ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 137-152

Evolutionary Algorithms and Neural Networks in Hybrid Systems

Thomas Villmann
University Leipzig
Clinic for Psychotherapy and Psychosomatic Medicine
D–04107 Leipzig, Karl–Tauchnitz–Str.25, Germany
email: villmann@informatik.uni-leipzig.de

Abstract. In this contribution we give a short overview about actual developments in hybrid systems combining artificial neural networks and evolutionary algorithms. In more detail we highlight such systems which take the self-organizing maps as neural architecture into account.

1. Introduction

In the area of *Computational Intelligence* several approaches are established for solving complex computational tasks which are inspired by nature. Main directions are *artificial neural networks*, *evolutionary* or *genetic algorithms*, *Fuzzy-approaches* and other. The applications range from clustering, pattern recognition, optimization and other to reasoning, data mining and knowledge discovery. A large number of specialized algorithms were developed for specific tasks. However, often the tasks require to combine several approaches which then are called hybrid systems. In this paper we want pay attention to such systems which combine evolutionary/genetic algorithms with artificial neural networks.

The article is structured as follows: In sec.2. we shortly review the actual state in development of hybrid systems combining neural and evolutionary paradigms. After this, in sec.3., we focus on hybrid approaches combining self-organizing maps and evolutionary algorithms.

2. Neural Networks and Evolutionary Algorithms

Artificial neural networks (ANNs) attempt to replicate the computational power of biological neural networks and should lead to algorithms with some of the *cognitive abilities* that biological organisms possess [43]. The basic characteristics are *adaptability*, *speed* (due to the massive parallelism), *robustness*, *ruggedness* and *optimality* with respect to certain criteria. The various network models differ in network structure, neuron (node) characteristics, learning dynamic regarding to their application tasks.

Evolutionary algorithms (EA)[1] are stochastic search algorithms which are adapted from principles of biological evolution working simultaneously on a

[1] In literature one distinguishes several types of genetic/evolutionary approaches. Besides

large number of *potential* problem solutions. These solutions are called individuals. The optimality of the individuals for a specific tasks is defined by a *fitness function* which plays the role of the assessing environment. The pool of individuals form a population which may be divided into interacting subpopulations. Comprehensive foundations of EAs are [37] and [44].

The combination of neural networks and EAs offers new possibilities to increase the power of adaptive approaches. Two main directions can be identified: the first one is to apply EAs to optimize a neural network whereas the second one deals with the transfer of methods from neural networks onto the frame work of EAs. As pointed out in [43] the latter case is not yet explored.

The application of EAs for performance improvement of neural networks has a long tradition (for a compact overview we refer to [17],[37],[43],[62],[63]). The most developments consider the optimization of the network topology which is a crucial point in network design [24]. The hardest limitations thereby is the large training effort: in each generation step of the EA all individuals representing an neural network have to be trained according the respective learning algorithm. Frequently, the backpropagation network is investigated because its large number of applications [38],[48],[50],[64]. However, especially for this network type the high learning effort is well known [24]. But also other network designs where developed using evolutionary algorithms: recurrent neural networks [2], general asymmetric neural networks [8], high ordered neural networks [14], cellular neural networks [41], Hopfield network [42], sparse feedforward neural networks [49] or radial basis function networks [61]. The widely ranged self-organizing map was also considered. We will concentrate on this network type in the next section, separately.

Beside the topology (structure) design of neural networks there exist a few number of papers to application of evolutionary algorithms to learning in neural networks [29],[35],[53],[54]. The usage of EAs for training may be helpfully in avoiding local minima in learning dynamics as pointed out in [53]. Yet, a general advantage is not recognizable because of the fine tuned neural learning schemes. In general, EAs gives an advantage only if specialized learning algorithms for global energy function minimizations are note available [3],[44].

3. Evolutionary Algorithms and Self-Organizing Maps

Let us now consider in more detail hybrid approaches which explicitly combine EAs with *Self-organizing maps* (SOM). In this area we find all directions of combinations: evolutionary design of SOMs, evolutionary weight adaptation and application of SOM learning strategies for increasing of convergence speed in EAs. First, for clarification we shortly introduce the basic concepts of self-organizing maps:

3.1. Self-Organizing Maps

Self-organizing maps [30] as a special kind of neural maps project data from some (possibly high-dimensional) input space $\mathcal{V} \subseteq \Re^{D_{\mathcal{V}}}$ onto a position in some

the differences in detail we denote all these algorithms in this paper as *evolutionary algorithms*. For a detailed description we refer to [37].

output space (neural map) $\mathcal{A}$, such that a continuous change of a parameter of the input data should lead to a continuous change of the position of a localized excitation in the neural map. This property of *neighborhood preservation* depends on an important feature of the SOM, its output space topology, which has to be specified prior to learning. Usually the topology is chosen hypercubically. This can be cast in a formal way by writing the output space positions as $\mathbf{r} = (i_1, i_2, i_3, ..., i_{n_j})$, $1 < i_k < n_k$ with $N = n_1 \times n_2 \times ... \times n_j$ where n_k, $k = 1, ..., j$ is the dimension of $\mathcal{A}$ (the length of the edge of the lattice) in the k^{th} direction .[2] Associated with each neuron $\mathbf{r} \in \mathcal{A}$, is a weight vector, or *pointer*, $\mathbf{w_r}$ in $\mathcal{V}$. The mapping $\Psi_{\mathcal{V}\to\mathcal{A}}$ is realized by a winner take all rule

$$\Psi_{\mathcal{V}\to\mathcal{A}} : \mathbf{v} \mapsto \mathbf{s} = \underset{\mathbf{r}\in\mathcal{A}}{\operatorname{argmin}} \|\mathbf{v} - \mathbf{w_r}\| \tag{1}$$

whereas the reverse mapping is defined as $\Psi_{\mathcal{A}\to\mathcal{V}} : \mathbf{r} \mapsto \mathbf{w_r}$. The two functions together determine the map $\mathcal{M} = (\Psi_{\mathcal{V}\to\mathcal{A}}, \Psi_{\mathcal{A}\to\mathcal{V}})$ realized by the SOM network. All data points $\mathbf{v} \in \Re^{D_\mathcal{V}}$ that are mapped onto the neuron $\mathbf{r}$ make up its receptive field $\hat{\Omega}_\mathbf{r}$. The masked receptive field of neuron $\mathbf{r}$ is defined as the intersection of its receptive field with $\mathcal{V}$:

$$\Omega_\mathbf{r} = \{\mathbf{v} \in\mathcal{V} : \mathbf{r} = \Psi_{\mathcal{V}\to\mathcal{A}}(\mathbf{v})\} . \tag{2}$$

