Evolved Developmental Strategies of Artificial Multicellular Organisms

Jean Disset[1], Sylvain Cussat-Blanc[1] and Yves Duthen[1]

[1]University Toulouse 1 Capitole - IRIT
{disset; cussat; duthen}@irit.fr

Abstract

We present the use of a new computationaly efficient 3D physics model for the simulation of cells in a virtual aquatic world. In this model, cells can freely assemble and disconnect along the simulation without any separation between the development and evaluation stages, as is the case in most evo-devo models which only consider one cell cluster. While allowing for the discovery of interesting behaviors through the addition of new degrees of freedom, this 3D center-based physics engine and its associated virtual world also come with their drawbacks when applied to evolutionnary experiments: larger search space and numerous local optima. In this paper, we have designed an experiment in which cells must learn to survive by keeping their genome alive as long as possible in a demanding world. No morphology or strategy is explicitly enforced; the only objective the cells have to optimize is the survival time of the organism they build. We show that a novelty metric, adapted to our evo-devo matter, dramatically improves the outcome of the evolutionary runs. This paper also details some of the developmental strategies the evolved multicellular organisms have found in order to survive.

Introduction

Over the past two decades, the artificial life community has seen the development of several models for the simulation of environments in which cells can freely evolve. Many 2-dimensional models have been used, mainly for their simplicity and their computational efficiency, (Doursat, 2009; Joachimczak et al., 2013), but also because they are often sufficient to let interesting cellular behaviors emerge. With the addition of the third dimension come both large possibilities in the exploration of artificial life and the exciting opportunity to more precisely compare and understand real world observations. While there are several 3D physics engines and simulators developed specifically for artificial life (Joachimczak and Wróbel, 2011; Fontana and Wróbel, 2013; Doursat and Sánchez, 2014; Cheney et al., 2014), combining low scale features of cells with efficient simulation at the scale of a whole organism can prove challenging. It requires either ignoring interesting aspects of cells such as their polarisation system, complex adhesive properties or variable stiffnesses, or abandoning computational efficiency.

Of course, many models of cellular simulations are not directly linked to the aLife community (although some have been used for artificial life experiments) and are more tightly related to bio-simulation, strongly focusing on the realism of the simulations they produce. Over the years, many models have been developed using various approaches, among which 2D lattice based cellular automata (Ouchi et al., 2003), various off-lattice 3D center-based models and even precise hybrid multi-scale systems which combine cell-level deformations as well as tissue-scale constraints (Lowengrub et al., 2009), to cite just a few. In the context of artificial life, and specifically when growing multicellular artificial organisms, the complexity of the simulated world directly impacts the developmental strategies and possible morphologies of the creatures. As this can make for some behaviors and strategies that are more desirable and might also help in the understanding of real-life behaviors by bringing more realism, it also comes with at least two obvious trade-offs. First, adding realism and complexity to the artificial world will often increase the required computational power, which is a resource of prime importance when using genetic algorithms that require the simulation of thousands of instances of these worlds. Secondly, and still in the context of artificial evolution, adding complexity to the world can dramatically broaden the search space, requiring even more simulations for evolutionary algorithms to come up with a convincing organism, and further complexifying the fitness landscape. It can thus be argued that the simulation of cells for the growth of artificial multicellular organisms is, while sharing obvious common roots, a different problem than the simulation of real world cells. In this context, while we take our inspiration from biology when designing a cell simulation engine, it is of prime importance to keep these trade-offs in mind and to try and see where the truly desirable features lie, those from which an evolved multicellular organisms might benefit, and those that can be simplified.

In this work, we propose to set up artificial life experiments in a 3-dimensional world using a fast cellular physics engine tailored to artificial life, MecaCell, that offers dynamic cell-cell interactions such as collision, adhesion and

volume conservation approximation while keeping the computational cost in reasonable limits. We have designed an experiment in which the virtual multicellular organism will have to face many local optima created by both the added degrees of freedom and the rules of the world in which it evolves. We show how novelty search with a morphology metrics can, when used in conjunction with a fitness function, help overcome many of these local optima. The experiment we present in this paper challenges one cell to preserve its genetic material in a sea-like environment as long as possible. In order to do so, the cell (which can choose to eventually become an organism after division) will have to face harsh conditions where energy is a difficult resource to harvest. Organisms, or rather same-DNA cell colonies, will thus have to balance their in-water morphology to collect light energy while maintaining solid roots in the ground in order to collect a second essential type of energy. While division of labor might play a determining role in the survival of the colony (harvesting nutrients and light, sharing energy, maintaining the structure of the organism), the rules of the simulated world should make for the appearance of different viable strategies. In the lineage of our previous work (Disset et al., 2014), and to reduce the clues provided by a heavily engineered fitness function as much as possible, the cell controllers, based on gene regulation, are only evolved for survival (duration of the simulation). In addition, we study the impact of a novelty search criterion.

