

The Evolution of Structural Design Through Artificial Embryogeny

Taras Kowaliw¹, Peter Grogono¹ and Nawwaf Kharma²

1. Department of Computer Science

2. Department of Computer and Electrical Engineering

Concordia University

1455 de Maisonneuve Blvd. Ouest

Montréal, Canada, H2X 1J2

taras@kowaliw.net

Abstract—We present a model of Artificial Embryogeny, *Deval*, with discussion of design rationale. The motivation for an externally-evaluated test-bed for models in Artificial Embryogeny is discussed. We review a simple model of structural design, the Plane Truss. The *Deval* model is applied to the evolution of designs for Trusses, showing that a variety of trusses satisfying distinct fitness-imposed criteria may be evolved; We evolve trusses capable of supporting external forces efficiently while either minimizing material usage or minimizing base size. Finally, we introduce a model-level perturbation, the inclusion of a *seed* rule, a metaphor for the initial cleavage found in real-world embryogeny. Through an empirical evaluation, we determine that the use of a seed rule is found to improve the overall quality of evolved designs, and to lessen the fitness variance between the runs.

I. INTRODUCTION

In this paper we introduce both a model of Artificial Embryogeny (AE), and a domain of application which may serve as a future testing grounds. We introduce *Deval*, a Cellular Automaton-like model where cells develop in a discrete space and under a discrete time. *Deval* is a general means of mapping between genotype and phenotype, where phenotype is realized in a dynamical system guided by the genome.

There exist many models of AE, and further, we believe that there are many more unexplored directions. In order to differentiate between the various models and their relative merits, there exists a need for independent and external means of evaluation. To that end, we introduce an interpretation of an organism as a Plane Truss, evaluated on its ability to form a stable structure and support external forces. Plane Trusses are common models in structural design — successful truss design is a challenging task, relevant to engineering today. It is our belief that this application forms a suitable starting point from which to evaluate various models of AE.

We test the *Deval* algorithm in the context of Plane Truss design, guiding evolution through our choice of fitness function — successful truss designs are found for each set of imposed constraints. Finally, we introduce the notion of a *seed* rule, an addition to the initial population of genotypes designed to force an initial cleavage of cells in the developmental phase. The seed rule is evaluated empirically in the context of growing Plane Trusses, and is found to increase overall

fitness and lessen the fitness variance between runs.

II. REVIEW

We review three relevant fields: Initially, we review the field of Artificial Embryogeny in a general sense; Next, we briefly introduce Plane Trusses, and discuss their evaluation through a matrix-base equilibrium method; Finally, we look at previous attempts at the evolution of structural form through Evolutionary Computation.

A. Artificial Embryogeny

There is much interest at present in the use of development in Evolutionary Computation (EC). Artificial Embryogeny (AE) is a term¹ used to describe a developmental phase in artificial evolution, that is, an indirect mapping between the genotype (representation) and phenotype (evaluated organism). It is common, although not necessary, that this mapping be inspired by biological embryogenesis.

The first computational models of embryogenesis include chemical diffusion work by Turing [21] and work with simple automata by Lindenmayer [11]. Perhaps the best known model, however, is an exceedingly simple proof of concept using recursive systems: Dawkins' BioMorphs, a simple user-guided evolutionary strategy that controls the development of stick-based trees [2]. Much research revolves around the attempt to reverse-engineer Evo Devo, or to create "plausible" models of embryogeny. These include the attempt to model plant growth by Prusinkiewicz *et al*, who have recently successfully re-created phenomena including meristems, branching, and tropism, amongst others [14], [15]. Kumar and Bentley present a system called the Evolutionary Development System, consisting of pre-programmed concepts of proteins, cells, cytoplasm, genes, etc., intended to model gene expression with cis-regulatory regions. The EDS has been shown to be capable of mimicking Genetic Regulatory Networks within cells, with some ability to grow embryos of particular shapes [10]. Eggenberger Hotz has supplied a model which includes significant modelling of the environment in which the cells

¹We follow the naming convention introduced by Stanley and Miikulainen [19] for reasons of pure aesthetic preference — other similar terms are "Computational Development" and "Artificial Ontogeny".

grow as well, including the force of cells on each other — this simulates the growth of three-dimensional structures, including invagination [3].

Most relevant to our current interest are cases where AE has been applied to the design of solutions to problems from engineering and related fields. AE possess several attractive properties which imply their potential use in situations where direct encoding might be impossible or intractable. AE techniques are believed to be capable of exploiting a canalization of development, allowing for the design of organisms too large for evolution via bijective encodings [9], [7]. AE is believed to be a mechanism by which large complex systems may maintain themselves, executing self-repair following damage [13]. AE allows for significant environmental influence on the development of organisms, allowing for the same representations to be used in the development of several different organisms [9]. Finally, it has been suggested that AE might be used to generate not only the final organism, but also a constructive map, detailing a plan for the assembly of the final design [17].

Eggenberger Hotz *et al* have used development to grow neural network architectures of impressive size and complexity, capable of controlling a robot arm intelligently [4]. Sekanina and Bidlo used evolution and a developmental algorithm to evolve sorting networks; Through their developmental scheme, they were able to create a series of ever-larger networks, reasonably independent of genome length [18]. Stoy and Nagpal use a Cellular-Automata-like technique to allow an undifferentiated mass of components to self-organize into a pre-determined shape; They have recently described the reconstruction of a CAD 3D model of a plane from an arbitrary configuration of atomic components [20].

