4 Evolutionary Pathways

Inman Harvey and Ezequiel A. Di Paolo

4.1 Introduction

Heredity, in both natural and artificial evolution, is often pictured in the form of a tree. This may be at the level of individuals, where the lines trace pathways back through parents and grandparents to distant ancestors. It could be at a higher level, where the ancestral relationships traced are those of the species, genera, and orders. This simple and compelling picture is made somewhat muddier by a belated recognition that there may be interesting amounts of horizontal gene transmission between many of these branches. Nevertheless, the concept of the evolutionary pathway may be useful for many purposes. For people using artificial evolution, as in evolutionary robotics (ER), one question that may be asked is this: given the present situation of some evolving population, and given the desired end goal, what pathways are there that evolution may take and is there any way that we can persuade evolution to head along the faster ones?

That question, in its very general form, is what we are tackling in this chapter. In doing so we shall move back and forth between rather abstract discussion of evolutionary search spaces and practical tips and hints that may be useful for people designing and applying evolutionary algorithms for some specific problem. Sometimes such tips have been found useful by practitioners and are passed on by word of mouth, yet do not find their way into the published literature. As heuristics, they are sometimes crude and unproven. This chapter is not intended as an exhaustive classification of evolutionary pathways; it should rather be considered as an album of postcard views drawn from several ER travelogues, in the hope that some of them may inspire and influence future explorers.

We discuss the various features of evolutionary paths that can make them more or less easy for evolution to navigate. These are illustrated by some case studies, and where the lessons or intuitions thus suggested may have some wider applicability they are summarized in the form of “Travel Tips.”

4.2 Where to Start From

The apocryphal Irish peasant, accosted in a rural lane by an American tourist in a rental car asking for the best route to Dublin airport, scratched his head and replied: "If I was going there, I wouldn't be starting from here." There are often good choices and bad choices for a starting population. Initializing the population at random sounds as though it should be simple, but there are significant subtleties. The Bayesian rationale behind initializing a population at random is that if we have no prior knowledge of where the solution or solutions may be found, then we should not bias the search in any way—but where we do have some prior knowledge we may be justified in incorporating it.

In evolutionary robotics it is common for all or part of the genotypes to be specifying real values for things like weights and biases of a neural network, time parameters, sizes of body parts, and so on. When any of these values are potentially unbounded in one or both directions, the search space is infinite in size. We must scatter the initial population over some bounded region that includes where the goal will be. A careless guess for the size of these bounds could be orders of magnitude too large or too small. The programmer may be hoping for evolution to do most of the work, but a reality check—needing some ability to reason with orders of magnitude—could help stop the search in the right ballpark. In general, if some of the parameters tend to evolve toward one of the extremes of their range, this may be a sign that the parameter range should be changed accordingly. Incidentally, when a range is indeed given hard boundaries then a decision has to be taken as to how to treat mutations that mutate values "out of range"; one solution often found effective is to "reflect" such a mutation back off the end-of-range value as if it was a reflecting wall.

In the absence of better knowledge, parameter ranges should be searched uniformly without bias. This is often done using linear mapping of genotype values onto the parameter range. However, the meaning of an unbiased search can depend on the parameter. When evolving CTRNNs (Beer 1995a, 1995b), time parameters τ are significant, and should often be allowed to range over several orders of magnitude, for instance from milliseconds to hours as the performance of the task may involve a variety of timescales (up to and including the timescale of the whole performance, which can include several trials when, for instance, evolving learning behavior). One trick to facilitate this is to encode time parameters logarithmically on the genotype. A value t on the genotype gets translated to τ via a formula such as $\tau = e^t$, or $\tau = 10^t$. This allows t to range through positive and negative values, while τ is always positive. It means that an initial population, drawn from a flat distribution in t values, will be realistically scattered across several orders of magnitude in τ values, and subsequent mutations also behave more appropriately.

