Genetic Algorithms and Genetic Programming
Lecture 8

Gillian Hayes

17th October 2006

School of informatics

Flatland: Evolving Controllers

- FlatLand
- The human controller
- The animal controller
- Learning using a GA
- Comparison learning mechanisms
- Performance

FlatLand

- Inspired by E. Abbott's book
- A 2D world with several agents: humans (evolve controller) and animals (fixed controller)
- Control agents through directing their motion
- Simple sensing system
- Task: moving to a fixed point and avoiding other agents
- Cooperative behaviour emerges
- Neural network controller, evolve its weights
- Compare: neural network controller trained by supervised learning

FlatLand

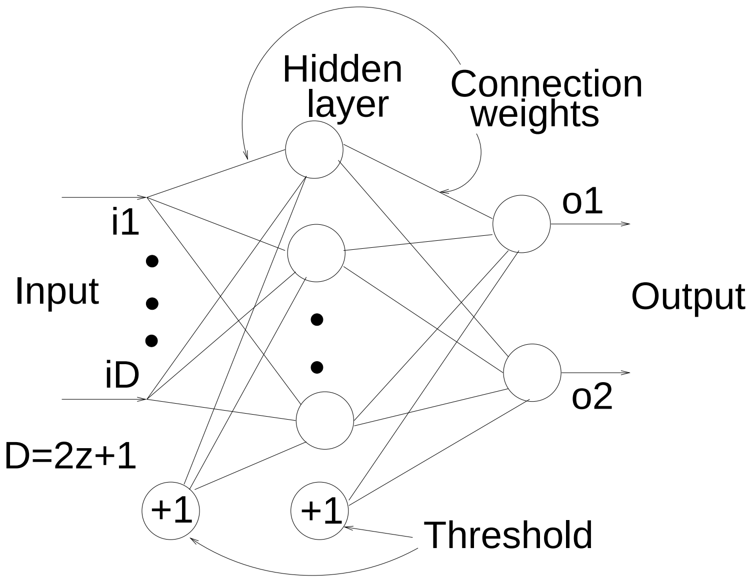
- 20 (in most cases) simple agents called “Humans” (small population size – computational cost of evaluation)
- Target position to achieve. Once achieved, target switches to new position.
- Other agents: “Animals”.
- Humans are controlled by a neural net, must avoid each other. Evolve weights in this net.
- Animals (for comparison) are controlled by Artificial Potential Fields.
- Evolutionary computation seems a good bet
  - Large, complex search space with many local optima
  - Non-exact objective function – what constitutes a good performance? Hard to make a gradient-based controller.

Controlling the "Human"

Simple feedforward neural network of sigmoid units.

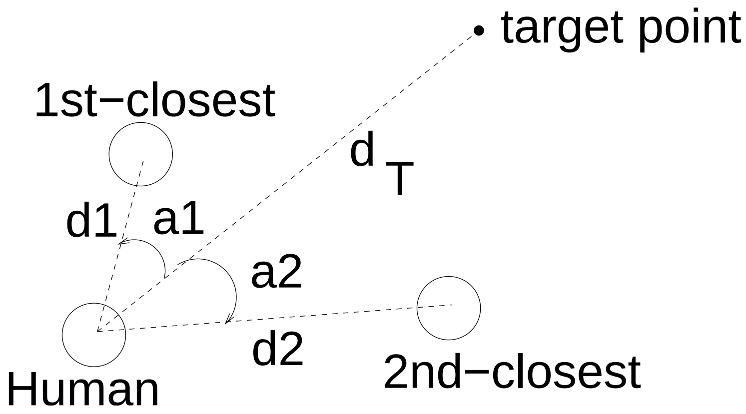
Sensory input, motor command output

Motor command gives polar coordinates of the human's next step (distance and turn angle – some post-processing of o_1 and o_2 needed, see paper for details)



“Human” Sensor System

Polar coordinates, number of inputs $D = 2z + 1$ where the Human takes into account the z closest other Humans (and their polar coordinates) and the extra input is the distance d_T of the Human from its current target.



