The Evolution of Structural Design Through Artificial Embryogeny

Taras Kowaliw, Peter Grogono and Nawwaf Kharma
Concordia University, Montréal, Canada

http://kowaliw.net

Presented at IEEE-ALIFE '07, Honolulu, USA

Motivation

- To develop a simple model of Artificial Embryogeny
- To create a non-trivial and externally-defined evaluation for “organisms”
- To explore the use of Evolutionary Computation as a means of testing model-level decisions.

Overview

- Review of Artificial Embryogeny
- Deva model
- Truss Interpretation
- Results of Evolution
- Future Directions

Review

Artificial Embryogeny

Usually studied in the context of Evolutionary Computation

A mid-step between genotype (representation) and fitness evaluation: phenotype (organism).

Inspired by, but not necessarily resembling, biological embryogenesis.

Usually undertaken to understand or exploit principles of biological development.

Many different models!

“Plausible Models”

Many researchers are interested in the creation of “plausible models” of actual embryogenesis.

Some models include:

- Various models of plant growth (Prusinkiewicz, Lindenmayer, Rolland-Lagan; 1990, 2006)
- Gene expression via cis-regulatory regions (Kumar, Bently; 2003)
- Cell growth, including inter-cellular forces (Eggenberger Hotz; 2003)

Embryogenesis is very complex and highly non-linear!

Exploitation

Other researchers are interested in exploiting principles of embryogenic growth for the design of solutions of engineering problems.

Including:

- Neural Networks (Eggenberger Hotz, Gmez, Pfeifer; 2003)
- Circuits (sorting networks) (Sekanina, Bidlo; 2005)
- Self-assembly of a CAD shape (Stoy, Nagpal; 2004)
- Micro-structures (Basanta *at al*; 2004)

Expected Gain

Some of the properties of AE models that have been demonstrated in particular contexts:

- Canalization of evolutionary space, allowing for the design of significantly larger organisms than is possible through direct encodings (Harding, Miller; 2006)
- Note: Consequently, many designs un-reachable!
- Self-maintenance and repair (Miller; 2004)
- Re-use of designs in new environments (Kowaliw, Grogono, Kharma; 2004, 2007)
- Simultaneous development of a construction plan (Reiffel, Pollack; 2004)

The Deva Model

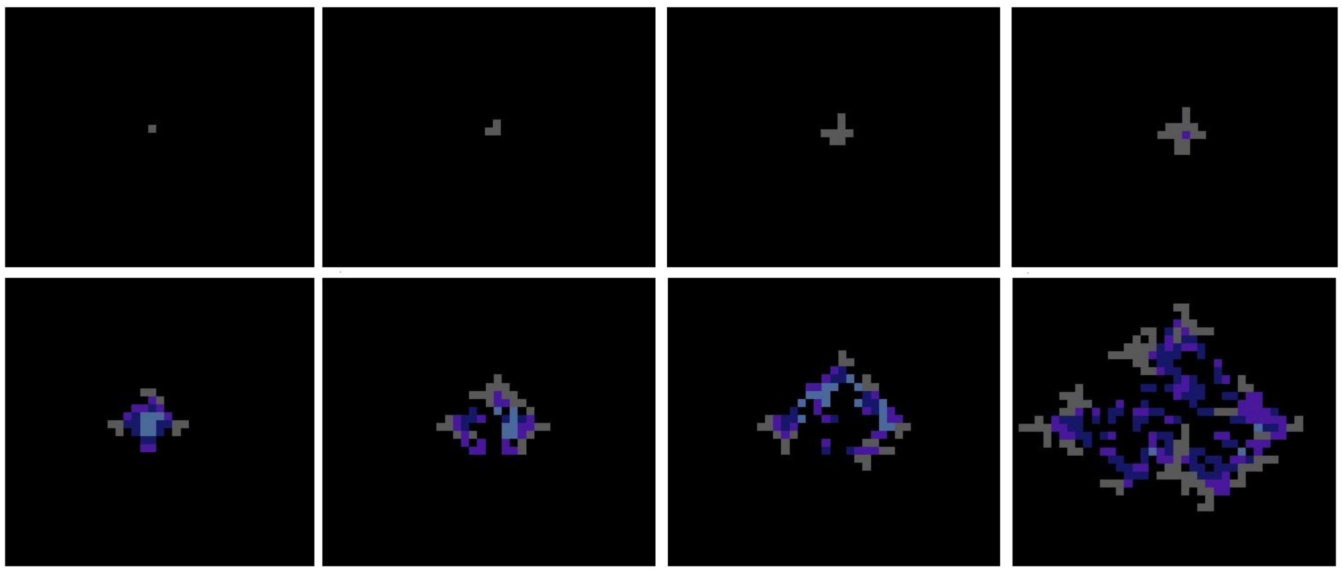
A Deva Algorithm

Consists of:

- A Developmental Space, $D \subset \mathbb{Z}^2$, with discrete time
- A set of cell types (colours) C, $|C| = n_c$
- A set of cell actions A
- A transition function, $\phi : N \to A$, where N is a description of a neighbourhood of cells from C.

A Deva Algorithm

Starting from a single cell in D, the cells execute actions from A, leading to some sort of growth.



