Natural Selection of Paths in Networks

Chrisantha Fernando[1,2], *Vera Vasas*[3], *Eors Szathmary*[4], *Phil Husbands*[1]

[1]Department of Informatics, University of Sussex, Falmer, Brighton, BN1 9RH

[2]MRC National Institute for Medical Research, The Ridgeway, Mill Hill, London.

[3]Departament de Genètica i de Microbiologia, Grup de Biologia Evolutiva (GBE), Universitat Autònoma de Barcelona, 08193 Bellaterra, Barcelona, Spain;

[4]Collegium Budapest, Institute for Advanced Study, Szentháromság u. 2, H-1014

December 2010

Abstract

We present a novel algorithm that exhibits natural selection of paths in a network. If each node and weighted directed edge has a unique identifier, a path in the network is defined as an ordered list of these unique identifiers. We take a population perspective and view each path as a genotype. If each node has a node phenotype then a path phenotype is defined as the list of node phenotypes in order of traversal. We show that given appropriate path traversal, weight change and structural plasticity rules, a path is a unit of evolution because it can exhibit multiplicative growth (i.e. change it's probability of being traversed), and have variation and heredity. Thus, a unit of evolution need not be a spatially distinct physical individual. The total set of paths in a network consists of all possible paths from the start node to a finish node. Each path phenotype is associated with a reward that determines whether the edges of that path will be multiplicatively strengthened (or weakened). A pair-wise tournament selection algorithm is implemented which compares the reward obtained by two paths. The directed edges of the winning path are strengthened, whilst the directed edges of the losing path are weakened. Edges shared by both paths are not changed (or weakened if diversity is desired). Each time a node is activated there is a probability that the path will mutate, i.e. find an alternative route that bypasses that node. This generates the potential for a novel but correlated path with a novel but correlated phenotype. By this process the more frequently traversed paths are responsible for most of the exploration. Nodes that are inactive for some period of time are lost (which is equivalent to connections to

and from them being broken). This network-based natural selection compares favourably with a standard pair-wise tournament-selection based genetic algorithm on a range of combinatorial optimization problems and continuous parametric optimization problems. The network also exhibits memory of past selective environments and can store previously discovered characters for reuse in later optimization tasks. The pathway evolution algorithm has several possible implementations and permits natural selection with unlimited heredity without template replication.

Introduction

Units of evolution (Maynard Smith, 1986) at the same level of selection (Okasha, 2006) are generally considered to be discrete non-overlapping individuals, for example, living organisms, B-cells undergoing somatic selection, ribozymes in the RNA world, and binary strings in a genetic algorithm. A unit of evolution is any entity that has multiplication, variation and heredity. If units have differential fitness they can evolve by natural selection. The mechanism by which the above units maintain unlimited heredity depends on template replication; in the language of networks, this involves a new node and edge being formed for each node and edge of the parent by formation of a topographic correspondence with the original (Szathmáry, 2006). The fundamental process of natural selection with template replication is shown in Figure 1. It was template replication based natural selection that inspired John Holland’s genetic algorithm (Holland, 1975).

This paper proposes *an alternative mechanism by which adaptations can arise and increase in frequency in a population by natural selection, in the absence of template replication.* It is based on the idea that natural selection need not act between physically independent individuals as shown in Figure 1. Instead, natural selection can act on paths in a directed graph, e.g. in a neuronal network, if the covariance between the phenotype of that path and the fitness of that path is not outweighed by transmission bias due to mutational exploration, and environmental change (Price, 1970). The number of possible paths in a network can be far greater than the number of nodes or edges because each node and edge can be part of many pathways (Izhikevich, 2006). The number of path phenotypes is less than (and rarely equal to) the number of paths.

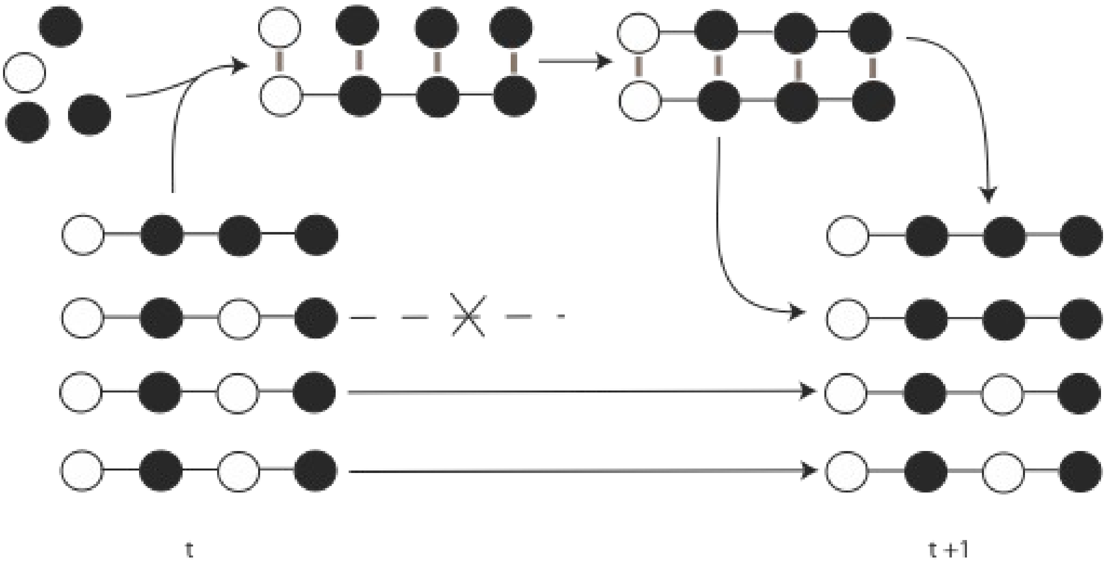


Figure 1. One generation of natural selection by template replication. At time t the population consists of 4 individuals with two phenotypes b_1 = 0101 and b_2 = 0111. The frequency of these phenotypes is q_1 = 3 and q_1 = 1. One generation involves template replication (possibly with mutation not shown) and removal of individuals to maintain the same population size. In the above diagram, this results in the same two phenotypes but with different frequencies $q_1 = 2$ and q_2 =2 respectively. The fact that phenotypic traits covary with fitness causes fitter traits to increase in the population (Price, 1970).

Let us consider some examples. Figure 2 shows two networks on the left, and all the paths they contain on the right. The top network contains two paths, each with a distinct phenotype. The pink path has phenotype 0101 and the green path has phenotype 0111. Unfilled circles represent nodes with node phenotype 0, and filled circles represented nodes with node phenotype 1. The network on the bottom of Figure 2 contains four paths, shown on the right. Three of the paths have the same phenotype (pink, 0101) and one path has phenotype 0111 (green).

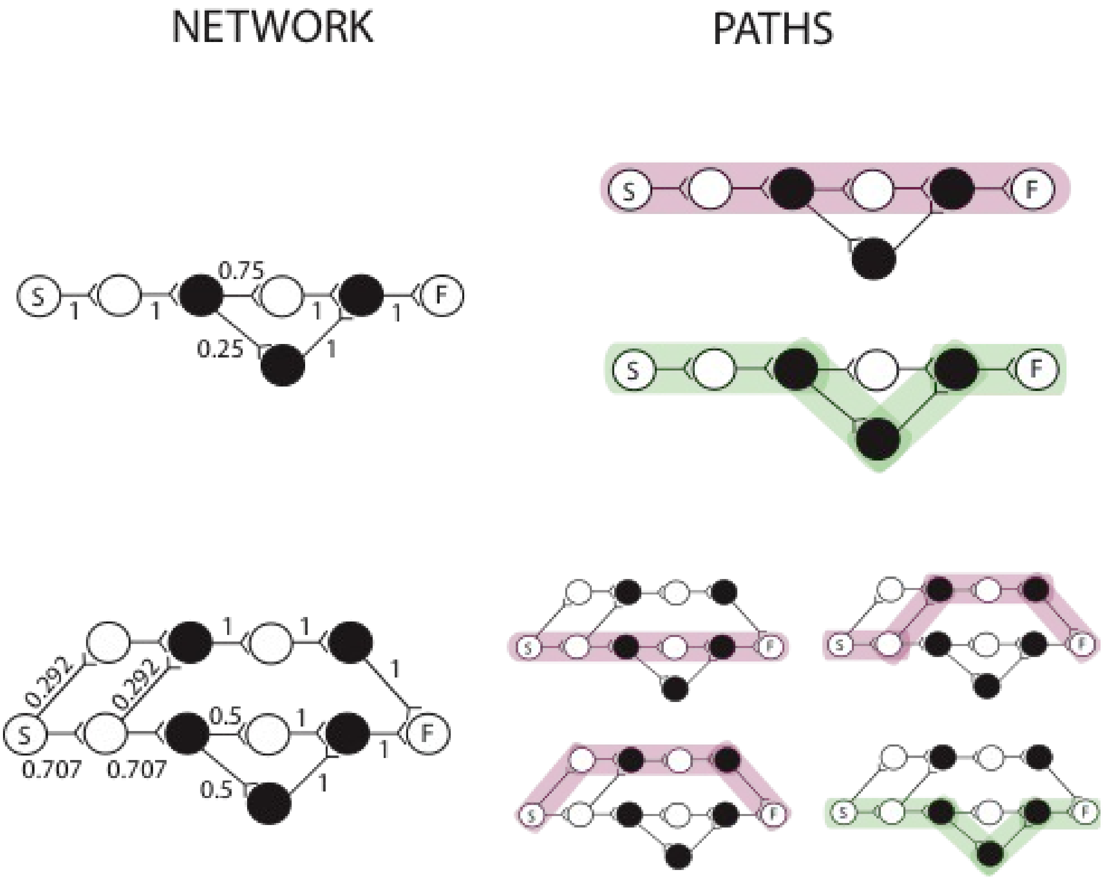


Figure 2. Two networks and the paths they contain. Paths with phenotype 0101 are shown in pink. Paths with phenotype 0111 are shown in green. The transition probabilities associated with each edge are marked.

Path Traversal

Note that each directed edge is associated with a weight between zero and one. The sum of weighs out of one node is always normalized to one after any weight change. Weights correspond to transition probabilities (weights) P_{ij} and are used to determine the frequency of a path. The probability of traversal (or in other words, the frequency) of that path is the product of the weights P_{ij} along that path. A node can be active or inactive. To generate a path, the start node is activated, and all other nodes are inactivated. In one time-step, the active node will then cause activation of *one* downstream node, chosen by roulette wheel selection over the outflow weights to all downstream nodes of the active node. The original active node is then inactivated. Therefore, at any one time, only one node is active in the network. This process iterates until the finish node becomes activated, at which point the path has been generated.

Given this probabilistic traversal scheme, it is easy to see that both networks at the top and bottom of Figure 2 have the same relative frequency of phenotypes as at time *t* in the traditional template based natural selection scheme shown in Figure 1. Each phenotype *b*, e.g. 0101, we will index with *i*, giving b_i. Each phenotype b_i has frequency q_i. The frequency q_i of a phenotype is defined as the sum of the frequencies of paths with that phenotype b_i. The frequency of an individual path is the proportion of times that that particular path is traversed when the start node is stimulated. Note that the fact that two different networks can produce the same frequency of phenotypes (as in the top and bottom networks in Figure 2) means there is a redundant (many-to-one) encoding of phenotypes by paths, and this may permit non-trivial neutrality (Toussaint, 2003), i.e. the probability distribution of phenotypes reachable by single mutations of paths may differ depending on the underlying configuration of paths that generated them.

Paths are Units of Evolution

Paths exhibit multiplicative growth. The increase or decrease of the frequency of a path occurs because there is strengthening (or weakening) of the transition probabilities P_{ij} along a path. Whether there is strengthening or weakening of these transition probabilities depends on the reward obtained by a path. Multiplicative growth of paths is necessary for them to be units of evolution.

