Training a Carbon-Nanotube / Liquid Crystal Data Classifier Using Evolutionary Algorithms

E. Vissol-Gaudin, A. Kotsialos, M. K. Massey, D. A. Zeze,
C. Pearson, C. Groves and M. C. Petty
E-mail:{eleonore.vissol-gaudin, apostolos.kotsialos, m.k.massey, d.a.zeze,
christopher.pearson, chris.groves, m.c.petty}@durham.ac.uk

School of Engineering and Computing Sciences, Durham University, Durham,
United Kingdom

Abstract. Evolution-in-Materio uses evolutionary algorithms (EA) to exploit the physical properties of unconfigured, physically rich materials, in effect transforming them into information processors. The potential of this technique for machine learning problems is explored here. Results are obtained from a mixture of single walled carbon nanotubes and liquid crystals (SWCNT/LC). The complex nature of the voltage/current relationship of this material presents a potential for adaptation. Here, it is used as a computational medium evolved by two derivative-free, population-based stochastic search algorithms, particle swarm optimisation and differential evolution. The computational problem considered is data classification. A custom made electronic motherboard for interacting with the material has been developed, which allows the application of control signals on the material body. Starting with a simple binary classification problem of separable data, the material is trained with an error minimisation objective for both algorithms. Subsequently, the solution, defined as the combination of the material itself and optimal inputs, is verified and results are reported. The evolution process based on EAs has the capacity to evolve the material to a state where data classification can be performed. PSO outperforms EA in terms of results' reproducibility due to the smoother, as opposed to more noisy, inputs applied on the material.

1 Introduction

Unconventional computing aims at investigating methods for designing systems able to perform a computation in different ways than the current paradigm. One such direction of research is evolution in materio (EIM) [10], which is concerned with computing performed directly by the materials. EIM focuses on the underlying properties of the materials aiming at exploring and exploiting them in such a way so that they are brought to a computation inducing state. Contrary to traditional computing with MOSFET technology, where everything is designed, produced and programmed very carefully, EIM uses a bottom up approach where computation is performed by the material without having explicit knowledge of its internal properties.

The idea of EIM can be found in early work of G. Pask [17] concerned with growing an electrochemical ear. In more recent work [22], observations were made when evolutionary algorithms were used for designing electrical circuits on FPGAs. The resulting circuit topologies were influenced by the material of the board used. Because of feedback provided by the iterative nature of stochastic optimisation interacting with the material, identified solutions were based on the specific FPGA's properties that were unaccounted for during the board's design. EIM replaced the FPGAs with material systems favouring exploitation of all physical properties by a search algorithm [11].

EIM has a broad scope, which can be delineated along four dimensions: (a) the type of material used, (b) the physical property manipulated for obtaining a computation, (c) the computation problem addressed and (d) the evolutionary algorithm used for solving the corresponding training problem.

Different materials have been used, including biological ones like slime moulds [6], bacterial consortia [1] and cells (neurons) [19]. In [20] it is argued that inorganic materials make a better medium for unconventional computing exploration. Nano-particles were used in [2] for developing a reconfigurable Boolean logic network. In [5] and [4] liquid crystals (LC) panels were used for evolving logic gates, a tone discriminator and a robot controller. Single walled carbon nanotubes (SWCNT) based materials have shown the potential to solve variety of computational problems [7], [9], [23], [12], [13], [14] and [15].

Candidate computational problems include Boolean function calculation, finding a minimum, evolving a controller, obtaining a tone discriminator, developing a neuron and data clustering problems. A more comprehensive review of potential problems can be found in [16]. Here, a simple binary data classification problem is considered.

Because of the complexity of the material EIM generally employs population based derivative free stochastic methods. Here a Particle Swarm Optimisation (PSO) algorithm as described in [8] is used.

2 Hardware Architecture and the SWCNT/LC Material

Figure 1(a) illustrates EIM's concept. An optimisation algorithm selects a set of incident signals applied on to the material (configuration voltages in our case) changing in effect its physical properties. During training, the state the material is brought into by the application of the configuration voltages is tested against a number of known input/output pairs of a correct computation. The material's response is recorded for each of those test inputs and a global error function is evaluated. Using the error function as part of a fitness function allows a swarm intelligence algorithm to explore the search space.

