including interaction between flocking and seeking energies, and interaction between flocking and hunting energies effect the lifetime most. The results of analyzing effect of each energy parameters and all possible interaction are indicated in appendix of this study. Additionally, software implementation of the proposed framework validates stability of simulation and visualization architecture. Ultimately, we validated our experimental results using

Nelson-Aalen cumulative hazard function and Cox-Snell variable, and scaled Schoenfeld residuals test.

5.1 Contributions

To the best of our knowledge, there are three contributions in this research:

As First, the high level of complexity in the structure of the proposed model in comparison with the other systems which mostly contains only one species of prey, one species of predator and a kind of resource. While this study introduces two species of prey, capability of competition among species, dynamic weather condition with two element of wind and rain and dynamic resources, various behavioral rules such as escaping, breeding, hunting, resting, etc.

Energy flow analysis within an artificial ecosystem is the second contribution. To the best of author's knowledge there is no similar comprehensive model in the previous literature that investigates the life span of a stochastic multi-species predator-prey artificial ecosystem based on energy flow using Survival Analysis method.

Lastly, the simulation results show that the flocking and seeking energy and flocking and hunting energy interactions are the most significant interactions which match with the Thompson et al. [65] observations in the real life. Their findings indicate that in real life birds use flocking behavior for better movement, more efficient food searching and social learning. Flocking motion also decrease predation risk as much as the flock size increases.

5.2 Future Works

As future works, we are planning to expand the research in three directions. First, we are planning to utilize evolutionary approach and reinforcement learning to provide the agents with required learning capabilities, so that they can autonomously learn behavioral rules that increase the lifetime and stability of the ecosystem based on survival of fittest paradigm and rewards from environment, respectively. Second, we will enhance the model by introducing more inhabitant types and increasing the levels of food chain. Moreover, in order to reach the steady state of the ecosystem with more inhabitants, we are planning to perform massive simulations and sensitivity analysis by using distributed processing paradigms. Third, we will make behaviors within system more realistic based on available empirical data and evaluating them on their own terms. Finally, we are planning to expand the visualization engine by providing an interactive interface as a serious-game based visualization tool in which not only user can tweak the parameters, but she can directly manipulate the visual environment in an interactive manner.

References

1. Abbott, R. (2011). From energy flows to purpose and back. In report of association for the advancement of artificial intelligence (AAAI) Fall Symposium: Complex Adaptive Systems. Arlington, Virginia: AAAI Press.
2. Allan, R. J. (2010). Survey of agent based modelling and simulation tools (pp. 1362-0207). Science & Technology Facilities Council.
3. Axelrod, R. (1998). The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration. Princeton University Press.
4. Bedau, M. A. (2003). Artificial life: organization, adaptation and complexity from the bottom up. Trends in cognitive sciences, 7(11), 505-512.
5. Ben-Shahar O., Shlomi Dolev, S., Dolgin A., & Sega M. (2014). Direction election in flocking swarms. Ad Hoc Networks. (12), 250–258.
6. Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). Swarm intelligence: from natural to artificial systems (No. 1). Oxford university press.
7. Bousquet and Page (2004): Multi-agent simulations and ecosystem management: a review, Ecological Modelling 176 (2004), 313-332.
8. Buhl, J., Sumpter, D. J., Couzin, I. D., Hale, J. J., Despland, E., Miller, E. R., & Simpson, S. J. (2006). From disorder to order in marching locusts. Science, 312(5778), 1402-1406.
9. Casti, J. L. (1997). Would-be worlds: How simulation is changing the frontiers of science. New York.
10. Chan, W. K. V., Son, Y. J., & Macal, C. M. (2010). Agent-based simulation tutorial-simulation of emergent behavior and differences between agent-based simulation and discrete-event simulation. In Proceedings of the Winter Simulation Conference. 135-150.
11. Collier, N. (2003). Repast: An extensible framework for agent simulation. The University of Chicago's Social Science Research. (36), 2003.
12. Cooper, T. F., & Ofria, C. (2003). Evolution of stable ecosystems in populations of digital organisms. In Artificial Life VIII: Proceedings of the Eighth International Conference on Artificial life. 227-232.

13. Couzin, I. D., Krause, J., James, R., Ruxton, G. D., & Franks, N. R. (2002). Collective memory and spatial sorting in animal groups. Journal of theoretical biology, 218(1), 1-11.

14. Cox, D . R . (1972). Regression Models and Life Tables (with Discussion). Journal of the Royal Statistical Society, Series B (34), 187- 220 .

15. Dorigo, M., Caro, G., & Gambardella, L. (1999). Ant algorithms for discrete optimization. Artificial life, 5(2), 137-172.

16. Dorin, A. (2005). A co-evolutionary epidemiological model for artificial life and death. In Advances in Artificial Life. 775-784. Springer Berlin Heidelberg.