Paths exhibit variation. Variation exists because each path can have a distinct path phenotype caused by the order of node phenotypes along that path. Paths exhibit heredity by two mechanisms. Firstly, when a path undergoes multiplicative growth by increasing P_{ij} along that path, i.e. when its frequency increases, this results in the increase of the frequency q_i of its associated phenotype b_i in the population of path phenotypes. Secondly, when a path mutates (to be described later) correlated variability exists because a new path phenotype, whilst not identical to the parental path phenotype, will still resemble the parent’s path phenotype because a mutant path is always a short bypass of the parental path and therefore overlaps with much of the parental path, i.e. like begets like. Correlated variability was shown to be a fundamental requirement for evolvability that was lacking in a previous proposal of an alternative to template replication due to compositional inheritance (Vasas, Szathmáry, & Santos, 2010). Hereditary variation of paths is necessary for them to be units of evolution.

Node Mutations

The mechanism of pathway mutation is shown in Figure 3A. This shows on the left a mutation of the first node of the network shown at the top of Figure 2. Mutants occur with a certain probability, μ, each time a node is activated. A node mutation involves creating a new node at the same layer (drawn in the figures above or below the parent node). The new node has weak initial connection strength from the node that activated the parent node, and a connection of strength 1 to the node that was activated by the parent node. This preserves the original paths, yet creates new alternative paths. The path phenotypes of the alternative pathways will be correlated with the path phenotypes of the paths that contain the node that underwent a mutation. Initially the alternative paths are traversed with low probability, in other words the frequency q_i^m of a mutant path phenotype b_i^m in a population of path phenotypes will be low, if that path phenotype did not previously exist in the population. Note that this kind of mutation could not occur in the population shown in Figure 1. Because a node can be involved in many paths each having different path phenotypes, a single node mutation can change the frequency of many path phenotypes at the same time. This is one of the features that distinguish the path evolution algorithm from a standard genetic algorithm.

Path Crossover

Path crossover occurs with probability χ whenever two distinct paths differ in reward, see Figure 3B. A weak weight is formed from a random layer in the loosing path to the next layer in the winning path. Another weak weight is formed from a random layer (after the first point of crossover) in the winning path to the next layer in the losing path. Thus, this is a two-point crossover that creates a new weak path that consists of part of the loosing path and part of the winning path.

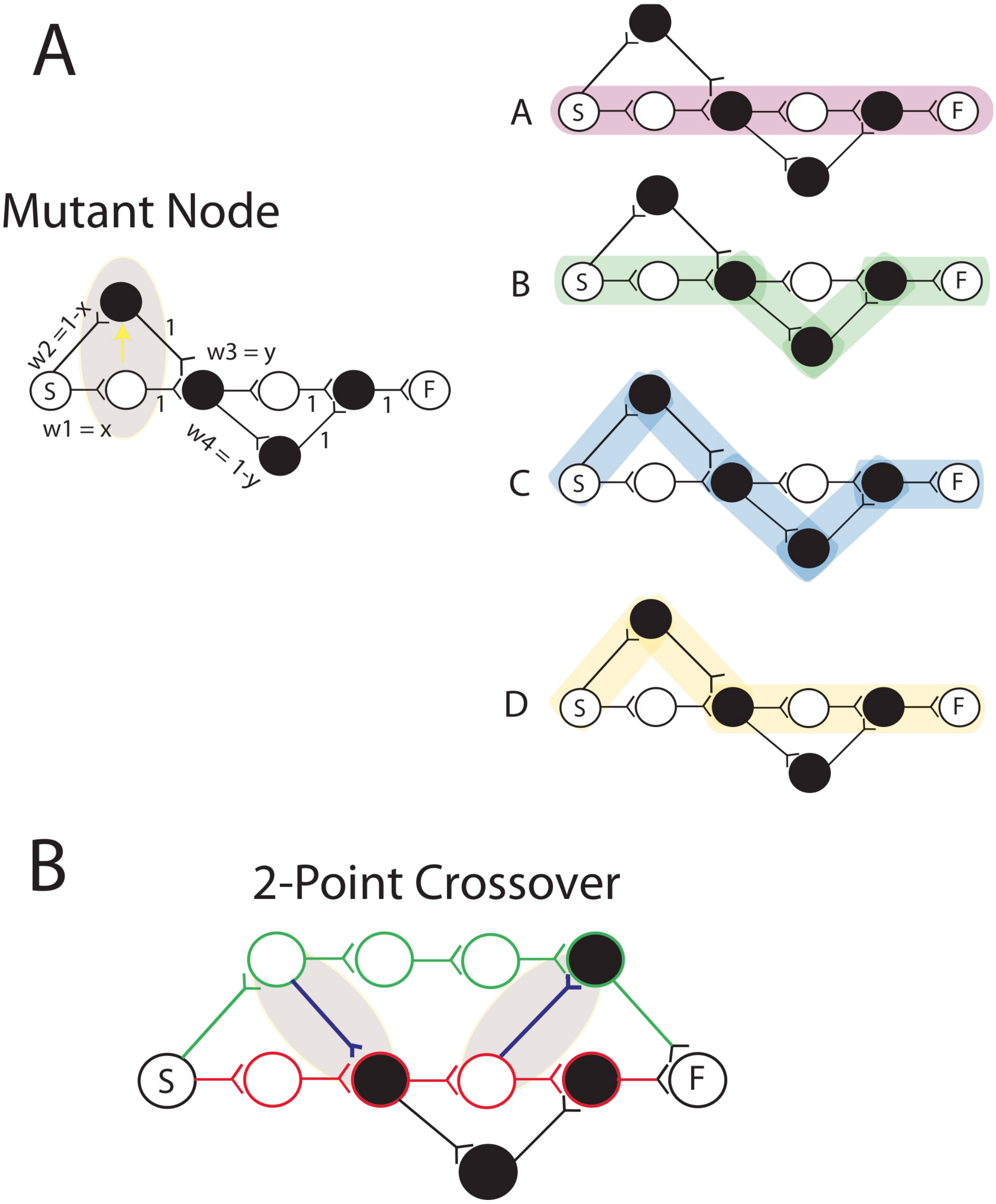


Figure 3 (Part A) A single mutation to the network in Figure 2 produces two new paths and two new path phenotypes (Right). (Part B) 2-point crossover between a winning path (green) and a losing path (red).

Evolutionary Dynamics of Paths in Fixed Networks

Let us consider the evolutionary dynamics over one generation of the simple network at the top of Figure 2. The frequency of the pink path is also the frequency of the path phenotype b_1 = *0101*, namely q_1 = 0.25, because only one path has that phenotype.

The frequency of the path phenotype b_2 = *0111* is q_2 =0.75, and is the frequency of the green path. For more complex networks the frequency of a path phenotype will be the sum of the probability of taking all paths with that phenotype. Now we have the frequencies of phenotypes in the ancestor generation at time *t*. Let us assume that b_1 has reward r_1 *=2* and b_2 has reward r_2 *=3*. Ignoring mutation for now, let one generation consist of choosing two paths. Each path is generated according to the roulette wheel traversal method described previously. From these two paths the winning path is chosen as the path with the highest reward associated with it. The probability of choosing path 1 twice is P(1,1) = $(0.75)^2$. The probability of choosing path 1 and path 2 is P(1,2) + P(2,1) = 2(*0.75)0.25* . The probability of choosing path 2 twice is P(2,2) = 0.25^2. Only when different paths (with distinct path characters) are chosen is a winner and looser defined. Therefore, with probability 0.375 per generation, path 2 will be chosen as the winner and path 1 as the looser. The transition probabilities P_{ij} are then modified as follows. The edges along the winning path (not shared by the losing path) will be strengthened according to the following rule…

$$\Delta P_{ij} = (1 + l\,)P_{ij} \qquad \ldots(1)$$

and the edges along the losing path (not shared by the winning path) will be weakened according to the following rule…

$$\Delta P_{ij} = (1 -\ l\,)P_{ij} \qquad \ldots(2)$$

for the losing path, followed by normalization over each set of outflow edges for which weights were changed. Specifically, if λ = 0.1 then the weight of the edge to path 1 will decrease from 0.75 to 0.75 x 0.9 and the weight on the edge to path 2 will increase by 0.25 x 1.1, which after normalization gives values new transition probabilities 0.71 and 0.29 respectively. By this learning rule the path character with the higher reward increases in the population and the path character with the lower reward decreases.

Let us consider a more general formulation of the above dynamics. Supplementary Material contains a Mathematica file that shows a deterministic model constructed with dynamical equations that captures the essence of natural selection in these path-based systems. A path is a genotype. A node on a path is an allele. A locus consists of all nodes on paths a certain distance away from the start node (i.e. in the same layer). The frequency of a path is the probability that activity passes along that path when the start node is stimulated. The frequency of a phenotype is the probability that that phenotype will be produced when the start node is stimulated. An understanding of the system will involve a description of the dynamics and links between these various entities.

The kind of network at the top of Figure 2 can be considered as a system with one locus and two alleles. The two alleles are the two parallel nodes at the same locus (layer) of each path. Let us set the initial weight to one of these nodes as w_1 and the other weight $w_2 = 1\text{-}w_1$ because the total outflow weight from the common preceding node must sum to 1. Weight change only occurs if two different paths are chosen in the two traversals available in each generation. Therefore, weight change occurs with probability $2w_1(1\text{-}w_1)$. With probability $1\text{-}2w_1(1\text{-}w_1)$ there is no weight change. Let us assume (without loss of generality) that the winning path (i.e. the path with higher reward) is associated with the node with weight w_1. Then the new weight at time t+1 of w_1 is given by

$$\mathrm{w}_{,t+11} = 2w_1(1-\ w_1)\frac{w_1(1+l\ )}{w_1(1+l\ )+(1-\ w_1)(1+l\ )} + (1-\ 2w_1(1-\ w_1))w_1 \qquad \ldots(3)$$

For initial values $w_1 = 0.1$, $w_2 = 0.9$, and $\lambda = 0.1$, this gives the dynamics shown in Figure 4.

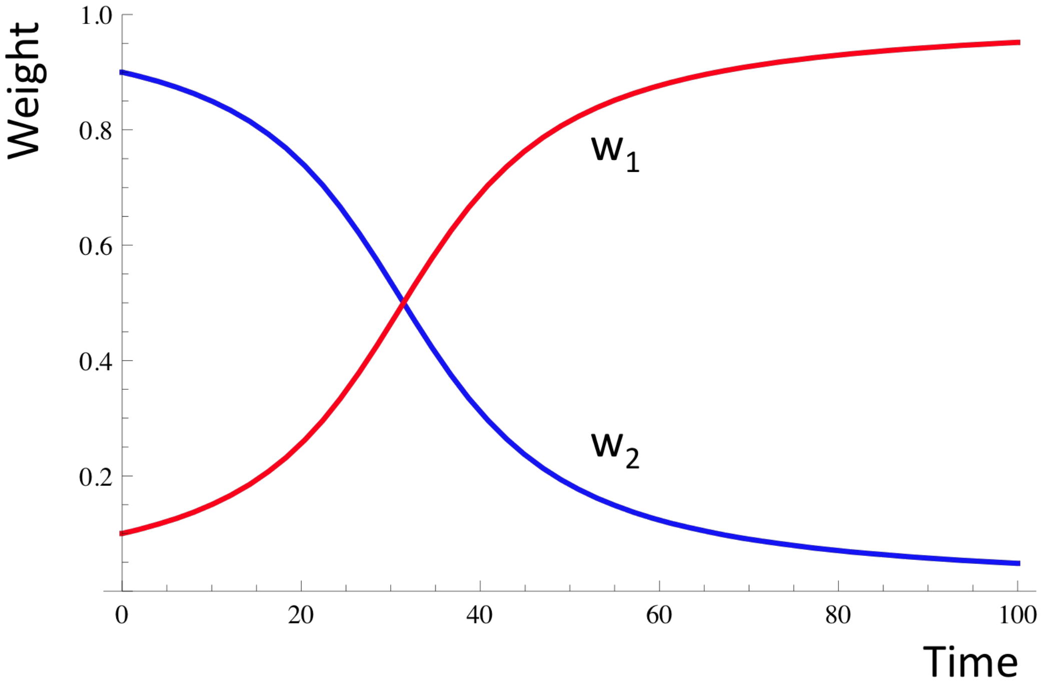


Figure 4. Selection between two alleles at one locus. The allele associated with higher reward reaches fixation, whilst the other allele goes extinct.