In our implementation, the signals sent are constant voltage charges applied by an *mbed* micro-controller and the outputs are direct current measurements. The voltages are sent to the SWCNT/LC compound via the motherboard through DACs. These are connected to a glass slide where the electrodes are etched, Figure 1. There are sixteen connections on the micro-electrode, but only

twelve of them are used due to hardware constraints. Ten connections are used for sending inputs and another two for collecting the output measurements used for transforming the material's response to a computation.

The nanotubes are dispersed in liquid crystals at varying concentrations. The purpose of the LC matrix is to provide a fluid medium in which the SWCNT can move in response to the applied electric field enabling the nanotubes to form reconfigurable and variable complex electrical networks. This adds an extra dimension to the problem compared to previous experiments, where SWCNT were mixed with a solid polymer [9], [7] and the resulting material system was in solid state, as opposed to the liquid state of the material used here.

The SWCNT/LC blend was drop-deposited within a nylon washer (5 mm internal diameter). The washer was glued to a glass microscope slide upon which an array of gold electrodes had previously been deposited using etch-back photolithography. The electrode array contacts are $50\mu m$ with $100\mu m$ pitch, Figure 1(b).

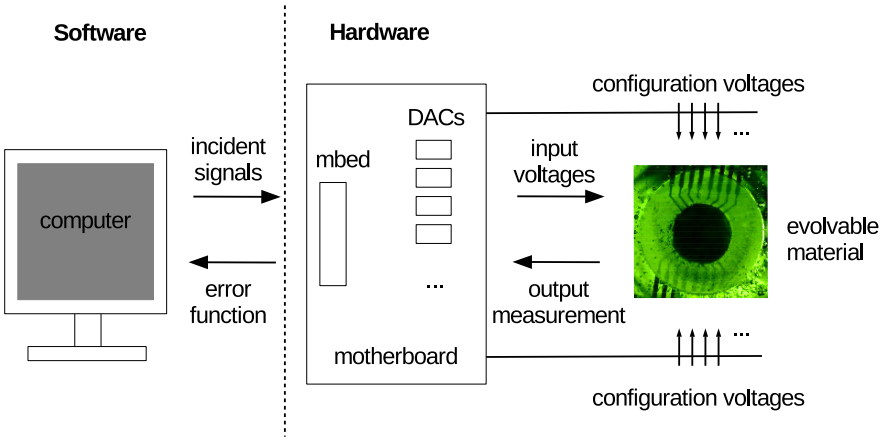
3 The Classification Problem

The computing problem considered here is that of data classification. Two different two-dimensional (2D) datasets are used for two problems of the same nature. A typical training and verification approach is followed for assessing the material's capability to act as a classifier. Figure 2 depicts the training datasets for the two problems. In both cases, two classes are formed, each covering a square area. In the first case the data are highly separable and don't overlap, resulting to the separable classes (SC) problem . In the second case, there is some small overlap where a pair of data can belong to any of the two, resulting to the merged classes (MC) problem. Training aims at evolving the material so that it is brought into a state such that when randomly selected input pairs are given as input, it can infer the class they belong to. The size of the training dataset for the SC and MC problems $K_t = 800$ pairs and the verification datasets' size is $K_v = 4,000$ pairs.

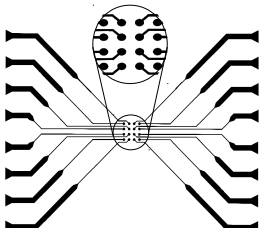
SC and MC are simple binary classification problems and a comparison scheme can result to the correct classification. However, the EIM approach taken here is not equipped intrinsically with such a capability. The material is trained by forcing it to change its shape and adapt its electrical properties so that an incident signal, in the form (V_1, V_2), results to an output that can be interpreted as a classification of that input. There is no explicit design of memory storage or bit comparison or a mechanism for numerical operations. It is just the material's shape and form that is evolved towards a state that produces the desired outcome.

