2

Supervised Learning with Hybrid Global Optimisation Methods. Case Study: Automated Recognition and Classification of Cork Tiles

Antoniya Georgieva[1] and Ivan Jordanov[2]
University of Oxford
University of Portsmouth
United Kingdom

1. Introduction

Supervised Neural Network (NN) learning is a process in which *input* patterns and known *targets* are presented to a NN while it *learns* to recognize (classify, map, fit, etc.) them as desired. The *learning* is mathematically defined as an optimisation problem, i.e., an error function representing the differences between the desired and actual output, is being minimized (Bishop, 1995; Haykin, 1999). Because the most popular *supervised learning* techniques are gradient based (Backpropagation - BP), they suffer from the so-called Local Minima Problem (Bishop, 1995). This has motivated the employment of Global Optimisation (GO) methods for the supervised NN learning. Stochastic and heuristic GO approaches including Evolutionary Algorithms (EA) demonstrated promising performance over the last decades (Smagt, 1994; Sexton et al., 1998; Jordanov & Georgieva, 2007; etc.). EA appeared more powerful than BP and its modifications (Sexton et al., 1998; Alba & Chicone 2004), but hybrid methods that combine the advantages of one or more GO techniques and local searches were proven to be even better (Yao, 1999; Rocha et al., 2003; Alba & Chicano, 2004; Ludemir et al., 2006).

Hybrid methods were promoted over local searches and simple population based techniques in Alba & Chicone (2004). The authors compared five methods: two BP implementations (gradient descent and Levenberg-Marquardt), Genetic Algorithms (GA), and two hybrid methods, combining GA with different local methods. The methods were used for NN learning applied to problems arising in medicine. Ludemir et al. (2006) optimized simultaneously NN weights and topology with a hybrid method combining Simulated Annealing (SA), Tabu Search (TS) and BP. A set of new solutions was generated on each iteration by TS rules, but the best solution was only accepted according to the probability distribution as in conventional SA. Meanwhile, the topology of the NN was also optimized and the best solution was kept. Finally, BP was used to train the best NN topology found in the previous stages. The new methodology compared favorably with SA and TS on four classification and one prediction problems.

Plaginakos et al. (2001) performed several experiments to evaluate various training methods – six Differential Evolution (DE) implementations (with different mutation operators), BP, BPD (BP with deflection), SA, hybridization of BP and SA (BPSA), and GA. They reported

Source: Theory and Novel Applications of Machine Learning, Book edited by: Meng Joo Er and Yi Zhou,
ISBN 978-3-902613-55-4, pp. 376, February 2009, I-Tech, Vienna, Austria

poor performance for the SA method, but still promoted the use of GO methods instead of standard BP. The reported results indicated that the population based methods (GA and DE) were promising and effective, although the *winner* in their study was their BPD method. Several methods were critically compared by Rocha et al. (2003) as employed for the NN training of ten classification and regression examples. One of the methods was a simple EA, two others were combinations of EA with local searches in *Lamarckian* approach (differing in the adopted mutation operator), and their performance was compared with BP and modified BP. A hybridization of local search and EA with random mutation (*macro-mutation*) was found to be the most successful technique in this study.

Lee et al. (2004) used a deterministic hybrid technique that combines a local search method with a mechanism for escaping local minima. The authors compared its performance with five other methods, including GA and SA, when solving four classification problems. The authors reported worst training and testing results for GA and SA, and concluded that their method proposed in the paper was substantially faster than the other methods.

Yao (1999) discussed hybrid methods combining EA with BP (or other local search), suggested references to a number of papers that reported encouraging results, and pointed out some controversial results. The author stated that the best optimizer is generally problem dependant and there was no overall winner.

In our recent research (Jordanov & Georgieva, 2007; Georgieva & Jordanov, 2008a; Georgieva & Jordanov, 2008c) we investigated, developed and proposed a hybrid GO technique called Genetic *LPτ* Search (*GLPτS*), able to solve high dimensional multimodal optimization problems, which can be used for local minima free NN learning. *GLPτS* benefits from the hybridization of three different approaches that have their own specific advantages:

- *LPτ* Optimization (*LPτO*): a GO approach proposed in our earlier work (Georgieva & Jordanov, 2008c) that is based on meta-heuristic rules and was successfully applied for the optimization of low dimensional mathematical functions and several benchmark NN learning tasks of moderate size (Jordanov & Georgieva, 2007);
- Genetic Algorithms: well known stochastic approaches that solve successfully high dimensional problems (De Jong, 2006);
- Nelder-Mead Simplex Search: a derivative-free local search capable of finding quickly a solution with high accuracy, once a region of attraction has been identified by a GO method (Nelder & Mead, 1965).

In this chapter, we investigate the basic properties of *GLPτS* and compare its performance with several other algorithms. In Georgieva & Jordanov (2008a) the method was tested on multimodal mathematical functions of high dimensionality (up to 150), and results were compared with findings of other authors. Here, a summary of these results is presented and subsequently, the method is be employed for NN training of benchmark pattern recognition problems. In addition, few of the more interesting benchmark problems are discussed here.

Finally, a case study of machine learning in practice is presented: the NNs trained with *GLPτS* are employed to recognize and classify seven different types of cork tiles. This is a challenging real-world problem, incorporating computer vision for the automation of production assembly lines (Georgieva & Jordanov, 2008b). Reported results are discussed and compared with similar approaches, demonstrating the advantages of the investigated method.

2. A novel global optimisation approach for training neural networks

2.1 Introduction and motivation

In Georgieva & Jordanov (2007) we proposed a novel heuristic, population-based GO technique, called *LPτ Optimization* (*LPτO*). It utilizes *LPτ* low-discrepancy sequences of points (Sobol', 1979), in order to uniformly explore the search space. It has been proven numerically that the use of low-discrepancy point sequences results in a reduction of computational time for small and moderate dimensionality problems (Kucherenko & Sytsko, 2005). In addition, Sobol's *LPτ* points have very useful properties for higher dimensionality problems, especially when the objective function depends strongly on a subset of variables (Kucherenko & Sytsko, 2005; Liberti & Kutcherenko, 2005). In *LPτO* are incorporated novel, complete set of logic-based, self-adapting heuristic rules (meta-heuristics) that guide the search through the iterations. The *LPτO* method was further investigated in Georgieva & Jordanov (2008c) while combined with the Nelder-Mead Simplex search to form a hybrid *LPτNM* technique. It was compared with other methods, demonstrating promising results and strongly competitive nature when tested on a number of multimodal mathematical functions (2 to 20 variables). It was successfully applied and used for training of neural networks with moderate dimensionalities (Jordanov & Georgieva, 2007). However, with the increase of the dimensionality, the method experienced greater computational load and its performance worsened. This led to the development of a new hybrid technique – *GLPτS* that combines *LPτNM* with evolutionary algorithms and aims to solve efficiently problems of higher dimensionalities (up to a 150).

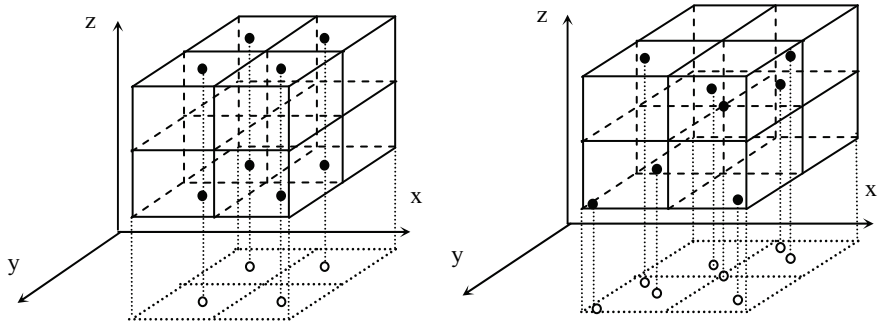
GAs are known for their very good exploration abilities and when optimal balance with their exploitation ones is found, they can be powerful and efficient global optimizers (Leung and Wang, 2001; Mitchell, 2001; Sarker et al., 2002). Exploration dominated search could lead to excessive computational expense. On the other hand, if the exploitation is favourable, the search is in danger of premature convergence, or simply of turning into a local optimizer. Keeping the balance between the two and preserving the selection pressure relatively constant throughout the whole run is important characteristic of any GA technique (Mitchell, 2001; Ali et al., 2005). Other problems associated with GA are their relatively slow convergence and low accuracy of the found solutions (Yao et al., 1999; Ali et al., 2005). This is the reason why GA are often combined with other search techniques (Sarker et al., 2002), and the same approach is adopted in our hybrid method, aiming to tackle these problems effectively by complementing GA and *LPτO* search.

