Selecting for Evolvable Representations

Joseph Reisinger
joeraii@cs.utexas.edu

Risto Miikkulainen
risto@cs.utexas.edu

Dept. of Computer Sciences
University of Texas at Austin
1 University Station, C0500
Austin TX, 78712-0233

ABSTRACT

Evolutionary algorithms tend to produce solutions that are not evolvable: Although current fitness may be high, further search is impeded as the effects of mutation and crossover become increasingly detrimental. In nature, in addition to having high fitness, organisms have evolvable genomes: phenotypic variation resulting from random mutation is structured and robust. Evolvability is important because it allows the population to produce meaningful variation, leading to efficient search. However, because evolvability does not improve immediate fitness, it must be selected for indirectly. One way to establish such a selection pressure is to change the fitness function systematically. Under such conditions, evolvability emerges only if the representation allows manipulating how genotypic variation maps onto phenotypic variation and if such manipulations lead to *detectable* changes in fitness. This research forms a framework for understanding how fitness function and representation interact to produce evolvability. Ultimately evolvable encodings may lead to evolutionary algorithms that exhibit the structured complexity and robustness found in nature.

Categories and Subject Descriptors

G.1.6 [**Optimization**]: Global Optimization; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

General Terms

experimentation, measurement, performance

Keywords

genetic algorithms, evolvability, modularity, representations, indirect encodings, development, estimation-of-distribution

1. INTRODUCTION

Evolvability is defined as "the capacity to generate heritable, selectable phenotypic variation"[10]. In terms of evolutionary search, evolvability means that the algorithm produces novel, high-fitness solutions with a high frequency. For example, when designing a coffee table, a representation

with natural parameters for height and surface area would be more evolvable than a direct encoding controlling the individual position of each block of wood; making parameter changes yields different kinds of tables, constraining variation in a meaningful way. This effect can also be seen in genetic programming, where parse-tree representations are more evolvable than searching code strings directly.

The main benefits of evolvability are to reduce the frequency of catastrophic mutations and to reduce the number of mutations required to produce phenotypically novel traits [10]. Thus evolvability provides a mechanism through which structured variation and complex adaptations can occur [9, 15, 22]. It is necessary to consider evolvability in situations where currently evolutionary search is inefficient and unable to produce reasonable solutions. Focusing on evolvability will yield more powerful genetic algorithms (GAs) that learn not only the highly-fit points in the search space, but also good *search strategies*, enabling GAs to solve more complex real-world problems.

Although evolvability is beneficial in the long term, it is not clear how evolvability is selected for in nature [10], or how it can be selected for in artificial evolution. Evolvability does not improve fitness immediately; in fact, in some cases of artificial selection it actually slows evolution. In artificial evolution, evolvability can be selected for directly, provided knowledge of good phenotypic variation can be built into the fitness function. However, in most interesting cases (e.g. open-ended domains), the algorithm designer does not know what constitutes good phenotypic variation and thus it must be learned by the algorithm. Therefore, evolvability must be selected for indirectly, e.g. as a side-effect of selection for some other characteristic.

One simple yet robust way of generating a selection pressure for evolvability is to gradually change the fitness function in a systematic manner over the course of evolution [9, 13]. Such methods are promising because they do not require specific information about evolvability to be built into the fitness function, but still provide a reward to genomes that learn the underlying structure of the problem. This paper analyzes various selection pressures for evolvability that are generated in this manner. In a series of experiments, representations are trained to exploit bilateral symmetry by presenting a sequence of bilaterally symmetric fitness functions. Only representations capable of manipulating phenotypic variation through the genotype-phenotype mapping are found to be evolvable. Furthermore, the degree to which representations do so is limited by how easily the resulting changes in fitness can be detected. Detectability is influ-

enced by several factors such as the fitness function, how much it changes per generation, the number of genotypic traits controlling evolvability, their mutation rates, and the size of the phenotype. This limitation suggests that representations should be designed to maximize the detectability of mutations that restructure phenotypic variation.

The paper is divided into six main sections: Section 2 reviews indirect encodings and evolvability, section 3 describes one method to induce selection for evolvability, section 4 compares the performance of two representations under several changing fitness functions, and sections 5 and 6 discuss the implications for representation design.

2. EVOLVABLE REPRESENTATIONS

The term evolvability is often used to describe an evolutionary system at the population level: populations with high genetic variance tend to be more evolvable than those with low variance. This paper focuses on evolvability at a more fundamental level, i.e. that of genetic representations. Intuitively, representational evolvability means the degree to which a particular genome is capable of extracting and exploiting the underlying structure of a particular fitness function. In general, representations that exhibit high evolvability are more capable of searching effectively through complex spaces. The benefits commonly associated with evolvability include increasing adaptive phenotypic variation (i.e. developmental constraints) [5, 15], decorrelating phenotypic traits (weak linkage) [20], and the emergence of both genotypic and morphological modularity [9, 13]. Evolvability is obtained by adaptively organizing the genotype-phenotype mapping in such a way that the search operators are more likely to produce highly fit phenotypes.