Simulated world

This section presents the different aspects of the simulated world we propose to investigate[1]. The main goal is to try various characteristics of the physics engine and to explore ways to mitigate the adverse effect of added degrees of freedom (comparatively to a 2D simulator or a 3D cell simulator which doesn't account for precise dynamic adhesions, for example). We want our virtual organisms to be able to evolve efficient and varied solutions to the problem of survival in a constrained environment.

Cell physics - MecaCell

MecaCell[2] aims to be an artificial life friendly and generic platform for the 3D simulation of cells. Its goal is to provide a continuous physics environment that is computationally efficient and versatile enough to tackle various ALife experiments and configurations (with exotic or simplified physics rules, for example).

[1]All the source code as well as images and videos are available at https://github.com/jdisset/seacells

[2]MecaCell is written in C++ and available (under LGPL license) at https://github.com/jdisset/MecaCell. It includes a custom OpenGL display engine with a plugin system for the extensibility of its interface.

Cell and volume conservation In MecaCell, each cell is an agent represented by a center, a membrane and an orientation. A cell can freely evolve in a 3D continuous environment, where it will collide and adhere with other cells. Here we consider cells to be spherical objects filled with a mostly incompressible fluid and wrapped in an elastic membrane. Every cell has a rest radius R_r and a dynamic radius R_d. The dynamic radius was introduced to enable an approximation of volume conservation: at each time step t, if a cell is cut (overlapping either another cell or a 3D object), we recompute both its membrane surface area A_t and its current volume V_t. The net difference in volume (relatively to its rest value V_r) is then translated into a pressure stress p_t:

$$p_t = \frac{I \times (V_t - V_r)}{A_t}$$

where I is the compressibility coefficient of the cell. Cell pressure acts as a force governing R_d's growth. When pressure increases under stress, the cell will compensate by expanding its radius in order to recover its original volume. This variation naturally implies a modification of its current membrane surface area A_t, which will also act as a shrinking force on the dynamic radius. The cell membrane is thus, in a computationally efficient manner, brought into equilibrium between volume conservation and surface area conservation, using the following explicit integration scheme:

$$R_{dt} = R_{dt-1} + \Delta t^2 \times (\Delta V - \Delta A - \frac{dR_d}{d_t} \times C)$$

where ΔV is the volume variation $V_t - V_r$, ΔA is the surface area variation $A_t - A_r$ and C is a damping coefficient.

Collisions In this model, collisions are easily handled by detecting two overlapping cells and by computing the normal and the area of the resulting contact plane. Each cell will then push on the other perpendicularly to this plane and according to their internal pressure (resulting from their deformation). The intensity of the force applied between a cell C_a of internal pressure p_a and a cell C_b (with internal pressure p_b) through a contact plane of area A_c is given by:

$$||\vec{F}|| = A_c \times (P_a + P_b)$$

$$\text{with } P_i = \begin{cases} 0, & \text{if } p_i < 0 \\ p_i, & \text{otherwise} \end{cases}$$

A tunable damping term C_{col} is also added.

Adhesions When wanting to simulate artificial multicellular organism in a 3D environment, the capability to maintain oriented connections is of prime importance. In MecaCell, cell-cell adhesions use the same kind of contact planes than for the collisions. A cell can choose its adhesive properties distribution across its membrane through the definition of an

adhesion function f_{adh} which associates an adhesive receptor density d_{adh} to a unit vector expressed in the local coordinate system of the cell (and represents the adhesive potential at a given membrane location). We simulate an adhesion between two cells by the creation of a dynamic mass-spring-damper system of length 0, attached to the centers of the contact surfaces on both cells membranes. This spring acts on both membranes but all of the generated forces and momentum is applied at the respective cells centers. When the two adhesive cells get closer from each other, the centers of the adhesion planes are updated, as well as all the mechanical properties of the adhesion mass-spring-damper system. The stiffness K and damping coefficient C are proportional to the contact plane surface area as well as the average receptor density on said surface (and to the intrinsic characteristics of these receptors, which can be different for every cells, or favor certain cell-cell affinities between cellular types). When two adhesive cells are pulled (or rotated) apart, the adhesive dynamic mass-spring-damper system can elongate up to a certain length defined by the maximum length reachable by an adhesion receptor. Thus, if the cells are pulled apart too strongly (relatively to the strength of their connection), they can actually come out of contact again. Similarly, if they experience a torque of too much intensity or a shear stress above a certain threshold, they will be able to slide on each other's membrane (the centers of their adhesion plane will have moved too far apart due to rotation).

