Heuristic Speciation for Evolving Neural Network Ensemble

Shin Ando
Yokohama National University
79-1 Tokiwadai Hodogaya-ku
Yokohama, Kanagawa, Japan
ando@ubicg.ynu.ac.jp

ABSTRACT

Speciation is an important concept in evolutionary computation. It refers to an enhancements of evolutionary algorithms to generate a set of diverse solutions. The concept is studied intensively in the evolutionary design of neural network ensembles. The diversity and cooperation of individual networks are among the essential criteria of the design. This paper proposes a speciation framework for ensemble design which integrates a collection of new techniques. Its characteristic features are: (a) the population of networks are speciated as such that the mutual information between the networks' outputs and genotypic representations is preserved. (b) The ensemble is designed incrementally, upon discovery of a *species* of networks which enhances the ensemble performance. (c) Multiple species are evolved and individual networks are evaluated according to the role of their respective species in the ensemble. This framework provides an implementation of evolutionary algorithm which performs simultaneous single-objective optimizations. The new algorithm is evaluated with a series of classification benchmarks and shows an improvement over other evolutionary training strategies and a statistical algorithm.

Categories and Subject Descriptors

I.5.2 [**Computing Methodologies**]: PATTERN RECOGNITION—*Design Methodology*; I.2.6 [**Computing Methodologies**]: ARTIFICIAL INTELLIGENCE—*Learning*

General Terms

Algorithms

Keywords

Niching, Evolutionary Network Design, Pattern Recognition

1. INTRODUCTION

Niching is one of the important concepts in evolutionary computation. It derives from a ecological term *niche*, which refers to a role in an ecosystem filled by a species or a population. In evolutionary computation, the terms *niching* and *speciation* refer to enhancements of evolutionary algorithms (EAs) to find multiple local optima or a set of good solutions, by emphasizing *species* of individuals to respectively evolve in a different *niche*. The concept has been studied extensively for multimodal and deceptive optimization, hybridization of local search, and cooperative coevolution [9, 24, 17, 13, 11]. There are also practical advantages for generating diverse solutions in engineering and design, such as fault tolerance and modularization.

This paper provides new perspectives on two essential aspects of speciation: the similarity measure of individuals and the restriction of mating and competition. Many implementations of niching employ a native distance measure of the feature domain, e.g., Hamming and Euclidean distance for binary and numerical optimizations problem. However, for complex optimization problems, naïve measurements may not sufficiently reflect the relevance of the genotype and the phenotype thus is unsuitable for defining the species. It is especially difficult to define an appropriate similarity when the genotypic representation implicitly includes a structure, such as a network or a graph.

Underlying the concept of speciation is an assumption that within the domain occupied by the species, the performance of the phenotype is quite predictable from its genotype. In such a perspective, a speciation should preserve the information in the genotypic representation about the phenotype as much as possible. This view lead us to a relatively new methodology in machine learning called Information Bottleneck (IB) [26]. IB clustering partitions the observed variable X such that its mutual information, thus the predictive power, for another variable Z is preserved maximally. Such clustering can be useful for speciating the structural genotypes, such as that of neural networks, without having to select an arbitrary similarity measure.

In the natural process of speciation, the new and original species become mutually *incompatible*, i.e., incapable of mating, at some point as the genetic crossover becomes too lethal. This provides an interesting analogy for multimodal optimization, since a crossover of individuals near different local optima is less likely to improve the solution. A major difficulty for implementing such a mechanism is formulating the transition, discrete or continuous, from compatibility to incompatibility. Due to the complexity of the EAs and the variety of the problems to which they are applied, the dynamics of the speciation is difficult to analyze.

In this paper, species are viewed as probabilistic distri-

GECCO'07, July 7–11, 2007, London, England, United Kingdom.

butions. Based on this view, we propose a speciation technique using Minority Detection [2], a method for identifying a small subset whose distribution has a significant divergence from that of the rest of the dataset. Using this method, a species is made incompatible after it differentiates from the global population in terms of the Kullback-Leibler divergence. As species are found heuristically, a radius or a number of niches are unnecessary. The distributions of species allows for the estimation of its niche, which can be useful for avoiding redundancy among species.

This paper addresses the problem of designing a neural network ensemble, in which the concept of niching and speciation has been studied extensively [11, 16, 19]. A typical ensemble is a linear combination of multiple networks. A linear ensemble of k networks' outputs $\mathbf{z} = \sum_{i=1}^{k} \alpha_i \mathbf{z}_i$, where $\mathbf{z}_i$ is the output of an individual network and α_i its weight. Generally, an ensemble approximates a complex function as a combination of decomposed modules or redundant networks. In classification tasks, ensembles exhibit advantages in generalization error and variance over large, single networks with less training time and risk of overfitting.

The task of designing an ensemble is significantly more complicated than that of a single network, as it involves decisions on the number of the ensemble members and their weights as well as designing and training individual networks. The performances of the individual networks as well as their diversity and cooperation are among the properties seen in good ensembles.

Analogically, an ensemble corresponds to an ecosystem whose members fill a respective role in performing classification. The goal of the speciation is to evolve species of networks which sufficiently fill the relevant niches, i.e., the modules or the components of a classifier function.

This paper proposes a simple yet effective speciation framework, which integrates a collection of new techniques. Its significant differences from previous works are: (a) the population of networks is speciated using IB clustering, as such that the predictive power of the genotypic representation is preserved as much as possible. (b) The ensemble is incrementally designed on a heuristic basis, i.e., the ensemble adds a new member if and when a species of networks which enhances the ensemble performance is found. (c) Multiple species are evolved and each individual is evaluated according to the role of the respective species in an ensemble.

