Salzburg Interest Group on Integrated Systems

siGis-00x-2000 March 14, 2000

Project Ideas

Proposal

Roland Schwaiger
rschwaig@cosy.sbg.ac.at

Department of Computer Science
University of Salzburg

Correspondence to:

Universität Salzburg
Institut für Computerwissenschaften und Systemanalyse
Jakob–Haringer–Straße 2
A–5020 Salzburg
Austria

Project Ideas

Roland Schwaiger
Department of Computer Science
University of Salzburg
A–5020 Salzburg, Austria
rschwaig@cosy.sbg.ac.at

1 Evaluation of netJEN (BioInf)

netJEN is a software tool used for evolution of artificial neural networks (ANN) by means of genetic algorithms (GA). Furthermore coevolution of training data sets (TDS) is implemented. Your task comprises the evaluation of

1. performance of the TDS evolution
2. verifying the distribution of the evolved data in the TDS, i.e. given two examples – unit square and spirale – what are the data points selected for learning?
3. end user test of the netJEN system.

Optional tasks are

1. encoding of the training data
2. investigation of n-dimensional encoding.

So the basic setup consists of

1. netJEN: where is it installed, how to use, any user docu
2. BugReport tool: how to use, where installed
3. test problems: reference, where to find, how to encode

2 GO with netJEN (BioInf)

Some prior project groups have implemented the game of GO with the former netGEN system. Based on their results, which are by the way available (CD-ROM, paper, ...), your task is to improve the encoding of the board and checker positions. A starting point for encoding the whole scene is a one to one mapping of the positions of the checkers and as output again use a one to one mapping of the future position of the checker. As can be seen easily this is a highly inefficient approach, so what's better? Some ideas are,

1. extraction and encoding of features (grouping, ...)
2. multi-resolutional encoding
3. transformation in a different domain
4. ...

We will use a 9×9 practice board for evaluation of our approaches.

3 Evolution of transfer functions (BioInf)

A quite new research direction, in the area of evolution of ANN by means of GA, focuses on the evolution of socalled transfer functions (TF). TF are responsible for the output of the single units in an ANN, the neurons, and as a whole for the overall output of the ANN. Till now primarily sigmoidal TFs were used partially because the most prominent learning algorithm – error back-propagation – requires differentiable TFs. In our setup ANNs are learned by evolution, i.e. weights are manipulated based on predefinded fitness functions. Using the evolutionary approach no differntiability of TFs is required. Your taks will be to investigate various TF types, especially the sin function, compared to standard TFs.

A paper will be provided by Schwaiger Roland.

4 Visualization of ANNs (PPKT)

As a result of the evolutionary cycle, netJEN saves the best ANNs to hard disk in a predefined format – StuttgarterNeuronaleNetzeSimulator (SNNS) file format. The number of ANNs to be saved for later processing may be set by the experimenter. Thus your taks will be to

1. show thumbnail images of the ANNs
2. display single networks
3. arrange ANN elements (neurons, connections, weights, ...) based on user requirements, such that optimal layout is guaranteed.

Especially the last subtask requires literature search in the area of graph theory and chip layout.

