Künstliche Intelligenz **1/2000** (2000): 26-32

LindEvol: Artificial Models for Natural Plant Evolution

Jan T. Kim

LindEvol is a family of computer simulation programs modelling the evolution of plants. Several mechanisms that shape plant evolution are integrated in LindEvol models. LindEvol models have been used to investigate evolution of structured taxonomic diversity. The concept of linking mutation rate modification to an energy cost was used as a starting point for developing a method for avoiding premature convergence in evolutionary algorithms. Finally, LindEvol has been used for a comparative analysis and characterization of methods for measuring biodiversity.

1 Introduction

Artificial Life has been defined as "a field of study devoted to understanding life by attempting to abstract the fundamental dynamical principles underlying biological phenomena, and recreating these dynamics in other physical media – such as computers – making them accessible to new kinds of experimental manipulation and testing." [Langton et al., 1992, preface]. One of these fundamental processes is evolution, which is a major focus of Artificial Life research. Morphogenesis is another central dynamical process which is linked to evolution by genetics, as the developmental process emerges through the biological interpretation of genetic information. At the molecular level, this interpretation is mediated by transcription factors. These are proteins that specifically bind to sequence motifs that are usually located in the upstream part of genes. By binding to these motifs, transcription factors activate or repress the transcription of their target genes. Transcription factors are themselves encoded by genes, and thus they form regulatory networks which constitute core systems for the biological interpretation of genetic information [Meyerowitz, 1994, Theißen and Saedler, 1995].

The LindEvol modelling system was developed to represent the evolution of plant morphogenesis processes. The name "LindEvol" is derived from "Lindenmayer systems" [Prusinkiewicz and Lindenmayer, 1990] and "evolution".

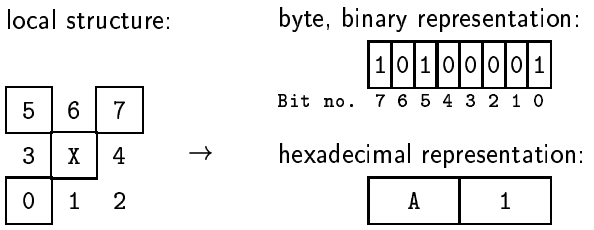


Figure 1: The scheme for indexing the local neighbourhood of a cell and the application of this scheme for mapping a local plant structure in a nine cell neighbourhood to a byte value. Boxes indicate cells belonging to one plant. Bits corresponding to sites occupied by cells of the plant are set to 1, all others are set to 0.

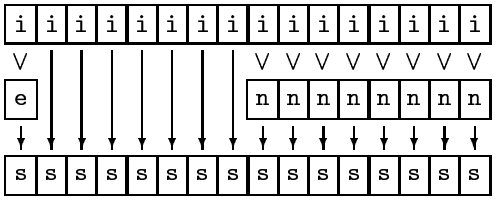


Figure 2: Computation of a sixteen bit cell state. Bits of the internal state are labelled `i`, the energy bit is labelled `e`, neighbourhood state bits are labelled `n` and the bits of the resulting cell state are labelled `s`. ∨ denotes the or operator.

2 Description of LindEvol

2.1 Modelling of plant phenotypes

Plants in LindEvol grow in a spatially extended environment which is modelled by a two-dimensional, orthogonal lattice. Vertically, the lattice has borders, called *ceiling* and *floor*. Horizontally, the lattice is circular. Plants are modelled as collections of contiguous cells. All plants of a population grow on the same lattice.

Energy enters the system as light units called *photons*. Photons are introduced into the lattice at the top row. From its introduction site, a photon travels vertically downward. If it encounters a cell, it is absorbed with a probability of 50%. Cells have a binary energy state, they are either energyless or energy-rich, an energyless cell that absorbs a photon becomes energy-rich.

Each *plant cell* has a state which consists of eight or sixteen bits. The neighbourhood state of a cell is defined by the pattern of cells that are located within the cell's nine-cell neighbourhood and that belong to the same plant. Fig. 1 shows the mapping of cell patterns to a byte value which is used to specify the neighbourhood state of a cell. In LindEvol-GA (see section 2.3.2), the cell state is eight bits wide and defined to be the

neighbourhood state.

In LindEvol-P (see section 2.3.2), the cell state is sixteen bits wide, and it is determined by the internal state consisting of sixteen bits (see section 2.2.1), the neighbourhood state and the energy state, as depicted in Fig. 2.

Time proceeds in discrete steps. A basic time step, called a *day*, consists of the simulation of light by running one photon through each column of lattice sites, followed by the simulation of plant growth. The genetic control of the growth process is described in the following subsection.

