Biol. Cybern. 69, 319–326 (1993)

Biological Cybernetics

Structure evolution and incomplete induction

R. Lohmann

Gesellschaft zur Förderung angewandter Informatik (GFaI), Rudower Chaussee 5, Geb. 13.7, D-12484 Berlin, Germany

Received: 4 April 1992/Accepted: 15 February 1993

Abstract. Evolutionary strategies such as the evolution strategy (Rechenberg 1965, 1973; Schwefel 1977) or genetic algorithms (Holland 1975; Goldberg 1989) have been widely applied to systems where parameters have to be determined according to a particular objective function. A necessary demand in all these experiments is that the structures of the objects to be optimised are well defined, because these structures are part of the objective function. With structure evolution the range of applications of evolutionary algorithms can now be expanded to tasks which are less accurately described, i.e. where the structures of the objects are fairly unknown. Heuristical effort is reduced first to defining structure components by combinations of which the structure space is generated. The structure space can be nearly infinitely large. Furthermore, the mutation procedures for structures have to be determined, complying with the demand for strong causality. In its computer model the algorithm of structure evolution involves the phenomenon of isolation, a feature of biological evolution additional to replication, mutation, and selection, which have already been implemented in other strategies. The idea of structure evolution is to let different but some what similar structures of an object compete in temporarily isolated populations where the respective parameter evolution is carried out. Thus structure evolution can perform a most effective search, both in structure and parameter space. The algorithm is demonstrated with two examples: a neural filter in a visual system and the topologies of frameworks. The first of the examples touches the problem of incompletely described tasks, and this paper will show that the effect of "overlearning" can be avoided by a learning procedure called "incomplete induction", which fits best with the algorithm of structure evolution.

1 Introduction

This paper considers the potential for systematically developing the design of a system that is asked to solve a particular problem. The design includes the system's structure and its parameters. It is only natural to begin with a discussion of task descriptions. The conclusion will be that the process of designing a system has to be regarded as an optimisation process of both the system's structure and its parameters with respect to the conditions and restrictions given by the task description. By means of extensive analytical investigation, the engineer usually has a good idea what the system that is expected to solve the task should look like. Only the determination of some parameters may be a bit uncertain, and he will subject the system to a parameter optimisation process. But what can the engineer do if he must admit that all analytical approaches to solve a task have failed? He will ask for an algorithm capable of systematically developing structures. At present the only promising algorithm for that purpose, seems to be structure evolution (Lohmann 1990, 1992). But the engineer is not released from all heuristical effort. He still has to define the structure components and to formulate rules about how to combine the components in order to constitute a sufficiently large structure space.

The structure of a system, i.e. an equation that describes a certain problem, represents an abstract comprehension of the problem to be solved. If the structure of the system is concurrently altered as proposed in structure evolution, one runs the risk that it focuses on the particular description of the task which is part of the objective function. This effect is well known as "overlearning" in neural networks. A learning procedure called "incomplete induction" is proposed to avoid unwanted specialisations in cases of incompletely described tasks. It can easily be implemented in the algorithm of structure evolution.

The most convincing proof that biological evolution works efficiently has always been its results. It may be appropriate, therefore, to report two examples where structure evolution has been successfully carried out.

2 Task description and optimisation

Discussing task descriptions without using an example does not make much sense. Let us, therefore, have a look at the patterns in Fig. 1 and ask what the features are that enable us to distinguish between the two leaves. Although we are far from being able to describe these features explicitly in technical terms, we suspect them to

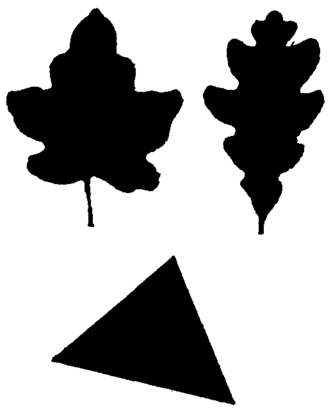


Fig. 1. Two leaves and a triangle

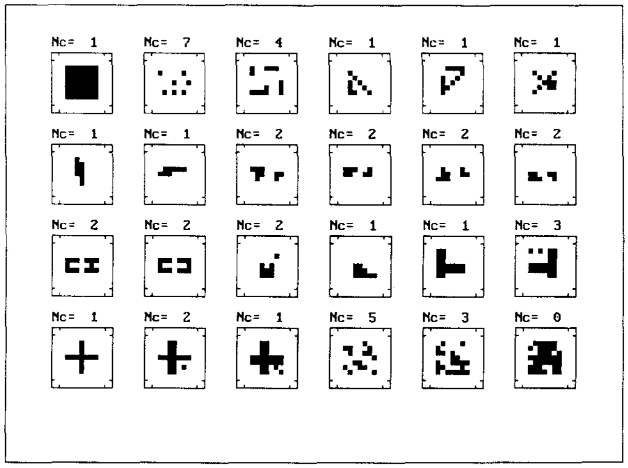


Fig. 2. A total of 24 different stimuli that may be representative for the task of a filter which should extract the number of coherent areas from any binary picture