17. Dorin, A., & Geard, N. (2014). The Practice of Agent-Based Model Visualization. Artificial Life, 20(2), 271-289.

18. Dorin, A., & Korb, K. B. (2007). Building virtual ecosystems from artificial chemistry. In Advances in Artificial Life. 103-112. Springer Berlin Heidelberg.

19. Dorin, A., Korb, K. B., & Grimm, V. (2008). Artificial-Life Ecosystems-What are they and what could they become?. In Artificial Life. 173-180.

20. Elsadany, A. E. A., El-Metwally, H. A., Elabbasy, E. M., & Agiza, H. N. (2012). Chaos and bifurcation of a nonlinear discrete predator-prey system. Computational Ecology and Software, 2(3), 169-180.

21. Gilbert, N., & Terna, P. (2000). How to build and use agent-based models in social science. Mind & Society, 1(1), 57-72.

22. Golestani, A., & Gras, R. (2010). Regularity analysis of an individual-based ecosystem simulation. Chaos: An Interdisciplinary Journal of Nonlinear Science, 20(4), 043120.

23. Gras, R., Devaurs, D., Wozniak, A., & Aspinall, A. (2009). An individual-based evolving predator-prey ecosystem simulation using a fuzzy cognitive map as the behavior model. Artificial life, 15(4), 423-463.

24. Gregoire, G., & Chate, H. (2004). Onset of collective and cohesive motion. Physical review letters, 92(2), 025702.

25. Hartman, C., & Benes, B. (2006). Autonomous boids. Computer Animation and Virtual Worlds, 17(34), 199-206.

26. Heleno, P., & Próspero dos Santos, M. (1998). Artificial animals in virtual ecosystems. Computer networks and ISDN systems, 30(20), 1923-1932.

27. Husbands, P., Harvey, I., Cliff, D., & Miller, G. (1997). Artificial evolution: a new path for artificial intelligence?. Brain and cognition, 34(1), 130-159.

28. Holling, C. S. (1959). The components of predation as revealed by a study of small-mammal predation of the European pine sawfly. The Canadian Entomologist, 91(05), 293-320.

29. Iba, H. (2013). Agent-based Modeling and Simulation with Swarm. Chapman & Hall/CRC. 139-145.

30. Kennedy, J., Kennedy, J. F., & Eberhart, R. C. (2001). Swarm intelligence. Morgan Kaufmann.

31. Kennedy, J.F., & Eberhart, R.C. (1995). Particle Swarm Optimization. In Proceedings of IEEE International Conference on Neural Networks. 1942-1948. Perth, WA: IEEE.

32. Kernbach, S., & Kernbach, O. (2011). Collective energy homeostasis in a large-scale microrobotic swarm. Robotics and Autonomous Systems, 59(12), 1090-1101.

33. Kubik A. (2003). Toward a formalization of emergence. Artif Life (9) 41-66.

34. Kunz, H., & Hemelrijk, C. K. (2003). Artificial fish schools: collective effects of school size, body size, and body form. Artificial life, 9(3), 237-253.

35. Lambert, P. C., & Royston, P. (2009). Further development of flexible parametric models for survival analysis. Stata Journal, 9(2), 265-290.

36. Lawless, J. F. (2011). Statistical models and methods for lifetime data (Vol. 362). John Wiley & Sons.

37. Levins, R. (1966). The strategy of model building in population biology. American scientist, 421-431.

38. Levy, P., & Bonomo, R. (1999). Collective intelligence: Mankind's emerging world in cyberspace. pp 13. Perseus Publishing.

39. Lotka, A. J. (1956). Elements of Physical Biology, 1924, republished as: Elements of Mathematical Biology. Dover Publishers, New York.

40. Macal, C. M., & North, M. J. (2010). Tutorial on agent-based modelling and simulation. Journal of Simulation, 4(3), 151-162.

41. MacArthur, R. H., & Connell, J. H. (1966). The biology of populations (p. 200). New York: Wiley.

42. Mascaro, S., Korb, K. B., & Nicholson, A. E. (2003). Alife investigation of parental investment in reproductive strategies. In ALIFE VIII-Proceedings of The 8th International Conference. 358-361.
43. May, R. M. (2001). Stability and complexity in model ecosystems (Vol. 6). Princeton University Press.
44. May, R. M. (1971). Stability in multispecies community models. Mathematical Biosciences, 12(1), 59-79.
45. Mitchell, M. (2006). Complex systems: Network thinking. Artificial Intelligence, 170(18), 1194-1212.
46. Mitchell, M., & Newman, M. (2002). Complex systems theory and evolution. Encyclopedia of Evolution, 1-5.
47. Nishimura, S., & Ikegami, T. (1997). Emergence of collective strategies in a predator-prey game model. Artificial Life, 3(4), 243-260.
48. North, M. J., Tatara, E., Collier, N. T., & Ozik, J. (2007). Visual agent-based model development with repast simphony. Tech. rep., Argonne National Laboratory.
49. Olfati-Saber, R. (2006). Flocking for multi-agent dynamic systems: Algorithms and theory. Automatic Control, IEEE Transactions on, 51(3), 401-420.
50. Parrish, J. K., Viscido, S. V., & Grünbaum, D. (2002). Self-organized fish schools: an examination of emergent properties. The biological bulletin, 202(3), 296-305.
51. Persson, I. (2002). Essays on the assumption of proportional hazards in Cox regression. Dissertation for the Degree of Doctor of Philosophy in Statistics. Acta Universitatis Upsaliensis.
52. Pitcher, T. J. (Ed.). (1993). Behaviour of teleost fishes (7). Springer.
53. Pratt, S. C. (2010), Collective Intelligence. Encyclopedia of Animal Behavior, Arizona State University, Tempe, AZ: Elsevier Ltd.
54. Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. ACM SIGGRAPH Computer Graphics, 21(4), 25-34.
55. Rodriguez, A., & Reggia, J. A. (2004). Extending self-organizing particle systems to problem solving. Artificial Life, 10(4), 379-395.