The *LPτO* technique can be summarized as follows: we *seed* the whole search region with *LPτ* points, from which we select *several* promising ones to be centres of regions in which we *seed* new LP_τ points. Then we choose few promising points from the new ones and again *seed* in the neighbourhood of each one and so on, until a halting condition is satisfied. By combining *LPτO* with GA of moderate population size, the aim is to explore the search space and improve the initial *seeding* with *LPτ* points by applying genetic operators in a few generations. Subsequently, a heuristic-stochastic rule is applied in order to select some of the individuals and to start *LPτO* search in the neighbourhood of each of the chosen ones. Finally, we use a local Simplex Search to refine the solution and achieve better accuracy.

2.2 *LPτ* low-discrepancy points

Low-discrepancy sequences (LDS) of points are deterministically generated uniformly distributed points (Niederreiter, 1992). *Uniformity* is an important property of a sequence

which guarantees that the points are *evenly* distributed in the whole domain. When comparing two uniformly distributed sequences, features as *discrepancy* and *dispersion* are used in order to quantify their uniformity. Two different uniform sequences in three dimensions are shown in Fig. 1. The advantage of the low-discrepancy sequences is that they avoid the so called shadow effect, i.e., when projections of several points on the projective planes are coincident.



(a) Cubic sequence (b) *LPτ* low-discrepancy sequence.

Fig. 1. Two different uniform sequences.

As it can be seen from Fig.1, the projections of the cubic sequence give four different points on the projective plane, each of them repeated twice, while the *LPτ* sequence gives eight different projection points. Therefore, the low-discrepancy sequence would describe the function behaviour in this plane much better than the cubic one; this advantage is enhanced with the increase of the dimensionality and the number of points. This feature is especially important when the function at hand is weakly dependent on some of the variables and strongly dependent on the rest of them (Kucherenko & Sytsko, 2005).

The application of LDS in GO methods was investigated in Kucherenko & Sytsko (2005), where the authors concluded that the Sobol's *LPτ* sequences are superior to the other LDS. Many useful properties of *LPτ* points have been shown in Sobol', (1979) and tested in Bratley & Fox (1988), Niederreiter (1992), and Kucherenko & Sytsko (2005). The properties of LDS could be summarized as follows:

- retain their properties when transferred from a unit hyper-cube to a hyper-parallelepiped, or when projected on any of the sides of the hyper-cube;
- explore the space better avoiding the *shadowing effect* discussed earlier. This property is very useful when optimising functions that depend weakly on some of the variables, and strongly on the others;
- unlike the conventional random points, successive LDS have *memory* and *know* about the positions of the previous points and try to fill the gaps in between (this property is true for all LDS and is demonstrated in Fig. 2);
- it is widely accepted (Sobol', 1979; Niederreiter, 1992) that no infinite sequence of N points can have discrepancy ρ that converges to zero with smaller order of magnitude than $O(N^{-1}\log^n(N))$, where n is the dimensionality. The *LPτ* sequence satisfies this estimate. Moreover, due to the way *LPτ* are defined, for values of $N = 2^k$, $k = 1, 2, \ldots, 31$, the discrepancy converges with rate $O(N^{-1}\log^{n-1}(N))$ as the number of points increases (Sobol', 1979).

2.3 The *LPτO* meta-heuristic approach

Stochastic techniques depend on a number of parameters that play decisive role for the algorithm performance assessed by speed of convergence, computational load, and quality of the solution. Some of these parameters include the number of initial and subsequent trial points, and a parameter (or more than one) that defines the speed of convergence (cooling temperature in SA, probability of mutation in GA, etc.). Assigning values to these parameters (*tuning*) is one of the most important and difficult parts from the development of a GO technique. The larger the number of such decisive parameters, the more difficult (or sometimes even impossible) is to find a set of parameter values that will ensure an algorithm's good performance for as many as possible functions. Normally, authors try to reduce the number of such *user defined* parameters, but one might argue that in this way, the technique becomes less flexible and the search depends more on random variables.

The advantage of the *LPτO* technique is that the values of these parameters are selected in a meta-heuristic manner – depending on the function at hand, while guided by the user. For example, instead of choosing a specific number of initial points N, in *LPτO*, a range of allowed values ($N_{\min}$ and $N_{\max}$) is defined by the user and the technique adaptively selects (using the *filling-in the gaps* property of *LPτ* sequences) the smallest allowed value that gives enough information about the landscape of the objective function, so that the algorithm can continue the search effectively. Therefore, the parameter N is exchanged with two other user-defined parameters ($N_{\min}$ and $N_{\max}$), which allows flexibility when N is selected automatically, depending on the function at hand. Since the method does not assume a priori knowledge of the global minimum (GM), all parts of the parameter space must be equally treated, and the points should be uniformly distributed in the whole region of initial searched. The *LPτ* low-discrepancy sequences and their properties fulfill this issue satisfactorily. We also use the property of *LPτ* sequences that additional points *fill the gaps* between the other *LPτ* points. For example, if we have an *LPτ* sequence with four points and we would like to double their number, the resulting sequence will include the initial four points plus the new four ones positioned in-between them. This property of the *LPτ* sequences is demonstrated in Fig. 2.

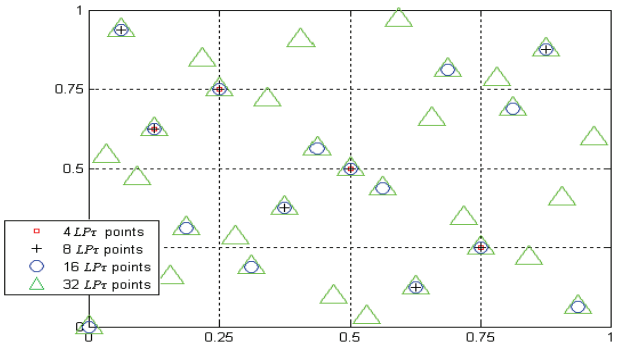


Fig. 2. *Fill in the gaps* property of the *LPτ* sequences.

As discussed above, when choosing the initial points of *LPτO*, a range of allowed values ($N_{\min}$ and $N_{\max}$) is defined and the technique adaptively selects the smallest possible value

that gives enough information about the landscape of the objective function, so that the algorithm can continue the search effectively. Simply said, after the minimal possible number of points is selected, the function at hand is investigated with those points, and if there are not *enough promising points*, additional ones are generated and the process is repeated until an *appropriate* number of points is selected, or the maximal of the allowed values is reached.

Another example of the meta-heuristic properties of *LPτO* is the parameter that allows switching between exploration and exploitation and, thus, controls the convergence of the algorithm. In *simulating annealing* (SA), this is done by the *cooling temperature* (decreased by *annealing shedule*); in GA - by the *probability of mutation*, etc. These parameters are user-defined at the beginning of the search. In the *LPτO* method, the convergence speed is controlled by the size of future regions of interest, given by a radius R, and, in particular, the *speed* with which R decreases (Georgieva & Jordanov, 2008c). If R decreases slowly, then the whole search converges slowly, allowing more time for exploration. If R decreases quickly, the convergence is faster, but the risk of omitting a GM is higher. In the *LPτO*, the decrease/increase step of R is not a simple user-defined value. It is determined adaptively on each iteration and depends on the current state of the search, the *importance* of the region of interest, as well as the complexity of the problem (dimensionality and size of the searched domain). The convergence speed depends also on a parameter M, which is the maximum allowed number of future regions of interest. M is a user defined upper bound of the number of future regions of interest M_{new}, while the actual number is adaptively selected at each iteration within the interval [1, M]. The GO property of *LPτO* to escape local minima is demonstrated in Fig. 3, where the method locates four regions of interest and after a few iterations detects the GM.