Deva Growth Algorithm

Time $t \leftarrow 0$

Initialize developmental space D_t

while $D_t \neq D_{t-1}$ **do**

  $t \leftarrow t + 1$

  $D_t \leftarrow D_{t-1}$

  for all Cell $c \in D_{t-1}$ **do**

    if c has sufficient age and c_{r_c} **then**

      Action $a \leftarrow \phi(\mu_c)$

      Decrement c_{r_c} appropriately for a

      Execute a in D_t

    end if

  end for

end while

Deva 1 Cell Actions

- *Nothing*, the empty action
- *Die*, which removes the cell
- *Divide*, which creates a clone of the cell
- *Specialize(x)*, which changes the cell's specialization to $x \in C$
- *Elongate*, which causes the cell to elongate

Deva 1 Transition Function

Cell-based: each cell decides next action.

Maps from a description of a cell's neighbourhood to an action, $\phi(\mu_c) = a$

Description is a count of cell types in the (extended von Neumann) twelve-neighbourhood,

$$e_1, ..., e_{12}$$

Deva 1 Transition Function con't

ϕ consists of a listing of $|\phi|$ descriptions, associated with actions.

$$(c, h_1, ..., h_{n_c}, a)$$

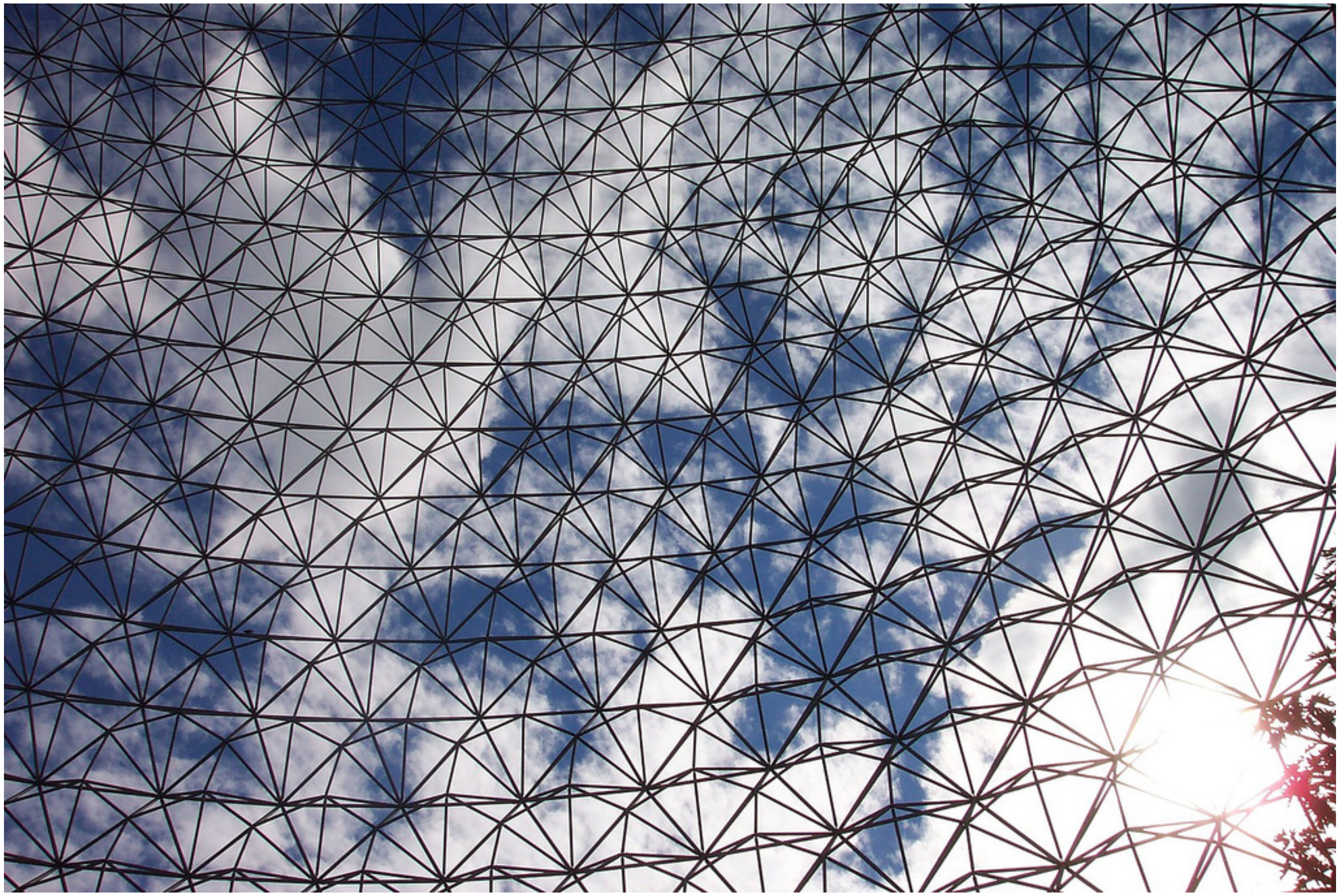
Given a description of the nbhd of a cell c, we may match the closest pattern:

if $r_{colour} = colour(c)$ **then**
$\quad distance \leftarrow (e_0 - r_{h_0})^2 + ... + (e_{n_c} - r_{h_{nc}})^2$
end if