56. Ruas, T. L., Marietto, M. D. G. B., de Moraes, A. F., Batista, R. D. S. F., Heideker, A., & Aguiar, E. (2011). Modeling artificial life through multi-agent based simulation. Multi-Agent Systems-Modeling, Control, Programming, Simulations and Applications, Intech.
57. Schut, M. C. (2007). Scientific Handbook for Simulation of Collective Intelligence, version 2. Available under Creative Commons License.
58. Schut, M. C. (2010). On Model Design for Simulation of Collective Intelligence, Information Sciences,180, 132-155
59. Scogings, C. J., & Hawick K. A. (2008). Energy Flow and Conservation in an Artificial Life Agent Model. In Proceedings of International Conference on Artificial Intelligence (ICAI'08). 113-119. Las Vegas, USA: CSREA.
60. Silva, A. R. D., Lages, W. S., & Chaimowicz, L. (2009). Boids that see: Using self-occlusion for simulating large groups on gpus. Computers in Entertainment (CIE), 7(4), 51.
61. Singh, V. K., Gautam, D., Singh, R. R., & Gupta, A. K. (2009). Agent-based computational modeling of emergent collective intelligence. In Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems. 240-251. Springer Berlin Heidelberg.
62. Sklar, E. (2007). Netlogo, a multi-agent simulation environment. Artificial life, 13(3), 303-311.
63. Tanner, H. G., Jadbabaie, A., & Pappas, G. J. (2003). Stable flocking of mobile agents part i: dynamic topology. In Decision and Control, Proceedings. 42nd IEEE Conference on (2). 2016-2021. IEEE.
64. Tatara, E., North, M., Howe, T., Collier, N., & Vos, J. (2006). An indroduction to repast simphony modeling using a simple predator-prey example. In Proceedings of the Agent 2006 Conference on Social Agents: Results and Prospects.
65. Thompson, W. A., Vertinsky, I., & Krebs, J. R. (1974). The survival value of flocking in birds: a simulation model. The Journal of Animal Ecology, 785-820.
66. Vicsek, T., Czirók, A., Ben-Jacob, E., Cohen, I., & Shochet, O. (1995). Novel type of phase transition in a system of self-driven particles. Physical review letters, 75(6), 1226-1229.

67. Volterra, V. (1926). Fluctuations in the abundance of a species considered mathematically. Nature, 118, 558-560.

68. Ward, C. R., Gobet, F., & Kendall, G. (2001). Evolving collective behavior in an artificial ecology. Artificial life, 7(2), 191-209.

69. Weisberg, M., & Reisman, K. (2008). The Robust Volterra Principle*. Philosophy of science, 75(1), 106-131.

70. Werner, G. M. (1991). Evolution of Communication in Artificial Organisms, Artifial Life II. In Proceedings of the Second International Conference of Artificial Life. 659-687.

71. Woodberry, O., Korb, K. B., & Nicholson, A. E. (2005, December). The evolution of aging. In The Second Australian Conference on Artificial Life (ACAL 2005).

72. Yaeger, L. (1994). Poly world: Life in a new context. Proc. Artificial Life (3), 263-263.

73. Zhou, J., Wu, X., Yu, W., Small, M., & Lu, J. A. (2012). Flocking of multi-agent dynamical systems based on pseudo-leader mechanism. Systems & Control Letters, 61(1), 195-202.

