GECCO 2004 Competitions

Results and Future Directions

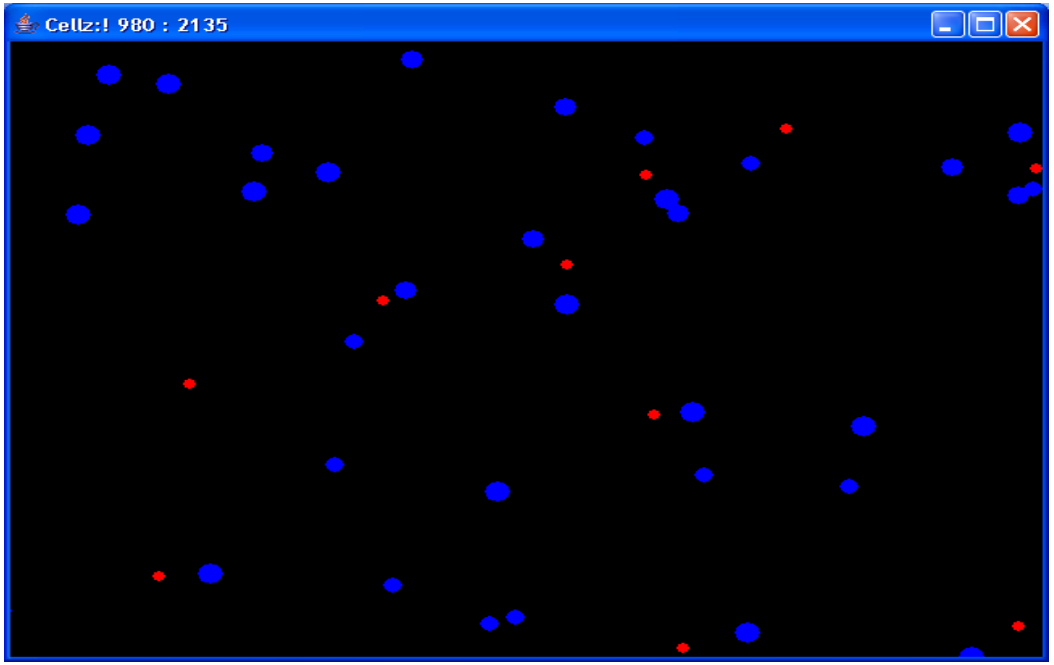
Simon Lucas (Competitions Chair)

Maarten Keijzer (Session Chair)

Competitions

- Cellz
  - A-life game
  - Challenging (too hard?)
  - 6 EOIs -> 1 entry
- Noisy DFA
  - Classic machine learning problem
  - 4 EOIs -> 3 entries
- TinyGP
  - A programming challenge
  - 5 EOIs -> 7 entries!

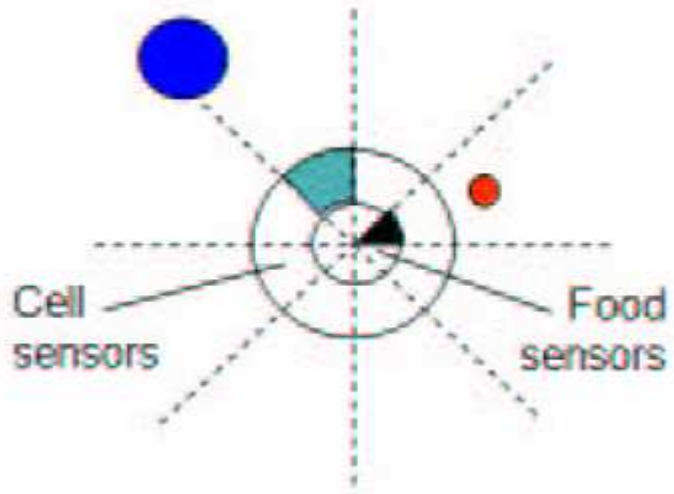
Cellz - the Game



- The cells (blue) move around to eat fixed food (red)
- Cells burn energy to move
- Cells divide when a certain mass is reached
- A piece of food once eaten is replaced (random location)
- Aim: maximize cell biomass at specified time limit (1000 steps)