4 Training Problem Formulation

The material training is formulated as an optimisation problem tailored for the evolvable material board. The classification task is about determining the class



(a)

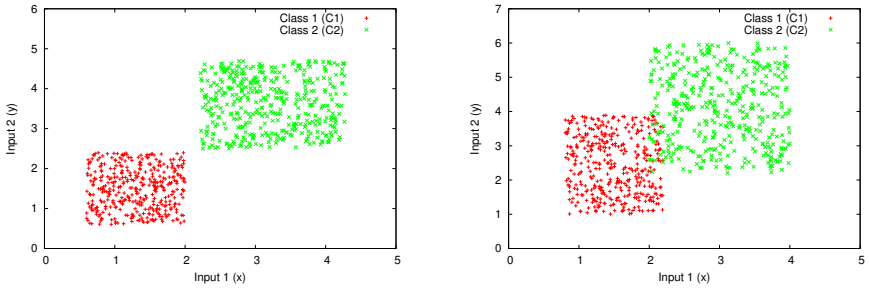


(b)

Fig. 1: (a) EIM hardware system structure. (b) Electrode array.

a pair of data $\mathbf{V}^{in} = (V_1^{in}, V_2^{in})$ belongs to. Hence, two of the ten available electrodes are reserved as data input connections. The inputs come in the form of voltage pulses of amplitude V_1^{in} and V_2^{in} (Volts). The remaining eight connections are used for applying configuration voltages to the material. They are realised as voltage pulses of amplitude $V_j \in [V_{\min}, V_{\max}]$, $j = 1, \ldots, 8$ (Volts). In order to evaluate a potential set of configuration voltages V_j, first the electrodes where each of the V_j is applied is decided. These voltages are then applied and the corresponding electrodes are kept charged while K_t known pairs of training inputs are send to the two electrodes selected for receiving the data inputs.

Two output connections are used for measuring the material response when it is constantly charged with the configuration voltages V_j and a pair of data $\mathbf{V}^{in}$ is send as input. Although the output locations are fixed because of hardware constraints, the connections where the inputs are going to be applied are variable and are part of the optimisation problem's vector of decision variables $\mathbf{x}$. The



(a) (b)

Fig. 2: Training datasets. (a) SC problem. (b) MC problem.

optimisation problem's vector of decision variables is defined as

$$\mathbf{x} = [V_1 \dots V_8 \ R \ p]^T . \tag{1}$$

where R is a scaling factor and $p \in \mathbb{N}$ an index running on the set of possible electrode assignments. An electrode assignment is a mapping from the set of data and configuration voltage inputs to the set of the ten electrodes. It is for a specific electrode assignment p and set of configuration voltages V_j, that the material's response to an input $\mathbf{V}^{in}$ is recorded. The response is a pair of measurements $\mathbf{I} = (I_1, I_2)$ (A) of the direct current at the two output locations, which are the basis of a comparison scheme using R for deciding the class $\mathbf{V}^{in}$ belongs to.

Let $\mathbf{I}^{(k)}$ denote the pair of direct current measurements taken when input data $\mathbf{V}^{in}(k)$ from class C_i, $i = 1$ or $i = 2$, are applied *while* the material is subjected to configuration voltages $V_j^{(k)}$, applied according to electrode assignment number $p^{(k)}$ using scaling factor $R^{(k)}$. Also, let $C(\mathbf{V}^{in}(k))$ denote $\mathbf{V}^{in}(k)$'s real class and $C_M(\mathbf{V}^{in}(k), \mathbf{x})$ the material's assessment of it calculated according to the following rule.

$$C_M(\mathbf{V}^{in}(k), \mathbf{x}) = \begin{cases} C_1 \text{ if } I_1(k) > RI_2(k) \\ C_2 \text{ if } I_1(k) \leq RI_2(k). \end{cases} \tag{2}$$

For every training pair of data $\mathbf{V}^{in}(k)$, $k = 1, \dots, K_t$ the error from translating the material response according to rule (2) is

$$\epsilon_{\mathbf{x}}(k) = \begin{cases} 0 \text{ if } & C_M(\mathbf{V}^{in}(k), \mathbf{x}) = C(\mathbf{V}^{in}(k)) \\ 1 \text{ otherwise.} \end{cases} \tag{3}$$

The mean total error is given by

$$\Phi_e(\mathbf{x}) = \frac{1}{K_t} \sum_{k=1}^{K_t} \epsilon_{\mathbf{x}}(k). \tag{4}$$