Appendix

```
  ___  ____  ____  ____  ____ (R)
 /__    /   ____/   /   ____/
___/   /   /___/   /   /___/   12.0   Copyright 1985-2011 StataCorp LP
  Statistics/Data Analysis            StataCorp
                                      4905 Lakeway Drive
                                      College Station, Texas 77845 USA
                                      800-STATA-PC        http://www.stata.com
                                      979-696-4600        stata@stata.com
                                      979-696-4601 (fax)

Single-user Stata perpetual license:
       Serial number:  30120584967
         Licensed to:  Ali Asgari
                       University Of Ottawa

Notes:
      1.  New update available; type -update all-

 1 . do "/var/folders/2s/yknc1_h12f52lpypfg2n75cr0000gn/T//SD02705.000000"

 2 . insheet using "/Volumes/6132652041/sim/final.csv", clear
   (11 vars, 10991 obs)

 3 .
 4 . capture log close

 5 . capture log using ali.txt, replace text

 6 .
   end of do-file

 7 . do "/var/folders/2s/yknc1_h12f52lpypfg2n75cr0000gn/T//SD02705.000000"

 8 . **survival function
 9 . stset lifetime

     failure event:  (assumed to fail at time=lifetime)
obs. time interval:  (0, lifetime]
 exit on or before:  failure

------------------------------------------------------------------------------
    10991  total obs.
        0  exclusions
------------------------------------------------------------------------------
    10991  obs. remaining, representing
    10991  failures in single record/single failure data
 2.20e+07  total analysis time at risk, at risk from t =         0
                             earliest observed entry t =         0
                                  last observed exit t =  51040.36

10 . sts graph, na

         failure _d:  1 (meaning all fail)
```

```
      analysis time _t:  lifetime

 11 .
 12 . *to know which vars should be included in the model.....We should look at the p value of chi square test
 13 . *yes
 14 . stcox breed_en, nohr

         failure _d:  1 (meaning all fail)
   analysis time _t:  lifetime

Iteration 0:   log likelihood = -91284.048
Iteration 1:   log likelihood = -91282.503
Refining estimates:
Iteration 0:   log likelihood = -91282.503

Cox regression -- Breslow method for ties

No. of subjects =        10991                     Number of obs   =     10991
No. of failures =        10991
Time at risk    =  21968002.05
                                                   LR chi2(1)      =      3.09
Log likelihood  =   -91282.503                     Prob > chi2     =    0.0788
```

_t	Coef.	Std. Err.	z	P>|z|	[95% Conf. Interval]	
breed_en	.0009699	.0005518	1.76	0.079	-.0001117	.0020515

```
 15 .
 16 . *NO
 17 . stcox compete_en , nohr

         failure _d:  1 (meaning all fail)
   analysis time _t:  lifetime

Iteration 0:   log likelihood = -91284.048
Iteration 1:   log likelihood = -91283.656
Refining estimates:
Iteration 0:   log likelihood = -91283.656

Cox regression -- Breslow method for ties

No. of subjects =        10991                     Number of obs   =     10991
No. of failures =        10991
Time at risk    =  21968002.05
                                                   LR chi2(1)      =      0.78
Log likelihood  =   -91283.656                     Prob > chi2     =    0.3762
```

_t	Coef.	Std. Err.	z	P>|z|	[95% Conf. Interval]	
compete_en	.0004958	.0005603	0.88	0.376	-.0006023	.001594

```
18 .
19 . *YES
20 . stcox escape_en , nohr

         failure _d:  1 (meaning all fail)
   analysis time _t:  lifetime

Iteration 0:   log likelihood = -91284.048
Iteration 1:   log likelihood =  -91202.28
Iteration 2:   log likelihood = -91202.278
Refining estimates:
Iteration 0:   log likelihood = -91202.278

Cox regression -- Breslow method for ties

No. of subjects =        10991                  Number of obs    =     10991
No. of failures =        10991
Time at risk    =  21968002.05
                                                LR chi2(1)       =    163.54
Log likelihood  =   -91202.278                  Prob > chi2      =    0.0000

------------------------------------------------------------------------------
          _t |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
   escape_en |   .0071256   .0005569    12.79   0.000     .0060341    .0082171
------------------------------------------------------------------------------

21 .
22 . *NO
23 . stcox feed_pred_en , nohr

         failure _d:  1 (meaning all fail)
   analysis time _t:  lifetime

Iteration 0:   log likelihood = -91284.048
Iteration 1:   log likelihood = -91283.282
Refining estimates:
Iteration 0:   log likelihood = -91283.282

Cox regression -- Breslow method for ties

No. of subjects =        10991                  Number of obs    =     10991
No. of failures =        10991
Time at risk    =  21968002.05
                                                LR chi2(1)       =      1.53
Log likelihood  =   -91283.282                  Prob > chi2      =    0.2158

------------------------------------------------------------------------------
          _t |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
feed_pred_en |   .0006886   .0005564     1.24   0.216    -.0004019    .0017791
------------------------------------------------------------------------------
```

```
24 .
25 .
26 . *YES
27 . stcox flock_en, nohr

         failure _d:  1 (meaning all fail)
   analysis time _t:  lifetime

Iteration 0:   log likelihood = -91284.048
Iteration 1:   log likelihood = -87306.692
Iteration 2:   log likelihood = -87295.643
Iteration 3:   log likelihood = -87295.642
Refining estimates:
Iteration 0:   log likelihood = -87295.642

Cox regression -- Breslow method for ties

No. of subjects =        10991                 Number of obs    =     10991
No. of failures =        10991
Time at risk    =  21968002.05
                                               LR chi2(1)       =   7976.81
Log likelihood  =   -87295.642                 Prob > chi2      =    0.0000
```