There are two existing models of embryogenesis closely related to Cellular Automata (CA). In Basanta’s model, EmbryoCA, a single seed cell is placed in the centre of a two-dimensional grid, with successive growth controlled by a modified two-dimensional CA. Basanta *et al* demonstrate the ability of their model to grow simple geometric shapes [1]. The work of Kowaliw *et al*, Bluenome, was an attempt to design a subset of all CA more suited to evolutionary pressures — Bluenome was shown to be capable of developing two-dimensional images maximizing several interesting fitness functions. Additionally, Bluenome was also applied to the problem of evolving an organism capable of surviving in an artificial two-dimensional world, locating, consuming and distributing food — several interesting agent strategies were discovered in this context. This current work may be viewed as an extension of the Bluenome work, with a re-designed and generalized model of embryogeny, and located in a more significant domain of application [9].

B. Trusses

Trusses are well studied examples of structural design, being used by architects and engineers in nearly all construction; Often, they are cited as the simplest such model. Still, as an approximation of actual real-world structures, trusses are

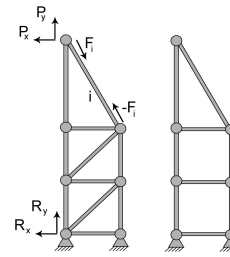


Fig. 1. Two plane trusses, the left is stable, the right unstable. Labelled on the left: external force P applied to the top joint, reactive force R from a base joint, member force F_i of the i th member.

close enough to be suitable models for most small construction projects, and are typically used at least in the initial design phase of nearly all large construction. Truss-based structures invisibly form the basis of nearly every large building or tower, but are most obviously visible in bridges, hydro towers, house roofing — although a simple model, truss design can be exceedingly complex. As such, trusses are an appropriate choice for evaluating a model’s ability to perform structural design, allowing for an evaluation of those designs from a completely independent context.

1) *Plane Trusses*: Plane trusses are two dimensional constructs consisting of (for our purposes) joints, beams and grounds. A truss is any connected collection of these three components, regardless of usefulness or triviality. All beams are connected via joints, which may be connected to grounds. The typical purpose of a truss is to support other structures, and to re-distribute any external forces so as to retain its original form; Hence, we typically talk about the stability of a truss, and the stress on any particular component.

Given some truss, our natural first question is whether or not it is stable — i.e., will it (approximately) retain its shape. The second question involves the stress placed on the members under some external force — if the maximum stress exceeds the yield strength of any particular beam, the truss may quickly become unstable. Another important issue involves the deformation of the truss members under strain; Given some beam and an external force, a beam will either compress or stretch, which in turn will cause the truss’ joints to dislocate. Some dislocation is to be expected, indeed, some dislocation is precisely the advantage to the use of joints rather than rigid connections. Too much, of course, will compromise the design. Figure 1 shows two trusses; The first is stable, but the second is not — any external force would cause the second to deform drastically.

In designing a plane truss, we typically operate under some set of criteria, and attempt to design a truss which can maximize those criteria while being both stable and as resistant to external forces as possible. We will assume, for all future discussions, that our trusses are topologically connected, pin-connected, friction-free, and that force is applied only at joints.

2) *Truss Stress Analysis*: We now examine the computation of member-forces in an arbitrary plane truss². There exist

²This analysis is taken largely from West’s treatment [22].

some simple counting tests that may determine if a given truss is unstable. Failing that, we must attempt to compute the equilibrium state given some external forces — in the process, we obtain values for all member forces. In our example, all truss members are identical in terms of material and area, grown in a developmental space where units (and hence also r_c) are measured in meters; We set $EA = 1.57 \times 10^4$ N, corresponding to a modulus of elasticity for steel [12] and a cylindrical member of diameter 1 cm.

Consider a general truss with n joints and m beams; We are provided with external forces to be applied at joints, and wish to determine the member forces. Let our structure forces be $\{P\} = \{P^1, \dots, P^n\}^T$, structure displacements be $\{\Delta\} = \{\Delta^1, \dots, \Delta^n\}^T$ and member forces be $\{F\} = \{F^1, \dots, F^m\}^T$. We may relate the individual member forces to displacement and structure forces as follows:

$$\{F\}^i = [k]_a^i [\beta]^i \{\Delta\} \quad (1)$$

where $[\beta]^i$ is the connectivity matrix for the i th member beam, and $[k]_a^i$ is its stiffness matrix, relating the deformation of the beam under a given force to the displacement at the joint. Hence, to solve for forces, it suffices to compute the displacements. The displacements may be computed through a truss stiffness matrix, a combination of the individual member stiffness matrices:

$$\{\Delta\} = [K]^{-1} \{P\} \quad (2)$$

Hence, given a plane truss, we may first compute the stiffness matrix, then compute the displacements, then the individual member forces. The entire process is bounded by the calculation of a matrix inversion (or LU-Decomposition), and hence has running time $O(m^3)$.

C. Evolution of Structures

There has been significant interest in the evolution of structural designs. This has included several frameworks for their analysis, including plane and space trusses, simplified models of Lego, and others. The Lego and related simple models have led to some interesting research in design, including the early development of buildable structures [6], or, more recently, the use of AE for the design of a simple arch, including scaffolding [17]. However, since we desire a notion of structural design which may be evaluated through means external to the A-Life community, we will instead concentrate on models taken directly from Engineering.