exist, similar to certain common features of the leaf patterns that are necessary to distinguish between leaves and geometrical patterns. Another feature in Fig. 1 that has at least a precise verbal description is the number of coherent areas, three in this particular picture. The development of a neural filter that extracts this feature from any binary picture will be treated in detail later in this paper. Here we discuss only the point of task description.

The filter in question produces an output signal P if it is stimulated by an input picture S. We also have the target output T which is the correct number Nc of coherent areas in pictures like those in Fig. 2. For the respective set of N different stimuli S_i, the quality of a filter is defined as the mean square of differences between system outputs P_i and their respective target outputs T_i:

$$Q = 1/N \sum_{i=1}^{N} (T_i - P_i)^2 \tag{1}$$

We replace the mean square by a more general function G which says how Q depends on T_i and P_i:

$$Q = G(T_i, P_i) \tag{2}$$

The system output is a particular function $P_i = F(S_i)$ whose value depends on the stimuli S_i, while the target outputs T_i are some given constants $T(S_i)$:

$$Q = G[T(S_i), F(S_i)] \tag{3}$$

The objective obviously is the determination of the function F in (3). Regardless of whether this function is represented in mathematical terms or graphically as a neural network or as electrical circuitry, we can say that the function F consists of a structure C and a number of parameters W. In mathematical terms the structure C is a set of instructions on how to calculate the system output from input data, and probably these instructions require a number of parameters. Thus the quality Q of the system depends on the set of stimuli S, the structure C and the parameters W contained by the structure:

$$Q = G(S, C, W) \tag{4}$$

The problem will be solved when the quality Q becomes optimal. The objective here is, therefore, minimisation of Q. As the task is to find a filter that extracts a particular feature from any binary picture, which are infinitely many, the task is incompletely described because (4) must remain executable in a finite time. An algorithm to solve the minimisation problem in (4) must take into account the variation of all three terms: stimuli S, structure C and parameters W.

We customarily formulate conditions for Q to become extreme by setting all partial derivatives to zero:

$$\partial Q / \partial W_i = 0 \tag{5}$$

The respective conditions according to the discrete terms S and C in (6), however, are of no practical use as long as an analytical approach is preferred.

$$\Delta Q / \Delta C \rightarrow \text{Min}; \quad \Delta Q / \Delta S \rightarrow \text{Min} \tag{6}$$

But these expressions give a hint of what has to be done to solve the minimisation problem in (4). Besides the parameter optimisation, the algorithm to be used for that purpose has to control changes of both C and S, providing for an improvement of quality Q.

3 Evolution strategy

The evolution strategy (Rechenberg 1973; Schwefel 1977) is a model of biological evolution in which the phenomena of replication, mutation and selection are involved. The algorithm includes a mechanism for adaptive step-size control, and it seems to be preferable to other evolutionary strategies (Goldberg 1989; Holland 1975) when used in real coded problems (Hoffmeister 1991). There is no need to explain evolution strategy in detail in this paper since there are a lot of basic publications.

According to Schwefel (1977) an evolution strategy with μ parents and λ offspring is written as in (7), where the " + " or "," respectively denotes whether parents take part in the selection procedure or not.

$$(\mu + , \lambda) - \text{ES} \tag{7}$$

4 Structure evolution

In structure evolution it is necessary to extend the strategy of (7) to a system with many populations which are separated from each other during a certain isolation time τ. Thus the phenomenon of isolation is introduced consequently to evolutionary strategies; it has to be considered whether "genetic load", proposed by Born (1978) and Born and Bellmann (1983), has a similar effect on evolution strategies as isolation. Here different structures are assigned to the populations in which the parameter optimisation is carried out. A set of well-determined parameters is necessary for the assessment of structures. The formulation of a strategy for μ' parent structures and λ' offspring is given in (8).

$$[\mu', \lambda'(\mu, \lambda)^{\tau}_{S1,V1}]_{S2,V2} - \mathrm{ES} \tag{8}$$

At the end of isolation time τ, selection and mutation take place among the different structures on the population level. Obviously the variation procedures $V2$ for structures are different from parameter variations $V1$ used within a single population. This is expressed in (8) by the indices $V1$ and $V2$ assigned to the respective strategies. $V2$ has to be explained explicitly according to the particular structure space for each problem, while $V1$ is the variation mechanism used in all evolution strategies. In a similar way we distinguish between selection criteria $S1$ and $S2$ assigned to the two levels of the strategy. They will be discussed in the next section.