Environment - Ground and sea

In addition to the cellular physics model presented previously, the world of this particular experiment is divided in two parts: the ground and the sea.

Ground The ground is a dense medium in which cells cannot easily move. In order to achieve this effect, we used a special integrator which does not take into account any inertia term, using only the force exerted on each cell to compute its next position. This ground acts as a solid when the forces exerted by the cells are below a certain threshold, only allowing cells to move if they push hard enough. This is, although in a simplified manner, a depiction of the mechanical characteristics of dense mud.

The ground contains nutrients, which are not available in the water. They are present in the mud at various depth, in small areas and finite amounts. At the beginning of the simulation, we initialize $N = 200$ nutrients sources. For a given nutrient source i placed at a random position (x_i, y_i, z_i) in the mud, the initial amount of nutrient n_i is given by:

$$n_i = Q_n \times (1 + C_n \times |y_i|^{P_n})$$

where Q_n is a constant and P_n and C_n are two parameters that determine how the amount of nutrient varies for each nutrient source according to its depth y_i. This is meant to mimic how the nutrient distribution can be different according to the type of soil. It also allows for the tuning of some aspects of the fitness landscape: with $C_n < 0$, the selective pressure would force the cells to expand laterally while a positive value of C_n should favor a vertical growth to find more reliable sources of nutrients. In this particular experiment, we use $Q_n = 0.03$, $P_n = 1.5$ and $C_n = 0.035$. These values have been chosen empirically in order to create an environment in which organisms can easily survive for a short amount of time but must develop complex strategies to survive longer.

Sea The second layer of the world is placed on top of the ground. We call it water, because its mechanical characteristics, namely density and viscosity, are supposed to mimic those of a still body of water. Here, a classic semi-implicit Euler integration scheme is used to update the cell positions and orientations. For computational efficiency purposes, no flows are simulated in this water. However, the cells are all slightly buoyant which means that they need to keep adhesions to cells that are still inside the mud in order to avoid being taken away.

Light is abundantly available in the water but stopped by the ground. It only comes in straight rays, perpendicular to the ground, and if one light ray shines upon a given cell, it won't be able to reach any other cell below that first one. In other words, cells block light and their shadows prevent other cells to be lit. We implemented this feature using a classical depth-buffer and depth-culling algorithm.

Cells

Cell life cycle In order to survive in this world, a cell has to fulfill one requirement: all its energy levels must stay above zero. In this particular experiment, a cell needs to handle two forms of energy: light and nutrient. At the initialisation stage, we place one unique "seed" cell in the mud, just below the water (precisely one cell diameter deep). When the simulation starts, the seed cell has maximum levels of light and nutrient, mimicking the seed endosperm (which provide the initial energy to the seed). At each time step, every cell consumes a fixed amount of light and nutrient energy. When any of the two levels of energy reach 0, the cell dies.

We implemented a simplified cell cycle in which every cell can choose between three actions: growth, quiescence, apoptosis. This lifecycle is controlled by an aGRN that will be detailed at the end of this section. When in quiescent mode, the cell consumes normal amount of nutrients and light. When choosing apoptosis, the cell will disappear and all the nutrients and light it contained will be lost. When a cell enters its growth phase, it will grow (while consuming 20% more energy) until its volume has doubled; at which point division will happen along a particular axis, determined by the cell's aGRN. When division occurs, the mother cell is replaced by two identical daughter cells whose energy

levels are exactly half those of the mother cell at the time of division. Only one variable, the age of the cell, differs between the two daughters cells: one is kept, the other is restarted at zero. This variable is incremented at each time step and is an input to the cells' aGRN.