The path (and phenotype) associated with higher reward reaches fixation, whilst the one with the lower reward goes extinct.

Now let us consider the more complex network in Figure 3A. Here there are four paths and four phenotypes, or two loci with two alleles at each locus. Let the two weights at the first locus be w_1 = x and w_2 = (1-x) and we two weights at the second locus be w_3 = y and w_4 = 1-y. The frequency of each path is then…

$$\begin{aligned} P(A) &= xy \\ P(B) &= x(1-y) \\ P(C) &= (1-x)(1-y) \\ P(D) &= (1-x)y \end{aligned} \qquad \ldots(4)$$

Again, the weights associated with the winning path are changed as in (1) and the weights associated with the losing path as in (2) followed by normalization. Consider the cases in which w_1 and w_2 will change. This happens only when the path pairs AC, AD and BC are traversed with probabilities *P(AC) = 2 P(A) P(C)*, *P(AD) = 2 P(A) P(D)* and *P(BC) = 2 P(B) P(C)*, respectively. When the other pairs are traversed, either fitness is identical and there is no change in weights, e.g. (B & D), or the paths do not differ at the w_1 and w_2 edge, e.g when paths (D&C) or (A&B) are taken. Assume that in this case we wish to minimize the number of 1's (filled circles) in each path. Looking at each case in turn then, A beats C, A beats D, and B beats C, and so w_1 will always be strengthened or not changed at all in each generation according to the following equation…

$$w_{1,t+1} = [P(AC)+P(AD)+P(BC)]\frac{w_1(1+l)}{w_1(1+l)+(1-w_1)(1+l)} + [1-(P(AC)+P(AD)+P(BC))]w_1$$

…(5)

Note that w_2 is just $1\text{-}w_1$. Similarly, w_3 an w_4 will only change when path pairs AB, AC, and CD are traversed in a generation. Figure 5 shows the vector field of the Δw_1 and Δw_3 for the various possible values of w_1 and w_3, and the dynamics of allele frequencies and phenotype frequencies over time for initial conditions w_1 = 0.2, and w_3 = 0.1, and λ = 0.1.

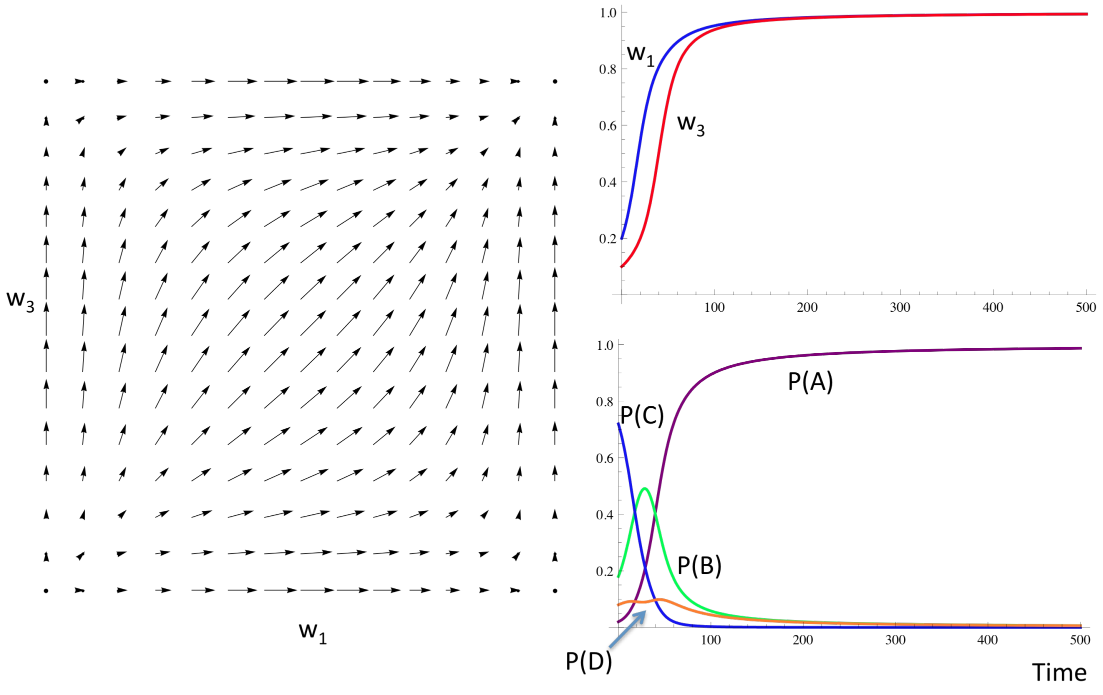


Figure 5. Selection at two loci, each locus having two alleles. The two fitter alleles (with weights w_1 and w_3) reach fixation whilst the other alleles (w_2 and w_4) go extinct.

The fittest alleles (w_1 and w_3) and the fittest path, A, go to fixation, whilst the other alleles and paths go extinct. As the vector field shows this is inevitable from any initial condition of w_1 and w_3. Effectively the two alleles are in linkage equilibrium.

Linkage Disequilibrium of Alleles in Paths

The network in Figure 6A is initially fully connected (in the forward direction). It has two loci, each with two alleles. We show that it is possible to establish linkage disequilibrium by weight change alone. Consider the case where the ordering of reward is 10 > 01 > 11 = 00. Supplementary Material shows a deterministic model of how the weights *x*,*y* and *z* change over time to send the fittest path 10 to fixation. Alternatively, if the fitness function is 10 = 01 > 11 = 00, both paths 10 and 01 are maintained at non-zero probability, the ratio depending on the initial value of the weight *x*. The initially more frequent of the 10 and 01 paths reaches a higher steady state value, see Figure 6B.

The capacity to maintain non-random assortment of the alleles by i. maintaining 10 and losing 01 (in the selective case) and ii. by maintaining B and C at different frequencies in the neutral case shows the capacity for linkage disequilibrium. The network converges to make one path in the selective case, and two non-overlapping paths in the neutral case. As we saw in Figure 3A, there are some networks that will not permit the maintenance of linkage disequilibrium because it is impossible to establish two non-overlapping paths because of a node bottleneck. In this case, mutations will be required to produce a greater number of nodes at that locus, so that paths can pass without overlapping with each other, thus maintaining multiple linkage disequilibria (pairwise associations) between alleles at loci on either side of the bottleneck.

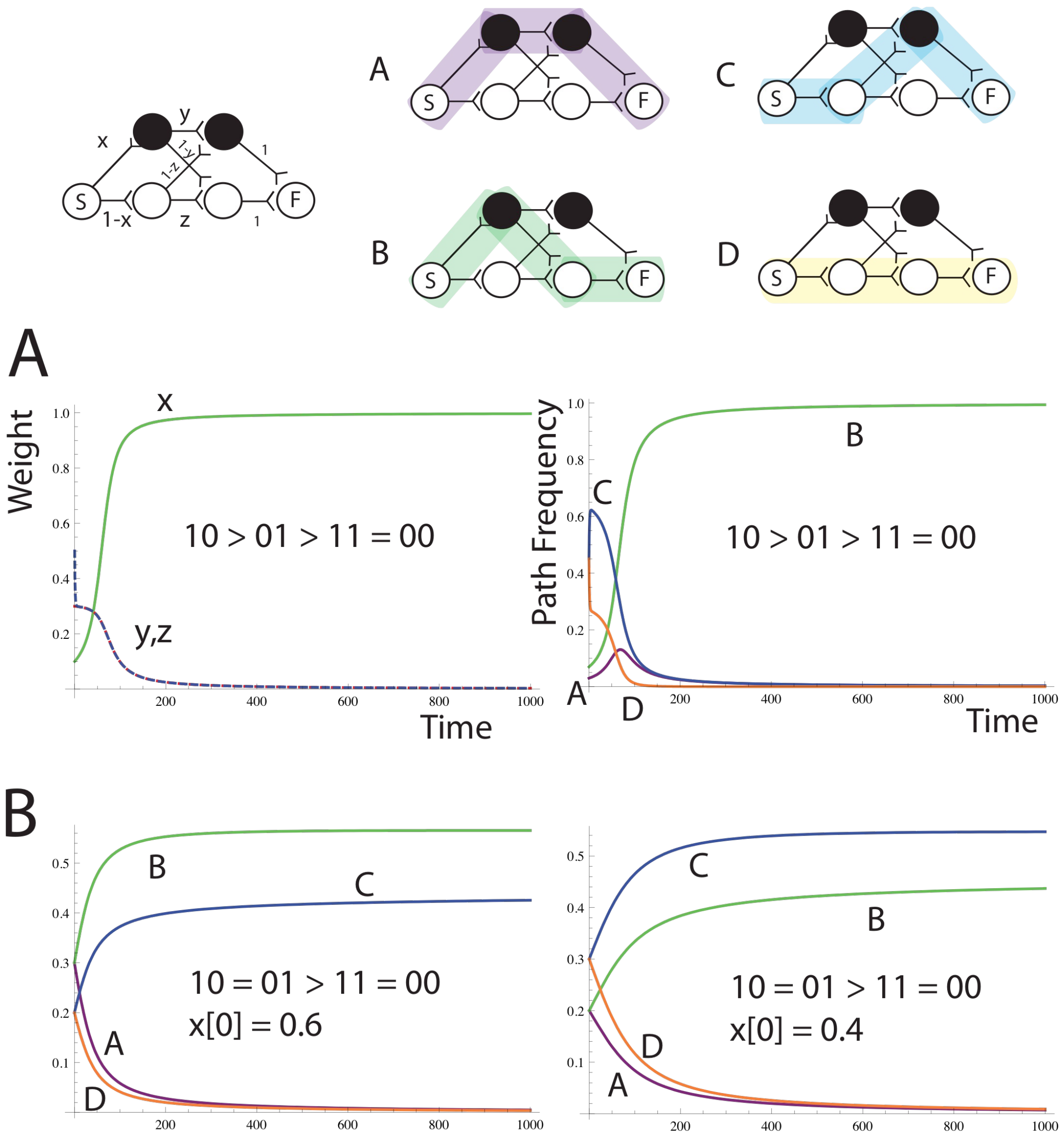


Figure 6. The network has three parameters x,y and z, and encodes four paths, A,B,C and D. Part A shows the dynamics of weights and path frequences for the fitness function 10 > 01 > 11 = 00. Path 10 (B, green) reaches fixation, and all other paths go extinct. Part B shows the dynamics of paths for the fitness function 10 = 01 > 11 = 00, for different initial weights of x of 0.6 and 0.4. Non-overlapping paths B and C are maintained at different concentrations that depend on the initial value of x. Paths A and D again go extinct.

Assignment of Path Phenotypes

The reward obtained by a path is a function of its phenotype b_i. The assignment of a phenotype to a path is determined by how the path interfaces with the environment. Figure 7 shows some examples of paths and their path characters and how these path characters may be associated with reward in various implementations of pathway evolution. See the Discussion for further implementation details.

