Artificial Life Lecture 14

Tierra and Avida [Plus PDW]

This line of Alife investigation can be traced back to early games of **COREWARS** -- A.K. 'Kee' Dewdney, 1983 and in Sci American columns..

Competing assembly language programs:-
'*My aim is for my program to over-write and destroy yours*.'

Obvious connection with computer viruses.

Assembly language

In 1950s, 60s, memory was on ferromagnetic cores, 1 bit per core. Programs are written in a simplified Assembly language Redcode, and run in a virtual computer memory of 8000 addresses.

Two human competitors write their own Assembly language programs, which are placed in memory at random positions. Two 'program counters' execute one instruction of each program in turn, with consequent writing into other parts of memory, and jumping execution position according to whatever instructions are met.

Redcode

The instruction set of Redcode, an assembly language for Core War

INSTRUCTION	MNEMONIC	CODE	ARGUMENTS		EXPLANATION
Move	MOV	1	A	B	Move contents of address *A* to address *B*.
Add	ADD	2	A	B	Add contents of address *A* to address *B*.
Subtract	SUB	3	A	B	Subtract contents of address *A* from address *B*.
Jump	JMP	4	A		Transfer control to address *A*.
Jump if zero	JMZ	5	A	B	Transfer control to address *A* if contents of address *B* are zero.
Jump if greater	JMG	6	A	B	Transfer control to address *A* if contents of address *B* are greater than zero.
Decrement: jump if zero	DJZ	7	A	B	Subtract 1 from contents of address *B* and transfer control to address *A* if contents of address *B* are then zero.
Compare	CMP	8	A	B	Compare contents of addresses *A* and *B*; if they are unequal, skip the next instruction.
Data statement	DAT	0		B	A nonexecutable statement; *B* is the data value.

Partial Recursion

Note that there are conditional jumps -- this provides enough for partial recursive programs, ie the works!

Variation
Add to the above list a 10th SPL instrn = split
-- this then allows programs to multiply.

One needs rules such as 'each players share of CPU time is divided between all their various programs'

Dwarf

DWARF, a battle program, lays down a barrage of "zero bombs"

ADDRESS	CYCLE1	CYCLE2	CYCLE9
0			
1	DAT —1	DAT 4	DAT 14
2	ADD #5 —1	ADD #5 —1	ADD #5 —1
3	MOV #0 @—2	MOV #0 @—2	MOV #0 @—2
4	JMP —2	JMP —2	JMP —2
5		— 0	— 0
6			
7			
8			
9			— 0
10			
11			
12			
13			
14			— 0
15			
16			
17			

Dwarf versus Imp

From AK Dewdney "*The Armchair Universe*"
WH Freeman 1988.

The simplest one-line battle program **'IMP'**:-

```
MOV 0 1
```

just chomps through memory. What happens when INP fights against **DWARF** ?

Here is an **IMP factory** :-

```
SPL 2
JMP -1
MOV 0 1
```

A bit of pre-history

Actually, 1983 Core Wars was preceded by a much earlier piece of what must be the earliest version of such work -- almost completely forgotten.

Nils A.Barricelli's 'Symbioorganisms' -- paper by him "*Symbiogenetic evolution processes realized by artificial methods*", Methodos 9 (35-36) 1957.

Other papers by him reprinted in "*Evolutionary Computation: The Fossil Record*" DB Fogel (ed) IEEE Press, New York 1998.

Old classics

This work demonstrated, at such an early date, evolution within a computer program, crossbreeding, growth and self-repair, parasitism ...

I accidentally came across a reference to this *decades ago*, which triggered my own interest in the area -- and then never saw any discussion of this work again until "*Darwin Among the Machines*" GB Dyson Addison Wesley 1997.

Well worth looking at some of these early papers.

Tierra

At the second Alife Conference, Santa Fe Feb 1990, proceedings in "Artificial Life II" eds Langton, Taylor, Farmer, Rasmussen, Addison Wesley 1992

Steen Rasmussen and co-authors presented an updated version of Core World in an Alife context -- see paper in those proceedings.

But also Tom Ray presented Tierra for the first time -- all the breakthrough work had been in the preceding month or so on his Toshiba laptop (20 MHz). TS Ray "*An Approach to the Synthesis of Life*" in same proceedings.