Two penalty terms are added to (4), H and U. $H(\mathbf{x})$ penalises solutions with high configuration voltages and is given by

$$H(\mathbf{x}) = \frac{\sum_{j=1}^{8} V_j^2}{8V_{max}^2}. \tag{5}$$

The rationale behind this penalisation is that incremental and generally low levels of configuration voltages are preferable. Solutions where high voltages are applied can destroy possible material structures favourable to the problem formed gradually during evolution. On the other hand, solutions that render the material unresponsive need to be avoided. A measure of such unresponsiveness is calculated at the end of each search iteration ι, where a sample equal to the population size S of error function evaluations is available. Let $\sigma_{o,\iota}^2$ denote the variance of $\Phi(\mathbf{x})$ and $\sigma_{V,\iota}^2$ the variance of $\sum_{j=1}^{8} V_j^2$ at iteration ι. A value of $\sigma_{o,\iota}^2$ close to zero indicates a non-responsive material and the penalty term takes the form

$$U_\iota = \left(1 - \frac{\sigma_{o,\iota}^2}{\sigma_{V,\iota}^2}\right)^2. \tag{6}$$

Hence, the total objective function $\Phi_s(\mathbf{x})$ for an arbitrary individual s at iteration ι is given by

$$\Phi_s(\mathbf{x}) = \Phi_e(\mathbf{x}) + H(\mathbf{x}) + U_\iota. \tag{7}$$

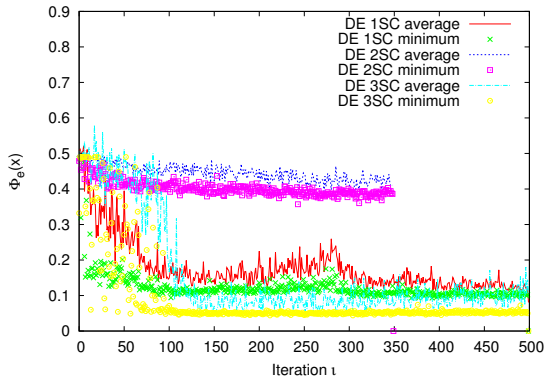
U_ι aims at leading the optimisation away from material states where the same response is given for different inputs.

The optimisation to be solved is that of minimising (7) for a population of size S, subject to voltage bound constraints $V_j \in [V_{\min}, V_{\max}]$, $R > 0$, electrode assignment p and classification rule (2). $V_{\min} = 0$ Volts and for the SC problem $V_{\max} = 4$ Volts whereas for the MC $V_{\max} = 7$ Volts.

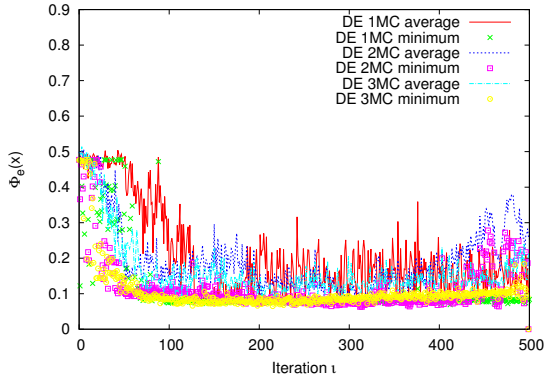
Two different evolutionary optimisation algorithms are used for solving this problem, differential evolution (DE) [21] and particle swarm optimisation (PSO) [3]. A constricted version of PSO with parameters taken from [8] is implemented. The DE algorithm implementation uses the parameters suggested in [18]. A population size of $S = 8$ is used for DE and $S = 10$ for PSO.

5 Results and Discussion

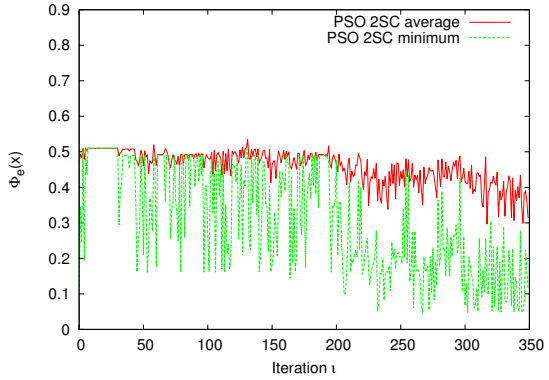
Training is performed by the DE and PSO algorithms solving the optimisation problem described in section 4 using the K_t pairs of data for problems SC and MC. The maximum number of optimisation iterations is used as termination criterion or lack of significant reduction for a number of iterations. The convergence profiles for the DE algorithm applied to the SC and MC data are shown in Figure 3(a) and (b), respectively.



(a)



(b)



(c)

Fig. 3: Convergence profile for different runs. (a) All runs of DE for the SC problem. (b) All runs of DE for the MC problem. (c) Second run of PSO for the SC problem.