Animals and Artificial Potential Fields (APFs)

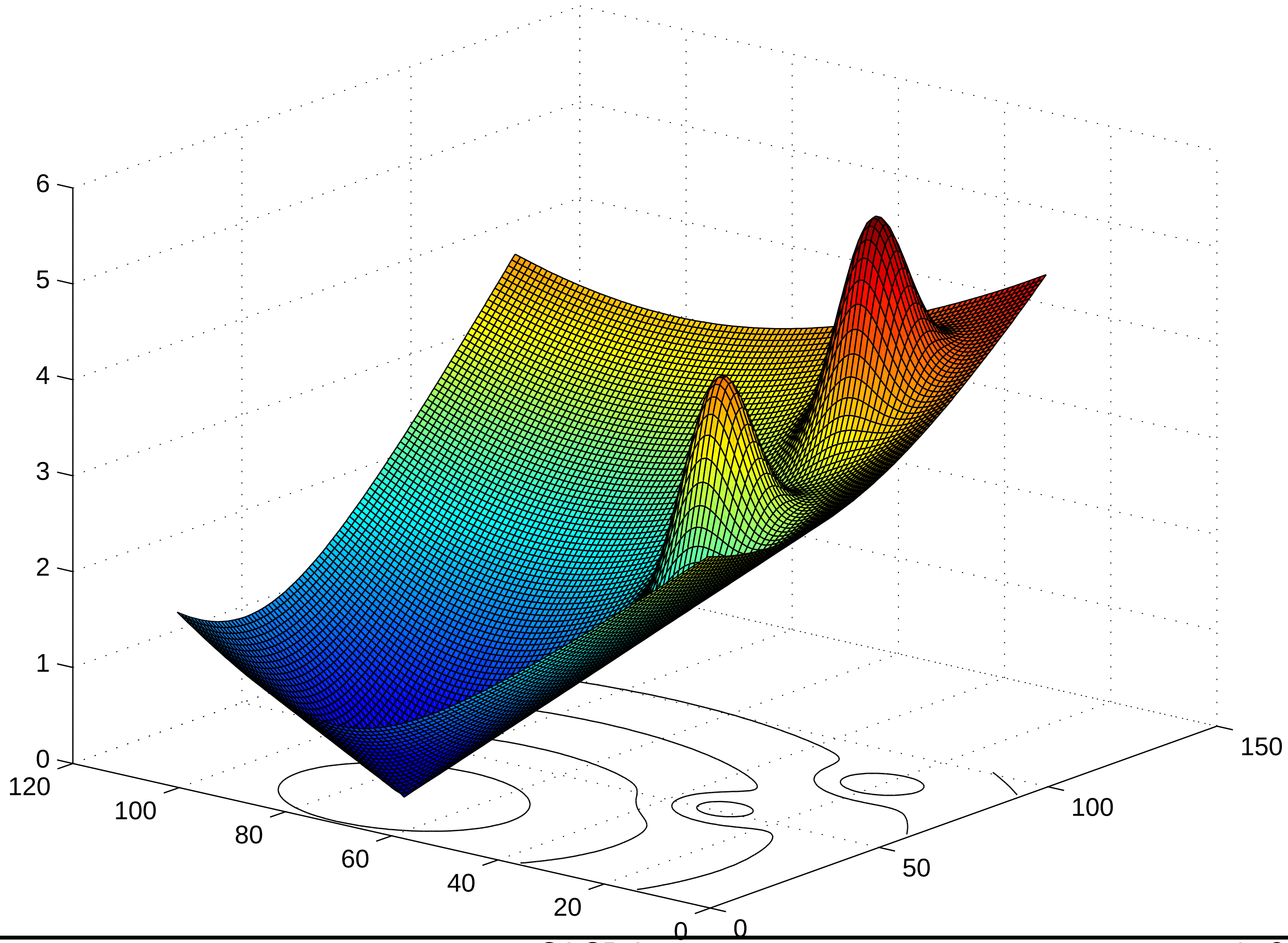
Another species for the environment, for comparison and training purposes.

