How Artificial Ontogenies Can Retard Evolution

Shivakumar Viswanathan
DEMO Lab, Computer Science Dept
Brandeis University, MS-018
Waltham, MA - 02139
shiva@cs.brandeis.edu

Jordan Pollack
DEMO Lab, Computer Science Dept
Brandeis University, MS-018
Waltham, MA - 02139
pollack@cs.brandeis.edu

ABSTRACT

Recently there has been much interest in the role of indirect genetic encodings as a means to achieve increased evolvability. From this perspective, artificial ontogenies have largely been seen as a vehicle to relate the indirect encodings to complex phenotypes. However, the introduction of a development phase does not come without other consequences. We show that the conjunction of the latent ontogenic stucture and the common practice of only evaluating the final phenotype obtained from development can have a net retarding effect on evolution. Using a formal model of development, we show that this effect arises primarily due to the relation between the ontogenic structure to the fitness function, which in turn impacts the properties being evaluated and selected for during evolution. This effect is empirically demonstrated with a toy search problem using LOGO-turtle based embryogenic processes.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods and Search

General Terms

Algorithms, Performance, Theory

Keywords

development, evolutionary algorithms, evolvability, generative representations, problem solving

1. INTRODUCTION

In the wake of advances in experimental embryology, the early 20th century saw the eclipse of a historical controversy surrounding the concept of *recapitulation* as the underlying relation between the ontogeny of individuals in a particular species and the phylogenic history of the species.[1] By this perspective, the ontogeny of an individual in a species was treated as being a condensed trace (or a recapitulation) of the phylogenic history of that species. This was in a literal sense of the intermediate ontogenic states being conceived as the adult phenotypes of an ancestral species. Furthermore, the ontogenic process was itself considered to be adaptive in that the phenotypes occuring later in the ontogeny were treated as being 'superior' to those phenotypes that occured earlier. Consequently, the increase in developmental complexity was conceptualized as being *directed* towards an idealized superior being. This view, based on a preformationist notion of development but within a Darwinian framework, lead to Ernst Haeckel's (infamous) biogenetic law that "ontogeny recapitulates phylogeny". Interestingly, the term *ontogeny* for the developmental history of an individual, was coined by Haeckel, who was one of the most famous proponents of a recapitulationist view within a Darwinian framework.

While Haeckelian recapitulation has long been discredited in biology, it no less presents some issues that are of relevance to artificial evolutionary-developmental systems. The key relation connecting the two is the observation that many computational implementations of development satisfy one of the fundamental premises of a recapitulationist perspective. Unlike biological development, the intermediate stages of an artificial ontogeny can in fact be valid phenotypes themselves. Familiar examples of this include CA and LOGO-turtle based embryogenies [10, 8] [2]

Given that artifical ontogenies involve a temporal series of valid phenotypes, it raises the question of where such systems stand with regard to the issues of (a) the relation between ontogeny and phylogeny, and (b) the correlation between the relative positions of phenotypes in the ontogeny with their fitness. In this paper, we demonstrate that the latter relation between the ontogenic structure and the fitness function is of particular importance to evolution with development that takes a quasi-preformationist form as expressed by the Genotype-Phenotype (G-P) map $\psi : \mathcal{G} \rightarrow \mathcal{P}$, i.e. where a genotype is a 'generative representation' or 'recipe' *for* the construction of a particular phenotype.

In the light of the rich ontogenic structure involving multiple phenotypes, rather than a genotype being associated with a single phenotype, it is effectively associated with a *collection of phenotypes* with a highly structured relationship with each other. From this perspective, development that satisfies a Genotype-Phenotype

[1] All references to theories of recapitulation in this paper are drawn from Gould's authoritative volume [5] that describes the rise and fall of these theories, and his attempt to re-interpret the apparent relation between ontogeny and phylogeny in the light of the findings of modern biology.

[2] This is not always the case as the embryonic stages could contain structural elements that are never present in 'adult' phenotypes. However, this is not to be conflated with the non-terminals that are present in grammatical genetic encodings, as they are distinct from the structural entities that undergo developmental transformation leading up to the 'adult' phenotype [6]. These intermediate structural entities could be valid phenotypes.

GECCO'05, June 25–29, 2005, Washington, DC, USA.

map can now be seen as additionally involving a *decision* to pick one phenotype from the generated collection that is to be evaluated by the fitness function to compute the reproductive fitness of the genotype.[3]

The commonly used policy has been to only evaluate the final or maximal phenotype of the ontogenic process, i.e. a decision made on the sole basis of the *temporal position of the phenotypes in the ontogeny*. By the classical view of recapitulation, the fitness of this final phenotype was deemed to be greater than that of all the preceding phenotypes in the ontogeny. However, this does not hold true in general with 'hardwired' artificial ontogenies that are unaffected by changes to the fitness function. Here, we show that as a direct consequence of this property, *development can cause a retardation in the rate of evolution*.

The organization of the discussion is as follows: Section 2 provides a formal definition of the evolutionary search problem and an order-theoretic description of the ontogenic premise of recapitulation. Next we use this formal description to analytically demonstrate how the ontogenic structure can impact selection (Section 3). A toy-problem using a LOGO-turtle based embryogeny is used to empirically demonstrate the retarding effect of the final-phenotype policy on evolution (Section 4).

2. MODEL OF EVO-DEVELOPMENT

2.1 Background

Angeline [3] describes a simple formulation to express the concept of an indirect genetic encoding which can be summarized as follows.

Let $\mathcal{P}$ be the set of objects of interest. The objective is to find an object in $\mathcal{P}$ that has certain desired properties expressed as an evaluation function $\mathbf{e} : \mathcal{P} \to \mathbb{R}$ that provides a measure of the relative suitability of each object to this desired end. So the general search problem is to find a $\phi \in \mathcal{P}$ that has a value at or "sufficiently near" a desired maximum in the range of $\mathbf{e}$ using an Evolutionary Algorithm.

