Acclivation of Virtual Fitness Landscapes

Ben Kovitz[1], David Bender[1], and Marcela Poffald

[1]Fluid Analogies Research Group, Indiana University, Bloomington, IN 47408
bkovitz@indiana.edu

Abstract

Any part of a genome, considered separately from the rest of the genome, evolves against a "virtual fitness landscape" that results when the rest of the genome is held constant. We show how analyzing a genome in this way can explain one form of progressively increasing evolvability.

When one part of a genome is a vector of numbers ("knobs") and the rest is a graph that determines the mapping from knobs to phenotype, the graph will respond to selective pressure to "acclivate" the virtual fitness function faced by the knobs—that is, to make it more hill-shaped. For as long as the knobs' virtual fitness function provides opportunity for distorting it to make knob-turning mutations improve fitness, the graph experiences pressure to evolve those distortions as a side-effect of responding to its own virtual fitness function.

As the knobs' virtual fitness function grows more hill-shaped, the knobs track upward paths more easily and hence so does the genotype as a whole. A synergy develops between incremental exploration of phenotypes by knob-mutations and discontinuous exploration by graph-mutations. A favorable condition for this is a global fitness function that frequently varies, changing constants but leaving structural invariants unchanged. The graph then accumulates a memory of the invariants as revealed across many previous epochs, held in the form of bias limiting and directing future evolution.

Introduction

In previous work (Kovitz, 2015), we found that cascading designs—organisms consisting of graphs that direct cascades of interactions among many parts—are well suited to evolve increasing evolvability, because a single mutation is likely to produce a coordinated change throughout the phenotype, preserving relationships among the parts of the phenotype that might be essential for survival while altering constants that incrementally improve fitness. The classic example of a cascading design is a metabolic network: a variety of enzymes, each catalyzing reactions that create, consume, speed, or slow other enzymes. Others include neural networks, genetic regulatory networks, and even software systems where cascades of activity are propagated by function-calls or message-passing.

In the present paper, we investigate a synergy between the "knobs" of a cascading design—elements subject to incremental, quantitative mutation—and the "graph" or "topology" of a cascading design—the structure of interactions that is subject to discontinuous, sometimes radical mutations. We find that under certain conditions, the graph faces selective pressure to map a rugged fitness landscape to a more hill-shaped virtual landscape for the knobs. The map often excludes the worst regions of the landscape from its range. The result is a mechanism by which evolvability can evolve (Colegrave and Collins, 2008).

Virtual Fitness Functions

Any part of a genome is selected against a *virtual fitness function* resulting from the interaction between the rest of the genome and the fitness function faced by the genome as a whole. If the whole-genome fitness function reflects the influence of the environment on the genome, then the virtual fitness function represents the same for a part of the genome, whose environment includes the rest of the genome.

Let a set of genotypes G have a mapping $m_G : G \to \Phi$ to a set of phenotypes Φ, and let $w_\Phi : \Phi \to \mathbb{R}$ be the fitness function for the phenotypes. Then $w_G : G \to \mathbb{R}$, the fitness function for the whole genome, is the composition of these functions, $w_G(g) = w_\Phi(m_G(g))$.

If we divide the genome into two parts G_1 and G_2, then each genotype $g \in G$ consists of a $g_1 \in G_1$ and a $g_2 \in G_2$, in which each g_2 defines a partial-genotype–phenotype mapping $m_{g_2} : G_1 \to \Phi$. That is, if we hold part of the genome constant, say by fixing g_2, this defines a mapping from all possible values of the rest of the genome, g_1, to corresponding phenotypes. If we reverse g_1 and g_2, then of course we get the opposite partial-genotype–phenotype mapping, $m_{g_1} : G_2 \to \Phi$.

These mappings, in turn, define virtual fitness functions $v_{g_2}(g_1) = w_\Phi(m_{g_2}(g_1))$ and $v_{g_1}(g_2) = w_\Phi(m_{g_1}(g_2))$. As mutations and crossovers can alter either or both of g_1 and g_2, the partial genomes G_1 and G_2 coevolve cooperatively, each selected by the fitness functions v_{g_1} and v_{g_2}, which vary among all the individuals and vary each generation.

Let *evolvability* be defined in some reasonable way (there are many), so that greater evolvability implies some advantage in navigating a fitness landscape upward faster or further over succeeding generations. Let $g_a, g_b \in G$ be two individuals in the same population and the same generation, g_a having parts g_{a1} and g_{a2}, and g_b having parts g_{b1} and g_{b2}. Assuming no other advantages favoring either g_{a1} or g_{b1}, if g_{a1} presents its mate g_{a2} with a virtual fitness function $v_{g_{a1}}$ that g_{a2} finds more evolvable than $v_{g_{b1}}$ is for g_{b2}, then the descendants of g_a will evolve faster or further than the descendants of g_b (according to how evolvability is defined).