4.2.1 CTRNNs or CTSNs

The choice of initial bounds, and the use of logarithmic encoding where appropriate, are examples where we do have some prior knowledge and are justified in incorporating this. A further example of this is the use of center-crossing (Beer 1995b; Mathayomchan and Beer 2002), where an understanding of the dynamics of CTRNNs is used to inform the choice of an initial population. The mathematical formalism of a CTRNN covers dynamical systems in general, and the inclusion of NN for neural networks in the name has misled some people who have been puzzled that the "weights" involved are fixed. The description CTSN, for continuous time sigmoidal networks (Beer and Daniels 2010), may have less misleading assumptions attached to it. Whichever term is used, the "nodes" of such networks refer to any modeled variable of a dynamical system—and could, for instance, include the weights of a conventional NN. The benefits of using CTRNNs or CTSNs as a default architecture for an evolved robot "brain" include:

1. They provide universal approximators for any smooth dynamical system (Funahashi and Nakamura 1993).
2. They are well understood, with a lot of accumulated experience and some analytical results.
3. They are easy to implement.
4. They are relatively easy to analyze.
5. They give good building blocks for dynamical systems, e.g., their activations are squashed so that values do not shoot off to infinity.

Each neuron in a CTRNN has a nonlinear activation function, with a central region (defined by the value of the bias term) that specifies where its output is most sensitive to varying input values. For summed input values significantly higher or lower than this, the nonlinear activation function saturates close to its upper or lower bounds, and ceases to be sensitive to further variations. In effect it is then switched (almost) fully on or fully off and plays (almost) no further role in the dynamics of the network as a whole. Small variations in connection weights associated with such a saturated neuron would make negligible changes to its function, and hence it becomes difficult for evolution to "explore viable pathways." Beer (1995b) conjectured that if an initial population was manipulated so as to ensure that all the neurons started off in their sensitive regions, then this would maximize the prospects for evolution. This can be done by ensuring that there is an equilibrium point for the network with every neuron's activation at its exact center of symmetry, in other words, with an output of 0.5 halfway between the saturation levels of 0.0 and 1.0.

A simple way to achieve this in an initial population that is otherwise random is to initialize the bias term for each neuron, as a function of the connection weights, so as to obey the “center-crossing” condition (Beer 1995b). In other words, although the connection weights are initialized randomly, the biases are initially constrained to meet this condition. Thereafter evolution is free to change (through mutation) all the parameters including these biases. “The richest dynamics should be found in the neighborhood of the center-crossing networks in parameter space, and one would expect that an evolutionary algorithm would benefit from focusing its search there” (Beer 1985b). In addition, if it were convenient for some neurons to work in the saturated regions, evolution could easily find suitable parameters to achieve this from such a starting point, while the opposite move from saturated to more dynamical regions is trickier.

In (Mathayomchan and Beer 2002) this strategy is compared against pure random initialization in evolving central pattern generators for a simulated walking creature, and found to give significantly superior performance. Analysis of CTRNNs (Beer and Daniels 2010) also shows that under some circumstances new regimes of dynamics become available when self-weights are allowed to exceed a value of 4 allowing neurons to become bistable, hence restrictions of weights to smaller values than this can limit the richness of possibilities. Randall Beer provides a primer (chapter 3, this volume) with pointers to such studies. It pays to pay attention to where an evolutionary pathway starts.

Travel Tip (TT) 1: Choose limits on initial values in a random population so as to span the full range in which you expect to find good results.
TT2: Consider “reflecting” mutations that take a value out of bounds as if the bounds were a reflecting wall.
TT3: Where an evolved value may scale several orders of magnitude, consider scaling it logarithmically on the genotype.
TT4: Where possible, ensure the initial population is free to evolve in any direction; for example, for CTRNNs, use the center-crossing method.

4.3 Where We Are Heading

In biology, we are often interested in pathways leading to the here and now of some particular individual or species. In evolutionary robotics we are typically looking to the future, and must decide what would count as a satisfactory end goal. This goal may be more or less explicit; it may demand solutions to specific tasks or the satisfaction of broad viability constraints (explore the world, survive, and so on). If it is some uniquely specified robot behavior, there are probably very many different genetic specifications, different genotypes, that could generate this. Usually we are seeking some satisfactory class of behaviors, rather than something unique, so the target is wider still. In terms of evolutionary search, we are seeking the uplands of some fitness landscape rather than one unique point at the peak of Mount Everest.