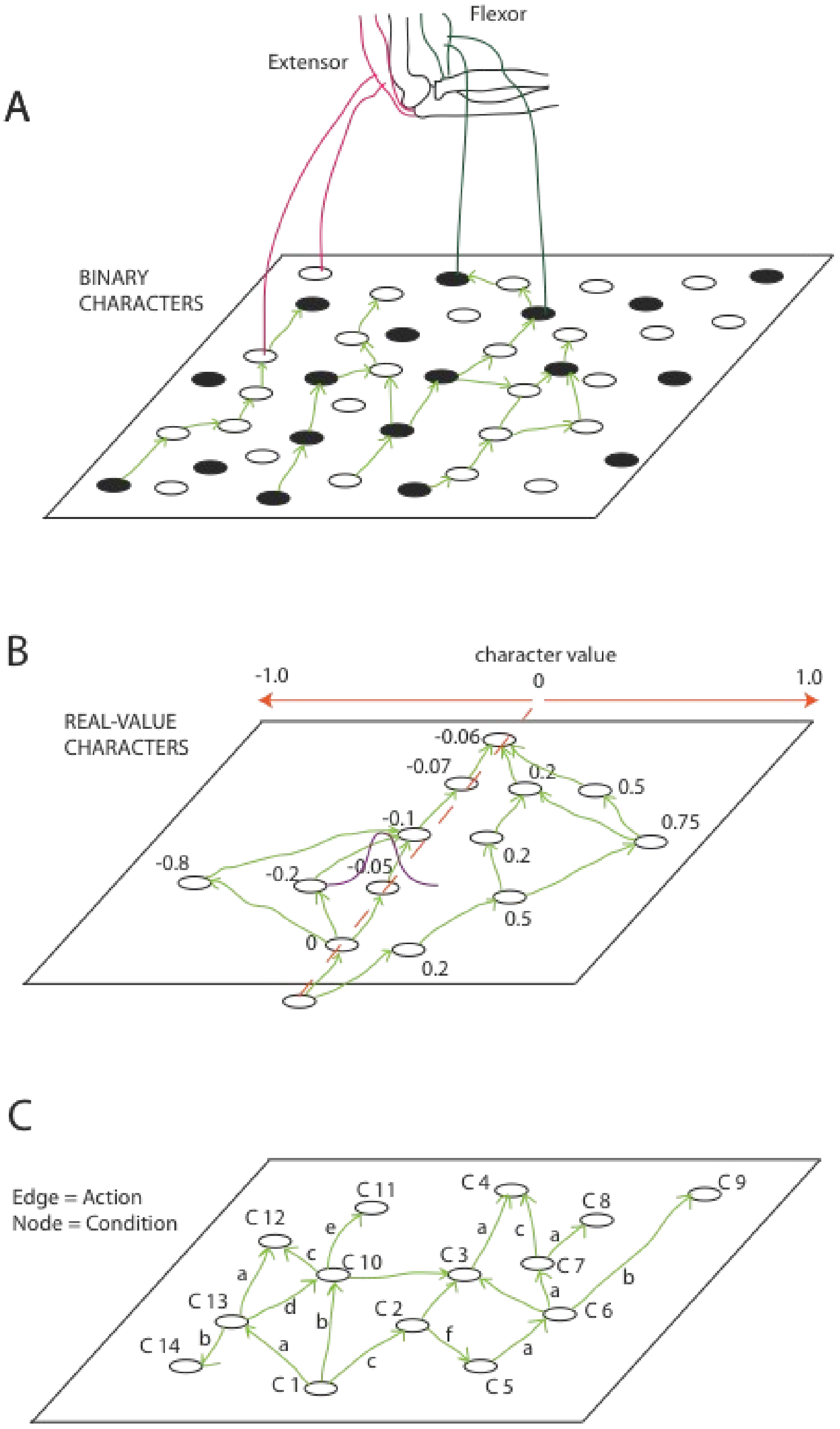
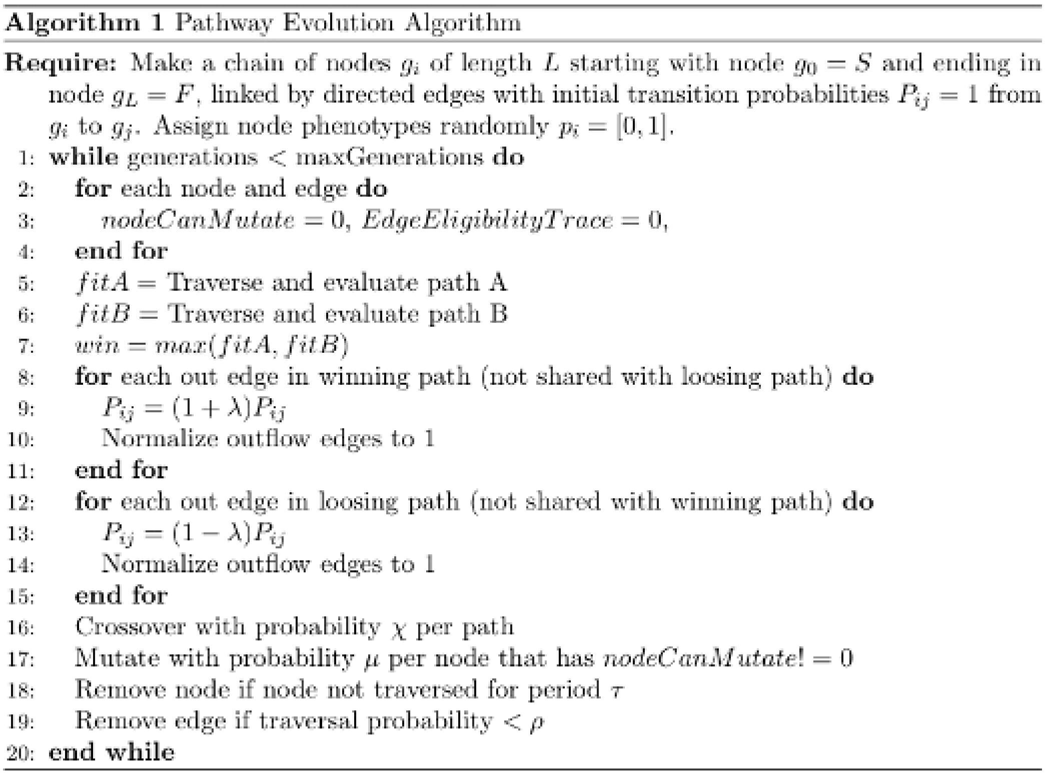


Figure 7. Different ways in which a path can have a character (phenotype). (A) Each node has a binary character, nodes labelled 0 activate an extensor muscle, nodes labelled 1 activate a flexor muscle. **(B)** The position of a node along the x-axis determines a real-valued character from -1 to 1. Mutation is Gaussian in position along the x-axis, i.e. a bypass mutant is likely to be to nearby neurons. **(C)** An anticipatory classifier system can evolve by a modification of PE if nodes are conditions and edges are actions. A condition (t) – action – condition (t+1) triplet is a classifier.

The full details of a pathway evolution (PE) algorithm are given in the Methods section, and the C++ code is available in Supplementary Material. The Results section compares the performance of PE algorithm with various parameter settings against a tournament selection based genetic algorithm (Harvey, 2009) on various combinatorial and real-value optimization problems, and for evolution in variable environments. Finally possible natural and artificial implementations of the PE algorithm are outlined.

Methods

The PE algorithm is described in Figure 8 and Figure 9.



```
Algorithm 1 Pathway Evolution Algorithm
Require: Make a chain of nodes $g_i$ of length $L$ starting with node $g_0 = S$ and ending in
    node $g_L = F$, linked by directed edges with initial transition probabilities $P_{ij} = 1$ from
    $g_i$ to $g_j$. Assign node phenotypes randomly $p_i = [0, 1]$.
while generations < maxGenerations do
    for each node and edge do
        $nodeCanMutate = 0$, $EdgeEligibilityTrace = 0$,
    end for
    $fitA$ = Traverse and evaluate path A
    $fitB$ = Traverse and evaluate path B
    $win = max(fitA, fitB)$
    for each out edge in winning path (not shared with loosing path) do
        $P_{ij} = (1 + \lambda)P_{ij}$
        Normalize outflow edges to 1
    end for
    for each out edge in loosing path (not shared with winning path) do
        $P_{ij} = (1 - \lambda)P_{ij}$
        Normalize outflow edges to 1
    end for
    Crossover with probability $\chi$ per path
    Mutate with probability $\mu$ per node that has $nodeCanMutate! = 0$
    Remove node if node not traversed for period $\tau$
    Remove edge if traversal probability $< \rho$
end while
```

Figure 8. The PE Algorithm Outline. See Figure 6 for details of the path traversal, crossover and mutation functions.

A network is initialized with *N* parallel linear directed paths (typically 1, 10 or 100 paths) of *L* nodes in length. The simplest case described above involves *N = 1,* i.e. the system starts with a single path of nodes. Let the first node be the start node that will be stimulated at the onset of each fitness evaluation. Each directed edge has associated with it a transition probability P_{ij}. Initially all probabilities along the chain are set to 1. If the system is initialised with more than one parallel chain then the sum of probabilities out of the start node to the first node of each chain are normalized to one so that each chain is equally likely to be traversed. Upon stimulating the start node each node in an activated chain will fire sequentially until the end of the chain is reached[1]. If there are many output edges from a node, only one of the post-synaptic nodes can become active. This ensures that only one path is active at one time. A noise term can be introduced to the transition probability to promote exploration.

Because each node has a particular characteristic, each path of activity also has a

[1] Pathways with loops are not described here for this complicates the issue considerably and requires nodes with a refractory period for example.

characteristic. For example, if we wish to implement a binary genetic algorithm using this network then a node should be interpreted as having a label (a phenotype unrelated to the network dynamics described here) of zero or one, see Figure 4A. We randomly initialize the node phenotypes of the initial chain[2]. For example, activity passing along the initial chain may produce the phenotypic sequence 0111010001. Let each phenotypic sequence so produced be associated with a reward *r*, as defined by a fitness function.

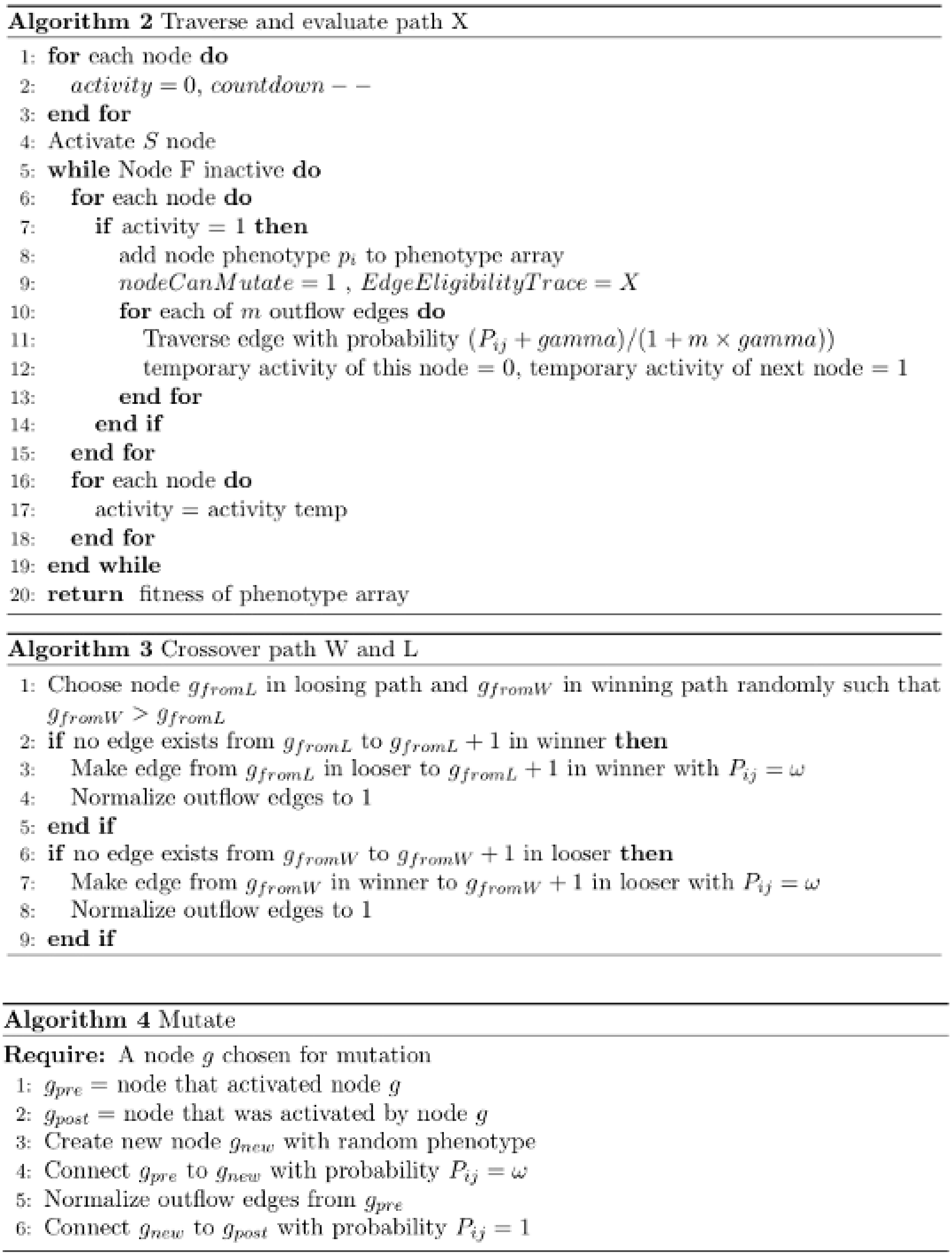
Now, at each generation, two paths are generated and the reward due to each path is determined on the basis of the path characteristic. If these two paths have differential reward they compete with each other for resources. This is a tournament selection method as used in steady-state genetic algorithms. The edges along the winning path are *multiplicatively* strengthened and the edges along the losing path are multiplicatively weakened, following which all outflow edge probabilities are normalized at each node in the path. Note that if the two paths in a single tournament spatially overlap and share edges, then these particular edges are not modified. Later we investigate a diversity maintenance mechanism that involves weakening shared edges. This multiplicative element of transition probability change contributes to the autocatalytic nature of a unit of evolution, with strong paths growing faster than weak paths. Note that each of the two paths should have distinct eligibility traces that can be used to allocate the delayed reward appropriately.

