Available online at www.sciencedirect.com
ScienceDirect
BioSystems 90 (2007) 769–782
BioSystems
www.elsevier.com/locate/biosystems

Complex adaptation and system structure

Marc Toussaint [a,*], Werner von Seelen [b,1]

[a] Honda Research Institute Europe GmbH, Carl-Legien-Strasse 30, 63073 Offenbach/Main, Germany
[b] Institut für Neuroinformatik, Ruhr-Universität Bochum, ND 04, 44780 Bochum, Germany

Received 5 January 2006; received in revised form 16 January 2007; accepted 7 March 2007

Abstract

The structural organization of biological systems is one of nature's most fascinating aspects, but its origin and functional role is not yet fully understood. For instance, basic adaptational mechanisms like genetic mutation and Hebbian adaptation seem to be generic and invariant across many species and are, on their own, fairly well investigated and understood. However, it is the organism's structure – the representations these mechanisms act upon – that bears the complex functional effects of these mechanisms. While typical technical approaches to system design require detailed problem models and suffer from the need to explicitly take care of all possible cases, the organization of biological systems seems to induce inherent adaptability, flexibility and robustness. In this discussion paper we address the concept of structured variability, particularly the role of system structure as implementing a certain representation on which basic variational mechanisms act on. The functional adaptability (or search distribution) depends crucially on this representation.

Keywords: Representations; Structural variability; Evolvability

1. Introduction

Conventionally, when designing a technical system the engineer assumes to have a full overview over all possible situations the system has to cope with and how the system should behave in each situation. Every execution step of the system is planned in detail by the designer assuming that he can anticipate and organize every eventuality. However, there is often a lack of a general model concerning the environment and disturbances and the necessary data and potential prior knowledge are expensive and incomplete. Numerous approaches to systematize the design process exist, but generally preserving traditional principles concerning the role of system structure. Eventually, the increasing complexity of control systems, their embedding in natural environments, and the necessary flexibility of these systems lead to a design complexity that requires changes within the traditional design approach. We will propose a point of view that focuses on the decisive role of the system structure in determining general variational system properties like adaptability and robustness.

Consider a system composed of many parts and assume that we can assess a quality measure for each system (we have a fitness function). Finding good systems becomes complex if there are dependencies between system parts w.r.t. the quality. Throughout this paper, this is what we mean by a complex problem. There are many ways to define this notion of problem complexity more precisely. Intuitively, the quality might depend on whether one part of the system fits with

* Corresponding author.
E-mail address: mtoussai@inf.ed.ac.uk (M. Toussaint).
[1] Currently with the Honda Research Institute Europe GmbH.

0303-2647/$ – see front matter © 2007 Elsevier Ireland Ltd. All rights reserved.
doi:10.1016/j.biosystems.2007.03.004

another. Probabilistically, one might interpreted the quality function as a distribution over system space (as in Eq. (2)) and find that the variables describing systems parts are dependent, measurably in terms of the mutual information between them. Classically, one might say that a main effects model (1st order ANOVA model) of the quality is not a good approximation; higher order dependencies need to be considered. In all these cases, system design is hard since independent local adaptations of the system often violate necessary dependencies between system parts and system functionality fully breaks down. To still achieve innovations, every adaptation step becomes increasingly complex in the sense that the system has to be coherently adapted at many places to account for the dependencies. The cost of such adaptation steps increases with the number of dependencies that have to be considered—which is exponential with the number of system parts.

However, natural processes that lead to complex systems typically rely on mutually independent local variations, for example, independent mutation incidents on the genome. Here a fundamental question arises: How can independent variation lead to systems which heavily rely on the precisely coordinated interplay of their parts? Or inversely: How can natural adaptation processes apparently decompose the problem of complex system design, such that independent adaptations lead to progress? In addition, natural systems are, despite their complexity, still robust against failure of different origins, and highly flexible to quickly adapt to a changed domain of application. We will review arguments that the ability to design complex systems based only on local independent adaptations and the robustness and flexibility of the developed systems has a common origin: the variational structure of these systems as it is induced by their architectural design, which was co-optimized during system evolution parallel to the actual system functionality. In turn, adaptability and robustness are not only additional attributes of natural systems, they can be seen as a method capable to yield complex systems. The notion of variational structure (or the structure of the variational distribution) will play a central role throughout the discussion. Similar to problem complexity it captures whether there are dependencies between system parts w.r.t. the variation (mutation, adaptation, learning) of the system and may also be interpreted as topological or metric structure in the search space.

In this paper we want to review various perspectives on how one can approach the problem of complex system design. For most parts, we will frame the problem as a search problem for two reasons. First, we need to discuss evolution, which can be understood as a search process, as a paradigm of how complex systems can be found. In particular, there are recent advances in understanding how evolution can find complex systems even though the basic search mechanism – mutation – seems primitive and unstructured in view of the mentioned dependencies between systems parts. The key here is that evolution can find a suitable language to describe systems, a suitable representation of the space of systems, i.e. a genetic system. We will discuss how it is possible to develop such genetic representations. Second, we want to include recent developments in the field of stochastic search (or optimization) algorithms in our discussion. Here, the problem of finding complex systems, but also the problem of finding suitable representations, can be formulated more rigorously, which will help us to gain a more theoretically grounded basis for defining and discussing variational structure and what it means that it is adapted to the problem structure.

The paper is organized as follows. The relations between search and representations are discussed in Section 2, with particular emphasis on genetic representations in evolutionary search, estimation-of-distribution algorithms and compact and factored representations. Thereafter we extend the discussion beyond the framework of search, also addressing systems which are themselves adaptive.

2. Evolutionary search and genetic representations

Evolution has developed an ingenious language to describe organisms: the genetic system. An interesting example are homeobox genes (hox genes), operons, or genes like the eye-less gene (Halder et al., 1995). A single gene may control the development of a large-scale module that is itself composed of many parts. For instance, a single mutation of the eye-less gene during the developmental process can lead to the additional growth of a whole functionally complete eye on some part of a Drosophila's body (though without functional connectivity to the nervous system). Clearly, this single gene does not describe the eye in all it's details. Rather, the gene controls in a strongly hierarchical way a process that leads to the growth of the eye and the expression of many other genes.

Genes like the eye-less gene correspond to a genetic macro, an abbreviation of a useful sub-processes of development, a word that is part of the language to describe organisms. In this language it is sufficient to express this single word to denote a complex compound. A gene as a macro for a compound is only one

example—a language has more possibilities to induce complex variability. Chenn and Walsh (2002) show that a mutation of a single gene can triple the size of the brain of a mouse. Such a variation is not an addition of a module but a correlated variation of the whole system at many parts. Riedl (1977) already claimed that the essence of the operon is to introduce correlations between formerly independent genes in order to adopt the functional dependence between the genes and their phenotypic effects and thereby increase the probability of successful variations. In fact, the meaning and advantages of this complex relation between genes and phenes are widely discussed ((Wagner et al., 1997, 1998; Hansen and Wagner, 2001a, b)).