$$f(x,y) = \alpha\sqrt{((x-x_T)^2 + (y-y_T)^2)} + \beta\sum_{i=1}^{z} e^{-[(\frac{\Delta x_i}{4R})^2 + (\frac{\Delta y_i}{4R})^2]}$$

(x_T, y_T) is the target position

$\Delta x_i = x - x_i$, $\Delta y_i = y - y_i$, where (x,y) is the position of the Animal and (x_i, y_i) is the position of its ith neighbour.

Target attracts, neighbours repel.



6
5
4
3
2
1
0
120
100
80
60
40
20
0
0
50
100
150

Animals: Why?

- APF provides almost optimal solution – animals slide down the (dynamically generated) surface to the target
- Use as a comparison for performance of humans
- Use as a source of training data for humans in supervised learning

Learning Mechanisms

1. Back propagation (supervised learning)

Get input/output pairs using APF strategy and train network using back propagation.

These are samples of good behaviour and eventually the human's controller will act like an animal controller.

2. Evolve the connection weights of the NN

- Generational GA, population size 20 (small)
- Chromosome: real numbers, -5 to +5, represent weights
- Fitness function: penalises crashes, rewards target achievement
- Clone each agent 20 times, evaluate for 200 simulation steps, fitness of agent = average fitness of its clones

- Pure elitism: two best agents breed, each parent cloning 9 offspring
- Mutation in each weight of each offspring with small probability, replace by random weight values
- Crossover: uniform or Montana and Davis

Characteristics of the FlatLand problem

- Dynamical multi-agent problem
- Partial information – only senses positions at current timestep, no speed or rate of change information
- Discontinuous time-varying information – closest agent changes
- Training problems – animals never get close to each other, trained humans do (so training doesn't cover all cases)
- Very few collision cases (1.5% of each human's lifetime) so sparse data (get to reward collision-avoidance rarely)

Results

Compare GA humans, BP humans, animals, random strategy, target achievers (agents that head straight for the target regardless of what is in the way).

Average of 10 evaluation runs of a 20-agent environment:

Species	Collisions	Targets	Speed	Twisting	Performance	Behaviour
Random	198620	7	0.46	10–60	0.0010	Random
Target Achiever	2000	3348	0.50	0.0	0.5	Target achiever
BP	663	3121	0.51	8–12	0.8219	Animal like
GA	268	3179	0.90	50–53	0.9297	Fear-circling
Animals	0	3200	0.5	2–3	1.0	Animal

- Performance is a composite measure relative to no. targets achieved by best Animals and no. collisions of the worst Target Achievers.

- GA Humans move much faster than the rest, but with more collisions and fewer targets than Animals.
- They achieve this by circling around other Humans – looks like fear (so high twisting).

Outcome

- Effort cost comparison also shows BP does better for a lowish required performance whereas GA does better for a high required performance.
- So if you're not fussy, use BP, but if you want good performance it will cost you less (cpu etc.) to use GA.
- GA always does better than BP (but sometimes costs more).
- BP strongly training-set dependent.

Characteristics of Solution

- Cooperative behaviours (humans avoiding each other) emerge from scratch
- Animal behaviour constitutes (i.e. "looks like") a selfish strategy. But the desired human behaviour constitutes a cooperative strategy.
- Could also incorporate learning by survival, e.g. collision leads to death
- Strong creature-environment interaction – ALife property. Evolution depends on the particular history of the creatures in their environment. In this case, the only environment **is** the creatures, apart from the plane.

Future Work

- Make environment more complex
- Introduce death from crashing
- Introduce energy levels and “food”, internal milieu of agent
- Give ability to choose between more behaviours
- Opportunism – what does an object “afford” you (see Ignasi Cos, Theodoros Damoulas, Lola Canamero, Gillian Hayes and others)
- Evolve controllers as interesting and adaptable opponents for human players (see Yannakakis)