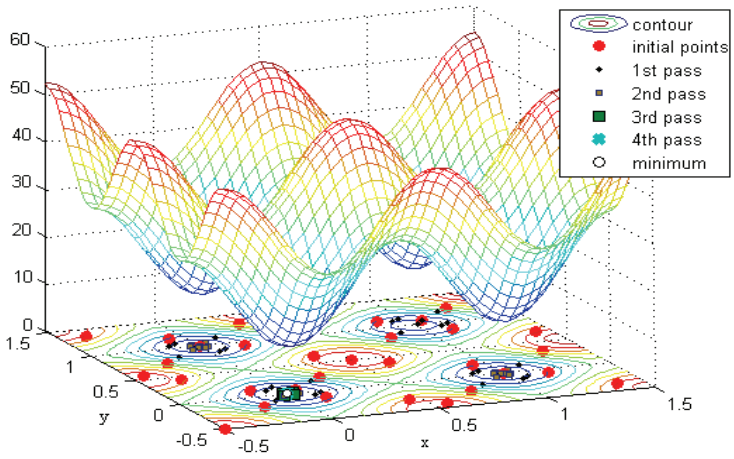


Fig. 3. Two-dimensional Rastrigin function with three local and one global minima, optimized with *LPτO*.

The convergence stability of *LPτO* with respect to these parameters (in particular M and N_{max}), the stability of the method with respect to the initial points and the searched domain, the analytical properties of the technique and the results from testing on a number of benchmark functions are further analysed and discussed in Georgieva & Jordanov (2008c).

2.4 GA and Nelder-Mead simplex search

General information for GA and their properties can be found in Mitchell (2001). We use conventional one-point recombination and our mutation operator is the same as in (Leung & Wang, 2001). We keep constant population size, starting with G individuals. The general form of the performed GA is:

Step 1. From the current population $p(G)$, each individual is selected to undergo recombination with probability P_r. If the number of selected individuals is odd, we dispose of the last one selected. All selected individuals are randomly paired for *mating*. Each pair produces two new individuals by recombination;

Step 2. Each individual from the current population $p(G)$ is also selected to undergo mutation with probability P_m;

Step 3. From the *parent* population and the *offspring* generated by recombination and mutation, the best G individuals are selected to form the new generation $p(G)$.

Step 4. If the halting condition is not satisfied, the algorithm is repeated from step 1.

Further details of the adopted GA can be found in Georgieva & Jordanov (2008a). The Nelder-Mead (*NM*) simplex method for function optimization is a fast local search technique (Nelder & Mead, 1965), that needs only function values and requires continuity of the function. It has been used in numerous hybrid methods to refine the obtained solutions (Chelouah & Siarry, 2003; 2005), and for coding of GA individuals (Hedar & Fukushima, 2003). The speed of convergence (measured by the number of function evaluations) depends on the function values and the continuity, but mostly, it depends on the choice of the initial simplex - its coordinates, form and size. We select the initial simplex to have one vertex in the best point found by the *LPτO* searches and another n vertices distanced from it in a positive direction along each of its n coordinates, with a coefficient λ. As for the choice of the parameter λ, we connect it with the value of R_1, which is the average distance between the testing points in the region of attraction, where the best solution is found by *LPτO*.

2.5 The *GLPτS* technique: hybridization of GA, *LPτO* and Nelder-Mead search

Here, we introduce in more detail the hybrid method called *Genetic LPτ and Simplex Search* (*GLPτS*), which combines the effectiveness of GA during the early stages of the search with the advantages of *LPτO*, and the local improvement abilities of NM search (further discussion of the method can be found in Georgieva & Jordanov (2008a).

Based on the complexity of the searched landscapes, most authors intuitively choose population size for their GA that could vary from 100s to 1000s (De Jong, 2006). We employ smaller number of points that leads to a final population with promising candidates from regions of interest, but not necessarily to a GM. Also, our initial population points are not random (as in a conventional GA), but uniformly distributed *LPτ* points.

Generally, the technique could be described as follows:

Step 1. Generate a number I of initial *LPτ* points;

Step 2. Select G points, ($G < I$), that correspond to the best function values. Let this be the initial population $p(G)$ of the GA;

Step 3. Perform GA until a halting condition is satisfied;

Step 4. From the population $p(G)$ of the last GA generation, select g points of future interest ($1 \le g \le G/2$);

Step 5. Initialize the *LPτO* search in the neighbourhood of each selected point;

Step 6. After the stopping conditions of the *LPτO* searches are satisfied, initialize a local NM search in the best point found by all *LPτO* searches.

To determine the number g of subsequent *LPτO* searches (**Step 4**), the following rule is used (illustrated in Fig. 4):

Let $p(G)$ be the population of the last generation found by the GA run. Firstly, all G individuals are sorted in non-descending order using their fitness values and then rank r_i is associated to the first half of them by using formula (1):

$$r_i = \frac{f_{\max} - f_i}{f_{\max} - f_{\min}}, \; i = 2, \ldots, G/2. \tag{1}$$

In (1), $f_{\max}$ and $f_{\min}$ are the maximal and minimal fitness values of the population and the rank r_i is given with a linear function which decreases with the growth of f_i, and takes values within the range [0, 1].

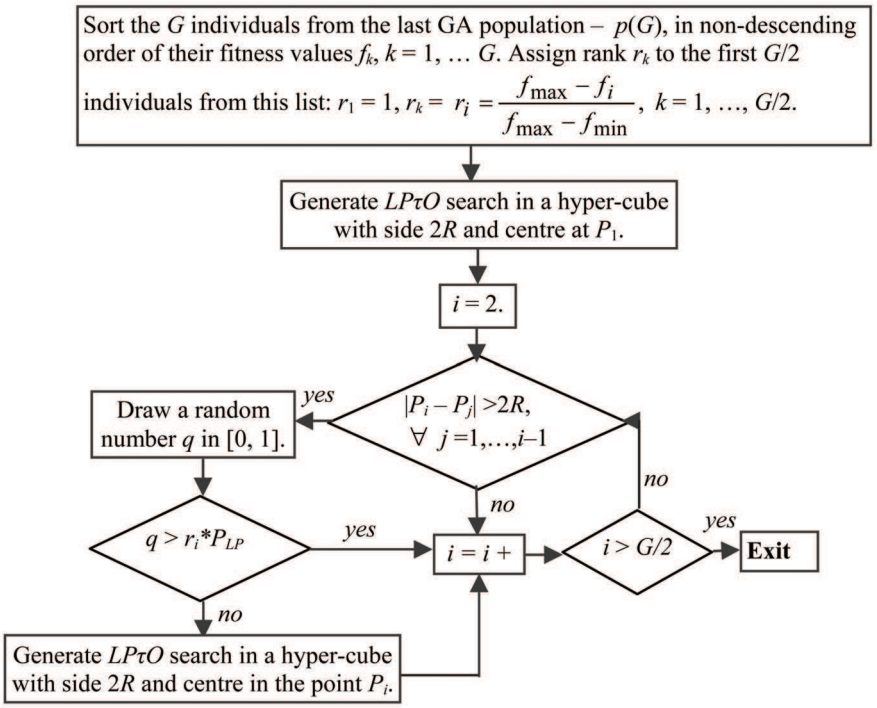


Fig. 4. Algorithm for adaptive selection of points of future interest from the last population of the GA run.

The best individual of the last population $p(G)$ has rank $r_1 = 1$ and always competes. It is used as a centre for a hyper-cube (with side $2R$), in which the *LPτO* search will start. The parameter R is heuristically chosen with formula (2)

$$R = 50/G + \text{int}_{\max}*0.001, \tag{2}$$

where int_{max} is the largest of all initial search intervals. This parameter estimates the trade-off between the computational expense and the probability of finding a GM. The greater the population size G, the smaller the intervals of interest that are going to be explored by the *LPτO* search. The next individual P_i, $i = 2, \ldots, G/2$ is then considered, and if all of the Euclidean distances between this individual and previously selected ones are greater than $2R$ (so that there is no overlapping in the *LPτO* search regions), another *LPτO* search will be initiated with a probability $r_i P_{LP}$. Here P_{LP} is a user-defined probability constant in the interval [0, 1]. In other words, individuals with higher rank (that corresponds to lower fitness) will have greater chance to initiate *LPτO* searches. After the execution of the *LPτO* searches is completed, Nelder-Mead Local Simplex Search is applied to the best function value found in all previous stages of *GLPτS*.

