EVOLUTIONARY MULTI-OBJECTIVE OPTIMIZATION OF NEURAL NETWORKS FOR FACE DETECTION

STEFAN WIEGAND and CHRISTIAN IGEL

Institut für Neuroinformatik, Ruhr-Universität Bochum,
44780 Bochum, Germany

UWE HANDMANN

Viisage Technology AG,
44801 Bochum, Germany

For face recognition from video streams speed and accuracy are vital aspects. The first decision whether a preprocessed image region represents a human face or not is often made by a feed-forward neural network (NN), e.g., in the Viisage-FaceFINDER® video surveillance system. We describe the optimization of such a NN by a hybrid algorithm combining evolutionary multi-objective optimization (EMO) and gradient-based learning. The evolved solutions perform considerably faster than an expert-designed architecture without loss of accuracy. We compare an EMO and a single objective approach, both with online search strategy adaptation. It turns out that EMO is preferable to the single objective approach in several respects.

Keywords: evolutionary algorithms, face detection, neural networks, model selection, multi-objective optimization, pattern recognition, structure optimization, strategy adaptation

1. Introduction

Face detection is usually the first step in face recognition for biometric authentication. The Viisage-FaceFINDER® video surveillance system[1] automatically identifies people by their faces in a three step process: first, regions of the video stream that contain a face are detected, then specific face models are calculated, and finally these models are compared with a database. The final face modeling and recognition is done using *Hierarchical Graph Matching* (*HGM*),[2] which is an improvement of the *Elastic Graph Matching* method.[3] It is inspired by human vision and highly competitive to other techniques for face recognition.[4] To meet real-time constraints, the Viisage-FaceFINDER® requires very fast and accurate image classifiers within the detection unit for an optimal support of HGM. In the detection step, different biologically motivated cues are fused to cluster the given images into regions of high and low significance similar to an approach for dynamic scene analysis in road

traffic.[5] The clusters of high significance are then classified as either containing or not containing an upright frontal face by a task specific feed-forward neural network (NN).[6]
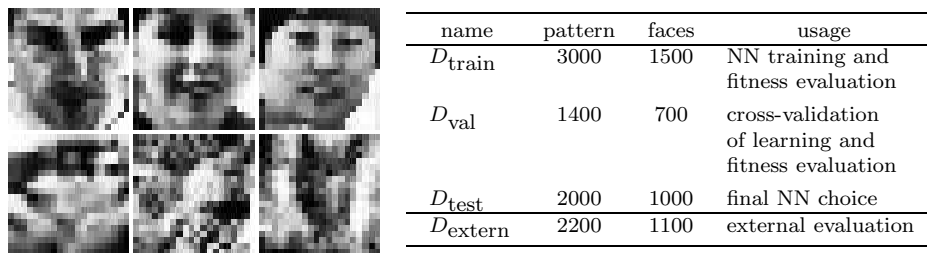
We address the task of optimizing the weights and the structure of the face detection NN in the Viisage-FaceFINDER® system. The first goal is to increase the speed of the neural classifier, because faster classification allows for a higher scanning rate leading to more accurate recognition. In addition, we would like to enhance the classification accuracy of the NN. This optimization problem can be tackled by evolutionary computation, which has become an established method for the design of NNs. We have already shown that NNs for face detection adapted by an hybrid algorithm combining recent developments from evolutionary and gradient-based optimization can outperform standard expert topologies proposed in the literature.[7] In our previous work, we simplified the optimization problem by transforming the two objectives, speed and NN classification accuracy, to a single one by weighted aggregation. In this approach, we had to fix the trade-off between speed and accuracy in advance by choosing the weighting factors of the objectives.

Increasing the speed and improving the accuracy of the classifier are two different, not necessary overlapping aims. Hence, we assume that our NN design problem is a—non-trivial—multi-objective optimization task. Advanced evolutionary multi-objective optimization (EMO) considers vector-valued objective functions, where each component corresponds to one objective. Such methods are capable of finding sets of trade-off solutions that give an overview of the space of possible solutions.[8,9] From such a set one can select an appropriate compromise, which might not have been found by a single-objective approach. Recently, EMO has been applied to the design of NNs.[10,11] In the following, we combine the basic idea of the vector-valued selection scheme from *NSGA-II*[12] with our hybrid algorithm for adapting NNs for face detection. The performance of the new approach is compared to our previous results. It turns out that the advanced multi-objective optimization is preferable to the approach using linear aggregation in several respects.

The article is organized as follows. In section 2 we briefly describe the optimization problem and the hybrid optimization algorithm with single- and multi-objective selection, respectively. This includes the description of techniques for online *operator adaptation* in both cases. In section 3 we will highlight the issue of performance assessment in the multi-objective framework. Afterwards, the empirical setup for the comparison of our algorithms is summarized. In section 4 we report on the experimental results. Finally, in section 5 we discuss the results and draw conclusions.

2. Structure Optimization of Neural Networks for Face Detection

In this section, we first briefly describe the optimization problem of improving the NN for face detection. Then we present our evolutionary optimization algorithm with single- and multi-objective selection, respectively.

name	pattern	faces	usage
D_{train}	3000	1500	NN training and fitness evaluation
D_{val}	1400	700	cross-validation of learning and fitness evaluation
D_{test}	2000	1000	final NN choice
D_{extern}	2200	1100	external evaluation

Fig. 1. Left, the input to the face detection NN are preprocessed 20×20 pixel grayscale images showing either frontal, upright face (positive) and nonface (negative) examples. The preprocessing comprises rescaling, lighting correction, and histogram equalization. Right, for optimization and evaluation we have partitioned the available patterns into 4 disjoint data sets.

2.1. *Optimization Problem*

Feed-forward NNs have proven to be powerful tools in pattern recognition.[13] Especially in the domain of face detection the competitiveness of NNs is widely accepted. As stated in a recent survey "The advantage of using neural networks for face detection is the feasibility of training a system to capture the complex class conditional density of face patterns. However, one drawback is that the NN architecture has to be extensively tuned (number of layers, number of nodes, learning rates, etc.) to get exceptional performance".[14] This drawback is addressed by the hybrid optimization algorithm used in this study.