2.2 Modelling of plant genomes and development

2.2.1 Cell actions

Plant development takes place by *actions* performed by plant cells. The actions are:

- `divide` *n*: A new cell is produced at the position *n* with respect to the acting cell. *n* denotes a neighbouring site of the mother cell according to the indexing scheme shown in Fig. 1.
- `flyingseed`, `localseed`: A new plant is generated on the lattice floor, either at a randomly chosen unoccupied site (`flyingseed`), or as closely as possible to the x-coordinate of the acting cell (`localseed`).
- `mut-`, `mut+`: Each plant has a mutation modification exponent, denoted by μ (see section 2.2.6). These actions decrement and increment μ.
- `statebit` *n*: The *n*-th bit in the cell's internal state is set to 1 in the subsequent time step.

All actions except `statebit` consume energy. Actions are encoded by action codes, which are six- or eightbit integer numbers. Action codes are extracted from genomes by genome interpreters (see below) and mapped to actions using a fixed lookup table.

2.2.2 Genome decoding basics

Genomes are modelled by strings of bytes which are processed by a system called genome interpreter (see sections 2.2.3 and 2.2.4). The genome interpreter determines substrings of the genome that constitute a *gene*. Each gene is decoded into one rule. A rule maps a set of cell states, specified by a state mask, to an action. A state mask is a bit mask consisting of the symbols 0, 1 and *, where the asterisk denotes "don't care". Rules are written in the form:

```
cell state mask -> action
```

In a given cell, all rules in which the state mask matches the cell state are activated, but only the first energy consuming action actually has a phenotypic effect, since performing such an action invariably renders the cell energyless. The genes that encode rules which are activated during the life of a plant are collectively referred to as the *developmental program* of that plant.

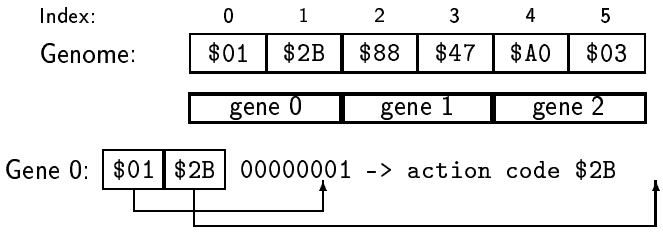


Figure 3: An example genome and its decoding by the block oriented genome interpreter. Hexadecimal values are prefixed with $.

2.2.3 Block oriented genome interpreter

For the block oriented genome interpreter, a gene consists of a block of two bytes length. At each even-numbered position in the genome (byte indexing starts with 0), a gene begins. The first byte determines the lefthand part of the rule encoded by the genome, it specifies exactly one cell state. The second byte specifies the eightbit action code for the righthand part of the rule. Fig. 3 illustrates how block oriented genome interpretation works.

2.2.4 Promoter oriented genome interpreter

The promoter oriented genome interpreter has been designed to capture the label-based structure of molecular genes which makes them largely independent of their positions and thus robust to frameshift effects [Ray, 1992]. The promoter oriented genome interpreter (see Fig. 4) distinguishes three types of bytes values. Bytes in which the most significant bit (bit number 7) is set are interpreted as *promoters*, bytes in which bit number 6 is set are interpreted as *terminators*, and bytes in which neither bit is set are considered to be *operators*. A gene is a sequence beginning with a promoter, followed by zero or more operators and ending with a terminator. Sequences starting with a promoter and ending with another promoter or the end of the genome are called incomplete genes.

For translating a gene into a rule, the six least significant bits are extracted from the terminator and interpreted as a sixbit action code determining the righthand part of the rule. Incomplete genes do not specify any action. Such genes do not have any effect on the development of the phenotype.

Each operator specifies one bit in the cell state mask (i.e. the lefthand part of the rule). The four least significant bits determine which bit is to be specified. Bit number 5 gives the value (0 or 1) of the bit in the state mask. If there are several operators for the same state bit, the last one takes precedence over all others. Positions in the cell state mask for which no value is specified by an operator become wildcards.

2.2.5 Visualization of regulatory networks

Activation of genes can result in cell divisions. In this case, the newly produced cell alters the local structure of its neighbouring cells, and the neighbouring cells form the structure surrounding the new cell. Because the daughter cell is placed within the neighbourhood of the mother cell, it is possible to determine the

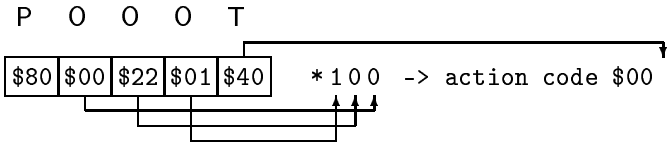


*Figure 4: Decoding of a gene by the promoter oriented genome interpreter. The promoter is labelled P, operators are labelled O and the terminator is labelled T. For clarity, only the least four significant bits of the state mask are shown (all twelve others are *).*

local structure surrounding the mother cell after division, and since the local structures of mother and daughter cell overlap, it is possible to partially determine the daughter cell's local environment (see [Kim, 1996b] details). The local structures of the cells involved determine which genes are activated in the next time step. Thus, by analyzing the local structures before and after division, it is, to a limited extent, possible to conclude which genes will be activated by the activity of a gene. These activation relations can be visualized, as shown in Fig. 5.