The genetic language is presumably not a universal one, i.e., it would not allow to describe (and let grow) all possible organisms. Instead, the vocabulary is efficient to specifically describe those kinds of organisms that evolution eventually has found: multi-cellular organisms, compounds of uniform cells, homogeneous and self-similar structures, etc. So, while it might not be surprising that evolution has utilized some language to describe organisms, the interesting question is why and how it has developed this specific language, i.e., why it developed this specific vocabulary instead of another one of all the possible ones. Since the choice of the genetic representation of a phenotype decisively determines the possibilities of phenotypic innovations, it has been argued that the genetic representations in todays natural organisms are not a mere incident. Instead, they should be the outcome of an adaptive process that optimized these representations with respect to the phenotypic variability and "innovatability" they induce (i.e., the chance of exploring new, functionally advantageous phenotypes, (Wagner and Altenberg, 1996)) or as stabilizing mechanisms (e.g., of canalization, (Wagner et al., 1997)). In fact, it is easy to believe that genes like the eye-less gene facilitate evolutionary search; the probability that an additional whole eye somewhere on the body is beneficial is not insignificant, while a few additional eye-cells somewhere on the body without the whole eye compounds are very unlikely to be useful. The role of the language here is to induce complex variations of the organisms and allow for complex innovations during evolution.

What we aim for is in the remainder of this section is a more rigorous formalism for such arguments. We will review some recent advances in the field of evolutionary computation that shed light on the evolution of genetic representations and the ability for complex adaptation from a more algorithmic and theoretical point of view.

2.1. Genotype–phenotype mappings, and genetic and phenotypic variability

To go beyond arguing for the plausibility of specific genetic representations one should propose a theory on how they evolve and what selectional mechanisms guide their evolution. Existing models concentrate on the evolution of gene interactions, e.g., on smooth landscapes ((Rice, 1998, 2000)) or on NK-landscapes that are themselves subject to evolution ((Altenberg, 1994, 1995)). Wagner (1996) also discusses the evolution of directly encoded networks of interacting genes and Wagner and Mezey (2000) propose a multi-linear model of gene interaction. We start our discussion with a general definition of genotype–phenotype mappings.

The way solutions are represented in evolutionary processes can be abstracted in terms of a genotype–phenotype mapping $\phi : G \to P$ (abbrev., GP-map). Here a genotype $g \in G$ describes the solution and comprises all the information that is passed to the next generation via mutation and recombination. The notion of a phenotype can be defined in various ways (Mahner and Kray, 1997); we choose to define a phenotype $\phi(g)$ roughly as all functional properties of g, i.e., essentially the "solution" g describes. Hence, $\phi(g)$ comprises all potentially selection-relevant properties of g.

Earlier work on the evolution of genetic representations ((Altenberg, 1994, 1995)) considered the GP-map itself as subject to evolution, i.e., ϕ being itself determined by the genotype (e.g., reflecting the intuitive point of view that the transcription factors themselves are encoded on the genome). Mathematically, however, saying that a mapping $\phi : g \to x$ depends itself on g is somewhat circular. A formaly cleaner and simpler approach is to think of ϕ as a constant map determined by the physics and chemistry of nature that maps any possible genetic system g as a whole to the corresponding organism.[1] Then, an evolution of genetic representations is possible when there exist in principle different genetic representations (genotypes) for the same solution (phenotype). Fig. 1 illustrates an implication of such a non-injective (many-to-one) genotype–phenotype mapping. We have two genotypes g_1, g_2 that map to the same phenotype x. The figure also illustrates the genetic variability σ_{g_1} and σ_{g_2} of the two genotypes, respectively. Formally, σ_g is a probability distribution over G, describing the probability of a mutation from g to another

[1] Formally, also an "inherited initial organism", e.g. an egg cell, should be thought of as part of g. In fact, in the grammar encoding in Section 2.3 the start string x_0 is defined part of g.

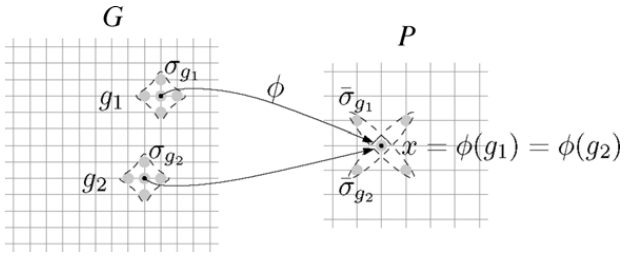

Fig. 1. Illustration of the effect of non-trivial neutrality: the phenotypic variability $\bar{\sigma}_{g_1}$ respectively $\bar{\sigma}_{g_2}$ depends on the choice of genetic representation g_1 or g_2 of the same phenotype x.

genotype g'. Both genotypes g_1 and g_2 have simple (unstructured, in the sense of uncorrelated) genetic variabilities σ_{g_1} and σ_{g_2} on the genotype space G. But when projected onto the phenotype space, they induce different phenotypic variation distributions $\bar{\sigma}_{g_1}$ and $\bar{\sigma}_{g_2}$ "around" the same phenotype. Here, $\bar{\sigma}_{g_i} = \Xi\, \sigma_{g_i}$ is defined via the phenotypic projection

$$\Xi : \sigma \to \bar{\sigma} : \quad \bar{\sigma}(x) = \sum_{g \in [x]} \sigma(g).$$

Here the set $[x] = \phi^{-1}(x) \subset G$ contains all genotypes that describe the same phenotype x (all genetic representations of x) and is also called neutral set. On the level of molecular evolution, Schuster et al. (1994); Fontana and Schuster (1998) analyzed an impressive example for non-fixed phenotypic neighborhoods that depend on the genetic representation. In their studies, the "organisms" are proteins whose functionality is determined by their three-dimensional molecular shape. This shape is the outcome of a complex folding process and eventually depends on the protein's amino acid sequence—the genotype. The authors showed that the same phenotype (3D shape) can be realized by a large number of different genotypes. Depending on the genotype, the phenotypic neighborhoods change so severely that almost any phenotype becomes possible as an offspring when only a suitable genetic representation is given.

To summarize, σ_g describes the genetic variability of a genotype g which, for simplicity, we assume to be non-adaptive and pre-determined by elementary mutation. And $\bar{\sigma}_g$ describes the functional variability of a phenotype $x = \phi(g)$ which, however, is not pre-determined but depends on the choice of genetic representation.

Clearly, for a given problem and a given solution x, some functional variability is preferable over others and hence some genetic representation is preferable over others. The functional variability of a solution captures properties like how adaptive the solution is, how likely a transition to a better solution is, how easily the solution can be adapted in case of a slight change of problem. We will discuss such system properties later in detail. Let us first consider how evolutionary search selects between different genetic representations in order to adapt the functional variability.

2.2. Matching the functional variability σ_g with the problem structure

In natural evolution theory as well as in the field of evolutionary computation there have been many approaches to describe the implicit evolutionary process on such neutral sets (Kimura, 1986; Schuster et al., 1994; Nimwegen and Crutchfield, 1999; Wilke, 2001; Toussaint, 2003b). For instance, the concept of effective fitness (Nordin and Banzhaf, 1995; Stephens and Vargas, 2000) describes the implicit selection pressure on different genotypes in a neutral set. At first it seems that, because all genotypes in a neutral set have the same phenotype and thus the same fitness, there should be no selectional pressure between them. However, effective fitness also captures the implicit effect of the genetic and functional variability of different genetic representations in a neutral set. The evolutionary process on such neutral sets describes the selection between different genetic representations depending on the functional variability they induce. In the words of Wagner and Altenberg (1996, section 6), such a seletional pressure is due to the "required match between the functional relationships of the phenotypic characters [in the selection] and their genetic representation [inducing the functional variability]." We want to capture this match formally as follows.