Tom Ray

Tom Ray is an ecologist who learnt to program, so as to see if computer-virus-like things could evolve in a computer, in virtual memory.

Actually the Barricelli work (of which he was unaware) prefigured some of the ideas -- but the results Ray got, with 1990 computer power, made an enormous impression.

Tierra code set

32 assembly language instructions, each of which can be represented by 5 bits including operands

(unlike redcode).

cf DNA 64 codons, 20 amino acids.

Uses stack (push and pop) and registers

nop_0	no op	pop_ax	pop into ax
nop_1		pop_bx	
or1	flip bit of cx	pop_cx	
sh1	shift left cx	pop_dx	
zero	zero cx	jmp	to template
if_cz	if zero	jmpb	backward
sub_ab		call	call procedure
sub_ac		ret	return
inc_a		mov_cd	
inc_b		mov_ab	
dec_c		mov_iab	mov instrn
inc_c		adr	adrs of templ
push_ax	onto stack	adrb	search backwd
push_bx		adrf	search fwd
push_cx		mal	alloc memory
push_dx		divide	'SPLIT'

Tierra code features

This is a very small instruction set, but computationally complete.

Addressing by template
Each JMP instrn is followed by a sequence of NOPs, a mix of NOP_0s or NOP_1s.

To find the right place to jump to, a search is made for nearest set of *complementary* NOPs -- 0010 matches 1101

Tierra World

The world
is virtual memory, eg 60,000 bytes, linear (and circular)
Organisms
are each blocks of memory where exclusive *write* access has been reserved for its own code – though other organisms can *read* and *execute* code that is not theirs

Reproduction
can arise from a *mal* instruction, allocating new memory for a daughter cell; followed by a *divide* -- which gives daughter cell its own instruction pointer

Tierra Dynamics (1)

Time-sharing
At any one time there will be many organisms running their own code, with their own boundaries (tho can execute code outside their 'cell walls').

Multi-tasking allocates share of CPU time to each organism in turn

The Reaper
The world is a limited resource, fixed size of memory. Natural copy instrns fill up memory, 'the Reaper' kills organisms at random when too full -- leaving their dead code in the soup.

Tierra Dynamics (2)

Mutation
Background 'cosmic rays' flip bits of memory at random (at a low rate). During any copying, there are replication errors at random, at a higher rate.

Kick-off
Ray designed by hand an 80-instruction 'ancestor' designed to be self-replicating. Put one in, and let it replicate (with mutations).

Wait and Watch

What happened

Memory filled up with copies, mutants appeared.

Smaller self-replicating mutants were favoured, because of the way CPU time was shared.

Then parasites appeared -- only 45 instrns long, not capable of self-replication but manage to 'borrow' the replication code of their neighbours

(-- organisms can read or execute others code).

Parasites and hyper-parasites …

An ecology of different organism types builds up.

Some immunity to parasites appears in some self-replicators

Hyper-parasites 'steal the instruction pointers' from parasites

Social parasites...

hyper-hyper-parasites...

etc etc...

Macro-dynamics

Tierra is a rich experimental domain for watching a diverse ecology grow, possible parallels to situations in the real world.

For each organism, the '*physical world*' consists of '*energy*' (CPU time = sunlight) and '*space*' in memory, which is limited.

But above all, the environment which affects their fitness includes the *other organisms* around.

Ray was interested in looking for such features as punctuated equilibrium (punk eek), and any light that could be shed on the Cambrian explosion of diversity.

...still going on...

This work has been a fruitful source of ideas, has spawned a whole industry, Tierra on the Internet 'with patterns of activity following the night around the globe'

etc etc.

Personally, I consider the early Tierra work was truly exciting and mind-opening, but I am not so sure about the continuation...

Avida

Several variations spun off from Tierra, with varying motives.

Avida is one of these, coming out of more a physics background than a biology one. Chris Adami, Titus Brown and others

"*Introduction to Artificial Life*" Christoph Adami
Springer-Verlag New York 1998

Plus (as with Tierra) many publications online on www

Main differences with Tierra

The linear memory of Tierra is replaced with a 2-D grid. It is claimed that Tierra to some extent lacked 'locality', since a new offspring could result in death (via Reaper) of any other organism anywhere else.