Suppose P is a vector representing a (multiset) collection of entities from $\mathcal{P}$ i.e. the population, and, for convenience, let $\mathbf{e}$ be such that $\mathbf{e}(P)$ produces a vector of real-numbered values corresponding to the evaluation of each of the members of P. A general evolutionary computation can be defined as an iteration of the equation:

$$P' = \beta(P, \mathbf{e}(P)) \tag{1}$$

The function β is called the *reproduction function* as it generates a new collection of entities from $\mathcal{P}$ from the previous collection P (for a given random seed). So evolution is in effect the iterative application of this function till some external stopping criterion is reached. Here, the objects in $\mathcal{P}$ are referred to as the *evaluated* entities and as they are also directly manipulated by β, they are also the *evolved* entities.

Morphogenic Evolutionary Computation (as Angeline calls it) is contrasted with this basic form by the introduction of an abstraction that separates the evaluated entities from the evolved entities. The reproduction function for a morphogenic evolutionary computation is defined as

$$G' = \beta'(G, \mathbf{e}(\psi(G))) \tag{2}$$

where, $\psi : \mathcal{G} \to \mathcal{P}$ is the *development function* (or as in more common usage, the Genotype-Phenotype map). G is a vector representing a (multi-set) collection of entities from $\mathcal{G}$, and $\psi(G)$ is considered to produce a vector P by the application of ψ to each of the elements of G. The key difference here is that the elements of $\mathcal{G}$ are the evolved entities as they are modified using β', while the elements of $\mathcal{P}$ remain the evaluated entities.

The main idea expressed by this formulation is that introducing a development function ψ enables a modification of the search problem from being primarily defined on $\mathcal{P}$ (i.e. the set of phenotypes), to being a search problem defined on a set of objects $\mathcal{G}$ (i.e. the set of genotypes). As ψ is a function where each genotype deterministically corresponds to exactly one phenotype, it follows that this change in problem definition does not result in a modification in the problem solving objective i.e. to find a $\phi \in \mathcal{P}$ that has a value at or 'sufficiently near' a desired maximum in the range of $\mathbf{e}$. Indeed, this is the very premise for considering multiple possible encodings for the same problem and the value of designing encodings that can improve the *evolvability* of the genetic representation for the problem [3, 12].

Though this formalism suggests that search on the set $\mathcal{G}$ would remain faithful to this objective, it remains to be assessed whether this is *actually* the case when the ontogenic structure is explicitly considered. Towards making this assessment, we next provide a formal description of what we specifically mean by 'ontogenic structure'.

2.2 Definition of ontogenic structure

Consider an *ontogenic computation* that takes the form

$$\phi_{t+1} = \mu_{g,t}(\phi_t) \tag{3}$$

where $\mu_{g,t}$ is the *ontogenic transformation function* for a given $g \in \mathcal{G}$. So a developmental process can be seen as being an iterative application of the ontogenic transformation function to the phenotype produced at each time step till some stopping condition is reached. Based on [5, 2], this temporally ordered sequence of phenotypic states $\phi_0 \to ...\phi_t \to ...\phi_{final}$ is referred to as an *ontogenic trajectory*, where $t = 0, ..., t_{final}$.

Clearly, the genome may itself undergo changes during this developmental process however our focus here is restricted to phenotypic changes and we assume that all these genetic effects are expressed in the temporally variable form of $\mu_{g,t}$. We assume that the development process for every genotype is guaranteed to halt so ψ is a total function.

If this operational aspect is neglected then the only change to the genotype-phenotype relation ψ would be in taking the expanded form ψ' given as:

$$\psi' : \mathcal{G} \to (\mathcal{P}_0 \to \mathcal{P}) \tag{4}$$

as expressed in its intuitive curried form, where $\mathcal{P}_0 \subset \mathcal{P}$ is the set of initial phenotype states.[4] To more transparently account for these operational properties within the evolutionary model described above, we formalize the notion of an ontogenic trajectory as follows.

Every triple $(g, \phi_0, \phi) \in \psi'$ is associated with a single *ontogenic trajectory* T. For clarity, an ontogenic trajectory T is treated here as a totally-ordered set $T = \langle P, \prec \rangle$ where P is a finite subset of $\mathcal{P}$ with $\phi_0, \phi \in P$, and $\prec$ is the "precedes" relation such that $\phi_a \prec \phi_b$ if phenotype ϕ_a appears prior to (precedes) phenotype ϕ_b in the ontogeny. So, the phenotypes ϕ_0 and ϕ are the minimal

[3]When development occurs in a rich physical environment (as in [4]), the ontogenic intermediates are in some sense being 'evaluated' at all stages. While this evaluation is of direct relevance to the viability of the developing embryo, it may be only indirectly relevant to the reproductive fitness, which is dependent on the composition of the population. So we will restrict our usage of the term 'evaluation' to imply this latter role.

[4]To maintain consistency with published work on development, $\mathcal{P}_0$ is assumed to have exactly one element $\mathcal{P}_0 = \phi_0$. With $\mathcal{P}_0$ being fixed, the equivalence to ψ is straightforward, as $\psi'_{\phi_0} : \mathcal{G} \to \mathcal{P}$.