2.2.6 Mutation

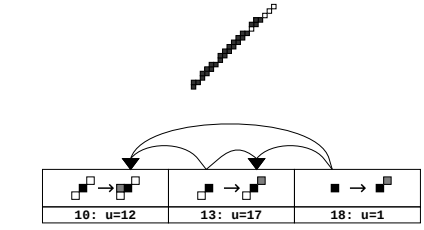
Mutations are modelled as random modifications of a genome. For LindEvol models, four different types of mutations are used:

- Point mutation. There are two types of point mutation. In a bytewise point mutation, a random value is written into the affected position in the genome. Bytewise point mutation is used in conjunction with block oriented genome interpretation. A bitwise point mutation is defined as the inversion of one randomly chosen bit in the affected byte; this is used with promoter oriented genome interpretation.
- Insertion. A block of random bytes is inserted. In models with block oriented genome interpretation, blocks of two bytes are inserted to avoid frameshifts. With the promoter oriented gene interpreter, which is much more robust to such frameshift effects, single bytes are inserted.
- Deletion. As with insertions, blocks of two bytes are deleted in models using block oriented genome interpretations, whereas single bytes are deleted in models using promoter oriented genome interpretation.
- Gene duplication. This type of mutation is only defined for models using promoter oriented genome interpretation. In a gene duplication, a copy of a stretch of the genome is appended to the genome. The copied stretch always starts with a promoter. The final byte which is copied is the next terminator, the next promoter, or the end of the genome, whichever is encountered first. This operation results in appending a copy of a gene or of an incomplete gene to the genome.

The control parameters for mutation are the global mutation rates for replacement, insertion, deletion and duplication, denoted by M_r, M_i, M_d and M_{dup}, respectively, and the mutation modification factor q. These are global parameters that remain constant during a simulation run, their allowed ranges are given by $0 \leq M_r, M_i, M_d, M_{dup} \leq 1$ and $q \geq 1$. The effective mutation rates, denoted by m_r, m_i, m_d and m_{dup}, determine the probability for a given mutation type to affect a byte in one mutation process. These are calculated according to the scheme

$$m = M \cdot q^{\mu} \tag{1}$$

where μ denotes the mutation modification exponent which can be incremented and decremented by energy consuming cell actions as described in section 2.2.1.




```
10: 10000001 -> divide 3
13: 00000001 -> divide 7
18: 00000000 -> divide 7
```

Figure 5: A plant and its developmental program listing (i.e. a listing of the rules encoded by the genome, showing only the genes which actually were activated during the growth process) and its graphical representation.

2.3 LindEvol models of evolution

2.3.1 LindEvol-GA

LindEvol-GA was designed to model the evolution of annual plants, i.e. plants which go through their entire life cycle within one vegetation period. A genetic algorithm [Goldberg, 1989] is used to model the process of evolution, according to a concept introduced in [Wilson, 1989]. Genomes are translated using the block oriented genome interpreter. Internal cell states are not used.

At the start of a simulation run, an initial population consisting of 50 randomly generated genomes that are 20 genes long is constructed. Fitness values for the genomes are computed by simulating a vegetation period. Initially, all plants consist of one single, energyless cell, called the germ cell. Germ cells are placed equidistantly on the floor of the lattice world, which is 150 sites wide and has a height of 30 sites. After 30 days, the number of energy-rich cells is determined for each plant and assigned as a fitness value to the corresponding plant.

After the computation of fitness values based on the simulation of a vegetation period, the population is subjected to selection and mutation. Selection is performed by replacing the genomes with the lowest fitness values with copies of genomes from the surviving part of the population. The proportion of genomes that are replaced in the selection step is determined by the selection rate s, which is a global control parameter. After selection, all genomes in the population are subjected to mutation as described in section 2.2.6, completing the computation of the population of the subsequent generation.

2.3.2 LindEvol-P

In LindEvol-P, promoter oriented genome interpretation and internal cell states are used. In contrast to LindEvol-GA, time steps are not grouped to vegetation periods.