Each cell on the grid can contain a complete program -- the instrn pointer loops back from end of program to beginning, unless a 'template-match' jmp takes into neighbouring cell/program.

More Avida

There are 8 nearest neighbours, and a preferred 'facing' direction for searching for template-matches.

Cell-division -- the divide command splits off the code and places it in a neighbouring cell, replacing whatever was there.

Avida has a local geometry, and a local reaper queue.

Instruction set

TABLE 9.1 Mnemonic of the 24 instructions available in the avida distribution as default.

Inst.	Inst.	Inst.	Inst.
nop-A	call	pop	allocate
nop-B	return	push	divide
nop C	shift-r	add	get
if-n-eq	shift-l	sub	put
jump-f	inc	nand	search-f
jump-b	dec	copy	search-b

The instruction set is smaller than Tierra, just 24, but there is a very clear family resemblance

Use made of Avida

This is designed to look at the evolution, and spread, of artifical life-forms in a space with 2-D toroidal (wrap-around) geometry, **cells** with neighbourhoods.

Conceptually, the designers see the execution of the programs within each cell as an artificial chemistry.

Programs can be rewarded if perform very simple computational tasks initially input/output capability, then simple Booleans of 1 or 2 inputs --- 'A or B'.

Dynamics of Evolution

This gives a 'fitness landscape', some programs artifically fitter than others.

In particular, the experimenters are interested in the dynamics of evolution in such a complex artificial world -- such factors as speciation, propagation of genetic 'information' through a population, etc etc.

Generally, as physicists, they were asking different sorts of questions from Tierra. In recent years, collaborating with a Microbial Evolutionary person, Rich Lenski, and testing biological ideas – fruitful collaboration.

EG some Avida-based publications

Adaptive Radiation from Resource Competition in Digital Organisms; S.S. Chow, C.O. Wilke, C. Ofria, R.E. Lenski, C. Adami. *Science* 305 (2004) 84-86

The Evolutionary Origin of Complex Features; R.E. Lenski, C. Ofria, R.T. Pennock, and C. Adami, *Nature* 423 (2003) 139-145.

Evolution of Digital Organisms at High Mutation Rate Leads To Survival of the Flattest; C.O. Wilke, J.L. Wang, C. Ofria, R.E. Lenski, and C. Adami, *Nature* 412 (2001) 331-333.

Genomic Complexity, Robustness, and Genetic Interactions in Digital Organisms R.E. Lenski, C. Ofria, T. C. Collier, C. Adami, Nature **400** (1999) 661-664.

References – Tierra and Avida

Tierra publications
http://www.his.atr.co.jp/~ray/pubs/
More general on Tierra
http://www.his.atr.co.jp/~ray/tierra

Avida publications
http://dllab.caltech.edu/avida/

Change of topic …

Time for a few slides relating to Passive Dynamic Walking and beyond.

Cf. work by Eric Vaughan, www.droidlogic.com

Walking without a nervous system

applied to humanoid walking

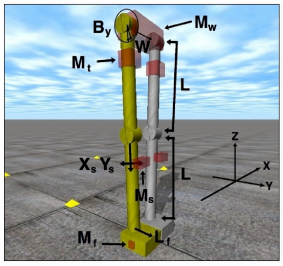
Passive Dynamic Walking is an equivalent of a glider

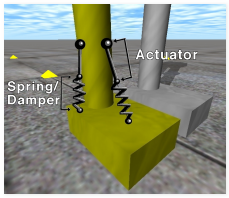
You could add more knees ….

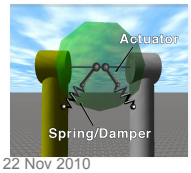
Or you could progressively add ***Control*** and ***Power***

All work by Eric Vaughan, www.droidlogic.com

10 degrees of freedom version







Damped ankle and hip springs

Some examples

Walking back and forward
(2D version, 6 DoF)

Balancing on a moving platform

Powered walking on the flat

All work by Eric Vaughan, www.droidlogic.com

ODE

These simulations are done within a 'Physics Engine' – that takes over most of the hard work of calculating how component parts move under forces of gravity, of motors, of meeting the ground or other objects.

We often use ODE (Open Dynamics Engine – Russell Smith and others)

http://www.ode.org/

Natural Motion

Natural Motion http://www.naturalmotion.com/

Torsten Reil – EASy MSc 1998/99