Therefore, the masked receptive fields $\Omega_\mathbf{r}$ are closed sets. All masked receptive fields form the Voronoi tesselation of $\mathcal{V}$. If the intersection of two masked receptive fields $\Omega_\mathbf{r}$, $\Omega_{\mathbf{r}'}$ is non-vanishing we call $\Omega_\mathbf{r}$ and $\Omega_{\mathbf{r}'}$ neighbored. The neighborhood relations form a corresponding graph structure $\mathcal{G}_\mathcal{V}$ in $\mathcal{A}$: two neurons are connected in $\mathcal{G}_\mathcal{V}$ if and only if their masked receptive fields are neighbored. The graph $\mathcal{G}_\mathcal{V}$ is called the induced Delaunay-graph (See, for example, [36] for detailed definitions). Because of the bijective relation between neurons and weight vectors $\mathcal{G}_\mathcal{V}$ also represents the Delaunay graph of the weights.

To achieve the map $\mathcal{M}$, SOMs adapt the pointer positions during the presentation of a sequence of data points $\mathbf{v} \in\mathcal{V}$ selected from a data distribution $\mathcal{P}(\mathcal{V})$, as follows:

$$\triangle\mathbf{w_r} = \epsilon h_\mathbf{rs}(\mathbf{v} - \mathbf{w_r}) . \tag{3}$$

$h_\mathbf{rs}$ is the neighborhood function, usually chosen to be of Gaussian shape:

$$h_\mathbf{rs} = \exp\left(-\frac{\|\mathbf{r} - \mathbf{s}\|^2}{2\sigma^2}\right) \tag{4}$$

Note that $h_\mathbf{rs}$ is dependent on the best matching neuron (1).

Topology preservation in SOMs is defined as the preservation of the continuity of the mapping from the input space onto the output space, more precisely it is equivalent to the *continuity* of $\mathcal{M}$ between the *topological spaces* with properly chosen metric in both $\mathcal{A}$ and $\mathcal{V}$. For lack of space we refer to [59] for detailed considerations. The topology preserving property can be used for immediate evaluations of the resulting map. Topology preservation also allows the applications of interpolating schemes such as the parametrized SOM

[2] Other spatial arrangements are also possible, which can be described by a connectivity matrix.

(PSOM) [47] or interpolating SOM (I-SOM) [19]. A higher degree of topology preservation, in general, improves the accuracy of the map [6]. There exist a lot of measure to determine the degree of topology preservation. An overview is given in [5].

3.2. Performance Improvement of SOM by Evolutionary Algorithms

3.2.1. Structure Adaptation in SOM by Evolutionary Algorithms

A crucial but strongly demanded feature of SOM is the above mentioned property of topology preservation. However, if the shape of the input space $\mathcal{V}$ does not match the structure of the lattice $\mathcal{A}$ violations of topology preservation are inevitable [59]. Several extensions of the usual SOM were developed to overcome this problem: growing cell structures (GCS) [18], the growing SOM (GSOM) [7] etc. The disadvantages of these algorithms consist in slowly increasing number of neurons (GCS) or remaining hypercube structure (GSOM).

Therefore, EAs were applied to structure optimization [21],[45],[46]. For this purpose the lattice structure of $\mathcal{A}$ is defined via a connectivity matrix $\mathcal{G}_{\mathcal{A}}$ representing the graph structure, i.e. $\mathcal{G}_{\mathcal{A}}(\mathbf{r},\mathbf{r}') = 1$ if and only if $\mathbf{r}$ and $\mathbf{r}'$ are connected in $\mathcal{A}$ elsewhere $\mathcal{G}_{\mathcal{A}}(\mathbf{r},\mathbf{r}') = 0$. Hence, the indices of neurons only define the positions in the neuron space. Now, each individual in an EA scheme codes one possible connectivity graph. The EA generates new graphs, the respective SOMs are trained according the usual learning rule (1) and assessed by a topology preservation measure serving as fitness function. As a remaining problem we have the large computational effort due to the time consuming network learning. A reduction of time can be achieved by usage of parallel techniques.

3.2.2. Evolutionary Weight adaptation

Other authors investigate the weight adaptation by evolution replacing the original learning dynamic (1). As pointed out in by Kohonen in [30] one can take the learning dynamic as an evolving system. We know two independently developed but nearly identical realizations [31] and [40] the first one published by Kohonen itself. Originally, the evolutionary SOM is designed for hypercubical lattice structures. However, it can easily be generalized to arbitrary graph structures $\mathcal{G}_{\mathcal{A}}$.

Let us denote by $\mathcal{S}_R(\mathbf{r})$ a subgraph of $\mathcal{G}_{\mathcal{A}}$ assigned to certain nodes $\mathbf{r}$ such that for a given radius R the nodes $\mathbf{r}'$ with distance $d_{\mathcal{G}_{\mathcal{A}}}(\mathbf{r},\mathbf{r}') \lesssim R$ belong to $\mathcal{S}_R(\mathbf{r})$. Now the evolution for a given input $\mathbf{v}$ is realized by the following steps:

1. determine the winning neuron $\mathbf{s}$ according 1 and the respective subgraph $\mathcal{S}_R(\mathbf{s})$
2. pick up randomly $n(R)$ nodes as population for EA
3. perform one evolution step (recombination/mutation and selection) with (1) serving as fitness function

Thereby, R and, hence, $n(R)$ decreases during map evolution as in the original SOM the neighborhood range σ. The original learning rate ϵ can be taken as a scaling value of the variance for the mutation.