Failure to realize this may underlie the common, but false, intuition that assumes big search spaces must be worse than small search spaces. After all, surely searching for a single needle in a large haystack is worse than searching for a single needle in a small haystack? But actually, if the density of needles is the same in each case, that intuition is misguided. Sometimes it may be the big haystack that is more navigable than the small one. Choices of genetic encoding will translate through to the size of the evolutionary search space. Any attempts to make the search space small, while ignoring consideration of the density of acceptable solutions or the navigability of the fitness landscape, are likely to be counterproductive.

If a fitness landscape only has two levels of fitness—in haystack terms, needles equals good and hay equals bad—then no search can be better than random search. The expected time to success depends only on the ratio of needles to hay. For evolutionary search to have some advantages, fitness should be measured in shades of gray and not just black and white. If one relates these different shades of fitness to contour lines on the fitness landscape, then any evolutionary pathway up the hills corresponds to passing through intermediate subgoals, which should be easier to find, en route to the more elusive uplands. Sometimes we can reshape the landscape by introducing intermediate targets, or stepping-stones, and we give some examples below. An intermediate target that is wider than the ultimate bull's-eye will of course be easier; and as archers know, surrounding the bull's-eye with concentric circles is one more way to add shades of gray to the fitness function.

TT5: Big search spaces may be as easy as, or sometimes even easier than, smaller search spaces—it is the density of satisfactory solutions that should be considered.
TT6: Fitness functions in shades of gray are usually easier than discrete black-and-white values.

4.4 Incremental Evolution

A classic strategy for tackling a difficult problem is to first carve it up into smaller ones: divide and conquer. When designing robots, such division is typically at the level of mechanism. Development of a motor mechanism can go ahead largely independently of any development of a vision system, since although each such module may be complex internally we expect the interactions between them to be rather simpler and easier to understand. But though humans find it easier to design complex systems in such a modular fashion, it is not so clear whether evolved systems, naturally or artificially evolved, have the same overwhelming need for modularity. It is significant that evolutionary robotics can and does produce evolved systems whose complexity of interactions is too difficult for humans to analyze and understand.

But there is a different way to carve up a large problem into smaller ones. A complex behavior can be seen as composed of simpler behaviors. Insofar as we can see, with hindsight, the evolutionary pathways from the origin of life to present-day mammals, we see phylogenetic development from relatively small and simple organisms with relatively simple behaviors through successive increases in complexity and variety of mechanisms and of behaviors. Necessarily every single individual in any one of these pathways must have been viable in its own right. This thought inspired Brooks (1991a, 1991b) to propose a subsumption architecture as a method for the incremental design of robots.

The way in which an evolutionary algorithm can be applied to such an incremental process and the use of such methods in ER were proposed in Harvey (1992), Harvey, Husbands, and Cliff (1993), and a body of work following from this. These resulted in the SAGA principles (Harvey 2001) of balancing mutation rates against selection pressure so as to optimize the exploitation/exploration balance in such scenarios. Incremental evolution is now quite widely practiced. Sometimes it is called robot shaping (Dorigo and Colombetti 1998), or incremental shaping (Auerbach and Bongard 2011). The term “scaffolding” is used in developmental psychology to refer to the framework that guides a child’s educational development, and can naturally extend to ER. It is clear that the choice of scaffolding methods can affect the ease or difficulty of evolutionary progress (Auerbach and Bongard 2011).

There are two separate issues to worry about when doing incremental evolution. First, one must assign the appropriate stepping-stones leading in the right direction and a suitable distance apart. This translates to designing a new fitness function for the next stage, to bias selection in favor of the desired behavior. An example can be seen in Vickerstaff and Di Paolo (2005), where an evolved agent was required to navigate via a series of beacons, with successive stages being also incremental in terms of complexity—evolving simple phototaxis first, then phototaxis to multiple sources, and finally homing to a nest.

The second issue is to decide whether new mechanisms are appropriate, for instance sensors, or extra brainpower in the form of a larger neural network, and if so, how to handle grafting such new material on to what is inherited from the previous stages.

Vaughan, Di Paolo, and Harvey (chapter 11, this volume) present a detailed case study on how incremental evolution was used to design a bipedal walking robot. An important lesson from this was that the effectiveness of results at one incremental stage, and the problems encountered there, informed the decision of what the next stepping-stone should be and what sorts of extra facilities might be needed to achieve that next step. So the evolutionary pathway was not prespecified from the beginning, but instead was laid out as evolution progressed in a dialogue between the analysis of the engineer and the high-dimensional parameter search of the genetic algorithm (GA).