3. Testing *GLPτS* on mathematical optimisation problems and benchmark NN learning tasks

3.1 Mathematical testing functions

Detailed testing results of *GLPτS* on multi-dimensional optimization functions are reported in Georgieva & Jordanov (2008a). Here, we only demonstrate the results of testing *GLPτS* on 30 and 100 dimensional problems for which a comparison with several other GO approaches was possible. The results, in terms of average (over 100 runs) number of function evaluations, are scaled logarithmically for better visualization and are shown in Fig. 5.

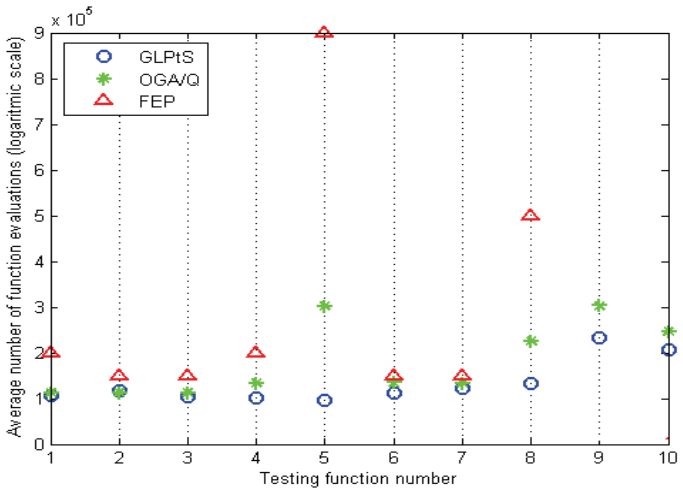


Fig. 5. Average number of function evaluations for ten test functions: comparison of *GLPτS* with needed Orthogonal Genetic Algorithm with Quanitsation (OGA/Q, Leung & Wang, 2001) and FEP (Yao et al., 1999).

When compared to the other evolutionary approaches, it can be seen from Fig. 5 that *GLPτS* performed very efficiently. In addition, the comparison with Differential Evolution in Georgieva & Jordanov (2008a) for lower dimensional problems helped us conclude that *GLPτS* is a promising *state-of-the-art* GO approach solving equally well both low and high-dimensional problems.

3.2 NN learning benchmark problems

Subsequently, we employed the *GLPτS* for minimizing the error function in NN learning problems and the results were reported in Georgieva & Jordanov, (2006). Here, we present only few interesting examples of using *GLPτS* for NN training.

The architectures of the investigated NNs comprise static, fully connected between the adjacent layers topologies with a standard sigmoidal transfer functions. The training is performed in a batch-mode, i.e., all of the training samples are presented to the NN at one go. The NN weight vector is considered an n-dimensional real Euclidean vector $\overline{W}$, obtained by concatenating the weight vectors for each layer of the network. The *GLPτS* global optimisation algorithm is then employed to minimize the objective function (the NN error function) and to perform optimal training. The proposed algorithm is tested on well-known benchmark problems with different dimensionalities. For comparison, a BP (Levenberg-Marquardt) is also employed and performed using Matlab NN Toolbox. Both methods are ran 50 times and their average values are reported.

Classification of XOR Problem

For the classification of the XOR, which is a classical toy problem (Bishop, 1995), the minimal configuration of a NN with two inputs, two units in the hidden layer, and one output is employed. The network also has a bias, containes 9 connection weights, and therefore, defines n = 9 dimensional optimization problem. There are P = 4 input-target patterns for the training set. It can be seen from the Fig. 6 that after the 20th epoch, BP did not improve the error function, while our method continued minimizing it. To assess the ability of the trained NN to generalize, tests with 100 random samples of noisy data are performed, where the noise is up to 15%. Obtained optimal results from the training and testing are given in Table 1 (Georgieva & Jordanov, 2006).

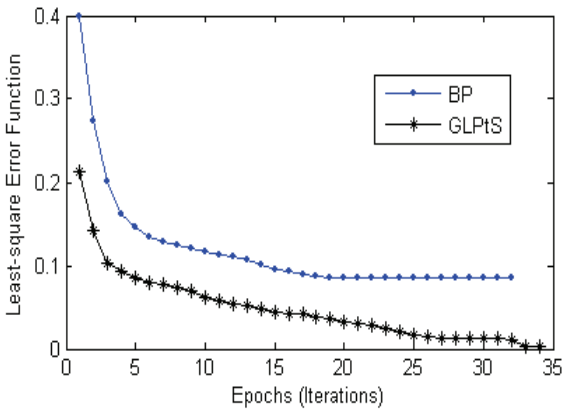


Fig. 6. Error function for the XOR problem when BP and *GLPτS* are used.

Predicting the rise time of a servo mechanism

The Servo data collection represents an extremely non-linear phenomenon (Quinlan, 1993; Rocha et al., 2003) – predicting the rise time of a servomechanism, depending on four attributes: two gain settings and two mechanical linkages. The database consists of 167 different samples with continuous output (the time in seconds). In order to avoid

Criterion Method	Error Function (Std. Dev.)	Mean Test Error (Std. Dev.)
BP	0.08 (0.09)	0.1987 (0.0290)
GLPτS	7.6e-08 (7e-08)	8.3e-07 (3.3e-7)

Method: BP – Backpropagation with Levenberg-Marquardt optimisation (the source of Matlab NN Toolbox is used).

Table 1. Optimal errors for the *GLPτS* and BP (XOR problem).

computational inaccuracies, we normalized the set of outputs to have a zero mean and unit standard deviation. A network with a 4-4-1 architecture (25-dimensional problem) is employed to produce a continuous output. The dataset is divided into two parts – one batch of 84 training samples and second batch of 83 testing ones. In this case, the transfer function in the output layer is changed to a linear function (instead of a sigmoidal one) in order to be able to produce output outside the [0, 1] interval. Obtained optimal solutions for the train and test errors are given in Table 2 and Fig. 7 illustrates the average values of the errors for each testing sample for both BP and *GLPτS*. One can see from the figure that there are more outliers in the case of BP and that overall, a smaller mean test value is achieved by the *GLPτS* method.

Criterion Method	Error Function (Std. Dev.)	Mean Test Error (Std. Dev.)
BP	0.0474 (0.06)	0.4171 (0.5515)
GLPτS	0.0245 (0.005)	0.2841 (0.4448)

Table 2. Optimal errors for the *GLPτS* and BP (*Servo* problem).

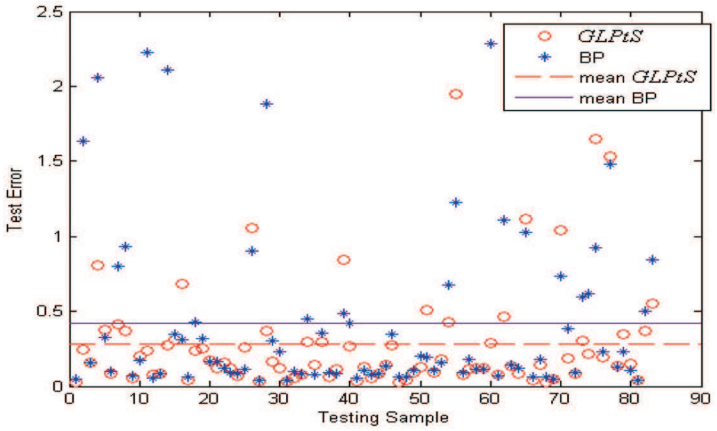


Fig. 7. Test errors and mean test errors for BP and *GLPτS*.