What remains to be done by the user of structure evolution is the definition of structure components and the determination of a set of appropriate mutation procedures which enable passage from one structure to another without violating the principle of strong causality.

5 Incomplete induction

If the parameters contained by a structure are optimised to some extent using a quality $Q1$, which is calculated from a limited set of stimuli $S1$, to control the selection, the result has the meaning of a first hypothesis concerning the whole task, which cannot be completely described, as mentioned before. To find out which of the different structures works most generally, these preliminary results have to be tested with another quality function $Q2$, calculated from a second set of stimuli $S2$. Both sets of stimuli, although different from one another, have to some extent to be representative concerning the whole task. As in (8) where the quality $Q2$ controls the selection $S2$ on the population level, there is a permanent selection pressure in the algorithm to ensure the preference for more generally working structures.

We propose to call this implementation of two representative but different selection criteria (one used to establish a first hypothesis, the other one to verify it) the "incomplete induction" procedure, because the reasoning is similar to that in the mathematical technique of "complete induction" used to check equations or mathematical sentences.

We do not pretend to give any proof concerning induction in this paper, but experimental evidence that induction works, in the way described above may be useful. Figure 3 shows the registered data of $Q1$ and $Q2$ versus time for an experiment carried out according to the strategy described in (8). $Q1$ is calculated from input data introduced in Fig. 2, whereas the second quality $Q2$ refers to quite different binary pictures, shown in Fig. 4.

There are two experiments in Fig. 3, the first one plotted as "×" and "+" for $Q1$ and $Q2$. Both qualities become smaller with the passage of time, as expected. At time $t1$ in Fig. 3 a second experiment branches from the first one, $Q1$ and $Q2$ now plotted as squares. In this

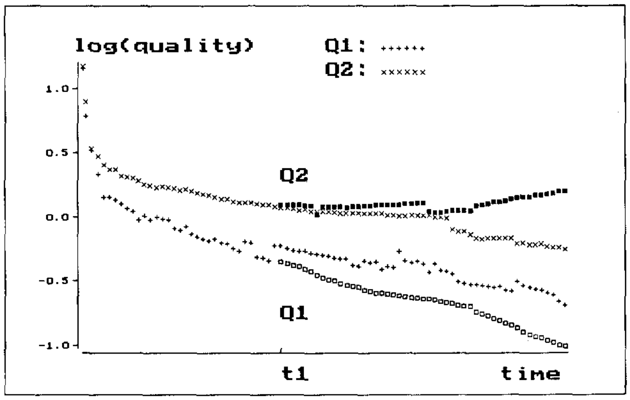


Fig. 3. Quality functions $Q1$ and $Q2$ versus time

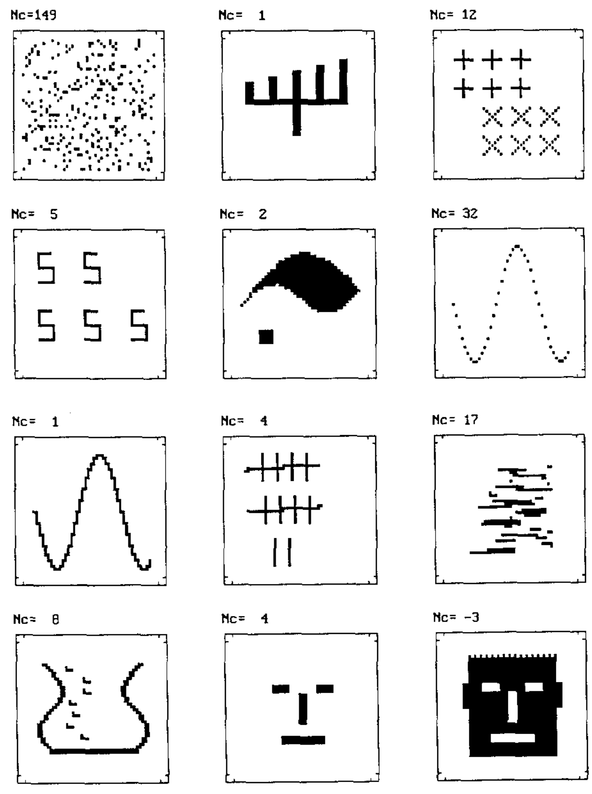


Fig. 4. Twelve pictures used to calculate $Q2$; compare with the 24 stimuli in Fig. 2