Vaughan (2007) used a significantly useful trick in this and related work, which only received a passing mention in his thesis. At a new stage in the incremental evolution, if there was for example a new sensor with an associated neural network to be added to the previously evolved system, then the weights and biases of this new neural network could be set at random. But the trick was to make sure all the connection weights between this new neural network and the previously evolved one were set to zero. In consequence, the new sensor and its locally associated neural network had absolutely no effect on the previously evolved system. Nevertheless further evolution was allowed to modify thereafter the parameters for the old network, for the new network, and for their interconnections.

By doing it this way, the potential new functionality was added in an initially neutral manner. As with neutral networks in general (to be discussed later in this chapter), the extra dimensionality opened up possibilities for new pathways. If the engineer's analysis and intuitions in making available the new sensor and deciding where it should connect up to the existing network were not sound, if incorporation of these new components would actually lead to loss of fitness, then evolution "had the option" of keeping the interconnection weights at zero and the new part effectively was ignored. But if the intuitions were useful, then the new component part could coevolve together with the previous system and interconnections between them.

TT7: Consider intermediate targets, less specialized than the ultimate goal, as stepping-stones.

TT8: When adding new component parts to the robot or its "brain" at a new stepping-stone, consider having them initially connected to the preexisting system in a neutral way (e.g., with zero weights), so as to preserve the previous functionality.

4.5 Showing the Way

Setting out a pathway of stepping-stones, through incremental changes in the tasks on which evolving robots are evaluated, is one way that the engineer can influence the direction taken. But there is another way of implicitly encouraging evolution to head toward the next goal via some particular direction. In this next example, the engineers crafted their fitness function to make a crucial factor salient. This could be related to the biological concept of exaptation, where something that "will be useful in the future," such as feathers on a flying creature, may have been originated with some different role, such as providing warmth. In natural evolution the Blind Watchmaker has no foresight, so that when one feature is exapted to take on a new role, this is serendipitous. But in artificial evolution, where there is scope for foresight, the Watchmakers may be partially sighted.

In one example (Tuci, Quinn, and Harvey 2002) experimenters introduced an artificial bias in the evolutionary process that was stimulated by a lack of success in evolving a second-order behavior. The task for a (simulated) robot was to navigate in a 2D arena toward a distant target, placed on the floor at random at either the left or right end of the arena. The target was not visible until reached, and the only visible indication of which end to head for was a light signal. Over some sequences of trials the light was associated with the same end as the target, but over other sequences the light was at the opposite end. Hence a good strategy, as expressed by a human, might be: on the first trial of a sequence, follow the light, and, if the target turned out to be where the light was, use the target-light correlation to guide the following trials; but if the target was not associated with the light, turn back to find it, and on the following trials head for the end opposite the light. The problem for the ER method to tackle is to evolve, from scratch, neural networks that implement some such strategy, based solely on feedback derived from the time taken to find the target, averaged over many sequences of trials.

The initial fitness function was indeed based simply on that average; given that the experimental design implied that half the time the light was correlated with the target and half the time it was not, there was no easy first-order relationship between the direction of the light and the direction of what increased the fitness score. As a consequence, evolution fairly soon “discarded” the light sensors as irrelevant (by reducing their connection weights to zero) and focused on the best simple strategy. This was as if the robot was following this rule: “first, head to one end at random, and if the target was not there, then backtrack to the other end.” Though this was a reasonable first-order strategy, the possibility of improving on this, with a second-order strategy exploiting the light, was no longer available if the light sensors had in effect been disconnected. The “feathers” were not available to be exapted for “flying.”

So the solution found (Tuci, Quinn, and Harvey 2002) was to put in an artificial bias to make the light sensors salient. The decision was made to weight more heavily (by a factor of 3) the score for those trials where the light was next to the target, while retaining the original score when the light and target were anti-correlated. It is the average of these scores over many sequences of trials that contribute to selective fitness. In this way the light becomes of first-order significance for fitness, rather than merely of second-order significance as a learning cue. As a result of this bootstrapping, a majority of the subsequent evolutionary runs using this modified fitness function showed the desired results. First, the robot used the light sensors to go toward the light in search of the target. If the target was not there, the robot “learned from its mistake,” headed in the other direction, and in subsequent trials headed away from the light.