There have been many attempts to use EC for structural design — an extensive recent review was conducted by Kicinger *et al* [8]. Typically, use of EC in structural design concentrates on optimizing the sizing or shape of existing frameworks — our work, however, involves topological design. Use of a GA to optimize a topological design through a relatively bijective relation between genotype and phenotype has been conducted by Rajan [16] (who also optimized sizing and shape). A more complex approach was undertaken by Yang and Soh, who used a GP approach to optimize topology in the context of tall buildings [23].

III. A DEVELOPMENTAL ALGORITHM

In this section we define the Deval model, then describe an interpretation of developed organisms as Plane Trusses. Finally, we describe the evolutionary engine.

A. The Deval Model

Let us consider a model which consists of a *developmental space*, D , a collection of *cell types* (or *colours*), C , a set of *actions*, A , and a *transition function*, ϕ . The developmental space, here simply $D = \mathbb{Z}^2$, is a space in which we may grow an organism, endowed with a discrete time. Each point in the lattice is a cell, possibly the empty cell — each non-empty cell may be viewed as an independent agent. Cells change in time by executing one of several actions; Which action is executed is determined by the cell's genome, the transition function.

We now describe the process of growth: developmental space is initialized empty everywhere, save the central point, which is initialized with a cell of type "1". At every time step, any non-empty cell examines its neighbourhood, and selects an action through the consultation of the transition function. If the cell has sufficient resources (measured via an internal counter, r_c), and has sufficient age, that action is executed. The set of possible actions is $A = \{\text{nothing}, \text{divide}, \text{elongate}, \text{die}, \text{specialize}(X)\}$, where $X \in C$. Through this process, the developmental space changes in time — termination occurs when the space is identical to the space that preceded it (guaranteed to occur due to a finite maximum value of r_c). This process may be written more explicitly as:

```

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

```

A *deval* transition function is a listing of descriptions of possible neighbourhoods of a specified length, $|\phi|$. These rules are tuples of the form:

$$(c, h_1, \dots, h_{n_c}, a)$$

where c is a colour, $n_c = |C|$ is the number of cell types, a is an action, and h_i is a count of the number of neighbours of cell type i , or a *hormone-level*. Hence, the size of the representation of such a transition function is $O(|\phi| \cdot n_c)$, and the number of possible transition functions³ is $n_c \cdot |\mu|^{n_c} \cdot |A|$, where A is the set of all actions and $|\mu| = 12$, the size of a neighbourhood.

³Note that many transition functions will be functionally equivalent.

The output of ϕ may be computed as follows, given a current cell c_0 and its neighbourhood μ_{c_0} :

```

int  $e_i \leftarrow 0$  for all  $e_i \in \{e_0, \dots, e_{n_c}\}$ 
for all Cells  $c \in \mu_{c_0}$  do
  int  $type \leftarrow cellType(c)$ 
   $e_{type} \leftarrow e_{type} + 1$ 
end for
Rule  $r_{min} \leftarrow \emptyset$ 
int  $minDistance \leftarrow \infty$ 
for all Rules  $r \in \phi$  do
  if  $r_{colour} = colour(c_0)$  then
    float  $distance \leftarrow (e_0 - r_{h_0})^2 + \dots + (e_{n_c} - r_{h_{n_c}})^2$ 
    if  $distance < minDistance$  then
       $minDistance \leftarrow distance$ 
       $r_{min} \leftarrow r$ 
    end if
  end if
end for
return  $a_{min}$ 

```

The null action is interpreted as “nothing”. The running time of a transition function lookup is hence $O(|\phi|)$.

Cell actions are the sole means through which the developmental space changes in time. The possible actions are:

- *Nothing*, the empty action
- *Die*, which removes the cell, leaving an empty location
- *Divide*, which creates a clone of the cell in the best free location
- *Specialize(X)*, which changes the cell’s specialization from one cell type to another, $X \in C$
- *Elongate*, which causes the cell to elongate in the direction of previous elongation, or, if un-elongated, in the best free location.

The best free location is defined to be the empty adjacent location (in the von Neumann neighbourhood surrounding the cell) which lies opposite to the greatest mass of non-empty cells (in the Moore neighbourhood)⁴. Most cell actions come with a cost, decrementing a cell’s r_c — this is meant to incorporate the notion of finite resources. If a cell cannot execute an action (no best free location, insufficient resources), it does nothing.

A Deval growth is controlled then through a genome (transition function), and several system parameters (number of cell types, n_c , initial setting of resource counter, r_c)⁵. See Figure 2 for one possible example of Deval growth.

B. Interpretation as Plane Trusses

We begin by defining a representation of general plane trusses constructively on a lattice. We aim to be able to map a specialized lattice of integers to some (possibly trivial or useless) plane truss.

⁴In the case of a tie, we select the left-most location, then clockwise.

⁵Of course, choice of developmental space may also be a factor, possibly imposing constraints on organism shape; We will not consider this interesting possibility here, instead assuming that the space is empty and sufficiently large.