The SNNS file format is given below showing an example for an XOR network

```
SNNS network definition file V1.4-3D
generated at Mon Apr 25 15:58:28 1994

network name : xor
source files :
no. of units : 4
no. of connections : 5
no. of unit types : 2
no. of site types : 2


learning function : Quickprop
update function   : Topological_Order


site definition section :

 site name | site function
-----------|------------------
 excite    | Site_WeightedSum
 inhibit   | Site_Pi
-----------|------------------


type definition section :

name          | act func     | out func     |sites
--------------|--------------|--------------|---------
LongeroutType | Act_Logistic | Out_Identity |
outType       | Act_Logistic | Out_Identity |
--------------|--------------|--------------|---------


unit default section :

act      | bias     | st | subnet | layer | act func     | out func
---------|----------|----|--------|-------|--------------|-------------
 0.00000 |  0.00000 | h  |      0 |     1 | Act_Logistic | Out_Identity
---------|----------|----|--------|-------|--------------|-------------


unit definition section :
```

```
no. | typeName       | unitName | act      | bias     | st | position | act func | out func | sites
----|----------------|----------|----------|----------|----|----------|----------|----------|---------
  1 |                | in_1     |  1.00000 |  0.00000 | i  | 3,5,0    |||
  2 |                | in_2     |  1.00000 |  0.00000 | i  | 9,5,0    |||
  3 |                | hidden   |  0.04728 | -3.08885 | h  | 6,3,0    |||
  4 |                | result   |  0.10377 | -2.54932 | o  | 6,0,0    |||
----|----------------|----------|----------|----------|----|----------|----------|----------|---------


connection definition section :

target | site    | source:weight
-------|---------|------------------------------------------------------------------------------------------------
     3 |         | 2:-4.83963, 1: 4.92521
     4 |         | 3:11.11523, 2: 4.53903, 1:-4.67122
-------|---------|------------------------------------------------------------------------------------------------
```

5 Visualization of netJEN problem domains and results (PPKT)

netJEN is applied to a variety of problem domains, ranging from stock exchange forecasts to classification of satellite images to extracted rules of the ANNs. Your task is to visualize

1. the problem domain, i.e. a satellite image
2. the result of the netJEN system

You will have to find and classify the possible application areas of ANNs. Based on your findings you will plan your software system ensuring to cover a broad spectrum of visualization the found domains. Some examples of problem domains, encoding in SNNS format and result files are available under

http://www.cosy.sbg.ac.at/ rschwaig/teaching/SNNSexamples/

6 Optimizing object groupings (PPKT)

You will use a GA to optimize groupings of objects $o_1, \ldots, o_n$ in a two dimensional area with respect to a given fitness function. The various objects o_i will be asigned values in a predefined interval $[l_i, u_i]$ measured on a discrete time interval, e.g. on a daily basis. The two dimensional area is discretized, where the single patches are

1. forbidden
2. belong to a predefined group
3. occupied or not by an object o_i

Your task will be to investigate the problem theoretically – is it possible to categorize this kind of problem – and implement a GUI and the GA. As a result you will give suggestions for the groupings in the area based on an overall fitness function, i.e. maximize assigned values of the objects with respect to an underlying model of the value assignment.

Basically the system consists of two components.

- The first component is used to simulate the agents within the world $\mathcal{W}$, with dimensions $w_x \in 1, \ldots, n$ and $w_y \in 1, \ldots, m$, where agents and objects coexist. The world is bounded by walls, where some spots are left open for exits and entrances. Agents $a_i \in \mathcal{A}$ may enter and leave the world according to some underlying distribution and the number of agents populating the world has an upper bound. Each agent a_i occupies a certain spot in the world which

  is not occupied by another agent,

  is not forbidden

  is not occupied by an object.

Each agent has assigned a certain value v_i^a where the value is reduced if the agent resides in front of an object. The amount deducted from the agents value depends on the object the agent is visiting and is denoted by c_i^o. The agents $\mathcal{A}$ may wander in the world randomly or according to a predefined or evolved scheme. Possibly trying to visit all objects with a $v_i^a > 0$.

The agents may wander and visit objects for $\mathcal{T}$ time units. We may call it a dynamic generational based population. Dynamic because agents are allowed to enter and leave the world and generational because after $\mathcal{T}$ time units another generation of agents will populate the world. The update of the simulation should be synchronous, i.e. let the generation be given at time t then the update for time $t+1$ depends solely on time step t. All collisions with other agents, etc. are calculated based on time step t[1]