TT9: When you know that some environmental feature will be salient at a later, more evolved stage, exploit your foresight by ensuring it is salient early on.

4.6 Neutral Pathways

When one interprets evolution as finding pathways toward the uplands of the fitness landscape, it is tempting to assume that any pathway that fails to rise in fitness should be ignored. This need not be the case. In natural evolution, many mutational changes are neutral, yet still play a role in facilitating subsequent nonneutral changes. The same may well be true in artificial evolution.

One way of visualizing this is to picture a multistory building with numerous interconnecting corridors and staircases. In order to reach the higher stories, it may be necessary to traverse a level corridor to find an available staircase. Even though walking down the corridor does not immediately increase one's height above ground, and even though there is no gradient to the corridor indicating whether you are moving toward or away from the nearest staircase, it is nevertheless a good strategy to search along these neutral pathways.

Such neutrality is associated with redundancy in a mapping from genotype to phenotype. If different genotypes are associated with one and the same phenotype, or indeed with a class of phenotypes of identical fitness, then this is a many to one mapping. It implies that the genotype search space is larger than the phenotype search space, and sometimes this may be good news. This is particularly the case if and when such neutral pathways provide wormholes, or escape routes in hyperspace, allowing a population to escape from what otherwise might have been a local optimum.

One can create artificial abstract fitness landscapes to illustrate this. The NKp landscape proposed by Barnett (1998), an extension of Kauffman's NK landscapes (Kauffman and Weinberger 1989), is one such example. Barnett demonstrated that by varying the parameter p one could increase or decrease the neutrality of the fitness landscape, without needing to alter the ruggedness. A low-neutrality version, with low value of p, might be rugged enough to have lots of local optima, low hills that can trap a population trying to climb to the uplands. A high-neutrality version of the same landscape, with high value of p, can be just as rugged according to standard measures of ruggedness; yet the neutral corridors, or neutral networks, provide escape routes in this case so that the population no longer gets trapped in local optima. The additional neutral pathways will completely transform the evolutionary dynamics.

Barnett (2001) showed that on versions ("epsilon-correlated") of his abstract fitness landscapes a "netcrawler" can be provably optimal. This is a version of a $1 + 1$ genetic algorithm; in other words, the population is in effect of size two. He demonstrated that one can optimize the mutation rate for the best balance between exploration and exploitation. Increasing mutation rates increases the rate at which a population "searches blindly along corridors for possible staircases," but also increases the probability of "falling off" such a neutral network. It is desirable to balance these two factors against each other. It is shown (Barnett 2001) that under certain circumstances the provably optimal mutation rate for an “adaptive” netcrawler is such that a fraction $1/e \cong 37$ percent of mutations are neutral. Hence the mutation rate can be adapted on the fly; if currently more than 63 percent of mutated variants prove to be deleterious, the mutation rate should be decreased, and if less, increased.

Such neutral networks are not confined to artificial abstract fitness landscapes. There is good reason to expect that any high-dimensional evolutionary search space, whether biological or artificial, which has sufficient redundancy in a nonarbitrary genotype to phenotype mapping will have such neutral networks. Schuster and colleagues (Schuster et al. 1994) have pioneered the study of these in RNA fitness landscapes. Gavrilets (1997) has applied a (much more abstract and broader) version of these ideas at the species level to explain the dynamics of biological species on a “holey” fitness landscape. Moving closer to ER, the existence and exploitation of neutral networks in an engineering design problem has been demonstrated by Thompson (Thompson 1997; Harvey and Thompson 1997; Thompson and Layzell 2000).