Representations capable of exhibiting such control over the genotype-phenotype mapping are known as indirect encodings and have been shown to exhibit many of the same characteristics as natural developmental systems [18]. Indirect encodings are powerful because they facilitate search in complex domains by making meaningful phenotypes more accessible to search. Given an appropriate selection pressure, such representations will adapt how genetic variation generated by the search operators maps onto phenotypic variation such that it becomes more structured and highly constrained [15]. For example, when evolving designs for tables, L-system based developmental encodings have been shown to find parameterizations of phenotypic variation that correspond to "natural" axes, such as the height and width of the table [6]. Evolvable representations thus allow search to manipulate its own bias over the course of evolution. Such restructuring cannot happen immediately, but must take place gradually as the representation exploits information gained about the fitness function. One promising method of designing indirect encodings is to utilize developmental processes [7, 18].

Although the phrase "evolvable representation" is commonly used in Evolutionary Computation (EC) literature, representations generally only have a *capacity* to become evolvable: evolvability emerges over the course of evolution with a specific fitness function, and is defined in terms of that function. Therefore, in this paper the term *latent evolvability* will be used to describe the representation's underlying capacity for becoming evolvable, and *acquired evolvability* will be used to refer to its evolvability produced in response to a particular fitness function.

The distinction between latent and acquired evolvability is important: as this study shows, representations with high latent evolvability may not exhibit acquired evolvability under certain fitness functions. Since evolvability confers mainly a *future* evolutionary benefit, it is unclear how it is selected for in nature. In general, evolvability can only be selected for in cases where an increase in evolvability is correlated with selection for some phenotypic trait. Kirschner and Gerhart [10] propose three possible methods at both the individual and clade level through which indirect selection for evolvability can be generated. First, more modular and robust organization can confer physiological advantages, such as making the organism more tolerant against stochastic development and environments. Second, because fewer mutations have harmful effects, more genetic variation can be stored in the population (i.e. they form neutral networks [8]). Third, populations with higher evolvability are more likely to survive when environmental conditions change [10].

Recent artificial evolution studies suggest several methods through which selection for evolvability, particularly through modularity, can arise [3, 4, 11, 19, 21, 23]. The simplest and most effective way is to make evolvability itself a component of fitness. A commonly studied example is Estimation-of-Distribution algorithms (EDAs), which attempt to learn evolvable phenotypic distributions by analyzing selection data from past generations [14]. Although less biologically plausible than traditional genetic algorithms, EDAs have been shown to be particular powerful in optimization problems. The primary difference in EDAs and indirect encoding methods is that EDAs adapt the search operators, whereas indirect encodings adapt the genotype-phenotype mapping.

Another effective way to select for evolvability is through a systematically varying fitness function. This approach is particularly useful when performing controlled experiments on evolvability. Evolvability can be defined purely in terms of the degree of exploitation of the underlying structure, and several different representations can be tested under the same conditions, since the selection pressure for evolvability is constant and continuous.

An experimental setup presented in this paper, consisting of a changing fitness function paired with an indirect encoding, allows acquired evolvability to be observed directly and manipulated in a controlled manner. It constitutes a first step towards isolating and analyzing the effects of various representational factors on evolvability. Specific issues regarding implementation are described in section 3.

3. GENERATING SELECTION PRESSURE FOR EVOLVABILITY

An implicit selection pressure for evolvability can be generated experimentally using two components: a changing fitness function with an exploitable structure and a genotypic representation capable of storing information regarding that structure. Section 3.1 introduces systematically changing fitness functions in general, section 3.2 describes a specific instance based on bilateral symmetry, and 3.3 describes the simplest indirect encoding capable of acquiring evolvability in that domain.

3.1 Fitness Regimes

A *fitness regime* is defined in this paper as a fitness function that gradually changes over the course of evolution.

The degree of change can be fixed or variable and is called the *target drift rate.* Fitness regimes with low target drift rates can be tracked by evolution, whereas fitness regimes that change too quickly can disrupt evolution to the point where fitness does not increase.

Of particular interest to the study of evolvability is a subset of fitness regimes where the fitness function does not change randomly, but rather has some underlying structure Γ that is invariant across all instances (i.e. a generalization of Kashtan and Alon's modularly varying fitness function [9]). When evolving under such a regime, it is beneficial for a population to identify and exploit Γ. Doing so increases the population's ability to adapt when the fitness function changes. Thus the fitness regime defines a fitness differential between genomes that exploit Γ and those that do not. Since exploiting Γ requires manipulating the phenotypic variation of the population, fitness regimes implicitly produce a selection pressure for evolvability. Furthermore, adapting the local search space in this way allows the population to find more general solutions and to escape local optima [9].

In terms of representation, each genome has a mutational cost associated with exploiting Γ. Some genomes may be able to exploit it with relatively few mutations whereas others may take many. The amount of evolvability that a representation acquires under a particular fitness regime is proportional to the number of mutations required for learning Γ. Large fitness differentials generate more selection pressure and thus more tolerance to adaptations that may take many mutations. If a certain selection pressure is not enough to meet the representation's threshold, then evolvability will not emerge. Representations that exhibit high evolvability even when there is little selection pressure are desirable, especially in cases like co-evolution, where large fitness differentials cannot necessarily be guaranteed.