Therefore each partial genome responds to any available selective pressure to create a virtual fitness landscape for the other partial genome that gives the latter greater evolvability. To illustrate with an unrealistically simple example, if the eye and the arm are governed by separate sets of genes, and some arm shapes make it easier for the eye to evolve—say, by providing cues that the eye can track to for hand-eye coordination—then there is selective pressure favoring alleles for those arm shapes. All other factors being equal, evolution favors arms that make eyes easier to evolve. This selective pressure happens indirectly; in any single generation, greater fitness wins. But over successive generations, descendants of organisms with greater evolvability will tend to have greater fitness than organisms with lesser evolvability.

The above considerations make no difference for homogeneous genomes, where every part of each genotype undergoes mutation and crossover the same as every other part and exerts the same effect on the phenotype or on the total fitness as every other part. However, if G_1 and G_2 vary according to different operators and/or affect the phenotype or total fitness differently, there is potential for each part to seek values that make the other part more evolvable, resulting in a period of progressively increasing evolvability for the organism as a whole.

In the rest of this paper, we examine a simple and natural way for this synergy to occur: when g_1 consists of a vector of real numbers ("knobs") and g_2 consists of a network that provides connections through which the numbers from g_1 interact.

Acclivation

As is well known, a genome consisting of a vector of numbers, where mutations alter the numbers by small amounts, evolves most easily against a hill-shaped fitness function. In a hill-shaped fitness function, local increases in fitness correlate with movement toward the peak of the whole fitness landscape (Kauffman and Levin, 1987). The more "rugged" the landscape, the weaker is this correlation, so that following the local gradient can lead organisms to become stuck at local maxima from which they cannot escape by local mutations (though they might escape by crossover).

Therefore, if a vector of numbers faces a rugged fitness landscape, with difficult features such as low local peaks and impassable moats, we can improve its evolvability for a vector of numbers by making its fitness function more hill-shaped. Let us call the process of making a fitness landscape more hill-shaped *acclivation.*[1]

So, in a genome where G_1 is a vector of numbers that mutate by small amounts, and G_2 is a directed graph that feeds the numbers in G_1 through nodes that perform some function on the numbers from their input edges, eventually leading to a phenotype whose fitness determines the fate of the whole organism, we should expect selective pressure for genotypes $g_2 \in G_2$ to produce mappings that induce acclivation on the virtual fitness functions v_{g_2}. Evolution should favor graphs that put knobs in a position where they can hill-climb successfully.

Genome for Experimentation

To test the preceding hypothesis in a form in which acclivation will be visually apparent on plots printed on paper, we limit ourselves to genomes where g_1 and the phenotype are 2-dimensional vectors and g_2 is a graph connecting them. The whole genome is a directed graph where:

1. Two nodes, called the *knobs*, k_1 and k_2, are designated to each hold a number in $[-1.0, +1.0]$, called an *initial activation*.
2. Two other nodes, p_1 and p_2, are designated to hold the phenotype.
3. Zero or more additional nodes $n_1, n_2, \ldots$.
4. Each edge has a weight of either $+1.0$ or -1.0.

Genotype–Phenotype Mapping

The phenotype is determined by a process of spreading activation, run for 10 timesteps. At each timestep, each node can have either an *activation* in $[-1.0, +1.0]$ or no activation. At timestep 0, only the knobs have activations: the numbers stored in the genotype. Each successive timestep, activations spreads from from nodes (the ones with activations) to their neighbors. If none of a node's incoming neighbors has an activation, its own activation (or lack of one) is unchanged. Otherwise, the activation of a node a_j at timestep $t+1$ is calculated according to the following function:

$$a_j(t+1) = T(a_j(t) + \sum_i W_{ij} a_i(t))$$

where W_{ij} is the weight of the incoming edge, if any, from node i to node j, and T is the following transfer function:

$$T(x) = \frac{2}{1 + \exp(-Sx)} - 1$$

[1] From Latin *clivus*, meaning a slope or a hill, combined with the prefix *ad-* indicating in this context an upward slope or becoming more sloped.

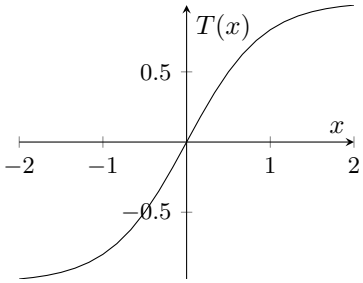


Figure 1: The transfer function T is a sigmoid function with attractors at ± 0.5 and a repellor at 0.

where $S = 2.1972274554893376$. This constant gives T, when iterated, attractors at ± 0.5 and a repellor at 0.

If a node does not have an activation at time t, then it does not figure into the above sum for calculating any other node's activation. At time $t = 0$, only the knobs have activations.