These experiments were based on evolving the connectivity of a field programmable gate array (FPGA): a chip composed of cells with multiplexers (“muxes”) in a square array. For these experiments a 10 × 10 region of cells was used, and binary genotypes (of length 1800 or more) specified the connectivity between cells, the configurations of muxes in each cell, and other synchronizing features. The real, physical FPGA, as thus genetically specified, was tested on its capacity to discriminate between two kinds of input signals: square waveforms of 1 kHz or 10 kHz. The genetic search space is immense; the fitness landscape depends upon the physical properties of the FPGA, which, since it can operate in asynchronous (unclocked) mode, cannot be reliably simulated by a deterministic simulation. In other words, it is a complex fitness landscape whose properties are largely unknown and noisy, and in this respect shares some aspects with biological fitness landscapes (in contrast to synthetic and abstract deterministic fitness landscapes). However it is still possible to keep accurate records of the evolutionary pathway taken.

One example (Thompson and Layzell 2000) used a 1 + 1 Evolution Strategy: that is, the population was in effect reduced to size two with the current parent and its mutated copy, very similar to Barnett’s netcrawler discussed earlier. If the mutant’s fitness is greater or equal to its parent, it then becomes the new parent for the next stage. Hence a unique pathway can be recorded, as contrasted with the moving cloud observed when population sizes are bigger. It can be seen (Thompson 2002) that the path did indeed avoid getting trapped in local optima by “escaping” via neutral networks. Subject only to some noise in the (physical) evaluations, the evolutionary pathway was constrained not to go downhill, and there were extended stages without increase in fitness but with genetic modification—exploration of a neutral network. Analysis of the (minimal) population “walking” along neutral pathways shows that—despite fitness remaining unchanged during this level segment—the jump in fitness at the end of the segment was due to a specific mutation that would not have had that beneficial effect at the beginning of the segment. In other words, the neutral drift along that segment was not wasted; it provided a new genetic environment in which that mutation now had a positive effect.

TT10: If there is redundancy in the genotype-phenotype mapping, so as to open up the possibility of neutral networks, then you shouldn't necessarily consider a period of fitness stasis over many generations to be a waste of time—the population may be "looking for a staircase along the corridor."

TT11: Particularly if there is a lot of neutrality in the evolutionary search space, consider a minimal population such as a 1 + 1 GA. . .

TT12: . . . and then consider whether it is appropriate to adapt the mutation rate according to the $1/e \cong 37$ percent rule (Barnett 2001).

4.7 Are We Nearly There Yet?

The experimenter using ER can often sympathize with the young child on an interminable car journey. Supposing one has done one's best to give the initial population the most favorable starting conditions, one has chosen appropriate stepping-stones to make incremental evolution easier, and encouraged the exploration of potentially fruitful pathways, then how long should one wait before becoming impatient for results?

Under some circumstances it is possible to at least put a lower bound on this waiting time. Here we draw on ideas from Worden (1995) and Haldane (1957), but these are revised for the purposes of a GA using a binary genotype. The simplest way to understand this point is via the game of Twenty Questions.

Suppose that this game is played with an agreed search space containing exactly 1,048,576 ($=2^{20}$) objects. Player_1 secretly chooses the target object, and Player_2 can ask twenty questions to elicit Yes/No answers. The optimum strategy for Player_2 is to choose questions that in effect bisect the search space. If exactly one half of the previously agreed-upon set of objects are bigger than a shoe and one half of the objects are smaller than a shoe, then the answer to the initial question "Is it [the target object] bigger than a shoe?" immediately gives 1 bit of information. A less efficient question, such as "Is it bigger than bus?" is less informative; the challenge of the game for Player_2 is to choose the most appropriate question at each stage.

One can view this process as iterative selection in a population of size 2^{20}. At each stage, and each question, the objects in the less fit half are discarded, while we may assume that the objects in the fitter half are duplicated; hence the population size remains unchanged. This is in effect a simple asexual GA, with no mutation, but a vast population size that initially completely spans the search space, then starts to genetically converge, under selection driven by appropriate questions, toward the target. In GA terms, we may treat this as a case of binary genotypes of length 20—and clearly 20 generations (or rounds of selection) is the absolute minimum length of time required, under optimum questioning strategy. This is assuming that the selection process picks the top half—which is in the ball park of a typical selection pressure used in many GAs; for instance, tournament selection with tournaments of size 2, as used in the Microbial GA (Harvey 2001) does just this. Generalizing, for an n-bit genotype, under these circumstances, the minimal expected time to wait is n generations; this is what Worden (1995) calls the "Speed Limit of Evolution." As with the speed of light, this forms an upper bound only achievable under ideal circumstances.