The effective fitness of a genotype g depends on the functional variability $\bar{\sigma}_g$ induced by that genotype and is related to the average fitness of the offspring (Nordin and Banzhaf, 1995; Stephens and Vargas, 2000):

$$\langle f \rangle_{\bar{\sigma}_g} = \sum_{p \in P} \bar{\sigma}_g(p)\, f(p). \tag{1}$$

(We omit to subtract $f(\phi(x))$ for convenience.) This simple measure for adapted variability can be interpreted in a more structural way. Consider a distribution over some random variables (genes, system features, etc.). By structure we mean more formally the dependencies between these random variables within this distribution, for instance, as they are captured by a graphical model (see (Toussaint, 2005) or the work on EDAs (Pelikan et al., 1999)). Given our definition of what we mean by a complex problem, capturing the dependencies between different phenotypic variables is decisive for finding solutions to complex problems. In other words, shaping the functional variability $\bar{\sigma}_g$ in a way that it captures

dependencies between variables of a phenotype is thus the crucial guideline for successful developmental processes.

So what are the structural implications of maximizing (1)? In order to make use of the notion of structure in probabilistic models we can recast the objective function f as a distribution, namely its Boltzmann distribution:

$$F(p) = \frac{1}{Z} \exp f(p), \tag{2}$$

with the normalization constant Z. Then the structure and the complexity of this objective function may be captured with the dependencies (mutual information) between phenotypic variables in F. The more and the higher-order dependencies there are, the more complex one would consider the problem.

The effective fitness $\langle f \rangle_{\bar{\sigma}_g}$ can now be rewritten in terms of the similarity between $\bar{\sigma}_g$ and F, namely:

$$\langle f \rangle_{\bar{\sigma}_g} = -D(\bar{\sigma}_g || F) - H(\bar{\sigma}_g) + \text{const.} \tag{3}$$

Here, $D(q||r) = \sum_p q(p) \ln \frac{q(p)}{r(p)}$ denotes the Kullback–Leibler divergence between two distributions and $H(q) = -\sum_p q(p) \ln q(p)$ the entropy of a distribution (Cover and Thomas, 1991). This is just a trivial rewriting of the original effective fitness in terms of measures on probability distributions. But this helps us to connect various points of views, including the "match" mentioned by Wagner and Altenberg (1996), work on probabilistic search algorithms, and work on compact representations.

The first observation is that $\langle f \rangle_{\bar{\sigma}_g}$ is maximized when the entropy of $\bar{\sigma}_g$ is zero, i.e., for the Kronecker-delta at some maximum of f. However, a Kronecker-delta means that we actually have no functional variability (no adaptability, flexibility, etc.). So it is natural to ask, under the constraint that there must be variability, what is the best variability? Given this constraint, maximization of $\langle f \rangle_{\bar{\sigma}_g}$ reduces to minimization of $D(\bar{\sigma}_g || F)$. The Kullback–Leibler divergence between two distributions is a similarity measure that, in particular, also measures the structural similarity between the distribution, i.e., whether they exhibit the same kind of correlations between variables (see also Amari (2001)). This means, a good variability $\bar{\sigma}_g$ obeys the same dependencies between phenotypic variables as f requires to; in other terms, implements precisely this coordinated interplay between system parts that are necessary for a reasonable solution of the problem.

Eventually, we found that the straight-forward objective to maximize the expected progress indeed includes a very structural aspect. A functional variability $\bar{\sigma}_g$ will maximize this expected progress only if it implements the dependencies between phenotypic variables required by the problem; i.e., when there is a match between the structure of the problem with the structure of $\bar{\sigma}_g$.

Concerning the entropy term in (3), reduction of phenotypic entropy can be interpreted as canalization. In fact, the decomposition of $\langle f \rangle_{\bar{\sigma}_g}$ into the Kullback–Leibler divergence and the entropy corresponds differentiating between the structure and the amount of variability. Since a lower limit on the amount of variability is sensible in any search approach, stochastic search algorithms often control the structure and the amount of variability separately (e.g., (Hansen and Ostermeier, 2001)).

Further, the Kullback–Leibler divergence can be related to compact and factored representations, as we will mentioned in Section 2.4. This closes the bridge between a well structured adaptability and the representation of solutions. Please refer to (Toussaint, 2006) for more details on this relation.

2.3. An example for the evolution of genetic representations

In the following we want to present a concrete example of the evolution of genetic representations for an illustrative toy problem: the evolution of artificial plants Toussaint (2003a). The aim of this demonstration is mainly to give an example for very complex representations where the neutral sets $[x]$ are actually of infinite size (where there exists an infinite number of possibilities to represent a solution). These representations are grammar-based and the implicit selection of genetic representations is indeed successful to find suitable representations, namely, a suitable language to describe solutions. We feel it is important to consider such complex examples (rather than, say, adaptation on simplified neutral networks on the hypercube) to remind us that in nature there exists a huge space G of possible genetic systems (e.g., including RNA as well as DNA encodings) between which transitions are in principle possible. The space of all grammars gives at lease a very crude picture of such complexity.

In this example (taken from (Toussaint, 2003a)), phenotypes are strings of growth commands (in the alphabet {F, +, −, &, ^, \, /, [,],.} to grow a plant (see (Prusinkiewicz and Lindenmayer, 1990), on details of this encoding of plants). The meaning of these symbols is briefly described in Table 1. For example, the sequence ⟨FF[+F][−F]⟩ represents a plant for which the stem grows two units upward before it branches in two arms, one to the right, the other to the left, each of which has one unit length and a leave attached

Table 1
Description of the plant grammars symbols, cf. (Prusinkiewicz and Hanan, 1989)

F	Attach a unit length stem
+, −, &, ^, \, /	Rotations of the "local coordinate system" (that apply on following attached units): four rotation of δ degree in all four directions away from the stem (+, −, &, ^) and two rotations of δ degree around the axis of the stem (\, /).
[	A branching (instantiation of a new local coordinate system)
]	The end of that branch: attaches a leave and proceeds with the previous branch (return to the previous local coordinate system); also if the very last symbol of the total sequence is not], another leave is attached
·	Does nothing

at the end. In the following we will use the normal alphabet ABC... to denote these symbols. A fitness function for plants is defined which rewards the amount of light that their leaves collect while punishing the plant's total weight. The genotypes are grammars (or L-Systems) $\langle x_0, \pi_1, \ldots, \pi_n \rangle$ which are given as a start string x_0 and a number of productions (rewriting rules) π_i. Productions are of the typical form A:BC...D, which would mean that every A in a string is replaced by the sequence BC...D. The genotype phenotype mapping reads $\phi(\langle x_0, \pi_1, \ldots, \pi_n \rangle) = \pi_n \circ \cdots \circ \pi_1(x_0)$.

In grammar encodings there exist many genotypes to encode the same phenotype and the neutral sets have very special structure. The following table gives two example genotypes, which both represent the same phenotype but induce very different phenotypic variability (cf. Fig. 1):

Genotype g	Phenotype x	Phenotypic variability $\bar{\sigma}$
⟨BCBC⟩	⟨BCBC⟩	⟨∗CBC⟩, ⟨B ∗ BC⟩, ⟨BC ∗ C⟩, ⟨BCB∗⟩
⟨AA, A:BC⟩	⟨BCBC⟩	⟨∗BC⟩, ⟨BC∗⟩, ⟨∗C ∗ C⟩, ⟨B ∗ B∗⟩

The phenotypic variability $\bar{\sigma}$ is given in terms of the offspring produced by single symbol mutations on the genotype, were the * indicates a mutated symbol. The first representation is a "direct" representation of the phenotype, while the second is a more "modular" one. Note that phenotypic variability is correlated in case of the second representation.