In the Viisage-FaceFINDER® system the inputs to the face detection NN are preprocessed 20×20 pixel grayscale images. These show either frontal, upright faces or nonfaces, see Fig. 1 (left). The preprocessing comprises rescaling, lighting correction, and histogram equalization. The assumption of fixed-size subimages as input to the classifier meets the realistic application scenario for the NN in the Viisage-FaceFINDER® system, although in the survey on face detection by Hjelmas and Low such input patterns are regarded as unrealistic for real world face detection.[15] In Viisage-FaceFINDER® the NN is only a part of a sophisticated face detection module, and its main task is to support the time consuming HGM procedure with appropriate face images.

In the efficient and hardware-friendly implementation of the face detection NN within Viisage-FaceFINDER® the speed of the classification scales approximately linearly with the number of hidden neurons and not with the number of connections. With every hidden neuron that is saved the detection costs are reduced by approximately one percentage point. Hence, the primary goal of the optimization is to reduce the number of hidden nodes of the detection NN under the constraint that the classification error should not increase. The second objective is to even improve the classification accuracy. These are possibly conflicting objectives. In particular "the smaller the network the better the generalization" does generally not hold.[16,17]

The data available for optimizing the face detection NN are split into a training D_{train}, validation D_{val}, and test D_{test} data set, see Fig. 1 (right). The reason for
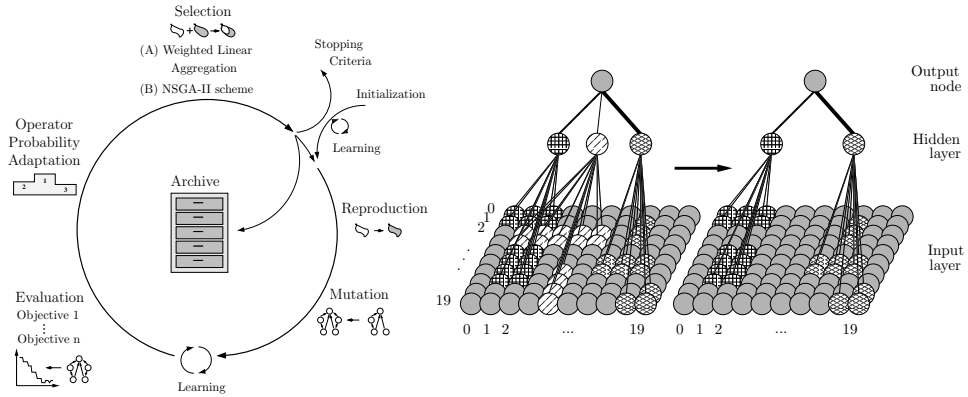
Fig. 2.   Left, the hybrid evolutionary algorithm in conjunction with two different selection variants (A) and (B), see text. Right, scheme of the delete-node operator. The linewidths indicate the magnitude of the corresponding weight values. The picture also visualizes the NN input dimension and the receptive field connectivity.

this partitioning will become apparent when we discuss our optimization algorithm below. In addition, there is a data set D_{extern}, which is not used during optimization, for the final evaluation of face detectors.

2.2. *Evolutionary algorithm*

Evolutionary algorithms have become established methods for the design of neural networks, especially for adapting their topology.[18,19] They are less prone to getting stuck in local optima compared to greedy algorithms like pruning or constructive methods.[20,21]

The basic optimization loop of our hybrid evolutionary algorithm is shown in Fig. 2 (left). This scheme might be regarded as canonical evolutionary NN optimization using direct encoding, nested learning, and Lamarckian inheritance. However, there are some special features described in the following.

We start with an explanation of how the first parent population is initialized. Then we sketch how offspring are created and mutated. After that, we outline the nested gradient learning procedure within the evolutionary loop. Then we highlight the two different approaches to selection considered in this work. The section ends with the description of the online strategy adaptation method for adjusting the operator probabilities.

2.2.1. *Initialization*

The comparison of our results will be held on the basis of the expert-designed 400-52-1 NN architecture, the *reference topology*, proposed by Rowley et al.[22] This NN has been tailored to the face detection task and has become a standard reference for NN based face detection.[14] We initialize the parent population with 25

individuals all representing the *reference topology* with different random weight initializations as done in the work of Wiegand et al.[7] The 400 inputs correspond to the pixels of the preprocessed image patterns, cf. Fig. 2 (right) and Fig. 1 (left). No hidden neuron is fully connected to the input but to certain receptive fields, see below. The total number of connections amounts to 2905. This is in contrast to more than 21,000 in a fully connected NN with an equal number of hidden neurons.

2.2.2. *Inheritance and variation*

Each parent creates one child per generation by reproduction. The offspring is then mutated by elemental variation operators. These are chosen randomly for each offspring from a set Ω of operators and are applied sequentially. The process of choosing and applying an operator is repeated $1 + x$ times, where x is an individual realization of a Poisson distributed random number with mean 1.

All operators are implemented such that their application always leads to valid NN graphs. A NN graph is considered to be *valid* if each hidden node lies on a path from an input unit to an output unit and there are no cycles. Further, the layer restriction, here set to a single hidden layer, has to be met. All new weight values are drawn uniformly from the interval $[-0.05, 0.05]$. There are 5 basic operators: *add-connection*, *delete-connection*, *add-node*, *delete-node*, and *jog-weights*:

add-connection   A connection is added to the NN graph.

delete-connection   This operator is inspired by *magnitude based pruning*. The operator is rank-based as discussed by Braun.[23] The connections of the NN are sorted by the absolute value of the corresponding weights. The connection with rank number r given by

$$r := \lfloor W \cdot (\eta_{\max} - \sqrt{(\eta_{\max}^2 - 4 \cdot (\eta_{\max} - 1) \cdot u)})/(2 \cdot (\eta_{\max} - 1)) \rfloor \quad (1)$$

is deleted, so that connections with smaller weight have a higher probability of being removed. Here $\lfloor x \rfloor$ denotes the largest integer smaller than x, W the number of weights, and $u \sim \mathcal{U}[0, 1]$ is a random variable uniformly distributed on $[0, 1]$. The parameter $1 < \eta_{\max} \leq 2$ controls the influence of the rank and is set to its maximum value.[24]

add-node   A hidden node with bias parameter is added to the NN and connected to the output. For each input, a connection to the new node is added with probability $p_{\text{in}} = 1/16$.

delete-node   In this rank-based node deletion operator, the hidden nodes are ordered according to their maximum output weight. The maximum output weight of a node i is given by $\max_j |w_{ji}|$, where w_{ji} is the weight of the connection from node i to node j. The nodes are selected based on (1), such that nodes with smaller maximum output weight values have a higher probability of deletion. If node k is deleted, all connections to or from k are removed, cf. Fig. 2 (right).

jog-weights This operator adds Gaussian noise to the weights in order to push the weight configuration out of local minima and thereby to allow the gradient-based learning to explore new regions of the weight space. Each weight value is varied with constant probability $p_{\text{jog}} = 0.3$ by adding normally distributed noise with expectation 0 and standard deviation $\sigma_{\text{jog}} = 0.01$.

