Travelling Salesman Problem Solved *'in materio'* by Evolved Carbon Nanotube Device

Kester Dean Clegg[1], Julian Francis Miller[1], Kieran Massey[2], and Mike Petty[2]

[1] The University of York, UK
{kester.clegg,julian.miller}@york.ac.uk
[2] Durham University, UK
{m.k.massey,m.c.petty}@durham.ac.uk

Abstract. We report for the first time on finding shortest path solutions for the travelling salesman problem (TSP) using hybrid "in materio" computation: a technique that uses search algorithms to configure materials for computation. A single-walled carbon nanotube (SWCNT) / polymer composite material deposited on a micro-electrode array is configured using static voltages so that voltage output readings determine the path order in which to visit cities in a TSP. Our initial results suggest that the hybrid computation with the SWCNT material is able to solve small instances of the TSP as efficiently as a comparable evolutionary search algorithm performing the same computation in software. Interestingly the results indicate that the hybrid system's search performance on TSPs scales linearly rather than exponentially on these smaller instances. This exploratory work represents the first step towards building SWCNT-based electrode arrays in parallel so that they can solve much larger problems.

1 Introduction

As material variability starts to limit the design of modern silicon-based processors [1,2], the NASCENCE project [3] has started investigating how nano-scale materials and devices that could be configured for computation [4]. While engineering for computation at such small scales remains extremely difficult, it is possible to use a method of automatically configuring materials so that they can perform a dedicated task [5]. In this paper, we illustrate a method to achieve this using a single-walled carbon nanotube (SWCNT) material configured by applying static voltages to specific locations on a micro-electrode array. The remaining electrodes act as inputs to the computation, in this case a small instance of the Travelling Salesman Problem (TSP). The hybrid system successfully finds the shortest path in many cases and its search performance is on a par to a comparable (optimised) evolutionary algorithm implemented in software. We believe this work represents the first time a benchmark search problem has been solved by a machine configured material and could pave the way toward designing novel computational materials: resulting in small, single purpose devices that are stable, low voltage and yet able to compute answers to challenging tasks.

Previous work in computer controlled configuration of physical systems for computation has had limited success [6,7,8]; results were sometimes unstable [9,10], able to

T. Bartz-Beielstein et al. (Eds.): PPSN XIII 2014, LNCS 8672, pp. 692–701, 2014.

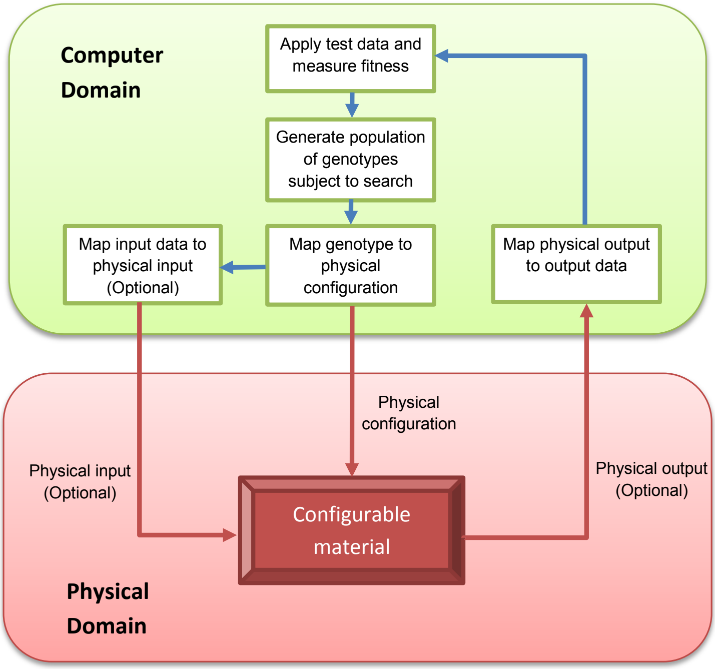
Computer Domain
Apply test data and measure fitness
Generate population of genotypes subject to search
Map input data to physical input (Optional)
Map genotype to physical configuration
Map physical output to output data
Physical configuration
Physical input (Optional)
Physical output (Optional)
Configurable material
Physical Domain

Fig. 1. Overview of hybrid computation with 'in materio' configuration performed by evolutionary search. The computer configures the application of physical signals to a material and tests the output. A genotype of configuration instructions is subject to evolutionary mutation. Physical output from the materials acts as computational input to the selection process. The configuration loop stops when the computational inputs have found a solution or no more improvement is required or possible.

carry out only low level processing tasks [6,11] or performed no better than exhaustive search [12,13]. Most work done on configuring materials to carry out computation has used search-based algorithms to exploit physical characteristics within the material without much concern as to how those characteristics could be formally understood or engineered [14,15]. An overview of the hybrid approach to materials-based computation is shown in Figure 1.