Given this genotype–phenotype mapping, a crucial ingredient for the evolution of genetic representations are mutations that allow to traverse the neutral sets, i.e., to mutate one genetic representation of a given solution x into another. Such neutral mutations can in our example be implemented by rearranging the grammar that is used to represent the phenotype. There are basically two such rearrangements: inserting a new production by encapsulating substrings of previous productions, and dissolving an existing production by expressing it within the grammar. Both of these operations can be made neutral, i.e., the phenotype the grammar is representing is not changed. But clearly, the resulting phenotypic variability changes as shown in the table above. We call such operations 2nd-type mutations, they correspond to neutral reorganizations of the genetic system, which prefectly complies with Kimura's Kimura (1986) point of view: "neutral mutations can be the raw material for adaptive evolution". There are several biological mechanisms that one might speculate to correspond to these neutral reorganizations of the genetic system; translocations, deletions, inversions, or transposons (or jumping genes) are candidates.

Some results of the experiments (taken from (Toussaint, 2003a)) are given in Fig. 2, which displays the best plant at certain generations in a single trial. The illustrations of the plants make clear that very specific search strategies have been developed during this evolution. The plants are very structured; it is implausible that such regular structures could have been found by independent phenotypic variations, as a direct encoding would require it. And also the size of the phenotypic solutions (the last plant corresponds to a phenotype string of length 9483) prohibits a search method without organizing a specifically structured search strategy and the ability to learn such strategies. In the experiments, such search strategies were developed by the evolution of an appropriate genetic representation of these structures.

The population was initialized with direct encodings of small random phenotypes. For a rather long time (almost half of the total evolutionary trial) the phenotypic variability remained unstructured and lead to small innovations and simple structures as the plant in generation 950. At around generation 1000, some interplay between the previously developed N- and F-productions (the genes N:IJFFFFFFFand F:IIAJJF in generation 1000) start to encode a repeated fan-like structure. Only 10 generations later, a few simple uncorrelated mutations on this representation have lead to a large phenotypic improvement exploiting the concept of the fan (the new duplications in the N-production generate more spokes (a complete fan), the single mutation in the F-production makes the spokes longer). Now that this fan concept has been discovered, it is further exploited. In generation 1650, the spokes have been further elongated and augmented with additional leaves at each segment (see the F-production). Note also that one N- and two B-

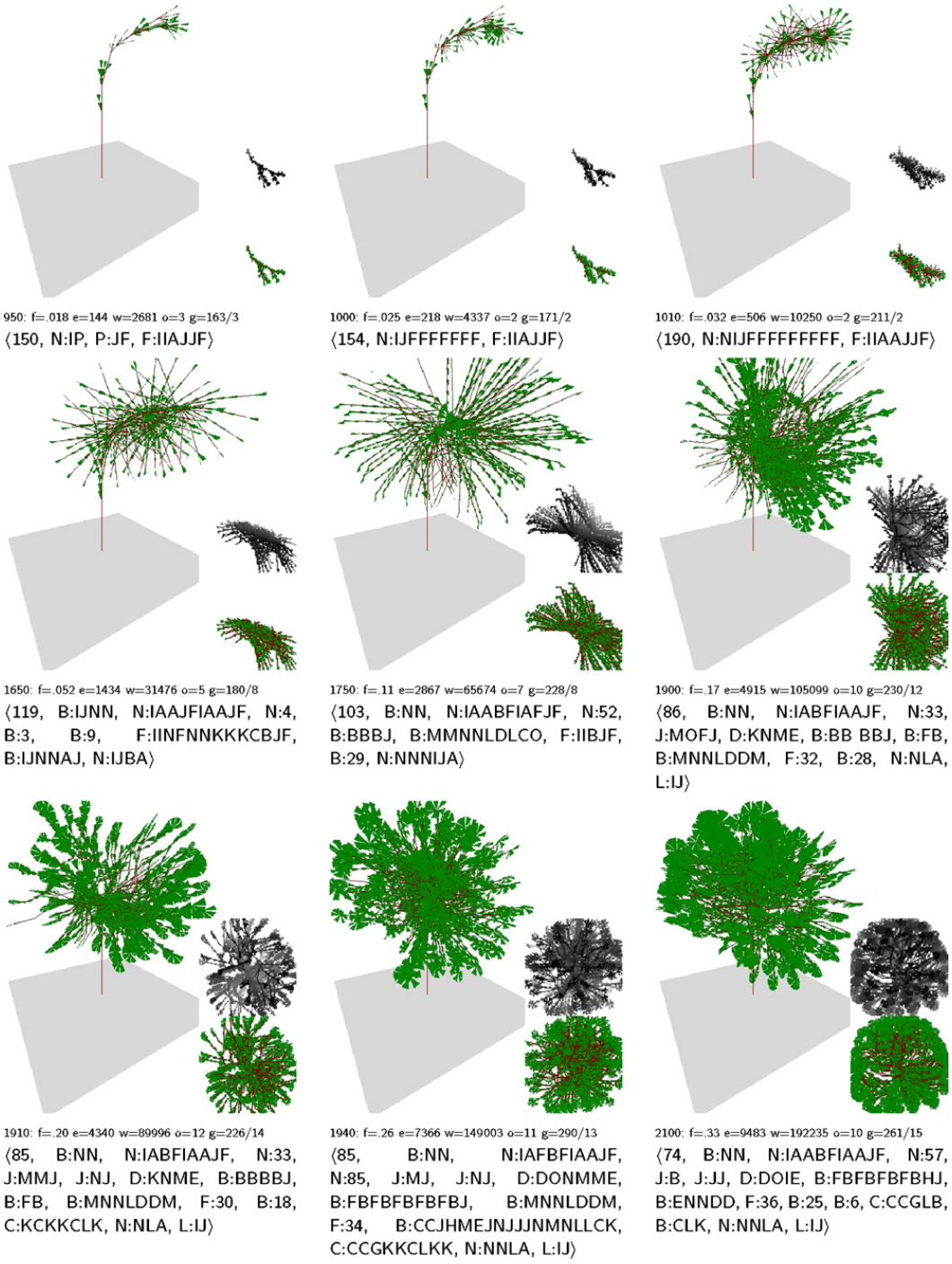

Fig. 2. A trial on a plant problem demonstrating the evolution of generative genetic representations. The best phenotypes at selected generations are displayed. The two squared pictures embedded in the lower right corner of each illustration display bird's view (and depth) perspectives that are used to calculate the plant's fitness. Below each illustration some data is given in the following syntax: ⟨generation⟩ : $f =$ ⟨fitness⟩ $e =$ ⟨number of elements⟩ $w =$ ⟨plant's total weight⟩ $o =$ ⟨number of used productions⟩ $g =$ ⟨total genome size⟩ / ⟨number of productions in the genome⟩. Below that, the genotype is given. Whenever a number instead of a symbol sequence is indicated, this number represents the length of the omitted sequence, which is too long to be displayed here. See the text for a discussion of this evolution.

productions are not expressed. In generation 1900 the plant's weight is already 105,099 and probably prohibits to elongate the spokes any further (since weight would increase exponentially). Instead a new concept develops: At the tip of each arm two leaves are now attached instead of one and this quickly evolves until there are three leaves, in generation 1910, and eventually a complete fan of six leaves attached at the tip of each arm. In generation 2100, a comparably short genome with 10 used productions encodes a very dense phenotypic struc-

ture of 9483 elements. In Toussaint (2003a) more such experiments can be found.