The phenotype is the vector $(a_{p_1}(10), a_{p_2}(10))$, i.e. the activations of the phenotype nodes after 10 timesteps. If p_1 or p_2 has not received an activation after 10 timesteps, then the genotype has no phenotype and is given a fitness of 0.0. This can happen if no edges provide a path from a knob to p_1 or p_2.

Example The following table shows step-by-step how the spreading-activation algorithm calculates the phenotype for the simple genotype in Figure 2.

t	k_1	k_2	n_1	p_1	p_2
0	-0.659	1.000			
1	-0.659	0.358	-0.619	-0.619	0.800
2	-0.044	-0.319	-0.970	-0.886	0.854
3	0.769	-0.379	-0.975	-0.771	0.529
4	0.958	0.404	-0.861	-0.002	0.164
5	0.964	0.905	-0.686	0.782	0.554
6	0.948	0.968	-0.420	0.958	0.922
7	0.906	0.971	-0.118	0.970	0.970
8	0.699	0.968	0.850	0.968	0.972
9	-0.164	0.950	0.990	0.950	0.971
10	-0.853	0.698	0.964	0.700	0.971

At t=0, the genotype provides the initial activations of the knob nodes.

At t=1, n_1 and p_1 each receive an input of -0.659 from k_1; each gets an activation of $T(-0.659) = -0.619$. Similarly, p_2 receives an input of 1.000 from k_2, giving p_2 an activation of $T(1.000) = 0.800$. Since the only input to k_1 comes from n_1, and n_1 had no activation on timestep 0, k_1's activation is unchanged.

At t=2, n_1 receives inputs along two edges: -0.659 from k_1 and -0.619 from itself. These add to n_1's preceding activation, so n_1's new activation becomes $T(-0.619-0.619-0.659) = -0.970$. k_1 now receives the -0.619 from n_1 but the edge has weight -1, so k_1's activation becomes $T(-0.659+0.619) = 0.044$. k_2 receives an input of -0.659 from k_1, so k_2's activation becomes $T(0.358 - 0.659) = 0.319$. p_1's activation becomes $T(-0.619 - 0.659) = -0.886$ and p_2's becomes $T(0.800 + 0.358) = 0.854$.

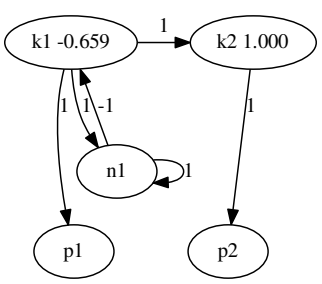


Figure 2: A simple genotype. The knob nodes are at the top, the phenotype nodes are at the bottom, and there is one additional node.

Now that all the nodes have activations, the cycle continues: k_1 and n_1 interact and p_1 and p_2 essentially accumulate output from k_1 and k_2, scaled back each timestep by the T function.

Finally, after 10 timesteps, p_1's activation is 0.700 and p_2's activation is 0.971, so this genotype's phenotype is $(0.700, 0.971)$.

Variation Operator

In generation 0 of the first epoch in each experiment, the population consists of genotypes containing only the two knob nodes and the two phenotype nodes, with up to four randomly placed edges with weights randomly chosen from $\{-1, +1\}$, and the knobs' initial activations chosen uniformly from $[-1.0, 1.0]$.

Each organism of each successive generation is generated by selecting one or two parents from the previous generation by tournament selection and making a child by a single mutation or by crossover. Crossover has a low probability, usually 0.02 or 0.05.

The possible mutations are: add a node, remove a node (but not a knob or phenotype node), add an edge, remove an edge, move an edge, or turn a knob. Knob-turning has a probability roughly equal to the sum of all the graph-edit mutations. Depending on the experiment, turning a knob chooses a knob delta from $\{-0.02, +0.02\}$ or from a normal distribution with mean 0 and $\sigma = 0.0$.

Population sizes range from 60 to 800 depending on the experiment. We omit some details of the variation operator here for lack of importance. The source code is publicly available at https://github.com/bkovitz/acclivation.

Experiments

In each experiment, we run the genome defined above against a different family of fitness functions and see what virtual fitness functions emerge. We only plot v_{g_2}, the virtual fitness function seen by the knobs, since we know of no way to plot fitness functions seen by a graph.

We found no reliably meaningful measure of acclivation. We tried running a hill-climbing algorithm on the virtual fitness functions but this yielded ambiguous results. For example, a higher fitness reached by the hill-climbing algorithm in many cases resulted not from acclivation but from extreme canalization: the graph forced the phenotype to a predetermined point regardless of the knobs. So, we plot no temporal dynamics of populations. Instead, we show only some representative individuals to illustrate the kinds of virtual fitness functions found.