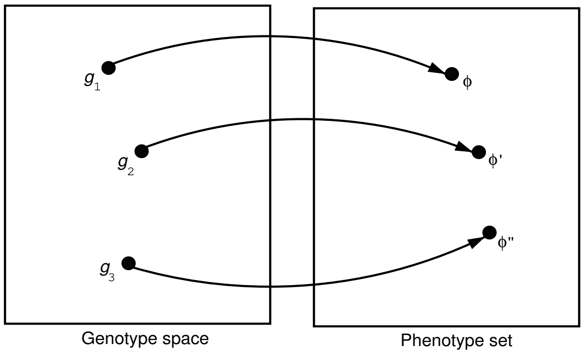

Figure 1: Functional view of genotype-phenotype map

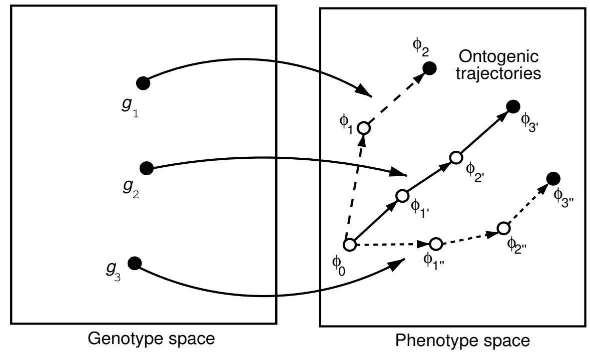

Figure 2: Operational view of genotype-phenotype map

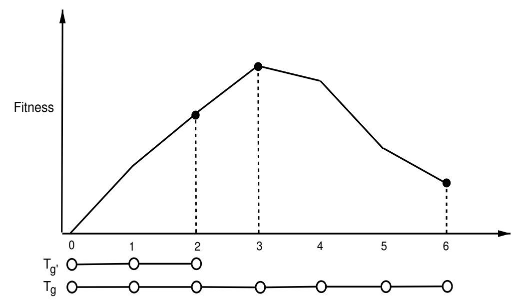

Figure 3: Fitness variation along trajectory

and maximal elements of T respectively i.e. no phenotype in P precedes ϕ_0, and ϕ precedes no phenotype in P. This definition assumes that there are no cycles in the trajectory. This is more for analytical convenience and does not effect the conclusions that we draw from the model in this paper.

With this concept of an ontogenic trajectory, we see that from a functional conception of the genotype-phenotype map as shown in Figure 1, we now have an operational conception of the form shown in Figure 2. In Figure 2, the relation between genotypes, phenotypes and ontogenic trajectories is such that for a given $T = \langle P, \prec \rangle$, a directed edge from ϕ_i to ϕ_j iff $\phi_i \prec \phi_j$ and there is no ϕ_k such that $\phi_i \prec \phi_k \prec \phi_j$ ($\phi_i, \phi_j, \phi_k \in P$). The maximal phenotypes of T are shown as solid vertices and the trajectories are distinguished by the type of edge.

As can be seen, this description abstracts away the mechanistic details of how the genotype is involved in morphogenesis and the component level structure of the phenotype but retains a focus on the temporally varying stages of development at the scale of the entire individual. A feature of our formulation is that it treats the ontogenic trajectory $T = \langle P, \prec \rangle$ as having two components - the collection of phenotypes generated P, and their relative ordering $\prec$. Since the genotype is now associated with multiple phenotypes, for such a scheme to be used with a typical EA based on the one-genotype→one-phenotype rationale (as described in Section 2.1), there needs to be a deterministic policy in place to decide which of these phenotypes is to be evaluated so as to compute the reproductive fitness of the genotype. We shall refer to this as the *ontogenic decision problem*.

The focus here is on the basic case where the relative ordering between the phenotypes is the sole basis for this decision, with the maximal phenotype of a trajectory being 'chosen' by default as the phenotype to be evaluated. We now use this formulation of ontogenic structure to deduce two immediate effects of this precedence-based policy on evolutionary search.

3. IMPLICATIONS FOR EVOLUTION

3.1 Effect on selection

Over an ontogenic trajectory there is effectively a generation of multiple phenotypes, of which only the maximal phenotype enters the population. Here we have a novel situation that is absent with a direct encoding, where there is no (non-trivial) developmental phase. With a direct encoding, every member of $\mathcal{P}$ that is generated over the evolutionary run is evaluated. However, with indirect encodings associated with development taking the form of ontogenic trajectories, every member of $\mathcal{G}$ generated over an evolutionary run is evaluated (from Equation 2) but such a guarantee clearly does not extend to all the members of $\mathcal{P}$ that are generated. This situation presents an akward problem as illustrated by the following example.

As mentioned earlier, the fitness function is defined on the phenotype set rather than on the set of all ontogenic trajectories. So, for any two phenotypes ϕ_i and ϕ_j in $\mathcal{P}$, it can be the case that $\mathbf{e}(\phi_i) > \mathbf{e}(\phi_j)$ for a fitness function $\mathbf{e}$, and $\mathbf{f}(\phi_i) < \mathbf{f}(\phi_j)$ for a different fitness function $\mathbf{f}$. Consequently, for a trajectory $T = \{\phi_0 \prec ... \prec \phi_i .... \prec \phi_j\}$, the maximal phenotype ϕ_j may or may not have the highest fitness in T depending on the particular fitness function in use. This is in contradiction to the philosophical view of development in the recapitulationist framework.

Now, consider two trajectories $T_{g'} = \{\phi_0 \prec \phi_1 \prec \phi_2\}$ and $T_g = \{\phi_0 \prec \phi_1 \prec ...\phi_6\}$ having fitness values that vary as shown in Figure 3. Here, the maximal phenotype of $T_{g'}$, namely ϕ_2, has a fitness value greater than that of ϕ_6, the maximal phenotype of T_g. However, T_g contains a phenotype ϕ_3 that has a fitness value higher than that of the maximal phenotypes ϕ_6 and ϕ_2. This, however, is a piece of information that is unavailable to the evolutionary evaluation and selection mechanisms *until* a genotype g'' having a trajectory $T_{g''}$ with ϕ_3 as its maximal element appears in the population. This could be the case *even if* ϕ_3 *was the globally optimal phenotype for the fitness function*. This dependence on the contingencies of genetic variation and the population dynamics to ensure the evaluation of the non-maximal phenotypes on the ontogenic trajectory is paradoxical given that ϕ_3 is generated as a tangible phenotype in T_g.