For the three runs, the training error averaged over the values achieved by the eight individuals of the population at iteration ι is shown along with the best value achieved in that iteration (not the best up to iteration ι). It can be seen that the DE algorithm trains the material and the average error follows the trend of the best result per iteration. This is not the case of the PSO algorithm, shown in Figure 3(c) (a single run is presented for the sake of clarity). The average error per iteration is much higher than the best achieved, although a positive correlation between the two is evident. The material is also evolved, but the PSO tends to explore more the search space. In both cases, during training the material morphology changes in order to provide a response that leads to a correct classification. However, the emphasis on exploitation displayed by DE does not provide the best verification result.

Once training is terminated, verification is performed on the trained material by applying back the optimal configuration voltages and sending as input K_v verification data pairs different from the K_t pairs used in training. The same verification experiment is repeated ten times and each time the mean error (4) is calculated and recorded.

Since the optimum may have been achieved several iterations before the algorithm's termination, the optimal solution will not have the same effect because the material would have undergone a number of non-reversible changes by that time. Hence, in order to achieve good verification results, structures inside the material need to be built that favour an error minimising response. It is the gradual evolution performed on the material that builds these structures of SWCNT conductive networks.

Table 1 provides the training Φ_e^* error, the best verification error $\Phi_{e,v}^*$ from the ten experiments conducted using the optimal solution, the worst verification error $\Phi_{e,v}^w$, and the mean verification error $\overline{\Phi_{e,v}}$ for three runs of the PSO and the DE for the SC problem. It can be seen that the PSO algorithm outperforms the DE particularly with respect to verification errors. In terms of training error, the second experiment of DE resulted to a material with over 30% error, which is too large and was terminated early. On the contrary, all PSO experiments resulted to a Φ_e^* less than 10%.

The solution degradation on the verification data is much lower for two of the three PSO runs, where practically the results are identical. Still, though the difference on the error value does not grow above 2.5% in the worst case. This indicates that the material has a consistent behaviour by the end of the search algorithms and the internal structures built inside it are not completely destroyed by the evolution process.

Table 2 provides the training and verification errors for the MC problem. Because this is a more difficult problem due to the small overlap of the data a bias of about 3% error is created. This is consistent with the training error, since the best PSO and DE values of Φ_e are larger by 3.6% and 1.9%, respectively. DE achieves better training error but the verification error is much larger. On the contrary, the PSO solutions generalise better and the verification errors are

Table 1: Problem SC training and verification errors for experiments using PSO and DE.

Experiment	$\Phi_e^*(\%)$	$\Phi_{e,v}^*(\%)$	$\Phi_{e,v}^w(\%)$	$\overline{\Phi_{e,v}}$ (%)
PSO 1SC	5.5	7.125	7.875	7.46751
PSO 2SC	9.6	9.85	10.1	9.9925
PSO 3SC	3.8	5.975	10.05	8.452
DE 1SC	9.2	11.625	12.9	12.1575
DE 2SC	35.7	36.925	39.8	38.8427
DE 3SC	4.4	6.4	9.775	6.9825

Table 2: Problem MC training and verification errors for experiments using PSO and DE.

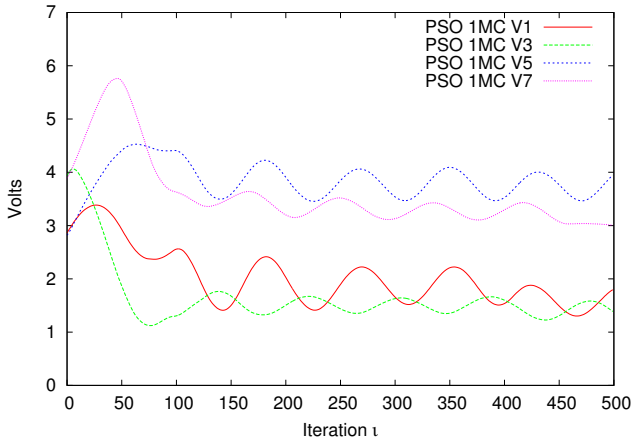
Experiment	$\Phi_e^*(\%)$	$\Phi_{e,v}^*(\%)$	$\Phi_{e,v}^w(\%)$	$\overline{\Phi_{e,v}}$ (%)
PSO 1MC	7.4	8.025	10.275	8.995
PSO 2MC	10.6	10.15	10.825	10.5775
PSO 3MC	7.5	7.4	8.075	7.6275
DE 1MC	6.9	8.45	9.075	8.7725
DE 2MC	6.3	27.825	31.25	29.7625
DE 3MC	6.5	12.075	15.525	13.2675

very similar to the training (in some cases, even marginally better). Hence, the PSO algorithm yields better solutions.