Reproduction takes place by seed production instead of being done by an external schedule. Plant death is modelled probabilistically on the basis of properties of the plant, i.e. without any global ranking. The probability of a plant to die in a time step depends on its total number of cells n_c, its number of energy-rich cells n_e, and its leanover term L. The leanover term reflects the deviation of the plant's shape from being balanced around the germ cell, its values range from 0 (indicating perfect balance) to 1 (indicating strongest leanover).

The death probability of a plant per time step is calculated based on the global control parameters d, d_n, d_e and d_l, which stay constant in a simulation run:

$$P_d = d \cdot n_c^{d_n} \cdot (n_e + 1)^{d_e} + d_l \cdot L$$

In addition to this probabilistic death, plants may also die as a result from being attacked. An attack is the attempt of a plant to produce a cell on a position that is already occupied. In this case, the plant to which the already existing cell belongs dies with the probability

$$P_{kill} = \frac{1}{n_e}$$

where n_e denotes the number of energy rich cells in the plant that is being attacked.

3 Selected LindEvol results

3.1 Mutation rate adaptation in LindEvol-GA

The mutation rates and the selection rate in LindEvol-GA limit the length of developmental programs that can evolve. According to a formal analysis of this error threshold effect [Kim, 1996b, Kim, 1996a], the maximal length of developmental programs can be estimated from the selection rate and the replacement mutation rate by

$$r_{max} \approx \frac{\log(1-s)}{2\log(1-M_r)} \qquad (2)$$

in the case that $M_i = M_d = 0$. Developmental programs with a length greater than r_{max} are not evolutionarily stable, since irrespective of their fitness, the fraction of their offspring which inherits the developmental program undamaged by mutation is so small that the developmental program in question are inevitably diluted out of the population.

According to equation 2, $r_{max} \approx 1.7$ for $s = 0.5$ and $M_r = 0.18$. This means that even basic developmental programs cannot evolve with such a high mutation rate. However, this limitation of developmental complexity can be overcome as the mutation modification factor is set to 2.0 in the run shown in Fig. 6.

In this run, a marked evolutionary step is observed around generation 1150. Before this step, average developmental program length is below 2, and plants growing vertical or diagonal shoots only appear sporadically. After the step, active lowering of effecitve mutation rates evolves as indicated by a drop in average mutation exponent values. At the same time, average developmental program length rises well above the threshold of 1.7, and more complex phenotypes appear. While fitness increases too, it remains comparatively low because plants use energy for performing `mut-` actions.

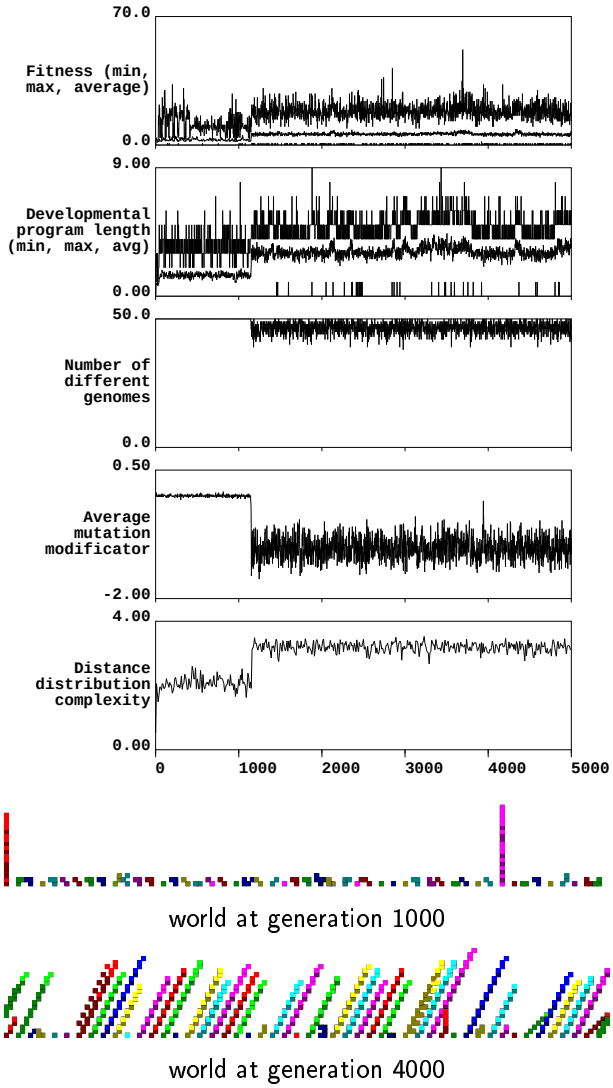


Figure 6: LindEvol-GA run with selection rate $s = 0.5$, mutation rates $M_r = 0.18, M_i = M_d = 0$, mutation modification factor $m_f = 2$.