In addition to the 5 basic operators, there are 3 task-specific mutations inspired by the concept of "receptive fields", i.e., dimensions of the input space that correspond to rectangular regions of the input image, cf. Fig. 2 (right). The RF-operators *add-RF-connection*, *delete-RF-connection*, and *add-RF-node* behave as their basic counterparts, but act on groups of connections. They consider the topology of the image plane by taking into account that "isolated" processing of pixels is rarely useful for object detection. The RF-operators are defined as follows:

add-connection-RF A valid, not yet existing connection, say from neuron i to j, is selected uniformly at random. If the source i is not an input, the connection is directly added. Otherwise, a rectangular region of the 20×20 image plane containing between 2 and $M = 100$ pixels including the one corresponding to input i is randomly chosen. Then neuron j is connected to all the inputs corresponding to the chosen image region.

delete-connection-RF An existing connection that can be removed, say from node i to j, is selected at random. If the source i is not an input, the connection is directly deleted. Otherwise, a decision is made whether a horizontal or vertical receptive field is deleted. Assume that a horizontal field is removed. Then *delete-connection-RF$_x$*(i, j) is applied recursively to remove the inputs from a connected pixel row:

delete-connection-RF$_x$ (i, j) Let (i_x, i_y) be the image coordinates of the pixel corresponding to the input i. The connection from i to j is deleted. If hidden node j is also connected to the input node k corresponding to pixel $(i_x + 1, i_y)$, *delete-connection-RF$_x$*(k, j) is applied. If j is connected to node l corresponding to $(i_x - 1, i_y)$, then the operator *delete-connection-RF$_x$*(l, j) is called.

Deletion of a vertical receptive field (i.e., a connected pixel column) is done analogously.

add-node-RF A hidden node with bias connection is added and connected to the output and a receptive field as done in the *add-connection-RF* operator.

2.2.3. *Embedded learning*

Let $\text{MSE}_a(D)$ and $\text{CE}_a(D)$ be the mean squared error and the classification error in percent on data set D of the NN represented by individual a and let $n_{\text{hidden}}(a)$ and $n_{\text{weights}}(a)$ be the corresponding number of hidden neurons and weights, respectively. The weights of every newly generated offspring a are adapted by gradient-

based optimization ("learning","training") of $\mathrm{MSE}_a(D_{\mathrm{train}})$. An improved version of the Rprop[25,26] algorithm is used for at most 100 iterations of training. This learning method solves the problem of choosing the learning rate by automatically adjusting individual step-sizes for each parameter to adapt. Training can stop earlier due to the *generalization loss* criterion GL_α as described by Prechelt.[27] The generalization loss is computed on D_{val} for $\alpha = 5$. Finally, the weight configuration with the smallest $\mathrm{MSE}_a(D_{\mathrm{train}}) + \mathrm{MSE}_a(D_{\mathrm{val}})$ encountered during training is regarded as the outcome of the training process and stored in the genome of the individual a.

2.2.4. *Evaluations and selection in presence of multiple objectives*

We are looking for sparse NNs with high classification accuracy. That is, we try to optimize two different objectives. There are several ways of dealing with multiple goals and we describe two of them in the following.

(A) Linearly aggregated objectives are subject to selection. In the first case the algorithm in Fig. 2 uses a scalar fitness $\Phi(a)$ for any individual a given by the weighted linear aggregation

$$\Phi(a) := \quad \gamma_{\mathrm{CE}} \cdot \mathrm{CE}_a^{(t)}(D_{\mathrm{train}} \cup D_{\mathrm{val}}) + \mathrm{MSE}_a^{(t)}(D_{\mathrm{train}} \cup D_{\mathrm{val}})$$
$$+\gamma_{\mathrm{hidden}} \cdot n_{\mathrm{hidden}}^{(t)}(a) \qquad + \gamma_{\mathrm{weights}} \cdot n_{\mathrm{weights}}^{(t)}(a) \tag{2}$$

supposed to be minimized. The weighting factors are chosen such that typically $\gamma_{\mathrm{CE}} \cdot \mathrm{CE}_a^{(t)}(D_{\mathrm{train}} \cup D_{\mathrm{val}}) \gg \gamma_{\mathrm{hidden}} \cdot n_{\mathrm{hidden}}^{(t)}(a) \approx \gamma_{\mathrm{weights}} \cdot n_{\mathrm{weights}}^{(t)}(a) \gg \mathrm{MSE}_a^{(t)}(D_{\mathrm{train}} \cup D_{\mathrm{val}})$ holds. Note that in our application we tolerate an increase in the number of connections as long the number of neurons decreases.

Let $\mathcal{O}^{(t)}$ contain all offspring produced at generation t by the parent population $\mathcal{P}^{(t)}$. Based on the fitness Φ EP-style tournament selection[28] with 5 opponents is applied to determine the parents $\mathcal{P}^{(t+1)}$ for the next generation from $\mathcal{P}^{(t)} \cup \mathcal{O}^{(t)}$.

(B) Vector-valued objectives are subject to selection. In the second case the evolutionary algorithm in Fig. 2 performs advanced EMO. It uses a selection method based on the Fast Non-Dominated Sorting Genetic Algorithm (*NSGA-II*).[12]