These experiments demonstrate the following:

- Evolution can adapt genetic representations in order to induce complex phenotypic variability. This adaptation is driven by neutral reorganizations of the genome.
- Significant phenotypic innovations (steps in fitness improvement) coincide with significant neutral reorganizations of the genome.
- In the experiments, evolution adapts genetic representations towards shorter and more modular genomes—which increases robustness in terms of phenotypic mutability but also has deeper implications in terms of structuring the phenotypic search distribution (see next section).
- Neutrality is not redundant in the sense of superfluous since different genetic representations within a neutral set induce different phenotypic variability. Hence, neutral traits carry information measurable in terms of phenotypic search distributions.

Similar genetic algorithms based on generative representations have been discussed before in the context of grammatical or L-system-type representations (Hornby and Pollack, 2001a, b; Hornby, 2003; de Jong, 2003) or so-called *symbiotic composition*(Watson and Pollack, 2002) to evolve complex phenotypic structures. Most of these interesting genotype–phenotype mappings are highly non-injective and the neutral sets $[x]$ comprise a large variety of genotypes inducing different phenotypic variabilities. However, their design was not based on the explicit idea of the evolution of genetic representations and none of these designs explicitly assures the mutational connectivity of these neutral sets (2nd-type mutations). In in the context of Genetic Programming, neutral reorganizations have been discussed in terms of automatically defined functions (ADFs, Koza, 1994) or module acquisition (Angeline and Pollack, 1993), which introduce very constructive ways of switching between different possible genetic representations of the same phenotype. For instance, modules of the expression can be encapsulated and re-used at several positions in the expression (see also Rosca and Ballard, 1994).

2.4. Estimation-of-distribution algorithms for computing representations

In the previous section we demonstrated evolutionary self-adaptation of the genetic representation in favor of a structured and adapted phenotypie variability $\bar{\sigma}$, which can be viewed as a search distribution in the actual search space P. Here we want to reason how a stochastic search algorithm can explicitly construct a genetic representations on the fly to induce a suitable search distribution. The idea is that during search the algorithm learns about the problem by analyzing the evaluations of previous search points and this knowledge can be used to design a new search distribution—or design a new genetic representation. The easiest heuristic in this direction would be to continue search in regions of the search space that have been successful previously (presuming a meaningful topology on the search space). But more interesting, and more important for our discussion, is a heuristic that tries to learn about the structure of the problem, i.e., about dependencies between parts of previously successful solutions and continues to search for solutions that obey the same kinds of dependencies. These kinds of algorithms learn a dependency structure in the search distribution.

Such algorithms are subsumed under the name of estimation-of-distribution algorithms (EDAs; Baluja and Davies, 1997; Mühlenbein et al., 1999; Pelikan et al., 1999, 2000). Given a search space P, a simple EDA is the following:

(1) Initialize a population $q = \{x_1, \ldots, x_\lambda\} \subset P$ of search points.
(2) Select the μ best from q, store them in a population p, and associate probabilities $1/\mu$ to each of them.
(3) Compute a model $q^* \in Q$ for this distribution p. Usually one that minimizes a cost function, e.g., $q^* = \underset{q \in Q}{\operatorname{argmin}}\ D(p||q)$. Here, Q is a class of feasible search distributions.
(4) Take λ samples $q \leftarrow \{x_1, \ldots, x_\lambda\}$ from q^*; and repeat from step 2.

Step 3 in the algorithm is the estimation-of-distribution step. The cost function measures the difference between two distributions and could for instance be the Kullback–Leibler divergence. It is in this step when the dependencies between variables of selected solutions (in p) are analyzed by "fitting" a probabilistic model q^* to it, which is then taken as the new search distribution.

In general, the crucial parameter of such an EDA is the choice of the class Q of feasible search distributions. On the one hand, the choice of Q determines the computational cost of the minimization (3) in every step. On the other hand, it determines the algorithm's capability to express and exploit the structure observable in p. Many

EDAs with different choices of Q have been presented: MIMIC (de Bonet et al., 1997) chooses Q to be the set of Markov chains, in PBIL (Baluja, 1994) Q is the set of factored distributions. Also the set of dependency trees (COMIT, Baluja and Davies, 1997), Bayesian networks (BOA, Pelikan et al., 2000), or Bayesian networks with local structures (hBOA, Pelikan and Goldberg, 2003) were considers as distribution classes Q.

In principle, these algorithms are able to detect necessary structure in good solutions and use this knowledge for further search. To draw the connection to the previous section we would like to discuss a recent approach to EDAs which slightly differs from the standard algorithms sketched above.

Given the discussion in the previous section it is clear that one can in principle design a search distribution by choosing a suitable representation of the search space, even when the basic distribution (mutational variability) on the coding space is constrained or pre-determined by elementary mutation. This is the idea of indirect EDAs (Toussaint, 2006). Instead of directly fitting a new search distribution $q*$ to the selected solutions q, one tries to find a representations $\phi : G \to P$ and a simple distribution $\tilde{q}^*$ on the coding space G such that $q^* = \tilde{q}^* \circ \phi^{-1}$ is a good model of p. This should be seen in analogy to the previous section: the distribution $\tilde{q}^*$ on the coding space G corresponds to the natural genetic variability σ while the induced distribution $q^* = \tilde{q}^* \circ \phi^{-1}$ on the search space P corresponds to the induced functional variability $\bar{\sigma}$ for a given representation ϕ.

It turns out that one can derive fundamental theoretical results on what suitable representations ϕ are in order to model selected solutions p (i.e., minimize the KLD $(Dp \,\|q^*)$) and implicitly capture the dependencies between solution variables in the way of representation. In Toussaint (2006) we derive two strongly related results: Given that the search distribution $\tilde{q}^*$ on the coding space must factorize (which is a model of independent genetic mutational variability),

(i) compact representations, in which previously selected solutions can be described with minimal description length, are optimal (w.r.t. minimizing $(Dp \,\|q^*)$);
(ii) factored representations, in which the distribution of previously selected solutions (when projected back on the coding space) has minimal mutual information between variables, are optimal.

Such compact representations are a special case of general factored ones. Note that finding such a factored representations is actually a variant of the independent component analysis problem—it basically means to find a problem decomposition.

These results give a theoretically grounded definition of what good genetic representations really are and they can be combined with the previously discussed results on the implicit selection of genetic representations in neutral sets. Given that genetic representations are implicitly selected based on the quality of the search distribution they induce we can follow that this evolutionary process aims towards compact and factored representations. This hypothesis has a potential to explain the driving force that lead to the evolution of the natural genetic language, its compactness and its modularity.

More concretely, considering the plant experiments described in the previous section, one can interpret the developed grammar-based representations as compressions of the phenotype. Indeed, in Toussaint (2006) we proposed a Compression EDA which basically replaces the estimation-of-distribution step 3 in algorithm 2.4 by computing a compact representation and a simple factored estimation on the coding space. Tests using a grammar-based compression technique indeed lead to hierarchical, compact representations of solutions similar to the ones observed for the plant evolution.

3. Simple adaptation on suitable representations

The previous discussions implicitly addressed the trade-off between Complex adaptation on arbitrary representations versus simple adaptation on suitable representations. Standard estimation-of-distribution algorithms (EDA) are a good example for learning complex search distributions on arbitrary representations. But we have seen that EDAs also provide us with a framework for the dual approach: learning a suitable representation of solutions and keeping the search distribution simple on that representation. With this background it becomes clearer how evolution is able to search for such complex organisms based on only simple mutation as the basic search mechanisms.