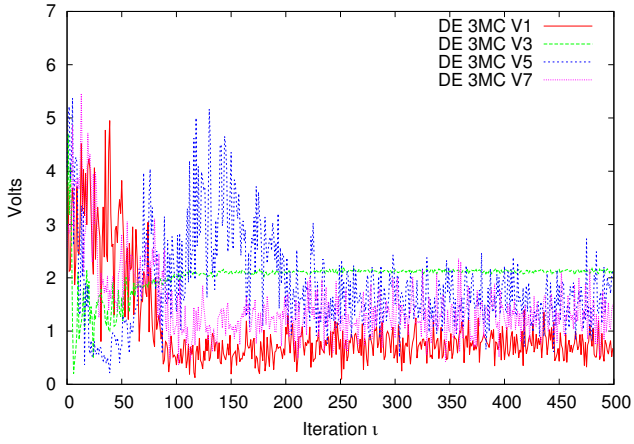
In the absence of analytical models of the material's dynamics, it is difficult to provide a rigorous explanation as to why PSO outperforms DE in verification. A distinctive difference between the two is the form of the configuration voltages' trajectories over iterations as they are exploring the search space. Figure 4 depicts the trajectories of a sample of the configuration voltages *averaged per iteration* for the PSO and DE algorithms. It can be seen that the search performed by DE is more noisy. On the other hand, PSO's exploration of the search space is much smoother. DE sends signals to the material that are noisy even when it aims to exploit a minimum. Hence, a conjecture about PSO algorithm's better performance is that the smoother trajectories of configuration inputs build more stable structures inside the material reinforcing at the same time responses minimising the classification error. The noisy configuration voltages applied by DE make the formation of such stable structures more difficult. This conjecture needs to be supported by more experiments and evidence, such as image analysis of the material before and after training.

6 Conclusions

This paper has presented the results of an investigation on evolution in materio for a new type of material, a mixture of single walled carbon nanotubes and liquid crystals. It is in liquid form and the nanotubes inside it form conductive



(a)



(b)

Fig. 4: Average configuration voltages per iteration for (a) PSO and (b) DE.

networks. Under the influence of different levels of voltage applied at various locations of its body, different networks are formed. The material is placed on a glass slide with electrodes etched on it and a custom made board based on the *mbed* micro-controller is used for evolving it as a data classifier.

Two simple classification problems are considered in an effort to evolve the material towards a state where measurements of electrical current can be interpreted following a pre-specified rule.

The training problem is formulated as an optimisation problem and results of both training and verification are reported. Two different algorithms have been used, PSO and DE. PSO outperforms the DE as it converges to better quality

solutions, which generalise more than those delivered by DE. The search pattern followed by PSO is very different than of DE. DE performs more detailed exploitation of a solution and generally sends noisy signals to the material. PSO has a stronger exploration element and sends much smoother input signals resulting to superior performance in the verification phase. The result is the evolution of an analogue classifier out of an initially unformed liquid state material.

This is a new area of research and a lot of issues need to be addressed. A more detailed investigation needs to be performed on the optimisation algorithms used and the impact of their search pattern on the solutions' quality. More recent variants of PSO, DE or other evolution-inspired algorithms need to be implemented. The impact of the concentration of SWCNT and LC in the mix needs to be evaluated. Finally, more complicated problems need to be considered and it would be very interesting to observe the material structure patterns formed for this purpose.

Acknowledgment

The research is supported by European's Community Seventh Framework programme (FP7/2007-2013) under the grant agreement No. 317662 (NAno Scale Engineering for Novel Computation using Evolution - NASCENCE (http://www.nascence.eu)).