Size of the representation of ϕ: $O(|\phi| \cdot n_c)$

Number of possible transition functions: $n_c \cdot 12^{n_c} \cdot |A|$

Truss Interpretation



Plane Trusses

Simple models of structure

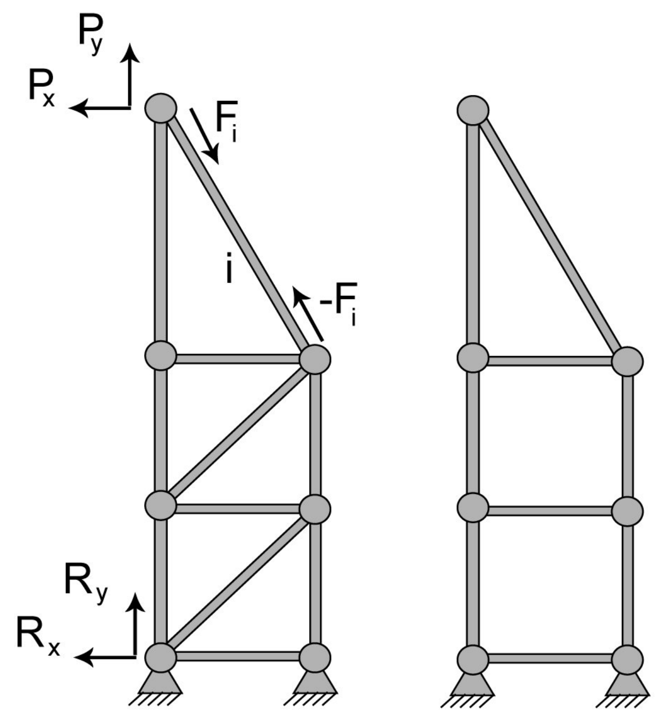
Good approximations of bridges, towers, etc.

Often form initial design stage of construction.

Plane Trusses con't

Consist of beams, joints, grounds.

Want: stability, ability to withstand (distribute) external force.



Evaluation

Let:
$\{P\} = \{P^1, ..., P^n\}^T$, external forces
$\{\Delta\} = \{\Delta^1, ..., \Delta^n\}^T$, displacements
$\{F\} = \{F^1, ..., F^m\}^T$, member forces

$$\{F\}^i = [k]_a^i[\beta]^i\{\Delta\}$$

where $[\beta]^i$ is the connectivity matrix for the ith member beam, $[k]_a^i$ is its stiffness matrix

$$\{\Delta\} = [K]^{-1}\{P\}$$

Evaluation con't

The brunt of the work is in computing $[K]^{-1}$. This takes $O(m^3)$ (or slightly better, with LU-Decomposition)

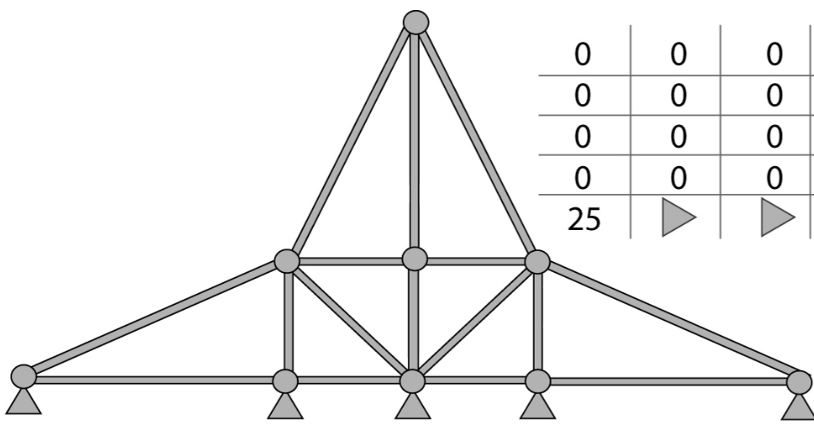
If K is non-singular, then our truss is unstable.

Otherwise, we may compute the pressure in each beam, and compare to material strength.

From Cell Types to Trusses

By considering the upper five directions as "genes", we can map between combinations of beams and integers in $\{0, .., 32\}$.