The theme "complex adaptation on arbitrary representations versus simple adaptation on suitable representations" transfers also to other fields. In machine learning and neural information processing models, latent variable models could been viewed as a general framework for such issues: latent variables are usually introduced to explain patterns in observed data. That is, instead of describing these patterns directly on the observed variables one invents latent variables to explain these patterns "away"; on the latent variable level there are less patterns left and data description is easier.

One could argue that a core difference between current technical approaches to problem solving and the natural paradigms is within this trade-off: technical approaches tend towards "complex adaptation on arbitrary representations" when developing efficient adaptation and learning algorithms whereas nature rather takes the "simple adaptation on suitable representations" approach. The apparent success of simple mutations in evolution and arguably simple basic adaptation mechanisms in the brain support this view. More emphasis needs to be spend on problems like learning representations, changing or switching back and forth between representations, inventing new representations—while keeping the processes that operate on these representations simple. In other terms: while many traditional approaches "fix the data format and develop complex processes thereon" the alternative approach is to "fix the basic processes but learn suitable data representations".

Another general aspect we can extract from the previous sections is the relation between the representation and structure of systems. The evolutionary paradigm illustrates well that using a certain representation of systems for search (the genetic language) has strong implications on what kind of systems are actually found, specifically, that their architectures are are highly structured in terms of hierarchies, modularity, self-similarity, etc. Note that this architectural notion of structure is not exactly the same as the structure in a distribution (the phenotypic variability $\bar{\sigma}$ or the selection F) that we previously discussed. The first is hard to define in general, while the latter can well be captured in terms of information theory. Still, this allows us to reason what the role of architectural system structure actually is. Clearly, for a given functionality there exist many systems, of structured as well as unstructured architectures, that could realize this functionality. As an example consider all possible C-codes one could write to program a certain function or software. From that point of view, structure is not a necessity for a system to be functional. Rather, the system structure is important for (1) the chance that the system can be found in a search process, (2) the chance that the system can be further improved, and (3) the chance that the system can be adapted to new constraints. This point of view ties the notion of system structure even stronger to the notion of system representation: all of the points (1) to (3) really depend on the way the system is represented (the representation of system space) and its implications on the functional variability. The architectural structure of the system is then merely a consequence of searching for and finding the system in the used representation.

4. Extensions of the concept

Generally, the processes of system design and system adaptation may be perceived as trajectories in the space of systems. This space surely is prohibitively large to be searched exhaustively and reasonable search processes must organize proper priors on how to transverse this space. We introduced the system variability σ_g in the case of evolutionary processes as a model of mutation. The key role of variability is that it also determines the functional variability $\bar{\sigma}_g$ in the actual solution space, which implicitly describes the topology on that space, i.e., the possible transitions and trajectories of continuous system evolution or the possible paths to design a system. So far we thought of σ_g as basic genetic mutations. However, in general the distribution σ_g can capture various kinds of system variability, for instance:

- *Intrinsic stochasticity*. Systems may have an intrinsic variability when parts or parameters of the system may fail, mutate, or be otherwise stochastic.
- *Parametric adaptability*. Parameters of a system may underlie a fixed adaptational process, like a learning rule, which induces the system variability.
- *Limitedness of innovations*. A designers "creativity" to modify a system, e.g. to reassemble parts of a system to a new system, is surely limited, leading to a limited range of possible new systems from an old one—which can also be captured in the definition of a specific system variability.
- *Input signal perturbations*. In the case of information processing systems, the perturbation of input signals can alternatively be modeled as perturbations in the system's input device (as it is typically done in classical system design). These perturbations of the system input device can again be captured as a system variability.

Although in all of these cases the system variability σ_g has a different concrete meaning, the conceptual approach to investigate the implicated functional variability $\bar{\sigma}_g$ remains the same. In all cases, this variability determins the possible transitions and evolution in the solution space. We briefly want to discuss how this point of view allows us to capture the notion of robustness and flexibility.

Robustness. One possibility is that σ_g is meant to model intrinsic perturbations of the system (like malfunction of parts) or perturbations of input signals which in turn may been modeled as perturbations of the input device of a system. Also in this case, all of what we argued so far remains valid, but the interpretations

change. While previously we considered σ_g to model possible adaptational variations of a system, thought of as sparse, now σ_g might really be a signal noise distribution. If the noise were Gaussian and the system linear, then $\bar{\sigma}_g$ (the variability on P) could be described by a covariance matrix.

Generally though, perturbations of the system or of the input signals lead to much more complex structured functional variations $\bar{\sigma}_g$. Being robust can then be measured (similar as within EDAs) as the average quality $\langle f \rangle_{\bar{\sigma}_g}$ of the system under all possible perturbations (see Section 2.2). The only difference being that now $\bar{\sigma}_g$ describes the functional effects of these perturbations instead of the system adaptability. All the arguments concerning the structural match equally apply: essentially, a system is robust if the functional variations $\bar{\sigma}_g$ still obey the necessary dependencies between variables of the phenotype, i.e., when these perturbations do not destroy the interplay between the parts of the system.

Flexibility. In some cases, the problem itself, i.e., the objective function f might change over time. A desirable property of a system is then that it will be able to adapt quickly (or be adapted easily) to the new problem. Again, the phenotypic variability $\bar{\sigma}_g$ plays the crucial role. This time though, $\bar{\sigma}_g$ should match to the possible changes of a problem, not to F. More precisely, assume that x solves the current problem and is implemented by the current system g. Now there exists another quality function $h(x') = \sum_{f'} \mathcal{P}(f'),\ f'(x')$ over P that describes the expected quality of a phenotype x' when the problem f changes stochastically to a new problem f' with probability $\mathcal{P}(f')$. The flexibility of a system can then be defined as the probability whether g (currently implementing x) will adapt to the new problem and offer a new phenotype x' with high quality. Again, this gives the same basic term as before, $\sum_{x'} \bar{\sigma}_g(x') h(x')$, now though measuring the match between $\bar{\sigma}_g$ and h instead of f. All arguments concerning the structural components of this term again apply equally.

5. Adaptation on structured systems

So far we have addressed the search or adaptation of systems g which implement a single fixed functionality or phenotype $\phi(g)$. In this section we more specifically consider $\phi(g)$ itself to be an adaptive system, the quality of which is determined by some developmental or learning process. A good example for such life-time adaptability is the evolution of brains. Considering the phenotypic variability in the individual's phase of development, sensory driven alterations of e.g. cortical maps determine the final functionality of cortical fields dependent on the environment. That means, that "not only has life evolved, but life has evolved to evolve" (Krubitzer and Kaas, 2005).

The strategy to adapt to environmental constraints is not restricted to the period of individual development but, in an altered form, a life-long ability. The acquired "knowledge" in a structure gives an additional basis for flexibility on a short time scale in a particular environment. The crucial problem for using the "experience" is its representation. That means the knowledge has to be structured dependent on the structure of the system and coded in an appropriate way. In cortices, associative memories combined in a feedback loop with the different analyzing parts of the system seems to be a promising model that assures correlations of possible solutions (Deco and Rolls, 2005). A general formal framework is not available. Subsequently we discuss some aspects of this knowledge-based flexibility. We see it on the background that it is too a part of the evolutionary strategy that allows individual adaptation, constrained by genetics on a molecular level. We restrict ourself to the representation of "knowledge" and the possibility to use it for further solutions. If one relates flexibility to the system's function it can be characterized by a knowledge-based "trajectory" on the phenotype space P from a solution x to another solution x'. This knowledge might be deduced from data and constraints on the desired solution.