_t	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
flock_en	.0636156	.0007335	86.73	0.000	.0621779	.0650532

```
28 .
29 . *YES
30 . stcox hunt_en, nohr

         failure _d:  1 (meaning all fail)
   analysis time _t:  lifetime

Iteration 0:   log likelihood = -91284.048
Iteration 1:   log likelihood = -91269.526
Iteration 2:   log likelihood = -91269.526
Refining estimates:
Iteration 0:   log likelihood = -91269.526

Cox regression -- Breslow method for ties

No. of subjects =        10991                 Number of obs    =     10991
No. of failures =        10991
Time at risk    =  21968002.05
                                               LR chi2(1)       =     29.04
Log likelihood  =   -91269.526                 Prob > chi2      =    0.0000
```

_t	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
hunt_en	.0029835	.0005535	5.39	0.000	.0018986	.0040685

```
31 .
32 .
33 . *NO
34 . stcox min_rest_en, nohr

         failure _d:  1 (meaning all fail)
   analysis time _t:  lifetime

Iteration 0:   log likelihood = -91284.048
Iteration 1:   log likelihood = -91283.515
Refining estimates:
Iteration 0:   log likelihood = -91283.515

Cox regression -- Breslow method for ties

No. of subjects =        10991                  Number of obs   =      10991
No. of failures =        10991
Time at risk    =  21968002.05
                                                LR chi2(1)      =       1.07
Log likelihood  =   -91283.515                  Prob > chi2     =     0.3021
```

_t	Coef.	Std. Err.	z	P>|z|	[95% Conf. Interval]	
min_rest_en	.0005637	.0005462	1.03	0.302	-.0005068	.0016343

```
35 .
36 . *YES
37 . stcox seek_pred_en, nohr

         failure _d:  1 (meaning all fail)
   analysis time _t:  lifetime

Iteration 0:   log likelihood = -91284.048
Iteration 1:   log likelihood =  -91175.78
Iteration 2:   log likelihood = -91175.772
Refining estimates:
Iteration 0:   log likelihood = -91175.772

Cox regression -- Breslow method for ties

No. of subjects =        10991                  Number of obs   =      10991
No. of failures =        10991
Time at risk    =  21968002.05
                                                LR chi2(1)      =     216.55
Log likelihood  =   -91175.772                  Prob > chi2     =     0.0000
```

_t	Coef.	Std. Err.	z	P>|z|	[95% Conf. Interval]	
seek_pred_en	.0082782	.0005618	14.73	0.000	.007177	.0093793

```
38 .
39 .
   end of do-file

40 . do "/var/folders/2s/yknc1_h12f52lpypfg2n75cr0000gn/T//SD02705.000000"

41 .
42 . stcox breed_en escape_en flock_en hunt_en seek_pred_en , nohr

         failure _d:  1 (meaning all fail)
   analysis time _t:  lifetime

Iteration 0:   log likelihood = -91284.048
Iteration 1:   log likelihood = -86658.652
Iteration 2:   log likelihood = -86651.275
Iteration 3:   log likelihood = -86651.275
Refining estimates:
Iteration 0:   log likelihood = -86651.275

Cox regression -- Breslow method for ties

No. of subjects =        10991                  Number of obs   =     10991
No. of failures =        10991
Time at risk    =  21968002.05
                                                LR chi2(5)      =   9265.55
Log likelihood  =   -86651.275                  Prob > chi2     =    0.0000
```

_t	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
breed_en	.0014602	.0005543	2.63	0.079	.0003739	.0025466
escape_en	.0147596	.0005632	26.21	0.000	.0136558	.0158634
flock_en	.0689399	.0007635	90.30	0.000	.0674435	.0704362
hunt_en	.0034723	.0005535	6.27	0.000	.0023875	.0045572
seek_pred_en	.0142068	.0005659	25.11	0.000	.0130977	.0153158

```
43 .
44 . * Now Interactions
45 .
46 . *NO
47 . stcox breed_en escape_en flock_en hunt_en seek_pred_en c.breed_en#c.escape_en, nohr

         failure _d:  1 (meaning all fail)
   analysis time _t:  lifetime

Iteration 0:   log likelihood = -91284.048
Iteration 1:   log likelihood = -86658.795
Iteration 2:   log likelihood = -86651.249
Iteration 3:   log likelihood = -86651.249
Refining estimates:
Iteration 0:   log likelihood = -86651.249
```

```
Cox regression -- Breslow method for ties

No. of subjects =        10991                  Number of obs   =      10991
No. of failures =        10991
Time at risk    =  21968002.05
                                                LR chi2(6)      =    9265.60
Log likelihood  =   -86651.249                  Prob > chi2     =     0.0000
```

