Background

Nature is very inspiring for computer scientists. Many general algorithms for searching solutions in a given state space are inspired by behaviour in the nature. Genetic algorithms are based on evolution and on the fitness function, simulated annealing based on the idea of controlled cooling of a metal material. In the field of multi agent systems and emergent behaviour the ants, bees, fish, birds are often given as examples of the swarm intelligence [1].

The topic of this coursework is to simulate the foraging of bees. The simulation will take place in the artificial environment provided as a Java package by Neil Madden and Julian Zappala. The goal of this paper is to give a brief outline of specification and high level design of the whole artificial system i.e. including environment and agent(s). However, the final implementation may differ from this specification and design.

Specification

Environment properties [2]

- discrete
  - infinite grid of cells (cell types are further described below)
  - time is represented by a sequence of time steps
  - duration of a time step can be modified
  - 10,000 time steps makes one day
  - the simulation takes ten days
- partially observable – a bee can see only 5 cells in each direction (Manhattan Metrics)
- dynamic – yield of flowers depends on the day time
- contains many agents – bees
  - hence **multi-agent architecture**
  - there can be many agents on one cell
- in each time step each agent have to perform an action (actions are further described below)
- execution of the actions is synchronous – agents are sorted and the result of an action performed by *nth* agent will be visible for the *mth* agents, where $m > n$

Agent's actions

- non-deterministic (Although the actions with higher utility will be prefered).
- sortable by utility function (depends on agent's state)
- infallible – action cannot fail in the environment
- have costs

Other

- simulation can be slowed down, speeded up, paused, stored and replayed (for testing and debugging purposes)
- GUI and CLI interfaces will be provided
- Java SE platform

List of actions

Name	Cost	Precondition	Postcondition
Move	*1*	true	Position of bee is at an adjacent cell
Rest	*-1*	true	Energy of bee is increased by 1
Create a nest	*100*	Actual cell has not to be a nest Maximum number[1] of nests has not been exceeded yet.	New nest
Lay an egg	*1*	Actual cell has to be a nest Food in nest is grater than 1000	New egg Food in nest is decreased by 1000
Deposit nectar	*x*	Actual cell has to be a nest. Bee has at least x energy.	Food in nest is increased by x. Energy of the bee is decreased by x.
Harvest	*-x*	Actual cell has to be a flower.	Energy of the bee is increased by x. x depends on daytime and type of flower.
Communicate*	*2*	Actual cell has to be occupied by another agent.	true

* Communication protocol will contain only some abstraction of bee's waggle dance [3] (showing the direction and distance of cell with high yield). Exchanging the whole representations between two agents won't be possible. In my opinion, communication protocol shouldn't be too much expressive, since it doesn't reflect the behaviour of bees in nature then, although for purposes of this course, there may will be more action than one to make the simulation more interesting.

Types of cells

On any cell there can be many bees => no collision of bees is allowed

Type	Description
Empty	GUI visualises it as a green place. It is possible to build a nest on empty cells.
Nest	Represents bee's hive. Eggs can be laid here.
Flower	Visualised as a flower. Bee can land on this cell and perform the harvest action. Each flower has a different number of clumps and the yield of a particular flower is determined by its type.

1 Small number (<3)

Types of flowers

There will be six types of flowers. Each type determines a function of yield. Each type has a different peak time and a peak duration. In general all these function follows a Gaussian curve and the task of the agent is not to assume these function (by hard coding them into their program), but learn them.

Design

The task is to design an intelligent agents which can cooperate to achieve the common goal. The common goal is to lay as many eggs as possible in ten days. The agents will not be aware of the fact that their life is limited by ten days. In other words they will not know when the simulation ends but they will know, how long the simulation runs. Each agent will have its own goals which may depend on its role and the knowledge it has got. For instance agent with no knowledge about the environment will likely choose such actions which leads to exploration of the environment or to get this knowledge from its co-workers.

Possible roles (approximate number of dedicated agents)

- **Queen (1)** – lays eggs, stays near the nest, communicates the knowledge with others, may decide to create a new nest at some better position on the basis of the obtained knowledge, the most centralized agent in this decentralized system
- **Scout (~2)** – Explorer, tents to go to undiscovered locations, which may offer better conditions for collecting nectar.
- **Harvester (~20)** – its goal is obvious – to harvest as much nectar as possible in the shortest amount of time

Except the queen role, the roles are not definite. When performing agent's selective function – **senseAndAct**, agent may accept some role with certain degree of probability. These probabilities may be hard coded, depend on environment and agent's representation or be communicated with the queen.

Bee's representation

A bee will store the map of cells it has visited and "measurements", where measurement is the value of yield of a given type of flower in a given time. On the basis of these measurements the bee will try to approximate the real yield function. Perhaps some methods of Regression Analysis could help or Backpropagation NN [4].

Bee's selective function

In each time step the environment requires from a bee to perform the *senseAndAct* function so that the bee have to select its action. The action should be selected in correspondence with its intentions. Beside the main goal (to lay eggs) there are another goals such as not to exhaust its energy. In this sense it is trade off amongst the goals.

My selective function will incorporate the search algorithm, probably some variant of A* or some variant of iterative DFS (with increasing horizon) which finds some sequence of actions i.e. plan [5]. This plan will be stored (List<Action> plan), first action of this plan will be used and in the next time step will be performed the second action and so on. The plan can be re-planed in each time step with some degree of probability or after certain amount of steps. The re-planing may will be dependent on the amount of resources, especially time.

Conclusion

It is hard to predict the values of all relevant parameters of agent and environment in advance, e.g. maximum of clumps in one flower. Parameters of the final version will be probably different from parameters described in this document. However, the whole concept of the multi agent architecture, bee's roles and deliberative agents will not vary.

Sources

[1] http://en.wikipedia.org/wiki/Swarm_intelligence

[2] some parameters from http://www.cs.nott.ac.uk/~bsl/G53DIA/slides/Project-Description.pdf

[3] http://en.wikipedia.org/wiki/Waggle_dance

[4] http://en.wikipedia.org/wiki/Regression_analysis

[5] idea based on the IRMA architecture http://www.cs.nott.ac.uk/~bsl/G53DIA/slides/Deliberative-Architectures-II.pdf