SIMULATING AND VISUALIZING COOPERATIVE HUNTING

Quach Phu Thanh Ha, Xuan Khang Nguyen, Le Anh Quan Nguyen
Ngoc Vinh Nguyen, and Hoang Viet Pham

Frankfurt University of Applied Sciences

Keywords: Environmental modelling, Collective Intelligence, Simulation, Hunting Strategies, Intelligent Agents, Predator/Prey Models

1 Introduction

1.1 General Topic

Every living form needs some kind of nutrition, and while some obtain it through "peaceful" ways, e.g. photosynthesis or from plants, others have adapted to more feral techniques. One such example is hunting, which is the process of pursuing and killing other live forms. The rich diversity of animals that have evolved over millions of years has also introduced a rich diversity of hunting techniques, each with its pros and cons. An especially interesting strategy is the cooperative hunting, which is related to the evolution of sociability between animals. It is a phenomenon where individuals work/hunt together to capture more prey at a lower over- all risk/cost. Moreover, solitary hunting may even be impossible, effectively forcing predators to cooperate in order to survive. Lions, wolves and humans are typical examples. Cooperative hunting is a complex behaviour as it requires social skills and coordination within the hunting party. For instance often the prey is much faster than the predators, who must employ cornering and surrounding approaches to counter this advantage. In other cases the prey is simply too strong for the individual predator, and thus a larger group is needed to bring it down.

A predator would generally stick to a single hunting strategy. Some, however, are versatile and adapt to the circumstances. After all, what counts in the end is surviving. Therefore, a generally solitary hunter may cooperate with others to kill prey otherwise too strong for him in order to survive, and a group hunter may see the opportunity of an easy solitary hunt and not inform the group, thus keeping everything for himself.

1.2 Similar problems in practice

- Agent-Based Simulation of Cooperative Hunting with UAVs

2 Team Work

2.1 Who What Does

- Quach Phu Thanh Ha
- Xuan Khang Nguyen
- Le Anh Quan Nguyen
- Ngoc Vinh Nguyen
- Hoang Viet Pham

2.2 Work Structure

- communication
- decisions
- bug tracking
- repository
- engineering

2.3 Ideas

3 Problem Description

3.1 Formal Description

The goal of this project is to simulate and visualize the various hunting strategies as well as manoeuvres used by predators to corner and surround their prey. For this purpose an application has to be programmed in Java consisting of an animation window displaying the simulation's state, a statistics window displaying information about the simulation (see 2.2), as well as at window containing the graphical user interface to operate the simulation and feed it the input parameters (see 2.1). The animation can be a simple 2D Cellular automaton l;3l. where the environment is represented by a grid and each object in it by a Coloured position or group of tiles (see Fig. 2.1). The application should be able to display the moving predator and prey individuals as well as the predator group radius (see Fig. 2.2). Changes in the environment are done by updating the respective tiles' colour, e. g. if a predator moves from position A to position 8, then position B will be changed to the predator's colour, whereas position A will receive the colour representing an empty position. You should implement the following rules regarding the hunt:

1. Each Predator individual must be able to make a decision regarding his hunting strategy based on the given circumstances (hunt small prey alone or notify group when help for larger prey is needed). Other hunting strategies apart from cooperative and solitary are not needed.
2. When hunting large prey in a group the predators must consider and stay within the group radius. while still trying to corner and surround the prey.

3. A prey is considered killed only if a predator within a group of the needed size reaches it.
4. If the predator group size is not of sufficient size the prey will retaliate by attacking the closest predator and killing it should it score a number higher than the predator's defence.
5. Predators being attacked by the prey should try to escape and join the group again (be within the group radius).
6. The predators may employ various manoeuvre tactics to corner the prey. without it being aware of them, i.e. the prey should simply try to get as far as possible from the predators without any specific strategy in mind.
7. Predators should be able to see prey and other predators from the group only within some radius (preferably greater than the group radius). Outside of this radius a predator may signal its presences and location to the others by "howling".
8. Each iteration all predators lose hit points due to starvation. These can be regained only through killing prey.

4 Related Work

4.1 Related Algorithms, Applications

5 Proposed Approaches

5.1 Input/Output Format

The input parameters for the application include, but are not limited to:

- Grid size – the width and the height of the grid (in number of tiles as in Fig. 2).
- Initial predator count – the number of predators at the beginning of the simulation.
- Prey size/position – position of a prey animal on the grid for instance as a pair of x, y coordinates and its size in grid tiles (small prey is easier to kill, but gives less nutrition).
- Auto-generate prey – when activated this option should enable the application to generate prey animals at random locations every x seconds.
- Predator run speed – how many tiles can the predator cross in a single iteration.
- Prey run speed – how many tiles can the prey cross in a single iteration.
- Predator starvation resilience – how many iterations can a predator individual endure without food intake (the predator dies if he does not successfully hunt a prey during this period).
- Predator defence chance – a larger prey would attack a lone predator and kill it if it scores a number greater than the predator's defence.
- Predator group radius – if a number of predator individuals are within a given distance of each other, they form a group and can hunt larger prey.

– Simulation object colours – for instance a drop down menu with several colour options for the predator objects and for the various prey objects (no duplicates!).
– Simulation speed – the overall speed of the simulation/animation (a lower speed may be desirable when observing the program in action, whereas a higher speed can deliver results more quickly).