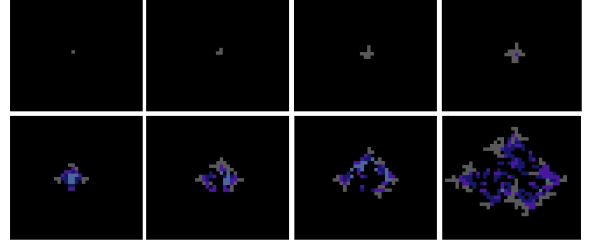


Fig. 2. An example of growth in developmental space by a Deval algorithm, where cell type is indicated by colour, black being empty. (left to right, top row): time 0; 2; 4; 6; (bottom row): time 10, 20, 30, 40.

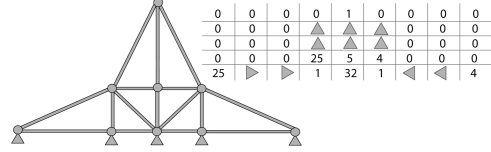


Fig. 3. Example of a translation between a lattice of integers and a plane truss. The triangles are interpreted as elongations of the adjacent cells.

Firstly, we define a set of cell types — each non-empty cell will contain a joint, and between zero and five beams. The beams will extend in directions π , $3\pi/4$, $\pi/2$, $\pi/4$ and 0 , labelled g_0 through g_4 respectively. Conversion from boolean gene values to an integer is accomplished through the following equation:

$$colour = 2^4 g_4 + 2^3 g_3 + 2^2 g_2 + 2^1 g_1 + 2^0 g_0 + 1$$

The zero cell type is reserved for the empty cell, the one value is for a joint with no beams, and all other combinations exist in the set $\{2, \dots, 32\}$.

Finally, we may allow cells to be elongated in one direction, by an arbitrary number of cell lengths. For example, a cell of type 9 has an angle of $3\pi/4$ with the x-axis, and a length of $\sqrt{2}$; A single elongation in the y-direction would lead to a length of $\sqrt{5}$, and an angle of $7\pi/8$ with the x-axis.

Hence, excluding elongation, any two-dimensional lattice of integers may be mapped to (some) truss⁶. One such mapping, including elongations, is shown in Figure 3.

In Figure 4, the growth of an agent is shown (in grey), whereas the final organism (in black) is much smaller — this is the result of a trimming process, applied to every organism following development. The trimming process serves to: (a) remove obviously unstable sections, such as beams which do not connect to joints at both ends; (b) to remove sections which are not connected to the base of the structure; and (c) to remove redundant joints, replacing them with longer beams. All three of these can be accomplished in a single pass of the un-trimmed truss structure, allowing for processing in $O(n)$ time, where n is the number of beams.

⁶Once elongation is included, we must make some constraints on the relative location of the elongation integers — this is, however, unimportant for our purposes, as deva growth would not allow for illegal combinations.

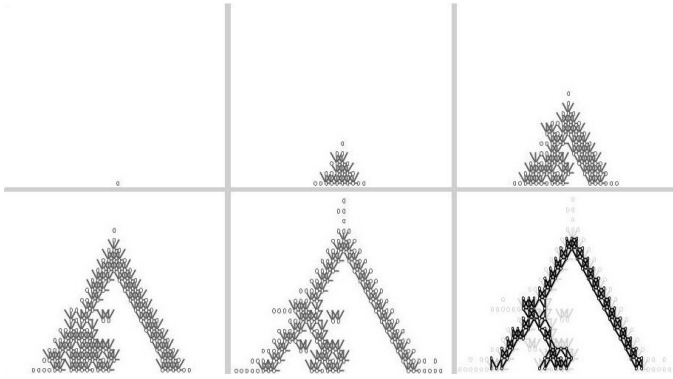


Fig. 4. An example of growth in developmental space by a Deva1 algorithm, under the plane truss interpretation (from left to right, top to bottom): time 0; time 10; time 20; time 30, time 40, the final trimmed agent at time 45.

C. Initialization and Genetic Operators

As previously mentioned, an organism may be represented by its transition function. The transition function, in turn, may be represented as a series of rules — that is, $|\phi|$ rules, each represented by $2 + n_c$ integers. Hence, a genome is simply a list of integers.

1) *Initialization*: The uniformly random initialization of a genome is not a desirable strategy, as the central section of the genome consists of a listing of hormone counts, (h_0, \dots, h_{n_c}) — selecting n_c random integers with uniform values between 0 and 12 will lead to an expected overall count of $6n_c$, far in excess of the maximum actual count of 12.

Instead, we desire a power-law distribution which favours 0:

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

We would like the expected value of the generation of any particular rule to be somewhere in the range of $\{0, \dots, 12\}$, where selecting a larger value is preferable, since the majority of computation will occur in a full neighbourhood; Hence, we take $E[n_c \cdot X] \approx 12$. Solving (numerically) the equation

$$\frac{12}{n_c} = \frac{1}{\sum_{j=0}^{12} \beta^j} \sum_{i=0}^{12} i \cdot \beta^{(12-i)}$$

we obtain $\beta \approx 3.6$ for $n_c = 32$.

We also require a probability distribution for possible actions:

$$Pr[Y = a] = \begin{cases} 1/4, & \text{if } a = \text{“divide”} \\ 1/4, & \text{if } a = \text{“die”} \\ 1/4, & \text{if } a = \text{“elongate”} \\ 1/4, & \text{if } a = \text{“specialize”} \end{cases}$$

where all colours of specialization are equally likely, save “0” or “1”, which are excluded.

We may generate a random rule by (uniformly) randomly generating an initial rule colour, then generating n_c hormones and one action according to the above distributions.