As well as traversal probability changes modulated by reward there are structural plasticity operations occurring in the network that create and destroy edges. Node mutations can occur with a certain probability μ per node whenever that node is active. Mutation of a node occurs by choosing an active node *g* in a traversed path and creating a new node. The node that activated the node *g* now activates the new node, and the new node activates the node that was activated *by* the node *g*. This biases mutation to make more bypass mutant grafts around the fitter paths. Also it is possible to imagine the mutation operation as not one of creating a new node, but rather of co-opting an unused node from an existing node in the vicinity[3]. Crossover from the

[2] In the knapsack problem the initial phenotypes of nodes are set to zero.

[3] Less specific variants of the mutation operator have been explored, e.g. allowing a new node to have connections from *all* nodes that were connected to its parent, or allowing it to connect to all nodes to which its parent node was connected.

losing path to the winning path and back again to the losing path may occur in some runs with a low probability χ. This is a 2-point crossover operation that allows utilization of the useful parts of a winning path by the losing path. If a node is not active in some time period it is removed. Also, if a transition probability sinks below some threshold value, it is removed.



```
Algorithm 2 Traverse and evaluate path X
for each node do
  $activity = 0$, $countdown - -$
end for
Activate $S$ node
while Node F inactive do
  for each node do
    if activity = 1 then
      add node phenotype $p_i$ to phenotype array
      $nodeCanMutate = 1$ , $EdgeEligibilityTrace = X$
      for each of $m$ outflow edges do
        Traverse edge with probability $(P_{ij} + gamma)/(1 + m \times gamma))$
        temporary activity of this node = 0, temporary activity of next node = 1
      end for
    end if
  end for
  for each node do
    activity = activity temp
  end for
end while
return fitness of phenotype array
```

```
Algorithm 3 Crossover path W and L
Choose node $g_{fromL}$ in loosing path and $g_{fromW}$ in winning path randomly such that
  $g_{fromW} > g_{fromL}$
if no edge exists from $g_{fromL}$ to $g_{fromL} + 1$ in winner then
  Make edge from $g_{fromL}$ in looser to $g_{fromL} + 1$ in winner with $P_{ij} = \omega$
  Normalize outflow edges to 1
end if
if no edge exists from $g_{fromW}$ to $g_{fromW} + 1$ in looser then
  Make edge from $g_{fromW}$ in winner to $g_{fromW} + 1$ in looser with $P_{ij} = \omega$
  Normalize outflow edges to 1
end if
```

```
Algorithm 4 Mutate
Require: A node $g$ chosen for mutation
$g_{pre}$ = node that activated node $g$
$g_{post}$ = node that was activated by node $g$
Create new node $g_{new}$ with random phenotype
Connect $g_{pre}$ to $g_{new}$ with probability $P_{ij} = \omega$
Normalize outflow edges from $g_{pre}$
Connect $g_{new}$ to $g_{post}$ with probability $P_{ij} = 1$
```

Figure 9. Details of path traversal, crossover and mutation operators.

We see that a path is a unit of evolution because it multiplies, i.e. the frequency of path in a population of paths can increase multiplicatively, there can be variants, i.e. there are many different path phenotypes, and there is heredity, that is, a node mutation will transform existing path phenotypes into new path phenotypes that resemble the original ones (like begets like).

Results

Several optimization benchmarks were used to characterise the PE algorithm with different parameter settings. Performance is compared to a microbial GA on the binary multiple knapsack problem and on a set of parametric optimization problems. The ability of the algorithm to show memory of past solutions is demonstrated.

Combinatorial Optimization Problems

The binary multiple knapsack problem is an extension of the simple 0/1 knapsack problem, on which genetic algorithms have been somewhat successful (Chu & Beasley, 1998). A knapsack has capacity *C*, and there are *n* objects. Each object has weight w_i, and a profit p_i. We aim to fill the knapsack for maximum profit but without exceeding its capacity, i.e. to find a vector $\boldsymbol{x} = (x_1, x_2 \ldots x_n)$ where $x_i \in [0,1]$, such that $\sum_{i=1}^{n} w_i x_i \leq C$ and for which $P(\vec{x}) = \sum_{i=1}^{n} p_i x_i$ is maximum. In the multiple knapsack problem, thee are *m* knapsacks. Each object is either placed in all *m* knapsacks, or in none at all. Each of the *m* knapsacks has capacity $c_1, c_2 \ldots c_m$, and each objects has a *different* profit in each knapsack, i.e. each objects is defined by a profit vector of length *m*. Again, no knapsack must be overfilled and maximum profit must be packed.

A typical run on the hard Weing8 instance of the knapsack problem (Khuri, Back, & Heitkotter, 1994) is shown in Figure 10. This is a hard knapsack problem with 105 objects and 2 knapsacks in which most pack vectors result in overfilling. To deal with this a punishment term is used in the fitness function that gives a negative fitness that is the extent of overfilling in all knapsacks. Otherwise the fitness is the profit of the knapsacks. On 30 independent runs, the PEA had a mean score of 615368 (sd = 7272) and the microbialGA with population size 100 had a mean score of 600236 (sd = 20003). The best solution obtained with the PEA was 622352 and the best score with the microbialGA was 623459 (the global optimum). In another knapsack problem (Weish25) the microbialGA obtained mean = 9900, sd = 40.8 and max = 9936 (the global optimum) over 30 trials, whilst the PEA obtained a mean of 9925, sd = 21.5, and max = 9936. The PEA and the microbialGA are comparable on these problems.

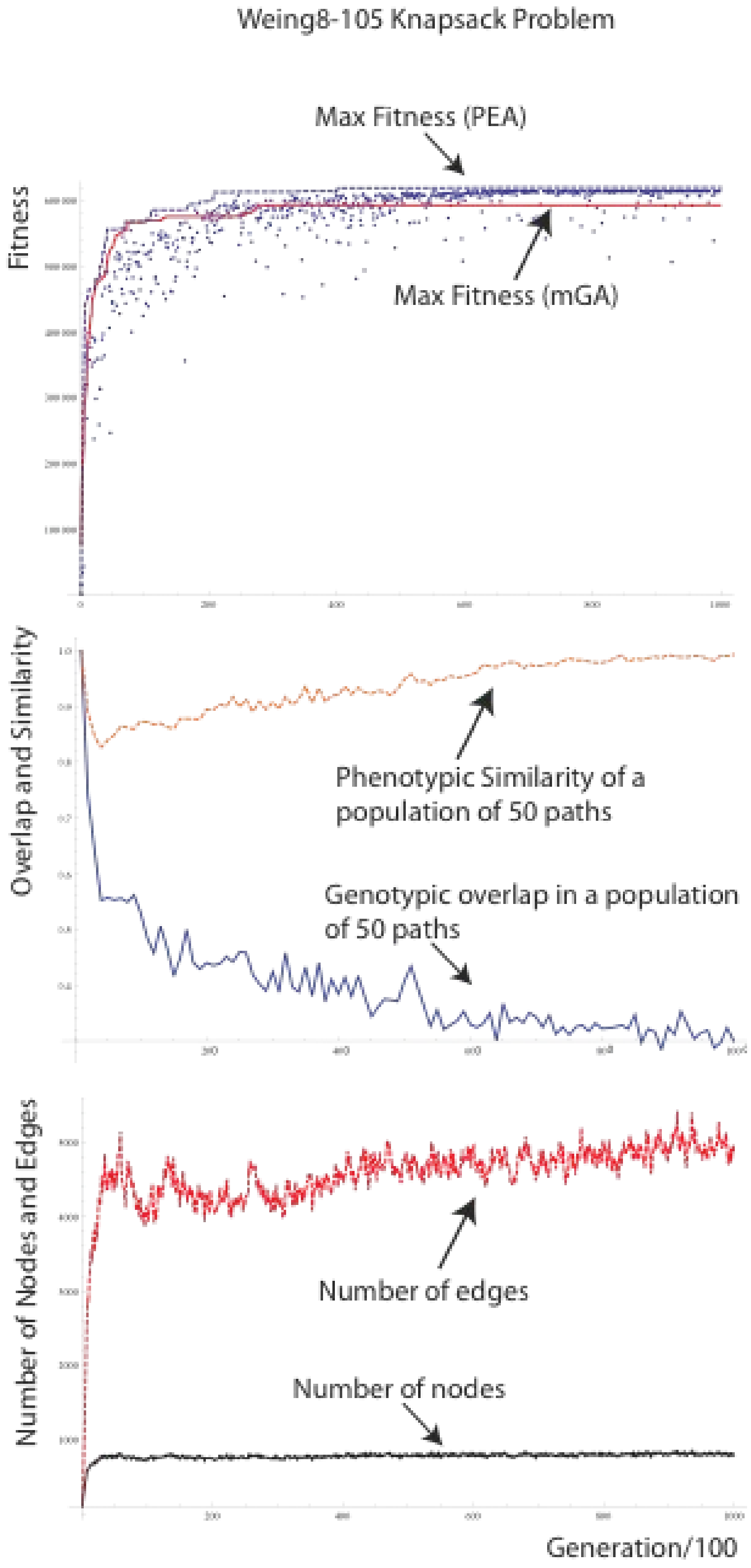


Figure 10. Performance of the PEA compared to a microbialGA with population size 100 and the same mutation rate on the Weing 8-105 knapsack problem. Max fitness achieved by the PEA = 620060 which is the 7th best possible packing, the maximum being 624319. The following parameters were used. $N = 1$, $L = 105$, $\lambda =$

0.1, μ = 1/L, χ = 0 (no crossover), τ = 200, ω = 0.01, ρ = 0, γ (gamma) = 0. The PEA is run for 10000 generations, i.e. 20000 pathway fitness evaluations.

Continuous Parametric Optimization Functions

A continuous value variant of the PE algorithm can be defined straightforwardly as in Figure 7B. Each node is associated with a real-value number character. Mutation involves the production of a bypass mutant to a nearby node chosen as a Gaussian function (mean centered on the parental value, s.d = 0.1) of distance from the parent node. The position along the x-axis determines the real number encoded by a node. Figure 8 shows performance on the Sphere (Eq. 6), Rosenbrock valley (Eq. 7) and Quartic with noise (Eq. 8) functions, the equations for which are shown below…

$$f_1(p) = \sum_{i=1}^{n} p_i^2 \qquad -5.12 < p_i < 5.12 \qquad \ldots(6)$$

$$f_2(p) = \sum_{i=1}^{n-1} 100(p_{i+1} - p_i^2)^2 + (1 + p_i)^2 \qquad -2.048 \quad p_i < 2.048 \qquad \ldots(7)$$

$$f_3(p) = \sum_{i=1}^{n} i p_i^4 + gauss(0,1) \qquad -1.28 \quad p_i < 1.28 \qquad \ldots(8)$$

Note that as opposed to the knapsack problem the function value must be minimized rather than maximized and so line 7 of Figure 8 is modified to read *win = min(fitA, fitB).* The GA tends to converge faster than PE to the solution. Note that PE with these settings behaves very much like a stochastic hill-climber (SHC), i.e. there is relatively little diversity of paths, and large path overlap between paths, see Figure 11. The number of simultaneously maintained phenotypes is low.