Since only ϕ_2 and ϕ_6 are evaluated and used as the basis for selection, g' would have a higher reproductive fitness than g, even though g is capable of generating a higher fitness phenotype than g'. As a result, depending on the structure of the population, g' would have a higher chance of being selected to produce variants in the next generation as compared to g.

From these observations, we see that by only evaluating the final phenotype the selection pressure is effectively for two properties, namely, (a) the production of high fitness phenotypes, and (b) their production at the maximal position in the ontogenic trajectory. Now, the search problem of interest was defined on the phenotype set and the desired solution that is to be ultimately returned is a phenotype rather than an ontogenic trajectory having a particular form. This implicit redefinition of the problem on the structure of the ontogenic trajectories can have a retarding effect as it limits the ability of evolution to discover and exploit the presence of high fitness phenotypes as and when they are tangibly generated over the evolutionary process, in whatever position on the ontogenic trajectory. This conflicts with the very motivation for introducing indirect encodings as a way to enable increased evolvability.

The 'blindness' of evaluation and selection leads to another related practical implication.

3.2 Uncertainty about solution quality

The absence of a guarantee that every generated phenotype would be evaluated over an evolutionary run can have an additional effect of bringing a degree of uncertainty with respect to the quality of the results obtained. This can be described as follows.

Let S_N be the set of *all* phenotypes that have been generated across N generations, independent of whether the phenotypes were maximal or intermediate phenotypes of ontogenic trajectories, and let E_N be the subset of these phenotypes that end up been evaluated. Let ϕ_E be a phenotype having the maximum fitness in E_N. If the selection mechanism used included elitism, where the highest fitness individual in a particular generation is retained unchanged in the next generation, then ϕ_E is guaranteed to be the highest fitness individual in the population at the N^{th} generation. Hence ϕ_E would also be the solution returned by the EA obtained as a result of search on $\mathcal{P}$. Now, let ϕ_S be a phenotype having the maximum fitness in S_N. It clearly can never be the case that $\mathbf{e}(\phi_S) < \mathbf{e}(\phi_E)$.

Ideally, the expectation of an 'efficient' problem-solving EA is that it returns the best possible result relative to the resources it has already consumed. So, at the minimum it is desirable that $\mathbf{e}(\phi_S) = \mathbf{e}(\phi_E)$ as each member of S_N has actually been generated over the evolutionary run by a consumption of computational resources. Notably, this is a guarantee that is trivially provided by EAs when the evolved and evaluated entities are identical as every phenotype generated over the evolutionary run is evaluated.

If $E_N \subset S_N$ then it is logically possible (as discussed above) that $\mathbf{e}(\phi_S) > \mathbf{e}(\phi_E)$, as high fitness phenotypes arising as intermediates on ontogenic trajectories, can be routinely lost as these phenotypes are 'invisible' to the evaluation and selection mechanisms of evolution. The net consequence is that even though ϕ_E may have a high fitness in absolute terms (and possibly a higher fitness than with a comparable direct encoding of the problem), the search process comes without the basic guarantee of being able to return the phenotype ϕ_S having the highest fitness of all the phenotypes that were generated during evolutionary search, as shown in Figure 4.

The uncertainty associated with this efficiency gap is absent when the globally optimal phenotype appears in E_N in which case $\mathbf{e}(\phi_S) = \mathbf{e}(\phi_E)$. However, this is hardly a property that can be guaranteed with non-trivial search problems where the fitness value of the globally optimal phenotype may be unknown.

So, the 'efficiency gap' can be seen as the opportunity cost of not evaluating every generated phenotype during the developmental process.

3.3 An informed G-P map, χ

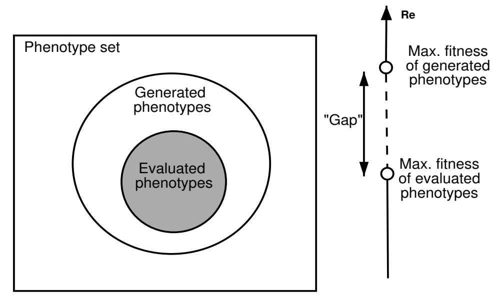

Figure 4: Efficiency gap

Before proceeding to the empirical demonstration of these effects, we will consider an alternative basis for the ontogenic decision problem to provide a contrast. The goal is not to present this as an alternative technique or a resolution to the above problems but merely as an experimental control.

As may already be evident to the reader, the above issues to do with selection and 'efficiency' become irrelevant when the relation between the fitness function and the ontogenic structure is such that $\mathbf{e}(\phi_{final}) \geq \mathbf{e}(\phi)$ ($\phi \in P$) is true for $T_g = \langle P, \prec \rangle$ for every $g \in \mathcal{G}$. To ensure that the G-P map behaves *as if* the fitness function satisfied this property, we can use an 'informed' version of ψ.

Rather than using an 'uninformed' policy based only on the natural structure of the ontogenic process as defined by $\prec$, an alternative is to completely ignore the ordering in the ontogenic trajectory and treat development as if it generated a flat subset of phenotypes P. A phenotype is then picked from P by a process of local search using the fitness function to return ϕ_{max} ($\phi_{max} \in P$), where $\mathbf{e}(\phi_{max}) \geq \mathbf{e}(\phi)$ ($\phi \in P$), as the phenotype 'encoded' by the genotype. This approach is superficially similar to the effect of life-time learning with phenotypic plasticity as in [7]. As the highest fitness is unknown *a priori*, local search to find ϕ_{max} here is effectively an exhaustive evaluation of all the phenotypes in P.