Each experiment tries a "family" of fitness functions, because each experiment's fitness function has a constant that changes randomly once per epoch: every 20 generations. This constant moves the peak of the fitness function to different places in phenotype space.

Experiment 1: Razorback

In this experiment, we ran the experimental genome against this fitness function, plotted in Figure 3(a):

$$w_\Phi(\phi) = 10.0 \cdot \hat{d}(\phi, P) \cdot \wedge(|p_2 - p_1|; R) + \text{waves}(\phi; 30)$$

where:

P is a point (the peak) chosen randomly along the $y = x$ line each epoch;

$\hat{d}(\phi, P)$ is a measure of the proximity of ϕ to P equal to 0.0 for the maximum possible distance and 1.0 for zero distance:

$$\hat{d}(\phi, P) = \frac{\max - d(\phi, P)}{\max}$$

$\wedge$ is the "inverted-v" function: like an inverted-U function but peaking sharply at $x = 0$ and returning zero outside the radius R, set to 0.1 or 0.2 on different runs of the experiment:

$$\wedge(x; R) = \begin{cases} 0 & \text{if } |x| > R \\ 1 - (\frac{x}{R})^2 & \text{if } |x| \leq R \end{cases}$$

and "waves" is a function that adds regular undulations, giving the overall fitness function an "egg carton" look, shown in Figure 3(a):

$$\text{waves}(\phi; \nu) = \cos(\nu p_1) \cdot \sin(\nu(p_2 + \frac{\nu}{2}))$$

So, this function rewards the phenotype up to 10 points for proximity to P, but only if the phenotype lies along a narrow ridge running diagonally across phenotype space, complicated by the addition of a regular pattern of undulations. The undulations add local minima throughout the fitness landscape to trap searches that merely follow the local gradient.

Figure 3 shows an organism that evolved in this experiment. The virtual fitness function illustrates acclivation: there is a steep slope leading to a "butte" containing the global fitness peak, and the narrow ridge of the phenotype function is widened and distorted, making it climbable from different directions.

This organism also illustrates another fundamental way, aside from acclivation, of gaining evolvability: by restricting the range of m_{g_2} to exclude bad parts of the phenotype space. The genotype–phenotype mapping does not allow access to any points in phenotype space other than those along the center of the ridge.

Experiment 2: Circle

In this experiment, we ran the experimental genome against this fitness function:

$$w_\Phi(\phi) = 10.0 \cdot \hat{d}(\phi, P) \cdot \wedge((p_1^2 + p_2^2) - r^2; R)$$

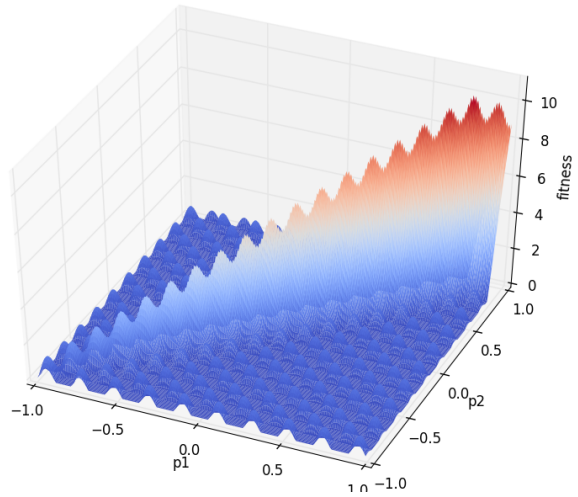
where r is the radius of a circle, R is the ridge radius as in the first experiment, and P is a point (the peak) chosen randomly along the circle at the start of each epoch. In words, the phenotype is rewarded up to 10.0 points for proximity to the peak, but only if the phenotype lies within R of the perimeter of the circle—a circular ridge. We set $r = 0.5$ and $R = 0.15$.

Figure 4 shows one organism that evolved in this experiment. It has evolved canalization for phenotypes near the circular ridge and decanalization for phenotypes in the center of the circle (knob-turnings quickly move the phenotype away from the center). All phenotypes outside the circle are inaccessible in this organism's genotype–phenotype mapping.