Let us assume that there is a robot capable of several basic solutions like recognition of objects, a generic grasping, shifting of objects and the ability to handle screws. Let us further assume that the structure of the system is not changed anymore, but the robot learns (with an integrated working memory) to "construct" different devices using his basic abilities. After some time of experience the robot has capacities which we might consider as the "knowledge" of the system which can be used to find further "solutions". The problems associated with this purpose can be interpreted as defining a "language" with element and syntactic rules that cover the behavioral space. This task contains three open questions:

(i) How can the "experience" of the system be represented?
(ii) What type of goals can be solved and how can they be communicated to the system?
(iii) In which way can an unrestricted search be replaced by a goal-directed search?

Complete answers can't be given but there exist some relations to the different points of view in this work. The different abilities e.g. of our robot like grasping, shifting

etc. are considered to be elements on a solution space P which contains all possible results that could be achieved by parameter variations on a given structure. Some of the "results" x' can be considered as solutions of other meaningful problems. To organize a trajectory from x to x' presupposes an "order" or a metric on this solution space in order to avoid an extensive search. It seems meaningful to use the correlations of the phenotypes (solutions) as a basis. An example can be seen in the structure of the mammalian cortices where different associative memories as learning matrices are assigned to the analyzing subsystems. Associative memories (AM) attach an input vector $\boldsymbol{e}$ to a meaning b, respectively a matrix C to a vector b'. Using b' as an input for the next learning matrix one can produce b'' and so on. The association of $\boldsymbol{e}$ to b_j relies on the degree of correlation. The steps from $\boldsymbol{e}$ to $b \rightarrow b' \rightarrow b''$ indicate an increasing level of abstraction. After learning one can use the structure in both directions $\boldsymbol{e} \rightarrow b$ and $b \rightarrow \boldsymbol{e}$ (Kohonen, 1980; Steinbuch, 1961). This anticipation of a structure (AM), which determines the representation seems to be a result of evolution (Deco and Rolls, 2005). This AM organizes a correlation-based metric that allows to combine solutions, it favors a transition from p to p' and allows to generalize the solution against disturbances. It is obvious that the chosen system represents a special case which supplies the solution space with a structure, a metric, and a correlation-based search strategy.

The goals of such systems can be different. In our robot example one has to combine solutions (e.g. detection, grasping, transportation). The sequences can be altered if e.g. grasping is not possible and shifting helps to do it. Disturbances can be reduced and alternative trajectories can be found but it remains open how to find new solutions that presupposes the usage of correlations that have not been learned before and have to be constructed autonomously by the system, e.g. by projecting the unsolved problem onto already solved problems in order to find similarities.

Correlations of the phenotype represent an essential ingredient of the way to decompose the problems in evolution (Section 2). In neural systems the structural equivalence is realized by an overlapping coupling in different types of neural nets. Experiments show that the structure of the networks especially for vision tasks allow a monotonous and definable trade-off between accuracy and safety (robustness) of the solutions (graceful degradation). The structural basis is the overlapping coupling of the elements that provide correlations and statistical expectations. This overlapping coupled elements (+, −) realized in the genotype–phenotype mapping as well as in neural nets seems to be a general principle in biological systems. This simple format of the processes is normally kept constant whereas the data format changes in space and time. Whatever this scheme of overlapping coupling does in different stages of the information processing, like filtering, decision making, combination or separation, it always assures correlations of the solutions. Their usage in the solution space has therefore a least a broad structural basis that allows "functional" correlation on a levels of achievable abstractions. How to do this systematically remains open, some special solutions are available.

As shown in Section 2 the evolutionary process adapts to the structure of the problem, flexibility, modularity and robustness can get a structural basis. This process can be extended to the developmental period of an individual so that the environment can influence the structure but under genetic control. The ability to "learn" in an environment remains a life-long possibility and creates a flexible structure in the form of integrated memories. This specific integration into the process to be realized assures a functional flexibility on the basis of the stored "knowledge". An appropriate order of this knowledge allows the flexibility to be a permanent program. The preposition for this flexibility is an appropriately evolved structure, the basic strategies in evolution, during the individual development, and in individual lifetime show great similarities in the different processes.

6. Conclusions: and how do we design complex systems?

Every design, search, or adaptation process is limited and characterized by the possible adaptation steps on the coding space. Such adaptation might be driven by fixed and simple mechanisms of learning, mutation or variation, or limited resource innovation. Despite these limitations, this does not mean that the functional adaptability cannot be complex structured. The view taken in this paper is that for every functional solution there exists a multitude of possibilities to represent this solution—and this can be exploited. Complex functional variability (complex structured search distributions) can be induced by the right choice of representation. Rather than relying on a static representation where increasing problem complexity normally amounts to growing computational resources, biological systems seem to flexibly adapt representations, thereby reformulating the problem, such that the same basic mechanisms of search and adaptation may find a solution.

One may argue that this merely transfers the problem of searching for a solution to the meta problem of searching for a representation that facilitates search. However,

we discussed rather concrete strategies to design representations and thereby allow for the design of structured adaptability. We want to conclude with summarizing such strategies:

- *Neutral reorganizations*. A simple strategy to exploit the existence of many possible representations is to allow for neutral reorganizations of the system. To give some illustrative examples: Reorganize the source code of a program without changing the software's functionality. Reorganize the tax laws without changing anyone's net loss or benefit in a society. Reorganize the genome without changing the organism's phenotypic properties. While naively such reorganizations seem redundant, they are not. The possibility for further structured adaptation crucially depends on such reorganizations. In the case of evolution, where there is a selectional preassure implicitly also on the probability of innovation of a species, only well-structured systems will survive.
- *Compact and factored representations*. Adaptation on many variables is simple when these can be adapted independently. The theory on EDAs and Minimum Description Length shows that such representations can be provided by compact representations, where there are few dependencies on the coding variables but adapted structured dependencies between the functional variables of the system. Roughly, such representations are found by recoding system properties that one has learnt, by experience, to be dependent.
- *Modularity*. A modular encoding where a single conding symbol is expressed to a whole system part in the decoding process is a means for (1) compactness and (2) inducing complex functional variability. On the coding space, a single variation allows to reproduce the same module multiple times and to reuse this module in other contexts.

References

Altenberg, L., 1994. Evolving better representations through selective genome growth. In: Proceedings of the IEEE World Congress on Computational Intelligence, pp. 182–187.

Altenberg, L., 1995. Genome growth and the evolution of the genotype–phenotype map. In: Banzhaf, W., Eeckman, F.H. (Eds.), Evolution and Biocomputation: Computational Models of Evolution. Springer, Berlin, pp. 205–259.

Amari, S., 2001. Information geometry on hierarchy of probability distributions. IEEE Trans. Inform. Theory 47, 1701–1711.

Angeline, P.J., Pollack, J.B., 1993. Evolutionary module acquisition. In: Proceedings of the Second Evolutionary Programming Meeting, pp. 154–163.

Baluja, S., 1994. Population-based incremental learning: a method for integrating genetic search based function optimization and competitive learning, Tech. Re CMU-CS-94–163. Comp. Sci. Dep., Carnegie Mellon U.

Baluja, S., Davies, S., 1997. Using optimal dependency-trees for combinatorial optimization: Learning the structure of the search space. In: Proceedings of the 14th International Conference on Machine Learning (ICML 1997), pp. 30–38.