_t	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
breed_en	.0016957	.0011783	1.44	0.150	-.0006136	.0040051
escape_en	.0149946	.0011806	12.70	0.000	.0126808	.0173084
flock_en	.0689398	.0007634	90.30	0.000	.0674435	.0704361
hunt_en	.0034713	.0005535	6.27	0.000	.0023864	.0045562
seek_pred_en	.0142063	.0005659	25.10	0.000	.0130972	.0153154
c.breed_en#c.escape_en	-7.22e-06	.0000319	-0.23	0.821	-.0000697	.0000553

```
48 .
49 . *NO
50 . stcox breed_en escape_en flock_en hunt_en seek_pred_en c.breed_en#c.flock_en, nohr

         failure _d:  1 (meaning all fail)
   analysis time _t:  lifetime

Iteration 0:   log likelihood = -91284.048
Iteration 1:   log likelihood = -86661.848
Iteration 2:   log likelihood = -86651.015
Iteration 3:   log likelihood = -86651.014
Refining estimates:
Iteration 0:   log likelihood = -86651.014

Cox regression -- Breslow method for ties

No. of subjects =        10991                  Number of obs   =      10991
No. of failures =        10991
Time at risk    =  21968002.05
                                                LR chi2(6)      =    9266.07
Log likelihood  =   -86651.014                  Prob > chi2     =     0.0000
```

_t	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
breed_en	.0021876	.001149	1.90	0.057	-.0000643	.0044396
escape_en	.0147617	.0005632	26.21	0.000	.0136579	.0158655
flock_en	.0696835	.0012811	54.39	0.000	.0671726	.0721944
hunt_en	.0034692	.0005535	6.27	0.000	.0023844	.004554
seek_pred_en	.0142087	.0005659	25.11	0.000	.0130995	.0153178
c.breed_en#c.flock_en	-.0000225	.0000311	-0.72	0.470	-.0000834	.0000385

```
51 .
52 . *NO
53 . stcox breed_en escape_en flock_en hunt_en seek_pred_en c.breed_en#c.hunt_en, nohr

         failure _d:  1 (meaning all fail)
   analysis time _t:  lifetime

Iteration 0:   log likelihood = -91284.048
Iteration 1:   log likelihood = -86658.113
Iteration 2:   log likelihood = -86650.697
Iteration 3:   log likelihood = -86650.697
Refining estimates:
Iteration 0:   log likelihood = -86650.697

Cox regression -- Breslow method for ties

No. of subjects =        10991                  Number of obs    =     10991
No. of failures =        10991
Time at risk    =  21968002.05
                                                LR chi2(6)       =   9266.70
Log likelihood  =   -86650.697                  Prob > chi2      =    0.0000
```

_t	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
breed_en	.0025758	.0011766	2.19	0.029	.0002696	.0048819
escape_en	.0147559	.0005632	26.20	0.000	.0136521	.0158597
flock_en	.0689401	.0007635	90.30	0.000	.0674437	.0704365
hunt_en	.0045787	.0011687	3.92	0.000	.0022881	.0068693
seek_pred_en	.0142195	.000566	25.12	0.000	.0131101	.0153289
c.breed_en#c.hunt_en	-.0000343	.0000319	-1.07	0.282	-.0000968	.0000282

```
54 .
55 . *NO
56 . stcox breed_en escape_en flock_en hunt_en seek_pred_en c.breed_en#c.seek_pred_en, nohr

         failure _d:  1 (meaning all fail)
   analysis time _t:  lifetime

Iteration 0:   log likelihood = -91284.048
Iteration 1:   log likelihood = -86658.931
Iteration 2:   log likelihood = -86651.053
Iteration 3:   log likelihood = -86651.053
Refining estimates:
Iteration 0:   log likelihood = -86651.053

Cox regression -- Breslow method for ties

No. of subjects =        10991                  Number of obs    =     10991
No. of failures =        10991
Time at risk    =  21968002.05
```

```
                                                 LR chi2(6)      =    9265.99
Log likelihood  =   -86651.053                   Prob > chi2     =     0.0000
```

_t	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
breed_en	.0021591	.0011872	1.82	0.069	-.0001678	.0044859
escape_en	.0147593	.0005631	26.21	0.000	.0136556	.015863
flock_en	.0689446	.0007635	90.30	0.000	.0674481	.0704411
hunt_en	.0034796	.0005536	6.29	0.000	.0023945	.0045647
seek_pred_en	.0149086	.0011966	12.46	0.000	.0125633	.017254
c.breed_en#c.seek_pred_en	-.0000215	.0000323	-0.67	0.506	-.0000848	.0000418

```
57 .
58 . *NO
59 . stcox breed_en escape_en flock_en hunt_en seek_pred_en c.flock_en#c.escape_en, nohr

         failure _d:  1 (meaning all fail)
   analysis time _t:  lifetime

Iteration 0:   log likelihood = -91284.048
Iteration 1:   log likelihood = -86819.584
Iteration 2:   log likelihood = -86651.563
Iteration 3:   log likelihood = -86651.241
Refining estimates:
Iteration 0:   log likelihood = -86651.241

Cox regression -- Breslow method for ties

No. of subjects =        10991                   Number of obs   =      10991
No. of failures =        10991
Time at risk    =  21968002.05
                                                 LR chi2(6)      =    9265.61
Log likelihood  =   -86651.241                   Prob > chi2     =     0.0000
```