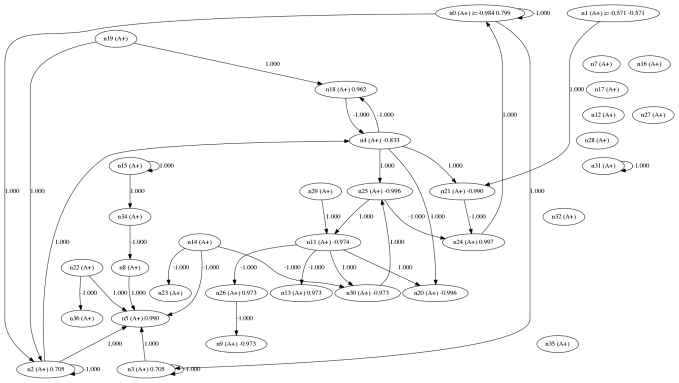
Experiment 3: Moats

In this experiment, we run a modified version of the razorback fitness function: wherever $\text{waves}(\phi) \leq 0.5$, fitness is zero rather than slightly reduced; organisms with fitness zero are not allowed to reproduce; the "islands" where $\text{waves}(\phi) > 0.5$ have flat, neutral plateaus, so there is no smooth gradient to climb within any one island; and the islands are spaced further apart than in the razorback experiment. So, organisms can only cross from one island to a higher island by a single mutation. Ending the lineage of an organism that falls into the "moat" between islands simulates the tendency in nature for fitness landscapes be to "holey" (Gavrilets, 1997), requiring leaps over regions of nonviable genotypes in order to improve fitness.

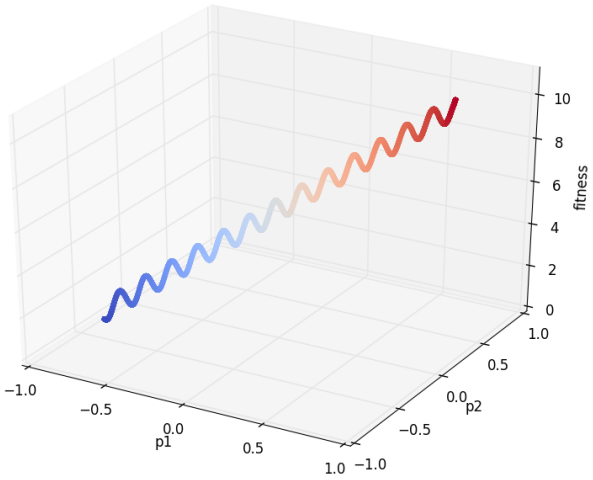
Figure 5 shows an organism that successfully climbs the chain of islands. It has evolved a genotype–phenotype map-



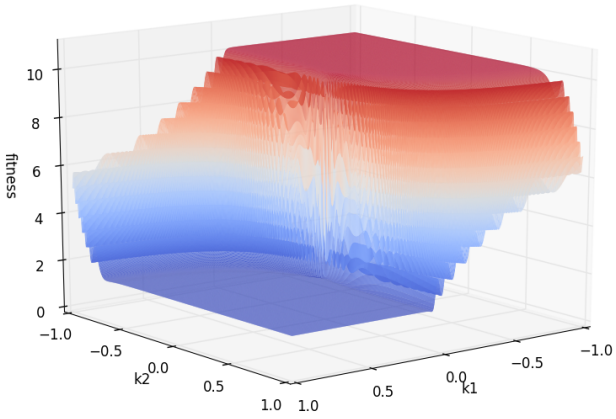
(a) Phenotype fitness



(b) Genotype



(c) Phenotype range



(d) Virtual fitness

Figure 3: One organism from experiment 1, "Razorback". The phenotype's fitness function, a landscape filled with local maxima and one wavy narrow ridge (a), has become distorted into the roughly hill-climbable virtual fitness function (d) seen by the "knobs" of the genotype (b). In (c), the phenotype range, x, y values indicate points in phenotype space that have a preimage in knob space when the knobs are mapped through m_{g_2}. The z values are the fitnesses of those phenotypes (the same as are plotted in (a)). In (b), the knob nodes are at the top, the phenotype nodes are at the bottom, numbers preceded by "i=" are initial activation levels, and the other numbers are activation levels after 10 timesteps.