We first map the elements of the genotype space (decision space) to n-dimensional real-valued vectors $\boldsymbol{z} = (z_1, \ldots, z_n)$ of the objective space. In our case we map the individual a that has already finished training to the vector $\boldsymbol{z}_a = (n_{\mathrm{hidden}}(a), \mathrm{CE}_a(D_{\mathrm{train}} \cup D_{\mathrm{val}}))$. Both objective components are subject to minimization. The elements of the objective space are partially ordered by the dominance relation $\succcurlyeq$ ($\boldsymbol{z}$ dominates $\boldsymbol{z}'$) that is defined by

$$\boldsymbol{z} \succcurlyeq \boldsymbol{z}' \in \mathbb{R}^n \quad \Leftrightarrow \quad \forall \, 1 \leq i \leq n : \; z_i \leq z_i' \; \wedge \; \exists \, 1 \leq j \leq n : \; z_j < z_j' \tag{3}$$

stating that vector $\boldsymbol{z}$ performs better than $\boldsymbol{z}'$ iff $\boldsymbol{z}$ is as least as good as $\boldsymbol{z}'$ in all objectives and better with respect to at least one objective. Considering a set M
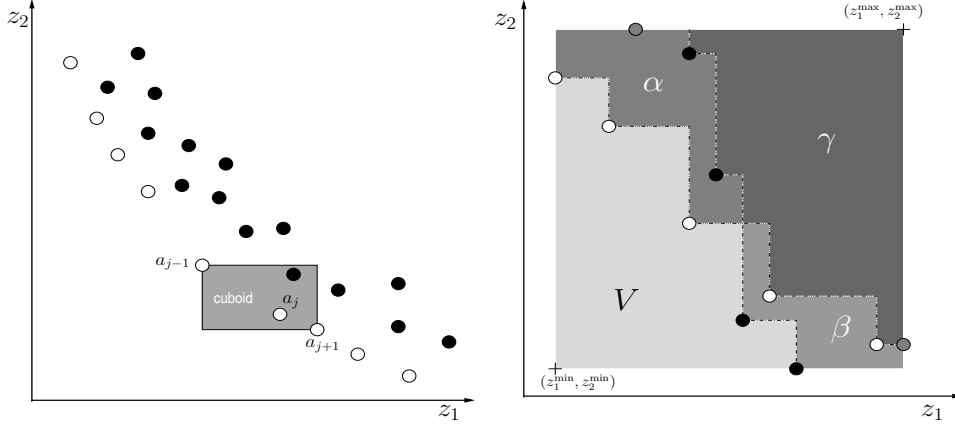
Fig. 3.   The figure on the left illustrates how the crowding distance[9] $C(a_j)$ is computed. The black dots are the elements of M_{i+1} and the white dots of belong to the Pareto front P_{M_i}. The figure on the right depicts the union of two different Pareto fronts P (white dots) and P' (black dots). The hypervolumes[29] and coverage differences of P and P', respectively, are given by $H_P = (\alpha + \gamma)/V$, $H_{P'} = (\beta + \gamma)/V$, $D_{P,P'} = \alpha/V$, and $D_{P',P} = \beta/V$. The grey dots do not necessarily belong to one of the two Pareto fronts shown in the figure but define the size of the reference area.

of n-dimensional vectors, the subset $P_M \subseteq M$ consisting only of those vectors that are not dominated by any other vector of M is called the Pareto front of M. As in the *NSGA-II* (environmental) selection scheme, we first assign to each individual $a \in \mathcal{P}^{(t)} \cup \mathcal{O}^{(t)}$ a rank value $R^{(t)}(a)$ based on its degree of non-domination in objective space. We define the chain of subsets M_i, $i \in \mathbb{N}$, by $M_1 \supseteq M_2 := M_1 \backslash P_{M_1} \supseteq M_3 := M_2 \backslash P_{M_2} \supseteq \ldots$, where $A \backslash B$ denotes the portion of the set A that is not part of set B. Then the rank operator $R^{(t)}(a)$ assigns each individual $a \in \mathcal{P}^{(t)} \cup \mathcal{O}^{(t)}$ the index i of the corresponding Pareto front P_{M_i} that includes the objective vector of a. Furthermore the NSGA-II ranking takes the diversity of the population (in the objective space) into account. The diversity is measured by the crowding distance $C(a)$, the size of the largest cuboid (precisely the sum of its edges) in objective space enclosing the vector $\boldsymbol{z}_a$, $a \in P_{M_i}$, but no other objective vector from P_{M_i}, see Fig. 3 (left). Then all individuals $a \in \mathcal{P}^{(t)} \cup \mathcal{O}^{(t)}$ are sorted in ascending order according to the partial order $\geq_n$ defined by

$$a_i \geq_n a_j \Leftrightarrow \left(R^{(t)}(a_i) < R^{(t)}(a_j) \right) \vee \left(R^{(t)}(a_i) = R^{(t)}(a_j) \wedge C(a_i) \geq C(a_j) \right) \quad (4)$$

and the first $|\mathcal{P}|$ individuals form the new parent population $\mathcal{P}^{(t+1)}$. We refer to the described selection method as *NSGA-II selection* throughout this article.

2.2.5. *Search strategy adaptation: Adjusting operator probabilities*

A key concept in evolutionary computation is strategy adaptation, i.e., the automatic adjustment of the search strategy during the optimization process.[30,31,32]

Not all operators might be necessary at all stages of evolution. In our case, questions such as when fine-tuning becomes more important than operating on receptive fields cannot be answered in advance. Hence, the application probabilities of the 8 variation operators are adapted using the method from Igel and Kreutz,[31] which is inspired by Davis' work.[33] The underlying assumption is that recent beneficial modifications are likely to be also beneficial in the following generations.

The 8 elemental operators are divided into five groups, those adding connections, deleting connections, adding nodes, deleting nodes, and solely modifying weights. Let Ω be the set of variation operators and let $p_o^{(t)}$ be the probability that $o \in \Omega$ is chosen at generation t. The initial probabilities for operators within a group are the same and add up to 0.2. Let $\mathcal{O}_o^{(t)}$ contain all offspring produced at generation t by an application of the operator o. The case that an offspring is produced by applying more than one operator is treated as if the offspring was generated several times, once by each operator involved. The operator probabilities are updated every τ generations. Here we set $\tau = 4$. This period is called an adaptation cycle. The average performance achieved by the operator o over an adaptation cycle is measured by

$$q_o^{(t,\tau)} := \sum_{i=0}^{\tau-1} \sum_{a \in \mathcal{O}_o^{(t-i)}} \max\left(0, \mathrm{B}^{(t)}(a)\right) \Big/ \sum_{i=0}^{\tau-1} \left| \mathcal{O}_o^{(t-i)} \right| , \tag{5}$$

where $\mathrm{B}^{(t)}(a)$ represents a quality measure proportional to some kind of fitness improvement. This is for the scalar value based selection scheme, case (A),

$$\mathrm{B}^{(t)}(a) := \Phi(a) - \Phi(\mathrm{parent}(a)) \tag{6}$$

and for the vector-valued selection scheme, case (B),

$$\mathrm{B}^{(t)}(a) := \mathrm{R}^{(t)}(\mathrm{parent}(a)) - \mathrm{R}^{(t)}(a) \tag{7}$$

respectively, where $\mathrm{parent}(a)$ denotes the parent of an offspring a. The operator probabilities $p_o^{(t+1)}$ are adjusted every τ generations according to equations

$$\tilde{p}_o^{(t+1)} := \begin{cases} \zeta \cdot q_o^{(t,\tau)}/q_{\mathrm{all}}^{(t,\tau)} + (1-\zeta) \cdot \tilde{p}_o^{(t)} & \text{if } q_{\mathrm{all}}^{(t,\tau)} > 0 \\ \zeta/|\Omega| \qquad\qquad + (1-\zeta) \cdot \tilde{p}_o^{(t)} & \text{otherwise} \end{cases} \tag{8}$$

and

$$p_o^{(t+1)} := p_{\min} + (1 - |\Omega| \cdot p_{\min}) \tilde{p}_o^{(t+1)} \Big/ \sum_{o' \in \Omega} \tilde{p}_{o'}^{(t+1)} . \tag{9}$$