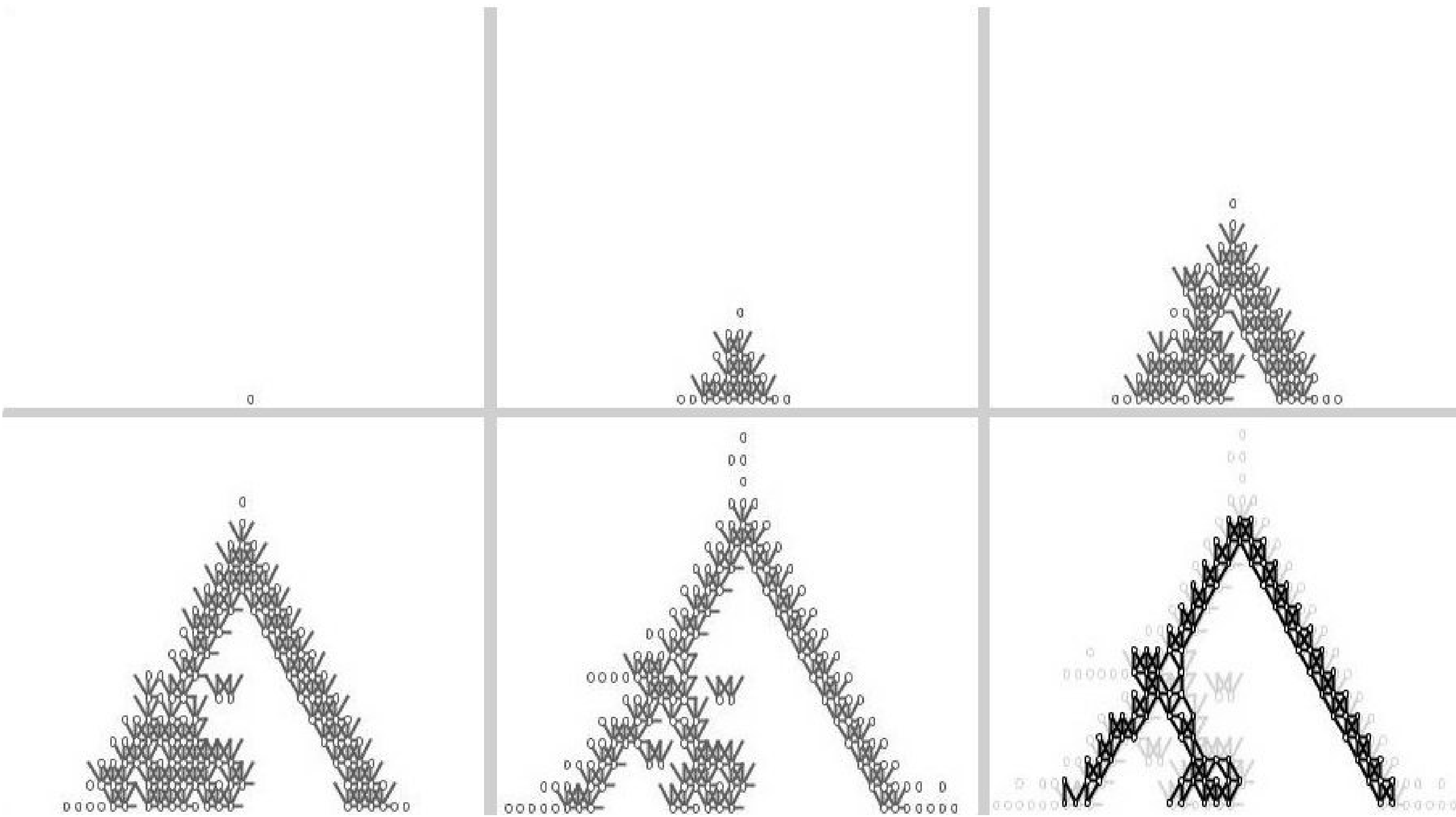
Elongations can serve to lengthen in some given direction



0	0	0	0	1	0	0	0	0
0	0	0	▲	▲	▲	0	0	0
0	0	0	▲	▲	▲	0	0	0
0	0	0	25	5	4	0	0	0
25	▶	▶	1	32	1	◀	◀	4

From Cell Types to Trusses con't

We also trim useless and redundant beams.



Evolutionary Algorithm

Representation

An organism is represented by its transition function, ϕ.

Each rule can be represented by a sequence of integers in appropriate range:

$$(c, h_1, ..., h_{n_c}, a)$$

ϕ consists of $|\phi|$ such rules.

We can define genetic operators for a list of integers.

Initialization

A listing of uniform random values for a count of cell types is a bad choice.

Want a distribution, X, that gives $E[n_c X] \approx 6$

We use a power-law distribution:

$$Pr[X = i \mid 0 \leq i \leq 12] = \frac{1}{\sum_{j=0}^{12} \beta^j} \beta^{12-i}$$

where $\beta \approx 3.6$ a good choice for $n_c = 32$

Fitness

- t penalize "trivial" forms
- h reward height
- m penalize materials used
- s reward stability
- b penalizes large bases
- p reward trusses that survive external forces, further rewarding lower maximum pressure: 20 kN down and 5 kN right at the highest joint(s), 50 N down and 50 N left at all other joints.

Fitness con't

$$f_{mat}(T) = t(T) \cdot h(T) \cdot m(T) \cdot s(T) \cdot p(T)$$

$f_{stoch} = f_{mat}$, save that the external force is applied to random joints.

$$f_{base}(T) = t(T) \cdot h(T) \cdot b(T) \cdot s(T) \cdot p(T)$$

Results of Evolution

Trials

We ran 10 trials for each fitness function, using two different phenotypic sizes (16 m, 24 m).