3.3. Advanced EA-schemes inspired by SOM

3.3.1. Convergence Improvement by Neighborhood Related Genetic Operations

The collective neighborhood oriented learning scheme of SOMs can be transferred to the evolution process in EAs. The individuals are now placed on lattice positions according to the SOM approach. In the approach provided by HUHSE a new genetic operator is introduced, called *neighborhood attraction operator*, which carries out a *directed variation* of the individual in direction of its fittest neighbor [27],[28]. The resulting EA dynamic is accelerated in comparison to usual variation by traditional (random) recombination.

TOTH&LÖRINCZ also incorporated neighborhood informations into the EA dynamic [55]. The specific task was to find simultaneously the maximum of a family of *parametrized* functions $f_{\mathbf{p}}(\mathbf{x}) : \Re^n \rightarrow \Re$, whereby $\mathbf{p} \in \mathcal{P} \subseteq \Re^m$ is a continuously varying parameter labeling for the different functions. In other words: we are seeking for mapping $X(\mathbf{p}) : \Re^m \rightarrow \Re^n$ such that $f_{\mathbf{p}}(\mathbf{x})$ is maximum for each value $\mathbf{p}$. Now the authors suggested to partition the space $\mathcal{P}$ into regions $\mathcal{P}_i$ using a SOM and assigning to each region a subpopulation of individuals for EA to maximize the respective function $f_{\mathbf{p}}(\mathbf{x})$. Hence, a neighborhood relation between the subpopulations is induced by the SOM structure. Thereby a recombination of individuals between neighbored subpopulations is allowed which again correspond to a neighborhood oriented collective adaptation. However, because of the unknown structure of $\mathcal{P}$ and the corresponding difficulties in topology preserving mapping by SOMs a better way would be to use the *topology representing neural network* (TRN) [36]. The TRN automatically determines the topological structure in $\mathcal{P}$ induce the neighborhood relations between the subpopulations.

3.3.2. SOM-like migration for subpopulations

In the previous section we only considered the convergence speedup resulting from the consideration of (nearest) neighbored individuals/subpopulations. This scheme can be further generalized:

We now consider a *multiple subpopulation approach* with a set Π of subpopulations Π_i also known as island model [37]. In the present approach the basic new point is that the set Π of subpopulations is *arranged on a topological structure* Ω. Multiple subpopulation approaches are extensively considered by CANTU-PÁZ [10],[12],[11],[13], and others [37],[39],[33],[44]. Usually, in the most approaches Ω is to be chosen of simple two-dimensional shape in agreement with biological migration systems (island model). During the evaluation individuals can *migrate* to subpopulations within a certain fixed neighborhood range according Ω. The small migration rate is always constant. Thereby, migration means in our approach that an individual of one subpopulation visits an other subpopulation for a short time (a few generation steps) to share the genetic information during this time with the members of the visited subpopulation [3].

[3] Yet, there are different realization of migration reported. An other way is to substitute members from a subpopulation by the migrating individuals. For an analysis of this kind of migration we refer to [12].

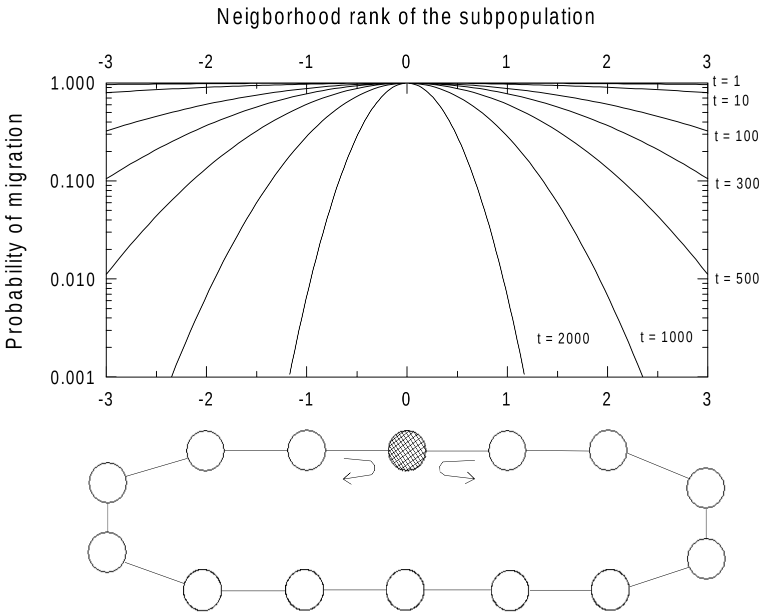

Figure 1: Exemplary plot of the neighborhood function $h_{i^*}(t,k)$ from eq.(2.2) determining the migration scheme in dependence on time t and neighborhood rank $r^{\Omega}_{i^*,k}$ from eq. (2.1) for a simple ring topology. Here, $\epsilon_h = 0$.

We favour a *dynamic neighborhood dependent migration scheme*. Let be given a structure graph $\mathcal{G}_\Omega$ representing the topological relations in Ω. To each node i in $\mathcal{G}_\Omega$ is assigned a subpopulation Π_i. Hence, we can define the rank of neighborhood for an actually considered subpopulation Π_{i^*} to another subpopulation Π_k according to the topological order in Π as rank of neighborhood:

$$r^{\Omega}_{i^*,k} = \mathrm{rank}\left(\Pi_{i^*}, \Pi_k, \Omega\right) \tag{5}$$

with $\mathrm{rank}\left(\Pi_{i^*}, \Pi_k, \mathcal{G}_\Omega\right)$ is defined as the minimal path length in the graph $\mathcal{G}_\Omega$. Now the migration scheme can be implemented locally based on the neighborhood relations between the subpopulations determined by a neighborhood function h which is motivated by the neighborhood function in SOMs [58]. The neighborhood function is decreasing in both neighborhood rank and time:

$$h_{i^*}(t,k) = (1.0 - \epsilon_h) \cdot \exp\left(-\frac{r^{\Omega}_{i^*,k}}{2\left(\sigma_h(t)\right)^2}\right) + \epsilon_h \tag{6}$$

with a small positive constant ϵ_h. In this way the degree of neighborhood becomes dynamically. Because of $h_{i^*}(t,k) \in (0,1]$ we immediately can interpret $h_{i^*}(t,k)$ as a *probability for migration* of an individual of the subpopulation Π_k into the actual evaluated Π_{i^*}. During the evaluation of a certain subpopulation Π_{i^*} the neighborhood function h is applied to determine the number of visiting individuals from each other subpopulation. At the beginning t_0 of the evolution process the range of the neighborhood nearly comprises the complete set of subpopulations and decreases exponentially during the time determined by the neighborhood range

$$\sigma_h(t) \underset{t\to\infty}{\longrightarrow} 0 \tag{7}$$

in (6) (see Fig. 1). In this way, σ_h determines a characteristic scale for the range of the information flow.

Following this approach one can take the individuals of all subpopulations at the beginning as an uniform population which performs a first rough adaptation process. During the further development the various subpopulations become more and more separated from each other and, hence, search in *different regions* of the solution space. On the other hand, ϵ_h in (6) preserves a *remaining probability for migration.* In this way one has a non–vanishing information flow through the topological ordered set Π of subpopulations. If one chooses $h_{i^*}(t,k) = h_{i^*}(k)$ independent on time t and, furthermore,

$$h_{i^*}(k) = \begin{cases} h^* & \text{for } k \le k^* \\ 0 & \text{else} \end{cases} \tag{8}$$

for a certain fixed k^* and $h^* \in (0,1]$ we can identify k^* as the degree of neighborhood in the Cantu-Páz early approach [10] whereas h^* is the now constant migration rate. In later approaches Cantu-Páz also introduced a time dependent degree of neighborhood which is adapted according to the fitness of the individuals within the subpopulations [11]. However, the determination of this value is computational expansive. Hence, our above introduced neighborhood function avoids this computational costs.

This approach of first rough adaptation of neighboring subpopulations together with a more fine tuning in the further process by more and more but not completely independent subpopulations arranged on a topological structure according to a neighborhood function is an analogy to the SOM-learning strategy: a first rough adaptation of the neuron weights takes place changing to a precise adjustment simultaneously with loosing the strong neighborhood conditions in the lattice [59].

As shown in the applications below, this dynamic migration scheme *accelerates* the adaptation process, especially, if the search space possesses many local minima [37] (see sect. 3.3.2.). Moreover, if a multiple processor system for instantiating is used in parallel computing, the communication overhead can be drastically reduced in comparison to the overall communication scheme. Thereby, we underline two important but different speed up factors: The first one is that the communication rate is reduced according to the decreasing migration probability determined by (6) and (7). On the other hand, as mentioned above, only a local communication takes place if the adaptation process is in the long convergence phase because of the exponentially decreasing neighborhood range σ_h. Hence, the several processors can communicate locally only with topological neighbors which reduces the communication overhead [13],[51],[60].

Application of the EA with SOM-like migration for clustering tasks

Clustering of dissimilarity data in psychotherapy research In the following we demonstrate the application of EAs for clustering of categorical data in psychology research. Thereby we will study some properties of the above introduced migration scheme. Beside the migration scheme here we also have to pay attention to the aspect of a proper choice of the fitness function for clustering of dissimilarities. The clustering of dissimilarities is recently studied in [9],[22],[26],[20] offering new perspectives for application. A respective energy function was found [56]:

$$\mathcal{F} = \frac{1}{2} \frac{\frac{1}{M \ln(c_{\max})}}{\left(1 - \frac{c_{\max}}{M}\right)} \sum_{i=1}^{M} \sum_{k=1}^{M} \frac{D_{i,k}}{M} \left(\sum_{v=1}^{c_{\max}} \frac{u_{i,v} u_{k,v}}{p_v} - 1 \right) \tag{9}$$

with $p_v = \sum_{i=1}^{M} u_{i,v}/M$ the normalized percentage of the data in that cluster and M the number of data and explicitly taking the number of clusters into account. The values $D_{i,k}$ are the *dissimilarities* between the data points. The $u_{i,v}$-values are the (fuzzy) cluster assignments. $\mathcal{F}$ serves as a cost function and has to be minimized. As pointed out in [26],[20] $\mathcal{F}$ is independent on non-symmetric dissimilarities and permutations of the cluster indices. Moreover, the cluster assignments $u_{i,v}$ are distributed according to the Gibbs distribution $\mathbf{P}(\mathcal{F}) = \exp\left(-\frac{\mathcal{F}-\mathbf{F}(\mathcal{F})}{T}\right)$whereby T plays the role of a temperature, and $\mathbf{F}(\mathcal{F})$ is the free energy. Yet, because of the statistical dependence of the assignments the Gibbs distribution $\mathbf{P}(\mathcal{F})$ can not be exactly rewritten in a factorized form, however a mean field approximation and corrections to the assignment correlations has been derived [26]. As it shown in [20] it is possible to minimize $\mathcal{F}$ by simulated annealing. Yet, it is difficult to generate a careful annealing scheme, because the of the large number of local minima. A possibility for convergence improvement is the application of evolutionary algorithms to optimize $\mathcal{F}$ [57].

