AntWorld

From FujabaWiki

Author: Albert Zündorf, University of Kassel, Germany

Table of contents

Motivation

The AntWorld simulation is a case study designed as benchmark for the comparison of graph transformation tools. This benchmark is proposed for the GraBats 2008 workshop in Leicester.

The AntWorld simulation has been inspired by the Ants example delivered with the Greenfoot environment http://www.greenfoot.org/scenarios/index.html

As a benchmark, the AntWorld simulation tries to complement the Szierpinsiky triangles benchmark studied in AGTIVE 2007. The Szierpinsiky triangles benchmark employed very view and very simple rules. It was possible to organize the rule execution such that only local search was necessary for rule matching. Thus, the Szierpinsiky triangles benchmark actually measured the memory footprint employed by the various tools.

The AntWorld simulation has been designed such that the tools will most likely not run into memory problems. Over time, the number of Ants grows to the square of executed rounds. Thus, the focus of the benchmark is the movement of Ants. This can again be done with reasonably simple rules that mainly employ local search. Thus, the Antworld simulation stresses local rule application.

The AntWorld Simulation case study

The AntWorld simulation consists of an ant hill sitting in the middle of a large area. The ants are moving around searching for food. If an Ant finds food, it brings the food home to its ant hill in order to grow new ants. On its way home, the ant drops feromons marking the path to the food reservoir. If an Ant without food leaves the hill or if a searching ant hits a feromon mark, the ant follows the feromon path to the food. This behavior already results in the well known ant trails.

The area grid

Since this is a case study for graph transformation tools, the area in which the ants move shall be modelled by a grid of nodes. In order to enable the ants to go home on a straight path, if they have found some food, the area grid shall look like a spider's web cf. Figure 1:

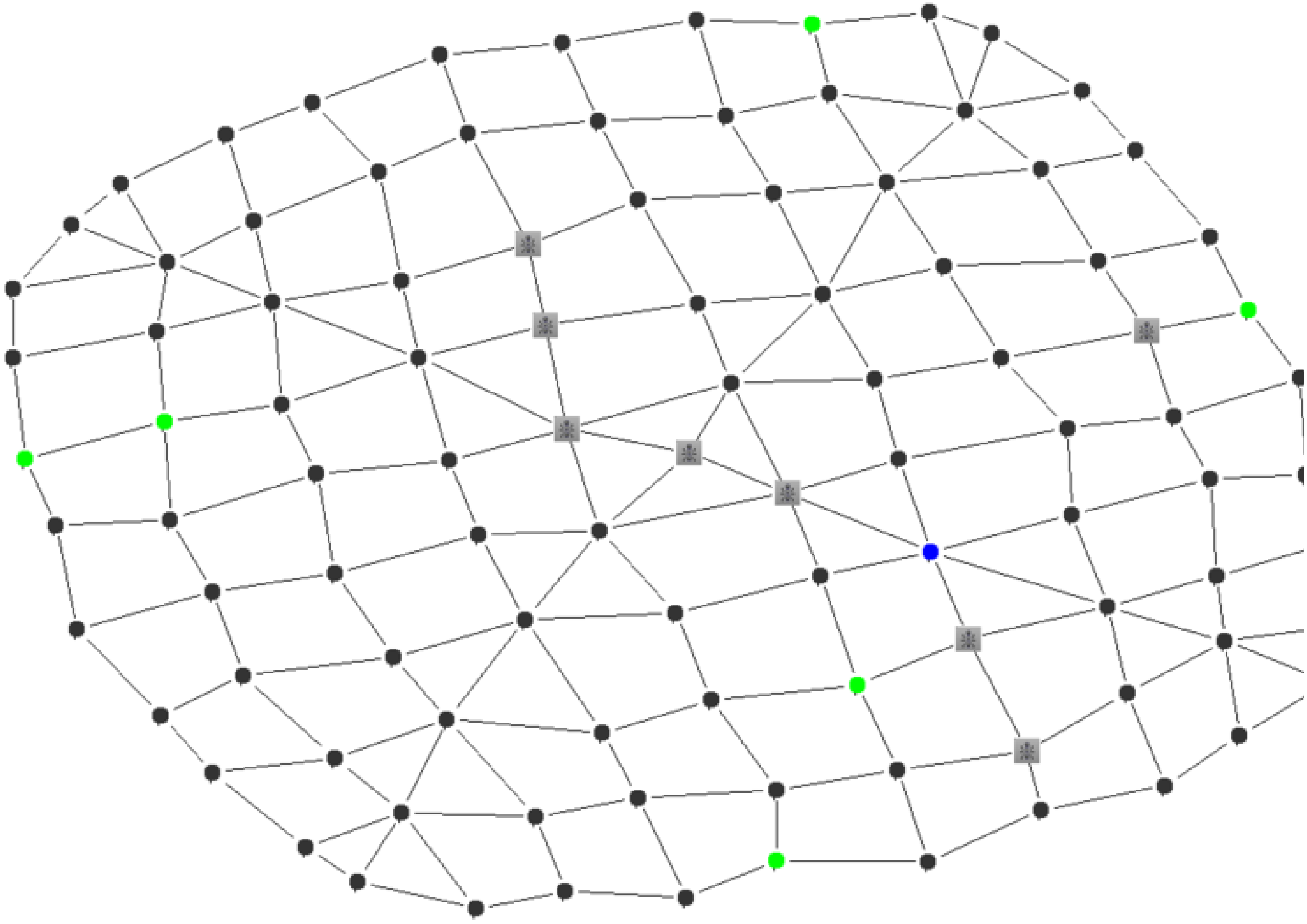

Figure 1: Ant World Example

In the center of the grid there is the ant hill. The first circle around the ant hill consists of 4 'exit' fields. In the second circle, each exit field has three child fields. In the next circle, the 4 fields on the main axis have three child fields while the normal fields have just one child field. This scheme creates a quite regular grid where each field has a quite straight path to the hill in the center of the grid.