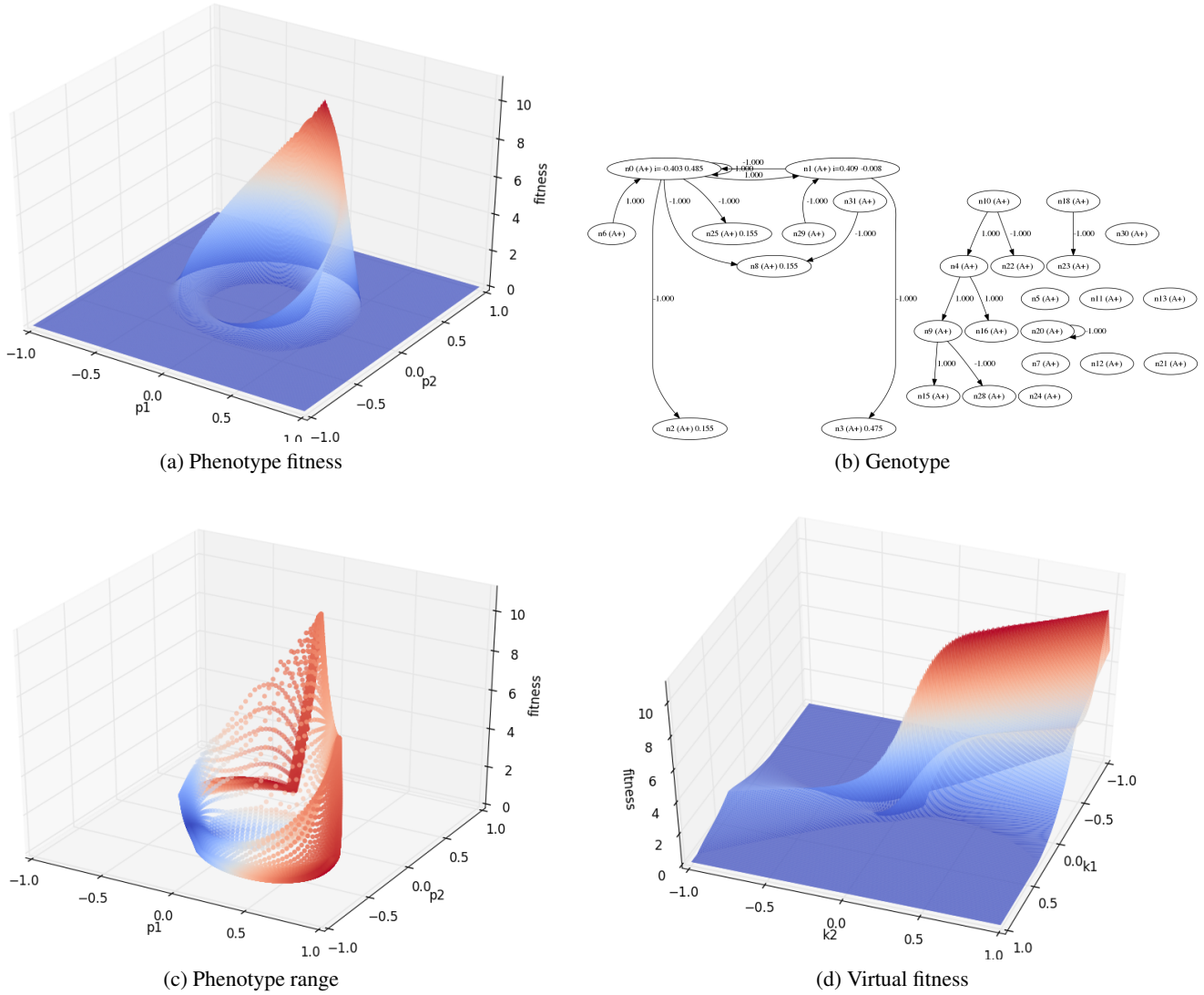
ping that squeezes the islands closer together in the virtual fitness function so that single knob-turns can leap the moats between them, as well as limiting the range of the virtual fitness function to the line along the centers of the islands.

Observations and Conclusions

The main result is that against these fitness functions, filled with traps that flummox direct evolution, a genome with continuously varying "knobs" mapped to its phenotype by a discontinuously varying topology or "graph" tends to evolve increasing evolvability by (a) presenting the knobs with a more hill-shaped virtual fitness function and (b) restricting the range of the knob–phenotype mapping to exclude "bad" parts of the phenotype space. Close observation of genotypes and lineages revealed a number of subtleties regarding how and when this process happens, explained below.

Limitations on Generality

Modelable Fitness Functions and Genetic Memory Over many generations, the graph accumulates a "memory"