Classification of Pima Indians Diabetes Database

In the *Diabetes* data collection, the investigated, binary-valued variable is used to diagnose whether a patient shows signs of diabetes or not (Rocha et al., 2003). All patients are females of at least 21 years old and of Pima Indian heritage. The data set comprises 500 instances

that produce an output 0 (non-positive for diabetes), and 268 with output 1 (positive for diabetes). Each sample has 8 attributes: number of times pregnant, age, blood test results, etc. In order to avoid computational inaccuracies, in our experiment all attributes are normalized to have a zero mean and a unit standard deviation. A network with 8-8-1 architecture (81-dimensional problem) is adopted to produce continuous output in the range [0, 1]. The dataset is divided into two parts – training subset of 384 samples (145 of which correspond to output 1), and testing subset of the same number of patterns. Table 3 shows the obtained optimal solutions for the training and testing errors.

Criterion Method	Error Function (Std. Dev.)	Mean Test Error (Std. Dev.)
BP	0.0764 (0.07)	0.2831 (0.2541)
GLPτS	0.001 (0.005)	0.2619 (0.3861)

Table 3. Optimal errors for the *GLPτS* and BP (*Diabetes* problem)

Function Fitting Regression Example

We also performed a function fitting example, for which the network is trained with noisy data. The function to be approximated is the Hermit polynomial:

$$G(x) = 1.1(1-x+2x^2)\exp(-x^2/2).$$

The set up of the experiment is the same as reported in Leung *et al.* (2001), with the only difference that we use batch-mode instead of on-line training. The test results from 2000 testing samples and 20 independent runs of the experiment are shown in Table 4. It can be seen from the table that our results improve slightly the best ones reported in Leung *et al.* (2001). Fig. 8 graphically illustrates the results and shows the Gaussian noise that we used for training, the function to be approximated, and the NN output.

Criterion Method	Average	Max	Min	Std. Dev.
RLS	0.1901	0.2567	0.1553	0.0259
IPRLS	0.1453	0.1674	0.1207	0.0076
TWDRLS	0.1472	0.1711	0.1288	0.0108
GLPτS	0.1349	0.1602	0.1184	0.01

Method: By Leung et al. (2001): RLS – Recursive Least Squares; IPRLS – Input Perturbation RLS; TWDRLS – True Weight Desay RLS.

Table 4. Test results for the *GLPτS* and the methods in Leung et al. (2001).

The results from the classification experiments (Table 1, Table 2, and Table 3) show that the achieved by *GLPτS* least-square errors are at least twice better than the BP ones. The multiple independent runs of our method also show that the obtained solutions are stable with small deviations. As it can be seen from Table 1, in the case of XOR, the *GLPτS* method outperforms BP considerably (BP with mean error of 0.08, in comparison with 7.6e-8 for the proposed here method). For this task Wang et al. (2004), also reported low success rate for BP with frequent entrapment in local minima. In the case of *Servo* problem, the superiority of our method is not so dominant (as in the case of XOR), but still the results in Table 2 show

better standard deviation of both measures – 0.005 against 0.06 for the error function, and 0.44 against 0.55 for the test error. This indicates a better and more stable solution for our method. The reported in Rocha et al. (2003) results from five different methods for the same task and architecture are also with worse error function values compared to ours. Those observations indicate that further improvement of the solution could not be found for the investigated 4-4-1 NN architecture, nevertheless, experiments with different architectures could lead to better results. The comparison of the training results for *Diabetes* given in Rocha et al. (2003), also confirms the advantages of the *GLPτS* method.

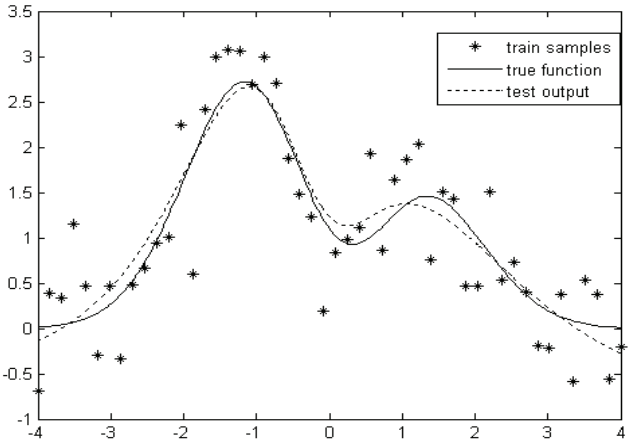


Fig. 8. Output of the network trained with *GLPτS* for the function fitting example.

4. Machine learning in practise: an intelligent machine vision system

4.1. Introduction and motivation

In (Georgieva & Jordanov, 2008b) we investigate an intelligent machine vision system that uses NNs trained with *GLPτS* for pattern recognition and classification of seven types of cork tiles with different texture. Automated visual inspection of products and automation of product assembly lines are typical examples of application of machine vision systems in manufacturing industry (Theodoridis & Koutroumbas, 2006). At the assembly line, the objects of interest must be classified in *a priori* known classes, before a robot arm places them in the right position or box. In the area of automated visual inspection, where decisions about the adequacy of the products have to be made constantly, the use of pattern recognition provides an important background (Davies, 2005).

Cork is a fully renewable and biodegradable sustainable product obtained from the bark of the cork oak tree. Although the primary use of cork is in the wine stoppers production (70% of the total cork market), cork floor and wall covering give about 20% of the total cork business (WWF, 2006). Cork oak plantations have proven biodiversity, environmental and economical values. Recent increase of alternative wine stoppers arises serious attention and concerns, since this is reducing the economical value of cork lands and might lead to abandonment, degradation and loss of irreplaceable biodiversity (WWF, 2006). On the other hand, in the past several years of technological advancement, cork has become one of the

most effective and reliable natural materials for floor and wall covering. Some of the advantages of the cork tiles are their durability, ability to reduce noise, thermal insulation, and reduction of allergens. Many of the cork floors installed during the "golden age" of cork flooring (Frank Lloyd Wright's *Fallingwater*; *St. Mary of the Lake* Chapel in Mundelein (USA); US Department of Commerce Building, etc.) are actually still in use, which is the best proof of their durability and ever-young appearance.
Image analysis techniques have been applied for automated visual inspection of cork stoppers in (Chang et al., 1997; Radeva et al., 2002; Costa & Pereira, 2006), and according to the authors, the image-based inspection systems have high production rates. Such systems are based on a line-scan camera and a computer, embedded in an industrial sorting machine which is capable of acquiring and real-time processing of the product surface image.

4.2 Database and features extraction

The aim of this case study was to design, develop and investigate an intelligent system for visual inspection that is able to automatically classify different types of cork tiles. Currently, the cork tiles are sorted "by hand" (e.g., see www.expanko.com), and the use of such a computerized system could automate this process and increase its efficiency. We experimented with seven types of cork wall tiles with different texture. The tiles used in this investigation are available on the market by www.CorkStore.com and samples of each type are shown in Fig. 9.

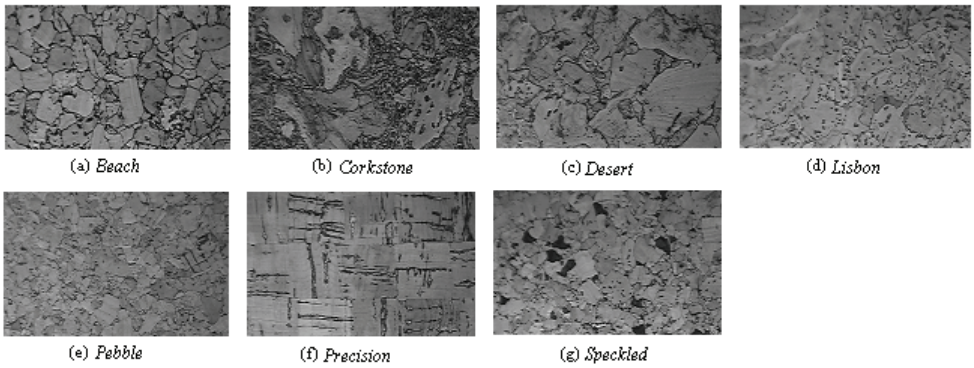
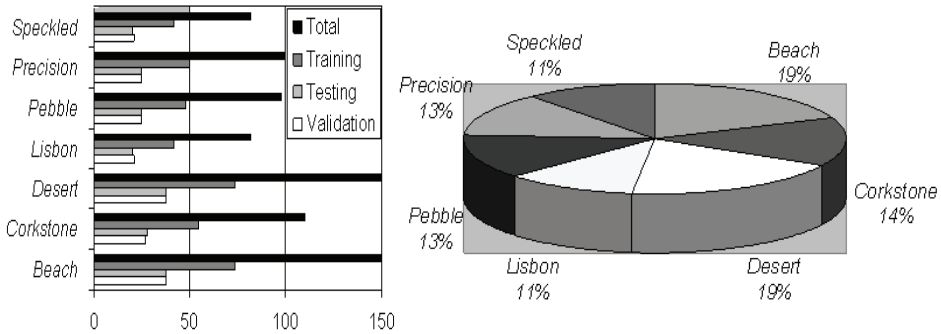


Fig. 9. Images taken with our system: samples from the seven different types of wall cork tiles.