Moving back from the Twenty Questions scenario to evolution, natural or artificial, one can view each round of selection as a very noisy, imperfect version of "the environment posing a binary question"—survive or die. In artificial evolution we may well start with a random initial population that does not span the whole search space, indeed is typically tiny compared to the search space. This, together with the role of mutations and the various inefficiencies combine to make the expected speed of evolution much slower than the theoretical speed limit. In artificial evolution, even under ideal conditions, one rule of thumb is to allow two orders of magnitude slower. Worden (1995) argues that recombination will not provide any added benefit, though his arguments are unclear and not accepted by many. Nevertheless, these back-of-the-envelope estimates are a starting point for estimating how long evolution might take.

So with an artificial evolution problem, with n-bit genotypes, the rule of thumb suggests one should wait for, say, 100n generations before getting impatient. This assumes the selection pressure mentioned earlier, and that one is waiting to find the unique best point in the search space. If redundancy or other reasons imply that any solution within the top-ranked 2^m of the search space would be acceptable, then the rule of thumb dictates $100(n - m)$ as a plausible waiting time.

This is presented here as a rule of thumb, which seems to bear up reasonably well in practice but is not rigorously proven. Extending beyond binary genotypes to those with real values, as might be the case when evolving parameters for neural networks, is controversial. But on the face of it, if a genotype contains n real values, and it is estimated that each such value needs to be defined to 4-bit, or 8-bit accuracy, then a starting place would be to compare this with binary genotypes of length 4n or 8n, for the purposes of relating this to a "speed limit."

TT13: When evolving binary genotypes of length n, under the circumstances previously indicated, consider waiting 100n generations before becoming impatient.
TT14: When evolving genotypes with real values, consider how many bits would give sufficient accuracy and use this to calculate the speed limit.

4.8 Conclusions

Evolutionary robotics can be pursued for scientific or for engineering motives (Harvey et al. 2005). Apart from such motives it also requires experimental and computational skills, and experience builds up a body of insights. Sometimes this results from theoretical analysis of what should work, but often it comes down to intuitions that have yet to be firmly grounded. ER practice, along with most other technical accomplishments, includes hints and practical knowledge that often does not make it into the textbooks.

This chapter has aimed to get some of these "Travel Tips" onto the record. This is not intended as an exhaustive list, and we do not claim to be covering the most important tips—it is a somewhat random set of postcard views from several ER journeys, and other travelers could have emphasized different views. Planning an ER expedition is challenging, but there is a sense of achievement in finding promising pathways toward one's goals.

References

Auerbach, J. E., and J. C. Bongard. 2011. Evolving monolith robot controllers through incremental shaping. In *New Horizons in Evolutionary Robotics, SCI 341*, ed. S. Docieux, N. Bredèche, J-B. Mouret, 55–65. Berlin, Heidelberg: Springer-Verlag.

Barnett, L. 1998. Ruggedness and neutrality—The NKp family of fitness landscapes. In *Alife VI, Proceedings of the 6th International Conference on Artificial Life*, ed. C. Adami, R. K. Belew, H. Kitano, and C. Taylor, 18–27. Cambridge, MA: MIT Press.

Barnett, L. 2001. Netcrawling—Optimal evolutionary search with neutral networks. In *Proceedings of the 2001 Congress on Evolutionary Computation (CEC2001)*, ed. J-H. Kim, B-T. Zhang, G. Fogel, and I. Kuscu, 30–37. Piscataway, NJ: IEEE Press.

Beer, R. D. 1995a. A dynamical systems perspective on agent-environment interaction. *Artificial Intelligence* 72:173–215.

Beer, R. D. 1995b. On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behavior* 3:469–509.

Beer, R. D., and B. Daniels. 2010. Saturation probabilities of continuous-time sigmoidal networks. ArXiv e-prints 1010.1714. http://arxiv.org/abs/1010.1714 (accessed June 21, 2013).

Brooks, R. A. 1991a. Intelligence without representation. *Artificial Intelligence* 47:139–159.

Brooks, R. A. 1991b. Intelligence without reason. In *Proceedings of the 12th International Conference on Artificial Intelligence (IJCAI-91)*, ed. J. Mylopoulos and R. Reiter, 569–595. San Mateo, CA: Morgan Kauffmann.