2 Hardware and Material Description

The NASCENCE consortium [3] is investigating candidate materials and techniques that could be employed to configure materials for computation. The project is looking at SWCNT held in fixed, gel and liquid mediums (such as liquid crystal). In this paper we report results using a fixed material system (supplied by Durham University) prepared on gold micro-electrode arrays with contacts arranged in either a 3x4 or a 4x4 grid. The early prototype 3x4 grid array has pads that are 40μm diameter with a pitch of

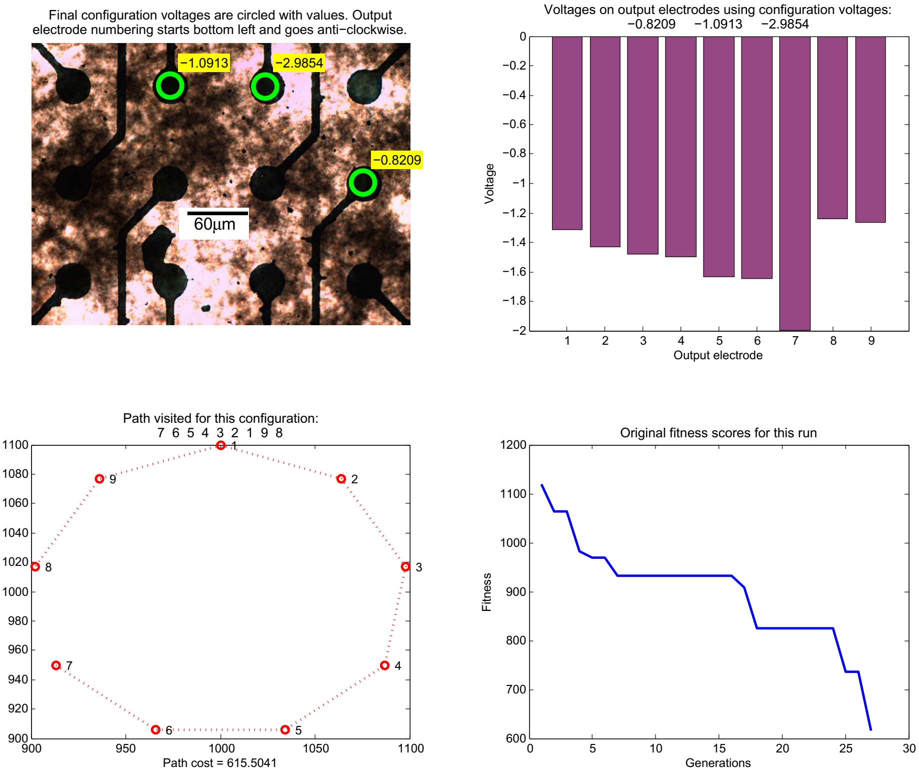

Fig. 2. Top left: shows the CNT dispersal over an earlier prototype 3x4 grid electrodes (x200 mag). Note unevenness of material over electrodes and the mask fault on the third electrode does not appear to affect the evolutionary search when finding the shortest path of the TSP (the evolutionary history is shown by the best performing genotype for each generation, bottom right). In this final configuration, voltages are applied to the circled electrodes and the remaining electrodes provide the floating point values into the TSP. Top right: recorded voltages which when sorted determine the path order to visit cities. Bottom left: Optimum path solution of the TSP.

100μm, leaving an electrode separation of approximately 60μm, while the later 4x4 grid has a spacing of 50μm pads and separation. The material is composed of 0.10% (by weight) of single-walled carbon nanotubes (purchased from Carbon Nanotechnologies Inc. Houston, TX, USA) mixed with poly-methyl methacrylate (PMMA, Aldrich Mw = 93,000) and dissolved in Anisole (VWR, analytical reagent grade). Approximately 20μL of material is dispensed on the electrode array which is then dried at 100°C for 30mins to leave a "thick film". The density of SWCNTs within the PMMA is quite variable, as is the accuracy of the mask to create the electrodes (see top left panel, Figure 2), however these inconsistencies in the substrate do not appear to present a problem to evolutionary search.