The functionality of our visual system is based on four major processing stages: image acquisition, features extraction (generation and processing), NN training, and finally NN testing. For the image acquisition stage, we used a Charge-Coupled Device (CCD) camera with a focal length 5-50 mm that is capable of capturing fine details of the cork texture. For all cork types we used grayscale images of size 230x340 pixels and, in total, we collected 770 different images for all classes. Fig. 10 shows the percentage distribution of each type of cork tiles. We used 25% of all images for testing (not shown to the NN during training) and assessing the generalization abilities of the networks.
The first step of the features generation stage was to reduce the effects of illumination. Subsequently, we used two classical approaches to generate image texture characteristics: the Haralick 'sco-occurrence method (Haralick et al., 1973) and the Laws' filter masks (Laws, 1980). Both methods were employed and the obtained features were used to generate one

dataset, without taking into account the feature generation technique. This approach resulted in obtaining 33 features for each image (8 co-occurrence characteristics and 25 Laws' masks). These features were further processed statistically with Principal Component Analysis (PCA) and Linear Discriminent Analysis (LDA) in order to extract the most valuable information and to present it in a compact form, suitable for NN training (Bishop, 1995). Before processing the data, we took out 120 samples to be used later as a testing subset, therefore, this data was not involved in the feature analysis stage. All additional details of this case study, can be found in Georgieva & Jordanov (2008b).



(a) (b)

Fig. 10. Dataset sample distribution (50% training, 25% testing, 25% validation): (a) Number of samples from each cork type; (b) The percentage distribution of each cork type.

4.3 Neural network training and testing

NNs with three different topologies (with biases) were employed and different coding of the seven classes of interest was used. In the first case, a NN with three neurons in the output layer (with *Heaviside* transfer function) was employed. The seven classes were coded as binary combinations of the three neurons ('*1-of-c*' coding, as proposed in Bishop, 1995), e.g., *Beach* was coded as (0, 0, 0), *Corkstone* as (1, 0, 0), etc. The last, (8th) combination (1, 1, 1) was simply not used. In the second designed topology, the output layer contained only one neuron (with *Tanh* transfer function and continuous output). Since the *Tanh* function has values in [-1, 1], the seven classes were coded as (-0.8571, -0.5714, -0.2857, 0, 0.2857, 0.5741, 0.8571) respectively. When assessing the system generalization abilities, we considered each testing sample as correctly classified if $|\text{output} - \text{target}| < 0.14$. For the last topology was used an output layer with seven neurons and a *Heaviside* transfer function. Each class was coded as a vector of binary values where only one output is 1, and all others are 0. For example, *Beach* was coded as (1, 0, 0, 0, 0, 0, 0), *Corkstone* as (0, 1, 0, 0, 0, 0, 0), etc.

The number of neurons in the input layer depends on the number of features (K) that characterize the problem samples. Utilizing the *rules of thumb* given by Heaton (2005) and after experimenting, the number of neurons in the hidden layer was chosen to be $N = 7$. The three different architectures were employed for both datasets, obtained by the PCA and LDA respectively, processing: K-7-3 (3-binary coding of the targets), K-7-1 (continuous coding of the targets), and K-7-7 (7-binary coding), where K is the number of features. At the system evaluation stage, 25% of the total data were used as a testing set, only 1/3 of which was present at the feature analysis phase (used in the preprocessing with PCA and LDA)

and the remaining 2/3 of the test set were kept untouched. Further on, we considered the testing results as average test errors for both testing subsets. Rigorous tests when a validation set is used were performed and the results can be found in Georgieva & Jordanov (2008b).

Feature Set	Measure	Three outputs (*binary coding*)	One output (*continuous coding*)
PCA	MSE (std), [min, max]	0.052 (0.0094) [0.03, 0.074]	0.014 (0.0044) [0.011, 0.036]
	Test rate, [min, max]	86% [79%, 94%]	66% [41%, 77%]
LDA	MSE (std), [min, max]	0.0038 (0.0029) [0, 0.014]	0.0037 (0.0022) [0.0005, 0.0113]
	Test rate, [min, max]	95% [88%, 99%]	88% [74%, 98%]

Feature set: Principal Component Analysis (PCA) and Linear Discriminant Analysis – discussed in Georgieva & Jordanov, 2008b.

Table 5. Neural Network Training with *GLPτS* and Performance Evaluation: two different datasets with binary and continuous output.

Table 5 shows training and testing results for both topologies with $K = 7$ for the PCA dataset and $K = 6$ for the LDA dataset. In Table 5 the MSE (mean squared error) and standard deviation (given in parentheses) for 50 runs are independently reported for each dataset. The minimal and maximal values obtained for the different runs are also shown in this table. The system was evaluated with the testing rate, given by the percentage of correctly classified samples from the test set. Similarly, Table 6 shows results for the same topologies and datasets, with the only difference being the NN training technique. For the training of the NNs in Table 5, *GLPτS* was used, and for Table 6 – the Matlab implementation of gradient-based *Levenberg-Marquardt* minimisation, denoted here as *Backpropagation* (BP). All test results are jointly illustrated in Fig. 11. The analysis of the results given in Table 5, Table 6, and Fig. 11, led to the following conclusions:

- The generalisation abilities of the NNs trained with *GLPτS* were strongly competitive when compared to those trained with BP. The best testing results of 95% were obtained for NN trained with *GLPτS*, LDA dataset, and three binary outputs;
- In general, the BP results were not as stable as the *GLPτS* ones, having significantly larger differences between the attained minimal and maximal testing rate values. This is due to entrapment of BP in local minima that resulted in occasional very poor solutions;
- The LDA dataset results had better testing rate and smaller MSE than those corresponding to the PCA dataset. In our view this advantage is due to the LDA property to look for optimal class separability;
- The three-output binary coding of the targets led to a NN architecture with higher dimensionality, but gave better results than the continuous one. This is not surprising, since the binary coding of the targets provided linearly independent outputs for the different classes, which is more suitable for classification tasks compared to continuous

coding (Bishop, 1995). However, in the case of seven binary outputs, the NN performance deteriorated, since the dimensionality was increased unnecessarily.

Feature Set	Measure	Three outputs (*binary coding*)	One output (*continuous coding*)
PCA	MSE (std), [min, max]	0.025 (0.053) [0.001, 0.245]	0.0489 (0.1473) [0.0113, 0.9116]
	Test rate, [min, max]	85% [39%, 94%]	71% [0%, 85%]
LDA	MSE (std), [min, max]	0.022 (0.06) [0, 0.244]	0.0049 (0.027) [0, 0.1939]
	Test rate, [min, max]	89% [40%, 98%]	90% [45%, 98%]

Table 6. Neural Network Training with BP and Performance Evaluation: two different datasets with binary and continuous output.

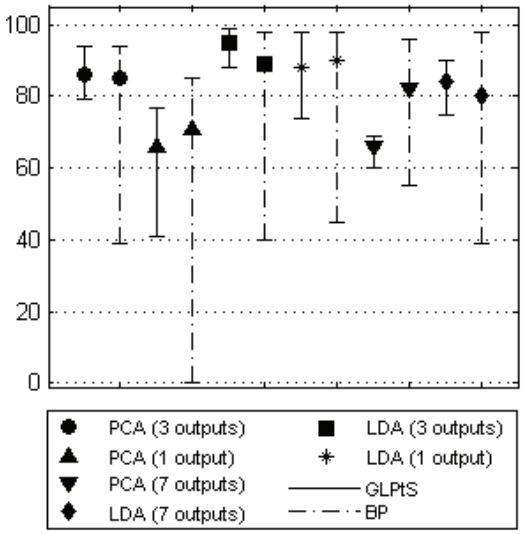


Fig. 11. Mean, min, and max test success rate (Table 5 and Table 6) for the experiments with different datasets, NN topologies, and learning approaches.