3.2 LindEvol-P

LindEvol-P is much more complex than LindEvol-GA, and less accessible to formal analysis – e.g. as generations overlap and are variable in length, the error threshold analysis which has been carried out for LindEvol-GA cannot be applied to LindEvol-P. However, LindEvol-P provides opportunities for re-examining phenomena observed in LindEvol-GA in a more realistic simulation system.

Fig. 7 provides an impression of the variability of phenotypes

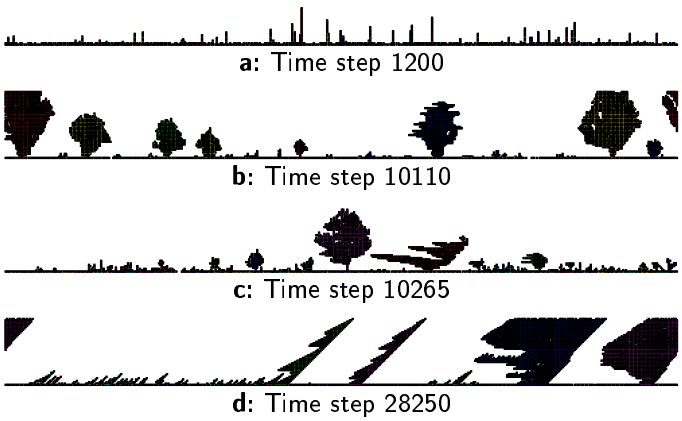


Figure 7: Lattice world pictures from a LindEvol-P run.

which can evolve in LindEvol-P simulations. Initially, the floor of the lattice world becomes covered with a unicellular plants. After 1000 time steps, the first vertically growing plants appear. The varying height of these plants reflects the different ages. Around time step 10000, a great phenotypic diversity of tree-like plants evolves within less than 300 time steps. While the plant shapes are quite different, the underlying genomes are very similar; most of them differ by one mutation in a gene which plays a key role during development (for details, see [Kim, 1996b]). Thus, the various phenotypes seen in Figures 7b-c are homeotic mutants of a common genetic theme [Meyerowitz, 1994].

Possibly due to their enormous phenotypic variability, the tree like plants are outcompeted by aggressive plants which extend only on the floor (not shown). Only after another 15000 time steps, a new type of plant emerges which grows diagonally and compensates the resulting imbalance by lateral extensions in the opposite direction (Fig. 7d). This plant type proves to be stable against aggressive plants on the floor, and continues to dominate for at least several ten thousand time steps. The improved robustness of these diagonal plants is partly due to their better efficiency in light absorption; a phenomenon also seen in LindEvol-GA.

4 LindEvol experiments

4.1 Evolution of complexity on different levels of biological organization

In LindEvol models, some key levels at which biological complexity and diversity is observed are represented. Complexity at the phenotypic level can be visually assessed by inspecting the growth processes and of plant communities. Energy usage or fitness levels can be used to quantitatively score the performance of the phenotypes. The length of developmental programs indicates the number of different cell types.

The emergence of complexity by evolution is only possible if there is an appropriate balance between variation (giving rise to new genetic configurations and thus to novel phenotypes) and selection (which purges variants of genetic information which give rise to phenotypes that fail to interpret, and interact with, their environment in a biologically meaningful way). The emergence of complexity on a level of biological organization is often correlated to complexity on other levels, and in some cases, plausible reasons for such correlations can be given. For example, the generation of more complex morphological structures may require more elaborate, and hence more complex, developmental processes.

In LindEvol-GA, mutation and selection are externally determined by control parameters (mutation rate and selection rate). As various levels of biological organization are represented by LindEvol, this system provides a basis for investigating the emergence of complexity on various levels and correlations between these phenomena by performing series of LindEvol-GA runs with varying settings of the selection rate and the mutation rates.

Comparing the phenotypes at generation 1000 and at generation 4000 shown in Fig. 6 demonstrates that without mutation rate adaptation, much less complex phenotypes evolve with a high global mutation rate. This is also indicated by the fitness value data and for developmental program length. Thus, the correlation between phenotypic and developmental complexity outlined above is captured by LindEvol-GA.