Since static fitness functions do not provide pressure to select for evolvability, even if a representation has high latent evolvability, it may exhibit little acquired evolvability under such circumstances. This insight is important because it may help explain the tendency for "brittle" evolved solutions, i.e. solutions whose local mutation space reflects simply how rugged the fitness function is. In general, static fitness functions may be a cause of low evolvability in artificial evolution, even with developmental systems [2].

3.2   Symmetry Domain

One conceptually simple fitness regime involves presenting a sequence of symmetric binary strings as the target for evolution. A target string of n bits is generated randomly with symmetry around the midpoint and used as the fitness function. This function is changed over time by flipping each bit with a predefined probability, maintaining bilateral symmetry. A distinction must be made between learning *general symmetry* of any target string that is symmetric about the midpoint and *specific symmetry* where the target string is defined explicitly. This distinction is important because the representation must generalize from each instance of specific symmetry in order to learn about general symmetry.

By varying how quickly the specific instances of bilateral symmetry change over evolution (the target drift rate), training sessions can be constructed that provide variable amounts of information about general symmetry. For example, a representation trained on only a single instance of bilateral symmetry has no pressure to exploit general
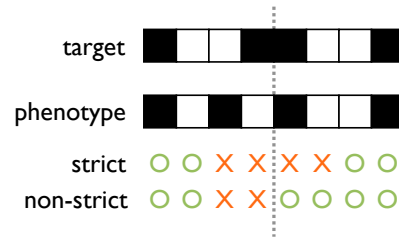

Figure 1: Fitness calculation for the strict symmetric and non-strict symmetric fitness regimes. In the strict regime, fitness is the number of traits (i.e. locations or genes) in the phenotype that match both the target string *and* the trait at the bilaterally symmetric point in the phenotype. Phenotypic traits that only match the target string do not count towards fitness. The non-strict regime relaxes this requirement, rewarding all traits that match the target, regardless of whether the bilaterally symmetric point matches. Although fundamentally similar, these two fitness regimes generate significantly different selection pressures for evolvability.

symmetry as a survival strategy. Therefore, the representation might not exhibit acquired evolvability, even though it has the capacity for it. Conversely, if the target string is changed slowly, then there *is* a pressure for such learning. Furthermore, higher rates of change yield stronger selection for evolvability.

Three fitness regimes are compared in this study, one random and two based on bilateral symmetry. Although all three fitness regimes employ a varying fitness function, each generates quantitatively different selection pressures for evolvability:

- *Random*, where the fitness function drifts without structure; the initial instance is generated randomly (with no bilateral symmetry) and all target string mutations are made at random. Individuals are rewarded for each trait (i.e. bit location or gene) that matches the target string.

- *Strict*, where target string mutations occur only on one half and are copied over to the other half to preserve symmetry. When fitness is calculated, only traits that match the target function and the trait at the bilaterally symmetric point on the genome are rewarded (figure 1).

- *Non-strict* is similar to the strict fitness regime except that all traits that match the target function are rewarded, regardless of whether they match the bilaterally symmetric point as well (figure 1).

This domain was chosen for two reasons: first, the fitness calculation is computationally simple to perform, and second, there is a clear underlying structure to the problem that remains even when the target is changed. Thus, if the representation is capable, there is an opportunity for exploiting the invariance in order to become more evolvable. Furthermore, no other opportunities exist to do so.

3.3   Linkage Representation

In order to make use of the selection pressure for evolvability, the solution representation must be able to adapt its genotype-phenotype mapping. For ease of analysis, the simplest indirect encoding capable of exhibiting acquired evolv-
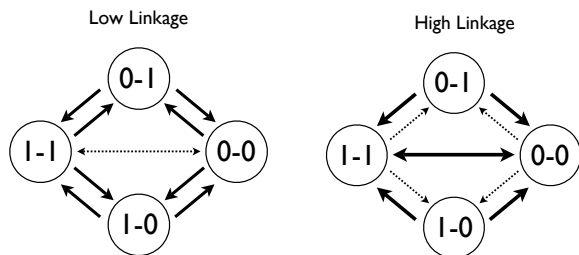
Figure 2: Effects of the linkage parameter on the local mutational space. Circles indicate symmetric phenotypic traits, bold arrows show high probability mutations, and dotted arrows indicate low probability mutations. With low linkage, mutations between each of the four combinations of phenotypic values (0-0, 0-1, 1-0, 1-1) are possible with equal probability. However, as the linkage parameter value increases (from low to high), the symmetric settings (0-0 and 1-1) become more probable and furthermore direct mutation between the two states becomes possible. Because bilateral symmetry is rewarded more than asymmetry (except in the random domain), the linkage parameters provide a way in which the genotype can become more evolvable.

	random	non-strict	strict
control	0.000	0.014	-0.060
linkage	0.011	0.066	0.319

Table 1: The acquired evolvability of the control and linkage representations over all three domains. The evolvability score represents the average ΔL over all target drift rates. Only the linkage representation under the strict fitness regime exhibits significant acquired evolvability.