The 4x4 electrode array is connected a 16x16 analogue cross point switch which controls connections from the substrate to a data acquisition (DAQ) card, enabling us to place configuration voltages anywhere within the array. The DAQ card first digitally configures the switch connections and then inputs analogue configuration voltages to the material and records the corresponding analogue outputs. The number of configuration voltages deployed depends on the problem being tackled and the availability of spare electrodes on the array. The configuration voltages and electrodes to which they connect are decided by a 1+4 evolutionary algorithm (see method). The range is restricted to ±3V and all connections are one to one (i.e. one configuration voltage can only go to one electrode). We apply the voltages for 1s and take mean values of the final 0.2s to minimize any noise or "settling periods" within the material. The time required to configure the analogue switch and set up channels on the DAQ card means that testing a configuration takes several seconds. While this is relatively "slow" to perform a computation, signals from the SWCNT-PMMA materials have negligible noise levels after the initial 50ms and our sampling times could be substantially reduced.

3 Problem and Method

The TSP was selected as our computational task as the problem belongs to a class of computationally hard problems known as NP-complete, it has been extensively studied [16] and there exist excellent free solvers against which we can check our solutions[1]. In terms of the search, the total number of paths within a given TSP increases factorially with the number of cities to visit, making it an excellent problem to see whether a solver can scale to larger problems.

A circular layout of cities has an easily identifiable shortest path. For symmetric TSPs, the minimum number of shortest paths is equal to double the number of cities, as a path can start at any city and go in either direction along the shortest path. Thus a circular 9 city TSP has 18 shortest paths, while an irregular 9 city layout may have more. A path can be expressed as a permutation, therefore the total number of paths for a given TSP is given by the factorial of the number of cities:

9 city	= 9!	= 362,880
10 city	= 10!	= 3,628,800
11 city	= 11!	= 39,916,800

Our representation maps the path visiting each city to a sorted vector of floating point numbers (the voltage outputs from the substrate, see below). The computational aim is to get the evolutionary algorithm to select the connections and configure the material with voltages so that it outputs a vector that represents the shortest path in the TSP (see bottom left of Figure 2). The sorted vector, or 'smallest position value' representation [17], can be readily applied to many other well-known computational problems such as classification tasks [18].

Currently, when we tackle a 10 city TSP with a 4x4 electrode array, we can in theory use the remaining 6 voltages to configure the SWCNT-PMMA material. In practice we are limited to a maximum of 4 analogue outputs from our current DAQ card and the results presented here reflect this limitation. However, we are having bespoke hardware

[1] http://www.math.uwaterloo.ca/tsp/concorde/

built for the NASCENCE consortium [3] that will allow many more analogue inputs and outputs to connect to the electrode arrays and permit us to look at a broader range of problems.

For example, using our current electrode array and DAQ card to configure the SWCNT-PMMA material on a 12 city TSP, we can use between 2–4 configuration voltages to configure the substrate and use the remaining electrodes of the array to provide the 12 input values to the problem. The input vector of values (1.a below — note just a few values are shown for clarity) is converted to a path by using the original indices after sorting (1.b); the sorted indices then become the order in which to visit the cities (see also bottom left Figure 2). In the following example city 6 would be visited first, then 2, then 4 and so on:

1. a)	1.892	0.321	2.064	0.984	1.664	0.056	2.391	2.907
	1	2	3	4	5	6	7	8
1. b)	0.056	0.321	0.984	1.664	1.892	2.064	2.391	2.907
	6	2	4	5	1	3	7	8

The use of a 16x16 cross point switch permits an evolutionary algorithm to select which electrodes in the array will be used for the configuration voltages and which are to be used as inputs to the TSP. This selection allows the search algorithm to exploit the physical resources available to different electrodes (this can be important, note the variable SWCNT dispersion shown top left of Figure 2). After the path order to visit cities is determined, the path length is calculated and this represents the evolutionary fitness for that configuration.

The evolutionary algorithm uses a 1+4 schema, where the best is kept and 4 copies are mutated for each generation. If a mutant scores an equivalent fitness to the previous best it is selected to form the subsequent generation so that the search continues to 'move on' despite not increasing solution fitness. The schema was chosen for its simplicity rather than its efficiency [19,20,21]. We impose a limit of 1500 generations on the search, as a run that fails to find the shortest path within this will have taken approximately 14 hours to complete. However, failure to solve these smaller instances of the TSP using the SWCNT-PMMA material is relatively rare (although the failure rate has been observed to increase on larger problems with fewer configurations); so allowing the search to continue beyond 1500 generations seems unlikely to impact the average number of generations to find the shortest path.