At the genetic level, the sequences from the entire population were subjected to a distance distribution complexity (DDC) analysis [Kim, 1996a, Kim, 1996b]. DDC is defined as the Shannon entropy of the relative abundances of distance values. This complexity measure was devised to characterize second order complexity, i.e. complexity which arises "at the edge of chaos" [Langton, 1992] by distinguishing populations with a rich taxonomic structure from unstructeredly randomized populations as well as from populations with a shallow taxonomy, e.g. due to convergence at a single fitness optimum. Fig. 6 shows that DDC increases as effective mutation rates are lowered and more complex phenotypes appear, indicating that there is indeed a positive correlation between the complexity characterized by DDC on the genetic level and complexity evolving at the developmental and at the phenotypic level. A systematic analysis revealed that high DDC values are strongly correlated to developmental program length and to high fitness values. Furthermore, the runs in which these elevated levels in complexity indicators were observed were shown to be located near an edge of chaos, which is formally defined by an error threshold analysis, according to their settings of the mutation rate and the selection rate [Kim, 1996a].

4.2 Evolutionary optimization by energy dependent mutation rate adaptation

Adapting mutation rates are useful for optimization by evolutionary algorithms [Rechenberg, 1994, Bäck, 1992]. However, mutation rates should be prevented from converging towards arbitrarily low values, as this may result in premature convergence of the evolutionary process.

In LindEvol-GA, mutation rate adaptation evolves only if this confers a substantial evolutionary advantage (see section 3.1), if mutation does not limit complexity, no adaptation of mutation rates is seen. This observation led to the development of a generalized scheme for implementing fitness dependent mutation rate adaptation which can be applied in a large class of evolutionary algorithms [Kim, 1998].

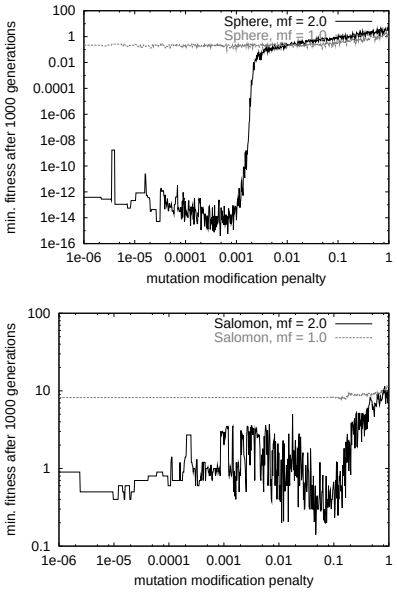


Figure 8: Optimal (i.e. minimal) fitness values after 1000 generations as a function of the penalty setting. Both axes have a logarithmic scale.

The key idea is to add a mutation modification penalty, denoted by p, to the control parameters of an evolutionary algorithm. Then, for an individual i, changing μ_i by one unit results in reducing its fitness value by p. Individual mutation rates are calculated as in equation 1.

This concept has been integrated into a basic evolution strategy [Rechenberg, 1994] in order to explore the effects of fitness dependent mutation rate on the performance of an evolutionary algorithm. Genomes are represented by vectors with real-valued components, and standard test functions (see [Salomon, 1996]) were used as fitness functions. Fig. 8 shows the results for two of these test functions, the sphere function $f(\vec{x}) = ||\vec{x}||$ and Salomon's function $f(\vec{x}) = -\cos(2\pi||\vec{x}||) + 0.1 \cdot ||\vec{x}|| + 1$.

For both functions, intermediate settings of the penalty parameter p yield best optimization results. Remarkably, this even holds true for the sphere function, which has no local optima. For multimodal functions like Salomon's function, the ideal setting of p is strongly shifted towards higher values, as too low penalties result in premature convergence with these functions. In the case of the sphere function, low penalty settings result in an uncounterbalanced slowdown of minimization speed, with nonzero settings of p, the slowdown is delayed and thus, optimization is able to proceed further towards the global optimum within 1000 generations.

4.3 Comparative analysis of various biodiversity measures

The phenomenon of biodiversity has attracted a significant amount interest during the 1990s [Ehrlich and Wilson, 1991]. Evolution is the key process that gives rise to biodiversity. Therefore, as it became increasingly clear that simple species numbers do not adequately reflect biodiversity, several methods for quantifying biodiversity based on genetic or phylogenetic information have been proposed [Crozier and Kusmierski, 1994, Faith, 1992, Nee and May, 1997, Williams et al., 1991]. However, a general definition of biodiversity does not yet exist, and the concepts and measures that have been developed in the recent years are difficult to compare and lead to contradictory results in some instances.

While biodiversity is generated by evolution, evolution does not necessarily lead to biodiversity, e.g. evolutionary algorithms often are designed to converge at an optimum in the fitness landscape, and convergence obviously implies decreasing diversity. In contrast to this, in Artificial Life open-endedness is considered a crucial property of evolution which is closely related to biodiversity [Bedau et al., 1998]. Thus, Artificial Life models are suitable systems for comparatively analyzing biodiversity measures, and conversely, biodiversity analyses of Artificial Life models may provide valuable information for developing them further towards open-ended evolution.