In social science and psychology often categorical data have to be clustered which some times can be transformed into dissimilarity data [1]. As an example we consider the most frequently used method for acquisition of structures of interpersonal relationships in the area of psycho–dynamic psychotherapy research: the '*Core Conflictual Relationship Theme*' (CCRT) [34]. The method investigates short stories (episodes) about relationships which are often reported by the patients in their therapeutic sessions. These episodes are coded in components *wish of the subject*, *response of the object* and *response of the subject* using a standardized category system. We have $s^{w}_{\max} = 34$ *standard categories* $S^{w}_{j} \in \mathcal{S}^{w}$ to classify the wishes, $s^{ro}_{\max} = 30$ categories $S^{ro}_{j} \in \mathcal{S}^{ro}$ for encoding the response of both the object and subject [15], respectively. The categories are collected in $c^{w}_{\max} = 8$ *clusters* $C^{w}_{k} \in \mathcal{C}^{w}$, $c^{ro}_{\max} = 8$ clusters $C^{ro}_{k} \in \mathcal{C}^{ro}$ and $c^{ro}_{\max} = 8$ clusters $C^{rs}_{k} \in \mathcal{C}^{rs}$ according their meaning [4]. Several considerations have shown that *the used scheme of assignment still leads to unsatisfactory reliability rates* because the cluster are not well separated [1]. For solving the re-clustering problem of standard categories at first we determined the similarities between the standard categories S^{w}_{i}, S^{w}_{j} on the basis of their (symmetric) conditional probabilities $p^{w}(i \mid j)$ obtained from an external study [1] (in analogy $p^{ro}(i \mid j)$, $p^{rs}(i \mid j)$ for the responses of object and subject). Now the reclustering can be done as a clustering of proximity data using EAs as outlined above.

For comparison we computed $\mathcal{F}$ according to (9) for the original clusters

database	$\mathcal{F}$ original clusters	$\mathcal{F}$ EA clusters	$c_{\max}$ EA clusters
P^w	6.88	6.24	10
P^{ro}	6.28	6.16	9
P^{rs}	6.16	6.09	9

Table 1: Values for the weighted concordance coefficient $\tilde{\kappa}$ and the fitness function $\mathcal{F}$ according to the original clusters and the EA generated solutions.

$\mathcal{C}^w$, $\mathcal{C}^{ro}$ and $\mathcal{C}^{rs}$. Applying $\mathcal{F}$ from (9) as fitness measure in the EA we result cluster solutions with higher number of clusters as depicted in Tab. (1) and considerable better fitness values. We used in our computations $\mu_{\text{all}} = 400$ individuals which were evenly distributed onto $s = 10$ subpopulations. We trained the ensemble during $t_{\max} = 5000$ time steps. According to the proposed SOM-like migration scheme the subpopulations were arranged on a topological structure in the present application simply chosen as a ring. The characteristic time scale for decreasing the neighborhood between the subpopulation was defined as linear shrinking of $\sigma_h(t)$ in (6) with $\sigma_h(0) = \frac{s}{2}$ and $\sigma_h(t_{\max}) = 0.2$. The remaining probability was $\epsilon_h = 0.01$. The parameters were chosen in agreement with experiences in agreement with the adaptation dynamic in the neural network model [16]. Thereby, the initial σ-value is related to a quasi single big population whereas the final value corresponds to a migration between neighbored subpopulations on a small level. However, further studies should investigate the adaptation regime of these parameters with respect to the dynamic of the fitness within the subpopulations.

To study the influence of the parameter choices in the SOM-like migration scheme we considered several regimes of parameter cooling for σ_h exemplary in clustering of $\mathcal{S}^w$. The results are summarized in Fig. 2. In the first regime a) the value σ_h always was kept constant on a zero level which means a complete separated set of (relative small) subpopulations leading to a slow convergence and bad fitness value because of the small diversity within the subpopulations and the non-existing information flow all the time. In regime b) again the value σ_h always was kept constant but now on a high level: $\sigma_h = 3$. Hence, all subpopulations can be taken as a large single population. This accelerates the adaptation process especially in the first rough adaptation phase whereas in the final phase the tendency to homogeneity prevent good optimization results. In contradiction to regime b) in the regime c) σ_h was on a low level $\sigma_h = 0.2$ but again chosen to be constant. In opposite to a) the non-vanishing information flow leads to significant improvements in the last phase of the adaptation whereas the beginning is indicated by minor improvements as in a) but in contradiction to c). The last curve d) clearly demonstrates the advantages of the SOM-like regime of a decreasing $\sigma_h(t)$ with the above used parameter choices $\sigma_h(0) = 5$ and $\sigma_h(t_{\max}) = 0.2$. Finally, we can state that the SOM-like migration scheme generated in neighbored subpopulations cluster solutions which are judged by psychotherapists as similar [32][4]. However, a mathematical proof of

[4] This observation is not surprising from the view of neural network approach. However,

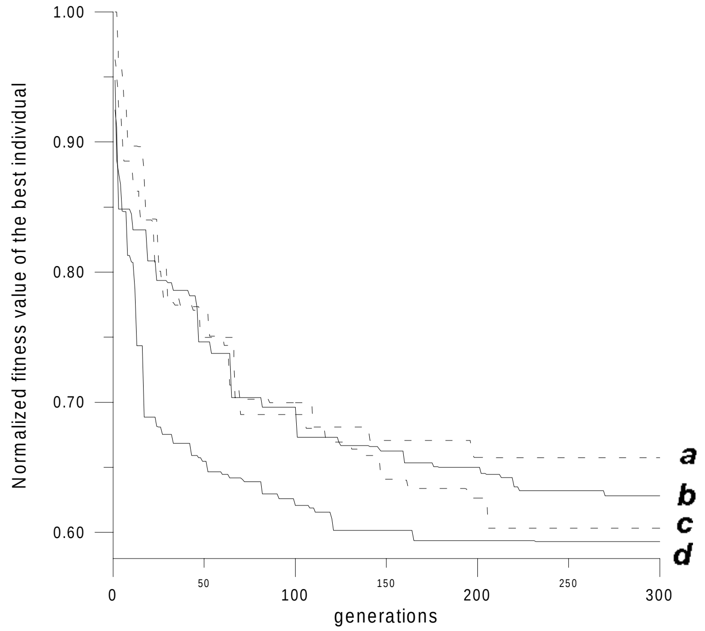

Figure 2: Development of the fitness of the best individual for mutiple subpopulation systems with different neighborhood range cooling strategies. (For further explanations - see text.)

this fact is difficult because of the complicate structure of the fitness function but should subject of further considerations as well as an improvement of the fitness function.