Note, in Figure 1, grid nodes that are occupied by one or more ants are rendered using an ant icon. Grid nodes that carry food are rendered by a green spot. Grid nodes that carry more than 9 pheromones are rendered as blue spots. All other grid nodes are rendered as black spots.

Note, in Figure 1 one may recognize two short ant trails, one leaving the center to the north west and one leaving the center to the south east. While the north west trail is crowed with ants, the south east trail shows one blue grid node marked by pheromones.

Note, Figure 1 shows a number of food places not yet detected by the ants.

Ant moves

The AntWorld simulation works in rounds. Within each round, each ant does one move. The ant behavior depends on the following modes:

- If the ant has no food and is on a field with food, it takes one piece of food and enters the food carrying mode. It may still move within the current round.
- If the ant carries some food, it follows the links towards the 'inner' circle. Thus, the ant moves towards the ant hill by one circle. During its way home, on each visited grid node (including the 'food' node, the ant drops 1024 parts of pheromones. This guides other ants to the food place. Note, if an ant drops new pheromones on an already marked grid node, this new pheromone parts are added to the already existing parts.
- If ant with food is on the hill node, it drops the food and enters the search mode. It may leave the hill within the

same round.

- An ant without food is in search mode. In search mode, the ant checks the neighbor node(s) of the next outer circle for pheromones. If there are neighbor nodes on the next outer circle with more than 9 parts of pheromones, the ant chooses one of these fields, randomly. The random choice shall be fair based on usual mechanisms for random choices, e.g. Java class Random may be used.

- If the ant is in search mode and no outer neighbor has sufficient pheromones, the ant moves to any of its neighbor fields based on a fair random choice. (However, an ant without food shall not enter the ant hill.)

General area management

- Initially, the area grid shall consist only of the hill and the first two circles. In addition, the hill shall contain 8 ants. No food provide on the grid, yet.

- Whenever during one round an ant enters the currently outmost circle (i.e. the border of the yet known area), a new circle of nodes shall be created. During the creation of this next circle, every 10th node shall carry 100 parts of food. If a circle has e.g. 28 nodes, node 10 and node 20 of that circle shall have food. Thus, this circle would need just 2 more nodes to create a third food place. Therefore, this 8 nodes are kept in mind and during the creation of the next circle (in our example with 36 nodes) we add another food place when 2 more nodes have been added. Thus, accross circles, every 10th node becomes a food place.

  Note, according to this schema, we create 10 parts of food per one grid node. Since each food part may result in one ant, one might expect pretty many ants in this simulation. However, in our test runs, we observed an average of 1 ant per 5 fields (on the long run). It seems, that due to the random search strategy, even this large number of ants have difficulties to find food far away from the hill.

- After each round, the pheromones shall evaporate. This is needed in order to erase an old ant trail once the food has drained. Thus, after each round, on each grid node, the number of pheromones shall be multiplied by 0.95. In case of a fraction, the number shall be rounded to the next smaller natural number.

  Note, using the factor 0.95 and 1024 start pheromones per drop, the number of pheromones on a grid node drop below 10 within roughly 100 rounds. Thus, a single ant may be able to travel about 50 steps home to its hill, drop the food and travel about 50 steps out following its own pheromone path back to the food place. Food places with a distance of more than 50 circles need the collaboration of multiple ants that pick up the ant trail and refresh it, continuously.

- After each round, the hill shall consume the food brought to it and it shall create one new ant per delivered food part. These new ant spread out in the next round.

Implementation remarks

- Grid nodes, ants and the hill shall be modeled as explicit graph nodes. Ant positions on grid nodes shall be modelled using a link. Multiple ants are allowed on the same grid node.

- The number of food parts and pheromones on a grid node or in the hill may be modeled as an integer attribute. Note, we have ten times as many food parts as grid nodes and pretty many pheromone parts. Modeling these parts as explicit objects would result in a benchmark measuring mainly the creation and deletion of food (and pheromone) objects which is not our desire. This benchmark is designed to measure the execution of rules with local search.

Goals

- Each tool shall model and run the AntWorld according to above rules.

- We would like to see some nice animations showing the ants and how they search for food and how the form ant trails.

- For performance measurement, the simulation may run without a GUI. The results shall be illustrated by a table and or diagram showing for reasonable numbers of rounds the number of circles of the grid, the number of nodes of the grid (4*noOfCircles^2), the number of food parts created, the number of ants created, and the total execution time.

  This shall be accompanied with a description of the deployed hardware.

  In order to achieve comparable results, University of Kassel may provide a common compute server for each participating team. Thus all measurements may be done on that same server and thus become comparable.

- This page was last modified 23:56, 20 May 2008.