Finally, we note a system option, *useSeed*, which forces the first rule in any initialized agent to be: $(1, 0, \dots, 0, \text{“divide”})$. This rule may be viewed as similar to the undifferentiated cleavage found in the initial stage of animal embryogenesis. Note that *useSeed* has no effect after initialization, hence allowing the “seeded” rules to possibly be removed from the population, depending on the whims of evolution.

2) *Genetic Operators*: In the case of *crossover*, we use a simple single-point crossover, with the tail ends of two parents’ genomes swapped. We use two kinds of point mutation: *power-mutation*, which replaces an integer with another selected from the same distribution used for initialization; and *copy-mutation*, which replaces the current integer with another selected randomly from the current genome.

D. Evolution

The use of Deva1 for the generation of designs is controlled overall via Evolutionary Computation (EC). That is, genomes are mapped to organisms via the Deva1 algorithm, and the organisms are assigned fitness through the truss interpretation. The fitness serves to select a set of genomes for the next generation, and the actual selection and recombination is controlled through a Genetic Algorithm (GA).

We use a typical GA, as described by Eiben and Smith [5]. The GA uses elitism, as well as crossover and mutation as defined above. Selection is accomplished through a tournament of five population members, using a tournament probability of $p = 0.7$. As will soon become clear, convergence is difficult to recognize, so trials were run for a fixed number of generations. Additionally, the initial population size was larger than the population size for successive generations, by a factor of *initMult*.

The evolution of Plane Trusses may be viewed as a multi-objective evaluation; There are many factors involved in the computation of fitness. These factors, defined for a general truss T , include:

- Selection for non-triviality, where T is trivial if it contains less than five cells, or less than three cell types: $t(T) = 1/2$ if T is trivial, $t(T) = 1$ otherwise.
- Selection for height, $h(T) = h/(r_c + 1)$, where h is the raw height of T .
- Selection for minimal material use, where $m \in [0, 1]$ varies linearly between 0 for maximal use of materials ($2(r_c + 1)^2$ in beam length), and 1 for no use of materials.
- Selection for a small base, $b \in [1/2, 1]$, minimized when a minimal number of joints support the structure at the center of the space. We define the base count, b' to be the sum of the distance of the existing base joints from the center,

$$b' = \sum_{i=-r_c}^{r_c} \begin{cases} |i| & \text{there exists a joint at location } i \\ 0 & \text{ow.} \end{cases}$$

and define the base factor to be

$$b = \frac{1}{2} + \frac{1}{2} \left(\frac{\sum_{i=-r_c}^{r_c} |i| - b'}{\sum_{i=-r_c}^{r_c} |i|} \right)$$

- Selection for stability, where T is considered stable if the inverse stiffness matrix is non-singular, and if under external force, there are no absurd deformations⁷. The stability criterion is then defined as $s(T) = 1$ if T is stable, $s(T) = 1/4$ otherwise.
- Selection for distribution of pressure, $p \in [1/2, 1]$. Having applied some external force, we measure the maximum absolute beam pressure in the truss, M . If pressure has exceeded our yield limit⁸ of 165 MPa, we return $p = 1/2$; Otherwise,

$$p = \frac{1}{2} + \frac{1}{2} \left(\frac{165 \text{ MPa} - |M|}{165 \text{ MPa}} \right)$$

At every joint, we apply 50 N down and 50 N left, simulating gravity and a mild horizontal force. Additional forces are defined by the fitness function in question.

Our first fitness function, f_{mat} , is designed to maximize height while minimizing overall material. We evaluate pressure by applying 20 kN down and 5 kN right⁹ at the highest joint; In the case of several joints, the force is divided evenly between them. Hence, we seek a tall, minimal structure, capable of supporting a large mass at the top, much like a tower supporting some additional structure at the peak. The fitness of a truss T is defined as

$$f_{mat}(T) = t(T) \cdot h(T) \cdot m(T) \cdot s(T) \cdot p(T) \quad (3)$$

Our second fitness function, f_{stoch} , is similar to f_{mat} in all ways except that rather than apply external forces at the highest joint, we apply them randomly. Hence, three non-base joints are selected at random, and at each we apply 5 kN down and 500 N either right or left (with equal probability at each joint).

Our final fitness function, f_{base} is again similar to f_{mat} in that again, we apply 20 kN down and 5 kN right divided between the highest joints. However, we remove the length minimization factor and instead include the base minimization factor — hence, f_{base} selects for tall trusses capable of supporting a load at the peak occupying as little ground space as possible:

$$f_{base}(T) = t(T) \cdot h(T) \cdot b(T) \cdot s(T) \cdot p(T) \quad (4)$$

IV. EXPERIMENTAL SET-UP

Many trials of many different sizes (values of r_c) were run. We chose to investigate first the use of different fitness functions (the Fitness Trials), and second the inclusion of the *useSeed* parameter (The Seed Trials). Data was collected for runs of size $r_c = 16, 24$ — we would have preferred larger runs as there is suggestion that AE approaches are particularly suited for problems with a large number of components [9],

⁷Where absurdity kicks in at ten meters or more; This is necessary as the equilibrium process may sometimes find stable points through profoundly unrealistic stretching of materials.