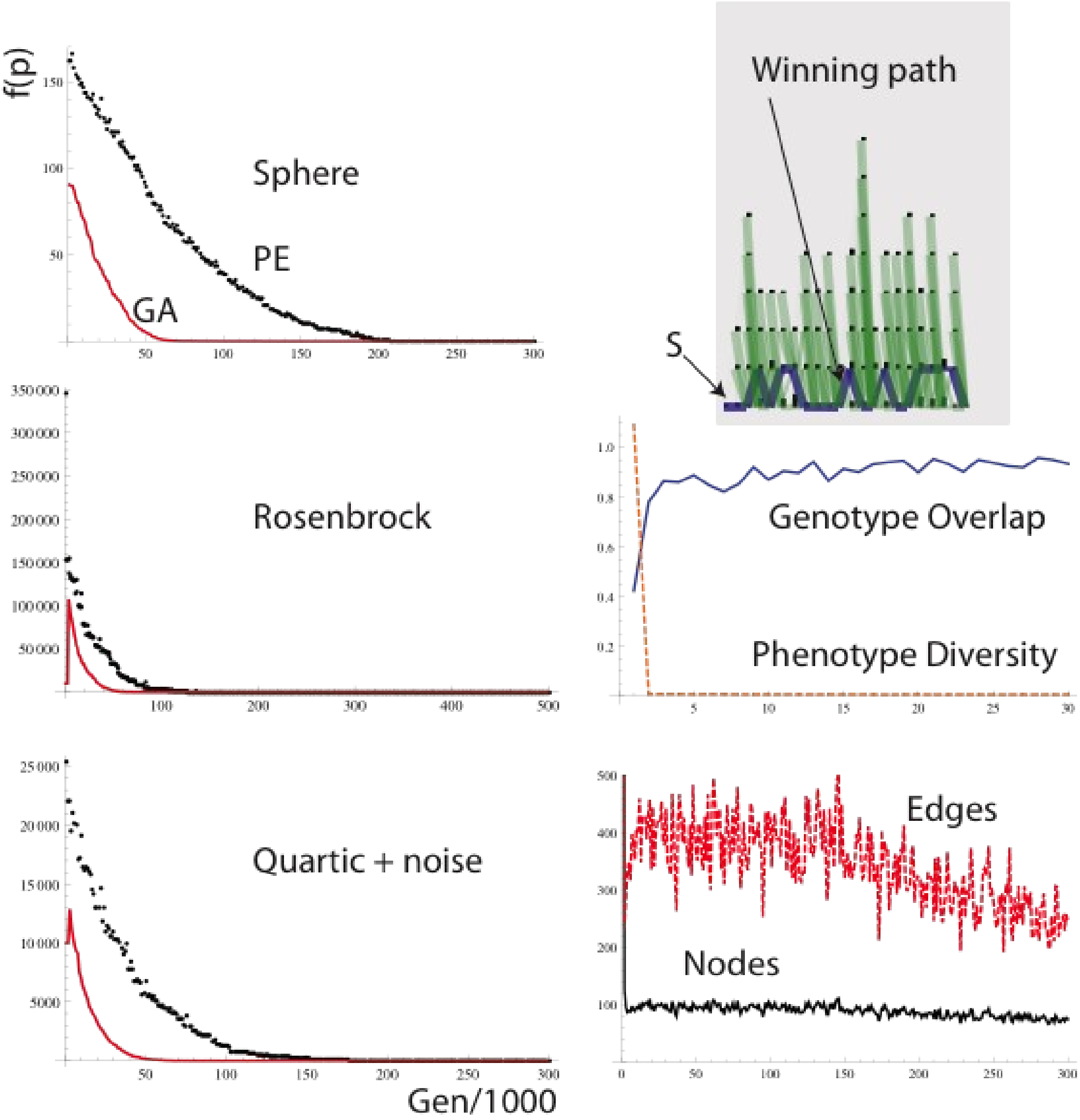


Figure 11. (Top) Performance on Sphere, Rosenbrock, and Quartic with noise functions. The following parameters were used: $N = 100, L = 20, \lambda = 0.1, \mu = 1/L, \chi = 0.01, \tau = 100, \omega = 0.01, \rho = 0, \gamma$ *(gamma)* $= 0$. The performance details on the right are for the Rosenbrock function.

Interestingly, we found that with the SHC like parameter settings for the PE, PE performed very poorly on Rastrigin's function, which is a cosine modulation of de Jong’s Sphere function used previously. It contains many local optima and is highly multimodal with regularly distributed minima locations, see Eq. 9 below…

$$f_4(p) = 10\ \ n + \overset{n}{\underset{i=1}{}} (p_i^2 - 10\ \cos(2pp_i)) \qquad -5.12 \quad p_i < 5.12 \qquad \ldots(9)$$

However, PE with a GA-like parameter setting (right), in which phenotype diversity of paths is preserved for a longer period in the run, performed about as well as the GA on the Rastrigin function. To make the PE behave more like a GA, and less like a SHC, we increase tau (the period after which a node dies if it is not activated), we apply a non-zero γ for exploration during a traversal so that even after a high fitness path has been found there is still a base level of exploration. Also, we implement a diversity maintaining change to the weight modification rule in which edges that are present in both paths are in fact punished in the same way as the losing path is punished in line 13, Figure 8.

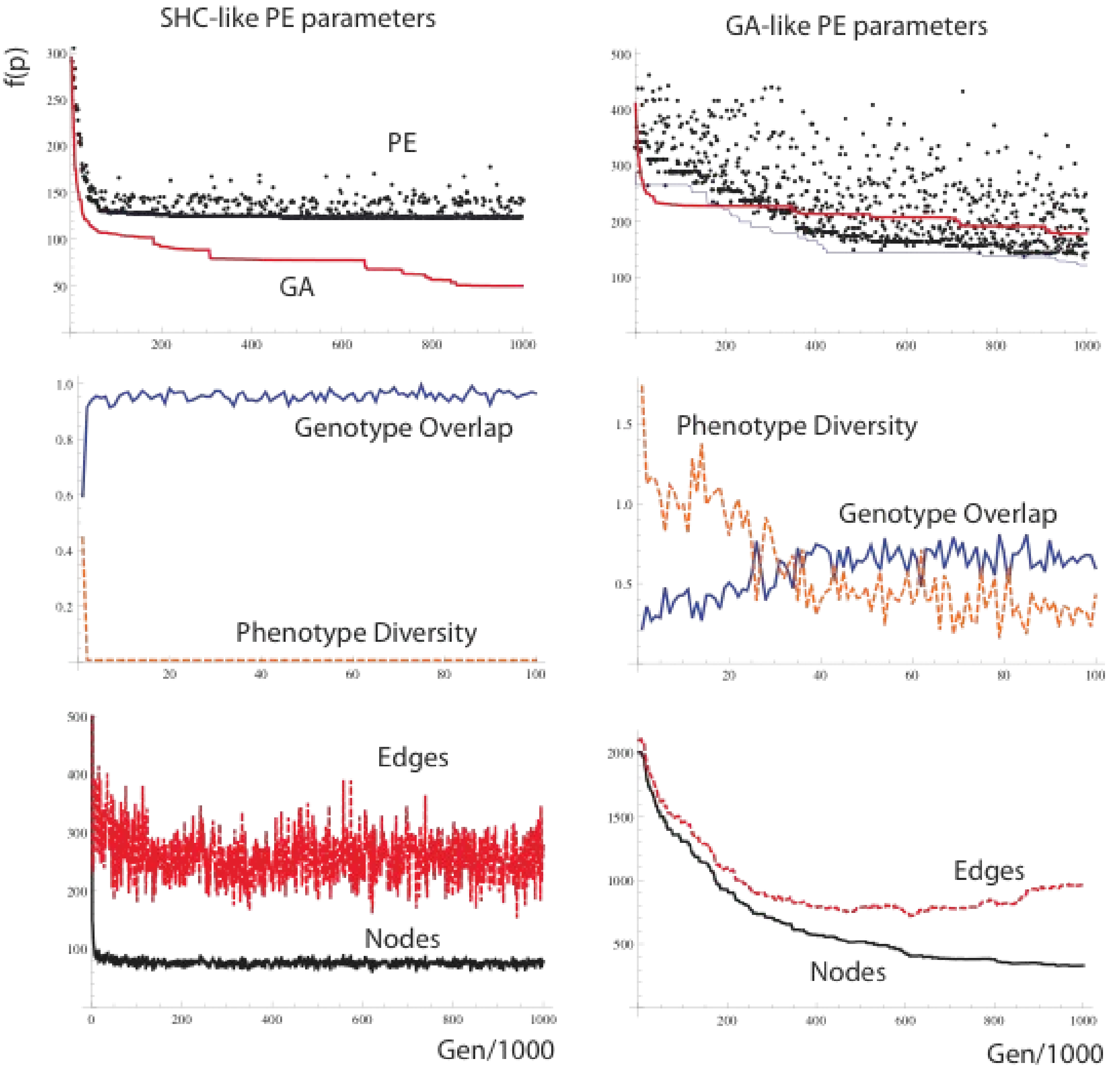


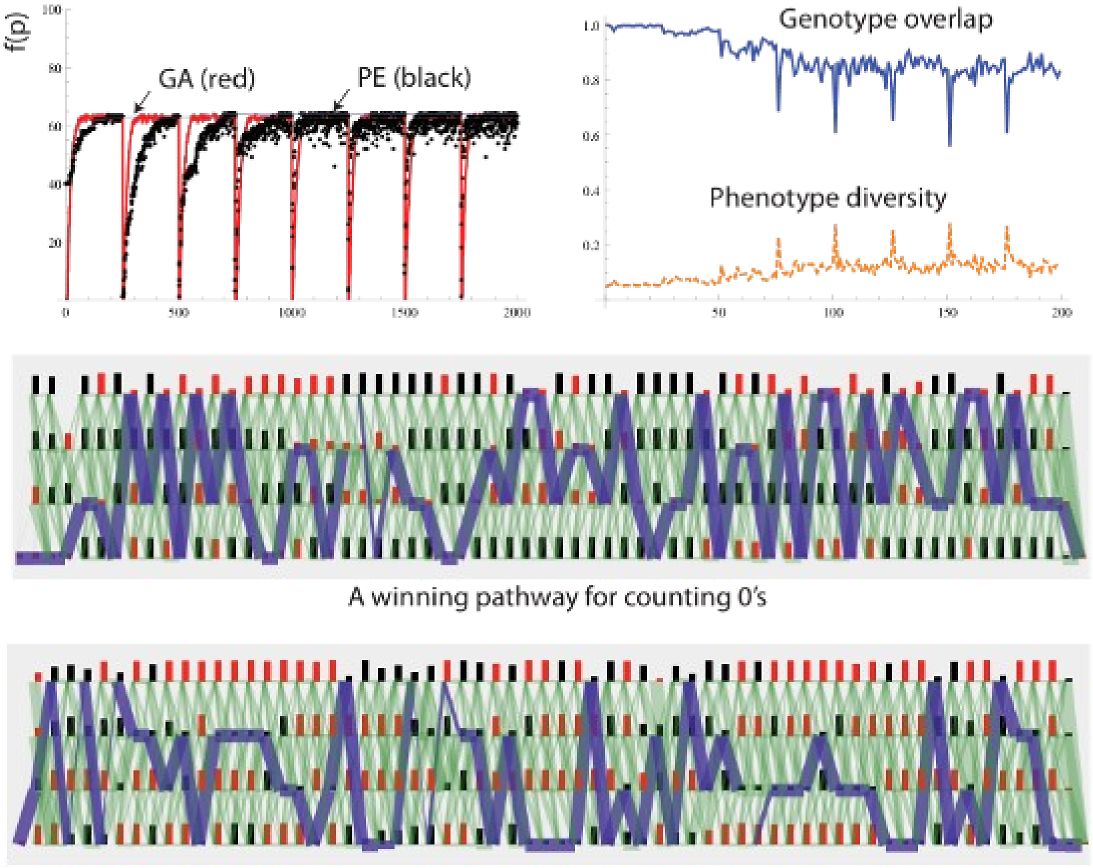
Figure 12. Performance of the PE algorithm on the Rastrigin function. **(Left)** PE with parameter settings as in Figure 6. **(Right)** PE with parameter settings as follows. *N = 100, L = 20, λ = 0.1, μ = 1/1000L, χ = 0.01, τ = 1000, ω = 0.01, ρ = 0, γ (gamma) = 0.01* + overlapping edges punished as losing path. Diversity maintenance is far greater with the GA like settings that preserve distinct phenotypic niches for a longer period of time.