The outputs delivered by the application should include, but are not limited to:

– Average food gain per iteration – the average nutrition from killed prey per iteration that the predator population has gained.
– Predator count – remaining predator individuals.

5.2 Benchemarks

5.3 Algorithms in Pseudocode

6 Implementation Details

6.1 Application Structure

6.2 GUI Details

6.3 UML Diagram

6.4 Used Libraries

6.5 Code Snippets

7 Experimental Results and Statistical Tests

7.1 Simulations

7.2 Benchmarks

7.3 Tables

7.4 Charts

7.5 Evalutaions

8 Conclusions and Future Work

8.1 How was the team work

– Quach Phu Thanh Ha
– Xuan Khang Nguyen
– Le Anh Quan Nguyen
– Ngoc Vinh Nguyen
– Hoang Viet Pham

8.2 What you have learned

- Quach Phu Thanh Ha
- Xuan Khang Nguyen
- Le Anh Quan Nguyen
- Ngoc Vinh Nguyen
- Hoang Viet Pham

8.3 Ideas for the future development of your application, new alforithms

8.4 Footnotes

The superscript numeral used to refer to a footnote appears in the text either directly after the word to be discussed or – in relation to a phrase or a sentence – following the punctuation sign (comma, semicolon, or period). Footnotes should appear at the bottom of the normal text area, with a line of about 2 cm set immediately above them.[1]

8.5 Program Code

Program listings or program commands in the text are normally set in typewriter font, e.g., CMTT10 or Courier.

Example of a Computer Program

```
program Inflation (Output)
  {Assuming annual inflation rates of 7%, 8%, and 10%,...
   years};
   const
     MaxYears = 10;
   var
     Year: 0..MaxYears;
     Factor1, Factor2, Factor3: Real;
   begin
     Year := 0;
     Factor1 := 1.0; Factor2 := 1.0; Factor3 := 1.0;
     WriteLn('Year  7% 8% 10%'); WriteLn;
     repeat
       Year := Year + 1;
       Factor1 := Factor1 * 1.07;
       Factor2 := Factor2 * 1.08;
       Factor3 := Factor3 * 1.10;
       WriteLn(Year:5,Factor1:7:3,Factor2:7:3,Factor3:7:3)
     until Year = MaxYears
end.
```

[1] The footnote numeral is set flush left and the text follows with the usual word spacing.

(Example from Jensen K., Wirth N. (1991) Pascal user manual and report. Springer, New York)

8.6 Citations

For citations in the text please use square brackets and consecutive numbers: [1], [2], [4] – provided automatically by LaTeX's `\cite` ... `\bibitem` mechanism.

8.7 Page Numbering and Running Heads

There is no need to include page numbers. If your paper title is too long to serve as a running head, it will be shortened. Your suggestion as to how to shorten it would be most welcome.

9 BibTeX Entries

The correct BibTeX entries for the Lecture Notes in Computer Science volumes can be found at the following Website shortly after the publication of the book: `http://www.informatik.uni-trier.de/~ley/db/journals/lncs.html`

Acknowledgments. The heading should be treated as a subsubsection heading and should not be assigned a number.

10 The References Section

In order to permit cross referencing within LNCS-Online, and eventually between different publishers and their online databases, LNCS will, from now on, be standardizing the format of the references. This new feature will increase the visibility of publications and facilitate academic research considerably. Please base your references on the examples below. References that don't adhere to this style will be reformatted by Springer. You should therefore check your references thoroughly when you receive the final pdf of your paper. The reference section must be complete. You may not omit references. Instructions as to where to find a fuller version of the references are not permissible.

We only accept references written using the latin alphabet. If the title of the book you are referring to is in Russian or Chinese, then please write (in Russian) or (in Chinese) at the end of the transcript or translation of the title.

The following section shows a sample reference list with entries for journal articles [1], an LNCS chapter [2], a book [3], proceedings without editors [4] and [5], as well as a URL [6]. Please note that proceedings published in LNCS are not cited with their full titles, but with their acronyms!

References

1. Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. J. Mol. Biol. 147, 195–197 (1981)
2. May, P., Ehrlich, H.C., Steinke, T.: ZIB Structure Prediction Pipeline: Composing a Complex Biological Workflow through Web Services. In: Nagel, W.E., Walter, W.V., Lehner, W. (eds.) Euro-Par 2006. LNCS, vol. 4128, pp. 1148–1158. Springer, Heidelberg (2006)
3. Foster, I., Kesselman, C.: The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, San Francisco (1999)
4. Czajkowski, K., Fitzgerald, S., Foster, I., Kesselman, C.: Grid Information Services for Distributed Resource Sharing. In: 10th IEEE International Symposium on High Performance Distributed Computing, pp. 181–184. IEEE Press, New York (2001)
5. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The Physiology of the Grid: an Open Grid Services Architecture for Distributed Systems Integration. Technical report, Global Grid Forum (2002)
6. National Center for Biotechnology Information, `http://www.ncbi.nlm.nih.gov`

11 Checklist of Items to be Sent to Volume Editors

Here is a checklist of everything the volume editor requires from you:

- [ ] The final LaTeX source files
- [ ] A final PDF file
- [ ] A copyright form, signed by one author on behalf of all of the authors of the paper.
- [ ] A readme giving the name and email address of the corresponding author.