The factor $q_{\mathrm{all}}^{(t,\tau)} := \sum_{o' \in \Omega} q_{o'}^{(t,\tau)}$ is used for normalization and $\tilde{p}_o^{(t+1)}$ stores the weighted average of the quality of the operator o, where the influence of previous adaptation cycles decreases exponentially. The rate of this decay is controlled by $\zeta \in (0, 1]$, which is set to $\zeta = 0.3$ in our experiments. The operator fitness $p_o^{(t+1)}$ is computed from the weighted average $\tilde{p}_o^{(t+1)}$, such that all operator probabilities sum to one and are not lower than the bound $p_{\min} < 1/|\Omega|$. Initially, $\tilde{p}_o^{(0)} = p_o^{(0)}$ for all $o \in \Omega$.

The adaptation algorithm itself has free parameters, $p_{\min}$, τ and ζ. However, in general the number of free parameters is reduced compared to the number of parameters that are adapted and the choice of the new parameters is considerably more robust. Both τ and ξ control the speed of the adaptation; a small ξ can compensate for a small τ ($\tau = 1$ may be a reasonable choice in many applications). The adaptation adds a new quality to the algorithm as the operator probabilities can vary over time. It has been empirically shown that the operator probabilities are adapted according to different phases of the optimization process and that the performance of the structure optimization benefits from this adaptation.[31]

3. Experimental Evaluation

It has already been shown that the size of the face detection network of the Viisage-FaceFINDER® system can be successfully reduced without loss of accuracy by the scalar fitness value approach (A).[7] Here, we want to study whether we can improve these results by using the NSGA-II selection scheme (B). This is done by comparing the performance of our hybrid algorithm using either selection variant (A) or (B). We assume that the runtime of our algorithm is strongly dominated by the number of fitness evaluations (due to the efforts spent for learning) and that the the number of fitness evaluations allowed is fixed. Then there is roughly no difference in runtime between the single- (A) and the multi-objective approach (B).

3.1. *Comparing multi-objective optimization algorithms*

In single-objective optimization the performance can be assessed by looking at scalar objective function values. In EMO performance comparisons are not straightforward because the the outcomes are sets of vectors.[34] Still, it is possible to make statements about the relative quality of different Pareto fronts. Let $\mathfrak{P}(\mathbb{R}^n)$ denotes the power set of $\mathbb{R}^n$. A set $A \in \mathfrak{P}(\mathbb{R}^n)$ can be defined to be better than a front $B \in \mathfrak{P}(\mathbb{R}^n)$ by the relation $A \rhd B$ (A weakly dominates B) given by

$$A \rhd B \text{ iff } A \neq B \text{ and } \forall\, \boldsymbol{b} \in B : \exists\, \boldsymbol{a} \in A : \quad \boldsymbol{b} \text{ is weakly dominated by } \boldsymbol{a} \ . \quad (10)$$

Weak dominance of $\boldsymbol{z}$ compared to $\boldsymbol{z'}$ means that objective vector $\boldsymbol{z}$ is not worse than $\boldsymbol{z'}$ in all objectives. We would also like to make some quantitative statements about how much Pareto fronts outperform each other. Independent of which quantitative indicator I we are going to apply it should always be compliant with the $A \rhd B$ relation, i.e., statements such as "A outperforms B concerning I" should also always imply $A \rhd B$ and vice versa. As shown by Zitzler et al.[35] it is impossible to provide a finite set of unary quantitative indicators $I_i : \mathfrak{P}(\mathbb{R}^n) \to \mathbb{R}$, $1 \leq i \leq m$, such that in general $I_1(A) > I_1(B) \wedge \ldots \wedge I_m(A) > I_m(B) \Leftrightarrow A \rhd B$. However, we can achieve

$$(A \rhd B \Rightarrow I(A) > I(B)) \text{ and } (I(A) > I(B) \Rightarrow B \not\rhd A) \ , \quad (11)$$

for example when using the hypervolume-indicator[29] H_P explained in Fig. 3 (right). It measures the portion of objective space that is weakly dominated by the Pareto front P.

Binary (not necessarily symmetric) indicators $I : \mathfrak{P}(\mathbb{R}^n) \times \mathfrak{P}(\mathbb{R}^n) \to \mathbb{R}$, which assign real numbers to ordered pairs of Pareto fronts, do not suffer from the restriction of unary indicators described above. Here we consider the coverage-difference-indicator $D_{A,B} := H_{A+B} - H_B$.[29] It reflects the size of the objective space that is weakly dominated by the set A but not by B, see Fig. 3 (right). It holds

$$(D_{A,B} > 0 \text{ and } D_{B,A} = 0) \Leftrightarrow (A \rhd B) \quad . \tag{12}$$

The coverage difference $D_{A,B}$ also allows to draw conclusions of the form $(D_{A,B} = 0) \wedge (D_{B,A} = 0) \Leftrightarrow (A = B)$, and $(D_{A,B} > 0) \wedge (D_{B,A} > 0) \Leftrightarrow (A||B)$, where $A||B$ denotes that A and B are incomparable. Following Zitzler's suggestion, we only consider normalized quantities. All hypervolumes are divided by $V := \prod_{i=1}^{n}(z_i^{\max} - z_i^{\min})$, where $z_i^{\max}$ and $z_i^{\min}$ are the maximum and minimum value the i-th objective of a vector z can take. In our experiments we determine $z_1^{\max}, z_2^{\max}$ and $z_1^{\min}, z_2^{\min}$ empirically by looking at all Pareto fronts obtained.

