Matthew Schieber, mschieber3

CSE/ECE 6730 Project 1: Predator and Prey Populations With Variable Grass Species

Project Description:

This project attempts to simulate the behavior of predator and prey populations to determine which species of grass seed to plant a field with. The customer of the simulation is a conservation agency which plans to cultivate a field of prey (rabbits) and predators (wolves). The agency needs to maintain a healthy field of both rabbits and wolves, but they are not sure which grass seed to plant in the field. They do know that the population behaviors will vary depending on which grass seed was planted. Each grass seed produces grass which absorbs light and fixes carbon at varying rates, meaning each grass will grow at different rates. Also, certain grass seeds are more expensive than others, so they need to optimize quality of the resulting populations per cost. This simulation project attempts to answer the question.

Conceptual Model:

The system under investigation (SUI) is perhaps most intuitively modelled under the cellular automata paradigm, however, I will formulate the conceptual model using the discrete event simulation (DES) model. The first consideration is to determine the granularity of behaviors to simulate. For example, if I adopt an approach such as the Lotka-Volterra equations (ref 1.), then I would merely evaluate behavior at the granularity of entire populations. However, I believe an event-driven simulator for this SUI necessitates finer-grained behaviors. For example, the activities at the level of individual organisms such as rabbits eating grass and wolves eating rabbits. If done correctly, modelling behaviour at this granularity should reproduce the correct population behaviors. Moreover, while individual organism entities will be used, it will also be necessary to evaluate some behaviors at the level of populations. For example, breeding and repopulating might be better evaluated at the population level. Therefore, I propose the following entities and their associated activities:

Table 1: Conceptual Model Entities

Entity	Type	Activities
Grass	Resource Unary	Grow
Rabbit	Consumer/Resource Class	Eat Grass, Die
Rabbit Population	Group Unary	Repopulate
Wolf	Consumer Class	Hunt Rabbit, Eat Rabbit, Die
Wolf Population	Group Unary	Repopulate

The “Grass” entity will keep track of how much grass is available in the field and provide services to the “Rabbit” entity. The grass count is kept as a state variable N_Grass and is initiated by the size of the field. The grass is constantly in the growing activity, in the same way as grass in a field. Each time the grass grows, the grass count is incremented. The growth rate of the grass will be dictated by the model input, g_i, which is the growth rate for a particular seed of grass. For the purpose of this investigation, I will assume that the grass grows strictly to the g_i parameter, without fluctuation. Also, when rabbits eat grass, the grass count is decremented.

The "Rabbit" entity will consume grass and provide services to the "Wolf" entity. Rabbits will constantly be in the eating activity while they are still alive. The eating activity will last a duration of "R_Starvation", which is a parameter dictating how long the rabbit will live until starving. The actual time for the eating activity will be drawn from a normal distribution with mean R_Starvation and a unit standard deviation. If there is no more food available in the field (i.e. N_Grass = 0), the rabbit will invoke the "Die" activity.

The "Wolf" entity will consume Rabbits. Wolves will constantly be in the hunting activity while they are still alive. The hunting activity will last a duration of "W_Starvation", which is a parameter dictating how long the wolf will live until starving. The actual time for the hunting activity will be drawn from a normal distribution with mean W_Starvation and a unit standard deviation. Since rabbits are fast, wolves are not always successful when hunting. Hunting is carried out as follows: first, the wolves get a fraction of the current rabbit population to hunt. This fraction will be a model parameter, RabbitsToHunt. After (RabbitsToHunt * N_Rabbits) rabbits are selected for hunting, the wolf will have a probability of success for each rabbit. If the wolf is successful, then the hunt is over, the Eat_Rabbit event is invoked, and another Hunt_Rabbit event is scheduled W_Starvation time in the future. If a wolf is not successful for any rabbit, then the wolf will starve and the Death event if invoked. This way, the more rabbits available in the field, the higher the probability for a wolf to eat a rabbit.