This informed G-P map is labelled as $\chi : \mathcal{G} \to \mathcal{P}$. Even though $\mathcal{G}$ and the ontogenic trajectories produced are identical with both ψ and χ, each genotype could now be associated with a different phenotype in χ. Hence $\mathcal{G}$ is behaviorally a different encoding of $\mathcal{P}$ when used with χ as compared to its usage with ψ, with χ being guaranteed to be efficient (in the sense described above) while ψ is not so. This comes with two important subtleties.

The use of the fitness function $\mathbf{e}$ here is to pick a phenotype to enter the population from among the *genetically identical* phenotypes in the same trajectory. So, these fitness evaluations do not provide information about the variational structure of the genotype space as with the genetically distinct phenotypes in the population. As a result, the fitness of ϕ_{max} alone determines the reproductive fitness of the genotype in the context of the population.

Another key requirement to enable such a local search is that the entire ontogenic trajectory be explicitly cached. During development a particular phenotypic state is transformed by $\mu_{g,t}$ to produce the next phenotypic state. Hence the ontogenic intermediates are irreversibly lost when the trajectory is non-cyclic rather than being being present in memory when the final phenotype is obtained. Given this irreversibility, it would be impossible to ascertain whether the present phenotypic state at time t is ϕ_{max} during run-time without the generation of the subsequent phenotypes. On generating the next phenotype, it would be impossible to return to the previous phenotype if that phenotype was indeed ϕ_{max}. So the mere introduction of run-time fitness evaluation does not imply that it would lead to an identification of ϕ_{max}, without maintaining

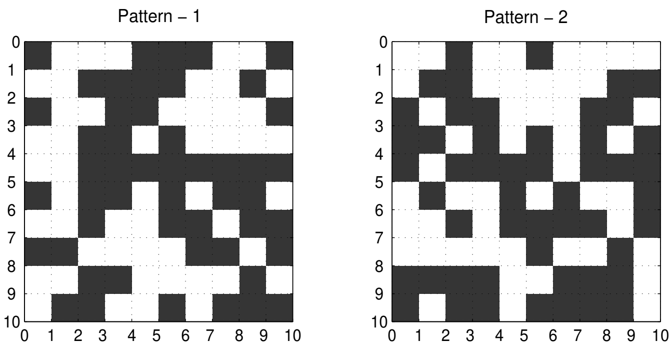

Figure 5: Target patterns (randomly generated)

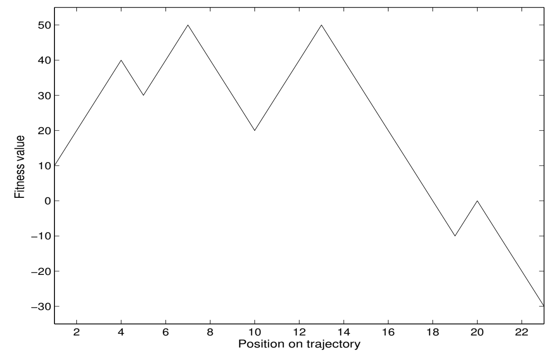

Figure 6: Fitness variation along ontogenic trajectory

additional state information.

In the next section we explore the characteristics of evolution with these two differing G-P maps ψ and χ using a strategy similar to the comparative studies described in [9, 8, 10]. The strategy in these studies was to keep the fitness function, selection and population size fixed over the evolutionary run for the encodings being compared. So the observed differences in the fitness increase and absolute fitness over a fixed number of fitness evaluations with these differing encodings were attributed to differences in the evolvability of the genetic encoding (and their associated variation operators). In this case the genetic encodings and variation operators are identical for both ψ and χ, the only difference is in the basis used for the ontogenic decision making.

4. EXPERIMENTS

4.1 Problem

The toy-problem chosen here is one of pattern construction, similar to that used in [10]. A target pattern consists of a bit-pattern on a 10×10 grid (Figure 5). The general objective is to evolve a pattern that corresponds to this target pattern. Here, the phenotype space $\mathcal{P}$ of interest is the set of *connected patterns* on this 10×10 grid, even though the target patterns are not connected. This is irrelevant as the goal is to study the evolutionary dynamics rather than the properties of the solutions obtained.

The fitness function $\mathbf{e}$ is defined as follows: Given a phenotype $\phi \in \mathcal{P}$, and the target pattern A, a value x is assigned to each cell of ϕ that matches the target pattern, and a value y is assigned to each cell that does not match this target. The overall fitness is the sum of the x and y values of each cell of ϕ. Here $x = +10$ and $y = -10$.

4.2 Ontogenic system: Turtle construction

The developmental system used is based on the the well known mathematical animal - the LOGO turtle. The turtle's movements are controlled by the execution of a procedure based on heading and orientation commands. The trace of the turtle's movements in space forms the basis for the construction of geometric objects (for details see [1]).

To focus on the canonical properties of evolution with a developmental phase, here we consider a simplified version of this turtle. The turtle moves on the 10×10 grid defined by the problem. The basic commands accepted by the turtle are: `forward, back, left, right`. The commands `forward` and `back` change the turtle's *position*, while the commands `left` and `right` change both the turtle's *heading and position*.

The turtle always moves in unit steps. The default input for the `right` command is $90°$ with respect to the current heading, and $-90°$ for the command `left` with respect to the current heading, with a single unit step taken in the new orientation for both commands. So the turtle always moves with the direction of its heading being parallel to one of the (globally defined) principal axes.

All the cells on the grid are assumed to have an initial state 0, and each time the turtle vists such a cell, the state changes to 1. Once the state changes from 0 to 1, it remains unchanged. As a result, the turtle's movements on executing a series of commands result in a connected pattern defined by the cells having the state 1. The start position for the turtle is always at $(5, 5)$ on the grid pointing upward, i.e. in the $Y-$ direction.