ability under the symmetry domain is employed. Genomes consist of n phenotypic traits (genes determining the phenotype), directly encoded as binary digits, and an additional set of real-valued *linkage* parameters, all of which are under evolutionary control. For genomes with high values for the linkage parameters, point mutations made to the phenotypic traits have a high probability of affecting the trait symmetrically opposite as well. Likewise, low values of the linkage parameters provide little mutational correlation between the two halves. Although the linkage parameter does not directly affect fitness, high linkage effectively adds a "shortcut" mutation, allowing pairs of symmetric phenotypic traits to mutate together. In addition, it reduces the probability that symmetric traits differ (figure 2). Thus when the target drifts, genomes with high linkage require fewer mutations to adapt the phenotypic traits. Using this representation, genomes that evolve high linkage values learn to exploit the domain structure, and thus have high evolvability; furthermore, no other methods exist to increase evolvability. Therefore, acquired evolvability can be measured directly as the average linkage values over the entire population.

The number of linkage parameters can be manipulated experimentally, ranging from one parameter per phenotypic trait to a single parameter controlling the linkage of all the phenotypic traits. This range corresponds to the amount of *a priori* information about bilateral symmetry in the representation. A single linkage parameter encodes knowledge of bilateral symmetry perfectly into the domain. If linkage is broken up over several parameters, the representation becomes capable of expressing many kinds of symmetry, but has less specific knowledge about bilateral symmetry.

In the experiment, the linkage representation is compared to a control representation with an identical genotype but where the linkage parameters control random pairs of phenotypic traits, not necessarily bilaterally symmetric ones. Since random linkage does not exploit the structure of symmetry, if the control representation evolves high linkage values, then evolution may just be selecting for a higher mutation rate, rather than evolvability. Thus comparing against this representation elucidates whether higher evolvability is actually being selected for.

4. RESULTS

A standard genetic algorithm with tournament selection, one-point crossover, and speciation [17] is run for 500 generations (until linkage values settled) on a population of 50 randomly seeded individuals. Each genome had a fixed length and consisted of binary-valued phenotypic traits and floating-point linkage parameters. For phenotypic traits, unless otherwise noted, the point mutation rate was 0.01 and the linkage parameter mutation rate was 0.2 (higher linkage parameter mutation was found to increase acquired evolvability). At the end of the run, all the linkage parameters in the population were averaged across 200 trials, yielding the linkage value L_{final}. Since linkage values are drawn initially from the uniform distribution over $[0.0, 1.0]$, the expected linkage value at generation 0, L_0, is $E(L_0) = 0.5$. Thus, $L_{\text{final}} = 0.5$ indicates neutral drift (that is, no acquired evolvability), 0.0 indicates selection for low linkage, and 1.0 selection for high linkage, i.e. acquired evolvability. Since only the linkage parameter affects the representation's evolvability, acquired evolvability can be measured directly as $\Delta L = L_{\text{final}} - E(L_0)$, which ranges from -0.5 (selection against evolvability), to 0.5 (acquired evolvability), with 0.0 indicating no change in evolvability.

The difference in average best fitness after 500 generations using the control representation and the linkage representation is shown in figure 4. Genomes consisted of 40 phenotypic traits and a single linkage parameter. Under the random fitness regime, the two representations performed equally well, indicating that the linkage parameters have no effect on fitness. Under both the strict and non-strict fitness regimes, symmetric linkage provided a fitness advantage. The fitness differential generated under the strict fitness regime was several times larger than under the non-strict regime (0.09 as opposed to 0.013). This larger differential generated a stronger selection pressure for evolvability (table 4). The fitness differential generated by the random representation was too small to induce selection for evolvability (table 4). As the target drift rate increased, average best fitness decreased from near 1.0 to 0.7 for the linkage representation, and 0.6 for the control representation.