_t	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
breed_en	.0014605	.0005543	2.64	0.008	.0003742	.0025469
escape_en	.0150287	.0011776	12.76	0.000	.0127206	.0173368
flock_en	.0692107	.0012906	53.62	0.000	.066681	.0717403
hunt_en	.0034766	.0005537	6.28	0.000	.0023913	.0045618
seek_pred_en	.014215	.0005668	25.08	0.000	.0131041	.0153258
c.flock_en#c.escape_en	-8.09e-06	.0000311	-0.26	0.795	-.000069	.0000528

```
60 .
61 . *NO
62 . stcox breed_en escape_en flock_en hunt_en seek_pred_en c.hunt_en#c.escape_en, nohr

         failure _d:  1 (meaning all fail)
```

```
   analysis time _t:  lifetime

Iteration 0:   log likelihood = -91284.048
Iteration 1:   log likelihood = -86659.174
Iteration 2:   log likelihood = -86651.261
Iteration 3:   log likelihood =  -86651.26
Refining estimates:
Iteration 0:   log likelihood =  -86651.26

Cox regression -- Breslow method for ties

No. of subjects =        10991                  Number of obs   =      10991
No. of failures =        10991
Time at risk    =  21968002.05
                                                LR chi2(6)      =    9265.57
Log likelihood  =    -86651.26                  Prob > chi2     =     0.0000
```

_t	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
breed_en	.0014611	.0005543	2.64	0.008	.0003747	.0025474
escape_en	.0145857	.0011661	12.51	0.000	.0123002	.0168713
flock_en	.0689378	.0007635	90.29	0.000	.0674413	.0704343
hunt_en	.0032949	.0011796	2.79	0.005	.0009828	.005607
seek_pred_en	.014207	.0005659	25.11	0.000	.013098	.0153161
c.hunt_en#c.escape_en	5.38e-06	.0000316	0.17	0.865	-.0000565	.0000673

```
63 .
64 . *NO
65 . stcox breed_en escape_en flock_en hunt_en seek_pred_en c.seek_pred_en#c.escape_en, nohr

         failure _d:  1 (meaning all fail)
   analysis time _t:  lifetime

Iteration 0:   log likelihood = -91284.048
Iteration 1:   log likelihood = -86660.985
Iteration 2:   log likelihood = -86651.124
Iteration 3:   log likelihood = -86651.121
Refining estimates:
Iteration 0:   log likelihood = -86651.121

Cox regression -- Breslow method for ties

No. of subjects =        10991                  Number of obs   =      10991
No. of failures =        10991
Time at risk    =  21968002.05
                                                LR chi2(6)      =    9265.85
Log likelihood  =   -86651.121                  Prob > chi2     =     0.0000
```

_t	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]

breed_en	.0014617	.0005543	2.64	0.008	.0003753	.0025481
escape_en	.0141865	.0011783	12.04	0.000	.011877	.016496
flock_en	.0689239	.0007639	90.22	0.000	.0674267	.0704212
hunt_en	.0034733	.0005535	6.27	0.000	.0023884	.0045582
seek_pred_en	.0136352	.0011772	11.58	0.000	.0113278	.0159425
c.seek_pred_en#c.escape_en	.0000176	.0000318	0.55	0.580	-.0000447	.0000799

```
66 .
67 . *NO
68 . stcox breed_en escape_en flock_en hunt_en seek_pred_en c.escape_en#c.escape_en, nohr

         failure _d:  1 (meaning all fail)
   analysis time _t:  lifetime

Iteration 0:   log likelihood = -91284.048
Iteration 1:   log likelihood = -86659.575
Iteration 2:   log likelihood = -86650.401
Iteration 3:   log likelihood = -86650.399
Refining estimates:
Iteration 0:   log likelihood = -86650.399

Cox regression -- Breslow method for ties

No. of subjects =        10991                  Number of obs   =     10991
No. of failures =        10991
Time at risk    =  21968002.05
                                                LR chi2(6)      =   9267.30
Log likelihood  =   -86650.399                  Prob > chi2     =    0.0000


69 .
70 . *YES
71 . stcox breed_en escape_en flock_en hunt_en seek_pred_en c.flock_en#c.hunt_en, nohr

         failure _d:  1 (meaning all fail)
   analysis time _t:  lifetime

Iteration 0:   log likelihood = -91284.048
Iteration 1:   log likelihood = -86652.211
Iteration 2:   log likelihood = -86637.763
Iteration 3:   log likelihood = -86637.761
```

```
Refining estimates:
Iteration 0:   log likelihood = -86637.761

Cox regression -- Breslow method for ties

No. of subjects =        10991                   Number of obs   =     10991
No. of failures =        10991
Time at risk    =  21968002.05
                                                 LR chi2(6)      =   9292.57
Log likelihood  =   -86637.761                   Prob > chi2     =    0.0000
```