⁸165 MPa is approximately 80% of the yield limit for structural steel [12])

⁹20 kN is approximately the force exerted by 2 000 kg of mass — on a single beam it exerts approximately 255 MPa of pressure, more than our yield.

but the computational expense of the repeated runs necessary for establishing a trend are great. Some runs with different values of r_c were conducted, including $r_c = 40$, the last of these having required more than one week of time on a dual-processor 2.4GHz machine.

A. Fitness Trials

Parameter settings for the Fitness Trials runs are:

population size	200	<i>initMult</i>	20
prob. crossover	0.8	rate. elitism	0.01
prob. copy-mut.	0.05	prob. power-mut.	0.05
<i>useSeed</i>	true	$ \phi $	100

There were ten runs of the Fitness Trials for each fitness and r_c setting, $r_c = 16, 24$. These runs are referred to as $fit.x.y.z$, where x is a fitness function, y is an r_c value, and $z \in \{0, \dots, 9\}$ is an index. Hence, $fit.stoch.24.3$ is the third run of the $r_c = 24$ trial using the fitness function f_{stoch} .

B. Seed Trials

Parameter settings for the Seed Trials runs are:

population size	200	<i>initMult</i>	10
prob. crossover	0.8	rate. elitism	0.01
prob. copy-mut.	0.05	prob. power-mut.	0.05
fitness function	f_{base}	$ \phi $	100

Twenty runs were executed with *useSeed* = *true*, then again with *useSeed* = *false* — these runs are labelled $us.x.y$, where x is 1 if *useSeed* = *true*, and $y \in \{0, \dots, 19\}$.

V. DATA AND ANALYSIS

A. Fitness Trials

In all trials, successful trusses were evolved — all runs found stable trusses, and 56 of the 60 found trusses capable of supporting the applied external forces¹⁰. In general, heights of approximately 9m were found in the $r_c = 16$ trials, and heights of 18m when $r_c = 24$. There were several general trends in evolved solutions for each fitness function. For the f_{mat} function, all high fitness population members somewhat resembled the exemplar, a simple triangle shaped truss. Organisms varied greatly, however, in the f_{stoch} and f_{base} runs. For the f_{stoch} function, sparse pyramids were common; Also, there were many agents with thin, tall upper sections and large bases¹¹. For the f_{base} function, some tall trusses with small, central bases were found; Additionally, large pyramid trusses with sparse bases were also common. Figure 5 show exemplar population members illustrating these phenomena.

The maximum fitness of agents in the $r_c = 16$ Fitness Trials are graphed in Figure 6. Note that frequent plateaus are found in each run, also present in the $r_c = 24$ runs; This suggests that the genetic operators are more frequently impotent or

¹⁰At least, in the case of the f_{stoch} trials, capable of supporting the forces as applied in the 100th generation.

¹¹Likely chosen for the increased probability of a random force being applied near the base, where the force is transferred easily to the grounds.

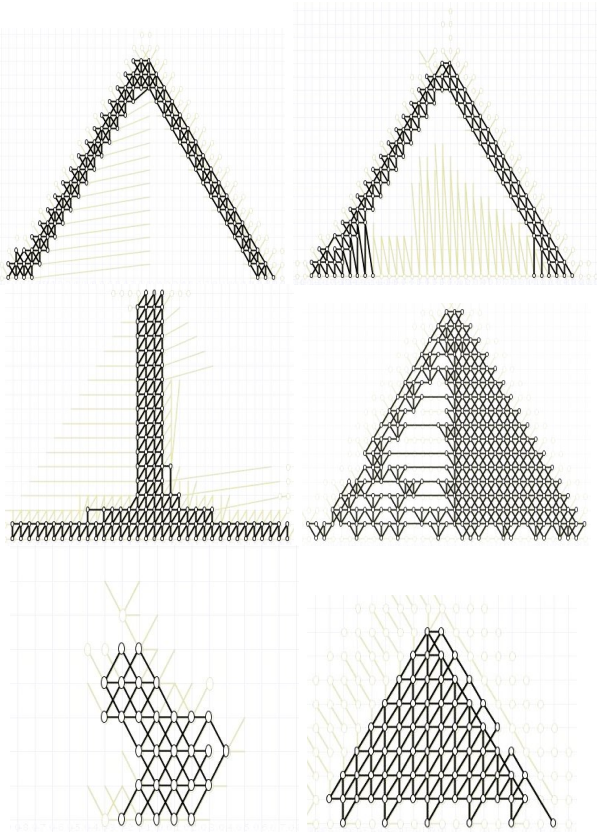


Fig. 5. Exemplar organisms from the Fitness Trials: (top) two organisms from the f_{mat} function, $fit.mat.24.4$, $fit.mat.24.1$; (middle) two organisms from the f_{stoch} fitness function, $fit.stoch.24.4$, $fit.stoch.24.3$; (bottom) two organisms from the f_{base} runs, $fit.base.16.6$ and $fit.base.16.1$

destructive, relative to more typical experiments in GAs. It is also reminiscent of theories of evolution via punctuated equilibria, where (real-world) evolution is believed to work via infrequent jumps.

As illustrated in Figure 7, a visual continuity between the phenotypes of members could typically be seen — in the example, agents show many similar qualities, including the presence of single-unit length crossbeams, hollowed-out centers, and elongated base supports.