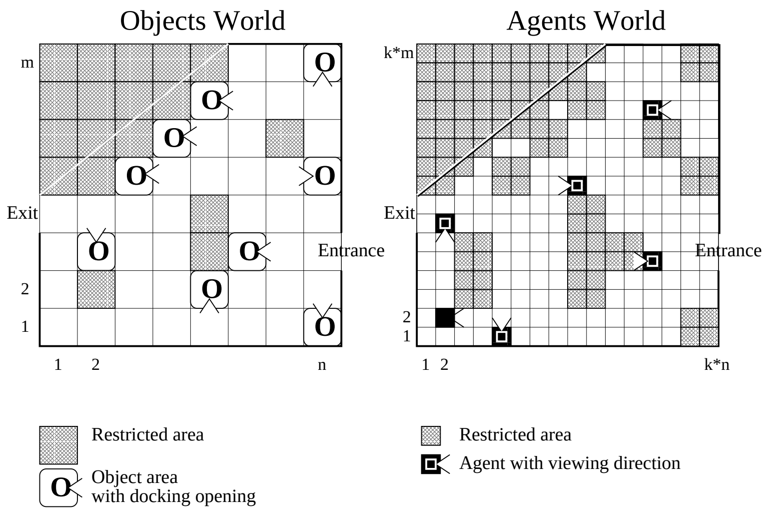

Figure 1: Object and agent world.

The setup of the objects is changed after one generation in order to collect enought data for the optimization cycle.

- The second component takes care of the object world. Basically the same size of the world is used as in the agent component case but the discretization of the world will be coarser. That is because agents do not, in general, occupy as much space as our objects do. The objects occupy one unit in the world. Objects have a certain type and therefore a certain cost to use c_i^o assigned which is deducted from an agents value if the object is visited by the agent. The objects collect the deducted values from the agents and after one generation the object has the value v_i^o. After each generation the setup of the objects is changed, basically to collect enough data for the optimization cycle. Some objects my be replaced by other objects from the object store.

Once the initial run has taken place, where e.g. data is collected over a period of $p \times \mathcal{T}$ time steps, the optimization of the object setup may take place. Through an evolutionary cycle, realized by a genetic algorithm, the setup of the objects is optimized according to a given fitness function. Usually we will try to maximize the sum of all object values.

7 ALGAE (PPKT)

evolution, learning and communication (elc) is an implementation of a multi agent system (MAS) conceptualized and implemented by Roland Schwaiger and Konrad Lang. They investigate the evolution of

[1] As you might imagine the calculation of time step $t+1$ from time step t needs a certain update rule. If the same order of the agents is used for each time step the first agent will be preferred all the time. Thus a shuffling of the agent update positions will be advantageous.

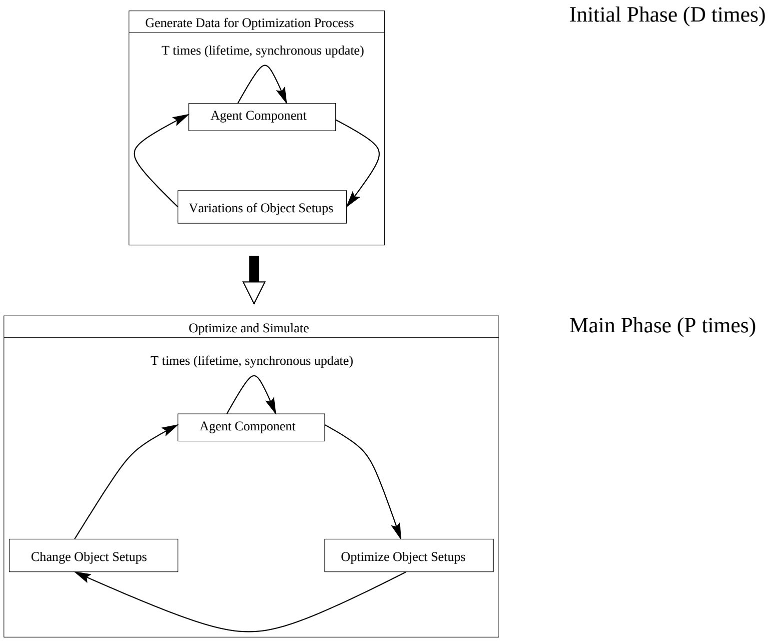

Figure 2: Overall optimization process.