Further on, we considered only the two cases with 3-binary and 1-continuous coding (as well as NN trained with *GLPτS*), as the most interesting and successful ones. Fig. 12 illustrates the testing success rate for the two NN topologies for both datasets (PCA and LDA) with respect to the increasing number of training samples. The idea was to assess whether the number of used samples and features gave comprehensive and reliable information for the different cork classes. We used 25% of the whole data as an unseen

testing subset and started increasing the percentage of used samples when training, keeping the NN topology unchanged. If the success rate increases proportionally to the increase of the training set size, then the features can be considered to be reliable (Umbaugh, 2005). The results illustrated in Fig. 12 were averaged over 20 runs. One can see from Fig. 12 that for both NN architectures, LDA gives better generalisation results than PCA. It can also be seen that for all combinations (datasets and coding), the test rate graphs are ascendant, but the increased of number of training samples above 60% hardly brings any improvement of the test error success rate (with the exception of the LDA – binary architecture).

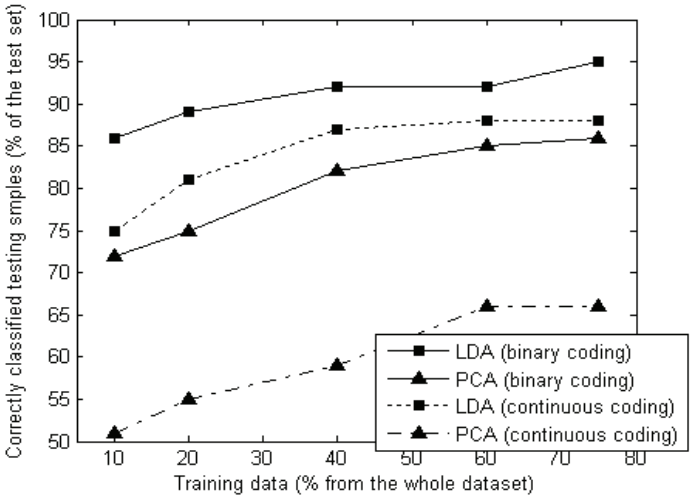


Fig. 12. Test success rate for increasing number of samples in the training set. PCA and LDA feature sets are considered with binary and continuous coding of the classes.

4.4 Comparison with results of other authors

Straightforward comparison of our results with findings for similar cork classification systems (Chang et al., 1997; Radeva et al., 2002; Costa & Pereira, 2006) is a difficult task, because of the many differences in the parameters and techniques. Some of the main differences can be listed as follows:

- Automated systems for cork products inspection have been developed only for cork stoppers and planks, but not for cork tiles;
- While natural cork stoppers are manufactured by punching a one-piece cork strip (which may have cracks and insect tunnels), cork tiles consist of various sizes of granules compressed together under high temperature, and cracks are not likely to be expected to appear. In (Chang et al., 1997; Radeva et al., 2002; Costa & Pereira, 2006), the authors are looking mostly for pores, cracks and holes (and their sizes) in cork stoppers, whereas in our case, gray density (texture) changes and overall appearance is of interest. We use feature generation techniques that capture the images texture information, while in (Chang et al., 1997; Radeva et al., 2002; Costa & Pereira, 2006) the authors use features that aim to identify cracks and holes;
- In Costa & Pereira (2006) the authors employ only LDA as a classifier and in (Chang et al., 1997) the investigation does not include any feature analysis techniques at all. In our

experiment, after using LDA and PCA to reduce the dimensionality of the problem space, we used *GLPτS* method for optimal NN learning. Other authors relay on different classifiers (Nearest Neighbor, Maximal likelihood, Bayesian classifier (Radeva et al., 2002), Fuzzy-neural networks (Chang et al., 1997), LDA (Costa & Pereira, 2006);

- The size of training and testing datasets and the size of the investigated images vary significantly.

In our study, we showed that LDA could reach up to 95% success rate for a task with seven classes, providing that the classifier is well designed and combined with NN (trained with *GLPτS* method). We claim that LDA is computationally efficient and very useful technique when the other stages of the system process – feature generation and appropriate classifier design are thoroughly thought and investigated. On the other hand, ICA is not suitable for all types of data, because it imposes independence conditions on the features and also involves additional computational cost (Theodoridis & Koutroumbas, 2006; Radeva et al., 2002). Considering the above-mentioned results, we can conclude that the intelligent classification system investigated has very good and strongly competitive generalization abilities (Table 7).

System	Costa & Pareira	Radeva et al.	Chang et al.	This Experiment BP training	This Experiment $GLP_{\tau}S$ training
Test Rate	46% –58%	46% –98%	93.3%	71% –90%	66% –95%

Table 7. Neural Network testing: comparison of our system with other intelligent visual systems employed for cork stoppers classification.

6. Conclusions

Here has been presented an overview of our recent research findings. Initially, a novel Global Optimisation technique, called *LPτO*, has been investigated and proposed. The method is based on *LPτ* Low-discrepancy Sequences and novel heuristic rules for guiding the search. Subsequently, *LPτO* has been hybridized with Nelder-Mead local search, showing very good results for low-dimensional problems. Nevertheless, with the increase of problems dimensionality, method's computational load increases considerably. To tackle this problem, a hybrid Global Optimisation method, called *GLPτS*, that combines Genetic Algorithms, *LPτO* method and Nelder-Mead simplex search, has been studied, discussed and proposed. When compared with Genetic Algorithms, Evolutionary Programming, and Differential Evolution, *GLPτS* has demonstrated strongly competitive results in terms of both number of function evaluations and success rate. Subsequently, *GLPτS* has been applied for supervised NN training and tested on a number of benchmark problems. Based on the reported and discussed findings, it can be concluded that the investigated and proposed *GLPτS* technique is very competitive and demonstrates reliable performance when compared with similar approaches from other authors.

Finally, an Intelligent Computer Vision System has been designed and investigated. It has been applied for a real-world problem of automated recognition and classification of industrial products (in our case study – cork tiles). The classifier, employing supervised

Neural Networks trained with *GLPτS*, has demonstrated reliable generalization abilities. The obtained and reported results have shown strongly competitive nature when compared with results from BP and other authors investigating similar systems.

7. References

Alba, E. & Chicano, J.F. (2004). Training neural networks with GA hybrid algorithms. *Lecture Notes in Computer Science*, Vol. 3102, pp. 852-863.

Ali, M.; Khompatraporn, Ch. & Zabinsky, Z. (2005). A numerical evaluation of several stochastic algorithms on selected continuous global optimisation test problems. *Journal of Global Optimisation*, Vol. 31, pp. 635-672.

Bishop C. (1995). *Neural networks for pattern recognition*, Clarendon Press, Oxford.

Bratley P. & Fox B., (1988). ALGORITHM 659 Implementing Sobol's quasirandom sequence generator, *ACM Transactions on Mathematical Software*, Vol. 14, pp. 88-100.

Chelouah, R. & Siarry, P. (2003). Genetic and Nelder-Mead algorithms hybridised for a more accurate global optimisation of continuous multidimensional functions. *European Journal of Operational Research*, Vol. 148, pp. 335-348.

Chelouah, R. & Siarry, P. (2005). A hybrid method combining continuous tabu search and Nelder-Mead simplex algorithms for the global optimisation of multiminima functions. *European Journal of Operational Research*, Vol. 161, pp. 636-654.

Costa, A. & Pereira, H. (2006). Decision rules for computer-vision quality classification of wine natural cork stoppers. *American Journal of Enology and Viticulture*, Vol. 57, pp. 210-219.