(a) Phenotype fitness

(b) Genotype

(c) Phenotype range

(d) Virtual fitness

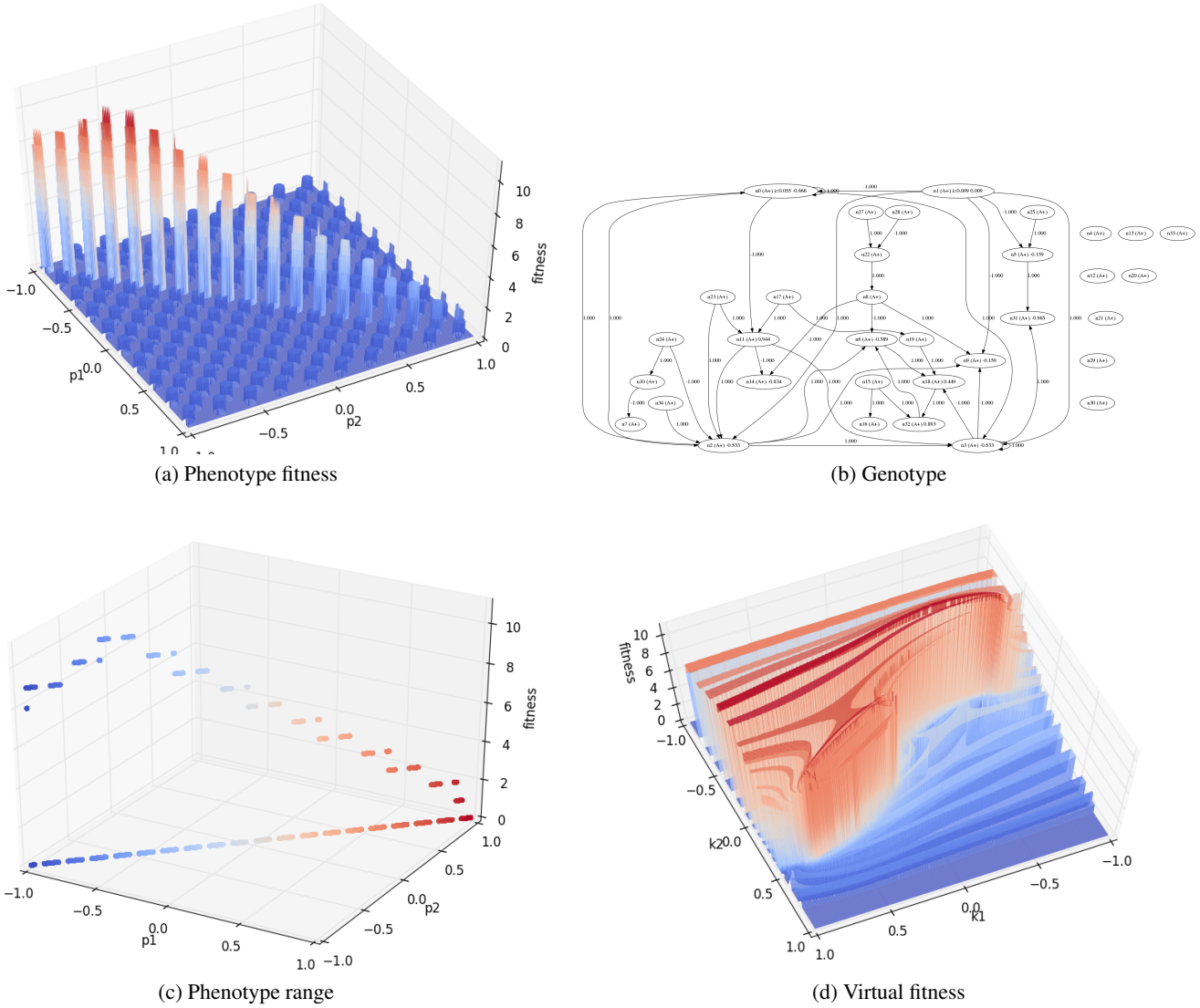
Figure 4: One organism from experiment 2, “Circle”.

of the family of whole-genome fitness functions, encoded in the form of *bias* in the way its lineage searches the phenotype space. This bias reflects invariants in the fitness family, such as where ridges occur and how they’re oriented, moat size, and where zero-fitness “deserts” consistently lie.

In effect, the graphs tended to evolve into models of the invariants in the family of fitness functions. This means that difficulties in modeling the invariants with a graph will shut down acclivation. For example, a graph can easily model the Razorback family because it lies along $y = x$, by disconnecting one knob and linking the other knob to both phenotype nodes. But a shifted Razorback family, say along $y = 2x - .4$, is much harder for the graph to model.

Knobs themselves cannot accumulate useful bias beyond being positioned where they will be mapped to high-quality phenotypes. This bias can be effective, though: we often observed several lineages in a single population with knobs positioned far apart, each ready to capitalize if the whole-genome fitness function or the genotype–phenotype mapping changes to favor them again.

Non-Stationary Fitness Function We frequently witnessed the decay of a population’s genetic memory. Many times, a population that was responding quickly to shifts of the fitness peak, moving rapidly toward it one knob-turn per generation, lost its ability to do this when a few epochs went by with little or no movement in the fitness peak. When the peak stayed constant too long, selective pressure favored canalization: genotype–phenotype mappings that held the phenotype at the peak in the face of most mutations. In



(a) Phenotype fitness

(b) Genotype

(c) Phenotype range

(d) Virtual fitness

Figure 5: One organism from experiment 3, “Moats”.

that circumstance, if knob-turning alters the phenotype at all, it lowers fitness, so selective pressure favors rendering it inert—that is, making a genotype–phenotype mapping in which all points in knob-space map to the same point in phenotype space.

Genetic memory also frequently decayed when a knob had a value of -1.0 or $+1.0$. A knob-turning mutation that goes beyond the limits has no effect. At these times, knob-turning often lost its sensitivity to the virtual fitness gradient—and so improvements in fitness had to come by graph-edits alone, which can spoil previous acclivation.