In general, only the linkage representation exhibits acquired evolvability, and then only when paired with the strict fitness regime (table 4). When the effects of linkage are random (control representation), there is no selection for evolvability. Furthermore, using the control representation, the strict fitness regime actually induces weak selection to *lower* the linkage parameter, and hence evolvability. Thus,
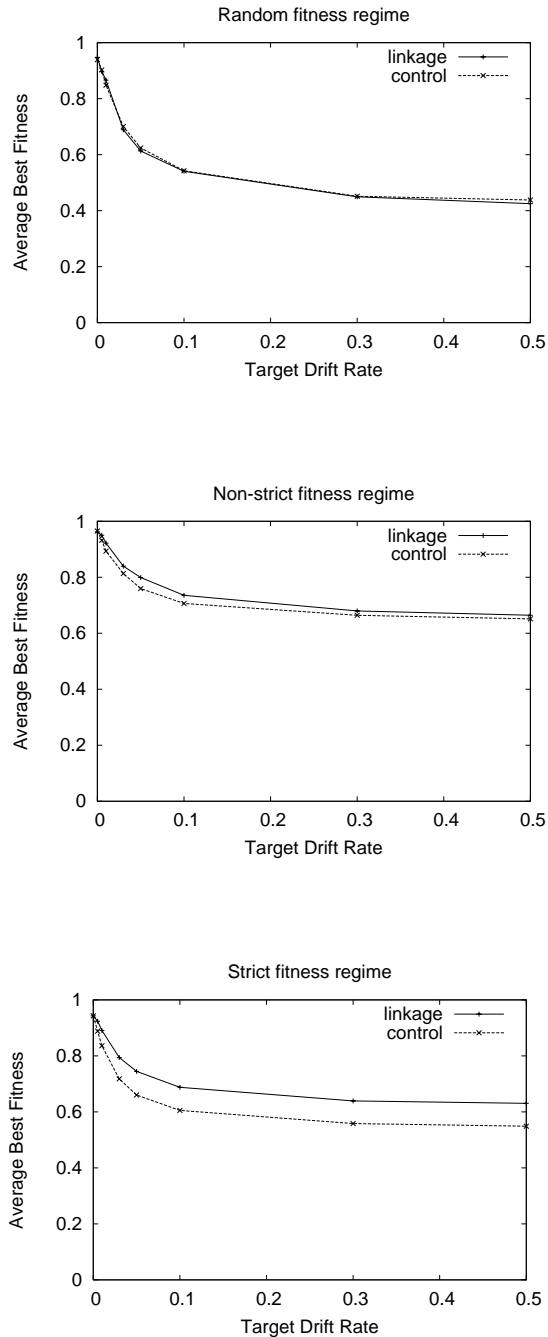
Random fitness regime

Non-strict fitness regime

Strict fitness regime

Figure 3: Fitness differentials between linkage and control representations. Average best fitness for the two representations is shown after 500 generations under the three fitness regimes: random, non-strict symmetric, and strict symmetric. Only the strict symmetric fitness regime induces a fitness differential large enough to generate a reliable selection pressure for evolvability, although the non-strict regime generates a non-zero fitness differential as well.

the linkage parameter is indeed being selected for because it affects the local mutation space, rather than simply because it leads to more mutations.

Under the strict fitness regime, as the target drift rate increases, the resulting selection pressure for evolvability also increases (figures 4 and 5). With a target drift rate of 0.0, the population with 40-bit phenotypes yields an acquired evolvability score of 0.07 after 500 generations, indicating little or no selection for evolvability. With a drift rate of 0.1, evolvability increases to 0.35 and with a target drift rate of 0.5, evolvability becomes 0.49, indicating strong selection.

As the number of phenotypic traits is increased in proportion to the number of linkage parameters, slightly higher evolvability scores are obtained under the strict fitness regime (figure 4). With a target drift rate of 0.1, genomes with 40 phenotypic traits acquire an evolvability score of 0.35, those with 80 acquire 0.39, and those with 160 acquire 0.48. Conversely, as more linkage parameters are added, the selection pressure proves too weak to generate any significant change in the linkage parameter. Under the strict fitness regime, with one linkage parameter, the maximum average linkage is 0.49, with two parameters, it falls to around 0.29, and with five to around 0.12. Under the unlinked fitness regime, one linkage parameter yields a maximum linkage parameter value of 0.16, two yields 0.09, and five yields 0.04 (figure 5). Varying the number of phenotypic traits and linkage parameters affects the size of the fitness differential created by linkage parameter point mutations: more phenotypic traits increases the fitness differential, whereas more linkage parameters decreases it. As the fitness differential is decreased, selection for higher linkage becomes untenable and thus the representation exhibits no acquired evolvability.

The linkage parameter mutation rate also has a profound effect on acquired evolvability. When the target drift rate is 0.5, a linkage parameter mutation rate of 0.01 yields an evolvability score of 0.246 after 500 generations. A mutation rate of 0.05 yields 0.31, 0.1 yields 0.37 and 0.2 yields 0.48. Evolvability can only be selected for when the fitness function changes in conjunction with a beneficial phenotypic mutation resulting from linkage. Since the probability of these events occurring in the same genome within the same generation is low, and since linkage mutation is not disruptive on average, a higher mutation rate can only be beneficial to selection for evolvability.

In conclusion, acquired evolvability is found to be bounded by the detectability of individual mutations shifting the phenotypic distribution. The threshold of detectability can be raised by increasing the fitness differential induced by the fitness regime, or lowering the number of mutations required to adaptively shift the phenotypic distribution.