second experiment the selection $S2$ is controlled by the same quality function $Q1$ that is used in $S1$. $Q2$ has been registered only for the purpose of documentation. We observe in the second experiment that $Q1$ approaches a minimum faster than in the first. But the price is that $Q2$ in the second experiment stagnates and even becomes worse. The result suggests that the faster convergence of $Q1$ in the second experiment comes with a higher specialisation of the system on the particular set of stimuli $S1$. The result will be comparatively worthless on different stimuli.

6 Example 1: local filter for the feature "number of coherent areas"

In this example we consider the development of a local filter that is part of a particular visual system. We have already described the task of the filter. The design of the visual system is demonstrated in Fig. 5.

The construction is partly a model of the biological design of the retina. It begins on the left side with a layer of receptors (RZ) which is followed by a homogeneous layer of local processing units (PZ). Each local filter can process only input data coming from a small neighbourhood, which is restricted to a $3*3$ matrix. By means of its structure and the respective parameters, the local filter calculates from its input data a local contribution to a global feature. The global feature (PM) is defined to be the sum of all local contributions.

The global feature here is identical with the system output P_i in (1). Target outputs T_i assigned to the stimuli used here have been reported in Sect. 2 as correct numbers of coherent areas Nc, written above each picture in Figs. 2 and 4. There is no difficulty in calculating quality functions according to (1).

The objective is to determine the structure of the local filter as well as the parameters contained by the structure. To apply structure evolution to this problem we first have to define the structure components. Figure 6 shows a graphical representation of a filter.

The structure is composed of input elements, a variable number of units to calculate different binary products, and finally a weighted sum.

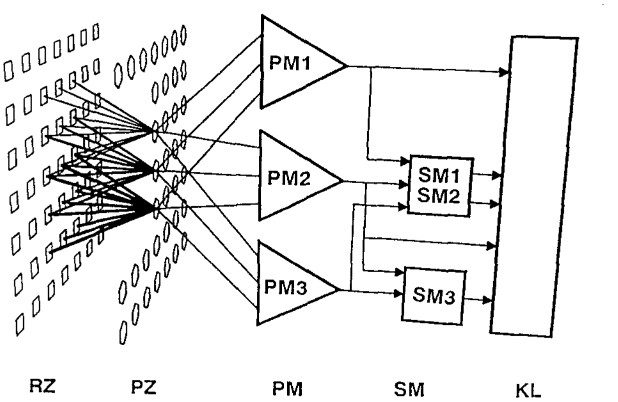


Fig. 5. Construction of a visual system (see explanations in the text)

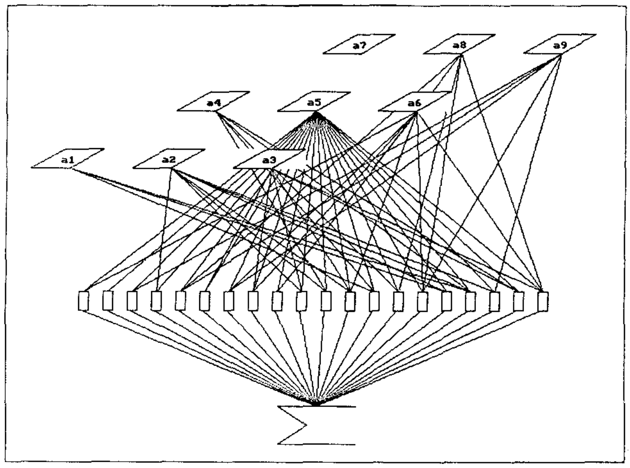


Fig. 6. Graphical representation of a local filter structure with nine input elements $a1$, $a2$,... $a9$ (*top*), several units that perform binary multiplication (*middle*) and another unit (*bottom*) that calculates a weighted sum of all binary products