B. Seed Trials

The simplest comparison of the us trials is a direct comparison of the fitness of the maximum agents — since all were evolved using the f_{mat} fitness function, this is a direct measure of the effectiveness of the models. Such a comparison may be seen in Figure 8, where the mean fitness of the twenty $us.0$ runs is contrasted against the mean fitness of the twenty $us.1$ runs. It is evident that there exists a large initial advantage to the $us.1$ runs — in later generations this advantage lessens, where final average fitness is within a standard deviation. However, there is a clear improvement shown overall for the $us.1$ runs, with a smaller variance.

One possible cause for the difference in fitness between the two us trials was the proportion of trivial trusses in each initial

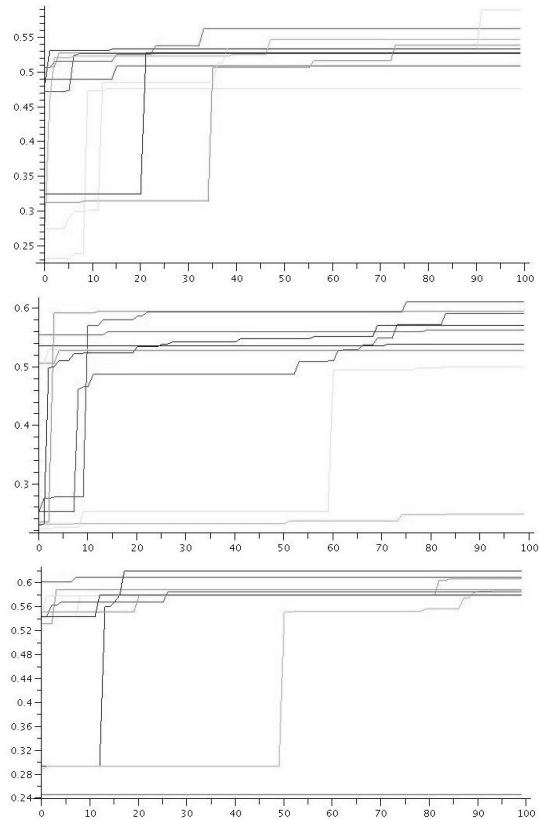


Fig. 6. Plots of fitness (y) versus generations (x). Maximum fitness of each of the (top to bottom) $fit.mat.16$, $fit.stoch.16$, and $fit.base.16$ runs.

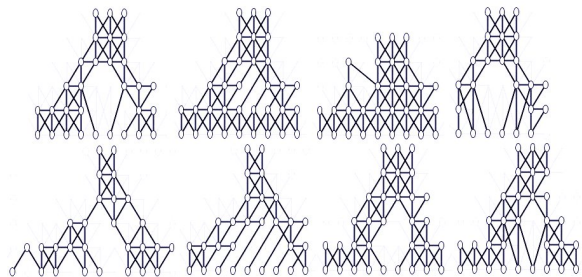


Fig. 7. The top ten (by fitness, excluding repeats) individuals of the 100th generation of an $r_c = 12$, f_{mat} run. All shown agents were stable, although the fifth and sixth had stress levels exceeding the yield of steel.

population. The occurrence of trivial organisms is common in the Deval model, typically comprising the majority of the population¹². In the $us.0$ trial, the population was 99% trivial (mean of 20 runs), compared with 95% for the $us.1$ trial. However, this initial advantage soon disappeared, as the proportion of trivial agents soon settled at approximately 60% for both trials. The initial advantage in the $us.1$ trial, allowed for a more diverse initial population as fodder for the remaining genetic search, likely explaining the initial fitness advantage.

¹²We are relatively unconcerned, since these trusses can usually be dismissed without significant processing — hence, we simply use large population sizes.

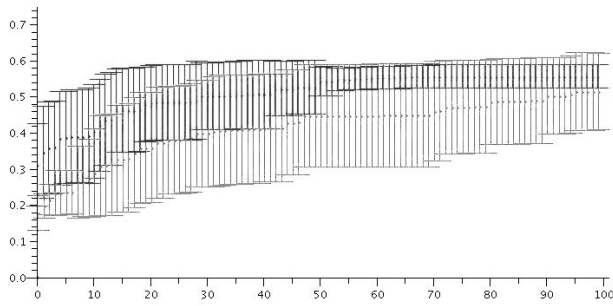


Fig. 8. Plot of fitness (y) versus generation (x). Mean fitness of the *useSeed* runs, *us.0* runs: (lighter) and *us.1* runs (darker) — the bars indicate a single standard deviation.

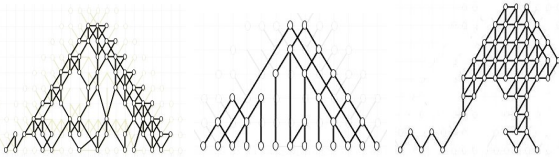


Fig. 9. Unusual organisms: (left) an organism notable for a lack of repetitive pattern, and use of odd polygons, in what could have been a simple support; (middle) a sparse pyramid in which the second highest support is completely unconnected to the first; (right) a truss which minimizes the number of base joints without minimizing mass at the top, distributing weight asymmetrically.

A final note about these trusses in general is their occasional “oddness” — it is both an advantage and a drawback of EC that evolution often finds solutions to problems which do not readily resemble human-designed solutions. In our trusses, this is often a result of a lack of right-left symmetry¹³ — in other cases, it takes the form of unusual (and typically unnecessary) complexity in solving simple problems. Figure 9 show several unusual agents, that is, agents which are reasonably functional, but bearing designs unlikely to have been created by an engineer.