_t	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
breed_en	.0014372	.0005543	2.59	0.010	.0003508	.0025236
escape_en	.014858	.0005634	26.37	0.000	.0137538	.0159622
flock_en	.0743883	.0012949	57.45	0.000	.0718504	.0769262
hunt_en	.0087676	.0011582	7.57	0.000	.0064976	.0110376
seek_pred_en	.0141855	.0005656	25.08	0.000	.013077	.0152941
c.flock_en#c.hunt_en	-.0001632	.0000314	-5.20	0.000	-.0002247	-.0001017

```
72 .
73 . *YES
74 . stcox breed_en escape_en flock_en hunt_en seek_pred_en c.flock_en#c.seek_pred_en, nohr

         failure _d:  1 (meaning all fail)
   analysis time _t:  lifetime

Iteration 0:   log likelihood = -91284.048
Iteration 1:   log likelihood = -86824.291
Iteration 2:   log likelihood = -86646.374
Iteration 3:   log likelihood = -86645.968
Refining estimates:
Iteration 0:   log likelihood = -86645.968

Cox regression -- Breslow method for ties

No. of subjects =        10991                   Number of obs   =     10991
No. of failures =        10991
Time at risk    =  21968002.05
                                                 LR chi2(6)      =   9276.16
Log likelihood  =   -86645.968                   Prob > chi2     =    0.0000
```

_t	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
breed_en	.0014643	.0005542	2.64	0.008	.0003782	.0025505
escape_en	.0148749	.0005643	26.36	0.000	.013769	.0159809
flock_en	.0724284	.001314	55.12	0.000	.069853	.0750038
hunt_en	.0034519	.0005535	6.24	0.000	.002367	.0045367
seek_pred_en	.0175893	.0011813	14.89	0.000	.015274	.0199046

c.flock_en#c.seek_pred_en	-.0001022	.0000313	-3.26	0.001	-.0001636	-.0000408

```
75 .
76 . *NO
77 . stcox breed_en escape_en flock_en hunt_en seek_pred_en c.hunt_en#c.seek_pred_en, nohr

         failure _d:  1 (meaning all fail)
   analysis time _t:  lifetime

Iteration 0:   log likelihood = -91284.048
Iteration 1:   log likelihood = -86657.429
Iteration 2:   log likelihood = -86649.694
Iteration 3:   log likelihood = -86649.694
Refining estimates:
Iteration 0:   log likelihood = -86649.694

Cox regression -- Breslow method for ties

No. of subjects =        10991                  Number of obs   =      10991
No. of failures =        10991
Time at risk    =  21968002.05
                                                LR chi2(6)      =    9268.71
Log likelihood  =   -86649.694                  Prob > chi2     =     0.0000
```

_t	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
breed_en	.0014796	.0005544	2.67	0.008	.000393	.0025661
escape_en	.0147608	.0005632	26.21	0.000	.013657	.0158647
flock_en	.0689335	.0007635	90.29	0.000	.0674372	.0704299
hunt_en	.0053237	.0011792	4.51	0.000	.0030125	.0076348
seek_pred_en	.0160698	.0011908	13.50	0.000	.013736	.0184037
c.hunt_en#c.seek_pred_en	-.000057	.000032	-1.78	0.475	-.0001198	5.83e-06

```
78 .
79 . *FINAL
80 . stcox breed_en escape_en flock_en hunt_en seek_pred_en c.flock_en#c.seek_pred_en c.flock_en#c.hunt_en , nol

         failure _d:  1 (meaning all fail)
   analysis time _t:  lifetime

Iteration 0:   log likelihood = -91284.048
Iteration 1:   log likelihood = -86817.507
Iteration 2:   log likelihood = -86633.032
Iteration 3:   log likelihood = -86632.636
Refining estimates:
Iteration 0:   log likelihood = -86632.636

Cox regression -- Breslow method for ties

No. of subjects =        10991                  Number of obs   =      10991
```

```
No. of failures =        10991
Time at risk    =  21968002.05
                                                  LR chi2(7)      =    9302.82
Log likelihood  =   -86632.636                    Prob > chi2     =     0.0000
```

_t	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
breed_en	.0014395	.0005542	2.60	0.009	.0003533	.0025258
escape_en	.0149708	.0005645	26.52	0.000	.0138645	.0160772
flock_en	.0777739	.0016701	46.57	0.000	.0745006	.0810471
hunt_en	.0087077	.0011576	7.52	0.000	.0064389	.0109766
seek_pred_en	.0175066	.0011804	14.83	0.000	.0151932	.0198201
c.flock_en#c.seek_pred_en	-.0001003	.0000313	-3.20	0.001	-.0001617	-.000039
c.flock_en#c.hunt_en	-.000162	.0000313	-5.17	0.000	-.0002234	-.0001005

```
81 . estimate store m1

82 .
83 .
   end of do-file
```