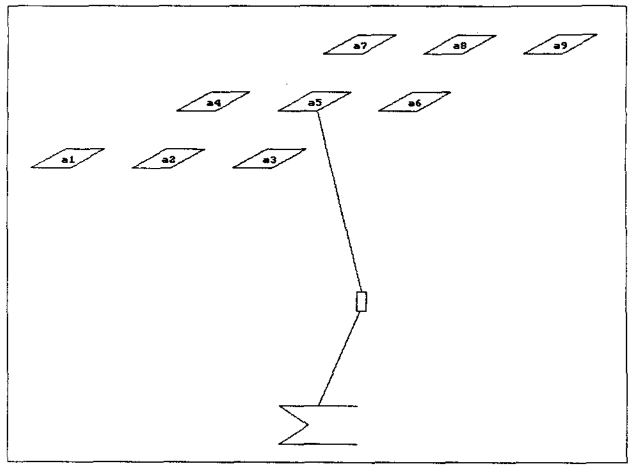


Fig. 7. Start configuration of the local filter

The number of parameters in this structure, called the filter length L, is the same as the number of products. If the filter length does not exceed $L = 20$, an estimation says that $Z = 10^{35}$ different structures can be composed by the structure elements.

There are three different variation procedures that comply with the demand for strong causality:

1. Elimination of a product in the structure, preferably one to which a parameter with a small absolute value is assigned
2. Duplication of an existing product and removal of one of the factors
3. Duplication of a product and addition of a new factor

In procedures 2 and 3 the filter length increases by one. The new parameter is set to zero when the mutation is executed, so the quality does not change.

The process of structure evolution has been started with the most simple configuration given in Fig. 7, and