wandering behavior of simple (reactive) agents in a highly dynamic environment. Those simple agents wander based on the outcomes of feed forward ANNs (multi layer perceptrons). The inputs to the network are calculated from positions of other agents within a sensor range of each single agent. As an outcome the ANN determines the steering direction (Forward, Back, Left, Right). The ANNs of the agents are evolved according to some given criteria, e.g. maximize the number of places visited or maximize the number of distinct partners within sensor range. Surprising results evolve over time s.a. simple steering rules and complex group behaviors.

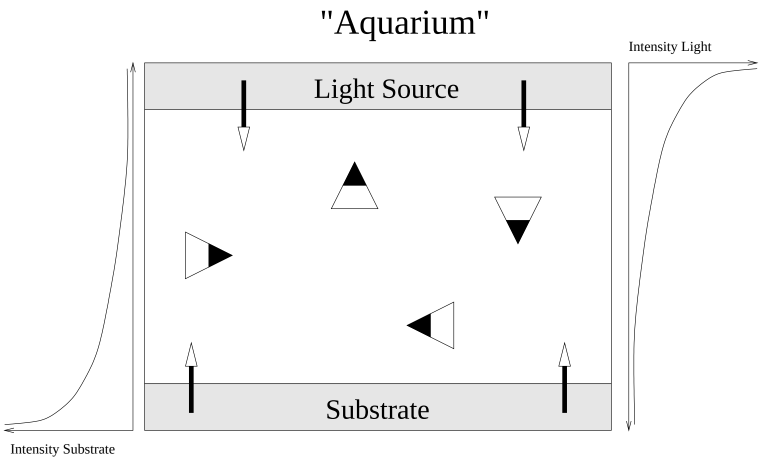

Figure 3: The world where the various agents live.

Based on the first findings in elc a followup system will be implemented based on the algae LOGOPODIUM. Imagine an aquarium where on top a uniform light source resides. At the bottom you may find some substrate. Algae (agents) live in the world between – for now we may view the aquarim as a two dimensional object. From time to time the light is switched on and off. As you might imagine, light has a higher intensity at the top of the aquarium than at the bottom. We might model the intensity by the function $1/x$ but any other function with similar shape will do, for now. The same holds for the substrate which diffuses into the water. The agents may consume the substrate, which after some time will be renewed.

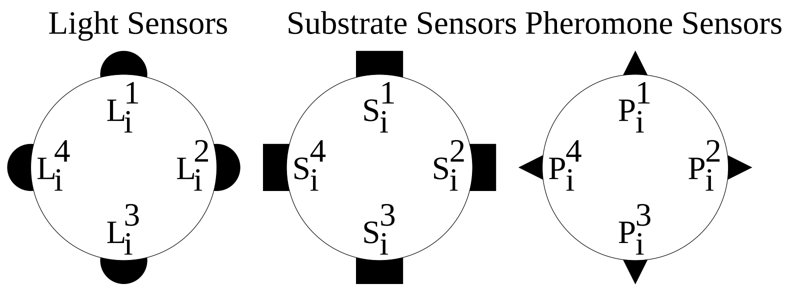

Figure 4: The various sensors of agents a_i.

All actions of one agent result from influences of its neighborhood. The agents (a_i) have sensors (receptors) for light ($L_i^j, j = 1,2,3,4$), substrate ($S_i^j, j = 1,2,3,4$) and pheromone ($P_i^j, j = 1,2,3,4$) which is released by other agents. According to the density of the various sources (light, substrate, pheromone) the sensors will show different values[2]. Through differnt values in the sensors an agent my "learn" to steer towards a target.