The procedure executed by the turtle takes the form of a list of these commands that are executed in sequential order. The set of all such procedures (with a minimum length = 5 and maximum length = 200) is taken to be the set of genotypes $\mathcal{G}$. Every connected trace produced by the turtle is a valid phenotype $\phi \in \mathcal{P}$. The ordered sequence of phenotypes produced during the execution of a particular genotype g starting from the first command is be considered to be an ontogenic trajectory T_g. The phenotype obtained when all the commands in g have been executed is therefore the maximal phenotype of T_g.

The fitness function is ontogenically non-monotonic for this system. Figure 6 shows the fitness variation along the ontogenic trajectory for a randomly generated sequence `LRBFLLR LFRLFFLFFR RFFLBFLB FBFBLLLL BFBFFLRFRLFL` (where `L = left, R = right, B = back, F = forward`). If the trajectory never leaves the starting state ϕ_0, the corresponding genotype is assigned a fitness of -500.

4.3 Variation and selection operators

The variational operators include both mutation and crossover. Three mutational operators were used. Mutational operator M_1 randomly replaces a randomly selected command (with uniform probability) on the given procedure by one of the other three commands. The operators M_2 and M_3 were specifically designed noting that the procedures are executed sequentially. Mutational operator M_2 reduces the length of the given procedure by removing a segment of randomly chosen length (with a maximum of 5 commands) from the end of the procedure upto the minimum permissible procedure length. M_3 adds a list of randomly generated commands (with a maximum of 5 commands) to the given procedure upto the maximum permissible procedure length. M_1 was applied with a probability 0.5, and M_2 and M_3 with probability 0.25 each. Crossover is at a single common locus that is randomly chosen on the shorter procedure with uniform probability over its

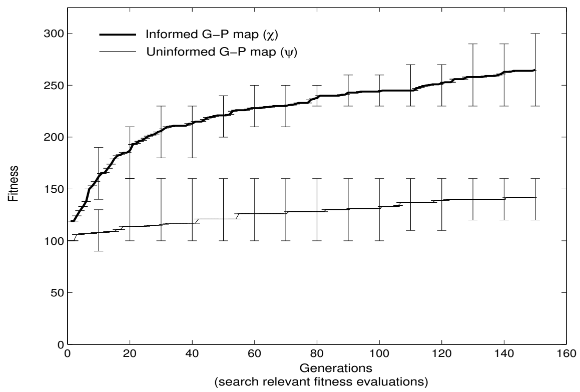

Figure 7: Fitness of best evaluated individuals (Pattern-1)

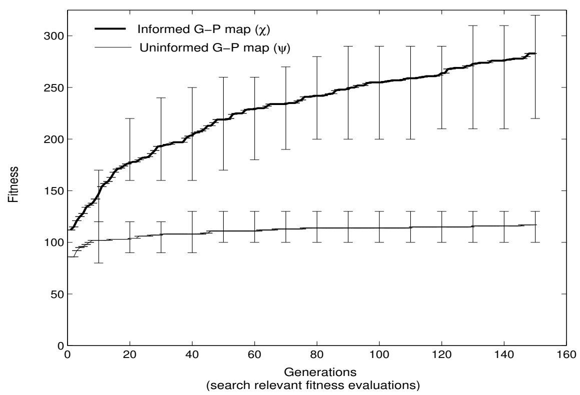

Figure 8: Fitness of best evaluated individuals (Pattern-2)

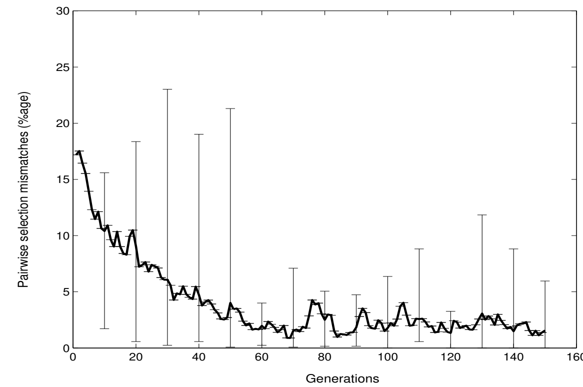

Figure 9: Selection mismatches with ψ (Pattern-1)

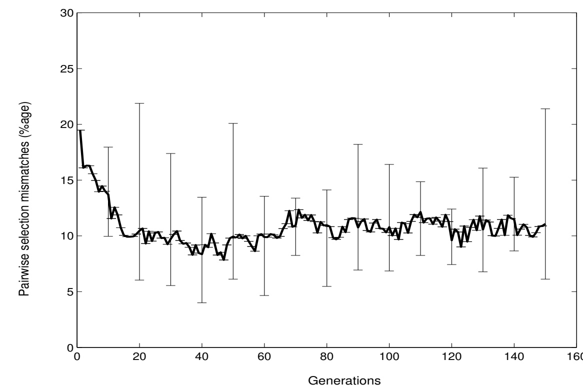

Figure 10: Non-selective mismatches with χ (Pattern-1)

entire length.

Selection is fitness proportionate. The initial population consists of randomly generated procedures of lengths ranging from 5 to 60. This population is evolved with a fixed-population size, generational EA with an elitism of 3. 20% of the remaining slots (rounded to the nearest even number) of every successive generation are reserved for genotypes obtained by crossover, while the remaining slots are filled by mutational variants.

4.4 Results

The results of evolution with the encodings ψ and χ with a population of 50 individuals over 150 generations (over 10 runs) for the randomly generated patterns in Figure 5 are described below.