Dorigo, M. and M. Colombetti. 1997. *Robot Shaping: An Experiment in Behavior Engineering*. Cambridge, MA: MIT Press.

Funahashi, K., and Y. Nakamura. 1993. Approximation of dynamical systems by continuous time recurrent neural networks. *Neural Networks* 6 (6): 801–806.

Gavrilets, S. 1997. Evolution and speciation on holey adaptive landscapes. *Trends in Ecology & Evolution* 12:307–312.

Haldane, J. B. S. 1957. The cost of natural selection. *Journal of Genetics* 55:511–524.

Harvey, I. 1992. Species adaptation genetic algorithms: A basis for a continuing SAGA. In *Towards a Practice of Autonomous Systems: Proceedings of the 1st European Conference on Artificial Life*, ed. F. J. Varela and P. Bourgine, 346–354. Cambridge, MA: MIT Press.

Harvey, I. 2001. Artificial evolution: A continuing SAGA. In *Evolutionary Robotics: From Intelligent Robots to Artificial Life*, LNCS 2217, ed. T. Gomi, 94–109. Heidelberg: Springer-Verlag.

Harvey, I., and A. Thompson. 1997. Through the labyrinth evolution finds a way: A silicon ridge. In *Proceedings of the First International Conference on Evolvable Systems: From Biology to Hardware (ICES96)*, ed. T. Higuchi, M. Iwata, and L. Weixin, 406–422. Heidelberg: Springer-Verlag.

Harvey, I., E. Di Paolo, R. Wood, M. Quinn, and E. A. Tuci. 2005. Evolutionary robotics: A new scientific tool for studying cognition. *Artificial Life* 11 (1–2): 79–98.

Harvey, I., P. Husbands, and D. Cliff. 1993. Issues in evolutionary robotics. In *From Animals to Animats 2: Proceedings of the 2nd International Conference on Simulation of Adaptive Behavior (SAB92)*, ed. J.-A. Meyer, H. Roitblat, and S. Wilson, 364–373. Cambridge MA: MIT Press.

Kauffman, S. A., and E. D. Weinberger. 1989. The NK model of rugged fitness landscapes and its application to maturation of the immune response. *Journal of Theoretical Biology* 141 (2): 211–245.

Mathayomchan, B., and R. D. Beer. 2002. Center-crossing recurrent neural networks for the evolution of rhythmic behavior. *Neural Computation* 14:2043–2051.

Schuster, P., Fontana, W., Stadler, P. F., and I. L. Hofacker. 1994. From sequences to shapes and back—A case study in RNA secondary structures. *Proceedings of the Royal Society B* 255:279–284.

Thompson, A. 1997. An evolved circuit, intrinsic in silicon, entwined with physics. In *Proceedings of the First International Conference on Evolvable Systems: From Biology to Hardware (ICES96)*, ed. T. Higuchi, M. Iwata, and L. Weixin, 390–405. Berlin: Springer-Verlag.

Thompson, A. 2002. Notes on design through artificial evolution: Opportunities and algorithms. In *Adaptive Computing in Design and Manufacture*, ed. I. C. Parmee, 17–26. Berlin: Springer-Verlag.

Thompson, A., and P. Layzell. 2000. Evolution of robustness in an electronics design. In *Proceedings of the Third International Conference on Evolvable Systems: From Biology to Hardware (ICES2000)*, LNCS 1801, ed. J. Miller, A. Thompson, P. Thomson, and T. Fogarty, 218–228. Heidelberg: Springer-Verlag.

Tuci, E., M. Quinn, and I. Harvey. 2002. An evolutionary ecological approach to the study of learning behaviour using a robot-based model. *Adaptive Behavior* 10 (3/4): 201–221.

Vaughan, E. 2007. The evolution of the omni-directional bipedal robot. Doctoral thesis, University of Sussex.

Vickerstaff, R. J., and E. A. Di Paolo. 2005. Evolving neural models of path integration. *Journal of Experimental Biology* 208:3349–3366.

Worden, R. 1995. A speed limit for evolution. *Journal of Theoretical Biology* 176:137–152.