The population entities for the wolves and rabbits behave in similar ways. The main responsibility for these entities will be to carry out the "Repopulate" event. During the Repopulate event, a fraction of the current respective population, dictated by "RabbitBreedingRate" and "WolfBreedingRate", respectively, will be generated and added to the population. This is a major feature of the SUI. If too many rabbits are allowed to reproduce, then the field will be overeaten, which can result in mass death for the rabbits and subsequently the wolves as well. Moreover, if the rabbit population becomes too sparse, then the wolf entity will become less successful hunting and could go extinct. Typically, there will be flux in the populations according to the above phenomenon.

Inputs/Outputs:

Input - model parameter , g_i , dictating the growth rate of grass seed species

Output - Populations over time. Final populations.

Assumptions:

- Rabbits and wolves are always fully grown
- Sunshine and rain are plentiful in the field, allowing for the designated growth rate.
- Breeding is a population behavior

Appendix

ABCMod Tables

Constants:

Name	Description	Value
InitialRabbitCount	Initial number of rabbits.	30
InitialWolfCount	Initial number of wolves.	4
R_Starvation	Days until a rabbit will starve.	2
W_Starvation	Days until a wolf will starve.	4
Rabbit_breeding	Fraction of population reproduced	0.2
Wolf_breeding	Fraction of population reproduced	0.05
Rabbits_to_hunt	Fraction of population for hunting	0.1
Wolf_catching_rabbit	Chance a wolf catching rabbit	0.25
Simulation_length	Number of days to simulate	TBD
Field_size	Size of simulation field	TBD

Parameters:

Name	Description	Value
g_i	Growth factor for grass	TBD - parameter

Entities:

Resource Unary: Grass	
Represents grass in the simulation field	
Attributes	**Description**
N_Grass	Amount of grass still in the field
g_i	Parameter, growth rate of grass Tuple - first element: units, second: per day
FieldSize	Size of field

<u>Resource/Consumer Class: Rabbit</u>	
Represents one rabbit	
Attributes	**Description**
R_Starvation	The number of days a rabbit goes before it eats food.

<u>Group Unary: Rabbit Population</u>	
Represents entire population of rabbits	
Attributes	**Description**
N_Rabbits	Number of rabbits in the field.
RabbitBreedingRate	Fraction of population reproduced

<u>Consumer Class: Wolf</u>	
Represents one wolf	
Attributes	**Description**
R_Starvation	The number of days a wolf can go without food.
RabbitsToHunt	Fraction of population for hunting
WolfCatchingRabbitRate	Chance a wolf catching rabbit

<u>Group Unary: Wolf Population</u>	
Represents entire population of wolves	
Attributes	**Description**
N_Wolves	Number of wolves in the field.
WolfBreedingRate	Fraction of population reproduced

<u>Entity Behaviors:</u>

<u>Field:</u>

Activity: Grow Grass	
Grows the grass in the field	
Precondition	none
TimeSequence	Occurs every G[1] days
Duration	G[1]
Terminating event	All cells of Grass are incremented by G[0]

Rabbits:

Activity: Eat Grass	
Simulates Rabbit eating grass	
Precondition	If Grass_count != 0 else die
Initiating event	Rabbit is alive
Duration	R_Starvation
Terminating Event	grass_count -= 1, schedule next Eat Grass activity

Activity: Die	
Rabbit dies	
Precondition	None
Causal	Wolf eats this rabbit or stavation
Duration	0
Terminating Event	N_Rabbits -= 1, remove this rabbit from simulation

Wolves

Activity: Hunt Rabbit	
Simulates wolf hunting a rabbit	
Precondition	None
Initiating event	Wolf is alive

Duration	W_Starvation
Terminating Event	If successful Eat_Rabbit and schedule next hunting activity, else Die

Activity: Eat Rabbit	
Simulates wolf hunting a rabbit	
Causal	
Initiating event	Successful hunt
Duration	0
Terminating Event	Selected rabbit dies

Activity: Die	
Wolf dies	
Precondition	None
Causal	Wolf was unsuccessful hunting
Duration	0
Terminating Event	N_Wolves -= 1, remove this Wolf from simulation