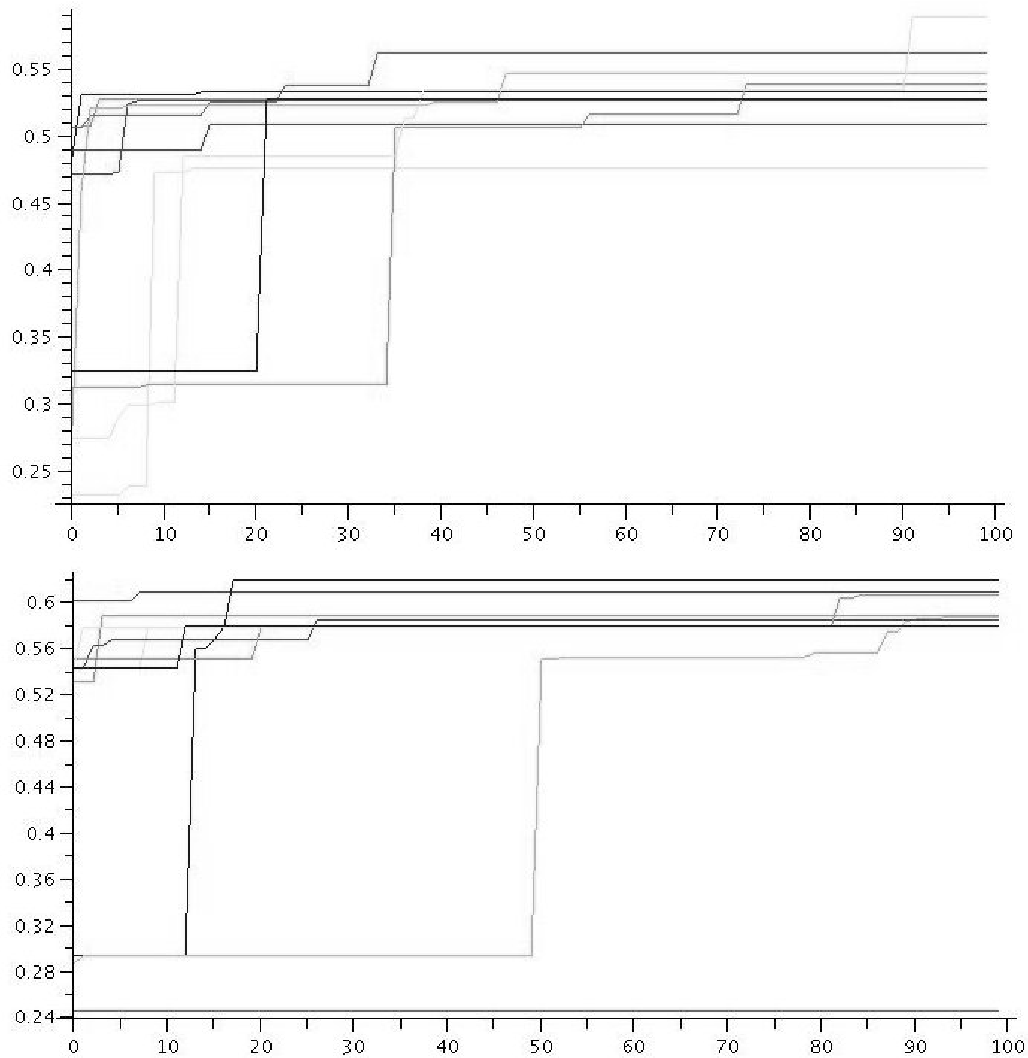
Stable trusses capable of supporting the external load were found in nearly all runs.

Some general trends were seen in runs using the same fitness functions.

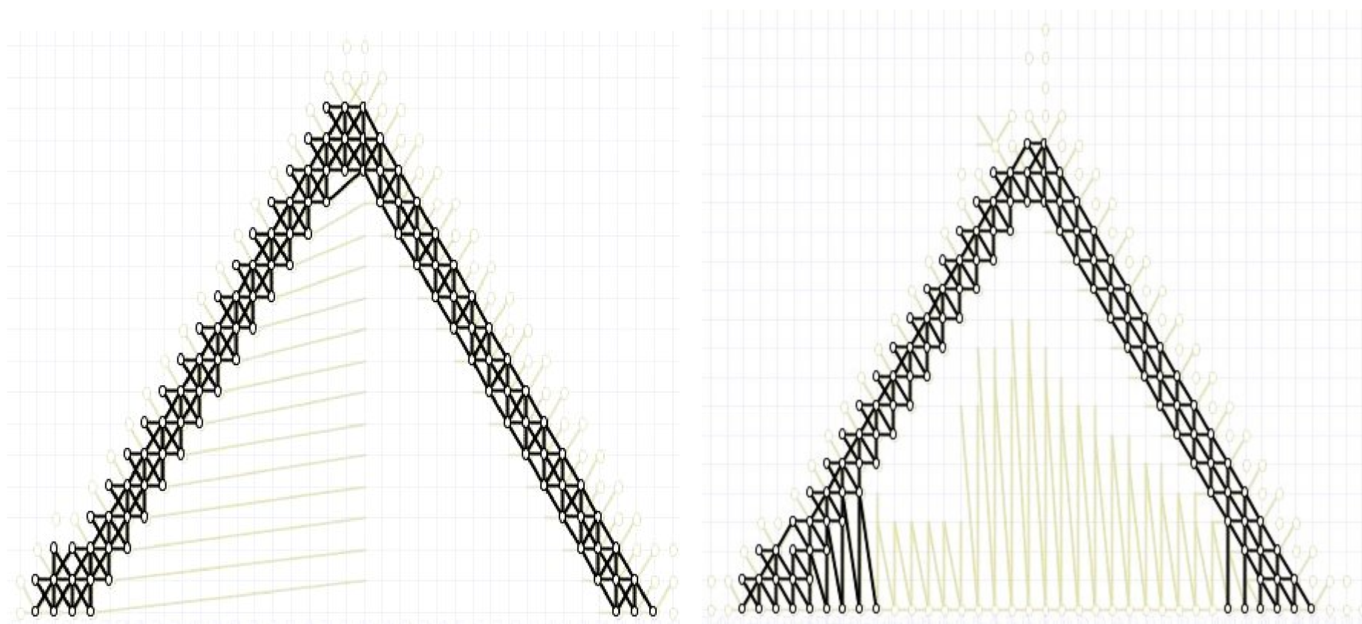
Different fitness functions tended towards different solutions.