References

[1] Amos, M., Hodgson, D., Gibbons, A.: Bacterial self-organisation and computation. arXiv preprint q-bio/0512017 (2005)

[2] Bose, S., Lawrence, C., Liu, Z., Makarenko, K., van Damme, R., Broersma, H., van der Wiel, W.: Evolution of a designless nanoparticle network into reconfigurable boolean logic. Nature Nanotechnology (2015)

[3] Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science. vol. 1, pp. 39–43. New York, NY (1995)

[4] Harding, S., Miller, J.: Evolution in materio: A tone discriminator in liquid crystal. In: Evolutionary Computation, 2004. CEC2004. Congress on. vol. 2, pp. 1800–1807. IEEE (2004)

[5] Harding, S., Miller, J.: Evolution in materio: Evolving logic gates in liquid crystal. Proc. Eur. Conf. Artif. Life (ECAL 2005), Workshop on Unconventional Computing: From cellular automata to wetware pp. 133–149 (2005)

[6] Jones, J., Whiting, J., Adamatzky, A.: Quantitative transformation for implementation of adder circuits in physical systems. Biosystems 134, 16–23 (2015)

[7] Kotsialos, A., Massey, M., Qaiser, F., Zeze, D., Pearson, C., Petty, M.: Logic gate and circuit training on randomly dispersed carbon nanotubes. International Journal of Unconventional Computing 10(5-6), 473–497 (2014)

[8] Laskari, E., Parsopoulos, K., Vrahatis, M.: Particle swarm optimization for integer programming. In: wcci. pp. 1582–1587. IEEE (2002)

[9] Massey, M., Kotsialos, A., Qaiser, F., Zeze, D., Pearson, C., Volpati, D., Bowen, L., Petty, M.: Computing with carbon nanotubes: Optimization of threshold logic gates using disordered nanotube/polymer composites. Journal of Applied Physics 117(13), 134903 (2015)
[10] Meyers, R.: Encyclopedia of complexity and systems science. Springer (2009)
[11] Miller, J., Downing, K.: Evolution in materio: Looking beyond the silicon box. In: Evolvable Hardware, 2002. Proceedings. NASA/DoD Conference on. pp. 167–176. IEEE (2002)
[12] Miller, J., Mohid, M.: Function optimization using cartesian genetic programming. In: Proceedings of the 15th annual conference companion on Genetic and evolutionary computation. pp. 147–148. ACM (2013)
[13] Mohid, M., Miller, J., Harding, S., Tufte, G., Lykkebø, O., Massey, M.K., Petty, M.: Evolution-in-materio: A frequency classifier using materials. In: Evolvable Systems (ICES), 2014 IEEE International Conference on. pp. 46–53. IEEE (2014)
[14] Mohid, M., Miller, J., Harding, S., Tufte, G., Lykkebø, O., Massey, M., Petty, M.: Evolution-in-materio: Solving bin packing problems using materials. In: Evolvable Systems (ICES), 2014 IEEE International Conference on. pp. 38–45. IEEE (2014)
[15] Mohid, M., Miller, J., Harding, S., Tufte, G., Lykkebø, O., Massey, M., Petty, M.: Evolution-in-materio: Solving machine learning classification problems using materials. In: Parallel Problem Solving from Nature–PPSN XIII, pp. 721–730. Springer International Publishing (2014)
[16] NASCENCE project (ICT 317662): Report on suitable computational tasks of various difficulties (2013), deliverable D4.2
[17] Pask, G.: Physical analogues to the growth of a concept. In: Mechanization of Thought Processes, Symposium. vol. 10, pp. 765–794 (1958)
[18] Pedersen, M.: Good parameters for differential evolution. Tech. rep., Technical report, Hvass Computer Science Laboratories (2010)
[19] Prasad, S., Yang, M., Zhang, X., Ozkan, C., Ozkan, M.: Electric field assisted patterning of neuronal networks for the study of brain functions. Biomedical Microdevices 5(2), 125–137 (2003)
[20] Stepney, S.: The neglected pillar of material computation. Physica D: Nonlinear Phenomena 237(9), 1157–1164 (2008)
[21] Storn, R., Price, K.: Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization 11(4), 341–359 (1997)
[22] Thompson, A.: An evolved circuit, intrinsic in silicon, entwined with physics. In: Evolvable systems: from biology to hardware, pp. 390–405. Springer Berlin Heidelberg (1996)
[23] Volpati, D., Massey, M., Johnson, D., Kotsialos, A., Qaiser, F., Pearson, C., Coleman, K., Tiburzi, G., Zeze, D., Petty, M.: Exploring the alignment of carbon nanotubes dispersed in a liquid crystal matrix using coplanar electrodes. Journal of Applied Physics 117(12), 125303 (2015)