Figure 7 is a comparison of the change in fitness values of the best evaluated phenotypes with the encodings ψ (uninformed) and χ (informed) for pattern-1, with respect to *search relevant fitness evaluations* (as discussed earlier in Section 3.3). As can be seen the rate of fitness increase as well as the fitness of the best individual after 150 generations is considerably greater than that obtained using ψ. This is also the case with Pattern-2 as shown in Figure 8. Due to the similarity of the results obtained with both patterns as well as several other randomly generated patterns, we will restrict our discussion to the results obtained with pattern-1 from this point on.

In Section 3.1, we argued that the presence of selection mismatches was a key property that could cause a retarding effect on evolution with a precedence-based ontogenic decision policy. Indeed, this is the only difference between ψ and χ in this turtle-implementation as all other aspects of evolution (i.e. genetic encoding, variation, fitness function, population size) are common to both. We see that such selection mismatches do indeed occur here.

A mismatch is deemed to occur when the value of $\mathbf{e}(\phi_{final})$ for genotype g_1 is greater than for genotype g_2, but where the value of $\mathbf{e}(\phi_{max})$ for genotype g_2 is greater than for genotype g_1. This is expressed as a percentage of the total number of pair-wise comparisons of genotypes in the population. With a population of 50, the total number pair-wise comparisons is equal to 1225. Figure 9 shows the frequency (averaged over 10 runs) with which these selection mismatches occur in the population with ψ. Here the number of such mismatches is high 17% (approx. 220 mismatched pairs) in the initial random population which then shows a decreasing trend, remaining at the order of 3% (approx. 60 mismatches) without entirely stabilizing to zero over the 150 generations.

To provide a contrast, consider the mismatches that occur between ϕ_{max} and ϕ_{final} with χ as shown in Figure 10. With χ, these mismatches do *not* have any consequences for selection as ϕ_{max} is explicitly identified. Even so, it is noticeable that the number of such mismatches consistently remains at a high level (at about 10% or 122 mismatches) with large fluctuations.

This suggests that the search strategy in each case is very different specific to the characteristics of the genetic representation and variation operators used here. This is also noticeable in the significant difference in the rate at which the average length of the ontogenic trajectories [5] changes over evolutionary time as shown in Figure 11. The difference in the rate of growth of the phenotypes suggests that a reason for the low number of selection mismatches

[5]The length of an ontogenic trajectory $T = \langle P, \prec \rangle$ is $l_T = |P|$

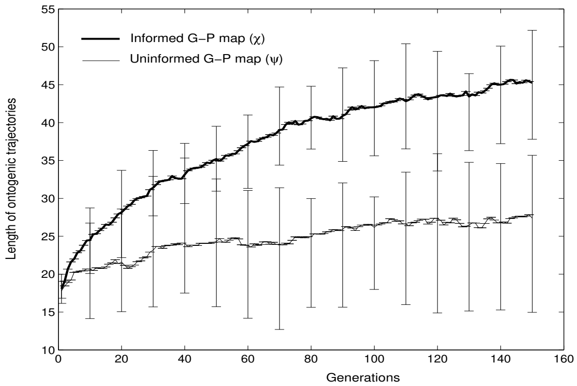

Figure 11: Average length of ontogenic trajectories (Pattern-1)

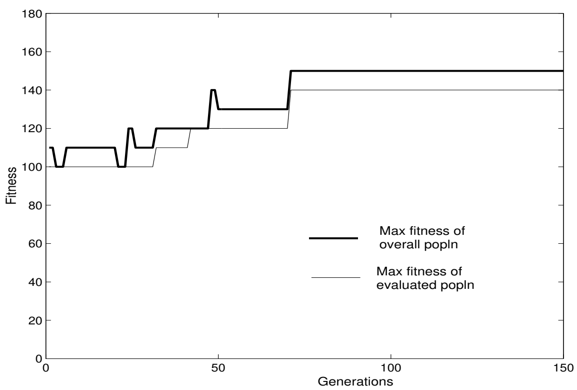

Figure 12: Example of efficiency gap (Pattern-1)

with ψ may be reflective of a population that has low diversity, which reduces the possibility of a mismatch.

Finally, this brings us to the uncertainty in the quality of the results or what we termed the 'efficiency gap'. An example of this gap arising between the maximum fitness of all the generated phenotypes in a particular generation and the maximum fitness of the evaluated phenotypes in that generation from a single evolutionary run with ψ is shown in Figure 12. Depending on the stopping condition, the solution returned may have a fitness less than that of phenotypes that have been generated but have remained unevaluated.

It with this plot that the retarding effect that artificial ontogenies of the form ψ have on evolution can be explicitly seen. There is a distinct time lag between when a high fitness phenotype is first generated, and when variation can produce an evaluated phenotype with a comparable fitness value. This is fundamentally different from the phenomenon of canalization where a high fitness phenotype is discovered by interaction with the environment, and there is a time lag before this phenotype can be produced directly independent of the environment. Unlike canalization, in this case the presence of these high fitness phenotypes appearing as ontogenic intermediates is invisible to selection and does not exert any selection pressure for canalization to occur. Rather than driving adaptation, here the ontogenic structure retards the rate of adaptation in preventing these high fitness phenotype from being discovered and exploited by selection and variation mechanisms. As can be seen in Figure 7, the removal of these constraints in χ make a significant difference on the rate of evolution.