Chang, J.; Han, G.; Valverde, J.M.; Grisworld, N.C. et al. (1997). Cork quality classification system using a unified image processing and fuzzy-neural network methodology. *IEEE Trans. Neural Networks*, Vol. 8, pp. 964-974.

Davies, E.R. (2005). *Machine Vision: theory, algorithms, practicalities*. Morgan Kaufmann.

De Jong (2006). *Evolutionary computation*, MIT Press, Cambridge.

Georgieva, A. & Jordanov, I. (2006). Supervised neural network training with hybrid global optimisation technique. *Proc. IEEE World Congress on Computational Intelligence*, Canada, pp. 6433-6440.

Georgieva, A. & Jordanov, I. (2008a). Global optimisation based on novel heuristics, low-discrepancy sequences and genetic algorithms. *European Journal of Operational Research* (to appear).

Georgieva, A. & Jordanov, I. (2008b). Intelligent visual recognition and classification of cork tiles with neural networks. *IEEE Transactions on Neural* Networks (to appear).

Georgieva, A. & Jordanov, I. (2008c). A hybrid meta-heuristic for global optimisation using low-discrepancy sequences of points. *Computers and Operations Research - special issue on hybrid metaheuristics* (to appear).

Georgieva, A.; Jordanov, I. & Rafik, T. (2007). Neural networks applied for cork tiles image classification. *Proceedings of IEEE Symposium on Computational Intelligence in Image and Signal Processing*, pp. 232-239, USA.

Haralick, R.M.; Shanmugam, K. & Dinstein, I. (1973) Textural features for image classification", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 3, pp. 610-21.

Haykin, S. (1999). *Neural networks - a comprehensive foundation.* Prentice-Hall, Inc.

Heaton, J. (2005). *Introduction to neural networks*, Heaton Research Inc.

Hedar, A.R. & Fukushima, M. (2003). Minimizing multimodal functions by simplex coding genetic algorithm. *Optimisation Methods and Software*, Vol. 18, pp. 265-282.

Jordanov, I. & Georgieva, A. (2007). Neural network learning with global heuristic search. *IEEE Transactions on Neural Networks*, Vol. 18, No. 3, pp. 937-942.

Kucherenko, S. & Sytsko, Y. (2005). Application of deterministic low-discrepancy sequences in global optimisation. *Computational Optimisation and Applications*, Vol. 30, pp. 297-318.

Laws, K.I. (1980). Rapid texture identification. *SPIE - Image Proc. for Missile Guidance*, Vol. 238, pp. 376-380.

Lee, D.W.; Choi, H.J. & Lee, J. (2004). A regularized line search tunnelling for efficient neural network learning. *Lecture Notes in Computer Science*, Vol. 3173, pp. 239-243.

Leung, Y.W. & Wang, Y. (2001). An orthogonal genetic algorithm with quantization for global numerical optimisation. *IEEE Transactions on Evolutionary Computation*, Vol. 5, pp. 41-53.

Leung, C.S; Tsoi, A., & Chan, L.W. (2001). Two regularizers for recursive least squared algorithms in feedforward multilayered neural networks. *IEEE Transactions Neural Networks*, Vol. 12, pp. 1314-1332.

Liberti, L. & Kucherenko, S. (2005). Comparison of deterministic and stochastic approaches to global optimisation. *International Trans. in Operational Research*, Vol. 12, pp. 263-285.

Ludemir, T.B.; Yamazaki, A. & Zanchettin, C. (2006). An optimisation methodology for neural network weights and architectures. *IEEE Transactions on Neural Networks*, Vol. 17, No. 6, pp. 1452-1459.

Mitchell, M. (2001). *An introduction to genetic algorithms*, MIT Press: Massachusetts.

Nelder, J. & Mead, R. (1965). A simplex method for function minimization. *The Computer Journal*, Vol. 7, pp. 308-313.

Niederreiter, H. (1992) *Random number generation and Quasi-Mounte Carlo methods,* Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania.

Plaginakos, V.P.; Magoulas, G.D. & Vrahatis, M.N. (2001). Supervised training using global search methods. *Advances in Convex Analysis and Global Optimisation, Kluwer Acad. Publishers, Dordrecht,* pp. 421-432.

Quinlan, J.R. (1993). Combining instance-based and model-based learning, *Proc. ML'93 (ed. P.E. Utgoff), San Mateo: Morgan Kaufmann*, pp. 236-243.

Radeva, P.; Bressan, M.; Tovar, A. & Vitrià, J. (2002). Bayesian classification for inspection of industrial products. *Proc. of the 5th Catalonian conference on artificial intelligence* – M.T. Escrig et al. (Eds.), pp. 399-407.

Rocha, M.; Cortez, P. & Neves J. (2003). Evolutionary neural networks learning. *Lecture Notes in Computer Science*, Vol. 2902, pp. 24-28.

Sarker, R.; Mohammadian, M. & Yao, X. (2002). Evolutionary Optimization, Kluwer Academic Publishers: Boston.

Sexton, R.S.; Alidaee, B.; Dorsey, R.E. & Johnson, J.D. (1998). Global optimisation for artificial neural networks: a tabu search application. *European Journal of Operational Research*, Vol. 106, pp. 570-584.

Smagt, P. (1994). Minimization methods for training feed-forward neural networks. *Neural Networks*, Vol. 7, pp. 1-11.

Sobol', I.M. (1979). On the systematic search in a hypercube. *SIAM Journal of Numerical Analysis*, Vol. 16, No. 5, pp. 790-792.

Theodoridis, S. & Koutroumbas, K. (2006). *Pattern Recognition*, Academic Press, 3rd edition.

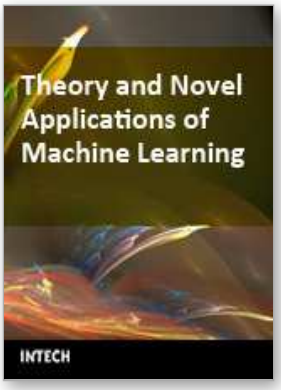
Umbaugh, S. (2005). *Computer imaging: digital image analysis and processing*, The CRC Press.

Wang, X.; Tang, Z.; Tamura, H.; Ishii, M.; & Sun, W. (2004). An improved backpropagation algorithm to avoid the local minima problem", *Neurocomputing*, vol. 56, pp. 455-460.

WWF/MEDPO, (2006). A WWF Report, Cork Screwed? Environmental and economic impacts of the cork stoppers market. .

Yao, X. (1999). Evolving artificial neural networks. *Proceedings of IEEE*, Vol. 87, No. 9, pp. 1423-1447.

Yao, X.; Liu, Y. & Lin, G. (1999). Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 2, pp. 82-102.



Theory and Novel Applications of Machine Learning

Edited by Meng Joo Er and Yi Zhou

ISBN 978-953-7619-55-4
Hard cover, 376 pages
Publisher InTech
Published online 01, January, 2009
Published in print edition January, 2009

Even since computers were invented, many researchers have been trying to understand how human beings learn and many interesting paradigms and approaches towards emulating human learning abilities have been proposed. The ability of learning is one of the central features of human intelligence, which makes it an important ingredient in both traditional Artificial Intelligence (AI) and emerging Cognitive Science. Machine Learning (ML) draws upon ideas from a diverse set of disciplines, including AI, Probability and Statistics, Computational Complexity, Information Theory, Psychology and Neurobiology, Control Theory and Philosophy. ML involves broad topics including Fuzzy Logic, Neural Networks (NNs), Evolutionary Algorithms (EAs), Probability and Statistics, Decision Trees, etc. Real-world applications of ML are widespread such as Pattern Recognition, Data Mining, Gaming, Bio-science, Telecommunications, Control and Robotics applications. This books reports the latest developments and futuristic trends in ML.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Antoniya Georgieva and Ivan Jordanov (2009). Supervised Learning with Hybrid Global Optimisation Methods. Case Study: Automated Recognition and Classification of Cork Tiles, Theory and Novel Applications of Machine Learning, Meng Joo Er and Yi Zhou (Ed.), ISBN: 978-953-7619-55-4, InTech, Available from: http://www.intechopen.com/books/theory_and_novel_applications_of_machine_learning/supervised_learning_with_hybrid_global_optimisation_methods__case_study__automated_recognition_and_c

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821