It is usually assumed that biodiversity increases as more complex life forms evolve. DDC exhibits this property (see Fig. 6 and section 4.1), suggesting that DDC may be suitable as a biodiversity measure which reflects the close relation between biodiversity, open-ended evolution, and complexity.

A comparative analysis of biodiversity measures, including DDC, has been carried out based on LindEvol-GA [Schwöbbermeyer and Kim, 1999]. Phylogenetic and genetic data were extracted from runs and used to compute time series of various biodiversity measures proposed in [Nee and May, 1997, Williams et al., 1991]. Quite strikingly, only measures which are computed from genetic data show any response to evolutionary transitions. No measure except DDC was found to increase as mutation rate adaptation sets in.

The lack of response to evolutionary events in phylogeny-based biodiversity measures is related to the use of different methods for obtaining phylogenies from LindEvol-GA and from molecular data. The phylogenies from LindEvol-GA were recorded independently of genetic information. In contrast to this, molecular phylogenies are reconstructed on the basis of genetic distances. Thus, phylogenetic and genetic information from LindEvol-GA are fully separated, while this separation cannot be achieved for molecular data. The finding that several biodiversity measures fail to extract hints about changing biodiversity from pure phylogenetic information provides a strong indication that phylogeny may not contain any significant information regarding biodiversity. Therefore, biodiversity should be measured from genetic data directly, as subjecting these to phylogeny reconstruction just obscures the biodiversity signal.

5 Summary and outlook

LindEvol combines various levels of biological organization into one model. The levels of genetic information, development, ecological interactions and evolution are represented, although each representation is necessarily coarse and incomplete in many respects. Nonetheless many fascinating processes can be observed in LindEvol simulation runs, and as LindEvol is accessible to systematic, computer aided analysis, such observations could be

used as starting points for steps toward a better understanding of complex biological phenomena:

- The observation of the emergence of structured taxonomic diversity was used as a basis to develop and test distance distribution complexity as a measure of complex diversity in evolutionary systems.
- The coupling of mutation rate adaptation to an energy cost in LindEvol models revealed that the link between protection from mutation to an energy expense in molecular nature may be a key mechanism that keeps evolution open-ended. This insight was transformed into an improvement of evolutionary algorithms.
- Using LindEvol as a basis for comparing biodiversity measures revealed that phylogeny may be unsuitable as a basis for assessing biodiversity, or that at least important aspects of biodiversity are neglected by an exclusive focus on evolutionary history.

Modelling regulatory networks is currently emerging as a new research focus [Mendoza and Alvarez-Buylla, 1998]. Integrating such models with the LindEvol concepts of genetic encoding and coevolution will certainly be useful for finding encodings of regulatory networks which are robust (in the sense of the Tierra language [Ray, 1992]). LindEvol-P is provides a promising basis for such studies.

Evolution and ecology are more and more recognized to be intimately interwoven, e.g. it is recognized that the separation between ecological and evolutionary time scales is arbitrary [Thompson, 1999]. Models combining these two key aspects of biology will be helpful in developing an integrated understanding of these two key levels. Again, LindEvol-P is a suitable basis for this direction of research, studies for integrating the ecological aspect of nutrient fluxes into LindEvol-P are already underway.

A well-known quote from Th. Dobzhansky says that "nothing makes sense in biology except in the light of evolution". LindEvol has been suitable as a basis for improving our understanding of the sense which life makes by evolution, and hopefully, it will continue to be a source of further insights in the future.

References

[Bäck, 1992] Bäck, T. (1992). Self-adaptation in genetic algorithms. In Varela, F. and Bourgine, P., editors, *Toward a Practice of Autonomous Systems*, pages 263–271, Cambridge, MA. MIT Press.

[Bedau et al., 1998] Bedau, M. A., Snyder, E., and Packard, N. H. (1998). A classification of long-term evolutionary dynamics. In Adami, C., Belew, R. K., Kitano, H., and Taylor, C., editors, *Artificial Life VI*, pages 228–237, Cambridge, MA. MIT Press.

[Crozier and Kusmierski, 1994] Crozier, R. and Kusmierski, R. (1994). Genetic distances and the setting of conservation priorities. In Loeschcke, V., Tomiuk, J., and Kain, J., editors, *Conservation Genetics*, pages 1311–1323, Basel, Switzerland. Birkhäuser Verlag.

[Ehrlich and Wilson, 1991] Ehrlich, P. R. and Wilson, E. O. (1991). Biodiversity studies: Science and policy. *Science*, 253:758–762.

[Faith, 1992] Faith, D. P. (1992). Conservation evaluation and phylogenetic diversity. *Biological Conservation*, 61:1–10.

[Goldberg, 1989] Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, New York.