The PE with GA like parameters is capable of performing similarly to a genetic algorithm on Restrigin's function, whereas a GA with the SHC like parameters sets easily stuck on a local optimum. With these GA type parameters, many more nodes and edges are maintained in the network at any one time than in the solution to the easier problems in Figure 11. Also, phenotypic diversity is slower to be lost, and genotype overlap is less throughout the run. Without punishing overlapping edges the PE did not reach the same level of performance as the GA.

Memory of Previous Solutions in Variable Environments

The extended evolutionary synthesis has begun to seriously study the evolution of evolvability (Kirchner & Gerhart, 1998; Pigliucci, 2008). A population undergoing natural selection can automatically learn from past environments to structure exploration distributions so as to have a higher probability of producing fit phenotypes in novel but related environments (Izquierdo & Fernando, 2008; Kashtan & Alon, 2005; Toussaint, 2003). This can occur if there is non-trivial neutrality, i.e. a many to one genotype to phenotype map in which genotypes can be discovered that produce phenotypic exploration distributions that suit the problem (Toussaint, 2003).

The PE algorithm can exhibit similar automatic structuring of exploration distributions in variable environments. We present the simple example of a fitness function that involves an alternating counting ones and counting zeros problem with a period of E = 1000 generations. Figure 13 shows that the PE algorithm is able to learn from previous environments to rediscover previous optima more quickly. The GA is a direct encoding with no capacity for non-trivial neutrality and so cannot do the same, and forgets the all 1's solution once it has worked out the all 0's solution. Therefore we demonstrate that the PE algorithm has the capacity for memory of previously discovered solutions.



A winning pathway for counting 0's

A winning pathway for counting 1's

Figure 13. The alternating counting 1s and counting 0s problem. The PE algorithm can retain memory of previously visited optima and rediscover these paths more rapidly the next time it is in the same selective environment. The GA did not improve over repeated presentations of selective scenarios. The parameters used were: $N = 100$, $L = 64$, $\lambda = 0.1$, $\mu = 1/100L$, $\chi = 0$ *(no crossover)*, $\tau = 1000$, $\omega = 0.01$, $\rho = 0$, γ *(gamma)* $= 0$, maximum number of nodes per layer = 4, no punishment of overlapping paths. Oscillation period = 25000 generations.

The HIFF Problem

Some problems have interdependency between variables, i.e., the fitness contribution of one variable is contingent upon the state of other variables, and there are structured dependencies that are potentially exploitable. The XOR problem considered in Figure 6 was such an example. An extension of this is the hierarchical IF-and-only-IF problem (HIFF) (Watson, Hornby, & Pollack, 1998) described by the following equations…

$$g(s_1,.....,s_n) = \begin{matrix} 1, & & \text{if n} = 1 \\ nf(S_1,.....,S_k) + & {}_{i=1}^{k} g(S^i), & \text{otherwise} \end{matrix} \qquad (5)$$

Where s_i is the i^{th} variable of the configuration, S^i is the i^{th} disjoint subpartition of the variables, $f(p_1,\ldots,p_k) = 1$ if $(\$(s \quad S)" \, i : p_i = s)$, and 0 otherwise; where Σ is the discrete set of allowable values for the problem variables; and $n = k^H$, where $H \quad Z^+$ is the number of hierarchical levels in the system or subsystem, and k is the number of submodules per module. In HIFF we consider only binary variables, i.e., $Z \quad \{0,1\}$ and where $k = 2$.

The lowest level of fitness contributions comes from examining adjacent loci in the phenotype and applying the transfer function and the fitness function. The transfer function is [0,0] -> 0, [1,1] -> 1, and all other pair types produce a NULL (N). The fitness function for each level just sums the 0 and 1 entries at that level. The second level is produced by applying the same transfer function to the output of the first transfer function. The fitness contribution of this next layer is again the number of 0s and 1s in this layer multiplied by 2. This goes on until there is only one highest-level fitness contribution. The fitness landscape arising from the HIFF problem is pathological for a hill-climber since there is a fractal landscape of local-optima, which means that the problem requires exponential time to solve. The global optima are either all 1’s and all 0’s.

Figure 14 shows performance of the PE algorithm on the HIFF compared to a microbial GA without crossover. PE performs consistently better than the GA. In all cases PE found the optimum by within 200000 generations, but the GA never found the optimum within this time. We propose that the good performance of the PE on the HIFF problem is because of its capacity to learn to achieve suitable linkage disequilibrium between nearby alleles.

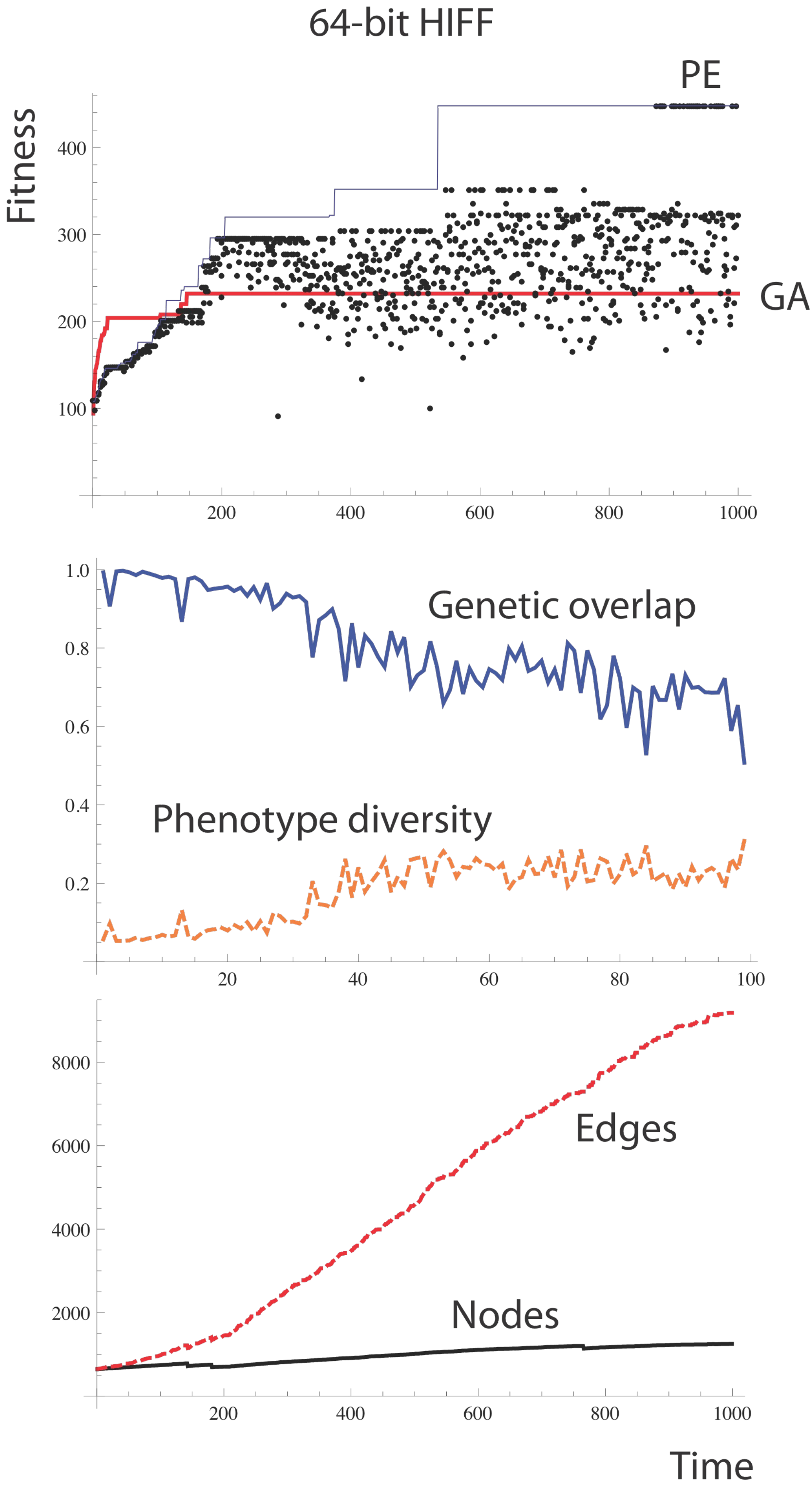

64-bit HIFF
PE
Fitness
400
300
200
100
GA
200
400
600
800
1000
1.0
0.8
0.6
0.4
0.2
Genetic overlap
Phenotype diversity
20
40
60
80
100
8000
6000
4000
2000
Edges
Nodes
200
400
600
800
1000
Time

Figure 14. The 64-bit HIFF Problem. The PE algorithm found the optimal solution but the microbial GA without crossover is stuck far from the optimum. The parameters used were: *$N = 10$, $L = 64$, $\lambda = 0.1$, $\mu = 1/100L$, $\chi = 0$ (no crossover), $\tau = 10000$, $\omega = 0.01$, $\rho = 0$, γ (gamma) $= 0$,* maximum number of nodes per layer = 20, no punishment of overlapping paths.

Expansion and Contraction Dynamics during Search and Discovery

Figure 15 shows the performance of PE on the royal road function. The simple royal road function is shown below…

$$\begin{aligned}
s_1 &= 11111111**;\\
s_2 &= ********11111111**;\\
s_3 &= ****************11111111**;\\
s_4 &= ************************11111111********************************;\\
s_5 &= ********************************11111111************************;\\
s_6 &= **11111111****************;\\
s_7 &= **11111111********;\\
s_8 &= **11111111;
\end{aligned}$$

from Figure 1 of Mitchell and Forrest[4]. For a bit string of length 64. 8 fitness points are obtained for each of the schemata s_i that is matched by the bit string. * indicates don't care. The royal road is a royal step pyramid because it does not matter in which order the schema are accumulated. When PE is applied to the royal road, one can immediately notice that during the exploration phase there is an increase in the number of nodes corresponding to loci of the 8-bit schemata that have not yet been found. Once a schema is found, the path corresponding to that schema gains dominance, and alternative paths are lost by node deletion gradually over time.