5. DISCUSSION

Several intuitive results were observed in the simulations: increasing linkage increases fitness especially under harsh environmental conditions (i.e. target drift rate greater than 0.1); more selection pressure for evolvability is generated with higher target drift rates; increasing the number of phenotypic traits per genome increases acquired evolvability; increasing the number of linkage parameters decreases acquired evolvability; increasing the mutation rate of the linkage parameters increases acquired evolvability; the choice of fitness regime determines the fitness differential between representations with and without acquired evolvability.
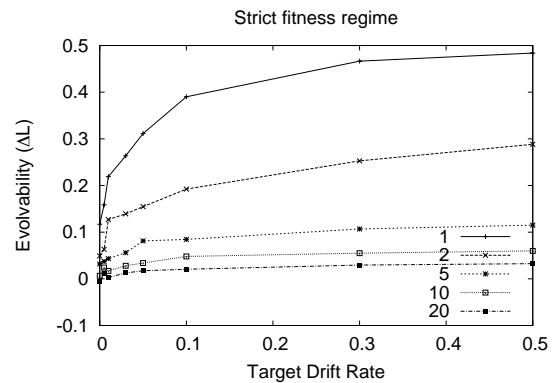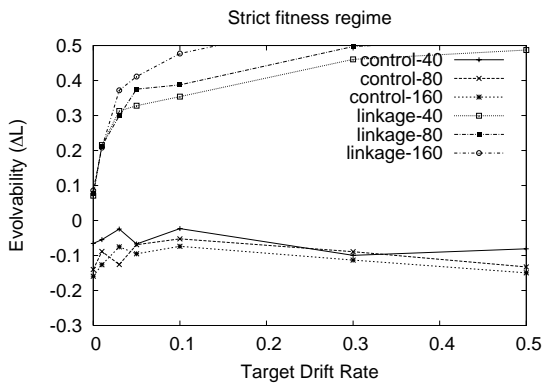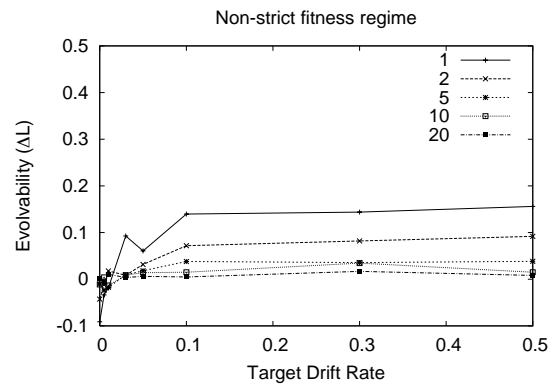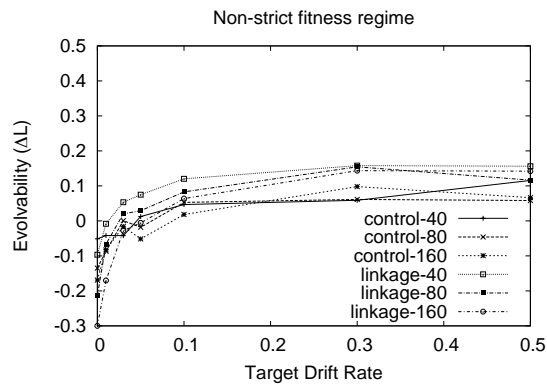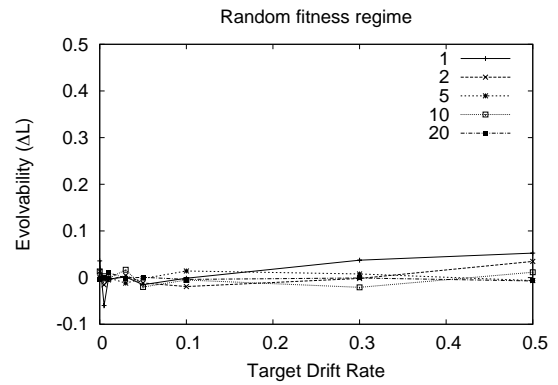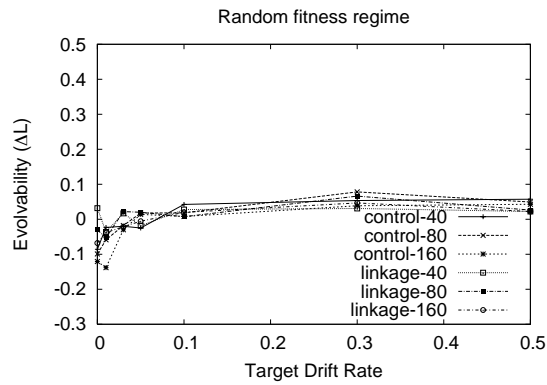
Figure 4: Evolvability with 40, 80, 160-bit phenotypes. Comparison of linkage vs. target drift rate after 500 generations. "Linkage" and "control" are the two different representations and "strict", "non-strict" and "random" are the fitness regimes. As target drift rate increases, acquired evolvability increases in the linkage representation under the strict and non-strict regimes. As the phenotype size increases, the acquired evolvability increases.

Figure 5: Effects of increasing the number of linkage parameters on acquired evolvability (40 bit phenotypes, linkage representation). Under both fitness regimes the acquired evolvability decreases, i.e. the average change in linkage tends towards 0.0 as more linkage parameters are added. Representations with more linkage parameters require correspondingly more linkage mutations to induce a measurable change in evolvability.

From the experimental results it is clear that evolvability, rather than a high mutation rate, is being selected for under the strict symmetric fitness regime. However, this particular selection pressure is exceedingly weak, unable to produce significant evolvability when the linkage parameter is split over more than two genes. This result suggests that there exists a lower bound on acquired evolvability based on the *detectability* of the linkage parameters' effect on fitness. Since linkage parameters affect future phenotypic variation, it may take many generations for evolution to detect that a linkage change has had any effect on average fitness. Furthermore, selection based on detected changes is not reliable because further mutations may have altered the linkage parameter in the interim. Detectability represents an interplay between representation and the fitness function: fitness regimes that induce a large selection pressure for evolvability make up for representations that require a large number of mutations to adapt phenotypic variation. Conversely, representations that require few mutations to acquire evolvability will do so even when the fitness regime induces a relatively weak selection pressure.