These characteristics collectively emphasize the diversity and cooperation among the ensemble without defining an explicit diversity objective. Therefore, the implementation of this framework is a set of single-objective evolutionary optimizations which are run simultaneously.

The rest of the paper is organized as follows. Section 2 discusses the existing works of niching and evolutionary ensemble design. Section 3 describes the enhancements in two aspects of speciation using machine learning techniques. Section 4 describes the proposed framework which integrates the speciation technique with EA. Section 5 evaluates the performance of the new algorithm using a set of classification benchmarks. Section 6 gives the conclusion of this paper.

2. BACKGROUND

2.1 Niching

While the survival of the fittest appears as a principle quite contrary to the diversity of the species, other facets of evolutionary process, e.g., geographical localization and genetic incompatibility, bring about a rich variety of species in nature. In evolutionary computation, many mechanisms to induce diversity and speciation have been studied as techniques of niching and speciation. Such techniques are used in problems of multimodal and deceptive optimization [24, 1], hybridization of local and global search [21, 13] and automatic modulation and coevolution [23, 6]. Two pioneering works among such studies are crowding and fitness sharing, from which derives numerous niching techniques [7, 20, 22, 12, 5, 24]. There are also various studies on incorporating clustering algorithms as a mean of speciation [28, 25, 1].

Fitness sharing induces niching by a distance-based penalty function. The fitness of an individual is degraded based on the level of overpopulation within a certain radius. Penalizing and suppressing of a regions of the fitness landscape is also employed in Clearing [24] and Sequential Niching [3].

Restricting the mating and competition of individuals is another important aspect of speciation. Crowding and its variants control the competition based on the distances between parents and offspring. Species Conserving GA [17] also restricts the survival competition to within a certain radius of superior individuals. Another group of niching techniques evolves multiple subpopulations and restricts mating and survival competition mostly or strictly to within each subpopulation [27, 1, 15]. The restricted mating encourages the discovery of diverse solutions as species are induced to converge to different niches. Another important effect of strictly restricted mating and competition is the faster convergence of subpopulation [15, 1], as crossovers among similar individuals are synonymous to fine tuning of the parameters and are less likely to be lethal.

2.2 Evolutionary Design of Ensembles

Designing an ensemble of neural networks is a significantly more difficult and delicate task than that of a single network, as it involves designing and training multiple neural networks, decisions on the size of the ensemble, selecting its members and their weights. Individual performances and diversity and cooperation among networks are among the criteria that is emphasized in the ensemble design.

Existing approaches to training multiple networks, aside from individual training, are incremental training and simultaneous training [11]. The basic concept of constructive training is to sequentially train a new network, with an additional objective to cooperate with the existing ensemble. Such strategy was initially studied in [16]. Simultaneous training has been employed by various studies [18, 19, 11]. In principle, a set of networks are trained simultaneously, with additional objectives of diversity and cooperation with one another. The two approaches have shown improvements over individual training in inducing diversity and cooperative behavior among individuals. However, a defect of the incremental training is the strong dependence to few initial networks. Simultaneous training demerits from the prerequisite ensemble size, which demands empirical knowledge of the task and the risk of redundancy among networks.

There have been several approaches for emphasizing multiple criteria of ensemble design. Negative correlation learning [19] penalizes the correlation of the networks' outputs in the fitness function. The drawback to the penalization approach is its arbitrary tradeoff parameter between the

penalty and the primary objective. The parameter is known to have a significant effect on the result.

[11] employs a multi-objective EA to allow for a multitude of criteria in ensemble design. While multi-objective EA is very useful for generating Pareto-fronts, it generally does not optimize individual objectives as well as a single-objective EA. As reported in [11], the multi-objective approach is not necessary superior to the single-objective approach, and introducing too many subsidiary objectives degrades the training and generalization error.

2.3 Notations

The basic notations of neural network ensembles and evolutionary algorithms are given as follows. The weights of connections are denoted by a vector $\mathbf{x} = [x_i]_p$. The input pattern and the network output is denoted by vectors $\mathbf{v} = [v_i]_q$ and $\mathbf{z} = [z_i]_r$. q and r correspond to the number of features and classes of the classification task. The number of nodes in the hidden layer is m; thus $p = m(q + r)$. Weights from x_1 to x_q correspond to connections between the first input and the hidden nodes. $x_{q+1} \sim x_{mq}$ correspond to connections between the rest of the input and the hidden nodes in similar a manner. $x_{mq+1} \sim x_{mq+r}$ correspond to connections between the hidden nodes and the first output. Rest of the weights correspond to connections between the hidden nodes and the rest of the outputs in a similar manner.

A network which corresponds to $\mathbf{x}$ is denoted by $N(\mathbf{x})$. An ensemble of k networks is denoted by $\mathbf{N} = \{N_i\}_{i=1}^k$. The output of an ensemble $\mathbf{N}$ is defined as the weighed sum of individual outputs

$$\mathbf{z}(\mathbf{N}, \mathbf{x}) = \sum_{i=1}^{k} \alpha_i \mathbf{z}(N_i, \mathbf{x}),$$

where α_i is the weight of an individual network.