Cell Controller Sensor Interface

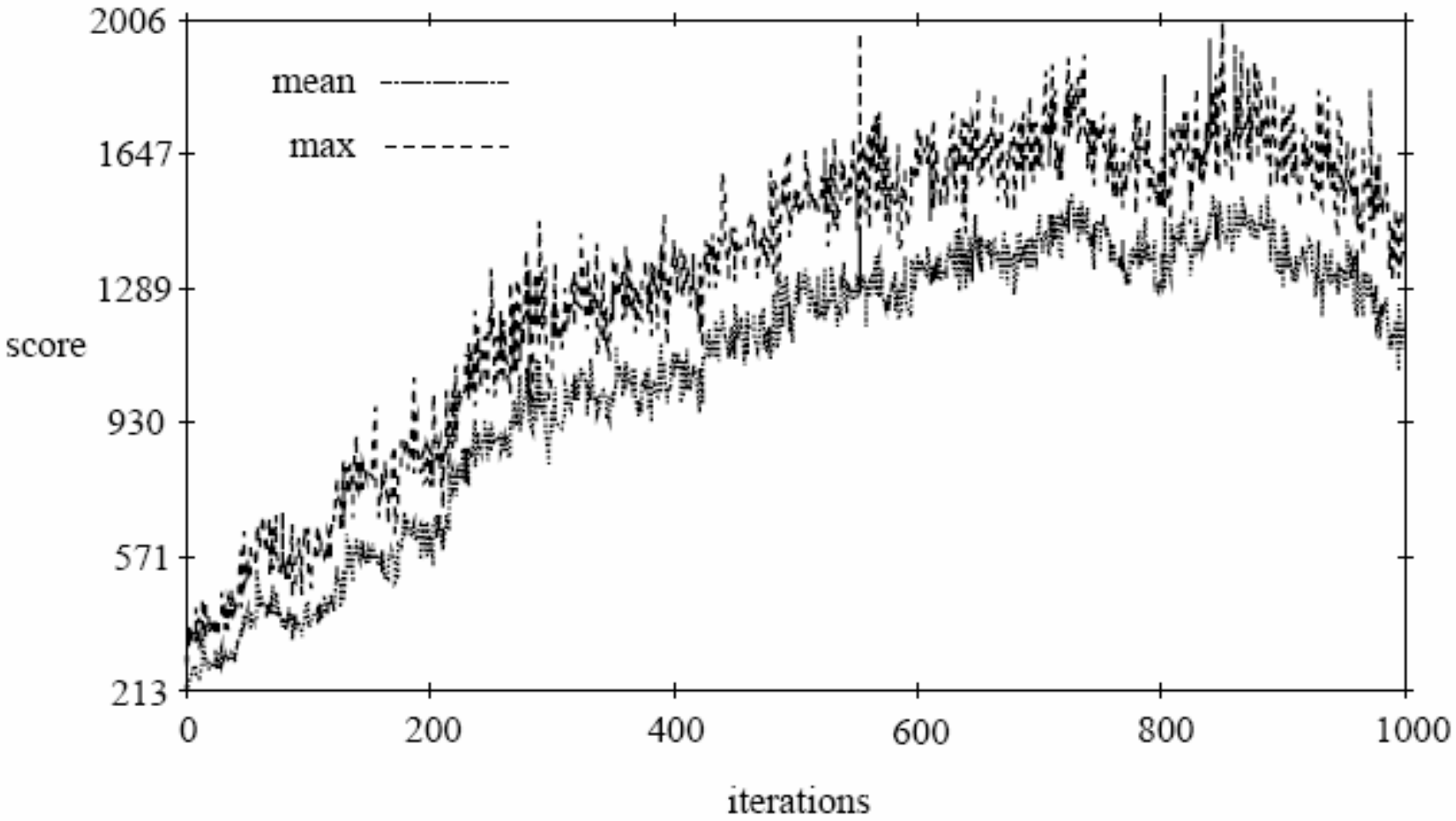
- Input: Wrap around retinas
- Activation: proportional to sum of reciprocals of particle distances
- Output: a force vector



Entering Cellz

- Entries took the form of a directed function graph
- Hence: neural nets and GP-style expression trees were possible controller architectures
- Cellz developer kit
  - Directed graph evolver + interpreter
  - Game SIM
  - Bonus: complete code for running your own web-based league (JSP based)
- Submissions were posted to web server
- Then evaluated, and entered into the league

Evolution must cope with noisy fitness: e.g.
1000 Fitness Evals of 10 samples each



On-line League Table

Position	Mean	Std. err	Controller name	Date submitted
1	1997	56	JB_Smart_Function_v1.1	Mon Apr 19 22:29:40 BST 2004
2	1920	123	JB_Smart_Function_v1	Mon Apr 19 22:27:32 BST 2004
3	1559	31	JB_Approx_Run_1	Sun Apr 04 04:32:03 BST 2004
4	1497	85	JB_Approx_2_FFNet	Sun Apr 04 07:27:41 BST 2004
5	1106	74	JB_SimpleSensor_v2	Sat Apr 03 23:04:41 BST 2004
6	943	43	JB_SimpleSensor1	Sat Apr 03 06:00:12 BST 2004
7	767	29	saveTest	Mon Mar 29 16:13:59 BST 2004
8	730	52	saveTest	Mon Mar 29 16:13:52 BST 2004
9	728	35	JB_Greedy_FFNet_1	Sat Apr 03 03:30:16 BST 2004
10	719	35	Mio	Tue May 18 11:44:12 BST 2004
11	717	31	saveTest	Mon Mar 29 16:16:39 BST 2004