Energy Nutrients and light are not available at the same place, which means the cells of our organism need to be able to absorb nutrients and light and share that energy with each other. More generally, a cell with large quantities of energy should be able to transfer part of it to any cell in need. In this experiment, we approximate this process through a passive diffusion based on Darcy's law, which describes the flow of an incompressible fluid throughout a porous isotropous medium in the laminar case (which is arguably the case here given the low Reynolds numbers involved). The energy (nutrient or light) flow F_n between two connected cells a and b is thus described by the following equation:

$$F_n = \frac{-k \times A \times \Delta p}{\mu L}$$

where Δp is the energy's pressure drop (here approximated by the difference in levels $n_b - n_a$ or $l_b - l_a$ where n_x and l_x are respectively the nutrient and the light level of cell x) between cell b and cell a. This flow is also determined by the intrinsic permeability of the medium k, the viscosity of the nutritive fluid μ as well as the connection area A and the distance L between the two cells centers. The value of this flow is computed at each time step for each active connection (i.e. real adhesions) between two cells using an explicit integration scheme. Using the free surface area of a cell's membrane, we also use this diffusion system to simulate the absorption of both light and nutrients from the environment. Any lit cell will perceive a light intensity proportional to its elevation (above the ground) until a certain altitude where this intensity is capped to one. Inside the ground and from any cell positioned at $\vec{P_c}$, the available nutrients concentration A_s coming from a nutrient source s at position $\vec{P_s}$, with current absolute content in nutrient C_t, initial diffusion radius of R_{t0} and an initial content of C_{t0} is given by

$$A_s = C_t \times (1 - (|\vec{P_s} - \vec{P_c}|/R_{t0} * (C_t/C_{t0})) * C_t/C_{t0})$$

Morphogens Bio-inspired communication through the diffusion of molecules in the environment has successfully been used in numerous artificial life experiment and has proven to be an efficient way to enable information transmission between agents. While some authors use detailed and realistic diffusion of signalling molecules, here we use a simple instantaneous diffusion system. Every cell can emit one or several of N_m morphogens through the m_i output protein concentration of its aGRN, and can sense the concentration of each morphogens through its c_i input proteins. The perceived intensity of a morphogen follows an inverse squared law. Thus, for any receiver positioned at $\vec{P_r}$, the perceived intensity I_m of a morphogen m emitted by N sources placed at positions $\vec{P_{si}}$ with intensity E_{mi} is given by:

$$I_m(\vec{P_r}) = \sum_i^N \frac{E_{mi}}{A_m \times ||\vec{P_{si}} - \vec{P_r}||^2 + 1}$$

where A_m is the attenuation coefficient of morphogen m. For each cell, we compute the gradient of a morphogen m as the averaged variation of its intensity along the x, y, and z axis, from one extremity of the cell to the other.

Cell adhesion In the early stages of this experiment, every cell would automatically establish a strong connection with every other cell upon contact. This led to the invariable collapsing of the morphology diversity, especially in the water part of the world, where inertia is not negligible. Indeed, as cells divide, they experience various forces that propagate along the entirety of the organism. As a result, opposite ends of an organism often come in contact, bouncing against each other; but the automatic creation of a strong connection would prevent cells to go back apart and will eventually make for the construction of an unordered blob of connected cells. In various multicellular artificial life models, this problem is avoided because the actual simulation stage, in which the organism is evaluated, is separated from the development phase, where the cells are positioned and linked without perturbations. While this simplifies things and allows for the creation of complex morphologies without the risk of discovering a spherical amalgamation of cells at the end of the evaluation, it also means that we lose some of the properties of real world organisms which can be of prime interest, especially for this experiment which aims to get closer to real world organism development: mainly self-repair and real time morphology adaptation to a changing environment. To tackle this problem, we once again take inspiration from biology by introducing a mechanism which lets the cell decide if it wants to create new connections or only keep the ones already existing and bounce off of a potential companion. This capacity, named "solidify", is managed by the cells' gene regulatory network. In MecaCell, the normal algorithm for adhesion creations between two cells is to "ask" them what are their reciprocal affinities at each time step. In order to let the cells decide when they are open to new adhesions, we add an "active connections" list to each cell that keeps track of all their "real" adhesions. At each time step, and for every cell, we compare this active connections list with a candidate list of cells that are currently colliding. A new bond is then created only if both candidates decide not to solidify. In combination with the other proteins provided as inputs to the aGRN (such as the cell age t and its mechanical pressure p), this, in theory, allows for the emergence of complex adhesions strategies.

Cell controller - aGRN Within our multicellular organism, each cell has its own gene regulatory network that controls the cell lifecycle. Even though the aGRNs are physically different in the cells of the same organism, as in nature, they share the same genetic code and thus, the same topology. When a cell division occurs, an exact clone of the mother cell's aGRN is copied into the daughter cell. In this work, the gene regulatory network used to control the cells is inspired by Banzhaf's model. This model has been designed for computational efficiency and is not meant to simulate a real biological gene regulatory network in all its complexity.