Detectability has practical implications for the design of evolvable representations. Both the linkage representation with a single linkage parameter and the linkage representation with five linkage parameters have the same latent evolvability: If the linkage parameters are all set to 1.0, both representations will exhibit the same increase in performance on the symmetry domain. However, only the linkage representation with a single linkage parameter actually acquires evolvability under the symmetric fitness regime. The reason is that many more linkage mutations are required to detect the fitness gradient in the five parameter case; single linkage mutations have 1/5 the chance of affecting a given phenotypic trait, rendering the change undetectable and thus unselectable in most cases.

A single linkage parameter affecting all phenotypic traits corresponds to designing a representation with perfect knowledge about bilateral symmetry. As the number of linkage parameters is increased, each individual parameter affects fewer phenotypic parameters, but the possibilities for other kinds of linkage (not just perfect bilateral symmetry) become available. Thus as the number of traits increases, the bias towards bilateral symmetry decreases. However, even in representations with as few as five linkage parameters, the selection pressure on evolvability proves to be too weak to generate significant acquired evolvability. Having too many linkage parameters diminishes the benefits of linkage to the point where it is no longer possible to select for. In other words, the fitness benefit per mutation is below the detectability threshold. Thus there is an inherent trade-off between creating tuned representations that exhibit high evolvability but require much *a priori* problem information, and more general representations that require less knowledge but behave poorly.

It is likely, however, that this trade-off can be circumvented by designing representations that minimize the number of mutations required to adapt phenotypic variation to reflect the problem structure, or by using co-evolutionary methods on open-ended problems. Co-evolution in particular is capable of generating sustained fitness differentials over the course of evolution, as opposed to fitness differentials that are fixed and thus decrease as the individuals in the population become more evolvable. Furthermore, when learning the problem structure is key to solving the problem, the effects of this trade-off will be diminished.

Analyzing the interplay between representation and fitness regime also sheds light on earlier results on measuring evolvability [16]. In that study, a version of the linkage representation was tested where each phenotypic trait had an associated linkage parameter. Despite using a bilaterally symmetric fitness regime, that representation only exhibited acquired evolvability when the individual fitness scores were averaged together with fitness scores of genotypes nearby in the local mutation space. Without a direct selection pressure for evolvability, the linkage representation exhibited no acquired evolvability because too many mutations were required in order to exploit symmetry. In other words, the selection pressure was too low to overcome the limitations imposed by the representation.

Ultimately these results shed light on designing evolvable representations: representations should be constructed in such a way that phenotypic variation can be adapted to match the structure of the fitness function in as few mutations as possible. More mutations requires a correspondingly larger selection pressure to be tenable. Thus, in order to be maximally evolvable, representations must be able to adapt the genotype-phenotype mapping at all levels, ranging from fundamental design changes to small phenotypic tweaks.

6. FUTURE WORK

The results of this study suggest several areas of future work: measuring evolvability, evolvability in co-evolutionary systems, the effects of evolvability on GA performance, and evolvability in nature.

Representations that are highly evolvable under some fitness regimes may not be evolvable under others. For example, the linkage representation exhibited little evolvability under the non-strict symmetric fitness regime because there was little evolutionary benefit to exploiting the linkage parameter. Thus when measuring evolvability, the score must be normalized by the selection pressure. Normalization ensures that representations that exhibit evolvability when there is little selection pressure for it are evaluated correctly, allowing results obtained under different fitness regimes to be compared directly.

Fitness regimes naturally cause evolution to select for higher evolvability when the representation has latent evolvability. With direct encodings [12, 24], on the other hand, mutations cannot be made in a way such that information about the fitness function is accumulated to bias phenotypic variation. Thus direct encodings have no latent evolvability, and will not respond adaptively to evolution under a fitness regime. This fact has particular implications for co-evolution: Past information about the opposing population will simply be discarded as evolution progresses, which may lead to populations cycling between behaviors or disengaging completely. Using co-evolution coupled with evolvable indirect encodings may alleviate this problem. Furthermore, the open-ended co-evolution fitness regime is more general and powerful than the static fitness regimes evaluated in this paper and in [9] because it can maintain large fitness differentials continually over the course of evolution. In contrast, with static fitness regimes the fitness differentials are finite and tend to zero as the representation exploits the underlying structure. An important direction for future work is to develop a measure of the selection pressure for evolvability

generated in co-evolutionary settings, to elucidate the factors that affect it. This line of inquiry could lead to more evolvable co-evolutionary systems, which should in turn help overcome common problems such as disengagement.

Selection for evolvability may in some cases retard GA performance. In a separate set of experiments (unpublished) using a more complex genetic regulatory network representation, first training under a fitness regime and then testing performance on static fitness function was found to perform significantly slower on average compared with populations first trained on a single static fitness function. This result is intuitive: selection for evolvability prefers long-term fitness gains that are averaged over many different instances of the fitness function, which may yield relatively poor fitness under any particular instance. However, ultimately representations that are both highly evolvable and highly efficient are desired. Therefore it is necessary to devise methods in which both evolvability and efficiency can be rewarded.