Application in VLSI model partitioning for logic simulation For the logic design of whole microprocessor structures time-extensive cycle-based simulation processes are necessary. Time spent for simulation can be drastically reduced using *parallel simulation* based on *model partitioning* which can be formulated as a combinational optimization problem characterized by a complex cost function [23] which estimates the run-time of one parallel simulation cycle of the corresponding hardware model parts. This cost function has to be minimized to reduce the expected parallel simulation time [25].

To achieve better partitions in shorter time we have parallelized the SOM-like EA. The migration is implemented by nonblocking Point-to-Point communication in the frame of the *Message Passing Interface* [51]. Here, the parameters was chosen again as in the above application, i.e. $\sigma_h(0) = \frac{s}{2}$, $\sigma_h(t_{\max}) = 0.2$ and $\epsilon_h = 0.01$.

it is in contrast to the investigations explained in [13].

Experimental results are given for an *IBM S/390* processor model which is partitioned into 15 blocks via the *STEP* pre-partitioning [23]. At the second hierarchy level three different EA strategies are realized with the parameters $\mu_{\text{all}} = 294$, $\lambda = 700$ in each run. A sequential one-population EA is opposed to two parallel multiple subpopulation EAs each consisting of $s = 7$ subpopulations, i.e., $\tilde{\mu} = 42$, $\tilde{\lambda} = 100$ for each subpopulation which corresponds to a selection pressure of $\rho \approx 2.4$. We compare the SOM-like migration scheme with an *all-to-all communication scheme* where the communication effort is much higher than for the SOM-like case [52]. Considering the partitioning effort for the multiple subpopulation approach, in Fig.3 the fitness of the best individual is plotted over the time t_{part} spent for the EA partitioning.

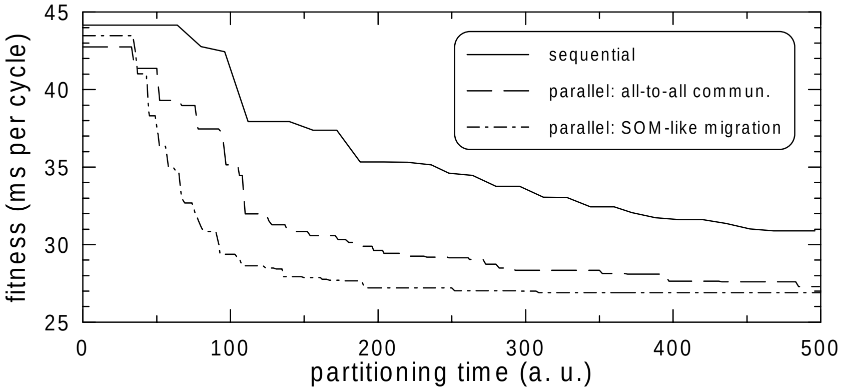

Figure 3: Fitness of the best individual of all subpopulations comparing a sequential run with two parallel ones in dependence on the *partitioning time* t_{part} measured in arbitrary units (a. u.).

Parallel partitioning drastically reduces the partitioning time because the partitioning effort is distributed to 7 processors. The fitness of the best individual significantly faster decreases than in the sequential case. But the all-to-all communication scheme is accompanied by a high communication overhead. Using our SOM-like migration approach this communication effort can be reduced in such a way that better individuals (partitions) are obtained in shorter partitioning time t_{part}. So, in the example discussed here an estimated run-time of $27\,ms$ per cycle (fitness) is reached in the half of the partitioning time comparing to the all-to-all communication scheme which is in agreement with results obtained in [13]. Although the individuals of our initial population are already equipped with expert knowledge and not randomly produced, EAs yield a reduction of estimated run-time from $\approx 44\,ms$ down to $27\,ms$ (see Fig.3).

4. Concluding remarks

We reviewed the actual development in hybrid systems for combination of ANNs and EAs. The most publications deal with ANN optimization with respect to the network topology while weight adaptation is not expansively considered. For the backward direction we can state some recently developed approaches incorporating neural network paradigms of SOMs into evolution dynamic of EAs.

References

[1] C. Albani, T. Villmann, B. Villmann, A. Körner, M. Geyer, D. Pokorny, G. Blaser, and H. Kächele. Kritik und erste Reformulierung der kategorialen Strukturen der Methode des Zentralen Beziehungs–Konflikt–Themas (ZBKT). *Psychotherapie, Psychosomatik und Medizinische Psychologie*, 49(11):408–421, 1999.

[2] P. Angeline, G. Saunders, and J. Pollak. An evolutionary algorithm that constructs recurrent neural networks. *IEEE Transactions on Neural Networks*, 5:54–65, 1994.

[3] T. Bäck. *Evolutionary Algorithms in Theory and Practice.* Oxford University Press, New York - Oxford, 1996.

[4] J. Barber, P. Crits-Christoph, and L. Luborsky. A guide to the CCRT Standard Categories and their classification. In L. Luborsky and P. Crits-Chrostoph, editors, *Understanding Transference*, pages 37–50. Basic Books New York, 1990.

[5] H.-U. Bauer, M. Herrmann, and T. Villmann. Neural maps and topographic vector quantization. *Neural Networks*, 12(4–5):659–676, 1999.

[6] H.-U. Bauer and K. R. Pawelzik. Quantifying the neighborhood preservation of Self-Organizing Feature Maps. *IEEE Trans. on Neural Networks*, 3(4):570–579, 1992.

[7] H.-U. Bauer and T. Villmann. Growing a Hypercubical Output Space in a Self–Organizing Feature Map. *IEEE Transactions on Neural Networks*, 8(2):218–226, 1997.

[8] S. Bornholdt and D. Graudenz. General asymmetric neural networks and structure design by genetic algorithms. *Neural Networks*, 5:327–334, 1992.

[9] J. Buhmann and H. Kühnel. Vector quantization with complexity costs. *IEEE Transactions on Information Theory*, 39:1133–1145, 1993.

[10] E. Cantu-Paz. Designing scalable multi-population parallel genetic algorithms. Technical Report 98009, Illinois Genetic Algorithms Laboratory, University of Illionois at Urbana-Champaign, Urbana IL, 1998.