4 Results

To the best of our knowledge, this work represents the first time the TSP has been solved using search algorithms to configure a material for computation. Knowing the exact size of the solution space for small instances of TSP allows us to calculate the proportion of samples taken from the total solution space that were required to find the shortest path. By comparing the number of samples taken from the total solution space to the number taken by doing the same computation entirely in software, we get an insight into whether performing the computation via the SWCNT-PMMA material has any advantages over doing the same task entirely in software. In order to do this, we wanted to chose an evolutionary algorithm that could be meaningfully compared to

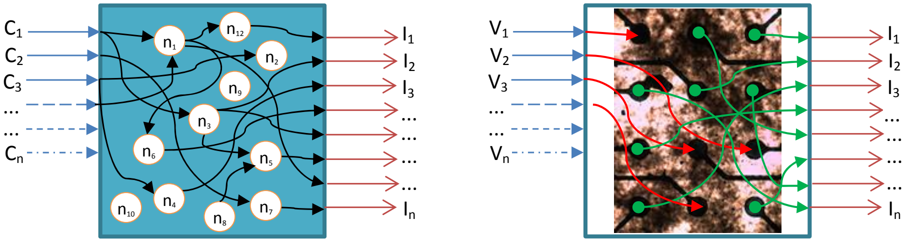

Fig. 3. The two evolutionary search algorithms are not exact correlates. The LHS represents the CGP evolved network of nodes and mathematical operators. In this schema, the inputs $C_{1..n}$ to the network are constants initially chosen at random but which do not change. The remaining connections and operators within each node are open to mutation, including those nodes selected as connecting to the inputs $I_{1..n}$ to the TSP. The RHS shows the SWCNT-PMMA material over an early 4x3 electrode. In this case the evolutionary algorithm is able to select which electrodes serve as inputs to the TSP and which will take the configuration voltages $V_{1..n}$ (note that in contrast to CGP, the input voltages are subject to evolutionary selection).

our hybrid material-based computation. Although we have no current means to analyse what electrical networks or properties are created within the SWCNT-PMMA material during its configuration, we can treat it as a ‘black box’ which takes certain inputs and transforms them into our desired outputs.

Cartesian Genetic Programming (CGP [22]) is a technique that creates a similar ‘black box’ of up to a 1000 networked nodes, within which mathematical and arithmetic operators transform inputs before passing on the values to other nodes or outputs [23]. The evolutionary mutation of the genotypes that describe the CGP network has parallels with our evolutionary mutation of the voltages and switch connections that affect the SWCNT-PMMA material’s configuration. We show a schematic comparison of the two methods in Figure 3. As CGP is known to be an efficient technique at solving well known benchmark problems [24] one would not normally expect a 1+4 evolutionary schema in software to outperform a similar schema in CGP. We encoded the same TSP instances to be solved by CGP and compared its performance against the SWCNT–PMMA material. It should be noted that as CGP running in software runs many times quicker than the hybrid SWCNT-PMMA system, it was possible to experiment with the CGP parameters to optimise its performance on the problem. Indeed, we discovered that we could improve the average number of generations to find the shortest path for CGP by a factor of ten or more. Time constraints meant this sort of experimentation was impossible for the SWCNT-PMMA material.

Since Thompson’s work it has been known that evolutionary algorithms with access to physical resources can make use of them in ways that are difficult to analyse or even predict [25]. Even so we were not expecting that instances of the TSP would be solved almost as efficiently (in terms of sampling the solution space) and reliably (in terms of successful runs) by the hybrid system using the SWCNT-PMMA material as an optimised encoding of CGP. While it is difficult to draw exact comparisons between the methods, both Table 1 and Figure 4 show that increasing the number of configuration

Table 1. Results for an optimised version of CGP and the hybrid evolutionary system with SWCNT-PMMA material for different instances of the TSP. Note proximity in the median number of generations of highlighted instances despite a decrease in the average number of samples from the solution space by a factor of almost ten.

Computation type	Size of TSP	No. of configuration voltages	Average no. of generations for successful runs	Median no. of generations	Average % sample of solution space
SWCNT-PMMA substrate	9	2	158.6	104.5	0.1751
	9	3	57.36	42.5	0.0741
	9	4	118.4	61.5	0.1308
	10	2	157.95	155	0.0174
	10	3	79.76	63.5	0.0086
	10	4	68.03	46.5	0.0075
	11	2			
	11	3			
	11	4	170.7	92	0.0017
Software (CGP encording)	9	n/a	34.04	29.5	0.4599
	10	n/a	48.96	40.5	0.0643
	11	n/a	91.96	65.5	0.0065

voltages exposed to evolutionary selection seems to improve search performance up to point, beyond which the performance starts to degrade again. We do not yet have a hypothesis as to why this is as it is impossible to know what physical characteristics in the material that the search algorithm is exploiting. However, the results hold true across different SWCNT-PMMA materials (i.e. changing the percentage of CNT within the PMMA) and across different city layouts for the TSP instances (e.g. random and irregular geometric arrangements).