In each trial of our optimization algorithm, we maintain an additional archive

$$\mathcal{A}^{(t+1)} := P_{\bigcup_{t'=0}^{t+1} \mathcal{P}^{(t')}} = P_{\mathcal{A}^{(t)} \cup \mathcal{P}^{(t+1)}} \tag{13}$$

starting from $\mathcal{A}^{(0)} = P_{\mathcal{P}^{(0)}}$. In our optimization scenario $|\mathcal{A}^{(t)}|$ is always small and we do not need to discard any non-dominated solutions. We regard the final archive as the outcome of an optimization trial.

Let $A_1, \ldots, A_T$ and $B_1, \ldots, B_T$ be the final outcomes (i.e., Pareto fronts) of our hybrid evolutionary algorithm using either selection method (A) or (B) starting from T independent random seeds. We calculate the median and the median absolute deviation (mad) of $H_{A_1}, \ldots, H_{A_T}$ and of $H_{B_1}, \ldots, H_{B_T}$. The Wilcoxon-rank-sum test is applied to decide whether the distributions are different. Then we calculate all D_{A_i,B_j} and D_{B_j,A_i} for $1 \leq i, j \leq T$ and compute the median and the median absolute deviation of the quantities $\Delta_{A_i,B_j} := D_{A_i,B_j} - D_{B_j,A_i} = -\Delta_{B_j,A_i}$ and D_{A_i,B_j}. Furthermore we calculate for $1 \leq i, j \leq T$

$$\mathcal{P}_{A_i \rhd B} := |\{(A_i, B_j) : D_{A_i,B_j} > 0 \wedge D_{B_j,A_i} = 0, \ 1 \leq j \leq T\}| \cdot {}^1\!/_T \ , \tag{14}$$

$$\mathcal{P}_{A_i || B} := |\{(A_i, B_j) : D_{A_i,B_j} > 0 \wedge D_{B_j,A_i} > 0, \ 1 \leq j \leq T\}| \cdot {}^1\!/_T \ , \tag{15}$$

$$\mathcal{P}_{B_j \rhd A} := |\{(A_i, B_j) : D_{A_i,B_j} = 0 \wedge D_{B_j,A_i} > 0, \ 1 \leq i \leq T\}| \cdot {}^1\!/_T \ , \tag{16}$$

$$\mathcal{P}_{B_j || A} := |\{(A_i, B_j) : D_{A_i,B_j} > 0 \wedge D_{B_j,A_i} > 0, \ 1 \leq i \leq T\}| \cdot {}^1\!/_T \ , \tag{17}$$

that is, in (14) the average number of trials from algorithm (B) that perform worse than trial A_i, in (15) the average number of trials from algorithm (B) that are incomparable to trial A_i, in (16) the average number of trials from algorithm (A) that perform worse than trial B_j, and finally in (17) the average number of trials from algorithm (A) that are incomparable to trial B_j.

3.2. *Experimental setup*

We want to quantify the benefits of hybrid optimization of NNs for face detection either using selection method (A) or (B) and not the performance of the complete Viisage-FaceFINDER® including preprocessing and face recognition. For comparison we trained the *reference topology* 100 times for 2000 iterations using the improved Rprop learning procedure on D_{train}. From all trials and all iterations we selected the network a_{ref} with the smallest classification error on $D_{\mathrm{val}} \cup D_{\mathrm{test}}$. In the following, all results are normalized by the performance of a_{ref}. For example, the normalized classification error of an evolutionary optimized NN a is given by $\mathrm{CE}'_a(D) := \mathrm{CE}_a(D)/\mathrm{CE}_{a_{\mathrm{ref}}}(D)$ and the normalized number of hidden neurons by $n'_{\mathrm{hidden}}(a) := n_{\mathrm{hidden}}(a)/52$.

   We start $T = 10$ trials of both variants (A) and (B) of the evolutionary algorithm described above for 200 generations (i.e., 5025 fitness evaluations per trial). For each evolved NN we calculate the value $\mathrm{CE}'(D_{\mathrm{test}})$. Although cross-validation is applied when training the NNs, the evolutionary optimization may lead to overfitting, in our case it overfits the patterns of $D_{\mathrm{train}} \cup D_{\mathrm{val}}$. Hence, we additionally introduce the data set D_{test} to finally choose models that generalize well. That is, we use D_{test} for some kind of cross-validation of the evolutionary process. When selecting the *reference topology* a_{ref}, we decide in a similar way as in picking a solution from the evolved architectures, but taking also D_{val} into account. This is reasonable, since D_{val} has not been applied during NN training.

4. Results

For the evolved Pareto NNs, the classification errors and the corresponding numbers of hidden neurons are depicted in Fig. 4 (left) as coordinates in a plane. The table in Fig. 4 (right) characterizes the Pareto fronts (i.e., the final archives) produced by using selection scheme (A) and (B), respectively.

   The median of the portion of the area weakly dominated by the outcomes of variant (B) is significantly larger than the area weakly dominated by Pareto fronts produced by selection scheme (A) (Wilcoxon-rank-sum-test, $p < 0.001$). Further, almost all of the space which is weakly dominated by unions $A_i \cup B_j$ is already weakly dominated by the front B_j for all $1 \leq i, j \leq T$. Evidently, selection method (B) seems to be superior to (A).

   No Pareto front which was generated by selection scheme (A) weakly dominates any of selection variant (B). When we look at the *medians* of our performance indicators shown in Fig. 4 (right) we can state: The outcome of selection method (B) is better than 35% of the outcomes of (A). A Pareto front with respect to (B) is incomparable to 65% of the outcomes of (A), and 55% vice versa. The outcome of (A) is worse than 45% of trials with regard to (B).