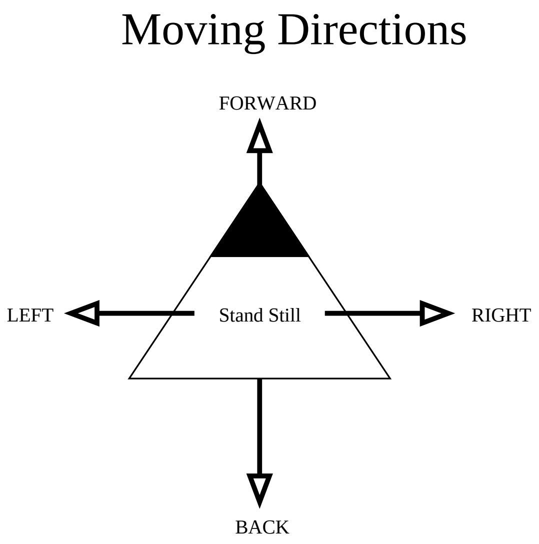

Figure 5: The possible moving directions of agent a_i.

The agents perform some actions according to their internal model. They may move in the world (aquarium). According to their local coordinate system they may move FORWARD, LEFT, RIGHT and BACK. The agents may stand still and do nothing. Obviously other agents or the borders of the aquarium hinter the agent to move according to the ANN output. Each action as described before requires a certain amount of energy. Agents consume substrate and light and collect and increase their energy level E. Agents may reproduce by mitosis (asexual) or meiosis (sexual).

The various actions of an agent are partially determined by its mapping from input of the environment to output, i.e. reactions. This type of agent may be calles reactive agent or stimulus/response agent. In our multi agent system we model this stimulus/response relationship by means of an artificial neural network – multi layer perceptron – (ANN). The input to the ANN consists of the sensor inputs and the energy level of the agent. Various hidden layers are used to perform transformations of the signals. Finally at the output level the calculated moving directions are shown. At the moment of writing the shown ANN is our first proposal, it will change during course of discussion.

[2]In order to model this situation a world object will set the sensor values

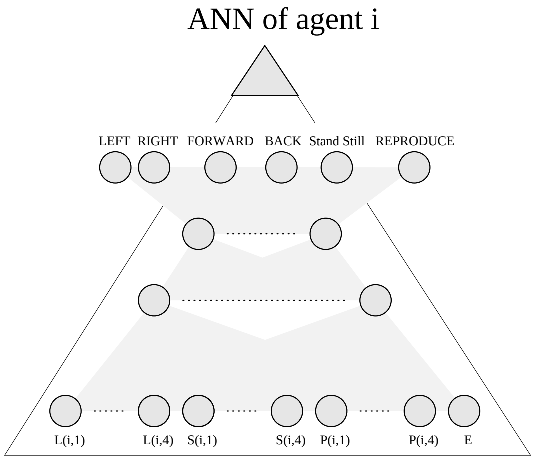

Figure 6: The ANN of agent a_i (first proposal).

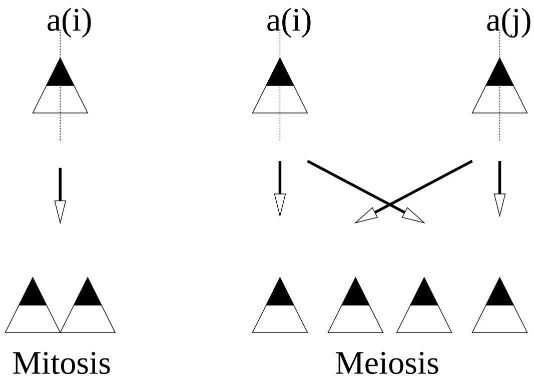

Figure 7: Reproduction of the agents a_i and a_j .

As has been pointed out, the algae live their lifes in a steady state population. That means that agents may life and prosper as long as their energy level does not reach 0. Once it has reached 0 the algae is damned to die. On the other hand if the energy level reaches a certain amount E_r, the algae may reproduce either sexually (meiosis) or asexually (mitosis). In order to find a partner the algae may produce pheromones. These pheromones may be detected by other agents throught their pheromone sensors. Possibly some algae learn to navigate to reproduction partners.