Fitness Plots

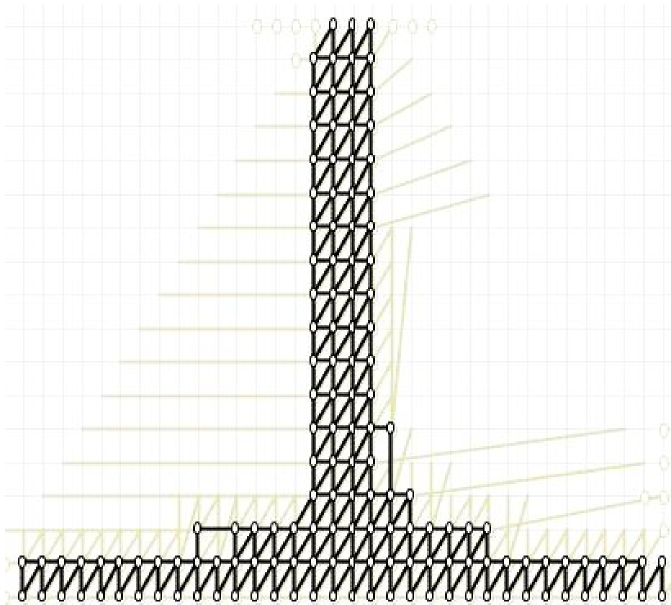
Plots of fitness versus generation for f_{mat} and f_{base} functions.