   In Tab. 1 we illustrate some details of the most interesting solutions from the "meta Pareto front". The latter denotes the Pareto front of all the NNs of all

PSfrag replacements

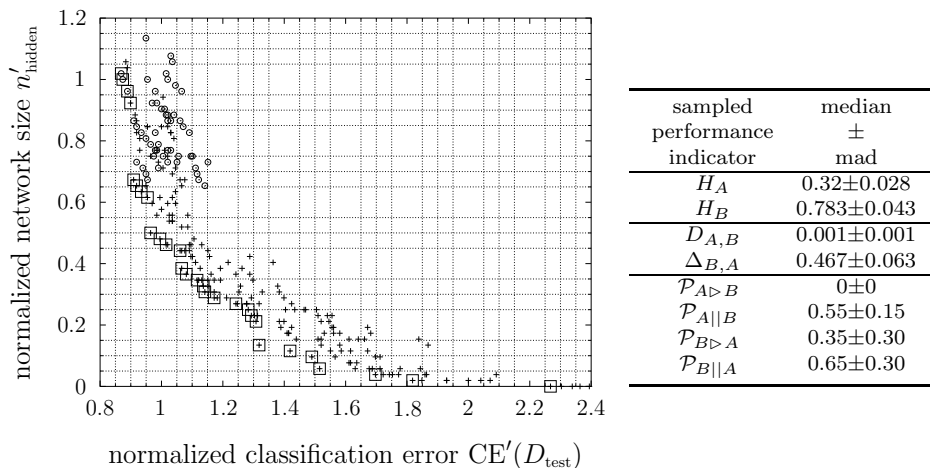| sampled performance indicator	median $\pm$ mad
H_A	0.32 ± 0.028
H_B	0.783 ± 0.043
$D_{A,B}$	0.001 ± 0.001
$\Delta_{B,A}$	0.467 ± 0.063
$\mathcal{P}_{A\triangleright B}$	0 ± 0
$\mathcal{P}_{A||B}$	0.55 ± 0.15
$\mathcal{P}_{B\triangleright A}$	0.35 ± 0.30
$\mathcal{P}_{B||A}$	0.65 ± 0.30

Fig. 4.　Evolved solutions by selection variants (A) and (B). The left plot shows the two objectives, the normalized classification error $CE'(D_{\text{test}})$ and the normalized number of hidden neurons n'_{hidden}, for all NNs of all Pareto fronts of all trials. The circles represent the outcomes of selection method (A), and the crosses the results of the variant (B). The non-dominated NNs, i.e., those constituting the "meta Pareto front", are highlighted. The right table shows the median and the median absolute deviation (mad) with respect to some performance indicators explained in the text. We define the multisets $H_R := \{H_{R_i}|\ i = 1,\dots,T\}, D_{R,S} := \{D_{R_i,S_j}|\ i,j = 1,\dots,T\}$, $\Delta_{R,S} := \{-\Delta_{R_i,S_j}|\ i,j,=1,\dots,T\}, \mathcal{P}_{R||S} := \{\mathcal{P}_{R_j||S}\,|\,j=1,\dots,T\}$, and $\mathcal{P}_{R\triangleright S} := \{\mathcal{P}_{R_j\triangleright S}\,|\,j = 1,\dots,T\}$.

fronts independent of the actual selection variant. There are only three solutions that were generated by selection variant (A) that belong to the "meta Pareto front". The other ones were evolved using selection method (B). Interestingly the outcomes of (A) are extreme solutions in the sense that the classification accuracy criterion is smallest at the cost of being achieved by the largest NN structures. It turns out that such solutions have been found in early stages of evolution. All NNs of the "meta Pareto front" in Tab. 1 which consist of at least 25 hidden nodes are comparable to the *reference topology* with respect to the classification error on D_{test}. This means, the numbers of hidden neurons are reduced by up to 50 % without a loss of classification accuracy. A generalization performance test on a fourth data set D_{extern}, which is independent from all data used for optimization and the final NN

Table 1.　Details of interesting NNs from the "meta Pareto front". The first three approximate Pareto optimal solutions were found with selection method (A), all other with variant (B).

n_{hidden}	53	52	50	48	35	34	33	32	26	25	$\dots$	0
n_{weights}	5860	5756	5447	2891	2355	2336	2310	2387	3428	3496	$\dots$	292
n'_{hidden}	1.02	1.00	0.96	0.92	0.67	0.65	0.63	0.62	0.50	0.48	$\dots$	0
$CE'(D_{\text{test}})$	0.869	0.87	0.89	0.90	0.91	0.92	0.93	0.95	0.96	0.99	$\dots$	2.27
$CE'(D_{\text{extern}})$	0.91	0.90	0.89	1.00	1.00	0.99	0.97	0.97	1.14	1.16	$\dots$	2.54

choice, demonstrates that most of our considerably smaller NNs perform at least as good as the expert-designed architecture.

5. Discussion and conclusions

The proposed hybrid evolutionary algorithms using either scalar fitness or the NSGA-II selection scheme successfully solve the problem of reducing the number of hidden neurons of the Viisage-FaceFINDER® face detection neural network (NN) without loosing detection accuracy. The speed of classification whether an image region corresponds to a face or not has been improved by up to 50 % compared to a reference topology proposed in the literature. By speeding up classification, the rate of complete scans of video-stream images can be increased leading to a more accurate recognition and tracking of persons.

We have revealed that in our face detection scenario structure optimization of NNs is indeed a non-trivial multi-objective problem. The results concerning the performance indicators H_R, $D_{R,S}$, $\mathcal{P}_{R||S}$, and $\mathcal{P}_{R \rhd S}$ are not very surprising, since we have compared a single- against a multi-objective selection scheme on the basis of multi-objective performance indicators. Pareto fronts obtained by evolutionary multi-objective optimization can be considered as approximations of the sets of optimal trade-offs between several objectives. By using the linear aggregation scheme for selection and choosing the weighting factors for the objectives already before search we take a major decision about what kind of trade-offs are accessible. That is, we probably restrict the optimization algorithm to sample only within a relatively close area in objective space—solutions that do not match our prior decision cannot be found. The application of a multi-objective selection scheme such as the one from NSGA-II facilitates a decision about the best compromise between multiple objectives after the optimization process. The multi-objective selection scheme requires less expert knowledge (it can be used more "out-of-the-box"), because no crucial weighting factors have to be chosen in advance.

The results of the multi-objective performance indicators also give some evidence that the NSGA-II selection leads to preferable solutions in a more robust way compared to the linear aggregation selection scheme. It should be noted that there are several evolved NNs (see Tab. 1) that have more weights than the initial one, but fewer hidden nodes. Such solutions cannot be found by a pure pruning algorithm. The convincing results encourage to adopt our hybrid algorithm with vector-valued selection for the automatic construction of NNs for other classification tasks.

When evolving NNs one is usually interested in well generalizing solutions. However, even if the classification error on a fixed additional data set that is not considered for adapting the weights (neither for training nor for early stopping) is—additionally or solely—used in the fitness calculation, the evolved NNs tend to overfit to the data responsible for their selection. Therefore we introduced additional data sets to reliably assess generalization performance. Our way of performing cross-validation of both, learning and evolution, is an improvement compared to

other methods. Nonetheless, the problem of evolving good generalizing NNs requires further investigation.