The ensemble classifies a sample by *winner-take-all* method. For convenience, we denote a binary vector $\mathbf{z}_b$ which corresponds to the output. 1 is assigned to the largest z and and 0 to others. A binary vector corresponding to the correct class of ith sample in dataset $V = \{\mathbf{v}_i\}_{i=1}^t$ is denoted by $\mathbf{e}_i$. The ensemble error f is defined from the number of correctly classified sample as

$$f(\mathbf{N}) = \sum_{i=1}^{t} \gamma_i \mathbf{e}_i^T \mathbf{z}_b(\mathbf{N}, \mathbf{v}_i) \tag{1}$$

where γ_i is a weight defined as

$$\gamma_i = \frac{1}{k} \sum_{j=1}^{k} \mathbf{e}_i^T \mathbf{z}_b(N_j, \mathbf{v}_i) \tag{2}$$

(2) represents the emphasis on samples that are difficult to classify by current ensemble members. This weighting has been proposed in [11] to cover a similar concept in boosting.

In the evolutionary algorithm, a global population is denoted by G. A species is a population of individuals denoted by $s = \{\mathbf{x}_i\}$. A set of existing species is denoted by $S = \{s_i\}$. The assignment of an individual $\mathbf{x}_i$ to a species s is denoted by $y_i = s$. The cardinality of a set or a population is denoted by $\#(\cdot)$. The functions of evolutionary algorithms, i.e., initialization, reproductive selection, survival selection, crossover operation, are denoted by $initialize()$, $RS(\cdot)$, $SS(\cdot)$, and $XO(\cdot)$ respectively. $initialize()$ generates an individual from a uniform distribution over the search domain. $RS(X)$ returns a random subset of $M \subset X$ as parents. $XO(M)$ is a simplex crossover (SPX) [14], which generates offspring from a uniform distribution over an expanded simplex formed by μ parents. The number of parents is set to $\mu = p + 1$ as suggested in [14]. $SS(X, L)$ compares the fitness of an individual randomly chosen from a population X and the best offspring in L. The former is replaced by the latter if it has a worse fitness. This survival selection is a variant of the Generalized Generation Gap Model [8]. The number of offspring is empirically set to $\lambda = p^2$.

3. SPECIATION TECHNIQUES

This section discusses new views on two critical aspects of speciation: similarity measure and incompatibility.

3.1 Speciation by Information Bottleneck

In clustering, selecting a similarity measure can be viewed as implicitly determining the relevance of the respective variables. As such, the impact of the similarity measure is very significant, often more so than the clustering method itself. The impact is just as significant with niching techniques, all of which define some measure of similarity between the individuals. A good measure differs from problem to problem, depending on the representation and the fitness function. For example, if the fitness function is significantly sensitive to some variables, or variables are highly correlated, use of Mahalanobis distance is more appropriate over Euclidean distance. The similarity of two numerical vectors depends on what they represent, e.g, points in Euclidean domain or weights of neural networks. For effective niching, it is essential that the similarity of genotype translates to the similarity of the phenotype or performance, although such a measure is not always apparent when the genotype includes an implicit structure, e.g., graphs and networks.

In speciating the population, it is implicitly assumed that within the domain occupied by the species, the performance of the phenotype strongly relate to the genotypic representation. As such, it seems essential that speciated population preserve the information about the phenotype as much as possible. Such view suggests the introduction of Information Bottleneck [26] methodology, which incorporates top-down domain knowledge into unsupervised learning.

IB formalizes a probabilistic clustering, using standard notations of mixture estimation: the observed variable X and the hidden variable Y which corresponds to respective clusters. The conditional probabilities $P(Y|X)$ represent the soft partitioning of the instances. Additionally, a variable which represents the domain knowledge, here denoted as Z, is introduced.

The clustering problem is formalized as a minimization w.r.t. $P(Y|X)$ as follows.

$$\min_{P(Y|X)} I(Y;X) - \beta I(Y;Z) \tag{3}$$

This formalization derives from the context of data compression, which shares basic concepts with unsupervised learning. Intuitively, (3) describes the compression of X to signal Y such that the information of Y is minimized, while preserving the information it has about Z.

Given the definition of mutual information

$$I(Y;X) = \sum_{\mathbf{x} \in X} \sum_{y \in Y} P(\mathbf{x}, y) \log \frac{P(\mathbf{x}, y)}{P(\mathbf{x}) P(y)}, \tag{4}$$

(3) rewrites as a functional of conditional probability $P(y|\mathbf{x})$.

$$\Lambda = \sum_{\mathbf{x},y \in X,Y} P(y|\mathbf{x}) P(\mathbf{x}) \log \frac{P(y|\mathbf{x})}{P(y)} - \beta \langle d(\mathbf{x},y) \rangle_{P(\mathbf{x},y)}$$

where d denotes the Kullback-Leibler divergence D_{KL} between two conditional probabilities

$$d(\mathbf{x},y) = D_{KL}(p(\mathbf{z}|\mathbf{x}) | p(\mathbf{z}|y)) \quad (5)$$

The minimum of Λ is obtained by an iterative algorithm described in [26]. $P(Y|X)$ corresponding to such minimum represents a partitioning of X which preserves the mutual information between X and Z. In a machine learning task such as speech recognition, X corresponds to the acoustic data, and Z represents one of the relevant information, e.g., speaker's identity or the transcripted text. Z thus is a perspective, to which the clustering result should be relevant. In IB, the Kullback-Leibler divergence between $P(\mathbf{z}|\mathbf{x})$ and $P(\mathbf{z}|y)$ serves as the implicit similarity, which allows for clustering without specifying the measure for respective Z.