[4] http://web.cecs.pdx.edu/~mm/handbook-of-ec-rr.pdf

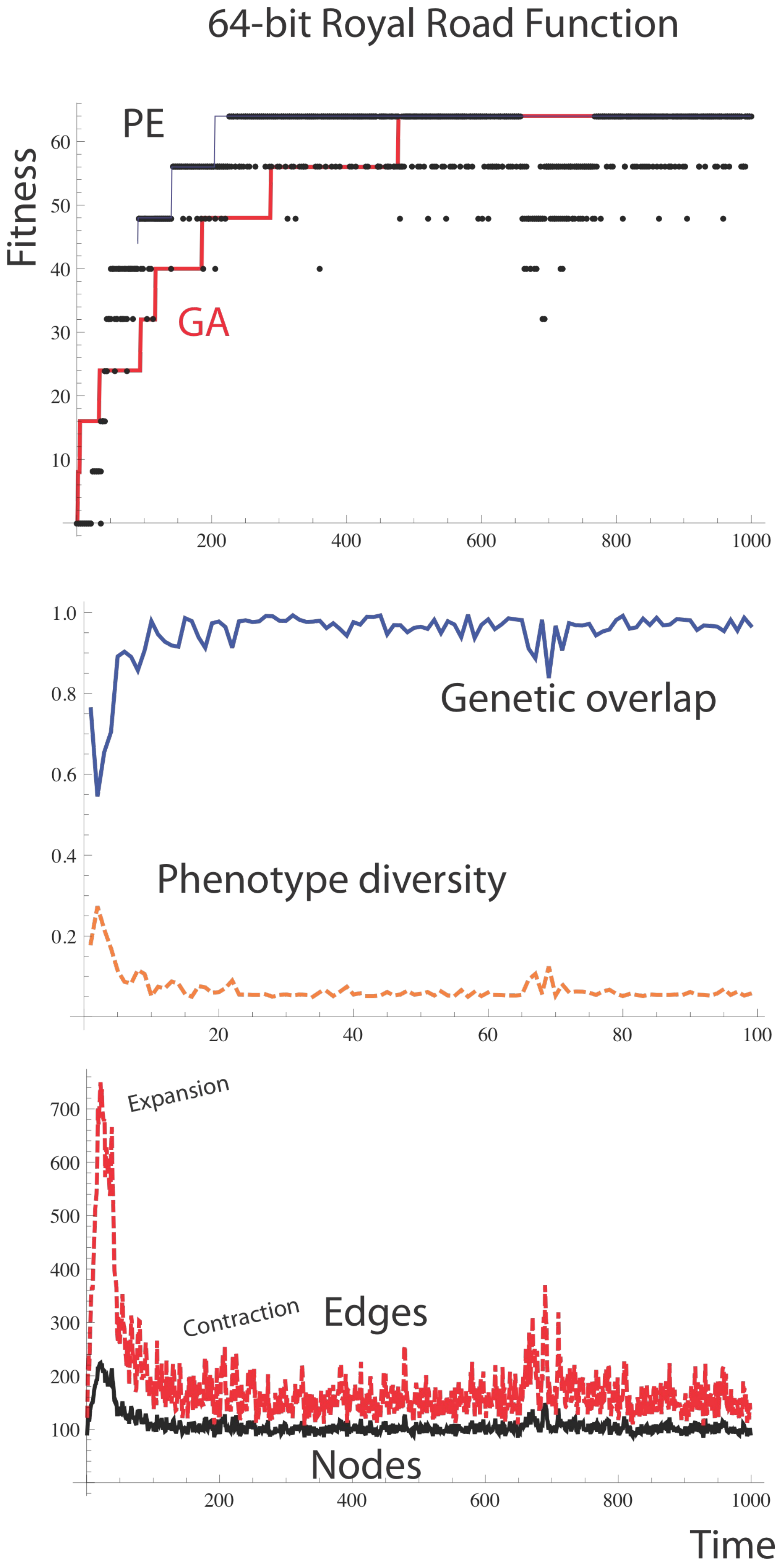

64-bit Royal Road Function
Fitness
PE
GA
10
20
30
40
50
60
200
400
600
800
1000
1.0
0.8
0.6
0.4
0.2
Genetic overlap
Phenotype diversity
20
40
60
80
100
700
600
500
400
300
200
100
Expansion
Contraction
Edges
Nodes
200
400
600
800
1000
Time

Figure 15. PE on the 64-bit royal road function shows automatic size changes. During exploration there is an expansion in the number of nodes and edges, followed by contraction after the solution is found.

Discussion

Can a path of activity be legitimately considered to be a unit of evolution? Maynard Smith said that group selection requires the existence of cohesive, spatially discrete groups, that ‘‘reproduce’’ by sending out propagules, and that can go extinct (1976, p. 282). He defined a population of units of evolution as “any population of entities with the properties of multiplication (one entity can give rise to many), variation (entities are not all alike, and some kinds are more likely to survive and multiply than others), and heredity (like begets like) will evolve. A major problem for current evolutionary theory is to identify the relevant entities’’ (p. 222, (Maynard-Smith, 1988)). We have identified a path as a unit of evolution. A path is capable of multiplicative growth in the population of paths, however, it does not give rise to a distinct spatially separate entity during growth, but strengthens the probability of traversal of its edges. We have demonstrated that path characters can have variation and heredity.

Implementations of the PE Algorithm

The neuronal networks of the brain provide the most natural implementation of the PE algorithm and its many possible variants[5]. The fundamental operation of node mutation and pathway crossover that the PE algorithm depends upon, closely relate to

[5] There are a great variety of possible rules for determining when a path should be strengthened and how a traversal should be made, just as there are a great variety of possible genetic algorithms. The pair-wise tournament selection algorithm is not well suited to control of on-line behaviour in which a network of pathways, e.g. implemented in neuronal networks, evolves subject to reward obtained in real time. So far the stability of a path has been related to the reward that is obtained by taking that path. However, it has been necessary to assess two paths from the same starting condition and compare the reward obtained between them. This is unsuitable for online behaviour control. The key to linking the PE algorithm to online control lies in the domain of learning classifier systems where it was discovered that a more effective search could be carried out if classifiers (which here can be interpreted as paths where a node is a condition and an edge is an action) were stabilized on the basis of the accuracy of their prediction of the reward (or sensory state) obtained in the next time step, whilst selection of edges (actions) was done using an e-greedy strategy based on the predicted reward associated with an edge (action) (Butz, 2006; Wilson, 1995). This would preserve pathways that accurately predict low reward as well as those that accurately predict high reward. Our critical insight here is that an edge is described not by one but two variables, a prediction accuracy that determines its growth rate, and a predicted reward that determines the traversal of paths.

the synaptic pathway mutations first proposed by Adams (Adams, 1998). Furthermore, the recent discovery of rapid structural plasticity (in the order of seconds) in the brain provides indirect support that cognitive and behavioral adaptations could be produced by evolution of neuronal pathways (Butz, Worgotter, & van Ooyen, 2009; Holtmaat & Sovoboda, 2009; Lohmann & Bonhoeffer, 2008). In real neuronal networks it is possible that mutations will be able to shortcut several layers, or add layers, producing variable path lengths. A reward biased spike-time dependent plasticity (STDP) rule with eligibility traces is a very natural biological implementation of the multiplicative weight update we use in the PEA (Izhikevich, 2007). The PE perspective invites us to think of the analogue of higher-order units of selection in pathway-based units of evolution. Polychronous groups are not linear pathways but trees of activity that may also evolve by the mechanisms described here (Izhikevich, 2006).

The fact that PE allows natural selection with unlimited heredity in the absence of template replication permits the possibility that in the origin of life, some yet unknown means of implementing PE may have been a precursor to template replication. Any system in which reward can be associated with an eligibility trace to strengthen a pathway is capable of PE, e.g. phosphorylation or other chemical modifications may mark metabolic pathways for strengthening on the basis of reward.

The pathway evolution viewpoint proposed for the first time here allows us to take a Darwinian population perspective on adaptive network processes and may inform experimentation, for example, in the neurosciences, what is the probability of fixation of a novel pathway or edge (synapse) in a real neuronal network as a function of its relative reward? Viewing a pathway as a unit of evolution that is capable of multiplication without template replication helps us understand how adaptation could occur by a true Darwinian process yet where *we see no obvious spatially distinct units.* This is because networks store an effective population of intertwined pathways. This proposal extends the neuronal replicator hypothesis (NRH) that argues that there exist informational replicators in the brain, i.e. autocatalytic entities capable of producing offspring that are correlated with their parent in fitness, and hence capable of accumulation of adaptations by natural selection. Previous formulations of the NRH focused on possible mechanisms of template replication in the brain (C.

Fernando, Goldstein, & Szathmáry, 2010; C. Fernando, Karishma, & Szathmáry, 2008; C. Fernando & E. Szathmáry, 2009; C. Fernando & E. Szathmáry, 2009). This is the first demonstration that the NRH can be satisfied in the absence of template replication if paths are considered to be units of evolution.

Acknowledgements: The work was funded by an EU FP7 grant E-FLUX. Many thanks to Richard Goldstein, Martin Butz, Simon McGregor, Istvan Zacher, Eugine Izhikevich, Inman Harvey, David Waxman, and Dario Floreano for useful discussions.

Adams, P. (1998). Hebb and Darwin. *J Theor Biol, 195*(4), 419-438.

Butz, M. (2006). *Rule-Based Evolutionary Online Learning Systems: A Principled Approach to LCS Analysis and Design* (Vol. 191). Heidelberg: Springer.

Butz, M., Worgotter, F., & van Ooyen, A. (2009). Activity-dependent structural plasticity. *Brain Research Reviews, 60*(2), 287-305.

Chu, P. C., & Beasley, J. E. (1998). A Genetic Algorithm for the Multidimensional Knapsack Problem. *Journal of Heuristics, 4*(1), 63-86.

Fernando, C., Goldstein, R., & Szathmáry, E. (2010). The Neuronal Replicator Hypothesis. *Neural Computation, 22*(11), 2809–2857.

Fernando, C., Karishma, K. K., & Szathmáry, E. (2008). Copying and Evolution of Neuronal Topology. *PLoS ONE, 3*(11), e3775.

Fernando, C., & Szathmáry, E. (2009). Chemical, neuronal and linguistic replicators. In M. Pigliucci & G. Müller (Eds.), *Towards an Extended Evolutionary Synthesis.* . Cambridge, Ma.: MIT Press.

Fernando, C., & Szathmáry, E. (2009). Natural selection in the brain. In B. Glatzeder, V. Goel & A. von Müller (Eds.), *Toward a Theory of Thinking*. Berlin.: Springer.

Harvey, I. (2009). *The Microbial Genetic Algorithm.* Paper presented at the Proceedings of the Tenth European Conference on Artificial Life, Budapest, Hungary.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press.

Holtmaat, A., & Sovoboda, K. (2009). Experience-dependent structural plasticity in the mammalian brain. *Nature Reviews Neurosicence, 10,* 647-658.

Izhikevich, E. M. (2006). Polychronization: computation with spikes. *Neural Computation, 18*(2), 245-282.

Izhikevich, E. M. (2007). Solving the Distal Reward Problem through Linkage of STDP and Dopamine Signaling. *Cerebral Cortex, 17*, 2443-2452.

Izquierdo, E., & Fernando, C. (2008). *The evolution of evolvability in gene transcription networks.* . Paper presented at the Proceedings of the 11th International Conference on Artificial Life.

Kashtan, N., & Alon, U. (2005). Spontaneous evolution of modularity and network motifs. *Proc Natl Acad Sci USA, 102*(39), 13773--13778.

Khuri, S., Back, T., & Heitkotter, J. (1994). *The zero/one multiple knapsack problem and genetic algorithms*. Paper presented at the Proc. 1994 ACM Symp. Applied Computing, .

Kirchner, M., & Gerhart, J. (1998). Evolvability. *Proc. Natl. Acad. Sci. USA, 95*, 8420-8427.

Lohmann, C., & Bonhoeffer, T. (2008). A Role for Local Calcium Signaling in Rapid Synaptic Partner Selection by Dendritic Filopodia. *Neuron, 59,* 253-260.

Maynard Smith, J. (1986). *The problems of biology*. Oxford, UK
: Oxford University Press.

Maynard-Smith, J. (1988). Evolutionary progress and the levels of selection. In M. H. Nitecki (Ed.), *Evolutionary Progress* (pp. 219–230). Chicago: University of Chicago Press.

Okasha, S. (2006). *Evolution and the levels of selection*. Oxford: Oxford University Press.

Pigliucci, M. (2008). Is evolvability evolvable? . *Nature Reviews Genetics, 9,* 75-82.

Price, G. R. (1970). Selection and covariance. *Nature 227,* 520-521.

Szathmáry, E. (2006). The origin of replicators and reproducers. *Philos. Trans. R. Soc. London. B. Biol. Sci. , 361*(1474), 1761-1776.

Toussaint, M. (2003). *The evolution of genetic representations and modular adaptation.* ND 04, 44780 Bochum---Germany.

Vasas, V., Szathmáry, E., & Santos, M. (2010). Lack of evolvability in self-sustaining autocatalytic networks constraints metabolism-first scenarios for the origin of life. . *Proc Natl Acad Sci U S A.*

Watson, R. A., Hornby, G. S., & Pollack, J. B. (1998). *Modelling Building-Block Interdependency.* Paper presented at the PPSN-V.

Wilson, S. W. (1995). Classifier Fitness Based on Accuracy. *Evolutionary Computation, 3*(2), 149-175.