Thus a non-stationary fitness function is most favorable for acclivation: one that changes frequently, shifting the peak, while retaining invariants that the graph can model in a stable way. See Reisinger et al. (2005) for discussion of evolvability in connection with this kind of nonstationary fitness function, including a measure, *acquired evolvability*, of a genome’s ability to “represent” its invariants.

Mutation Rate When instead of limiting each offspring to a single mutation from its parent, we allowed a number of mutations proportional to the size of the parent (the number of nodes), genotypes tended to “bloat”, acquiring hundreds of disconnected nodes and edges. The problem is that when larger genotypes can make more mutations per generation, they have no incentive to optimize the way they respond to graph mutations. When each organism can only vary from its parent by a single mutation, those who do not optimize their mutation exposure are at a disadvantage in the race to the new peak at the start of each epoch. A lineage with an

unnecessarily large number of ways to make neutral mutations will tend to lose those races to lineages with the minimal amount of "padding".

The Transfer Function We expected that nearly any transfer function typically used in simulated activation networks would induce acclivation of virtual functions (given appropriate fitness functions, etc.), but this was decidedly not the case. When we tried a simple $y = x$ function clamped within $[-1.0, 1.0]$, step functions, rectifier functions, and letting the constant in the T function stray far from S all produced much less acclivation as well as phenotypes of much lower fitness. The graphs could not "lock on" to the ridge, making knob-turning nearly useless for navigating up the fitness functions.

The T function has a peculiar characteristic that makes it suitable for these experiments, where the only constants in the genotype that are allowed to vary in small increments are those in the knobs. T is expansive in the range $-.28 < x < .28$ and contractive everywhere else. When a constant input, as from a knob, is fed into T repeatedly ten times, this yields a function $T(x + T(x + T(\ldots)))$, which is expansive in $-.14 < x < .14$ and contractive everywhere else. This makes T well suited to forming a wide variety of functions that simultaneously dilate and compress different ranges of the phenotype space, by composition with itself alone—without constants. Activations from incoming edges a_i beyond the first edge make a node calculate the function $T(a + \sum_i a_i)$, giving compositions of T the ability to shift their output right or left.

Compositions of linear transfer functions can shift phenotype space but they can't dilate or compress it. This makes it harder, perhaps impossible, to evolve an acclivated virtual fitness function. When the constant S is varied too far from that in T, the resulting function's range of expansion quickly shrinks to a tiny region around $x = 0$ or grows to nearly the whole interval $[-1.0, 1.0]$.

Virtual Knobs

We ran variations on the above experiments where nodes other than the knobs were allowed to inherit constants. For example, we tried allowing non-knob nodes to inherit an initial activation. Under this condition, successful organisms tended to accumulate a collection of nodes with different constants, none of which were connected to the knobs and only one of which was connected to a phenotype node. They exploited the "move edge" mutation to make these collections of nodes function as a virtual knob. Both knob nodes were often disconnected from the rest of the graph.

The organisms seemed to prefer their virtual knobs. Virtual knobs are subject to evolutionary pressure determining how fast they turn, i.e. the probability distribution of knob-turning deltas. The hard-coded knobs are limited to deltas in the range of about ± 0.02. The evolved virtual knobs tended to turn much faster than our hard-coded knobs.

When we removed nodes with constants, we tried allowing more than one edge between nodes. The organisms evolved to exploit the "add edge" and "remove edge" mutations as knobs. The number of edges between two nodes effectively served as an adjustable multiplier.

To get the organisms to make use of our hard-coded knobs, necessary to examine virtual fitness functions whose domain is the knob settings, we had to purge the graph of all other constants capable of varying in small increments. This severely reduces evolvability.

Acknowledgements

We thank Jerry Swan (University of York), Zoltán Kocsis (University of Manchester), and Etienne Barnard (North-West University, South Africa) for helpful discussion and suggestions. We thank David Landy and Brad Rogers (Indiana University) and the anonymous reviewers for many valuable suggestions for improving the article. And thanks to Joonas Ilmavirta (University of Jyväskylä) for suggesting *acclivus* as the appropriate Latin root for "making sloped or hill-shaped."

References

Colegrave, N. and Collins, S. (2008). Experimental evolution: experimental evolution and evolvability. *Heredity*, 100(5):464.

Gavrilets, S. (1997). Evolution and speciation on holey adaptive landscapes. *Trends in ecology & evolution*, 12(8):307–312.

Kauffman, S. and Levin, S. (1987). Towards a general theory of adaptive walks on rugged landscapes. *Journal of theoretical Biology*, 128(1):11–45.

Kovitz, B. (2015). Experiments with cascading design. In *Proceedings of the 13th European Conference on Artificial Life (ECAL 2015), Workshop EvoEvo*.

Reisinger, J., Stanley, K. O., and Miikkulainen, R. (2005). Towards an empirical measure of evolvability. In *Proceedings of the 2005 workshops on Genetic and evolutionary computation*, pages 257–264. ACM.