This model is composed of a set of abstract proteins. A protein a is composed of three tags: (1) the *protein tag* id_a that identifies the protein, (2) the *enhancer tag* enh_a that defines the enhancing matching factor between two proteins, and (3) the *inhibitor tag* inh_a that defines the inhibiting matching factor between two proteins. These tags are coded with an integer in $[0, p]$ where the upper bound p can be tuned to control the precision of the network. In addition to these tags, a protein is also defined by its concentration that will vary over time with particular dynamics described later. A protein can be of three different types: *input*, a protein whose concentration is provided by the environment, which regulates other proteins but is not regulated; *output*, a protein with a concentration used as output of the network, which is regulated but does not regulate other proteins; and *regulatory*, an internal protein that regulates and is regulated by others proteins.

With this structure, the dynamics of the aGRN are computed by using the protein tags. They determine the productivity rate of pairwise interaction between two proteins. For this, the affinity of a protein a for another protein b is given by the enhancing factor u^+_{ab} (resp. the inhibiting factor u^-_{ab}) calculated with the euclidean distance between protein b id tag and protein a enhancer (resp. inhibitor) tag. The proteins are then compared pairwise according to their enhancing and inhibiting factors. For a protein a, the total enhancement g_a and inhibition h_a are given by the sum of the exponential influences between the proteins. Two parameter β and δ are used to control the dynamics of the system: β affects the importance of the matching factors and δ is used to modify the production level of the proteins in the differential equation. In summary, the lower both values are, the smoother the regulation is; the higher the values are, the more sudden the regulation is. The concentrations are updated with a simple differential equation taking into account the newly produced proteins and the destroyed one. More details on the aGRN dynamics can be found in (Cussat-Blanc et al., 2015).

Table 1 describes the configuration of our aGRN input and output proteins when applied to this artificial embryogenesis experiment. A few clarifications on the role of some of these inputs and outputs is necessary. First, the sensed nutrients (c_n) input represents the actual concentration in nutrients sensed by the cell in its surrounding environment. The current nutrients level (n) input is the actual current level of nutrients in the cell. The same goes for the light intensity sensed by the cell (c_l) and the current amount of light energy accumulated in it (l).

Name	Type	Description or use
$c_{i,\forall i\in[0,2]}$	input	concentration of morphogen i
c_n	input	sensed nutrients
n	input	current nutrients level.
c_l	input	sensed light intensity
l	input	current light level.
t	input	age of the cell
p	input	mechanical pressure
$o_{i,\forall i\in[0,2]}$	output	morphogen i production
o_N	output	normalisation of o_i
$d_{i,\forall i\in[0,2]}$	output	divide along morphogen i gradient
d_n	output	divide along nutrient gradient
a	output	apoptosis
q	output	quiesence
s	output	solidify: no new adhesion
s_T	output	threshold for s activation
pd	output	perpendicular division

Table 1: List of our artificial grn inputs and outputs proteins.

The cells express their choices between division, quiescence or apoptosis through the concentrations of the output proteins d_i, a and q respectively. The protein with the biggest concentration represents the cell's choice. In addition to starting a division, the d_i outputs proteins of the aGRN also controls the cells' division plane: each d_i output protein corresponds to a morphogen, and the d_i or d_n protein with maximum concentration is used to determine the gradient (morphogen or nutrient) along which the cell must divide. If no gradient of said morphogen or nutrient is present, the axis of division is randomly chosen. The pd protein allows the cell to choose between a division along the morphogen gradient or perpendicular to it when the concentration of protein pd is greater than the concentration of the selected division protein.

The solidify output protein s controls the solidify capacity of a cell: if the concentration of protein s rises above the threshold protein s_T, the cell solidifies and will not accept any more adhesion from not yet connected cells until the concentration of protein s decreases again.

To obtain a usable GRN, both the protein tags and the dynamics coefficients need to be optimized. The next part presents the specificities of the genetic algorithm used in this work.

Evolution

One of the goals of this experiment is to explore how artificial multicellular organisms could survive in a harsh envi-

ronment without explicitly being led toward a given strategy or morphology. We want to explore the organisms that the rules of this world could create without constraining the creativity of evolution through some restrictive objective function. Therefore, the only objective for the cells is to survive as long as they can, or more precisely, to keep at least one copy of their DNA in our virtual world for as long as possible. This gives full freedom to the cells on the developmental strategies they can use and opens a wide range of possibilities of morphologies the organisms can develop. The drawback is that it also dramatically increase the search space and fills it with many local optima that pave the way to increased longevity. The evolutionary algorithm we use in this work to evolve the aGRN is based on the Gene Regulatory Network Evolution through Augmenting Topology algorithm (GRNEAT) (Cussat-Blanc et al., 2015).