Cellz Winner
Jason Brownlee (Australia)

- Designed neural architecture by hand
- Closely modelled on supplied hand-designed controller (implemented in Java)
- Used ECJ to evolve the weights of the neural net
- Achieved similar scores to the Java controller
- But with a neural net: now see it run
- Cellz league will continue to run – enter for fun!

Noisy DFA

- Classic machine learning problem
- Studied in the literature for well over thirty years
- Interesting for EC, Neural Net, Grammatical Inference and Machine Learning communities

The Problem Setup

- A random DFA (deterministic finite automaton) is constructed
  - By choosing the maximum number of states
  - Randomly selecting entries in the state transitions matrix
  - Randomly selecting state labels (accept or reject)
- This DFA (called the target DFA) is used to label two sets of randomly chosen strings (train + test)
- By learning only from the training set, must try to predict test set labels
- Twist: training set labels are corrupted with 10% noise (i.e. 1 in 10 chance of flipping each label)
- Target DFA ranging from 10 through to 50 states

Heuristic State Merging

- Basic idea: build prefix-tree acceptor
- This fits only the training data
- Then merge states while maintaining training set consistency
- Best methods have emerged from Abbadingo One DFA inference problem
- E.g. EDSM: evidence driven state merging
- Clever algorithms – carefully designed and tuned

Evolutionary Approaches

- Typically much simpler
- Encode DFA
- Evolve it!!!
- Heuristic methods work very well on noise-free problems
- But is evolution better able to cope with noise?

Entries (more details on web)

- Gomez (Jonatan Gomez)
  - Encoded DFA in biologically-inspired genotype
  - Evolved only the state transition matrix (assigned state labels optimally)
- Gauge (Miguel Nicolau)
  - Direct application of grammatical evolution
- Blue
  - Blue fringe EDSM with search extensions to cope with noise (see ICML 2003 paper)
- <Presentation from Miguel>

Results (10 States)

Entry	Success (> 99%)	mean	s.e.	Min:Max
Gomez	6/10	99	0.2	97 : 100
Gauge	7/10	92	5	42 : 100
Blue	1/10	89	3.0	74 : 100

Results (20 States)

Entry	Success (> 99%)	mean	s.e.	Min:Max
Gomez	5/10	98	1.6	83 : 99.9
Gauge	3/10	96	1.2	89 : 100
Blue	0/10	90	1.5	82 : 98

Results (30 States)

Entry	Success (> 99%)	mean	s.e.	Min:Max
Gomez	5/10	90	4.6	60 : 99.7
Gauge	0/10	75	3.7	57 : 90
Blue	0/10	82	2.3	70 : 96

Noisy DFA Summary

- Winner: Gomez
  - Solved largest problem (30 state)
- General conclusions
  - EC methods outperformed state of the art heuristic state merging
  - Interesting result
  - Perhaps not surprising: EC methods generally cope well with noise
- Future challenges:
  - Can EC methods scale up to cope with larger problems (i.e. more states)?

Tiny GP

- Challenge
  - Tiny, but reasonably complete implementation of GP
- Judging criteria
  - Clarity of implementation and accompanying documentation.
  - Readability and formatting of the code.
  - Number of lines of code
  - Source file size
  - Size of compiled version of program (if any)
  - Memory footprint when running.
  - Degree to which the requirements were met

Tiny GP Entries

- 7 entries
- 5 in C / C++
- 1 in Java
- 1 in Mathematica
- C/C++ versions were typically tinier
- Though many of the features used *could* have been done equally well in Java
  - Would be nice to see some of these ported to Java for the benefit of Java programmers (Simon)

Tiny GP Results

- Winner: Riccardo Poli
  - GP in about 2.7k (Compiled size with some cunning tricks!)
  - C – clean and simple implementation
- Close runner-up:
  - Maarten Keijzer
  - C++ with clever exploitation of the STL
- <Presentations from Riccardo + Maarten>

Competitions Summary

- First time for GECCO – a promising start!
- Encourage greater participation for future
- Make entrance easier – pick more standard problems?
  - E.g. regression or time series prediction?
- Another programming contest?
  - Tiny GA?
- Suggestions welcome…