With regards to evolutionary algorithms, X corresponds to the genotype of the individuals and Z to their phenotype. Y indicate the assignment of individuals to respective species. This technique provides a clustering with an unarbitrary similarity measure when the relevance between the genotype and the phenotype is complex. It is important to note that clustering described above is different from clustering individuals based on the neural networks' output.

3.2 Detection of Species as Minority Subsets

Given the nature of the optimization task, it is very unlikely that the properties of the landscape, e.g., the radii or the number of local optima, be provided. Further, it is cautious to assume that the local optima and their surrounding landscapes are not uniform, therefore species corresponding to some niche may emerge slower than others, or not appear at all in deceptive problems. Most speciation techniques therefore subjectively analyze individuals to estimate the property, such as range or density, of *species*. The dynamics of the speciation, however, is generally difficult to formalize due to the complexity of EAs and the variety of the problems addressed. In the following, an empirical view of the species as a probability distribution is introduced from an analysis of a simple evolutionary optimization setting.

The fitness function is $g : x \in \Omega \subset \mathbb{R} \to \mathbb{R}$, which has a set of local optima $\{c_i\}$. With regards to each c_i and its niche Φ_i, g is approximated $g_i(x) \approx \alpha (x - c_i)^2$ for $x \in \{x | x - c_i < d\} \equiv \Phi_i$ and $g_n(x) \approx b$ otherwise. $g_n(x)$ is always smaller than $g_i(x)$ for all i.

The initial population is distributed uniformly, $P(X_0) = U(\Omega)$. The crossover, $XO(x_a, x_b) = U([x_a, x_b])$, is uniform as well. We assume that the subdomain of the niches are relatively small, i.e., $\int_{x \in \Phi_i} dx \ll \int_{x \in \Omega} dx$, formally. RS is a random selection and SS is an elitist strategy which replaces the worst parents with the best offspring.

When κ offspring are generated within the niche Φ_i, the probability that one at x_a is selected by SS is

$$Q(x|\Phi_i) = \frac{1}{\int_{\Phi_i} dx} \left(1 - \frac{\int_{g(x_a) < g(x)} dx}{\int_{\Phi_i} dx}\right)^{\kappa - 1}$$

Assuming that the offspring generated outside of Φ_i are irrelevant, the probability distribution over Φ_i is defined as

$$P(x|\Phi_i) = \frac{1}{\zeta} Q(x|\Phi_i) \quad (6)$$

with ζ being the normalization constant. $P(x|\Phi_i)$ has an exponential derivative and a limited variance depending on the value of κ.

The simple analysis of EA provides an intuitive definition of a species, which is a small subset of a individuals with a limited variance and a significantly different distribution than the rest of the population. Subsequently, speciation is defined as a detection of such subset as a species.

We propose an extension of a machine learning technique called Minority Detection (MD) [2] for this detection problem. MD identifies a subset whose distribution has a significant divergence from that the majority of the dataset. For more details, readers are referred to [2].

MD is an extension of probabilistic mixture estimation; the same notations from Section 3.1 is used. A minority detection problem is characterized by following two settings.

1. An instance $\mathbf{x}_i$ belongs to either a majority $y_i = a$ or a minority $y_i = b$.
2. The cardinality of the subset $S = \{\mathbf{x}_i; y_i = b\}$ is trivial to that of the entire dataset.

Accordingly to 1. and 2., an approximation $P(\mathbf{x}|a) \approx P(\mathbf{x})$ is introduced. This approximation trivializes the first term of the following expansion of mutual information,

$$\begin{aligned} I(Y;X) &= P(a) \sum_{\mathbf{x} \in X} P(\mathbf{x}|a) \log \frac{P(\mathbf{x}|a)}{P(\mathbf{x})} \\ &\quad + P(b) \sum_{\mathbf{x} \in X} P(\mathbf{x}|b) \log \frac{P(\mathbf{x}|b)}{P(\mathbf{x})} \end{aligned}$$

therefore,

$$\approx P(b) \sum_{\mathbf{x} \in X} P(\mathbf{x}|b) \log \frac{P(\mathbf{x}|b)}{P(\mathbf{x})} \equiv I(b;X)$$

Introducing a similar approximation for Z, i.e., $P(\mathbf{z}|a) \approx P(\mathbf{z})$, the approximation of $I(Y;Z)$ is denoted by $I(b;Z)$. Using $I(b;X)$ and $I(b;Z)$, the clustering objective Λ is rewritten as follows.

$$\begin{aligned} \Lambda &= I(b;X) - \beta I(b;Z) \\ &= P(b) \sum_{\mathbf{x} \in X} P(\mathbf{x}|b) \log \frac{P(\mathbf{x}|b)}{P(\mathbf{x})} - \beta \langle d(\mathbf{x},b) \rangle_{P(\mathbf{x},b)} \end{aligned}$$

where d follows the definition of (5).

[2] shows an iterative algorithm for minimizing Λ. The time/space complexity of each step is $O(\#(S))$. The pseudo code of the algorithm is shown in Fig. 1.

An important characteristic of the MD is that if a minority subset is not present or its density is smaller than the rest of the dataset, the estimated variance of minority reduces to 0, which corresponds to an empty minority set. As such, the speciation based on MD is deterred from speciating a small fluctuation of the initial population or a premature species which may not yet reflect the property of a niche. Presumption of the radii or the number of species is not required, since species' parameters are estimated from individuals.