[Kim, 1996a] Kim, J. T. (1996a). Distance distribution complexity: A measure for the structured diversity in evolving populations. In Langton, C. G. and Shimohara, K., editors, *Artificial Life V*, pages 281–288, Cambridge, MA. MIT Press.

[Kim, 1996b] Kim, J. T. (1996b). *Untersuchungen zur Evolution von morphologischer und taxonomischer Diversität und Komplexität anhand von Computermodellen*. PhD thesis, Universität zu Köln.

[Kim, 1998] Kim, J. T. (1998). Energy dependent adaptation of mutation rates in computer models of evolution. In Adami, C., Belew, R. K., Kitano, H., and Taylor, C., editors, *Artificial Life VI*, pages 248–255, Cambridge, MA. MIT Press.

[Langton et al., 1992] Langton, C., Taylor, C., Farmer, J., and Rasmussen, S., editors (1992). *Artificial Life II*, Redwood City, CA. Addison-Wesley.

[Langton, 1992] Langton, C. G. (1992). Life at the edge of chaos. In Langton, C. G., Taylor, C., Farmer, J. D., and Rasmussen, S., editors, *Artificial Life II*, pages 41–91, Redwood City, CA. Addison-Wesley.

[Mendoza and Alvarez-Buylla, 1998] Mendoza, L. and Alvarez-Buylla, E. R. (1998). Dynamics of the genetic regulatory network for *Arabidopsis thaliana* flower morphogenesis. *J.theor.Biol.*, 193:307–319.

[Meyerowitz, 1994] Meyerowitz, E. M. (1994). The genetics of flower development. *Scientific American*, 271(5):56–65.

[Nee and May, 1997] Nee, S. and May, R. M. (1997). Extinction and the loss of evolutionary history. *Science*, 278:692–694.

[Prusinkiewicz and Lindenmayer, 1990] Prusinkiewicz, P. and Lindenmayer, A. (1990). *The algorithmic beauty of plants*. Springer-Verlag, New York.

[Ray, 1992] Ray, T. (1992). An approach to the synthesis of life. In Langton, C., Taylor, C., Farmer, J., and Rasmussen, S., editors, *Artificial Life II*, pages 371–408, Redwood City, CA. Addison-Wesley.

[Rechenberg, 1994] Rechenberg, I. (1994). *Evolutionsstrategie '94*. Fromann-Holzboog, Stuttgart-Bad Cannstadt.

[Salomon, 1996] Salomon, R. (1996). Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. a survey of some theoretical and practical aspects of genetic algorithms. *BioSystems*, 39:263–278.

[Schwöbbermeyer and Kim, 1999] Schwöbbermeyer, H. and Kim, J. T. (1999). A comparative analysis of biodiversity measures. In Floreano, D., Nicoud, J.-D., and Mondada, F., editors, *Advances in Artificial Life*, Lecture Notes in Artificial Intelligence, pages 119–128, Berlin, Heidelberg. Springer Verlag.

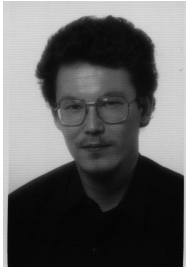
[Theißen and Saedler, 1995] Theißen, G. and Saedler, H. (1995). Madsbox genes in plant ontogeny and phylogeny: Haeckel's 'biogenetic law' revisited. *Current Opinion in Genetics and Development*, 5:628–639.

[Thompson, 1999] Thompson, J. N. (1999). The evolution of species interactions. *Science*, 284:2116–2118.

[Williams et al., 1991] Williams, P., Humphries, C., and Vane-Wright, R. (1991). Measuring biodiversity taxonomic relatedness for conservation priorities. *Australian Systematic Botany*, 4:665–680.

[Wilson, 1989] Wilson, S. W. (1989). The genetic algorithm and simulated evolution. In Langton, C. G., editor, *Artificial Life I*, pages 156–166, Redwood City, CA. Addison-Wesley.

Contact:
Jan T. Kim
Max-Planck-Institut für Züchtungsforschung
Carl-von-Linné-Weg 10, 50829 Köln, Germany
email: `kim@mpiz-koeln.mpg.de`
phone: ++49 (0)221 5062-538, fax: ++49 (0)221 5062-113



Jan T. Kim studierte Biologie an der Universität zu Köln und ist seit 1991 am MPI für Züchtungsforschung tätig. Sein Interessenschwerpunkt ist die Dynamik von Information in biologischen Systemen. Neben der Modellierung von Evolutions- und Morphogeneseprozessen beschäftigt er sich mit der Analyse von Transkriptionsfaktorsequenzen und entwickelt Anwendungen im Bereich der Bioinformatik.