VI. CONCLUSIONS AND FUTURE DIRECTIONS

We have presented a new model of AE, *Deva1*. *Deva1* has been shown to be capable of evolving plane trusses — that is, evolving designs of structure which are stable, capable of effectively distributing external forces, and also optimizing other constraints imposed by a fitness function.

Additionally, we have empirically explored a model-level addition, the *useSeed* parameter, a metaphor for the initial cleavage in real-world embryogeny. Relative to the task of designing a stable, effective, tall truss with minimal material usage, the *useSeed* strategy has proven effective — that is, inclusion of the seeded rules leads to an improvement in expected fitness generally, and a decrease in the variance between runs, suggesting a greater probability of generating a good truss design on a typical run.

The value of carrying out an empirical evaluation of a model-level feature using the above trials should not be under-

¹³It is expected that an addition to the fitness function could help induce symmetrical designs.

estimated. As the model name might suggest, *Deva1* is not the only envisioned model of AE we wish to explore. There are several other *Deva* algorithms currently being developed, we plan to evaluate them under the application described above. Given the non-linearity of recursive growth, predicting the aggregate differences (or lack thereof) resulting from model-level decisions is very difficult, perhaps impossible — considering the breadth of models currently existing in AE, empirical comparison on an accepted test problem should prove valuable.

REFERENCES

- [1] Basanta, D., Miodownik, M., Bentley, P., Holm, E. *Evolving Automata to Grow Patterns*, Proc. of the 2004 NASA/DoD Conference on Evolvable Hardware, 2004
- [2] Dawkins, R., *Climbing Mount Improbable*, W.W. Norton & Company, 1996
- [3] P. Eggenberger Hotz, *Combining Developmental Processes and Their Physics in an Artificial Evolutionary System to Evolve Shapes*, in Kumar, S., Bentley, P. (Eds.), *On Growth, Form and Computers*, Elsevier Academic Press, pp. 302-318, 2003
- [4] Eggenberger Hotz, P., Gmez, G., Pfeifer, R., *Evolving the morphology of a neural network for controlling a foveating retina and its test on a real robot*, Artificial Life VIII: 8th Int. Conf. on the Simulation and Synthesis of Living Systems, 2003
- [5] Eiben, A. E., Smith, J. E., *Introduction to Evolutionary Computing*, Springer 2003
- [6] Funes, P., Pollack, J., *Computer Evolution of Buildable Objects*, in P. Husbands and I. Harvey (eds.), *Fourth European Conference on Artificial Life*, pp 358-367, 1997
- [7] Harding, S., Miller, J., *The Dead State: A Comparison Between Direct and Developmental Encodings*, GECCO 2006
- [8] Kicinger, R., Arciszewski, T., De Jong, K. A., *Evolutionary Computation and Structural Design: a Survey of the State of the Art*, Computers & Structures, Vol. 83(23-24), pp. 1943-1978, 2005
- [9] Kowaliw, T., Grogono, P., Kharna, N., *Bluenome: A Novel Developmental Model of Artificial Morphogenesis*, GECCO 2004
- [10] Kumar, S., Bentley, P., *On Growth, Form and Computers*, Elsevier Academic Press, 2003
- [11] Lindenmayer, A., *Mathematical Models for Cellular Interaction in Development*, Journal of Theoretical Biology, 18, pp. 280-315, 1968
- [12] Megson, T. H. G., *Structural and Stress Analysis, 2nd Ed.*, Elsevier Butterworth Heinmann, 2005
- [13] Miller, J., *Evolving a Self-Repairing, Self-Regulating, French Flag Organism*, GECCO 2004
- [14] Prusinkiewicz, P., Lindenmayer, A., *The Algorithmic Beauty of Plants*, Springer-Verlag, New York, 1990
- [15] Prusinkiewicz, P., Rolland-Lagan, A., *Modeling plant morphogenesis*, Current Opinion in Plant Biology, volume 9, pp. 83-88., 2006
- [16] Rajan, S. D., *Sizing, Shape and Topology Design Optimization of Trusses Using a Genetic Algorithm*, Journal of Structural Engineering, Vol. 121, pp. 1480-1487, 1995
- [17] Rieffel, J., Pollack, J., *The Emergence of Ontogenic Scaffolding in a Stochastic Development Environment*, GECCO 2004
- [18] Sekanina, L., Bidlo, M., *Evolutionary Design of Arbitrarily Large Sorting Networks Using Development*, Genetic Programming and Evolvable Machines, Vol. 6-3, pp. 319-347, 2005
- [19] Stanley, K., Miikkulainen, R., *A Taxonomy for Artificial Embryogeny*, Artificial Life, Vol. 9-2, pp. 93-130, 2003
- [20] Stoy, K., Nagpal, R., *Self-Reconfiguration Using Directed Growth*, Proc. 7th Int. Symp. on Distributed Autonomous Robotic Systems, pp. 1-10, 2004
- [21] Turing, A., *The Chemical Basis of Morphogenesis*, Philosophical Transactions of the Royal Society B, 237, pp. 37-72, 1952
- [22] West, H. H., *Analysis of Structures: An Integration of Classical and Modern Methods*, John Wiley and Sons, 1989
- [23] Yang, Y., Soh, C. K., *Automated Optimum Design of Structures Using Genetic Programming*, Computers & Structures, Vol. 80, 18-19, pp. 1537-1546, 2002