We define a function *speciate*(X), whose outputs are a subset $s_i = \{x_j; y_j = s_i\}$, and its distribution $P(\mathbf{x}|s_i)$ estimated by MD.

INPUT: observed values X , initial labels Y
OUTPUT: subset s, distribution parameter θ
METHOD:
define $s \equiv \{\mathbf{x}_j \in X; y_j = b\}$, $P(\mathbf{x}|\theta) \equiv P(\mathbf{x}|b)$
initialize θ_0 s.t. $P(\mathbf{x}|\theta_0) = P(\mathbf{x}|s)$, $\Lambda_0 = \Lambda(P(\mathbf{x}|\theta_0))$
$t = 0$
repeat
  $Y \rightarrow Y_t$,$\theta \rightarrow \theta_t$, $t+1 \rightarrow t$
  $a \rightarrow \arg\min_{y_i \in Y} P(\mathbf{x}_i|b)$
  update θ s.t. $P(\mathbf{x}|\theta) = P(\mathbf{x}|s)$
  $\Lambda_t \leftarrow \Lambda(P(\mathbf{x}|\theta))$
until $\Lambda_t > \Lambda_{t-1}$
return W, θ

Figure 1: Pseudo-code of Minority Detection

4. HEURISTIC SPECIATION

4.1 Incremental Framework

This section describes the Heuristic Speciation (HS), a framework which alternates between the evolutionary algorithm and a speciation procedure. HS evolves a global population G and an set of species S. The population of each species derives from the global population, but the mating and competition is restricted to its members once it has speciated. The members of the ensemble $\mathbf{N}$ respectively correspond to the best individual of a species.

HS iterates following steps for the global population illustrated in Fig. 2 (a).

1. Initialize each individual in G with *initialize* (). Also initialize S and $\mathbf{N}$ as empty sets.
2. Apply genetic operators RS, XO, SS to G until κ new offspring are added to G.
3. Execute *speciate* (G).
4. If new species are found, update S and G; otherwise skip to 5.
5. Terminate upon reaching the target performance or number of evaluations; otherwise repeat from 2.

The update of S and G in Step 4. is processed as follows. If a new species s_i is found, the individuals $\{x_j\} \in s_i$ are removed from the global population G and replaced by the same number of individuals generated by *initialize* (). s_i is added to S and its best individual is added to $\mathbf{N}$.

Each new offspring $x \in X$ is considered for reassignment to all existing in S. If there exists a species $s \in S$ s.t. $F(s) < F(s+x)$, x is added to the species s. The pseudo-code of the update procedure is shown in Fig. 3.

Each speciated population is evolved as an independent population using a standard evolutionary algorithm. Subsequent to its speciation, HS iterates with following steps with each species s_i, as illustrated in Fig. 2 (b).

1. Execute selection $RS(s_i)$ to select parents M.
2. Execute crossover $XO(M)$ to generate offspring L.
3. Execute survival selection $SS(L, s_i)$.
4. Update the best individual in s_i to the ensemble.

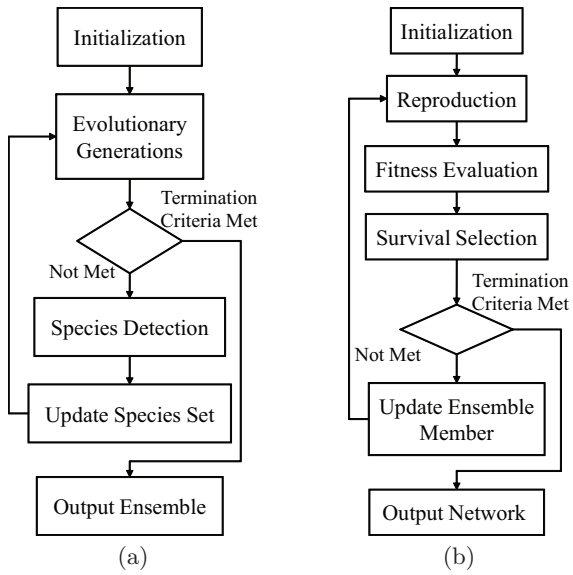

Figure 2: Flowchart of evolution for global population (a) and species (b)

5. Halt on the increase in validation error or termination of global evolution; otherwise repeat from 1.

As described above, speciation restricts direct flow of information from a species to the global population. However, new individuals from global population are continually added to the species via the update of S. To maintain a finite species population, the worst individuals are eliminated after every κ generations.

The population size of the global and species population are determined accordingly to their respective goal of evolution. A small population size is used for species to converge quickly to a local optima. The diversity of the species population is sustained by new individuals supplied from the global population. Much larger population size is used for the global population to cover multiple niches in a multimodal landscape.

As discussed in Section 3.2, the algorithm for detecting species requires $O(\#(s)\, p^2)$ per iteration step considering the computation of vector statistics in p dimension, where

INPUT: global population G, existing species $S = \{s_i\}_{i=1}^{k}$, new species $S' = \{s_i\}_{i=k+1}^{k+m}$, new offspring $X = \{\mathbf{x}_i\}_{i=1}^{t}$
OUTPUT: G, S
METHOD:
for all $s \in S'$ **do**
  $G = G - s$, $S = S + s$.
  $G = G +$ *initialize* $(\#(s))$
end for
for all $\mathbf{x}_i \in X \cap G$ and $s_j \in S$ **do**
  if $F(s_j) < F(s_j + x_i)$ **then**
    $G = G - \mathbf{x}_i$, $s_j = s_j + \mathbf{x}_i$
  end if
end for

Figure 3: Pseudo-code for Updating G and S

$\#(s)$ is the tentative size of the species. The number of steps does not exceed $\#(s_0)$, the initial size of the species, as only the niches near the new individuals are relevant. The iteration is repeated using each of the κ new offspring as a *seed* for different initialization. The computational complexity of the speciation is therefore $O\left(\kappa\#(s_0)^2 p^2\right)$. Meanwhile, the complexity of the genetic operator is $O\left(p^2\right)$ as discussed in [14]. Overall complexity of the EA with HS is $O\left(\#(s_0)^2 p^2\right)$ per each generation of an offspring.