GRNEAT

In this algorithm inspired by the NEAT algorithm (Stanley and Miikkulainen, 2002) and adapted to evolve gene regulatory network, the first population of aGRNs is initialized with small topologies, containing only input and output proteins. The population is evaluated standardly with a fitness promoting survival time. After a 3-player tournament selection, offsprings are crossed over using a protein alignment operator. This operator uses a genetic distance metric to compute topological distances between two aGRN proteins. Each type of proteins is processed separately. Both the input and the output proteins are treated with the same method. One of each input (or output) protein linked to a sensor (or an actuator) is randomly selected from one of the parents. The regulatory proteins are then aligned before being crossed: for each regulatory protein p_i^1 from the first parent, the closest regulatory protein p_j^2 not yet aligned is selected from the second parent. The distance between two proteins is computed as follows:

$$D(A,B) = \frac{1}{p}\big(a|id_A - id_B| + b|enh_A - enh_B| + c|inh_A - inh_B|\big)$$

where id_x is the tag, enh_x is the enhancer tag and inh_x is the inhibiter tag of protein x and p is the precision of the aGRN. $a = 0.75$, $b = 0.125$ and $c = 0.125$ are constants that weight each part of the protein properties. If the distance $D(p_i^1, p_j^2)$ is lower than a given alignment threshold σ_a, both proteins are aligned. Once alignment of all proteins has been attempted, one protein of each aligned pair is randomly selected and added to the offspring. The regulatory proteins that failed to align in both parents are also added to the offspring. This ensures that no crucial genetic material is deleted during the crossover. Finally, the dynamics coefficients are also crossed. One of the β and the δ coefficients is randomly selected from the parent genomes and used in the offspring genome.

Crossed-over aGRNs represent 30% of the offsprings. The rest of the offsprings are built using tournament selected genomes from the previous generation. All offsprings except the elite (the best genome) are then subject to mutation with a 75% rate. When mutated, a genome can be modified in three different ways: (1) delete a protein, with a 15% probability, randomly select a regulatory protein, if any, that is removed from the aGRN; (2) add a protein, with a 15% probability, adds a randomly generated regulatory protein; (3) modify a protein, with 70% probability, randomly modify exactly one parameter of the aGRN, either one protein tag or one of the dynamics coefficients.

Novelty metrics

In order to try to mitigate the adverse effects of increased degrees of freedom and numerous local optima in the morphological parameter space, we added a novelty metric as defined in (Lehman and Stanley, 2008). We combined this novelty score with our main survival objective by modifying the selection phase of our genetic algorithm: each potential parent is selected through a tournament based on either novelty or survival time, with a 50% chance. While not as complex as some other integrations of novelty in a multi-objective genetic algorithm (Mouret, 2011), this proved sufficient to harness some of the exploratory power of novelty. In this experiment, we tried three different novelty metrics, which are based on the capture (and comparison) of various aspects of a developing phenotype:

- Nm_0 is composed of three numbers: the maximum number of cells during the simulation, the maximum depth reached by a cell and the total survival time (which is also the main objective).
- Nm_1 is composed of 5 snapshots of the simulation (at times $t = 10$, $t = 20$, $t = 50$, $t = 75$ and $t = 100$). Each snapshot contains 2 numbers: the number of cells and the maximum depth of a cell at the time of the capture.
- Nm_2 is a set of 5 captures (taken at the same time steps as for Nm_1) represented as a 20×20 integer matrix. It is actually a set of pictures in which each pixel's value represents the number of cells stacked. The plane of the shot is determined through a Principal Component Analysis on the cells position (it is the most discriminant plane). This metric is meant to capture the morphologies of the organisms in all their subtleties

Results

Influence of novelty

In Figure 1, we can see the median (with first and third quartile) survival times of the best genomes evolved during 300 generations in 10 independent runs. On a 2014 high-end laptop, the average evaluation time for an individual was of 0.2s during the first 5 generations and ended at an average of

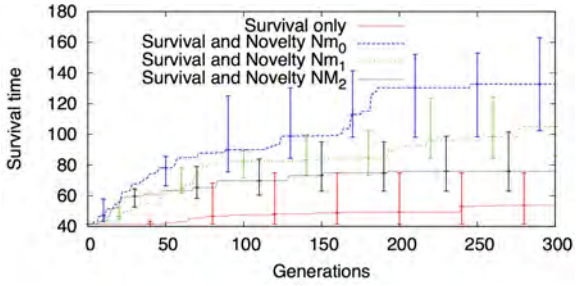

Figure 1: Error bar plots of the best individuals obtained on 10 independent runs. Error bars represents the median, the first and third quartiles. All novelty objectives are obviously helping to escape local optimum. However, the novelty measure Nm_0 is giving better results. The initial value of 41 obtained at generation 0 represents the survival time for a seed cell that stays quiescent during the whole simulation.