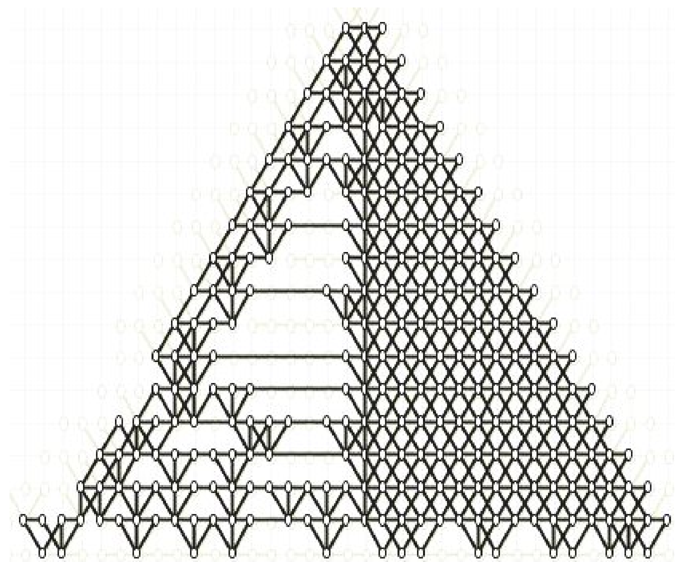


Exemplars for f_{mat}

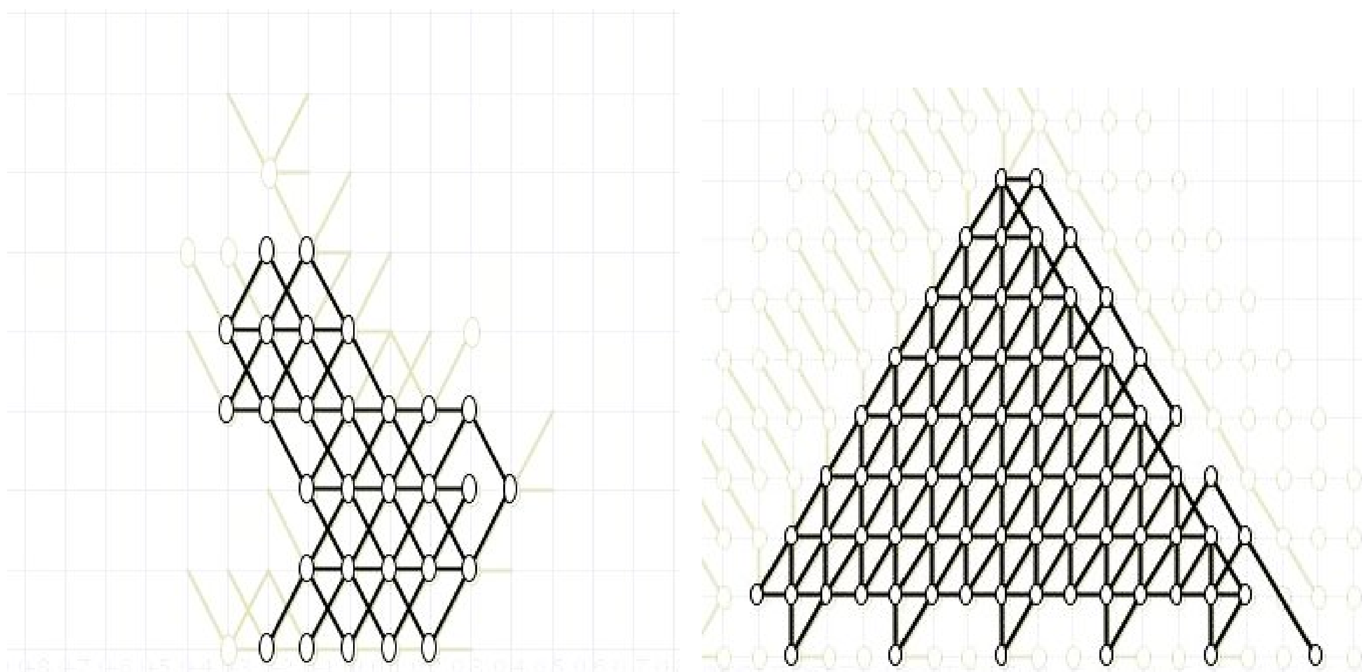


Exemplars for f_{stoch}





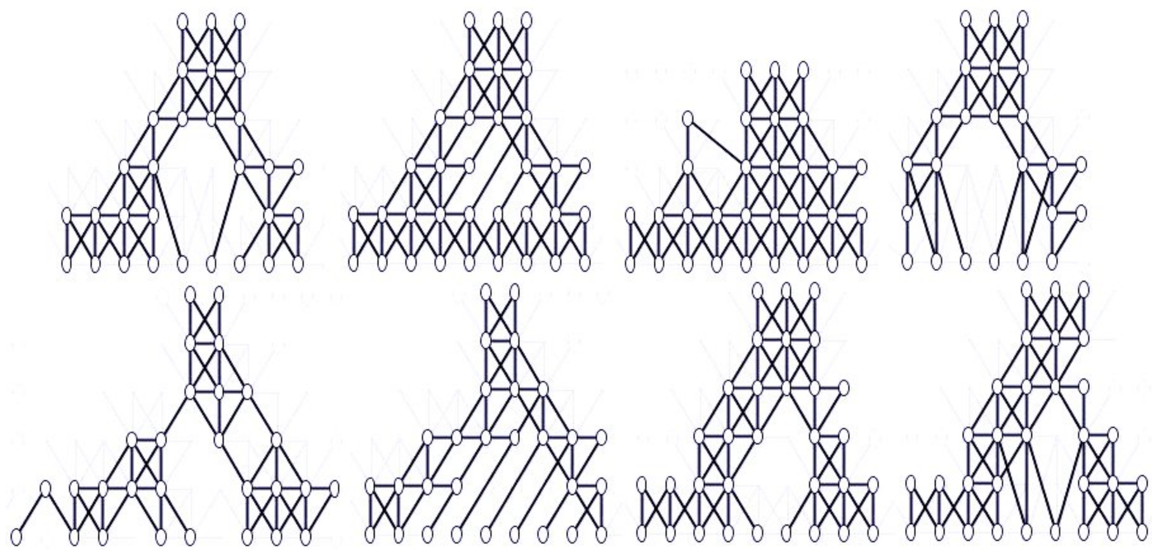
Exemplars for f_{base}



Phenotypic Continuity

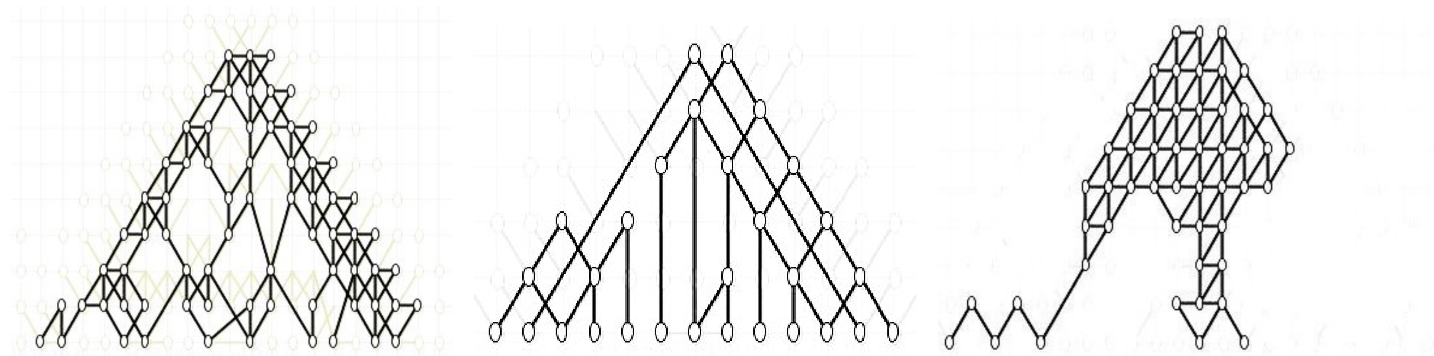
Genetic operators often destructive/impotent: suggests non-linearity in genotype-phenotype map.

However, some phenotypic continuity can be seen visually.



Unusual Solutions

Designs often do not resemble solutions that engineers would choose.



“Seed” Trials

The $useSeed$ parameter causes the first rule of a randomly initialized agent to be:

$$(1, 0, ..., 0, \text{“divide”})$$

This is a loose analogy for the initial cleavage that affects a zygote.

Designed to reduce proportion of uninteresting organisms in randomly generated populations.