4.2 Niche-wise Evaluation

Since each species represents a different member of the ensemble, it is natural for its individuals to be evaluated according to the species' role in the ensemble. As such, we propose a Niche-wise Evaluation described as follows. Given the ensemble error (1) for training samples V and the current ensemble $\mathbf{N}$, the fitness function of an individual $\mathbf{x}$ in the global population is defined

$$f_0 = f\left(\mathbf{N} + N\left(\mathbf{x}\right)\right). \tag{7}$$

The fitness of an individual $\mathbf{x}$ in a species s_i, given the best network of the species N_i, is defined

$$f_i = f\left(\mathbf{N} - N_i + N\left(\mathbf{x}\right)\right). \tag{8}$$

Despite the simple formulation, (7) and (8) formulate some of the important criteria of ensemble design. The individuals are not rewarded for classifying samples that are classifiable by a member of the ensemble, as all individual has the access to the members of the ensemble.

[11] has introduced the *Substitution* objective similar to (8). It was shown to be an effective objective for single and multi-objective EA. In this paper, (8) is used as a primary emphasis to establish the species' respective role, in conjunction with Heuristic Speciation framework and the evaluation principal of the global population.

Under niche-wise evaluation, it is unlikely to find two species which correspond to the similar component of a function as finding one such species modifies the fitness function and negates the advantage of similar species. Meanwhile, species with a small population described in Section 4.1 often converge quickly to a single local optimum. As such, the reconfiguration of the species, such as merging and splitting were practically unnecessary thus not implemented in HS framework.

4.3 Weighting the Ensemble

As seen in (7) and (8), each evaluation is based on a different ensemble, thus the weights of individual networks α, is not fixed. In this framework, two methods are used for computing weights: Stacked Regression (SR) [4] and gradient decent method (GD). SR is a weight optimization measure based on cross-validation and computationally expensive than GD. HS updates α at following points.

1. Upon discovery of a new species; calculated with SR.
2. At each evaluation; re-adjusted with GD.

5. EXPERIMENTAL RESULTS

5.1 Setup

In this section, the proposed method is evaluated using three datasets, Glass, Heart Disease, and Pima, available at

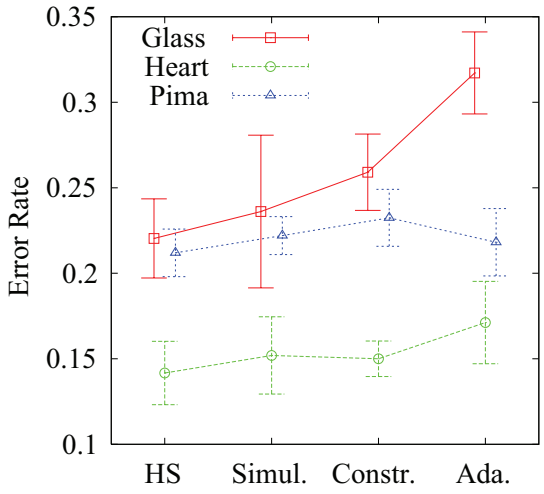

Figure 4: Comparison of Error Rates. □, ◯, and △ indicate the means for Glass, Heart, Pima datasets respectively. Lengths of error bars correspond to the standard deviation.

UCI Machine Learning Repository. All datasets are divided into three sets following the specification provided with the original dataset, i.e., first 50% as a training set, next 25% as a validation set, and the final 25% as a test set. It should be noted that the results using different separations or other schemes, e.g. n-fold cross-validation, are not directly comparable to this and many of the previous works which follow this separation.

The Glass dataset comes from the chemical analysis of glass splinters. There are nine numerical inputs and six types of glass. The Heart Disease data comes from the record of Cleveland Clinic. There are 13 numerical inputs and two classes, i.e., absence and presence of heart disease. The Pima data has 8 numerical inputs and two classes, positive or negative for diabetes.

The proposed method is compared to two other strategies of evolutionary ensemble design: constructive and simultaneous training. We implemented the constructive training as an iteration of single population EA using the fitness function defined in (7). For simultaneous training, k independent subpopulations are trained simultaneously using the fitness defined in (8). The ensemble is comprised of the best individuals of the subpopulations. The training of each population halts at the increase in validation error. As with the proposed method, a sample is classified with the winner-take-all method, and the weights of the individual networks are computed by SR and GD.

The maximum number of evaluations for evolutionary methods is 10^6. For HS, the size of the global population is 1000 and the maximum size of the species population is 100. The speciation takes place after every $\kappa = 20$ addition of new offspring. The populations of constructive and simultaneous training consist of 100 individuals respectively. The number of populations for simultaneous training is 15.