[11] E. Cantu-Paz. Migration policies and takeover times in genetic algorithms. In W. Banzhaf, J. Daida, A. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. Smith, editors, *Proc. Of the Genetic and Evolutionary Computation Conference GECCO-1999*, page 775, San Franzisco, CA, 1999. Morgan Kaufmann.

[12] E. Cantu-Paz. Topology, migration rates and multi-population parallel genetic algorithms. Technical Report 99007, Illinois Genetic Algorithms Laboratory, University of Illionois at Urbana-Champaign, Urbana IL, 1999.

[13] E. Cantu-Paz. Topology, migration rates and multi-population parallel genetic algorithms. In W. Banzhaf, J. Daida, A. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. Smith, editors, *Proc. Of the Genetic and Evolutionary Computation Conference GECCO-1999*, pages 91–98, San Franzisco, CA, 1999. Morgan Kaufmann.

[14] I. Chalkiadakis, G. Rovithakis, and M. Zervakis. A structural gentic algorithm to optimize high order neural networks architecture. In M. Verleysen, editor, *Proc. Of European Symposium on Artificial Neural Networks (ESANN'2001)*, Brussels, Belgium, 2001. D facto publications.

[15] P. Crits-Christoph and A. Demorest. List of standard categories (edition 2). In L. Luborsky and H. Kächele, editors, *Der Zentrale Beziehungskonflikt - ein Arbeitsbuch.* PSZ Verlag, Ulm, Germany, 1988.

[16] R. Der and T. Villmann. Dynamics of self-organized feature mapping. In J. Mira, J. Cabestany, and A. Prieto, editors, *New Trends in Neural Computation. International Workshop on Artificial Neural Networks. IWANN '93 Proceedings*, pages 312–15, Berlin, Germany, 1993. Springer-Verlag.

[17] D. Fogel, L. Fogel, and V. Porto. Evolving neural networks. *Biological Cybernetics*, 63:487–493, 1990.

[18] B. Fritzke. Growing cell structures—a self-organizing network in k dimensions. In I. Aleksander and J. Taylor, editors, *Artificial Neural Networks, 2*, volume II, pages 1051–1056, Amsterdam, Netherlands, 1992. North-Holland.

[19] J. Goppert and W. Rosenstiel. The continuous interpolating self-organizing map. *Neural Processing Letters*, 5(3):185–92, 1997.

[20] T. Greapel and K. Obermayer. A stochastic self-organizing map for proximity data. *Neural Computation*, 11(1):139–155, 1999.

[21] A. Hämäläinen. Using genetic algorithm in self-organizing map design. In D. Pearson, N. Steele, and R. Albrecht, editors, *Artificial Neural Nets and Genetic Algorithms - Proc. of ICANNGA'95*, pages 364–367. Springer-Verlag, 1995.

[22] M. Hasenjäger, H. Ritter, and K. Obermayer. Active data selection for fuzzy topographic mapping of proximities. In G. Brewka, R. Der, S. Gottwald, and A. Schierwagen, editors, *Fuzzy-Neuro-Systems'99 (FNS'99), Proc. Of the FNS-Workshop*, pages 93–103, Leipzig, 1999. Leipziger Universitätsverlag.

[23] R. Haupt, K. Hering, U. Petri, and T. Villmann. Hierarchical model partitioning for parallel VLSI–simulation using evolutionary algorithms improved by superpositions of partitions. In K. Lieven, editor, *Proceedings of European Congress on Intelligent Techniques and Soft Computing (EUFIT'97)*, pages 804–808, Aachen, Germany, 1997.

[24] S. Haykin. *Neural Networks - A Comprehensive Foundation.* IEEE Press, New York, 1994.

[25] K. Hering, R. Haupt, and T. Villmann. Hierarchical Strategy of Model Partitioning for VLSI–Design Using an Improved Mixture of Experts Approach. In *Proc. Of the Conference on Parallel and Distributed Simulation (PADS'96)*, pages 106–113. IEEE Computer Society Press, Los Alamitos, 1996.

[26] T. Hofmann and J. Buhmann. Pairwise data clustering by deterministic annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(1):1–14, 1997.

[27] J. Huhse and A. Zell. Evolutionary strategy with neighborhood attraction. In H. Bothe and R.Rojas, editors, *Neural Computation 2000*, pages 363–369, Zürich, 2000. ICSC Academic Press.

[28] J. Huhse and A. Zell. Investigating the influence of the neighborhood attraction factor to the evolution strategies with neighborhood attraction. In M. Verleysen, editor, *Proc. Of European Symposium on Artificial Neural Networks (ESANN'2001)*, Brussels, Belgium, 2001. D facto publications.

[29] K. Kalveram. Motor control and movement optimization learned by combining auto-imitative and genetic algorithms. In M. Verleysen, editor, *Proc. Of European Symposium on Artificial Neural Networks (ESANN'2001)*, Brussels, Belgium, 2001. D facto publications.

[30] T. Kohonen. *Self-Organizing Maps.* Springer, Berlin, Heidelberg, 1995. (Second Extended Edition 1997).

[31] T. Kohonen. Fast evolutionary learning with batch-type self-organizing maps. *Neural Processing Letters*, 9(2):25–34, 1999.

[32] A. Körner. *Kategorisierung Von Bezeihungsschemata Mi der Methode "Das Zentrale Beziehungskonfliktthema".* PhD thesis, University Leipzig, Germany, 2000.

[33] S. Kovács, G. J. Tóth, R. Der, and A. Lőrincz. Output sensitive discretization for genetic algorithm with migration. *Neural Network World*, 6:79–88, 1996.

[34] L. Luborsky. The core conflictual relationship scheme. In N. Freedman and S. Grand, editors, *Communicative Structure and Psychic Structures.* Plenum Press New York, 1977.

[35] V. Maniezzo. Genetic evolution of topology and weight distribution of neural networks. *IEEE Transactions on Neural Networks*, 5:39–53, 1994.

[36] T. Martinetz and K. Schulten. Topology representing networks. *Neural Networks*, 7(3):507–522, 1994.