1.3s. The best organisms obtained with these runs are presented in Figure 2(a, c-h). This graph reveals both the deceptiveness of the fitness landscape when the survival time is used as only fitness objective as well the beneficial impact of novelty. This is undeniable (Student t-test p-values are provided in table 2): where a classical objective based evaluation struggles to find solutions that pass the first local optima (for example: not dividing and surviving on the initial resources of the seed cell, or just doing a few divisions in order for some cells to reach the surface and bring in a little bit of light), the novelty based approaches successfully find solutions to overcome these optima and efficiently pave the way to more robust organisms.

The three novelty measures tested in this experiment show that too much information loses the evolution in the vast search space: the novelty measure Nm_0 globally does better than both other measures. This measure is the one that includes the fewer parameters. In our opinion, when too much parameters are used to describe a phenotype, the exploration space becomes too large and individual with minor differences are considered too novel. Therefore, it is of high importance to wisely choose parameters that describe the phenotypes. As depicted in table 2, the relatively high p-values between novelty based runs reveal the necessity to make a broader study on the influence of the novelty parameters in order to find the best possible measures for evo-devo models and validate our preliminary results.

Developmental strategies and world setup influence

Along all the evolutionary runs, we observed an important diversity of developmental strategies and morphologies, especially when any form of novelty was involved. Figure 2(a,b) shows examples of cells arrangements obtained with different worlds parameters. The distribution of nutrients in the world was also found to be of huge influence over the preferred strategies: as expected, large values of C_n and P_n

Wins\Loses	Survival	Nm_0	Nm_1	Nm_2
Survival	-	0.002	0.011	0.089
Nm_0	0.002	-	0.360	0.050
Nm_1	0.011	0.360	-	0.250
Nm_2	0.089	0.050	0.250	-

Table 2: p-values of the paired Student t-test run comparison between runs with survival fitness and the different novelty measures calculated on 10 runs at generation 300.

favored a very vertical growth of the cell colony, with the formation of a relatively thick trunk in the ground enabling fast nutrient and light transfer between the deep roots cells and the emerged ones. One of the most interesting results might be the emergence of a form of reproduction through parthenogenesis when the nutrients concentration was uniform. Cells indeed understandably found the benefits of a vertical growth to be incomparable with the efficiency of a vertical growth. They also adopted, as shown in Figure 2(b), a spread method where they would laterally develop just below the surface. When a root cell encountered a nutrient source, it would also divide upward (to the surface) and the cells between the two formed cluster would undergo apoptosis, thus creating a simple form of parthenogenesis reminiscent of the biological reproduction of some plants.

Conclusion

We have presented a new developmental model based on MecaCell, a physics engine tailored for artificial life experiments. This model shows how novelty search can help when stepping artificial embryogenesis up to the third dimension. Indeed, this allows for more degrees of freedom for the multicellular organisms but also adds complexity for the cell controller to handle. As a result, this makes the search space much harder to explore with standard fitness function. In addition to the use of a 3D developmental model, we also wanted to remove all engineering from the main fitness objective: it only targets to the survival duration of the organisms. By only using this objective, we showed that the evolution is stuck in one or few local optima, but by adding different novelty metrics based on the organisms morphologies and capabilities to explore its environment, we showed that evolution can escape from these local optima and develop more complex morphologies and behaviors able to survive longer in the exact same environment.

This new developmental model opens many research perspectives. Firstly, we need to study more precisely the influence of the environment parameters on the multicellular organisms. During the development of the presented experiment, one of the major difficulty was to produce a viable environment, easy enough to allow the organisms to grow but difficult enough to require complex behaviors. Balancing this is difficult task and needs to be studied in detail.