In addition to the evolutionary methods, AdaBoost [10], a very widely used population learning method, is applied to the same problems. For all methods, the number of hidden nodes was set to $m = 12$. For each method, 30 runs were performed on each dataset and the error rate $\frac{1}{T}f$, i.e.,

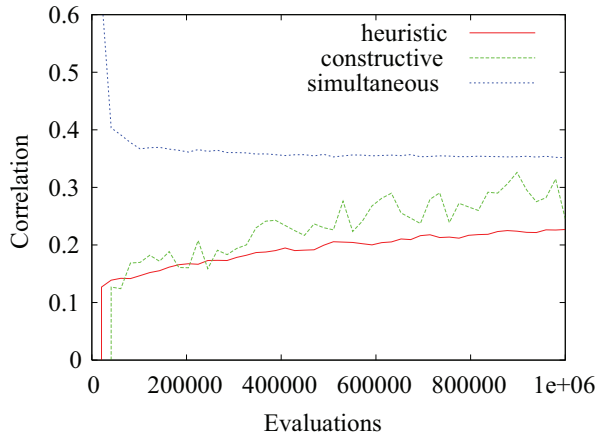

Figure 5: Average Correlation among the Ensemble in Heart Disease dataset

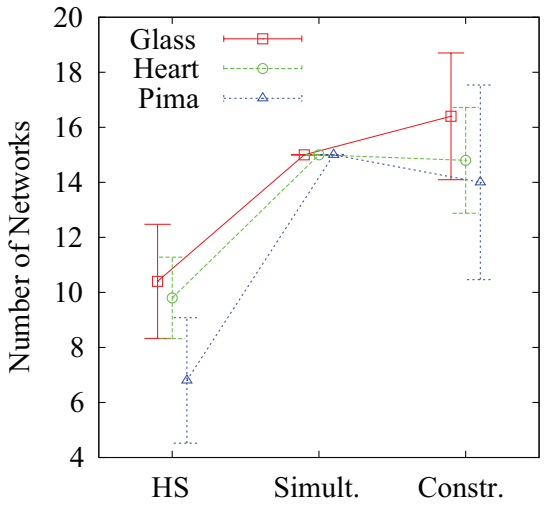

Figure 6: Number of Networks in an Ensemble

number of correctly classified test samples divided by the number of samples, is recorded.

5.2 Results

Fig. 4 illustrates the comparison of the error rates in test dataset for respective methods. The performances of evolutionary methods in Fig. 4 were of excellent quality. All methods exhibits similar or better error rates compared to the boosting algorithm. Notably, the error rates for Glass dataset, a multi-class classification task, are better than that of the boosting algorithm with a significance level of 5% or larger. Although Heuristic Speciation exhibits lower mean error rate than constructive and simultaneous training strategies in all datasets, the advantage is statistically inconclusive at this point.

To study the effect of respective evolutionary methods on the diversity of the networks, we observed the correlation among the members of the ensemble during the evolution. Following [19], the correlation ρ is defined as follows.

$$\rho = \frac{1}{\#(V)} \sum_{\mathbf{v} \in V} \sum_{i=1}^{k} \rho_i$$

$$\rho_i = (\mathbf{z}(\mathbf{N}, \mathbf{v}) - \mathbf{z}(N_i, \mathbf{v}))^T \sum_{N_j \neq i \in \mathbf{N}} (\mathbf{z}(\mathbf{N}, \mathbf{v}) - \mathbf{z}(N_j, \mathbf{v}))$$

Fig. 5 illustrates the correlation for Heart Disease dataset averaged over all runs. Each evolutionary method shows a distinguishable pattern which are consistent with other datasets as well. The correlation of simultaneous training exhibits a rapid, initial decreases, while a gradual increase is observed in other two methods. Of the two, Heuristic Speciation shows a more stable pattern of increase. The surges in constructive training correspond to an addition of a new network to the ensemble. Fig. 5 indicate that Heuristic Speciation exhibits a generally lower level of correlation over the course of the evolution, which suggests that the evaluation is reserved to ensembles of lower correlation in HS.

Fig. 6 illustrates the final ensemble size of three evolutionary methods. The ensemble size of simultaneous training is constant therefore its variance is zero. HS designs smaller ensembles for all datasets with significant difference from constructive training. It should be noted that the speciation is an intrinsically *heuristic* approach rather than an optimization of the ensemble size, which attributes to the non-trivial variance of the ensemble size.

6. CONCLUSION

In this paper, a new strategy for speciation is proposed and applied to the design of neural network ensemble. Heuristic Speciation can be viewed as a framework which combines the merits of the constructive and simultaneous training approaches. It reduces the dependence to the initial networks and the prerequisite ensemble size, while inducing the discovery and establishment of roles in a classification task.

The implementations of key features, i.e., detection of species and discrete incompatibility, were addressed by two machine learning techniques which subsequently introduce new perspectives of speciation.

The proposed method improved the average performance of the conventional training strategies as seen in Fig. 4. The difference in the performances can be attributed to the lower correlation among the ensemble. As exhibited in Fig. 5, The training of HS is more reserved to ensembles with lower correlation compared to other strategies. Since the objective of diversity is not explicitly defined as fitness function, it is a collective effect of the probabilistic clustering, the restriction of mating and competition, and niche-wise evaluation.

As observed in Fig. 6, the ensemble of HS were generally smaller than that of constructive training. This result can be credited to the heuristic property of the proposed method which increments to the ensemble only when a new role is found as a population of networks.

An important work for the future is to incorporate the topological design and the objective of regularization, both of which are of significant importance in neural network design. It remains to be seen how the concept of similarity discussed in this paper can be extended to a topologically flexible network structures.