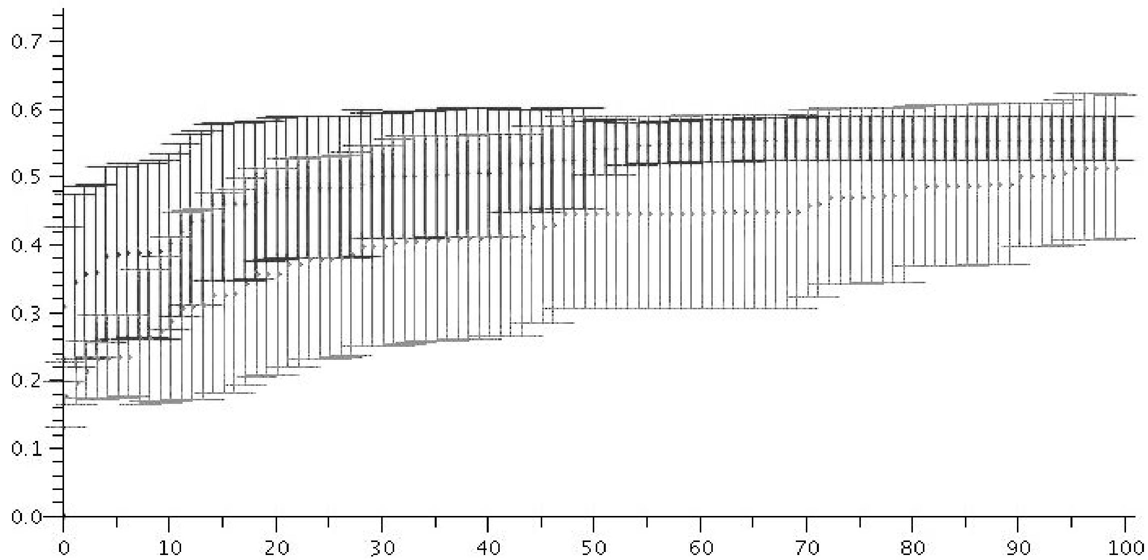
“Seed” Trials Results

We ran 20 runs with $useSeed$ enabled, 20 disabled, with a phenotypic size of 16 m.

There was a significant decrease in the number of uninteresting organisms in the initial generation.

There was slightly better performance, by fitness, in the $useSeed$ enabled trials; Also, a much smaller variance between runs.

“Seed” Trials Plot



Future Directions

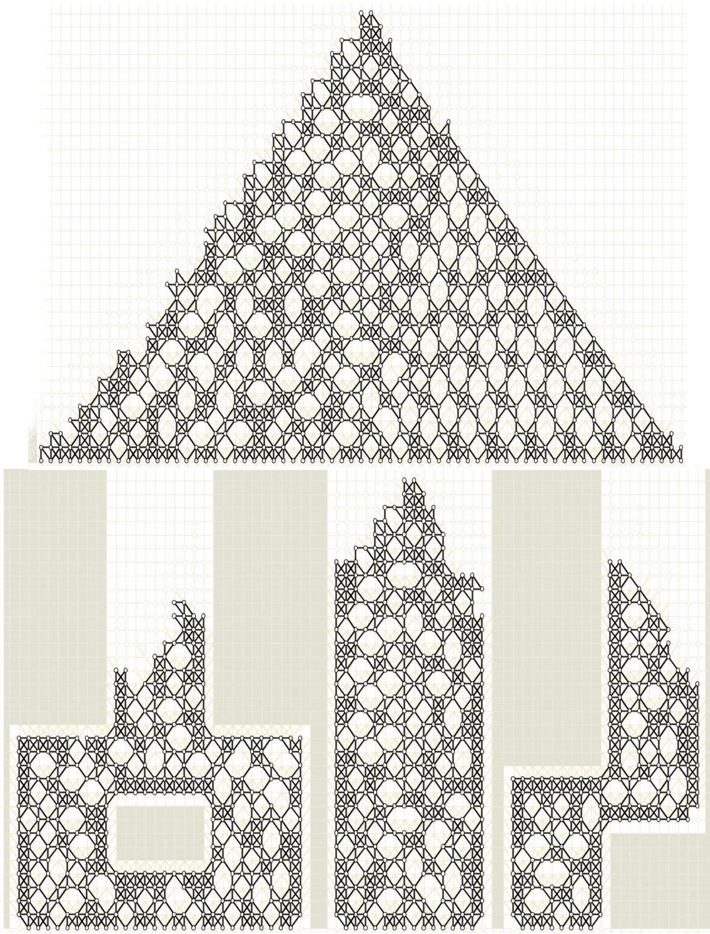
Developmental Environments

The geometry of developmental space can serve as an environmental constraint

Genomes evolved in some context can be re-grown in a different one

The robustness of AE systems to environmental perturbations may be explored.

To be presented at GECCO 2007.



Comparative Analysis

There are many models of AE in the literature.

AE is essentially the use of dynamical systems as a tool for design. Highly non-linear, probably unpredictable!

Evolutionary Computation pushes systems towards optima defined by the fitness and representation — largely development-free.

Through comparative empirical analysis, we can see the interplay between the two, and see how significant model-level decisions are.

FIN

"The Evolution of Structural Design
Through Artificial Embryogeny"

Taras Kowaliw, Peter Grogono, Nawwaf Kharma
Concordia University, Montréal, Canada
http://kowaliw.net