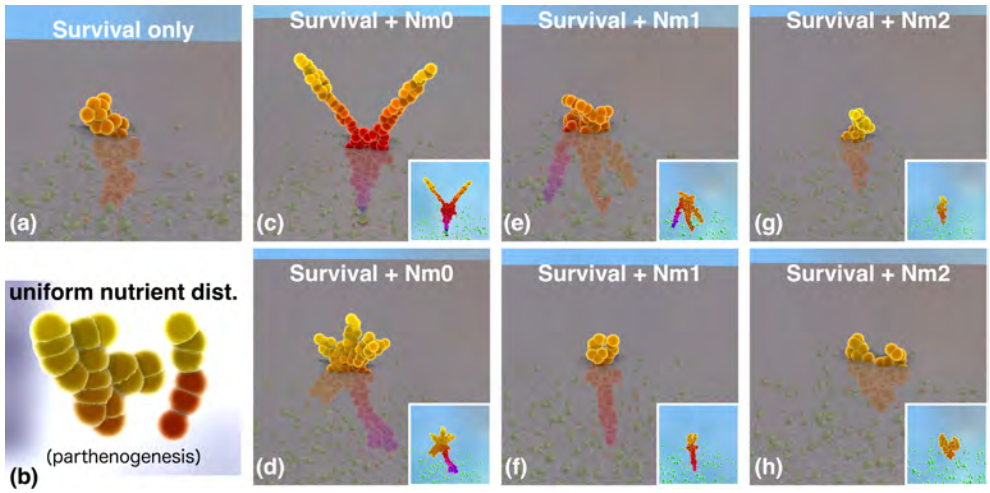

Figure 2: Examples of organisms obtained with the different fitnesses. (a, c-h) survival only and novelty metrics Nm_0, Nm_1 and Nm_2 in the novelty impact study. (a, b) same fitness with different environmental conditions.

Once done, we want to produce an artificial world in which different organisms would coexist, cooperating or competing for survival and reproduction. This will require specialization capacities of the cells in order to balance the capacities of the organisms, with for example a light extracting cell type and a reproductive one. We hope to produce more complex organisms, further mimicking some aspects of the early stages of the appearance of life on earth.

References

Cheney, N., Clune, J., and Lipson, H. (2014). Evolved electrophysiological soft robots. In *ALIFE 14: The Fourteenth Conference on the Synthesis and Simulation of Living Systems*, volume 14, pages 222–229.

Cussat-Blanc, S., Harrington, K., and Pollack, J. (2015). Gene regulatory network evolution through augmenting topologies. *Evolutionary Computation, IEEE Transactions on*, 19(6):823–837.

Disset, J., Cussat-Blanc, S., and Duthen, Y. (2014). Self-organization of symbiotic multicellular structures. In *the Fourteenth International Conference on the Synthesis and Simulation of Living Systems-ALIFE 2014*, pages pp–541.

Doursat, R. (2009). Organically grown architectures: Creating decentralized, autonomous systems by embryomorphic engineering. In *Organic computing*, pages 167–199. Springer.

Doursat, R. and Sánchez, C. (2014). Growing fine-grained multicellular robots. *Soft Robotics*, 1(2):110–121.

Fontana, A. and Wróbel, B. (2013). An artificial lizard regrows its tail (and more): regeneration of 3-dimensional structures with hundreds of thousands of artificial cells. In *Advances in Artificial Life, ECAL*, volume 12, pages 144–150.

Joachimczak, M., Kowaliw, T., Doursat, R., and Wrobel, B. (2013). Controlling development and chemotaxis of soft-bodied multicellular animats with the same gene regulatory network. In *Advances in Artificial Life, ECAL*, volume 12, pages 454–461.

Joachimczak, M. and Wróbel, B. (2011). Evolution of the morphology and patterning of artificial embryos: scaling the tricolour problem to the third dimension. *Advances in artificial life. Darwin meets von Neumann*, pages 35–43.

Lehman, J. and Stanley, K. O. (2008). Exploiting open-endedness to solve problems through the search for novelty. In *ALIFE*, pages 329–336.

Lowengrub, J. S., Frieboes, H. B., Jin, F., Chuang, Y., Li, X., Macklin, P., Wise, S., and Cristini, V. (2009). Nonlinear modelling of cancer: bridging the gap between cells and tumours. *Nonlinearity*, 23(1):R1.

Mouret, J.-B. (2011). Novelty-based multiobjectivization. In *New horizons in evolutionary robotics*, pages 139–154. Springer.

Ouchi, N. B., Glazier, J. A., Rieu, J.-P., Upadhyaya, A., and Sawada, Y. (2003). Improving the realism of the cellular potts model in simulations of biological cells. *Physica A: Statistical Mechanics and its Applications*, 329(3):451–458.

Stanley, K. O. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127.