References

1. Viisage Technology AG, `http://www.viisage.com`.
2. M. Hüsken, M. Brauckmann, S. Gehlen, K. Okada and C. von der Malsburg, Evaluation of implicit 3D modeling for pose invariant face recognition, in *Defense and Security Symposium 2004: Biometric Technology for Human Identification*, eds. A. K. Jain and N. K. Ratha *Proceedings of SPIE* **5404**, (The International Society for Optical Engineering, 2004).
3. M. Lades, J. C. Vorbrüggen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Würtz and W. Konen, Distortion invariant object recognition in the dynamic link architecture, *IEEE Transactions on Computers* **42** (1993) 301–311.
4. W. Zhao, R. Chellappa, P. Phillips and A. Rosenfeld, Face recognition: A literature survey, *ACM Computing Surveys (CSUR)* **35**(4) (2003) 399 – 458.
5. U. Handmann, T. Kalinke, C. Tzomakas, M. Werner and W. von Seelen, An image processing system for driver assistance, *Image and Vision Computing* **18**(5) (2000) 367–376.
6. H. M. Hunke, Locating and tracking of human faces with neural networks, Master's thesis, University of Karlsruhe (1994).
7. S. Wiegand, C. Igel and U. Handmann, Evolutionary optimization of neural networks for face detection, in *12th European Symposium on Artificial Neural Networks (ESANN 2004)*, ed. M. Verleysen (Evere, Belgium: d-side publications, 2004), pp. 139–144.
8. C. Coello Coello, D. Van Veldhuizen and G. Lamont, *Evolutionary Algorithms for Solving Multi-objective Problems* (Kluwer Academic Publishers, New York, 2002).
9. K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms* (John Wiley & Sons, Chichester, UK, 2001).
10. H. A. Abbass, Speeding up backpropagation using multiobjective evolutionary algorithms, *Neural Computation* **15**(11) (2003) 2705–2726.
11. Y. Jin, T. Okabe and B. Sendhoff, Neural network regularization and ensembling using multi-objective evolutionary algorithms, in *Proceedings of the Congress on Evolutionary Computation (CEC'04)*, (IEEE Press, 2004), pp. 1–8.
12. K. Deb, S. Agrawal, A. Pratap and T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation* **6**(2) (2002) 182–197.
13. G. P. Zhang, Neural Networks for Classification: A Survey, *IEEE Transactions on System, Man, and Cybernetics – Part C* **30**(4) (2000).
14. M.-H. Yang, D. J. Kriegman and N. Ahuja, Detecting faces in images: A survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(1) (2002) 34–58.
15. E. Hjelmas and B. K. Low, Face detection: A survey, *Computer Vision and Image Understanding* **83** (2001) 236–274.
16. P. L. Bartlett, The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network, *IEEE Transactions on Information Theory* **44**(2) (1998) 525–536.
17. R. Caruana, S. Lawrence and C. L. Giles, Overfitting in neural networks: Backpropagation, Conjugate Gradient, and Early Stopping, in *Advances in Neural Information Processing Systems*, **13**, (MIT Press, Denver, Colorado, 2001), pp. 402–408.

18. S. Nolfi, Evolution and learning in neural networks, in *The Handbook of Brain Theory and Neural Networks*, ed. M. A. Arbib (MIT Press, 2002) pp. 415–418, 2 edn.
19. X. Yao, Evolving artificial neural networks, *Proceedings of the IEEE* **87**(9) (1999) 1423–1447.
20. R. D. Reed and R. J. Marks II, *Neural Smithing* (MIT Press, 1999).
21. A. Stahlberger and M. Riedmiller, Fast network pruning and feature extraction by using the unit-OBS algorithm, in *Advances in Neural Information Processing Systems*, eds. M. C. Mozer, M. I. Jordan and T. Petsche **9**, (The MIT Press, 1997), pp. 655–661.
22. H. A. Rowley, S. Baluja and T. Kanade, Neural network-based face detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(1) (1998) 23–38.
23. H. Braun, *Neurale Netze: Optimierung durch Lernen und Evolution* (Springer-Verlag, 1997).
24. L. D. Whitley, The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best, in *Proceedings of the Third International Conference on Genetic Algorithms, ICGA'89*, ed. J. D. Schaffer (Morgan Kaufmann, Fairfax, VA, USA, 1989), pp. 116–121.
25. C. Igel and M. Hüsken, Empirical evaluation of the improved Rprop learning algorithm, *Neurocomputing* **50**(C) (2003) 105–123.
26. M. Riedmiller, Advanced supervised learning in multi-layer perceptrons – From back-propagation to adaptive learning algorithms, *Computer Standards and Interfaces* **16**(5) (1994) 265–278.
27. L. Prechelt, Early stopping – but when?, in *Neural Networks: Tricks of the Trade*, eds. G. B. Orr and K.-R. Müller, *LNCS* **1524** (Springer-Verlag, 1999) pp. 57–69.
28. D. B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence* (IEEE Press, Piscataway, NJ, USA, 1995).
29. E. Zitzler, *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications* (Shaker Verlag, Aachen, Germany, 1999).
30. A. E. Eiben, R. Hinterding and Z. Michalewicz, Parameter control in evolutionary algorithms, *IEEE Transactions on Evolutionary Computation* **3**(2) (1999) 124–141.
31. C. Igel and M. Kreutz, Operator adaptation in evolutionary computation and its application to structure optimization of neural networks, *Neurocomputing* **55**(1–2) (2003) 347–361.
32. J. E. Smith and T. C. Fogarty, Operator and parameter adaptation in genetic algorithms, *Soft Computing* **1**(2) (1997) 81–87.
33. L. Davis, Adapting operator probabilities in genetic algorithms, in *Proceedings of the Third International Conference on Genetic Algorithms, ICGA'89*, ed. J. D. Schaffer (Morgan Kaufmann, Fairfax, VA, USA, 1989), pp. 61–69.
34. T. Okabe, Y. Jin and B. Sendhoff, A critical survey of performance indices for multi-objective optimisation, in *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'03)*, (IEEE Press, 2003), pp. 1053–1060.
35. E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca and V. G. da Fonseca, Performance assessment of multiobjective optimizers: An analysis and review, *IEEE Transactions on Evolutionary Computation* **7**(2) (2003) 117–132.