Chenn, A., Walsh, C.A., 2002. Regulation of cerebral cortical size by control of cell cycle exit in neural precursors. Science 297, 365–369.

Cover, T.A., Thomas, J.A., 1991. Information Theory. John Wiley & Sons, New York.

de Bonet, J.S., Isbell, C.L., Viola Jr., P., 1997. MIMIC: finding optima by estimating probability densities. In: Mozer, M.C., Jordan, M.I., Petsche, T. (Eds.), Advances in Neural Information Processing Systems, vol. 9. The MIT Press, p. 424.

de Jong, E.D., 2003. Representation development from Pareto-Coevolution. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2003), pp. 265–276.

Deco, G., Rolls, E.T., 2005. Attention, short-term memory, and action selecetion: a unifying theory. Prog. Neurobiol. 76, 236–256.

Fontana, W., Schuster, P., 1998. Continuity in evolution: on the nature of transitions. Science 280, 1431–1433.

Halder, G., Callaerts, P., Gehring, W., 1995. Induction of ectopic eyes by targeted expression of the eyeless gene in Drosophila. Science 267, 1788–1792.

Hansen, N., Ostermeier, A., 2001. Completely derandomized self-adaption in evolutionary strategies. Evol. Comput. 9, 159–195.

Hansen, T.F., Wagner, G.P., 2001a. Epistasis and the mutation load: a measurement-theoretical approach. Genetics 158, 477–485.

Hansen, T.F., Wagner, G.P., 2001b. Modeling genetic architecture: a multilinear model of gene interaction. Theor. Popul. Biol. 59, 61–86.

Hornby, G.S., 2003. Generative representations for evolutionary design automation, Ph.D. Dissertation, Brandeis University Dept. of Computer Science.

Hornby, G.S., Pollack, J.B., 2001a. The advantages of generative grammatical encodings for physical design. In: Proceedings of the 2001 Congress on Evolutionary Computation (CEC 2001). IEEE Press, pp. 600–607.

Hornby, G.S., Pollack, J.B., 2001b. Evolving L-systems to generate virtual creatures. Comput. Graph. 25, 1041–1048.

Kimura, M., 1986. DNA and the neutral theory, philosophical transactions. Roy. Soc. London B312, 343–354.

Kohonen, T., 1980. Self-Organisation and Associative Memory. Springer, Berlin.

Koza, J.R., 1994. Genetic Programming II: Automatic Discovery of Reusable Programs. MIT Press, Cambridge Massachusetts.

Krubitzer, L., Kaas, J., 2005. The evolution of the neocortex: how is phenotypic diversity generated? Curr. Opin. Neurobiol. 15, 444–453.

Mahner, M., Kray, M., 1997. What exactly are genoes, genotypes and phenotypes? And what about phenomes? J. Theor. Biol. 186, 55–63.

Mühlenbein, H., Mahnig, T., Rodriguez, A.O., 1999. Schemata, distributions and graphical models in evolutionary optimization. J. Heuristics 5, 215–247.

Nimwegen, E., Crutchfield, J.P., 1999. Neutral evolution of mutational robustness, Santa Fe Working Paper 99–07-041.

Nordin, P., Banzhaf, W., 1995. Complexity compression and evolution. In: Eshelman, L. (Ed.), Genetic Algorithms: Proceedings of the Sixth International Conference (ICGA 1995). Morgan Kaufmann, Pittsburgh, pp. 310–317, 15–19.

Pelikan, M., Goldberg, D.E., 2003. Hierarchical BOA solves Ising spin glasses and MAXSAT. In: Proceedings of the Genetic and Evolutionary Computation Conference, 2003 (GECCO 2003). Springer, pp. 1271–1282.

Pelikan, M., Goldberg, D.E., Cantú-Paz, E., 2000. Linkage problem, distribution estimation, and Bayesian networks. Evol. Comput. 9, 311–340.

Pelikan, M., Goldberg, D.E., Lobo, F., 1999. A survey of optimization by building and using probabilistic models, Tech. Rep. IlliGAL-99018, Illinois Genetic Algorithms Laboratory.

Prusinkiewicz, P., Hanan, J., 1989. Lindenmayer Systems, Fractals, and Plants, Vol. 79 of Lecture Notes in Biomathematics. Springer, New York.

Prusinkiewicz, P., Lindenmayer, A., 1990. The Algorithmic Beauty of Plants. Springer, New York.

Rice, S.H., 1998. The evolution of canalization and the breaking of von Bear's laws: Modeling the evolution of development with epistatis. Evolution 52, 647–656.

Rice, S.H., 2000. The evolution of developmental interactions: Epistasis, canalization, and integration. In: Wolf, J.E.B., Wade III, M. (Eds.), Epistasis and the Evolutionary Process. Oxford University Press, New York, pp. 82–98.

Riedl, R., 1977. A systems-analytical approach to macro-evolutionary phenomena. Quart. Rev. Biol. 52, 351–370.

Rosca, J.P., Ballard, D.H., 1994. Genetic programming with adaptive representations, Tech. Rep. TR 489, Computer Science Department, University of Rochester, NY.

Schuster, P., Fontana, W., Stadler, P.F., Hofacker, I.L., 1994. From sequences to shapes and back: a case study in RNA secondary structures. Proc. Roy. Soc. B255, 279–284.

Steinbuch, K., 1961. Die Lernmatrix. Kybernetik 1, 36–45.

Stephens, C.R., Vargas, J.M., 2000. Effective fitness as an alternative paradigm for evolutionary computation I: general formalism. Genet. Prog. Evol. Mach. 1, 363–378.

Toussaint, M., 2003a. Demonstrating the evolution of complex genetic representations: an evolution of artificial plants. In: Proceedings of the Genetic and Evolutionary Computation Conference. (GECCO 2003), pp. 86–97.

Toussaint, M., 2003b. On the evolution of phenotypic exploration distributions. In: Cotta, C., De Jong, K., Poli, R., Rowe, J. (Eds.), Foundations of Genetic Algorithms vol. 7. Morgan Kaufmann, pp. 169–182.

Toussaint, M., 2005. Factorial representations to generate arbitrary search distributions. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2005); Theory of Representations workshop. ACM Press, Washington, D.C., USA, pp. 339–345.

Toussaint, M., 2006. Compact representations as a search strategy: compression EDAs. Theor. Comput. Sci. 361, 57–71.

Wagner, A., 1996. Does evolutionary plasticity evolve? Evolution 50, 1008–1023.

Wagner, G.P., Altenberg, L., 1996. Complex adaptations and the evolution of evolvability. Evolution 50, 967–976.

Wagner, G.P., Booth, G., Bagheri-Chaichian, H., 1997. A population genetic theory of canalization. Evolution 51, 329–347.

Wagner, G.P., Laubichler, M.D., Bagheri-Chaichian, H., 1998. Genetic measurement theory of epistatic effects. Genetica 102/103, 569–580.

Wagner, G.P., Mezey, J., 2000. Modeling the evolution of genetic architecture: a contimuum of alleles model with pairwise A $\times$ A epistatis. J. Theor. Biol. 203, 163–175.

Watson, R.A., Pollack, J.B., 2002. A computational model of symbiotic composition in evolutionary transitions, Biosystems. Special Issue Evolv. 69, 187–209.

Wilke, C.O., 2001. Adaptive evolution on neutral networks. Bull. Math. Biol. 63, 715–730.