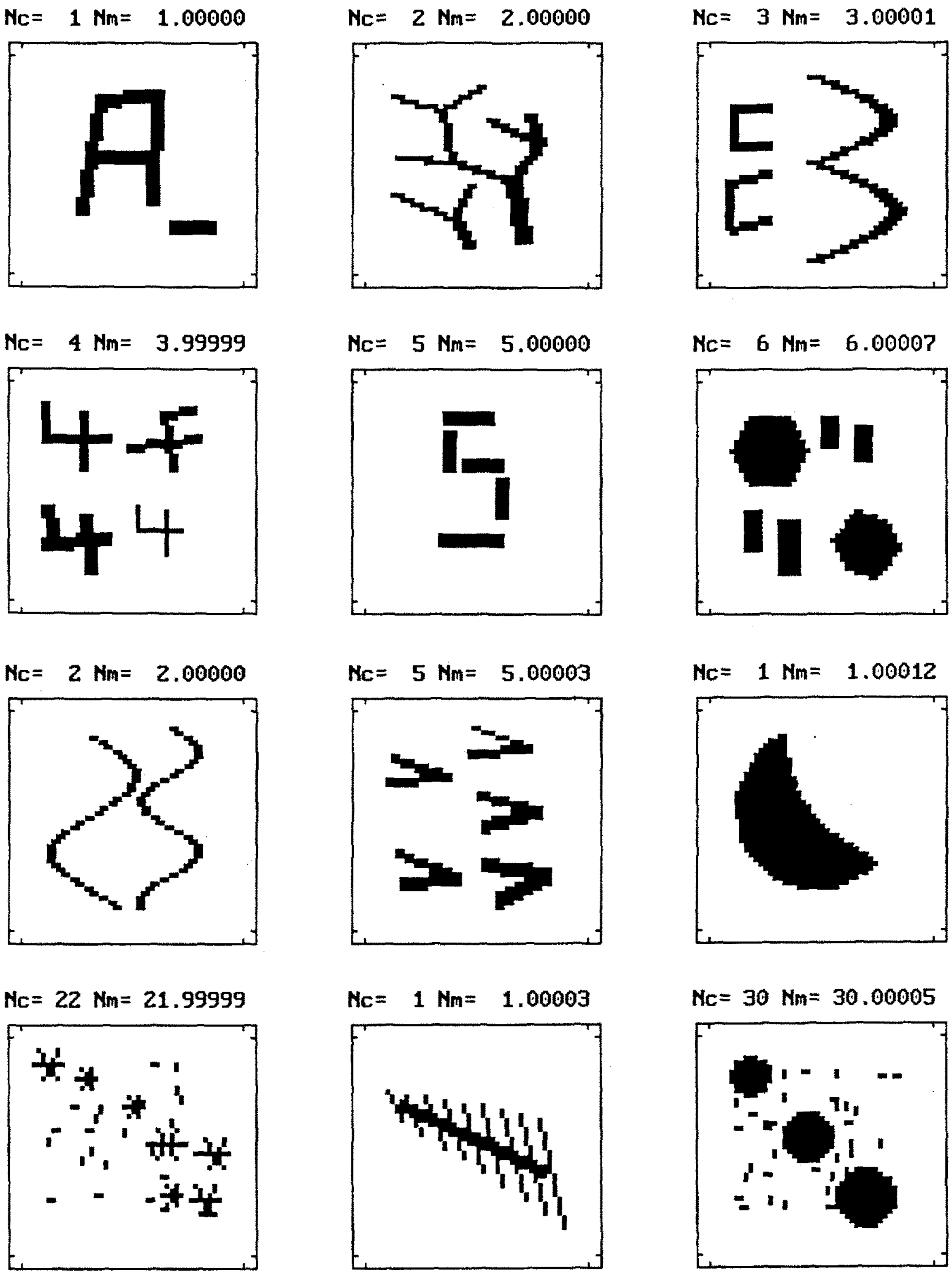


Fig. 8. Test set of 12 stimuli that never occurred in the evolution process. Above each picture the correct (*Nc*) and the measured numbers (*Nm*) of coherent areas are written

the result has already been reported in Fig. 6. During the whole process only 10^5 different structures have occurred, some of them only for a few generations in parameter evolution. Thus we can claim that a most effective search in a nearly infinitely large structure space and the respective parameter spaces has taken place.

In the end we like to test the filter on a set of pictures that have never occurred in the process of structure evolution (see Fig. 8). Above each picture one can read again the correct number of coherent areas Nc, followed by the measured number Nm. As differences between Nm and Nc are rather small, we can say that structure evolution has worked sufficiently and that incomplete induction has taken place.

7 Example 2: topologies of frameworks

A convincing example for the development of structures is provided by frameworks, because of their comprehensive graphical representations. Obviously frameworks are not dealing with input and output data, and in fact this example has been chosen to show that structure evolution does not require the reasoning of incomplete induction in any case.

The task of the two-dimensional framework, which is not exposed to any internal or external force, is to connect six fixed points plotted as in the left side of Fig. 9. The first restriction is that the framework is non-cinematic. Furthermore, we want the bars of the framework to comply as well as possible with a target length L_t which is shown on the left edge in Fig. 9. In order to save resources the total length of all bars is set to become a minimum. The corresponding quality function for R bars is given in (9):

$$Q = 1/R \sum_{i=1}^{R} (L_t - L_i)^2 + \sum_{i=1}^{R} L_i \tag{9}$$

The number of variables in the system is $Z = 2*(R - 6)$. We recognize that the structure components are the bars and the nodes whose numbers may vary. We must admit that the number of different structures seems to be rather large; we also admit some difficulties in the estimation of a lower bound of that number.

There are four variation procedures for the topologies of frameworks. The first one inserts a node on an existing bar and connects it with another node in its neighbourhood. The second procedure simply inserts an additional bar between two existing nodes that are not too far from each other. The third and fourth variation procedures are inversions of the first two.

The strategy used in this experiment worked with a ten-population system and an isolation time of 20 generations.

$$[10(1, 10)^{20}{}_{S1,\,V1}]_{S2,\,V2} - \mathrm{ES} \tag{10}$$

In order to provide the system with a good variability all structures suffer one of the above mutations after 20 generations of parameter evolution. The selection procedure $S2$ is chosen to be rather weak by replacing only the worst structure by the best one in each cycle on the population level. There is only one selection criterion, quality Q according to (9).

The best result (right side of Fig. 9) has been achieved after 1559 generations on the structure level. It is surprising that this framework is not symmetrical, but we understand that a ring-like framework requires a bigger total length of its bars than the "C"-shaped topology in Fig. 9. The results of structure evolution in this example are only near-optimal, since the strategy is a non-conservative one. Continuation of the process sometimes leads to worse topologies, as can be recognized in Fig. 10.

Of some interest may be a brief look at the small choice of topologies in Fig. 10, taken from the fossil collection due to the evolution process. Note that there are only 20 generations of structure evolution between structures 4 and 5. Although they look completely different at first glance, their distance in structure space is rather small. This observation might explain Darwin's great difficulties in finding evidence for his theory of biological evolution. He had to rely on the much smaller choice of fossils found upto then.