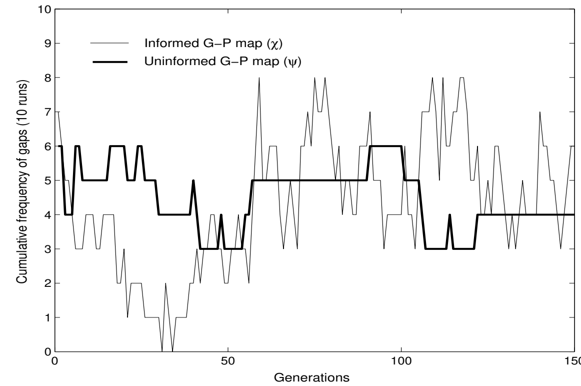

Figure 13: Frequency of efficiency gap occurence (Pattern-1)

Figure 13 shows the frequency with which these gaps occured across the 10 evolutionary runs with ψ, and for contrast, with χ. Here we see that the efficiency gaps appear and disappear sporadically and are neither isolated occurences on a particular evolutionary run or necessarily a constant presence.

5. DISCUSSION AND CONCLUSIONS

One of the main motivations for using indirect genetic encodings in evolutionary problem solving has been the possibility of improving the *evolvability* of the genetic representations by design [3, 12]. From this evolvability perspective, the genetic representation is a 'rate-limiting factor' [13] in that different representations of a given problem are seen as effecting differences in the rate at which high fitness variants are produced under variation and selection.

To this end, development has largely been treated as a vehicle for an indirect genetic encoding of the search problem. However, in this paper, we have demonstrated that evolution with an explicit developmental phase can bring some novel issues to bear on evolutionary search different from the genetic issues. By explicitly considering the structure latent in developmental processes, we have shown that this additional structure can have the net effect of *retarding* rather than promoting evolution. Where phenotypic plasticity aided by life-time learning by interaction with the environment leads to an expediting effect on evolution, here the plastic phenotypic transformations occuring under the control of the genotype has the contrary effect of retarding evolution. Analogous to the *Baldwin expediting effect* [7], we can refer to this contrary phenomenon as the *Haeckel retarding effect* based on the classic recapitulationist conflation of the awe-inspring process of morphogenesis with evolutionary progress.

Looking forward, the key conceptual issue that this phenomenon raises is that the genotype needs to be viewed as more than just a recipe for how a phenotype is to be constructed but also as a *strategy for the evaluation of the products of development*. While the 'informed' strategy of evaluating every phenotype generated was presented as a basic resolution of the problem posed by evaluating only the final phenotype, it is far from being a satisfactory natural resolution. As described earlier, by completely ignoring the inherent structure in the ontogeny, it takes a hammer to the problem by converting it into one of local search. However, this involves the caveats of having to cache the entire trajectory rather than providing a solution that is integrated into the generative character of the development process. Finding a more sound solution is an open question that needs to be resolved. To this end, we have begun to explore the possibility of a game-theortic conception of development [11].

There are also several empirical issues associated with addressing this problem. The analysis and the demonstration presented here is clearly simplistic and, among other things, does not (a) adequately address the properties of several existing developmental approache, and (b) says little about the prevalence or the importance of this effect in 'real-world' problems. Furthermore, the critical issue of whether the cost of resolving this issue is commensurate with the gains obtained has remained largely unaddressed here.

Evolution with indirect encodings have already produced numerous successes even without the recognition of this underlying issue. The recognition of this retarding effect therefore suggests an opportunity, rather than a shortcoming, to tap the structure provided by the development processes to further enhance the evolutionary capabilities with such encodings.

6. ACKNOWLEDGEMENTS

The authors would like to thank Anthony Bucci, Sevan Ficici, Richard Watson, Keki Burjorjee and the three anonymous reviewers for their constructive criticism and comments.

7. REFERENCES

[1] H. Abelson and A. diSessa. *A. Turtle Geometry: The Computer as a Medium.for Exploring Mathematics*. MIT Press, Cambridge, MA, 1981.

[2] P. Alberch, S. Gould, G. F. Oster, and D. B. Wake. Size and shape in ontogeny and phylogeny. *Paleobiology*, 5(3):296 – 317, 1979.

[3] P. Angeline. Morphogenic evolutionary computations: Introduction, issues and example. In J. R. McDonnell, R. G. Reynolds, and D. B. Fogel, editors, *Evolutionary Programming IV: Proceedings of the Fourth Annual Conference on Evolutionary Programming*, pages 387–401. MIT Press, March 1995.

[4] J. C. Bongard and R. Pfeifer. Evolving complete agents using artificial ontogeny. In *Morpho-functional Machines: The New Species (Designing Embodied Intelligence)*. Springer-Verlag, 2003.

[5] S. J. Gould. *Ontogeny and phylogeny*. Belknap press, 1985.

[6] F. Gruau. *Neural Network Synthesis using Cellular Encoding and the Genetic Algorithm.* PhD thesis, Laboratoire de l'Informatique du Parallilisme, Ecole Normale Supirieure de Lyon, France, 1994.

[7] G. E. Hinton and S. J. Nowlan. How learning can guide evolution. *Complex Systems*, 1:495–502, 1987. reprint in: Adaptive Individuals in Evolving Populations: Models and Algorithms, R. K. Belew and M. Mitchell (eds.), 1996, pp. 447–454, Reading, MA: Addison Wesley.

[8] G. Hornby and J. Pollack. Creating high-level components with a generative representation for body-brain evolution. *Artificial Life*, 2002.

[9] M. Komosinski and A. Rotaru-Varga. Comparison of different genotype encodings for simulated three-dimensional agents. *Artificial Life*, 7(4):395–418, 2001.

[10] S. Kumar and P. J. Bentley. Computational embryology: past, present and future. *Advances in evolutionary computing: theory and applications*, pages 461–477, 2003.

[11] S. Viswanathan and J.B.Pollack. On the robustness achievable with stochastic development. In *Proceedings of the 2005 NASA/DoD Conference on Evolvable Hardware*, 2005. (to appear).

[12] G. Wagner and L. Altenberg. Complex adaptations and the evolution of evolvability. *Evolution*, 50:967–976, 1996.

[13] A. S. Wilkins. *The Evolution of Developmental Pathways*. Sinauer Associates, 2001.