7. REFERENCES

[1] S. Ando, J. Sakuma, and S. Kobayashi. Adaptive isolation model using data clustering for multimodal function optimization. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pp. 1417–1424, New York, NY, USA, 2005. ACM Press.

[2] S. Ando and E. Suzuki. An information theoretic approach to detection of minority subsets in database. In *ICDM '06: Proceedings of the Sixth International Conference on Data Mining*, pp. 11–20, Washington, DC, USA, 2006. IEEE Computer Society.

[3] D. Beasley, D. R. Bull, and R. R. Martin. A sequential niche technique for multimodal function optimization. *Evolutionary Computation*, 1(2):101–125, 1993.

[4] L. Breiman. Stacked regressions. *Machine Learning*, 24(1):49– 64, 7 1996.

[5] A. D. Cioppa, C. D. Stefano, and A. Marcelli. On the role of population size and niche radius in fitness sharing. *IEEE Trans. Evolutionary Computation*, 8(6):580–592, 2004.

[6] P. J. Darwen and X. Yao. Speciation as automatic categorical modularization. *IEEE Trans. Evolutionary Computation*, 1(2):101–108, 1997.

[7] K. A. De Jong. An analysis of behavior of a class of genetic adaptive systems. *Ph.D. thesis,University of Michigan*, 1975.

[8] K. DEB, A. Anand, and D. Joshi. A computationally efficient evolutionary algorithm for real-parameter optimization. *Evolutionary Computation*, 10:371–395, 2002.

[9] K. Deb and D. E. Goldberg. An investigation of niche and species formation in genetic function optimization. In J. D. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pp. 42–50, San Mateo, California, 1989. Morgan Kaufmann.

[10] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *ICML*, pp. 148–156, 1996.

[11] N. García-Pedrajas, C. Hervás-Martínez, and D. Ortiz-Boyer. Cooperative coevolution of artificial neural network ensembles for pattern classification. *IEEE Trans. Evolutionary Computation*, 9(3):271–302, 2005.

[12] D. E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the Second International Conference on Genetic Algorithms*, pp. 41–49, Hillsdale, New Jersey, 1987.

[13] V. Hanagandi and M. Nikolaou. A hybrid approach to global optimization using a clustering algorithm in a genetic search framework. *Computers and Chemical Engineering*, 22(12):1913–1925, November 1998.

[14] T. Higuchi, S. Tsutsui, and M. Yamamura. Theoretical analysis of simplex crossover for real-coded genetic algorithms. In *PPSN VI: Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, pp. 365–374, London, UK, 2000. Springer-Verlag.

[15] K. Ikeda and S. Kobayashi. GA based on the UV-structure hypothesis and its application to JSP. In *Parallel Problem Solving from Nature(PPSN VI)*, volume 1281 of *Lecture Notes in Computer Science*, pp. 273–282. Springer-Verlag, March 2000.

[16] M. M. Islam, X. Yao, and K. Murase. A constructive algorithm for training cooperative neural network ensembles. *IEEE Transactions on Neural Networks*, pp. 820–834, 2003.

[17] J. Li, M. Balazs, G. Parks, and P. Clarkson. A genetic algorithm using species conservation for multimodal function optimization. *Evolutionary Computation*, 10(3):207–234, 2002.

[18] Y. Liu and X. Yao. Simultaneous training of negatively correlated neural networks in an ensemble. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 29(6):716–725, December 1999.

[19] Y. Liu, X. Yao, and T. Higuchi. Evolutionary ensembles with negative correlation learning. *IEEE Trans. Evolutionary Computation*, 4(4):380–387, 2000.

[20] S. W. Mahfoud. Crowding and preselection revisited. In R. Männer and B. Manderick, editors, *Parallel problem solving from nature 2*, pp. 27–36, Amsterdam, North-Holland, 1992.

[21] A. Martinez-Estudillo, C. Hervas-Martinez, F. Martinez-Estudillo, and N. Garcia-Pedrajas. Hybridization of evolutionary algorithms and local search by means of a clustering method. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 36(3):534 – 545, 2006.

[22] O. Mengshoel and D. Goldberg. Probabilistic crowding: Deterministic crowding with probabilistic replacement. In W. B. et al., editor, *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 409–416. Morgan KauRmann, 1999.

[23] M. A. Potter and K. A. D. Jong. Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evol. Comput.*, 8(1):1–29, 2000.

[24] G. Singh and D. Kalyanmoy Deb. Comparison of multi-modal optimization algorithms based on evolutionary algorithms. In *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pp. 1305–1312, New York, NY, USA, 2006. ACM Press.

[25] F. Streichert, G. Stein, H. Ulmer, and A. Zell. A clustering based niching ea for multimodal search spaces. In *Proceedings of Evolution Artificielle (LNCS 2935)*, pp. 293–304. Springer-Verlag, 2003.

[26] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. *Computing Research Repository(CoRR)*, physics/0004057, 2000.

[27] R. K. Ursem. Multinational gas: Multimodal optimization techniques in dynamic environments. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 19–26. Morgan Kaufmann, 2000.

[28] X. Yin and N. Germay. A fast genetic algorithm with sharing scheme using cluster methods in multimodal function optimization. In R. F. Albrecht, C. R. Reeves, and N. C. Steele, editors, *Proceedings of the International Conference on Artificial Neural Nets and Genetic Algorithms*, pp. 450–457. Springer-Verlag, 1993.