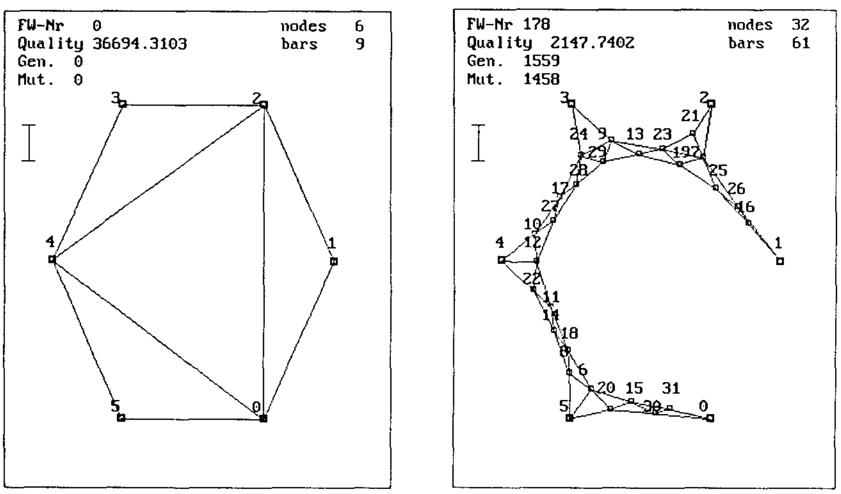
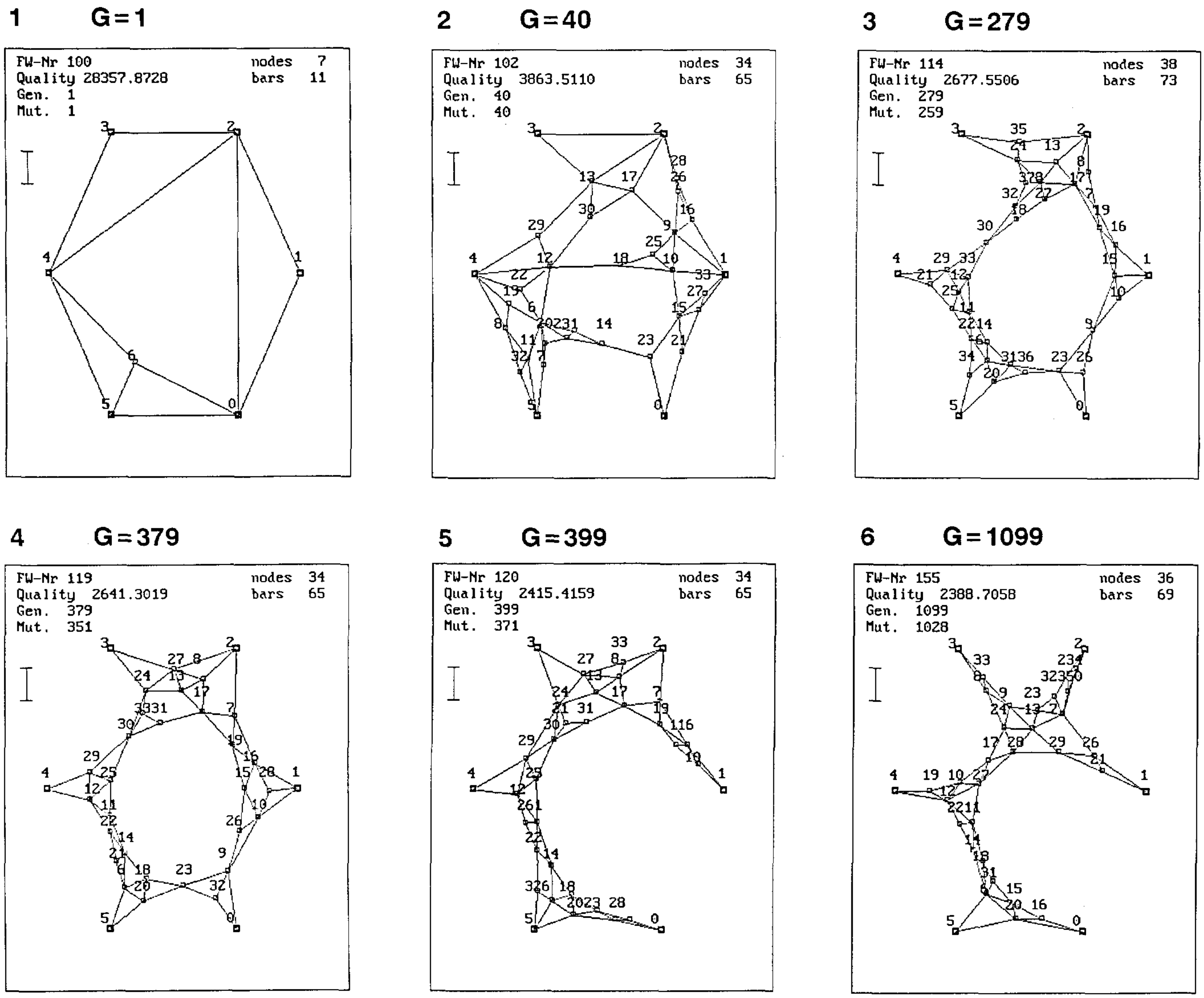