[37] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer–Verlag Berlin Heidelberg New York, third, revised and extended edition, 1996.

[38] H. Mühlenbein. Limitations of multi-layer percepton networks-steps towards genetic neural networks. *Parallel Computing*, 14:249–260, 1990.

[39] H. Mühlenbein, M. Gorges-Schleuter, and O. Krämer. Evolution Algorithm in Combinatorial Optimization. *Parallel Computing*, (7):65–88, 1988.

[40] A. Nissinen and H. Hytyniemi. Evolutionary self-organizing map. In H.-J. Zimmermann, editor, *Proceedings 6th European Congress on Intelligent Techniques Soft Computing (EUFIT'98)*, volume 3, pages 1596–1600. Verlag Mainz, Aachen, Germany, 1998.

[41] S. Pal, D. Bhandari, P. Harish, and M. Hundu. Cellular neural networks, genetic algorithms and object extraction. *Far East Journal of Mathematical Science*, 1:139–155, 1993.

[42] S. Pal, S. De, and A. Ghosh. Designing Hopfield type neural networks using genetic algorithms and its comparison with simulated annealing. *International Journal of Pattern Recognition and Artificial Intelligence*, 11:447–461, 1997.

[43] S. Pal and S. Mitra. *Neuro-Fuzzy Pattern Recognition.* Wiley Series on Intelligent Systems. Wiley Sons, New York, 1999.

[44] H. Pohlheim. *Evolutionäre Algorithmen.* Springer-Verlag, 1999.

[45] D. Polani. On the optimization of self-organizing maps by genetic algorithms. In E. Oja and S. Kaski, editors, *Kohonen Maps*, number ISBN 951-22-3589-7, pages 157–170, Amsterdam (Holland), June 1999. Helsinki, Elsevier.

[46] D. Polani and T. Uthmann. Adaptation of kohonen feature map topologies by genetic algorithms. In R. Mnner and B. Manderick, editors, *Parallel Problem Solving from Nature*, volume 2, pages 421–429. Elsevier, 1992.

[47] H. Ritter. Parametrized self-organizing maps. In S. Gielen and B. Kappen, editors, *Proc. ICANN'93 Int. Conf. on Artificial Neural Networks*, pages 568–575, London, UK, 1993. Springer.

[48] P. Robbins, A. Soper, and K. Rennolls. Use of genetic algorithms for optimal topology determination in back-propagation neural networks. In *Artificial Neural Networks and Genetic Algorithms - Proc. of ICANNGA'93 Vienna*, pages 726–730. Springer-Verlag, 1993.

[49] S. Saha and J. Christensen. Genetic design of sparse feedforward neural networks. *Information Science*, 79:191–200, 1994.

[50] J. Schaffer, R. Caruana, and L. Eshelman. Using genetic search to exploit the emergent behavior of neural networks. *Physica D*, 42:244–288, 1990.

[51] H. Schulze. Entwicklung, Untersuchung und Implementierung von Parallelen Evolutionären Algorithmen für die Modellpartitionierungskomponente parallelMAP. Master's thesis, University Leipzig, 1998.

ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 137-152

[52] H. Schulze, R. Haupt, and K. Hering. Experiments in parallel evolutionary partitioning. In H. Sips and E. D'Hollander, editors, *Proc. Of Conference on Parallel Computing 1999*, Delft, Netherlands, 1999. Imperial Collage Press.

[53] U. Seiffert. Multiple layer perceptron learning using genetic algorithms. In M. Verleysen, editor, *Proc. Of European Symposium on Artificial Neural Networks (ESANN'2001)*, Brussels, Belgium, 2001. D facto publications.

[54] S. Sin and R. deFigueiredo. An evolution-oriented learning algorithm for the optimal interpolative net. *IEEE Transactions on Neural Networks*, 3:315–323, 1992.

[55] G. J. Tóth and A. Lőrincz. Genetic algorithm with migration on topology conserving maps. *Neural Networks World*, 2:171–181, 1995.

[56] T. Villmann. Application of evolutionary algorithms for clustering of non-metric data in medicine. In G. Brewka, R. Der, S. Gottwald, and A. Schierwagen, editors, *Fuzzy-Neuro-Systems'99 (FNS'99), Proc. Of the FNS-Workshop*, pages 155–165, Leipzig, 1999. Leipziger Universitätsverlag.

[57] T. Villmann. Neural networks approaches in medicine – a review of actual developments. In *Proc. Of European Symposium on Artificial Neural Networks (ESANN'2000)*, pages 165–176, Brussels, Belgium, 2000. D facto publications.

[58] T. Villmann. Evolutionary algorithms with subpopulations using a neural network like migration scheme and its application to real world problems. *Integrated Computer- Aided Engineering*, page to appear, 2001.

[59] T. Villmann, R. Der, M. Herrmann, and T. Martinetz. Topology Preservation in Self–Organizing Feature Maps: Exact Definition and Measurement. *IEEE Transactions on Neural Networks*, 8(2):256–266, 1997.

[60] T. Villmann, R. Haupt, K. Hering, and H. Schulze. Parallel evolutionary algorithms with som-like migration. In A. Dobnikar, N. Steele, D. W. Pearson, and R. Albrecht, editors, *Artificial Neural Networks and Genetic Algorithms (Proc. Of ICANNGA'99)*, pages 274–279, Wien - New York, 1999. Springer-Verlag.

[61] B. Whitehead and T. Choate. Cooperative-competitive gentic evolution of radial basis function centers and widths for time series prediction. *IEEE Transactions on Neural Networks*, 7:869–880, 1996.

[62] D. Whitley, T. Starkweather, and C. Bogart. Genetic algorithms and neural networks: Optimizing connections and connectivity. *Parallel Computing*, 14:347–361, 1990.

[63] X. Yao. A review of evolutionary artificial neural networks. *International Journal of Intelligent Systems*, 8:539–.567, 1993.

[64] Q. Zhao and T. Higuchi. Evolutionary learning of nearest neighbor mlp. *IEEE Transactions on Neural Networks*, 7(3):762–767, 1996.