There is growing evidence both from theory [13] and computational simulation [9] that more evolvable genotypes can be selected for. However there is little evidence for such selection from biological studies [10]. Understanding how evolvability arises naturally will shed light on how immense morphological complexity and variation is generated in the metazoans. Can these results from artificial evolution be used to augment biological research? One important next step would be to study selection pressures for evolvability on more biologically plausible platforms such as Avida [1].

7. CONCLUSION

Using a conceptually simple framework of bilateral symmetry, some limitations associated with selection for evolvability were demonstrated. Selection pressure for evolvability was generated by making systematic changes to the symmetric fitness function during the course of evolution. Evolvability was only exhibited by representations that store information learned from the changing fitness function. Furthermore, acquired evolvability was limited by the number of mutations required to detect the induced fitness gradient. Finding effective methods through which to overcome this limitation is an important direction for future work in designing representations and fitness functions. Ultimately, more evolvable evolutionary algorithms will aid in finding solutions to more complex problems by exploiting their underlying structure.

8. REFERENCES

[1] C. Adami. *Introduction to Artificial Life*. Springer, New York, 1998.

[2] P. Bentley and S. Kumar. Three ways to grow designs: A comparison of embryogenies for an evolutionary design problem. In *Proc. of the Genetic and Evolutionary Computation Conference*, pages 35–43, Orlando, Florida, USA, 1999. Morgan Kaufmann.

[3] R. Calabretta, S. Nolfi, D. Parisi, and G. P. Wagner. Duplication of modules facilitates the evolution of functional specialization. *Artif. Life*, 6(1):69–84, 1999.

[4] M. Ebner, M. Shackleton, and R. Shipman. How neutral networks influence evolvability. *Complexity*, 7(2):19–33, 2001.

[5] T. F. Hansen. Is modularity necessary for evolvability? Remarks on the relationship between pleiotropy and evolvability. *BioSystems*, 69:83–94, 2002.

[6] G. S. Hornby. Functional scalability through generative representations: The evolution of table designs. *Environment and Planning B: Planning and Design*, 31(4):569–587, 2004.

[7] G. S. Hornby and J. B. Pollack. The advantages of generative grammatical encodings for physical design. In *Proc. of CEC 2001*, 2001.

[8] M. A. Huynen. Exploring phenotype space through neutral evolution. *Journal of Molecular Evolution*, 43:165–169, 1996.

[9] N. Kashtan and U. Alon. Spontaneous evolution of modularity and network motifs. *Proc. of the National Academy of Sciences*, 102(39), 2005.

[10] M. Kirschner and J. Gerhart. Evolvability. *PNAS*, 95:8420–8427, 1998.

[11] V. Kvasnička and J. Pospíchal. Emergence of modularity in genotype-phenotype mappings. *Artificial Life*, 8(4):295–310, 2002.

[12] G. E. Liepins and M. D. Vose. Representational issues in genetic optimization. *Journal of Experimental and Theoretical Artificial Intelligence*, 2:101–115, 1990.

[13] H. Lipson, J. Pollack, and N. Suh. On the origin of modular variation. *Evolution*, 56(8):1549–1556, 2002.

[14] M. Pelikan, D. E. Goldberg, and F. G. Lobo. A survey of optimization by building and using probabilistic models. *Computer Optimization and Applications*, 21(1):5–20, 2002.

[15] R. A. Raff. *The Shape of Life: Genes, development, and the Evolution of Animal Form*. The University of Chicago Press, 1996.

[16] J. Reisinger, K. O. Stanley, and R. Miikkulainen. Towards an empirical measure of evolvability. In *Proc. of the Genetic and Evolutionary Computation Conference Workshop Program*, 2005.

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

[18] K. O. Stanley and R. Miikkulainen. A taxonomy for artificial embryogeny. *Artif. Life*, 9(2):93–130, 2003.

[19] A. Thompson and P. J. Layzell. Evolution of robustness in an electronics design. In *Proc. of the Third International Conference on Evolvable Systems*, pages 218–228, London, UK, 2000. Springer-Verlag.

[20] M. Toussaint. Compact representations as a search strategy: Compression EDA. *Theoretical Computer Science*, In press.

[21] E. A. Variano, J. H. McCoy, and H. Lipson. Networks, dynamics and modularity. *Physical Review Letters*, 92(18), 2004.

[22] G. Wagner and L. Altenberg. Complex adaptations and the evolution of evolvability. *Evolution*, 50(3):967–976, 1996.

[23] R. A. Watson and J. B. Pollack. Symbiotic composition and evolvability. In *Proc. of the 6th European Conference on Advances in Artificial Life*, pages 480–490, London, UK, 2001. Springer-Verlag.

[24] D. Whitley, S. Rana, and R. Heckendorn. *Representation Issues in Neighborhood Search and Evolutionary Algorithms*, pages 39–58. John Wiley and Sons Ltd., 1997.