Fig. 9. *Left*: Start configuration of a non-cinematic framework connecting six fixed points. *Right*: The best framework structure occurring in the evolution process



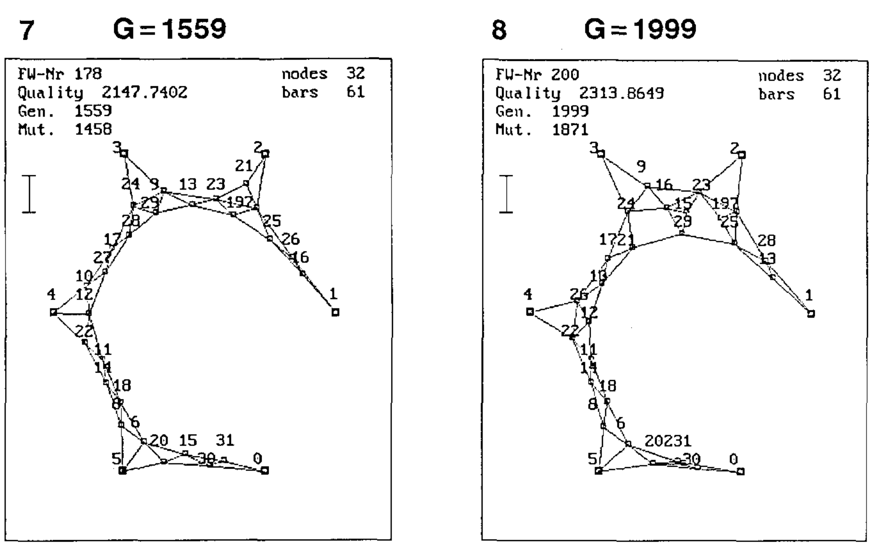


Fig. 10. (1–8) A choice from the fossil collection due to a structure evolution of a framework. G is the number of generations run off on the population level of the strategy when the configurations were stored

8 Conclusion

This paper presents the algorithm of structure evolution as a further-developed evolution strategy. It benefits from modelling the biological phenomenon of isolation in strategies with many populations. It is capable of developing structures including the parameters of a system, as long as the task is appropriately described and the user successfully defines structure space and attendant mutation procedures. Tightly connected with the algorithm is a learning procedure called incomplete induction, which refers to incompletely described tasks. Both structure evolution and incomplete induction seem to open the door to a systematic approach to developing structures of systems. Since there is not much theory on structure evolution to date, one has to be content with results from experiments in which the algorithm worked effectively.

References

Born J (1978) Evolutionsstrategien zur numerischen Lösung von Adaptationsaufgaben. PhD thesis, Humboldt Universität zu Berlin

Born J, Bellmann K (1983) Numerical adaptation of parameters in simulation models using evolution strategies. In: Bellmann K (ed) Molecular genetic information systems: modelling and simulation. Akademie, Berlin, pp 291–320

Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Reading, Mass

Holland JH (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor

Hoffmeister F, Bäck T (1991) Genetic algorithms and evolution strategies: similarities and differences. In: Schwefel H-P, Männer R (eds) Parallel, problem solving from nature. (Lecture notes in computer science, vol 496) Springer, Berlin Heidelberg New York, pp 455–469

Lohmann R (1990) Selforganization by evolution strategy in visual systems. In: Voigt H-M, Mühlenbein H, Schwefel H-P (eds) Evolution and optimization '89. Akademie, Berlin, pp 61–68

Lohmann R (1992) Structure evolution in neural systems. In: Suocek B, IRIS Group (eds) Dynamic, genetic and chaotic programming. Wiley, New York, pp 395–411

Rechenberg I (1965) Cybernetic solution path of an experimental problem, Library Translation 1122. Royal Aircraft Establishment, Farnborough, UK

Rechenberg I (1973) Evolutionsstrategie – Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Frommann-Holzboog, Stuttgart

Schwefel, H-P (1977) Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie. Birkhäuser, Basel