Some instability observed in the population's fitness scores during the evolutionary runs for TSPs with 2 configuration voltages (first column in Figure 4) led us to suspect that increasing the number of configuration voltages on larger instances would lead to better performance. While we observed almost identical levels of performance for 10 city with 4 configuration voltages as for 9 city with 3, we are unable to verify the performance of the hybrid system on an 11 city TSP beyond 4 configuration voltages as this is the maximum number of analogue outputs that our current equipment supports. It may be the case that there is a "sweet spot" for optimal performance based on the proportion of configuration voltages to the number of cities in the TSP instance, which from our (limited) initial results might be somewhere between a third and a half (e.g. 12 city might need around 5).

The instability during some evolutionary runs is caused by voltages that are recorded as floating point numbers in the sorting vector converging to nearly identical values. Noise (in the μV range) can then cause a radical difference to the permutation that determines which order the cities are visited. This re-ordering of vector values can mean that a search run which had reached a plateau representing a local fitness optima can escape and go on to find the global optimum. We suspect this "feature" of the hardware may be one factor that helps the SWCNT-PMMA materials outperform CGP in one case, although we are aware this argument is contrary to results for the 9 city TSP with 3 and 10 city with 4 configuration voltages as in these instances there is almost no instability recorded during the evolutionary runs and there may well be more important factors.

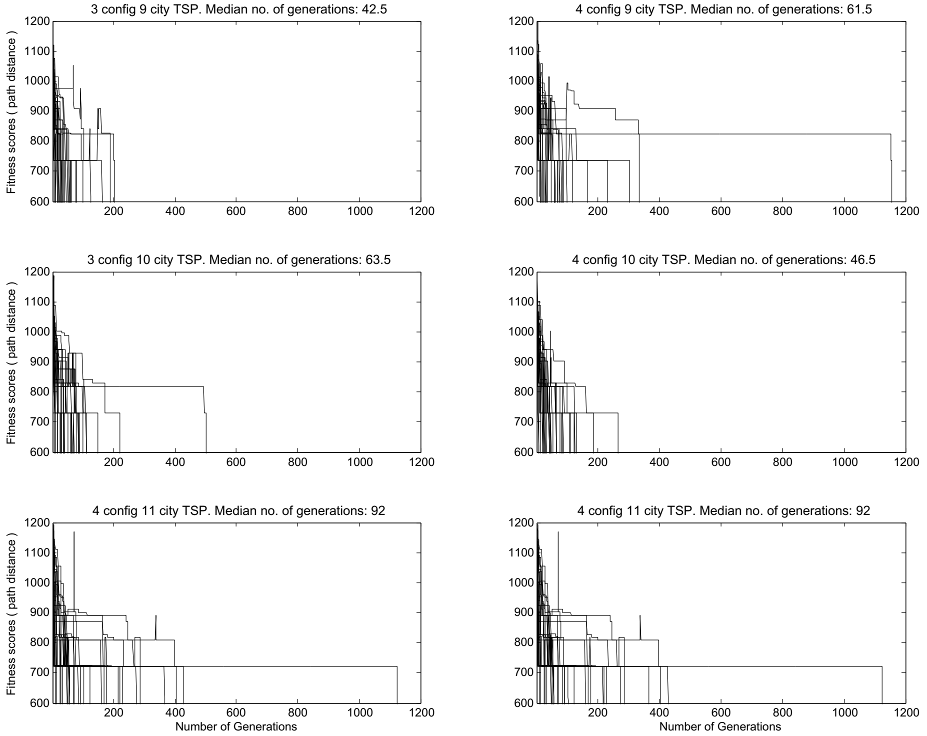

Fig. 4. The plots show the evolutionary histories of 30 runs of 9, 10 and 11 TSP instances using the same SWCNT-PMMA material over 1500 generations. There appears to be a "sweet spot" for the number of configurations voltages for optimal search performance that may increase as the number of cities increases. Unfortunately we cannot confirm this as our current hardware is limited to a maximum of 4 configuration voltages. Note how the material struggles with insufficient configuration voltages; those runs using 2 configuration voltages all show instability and occasional failure to find the shortest path. The median number of generations to find the shortest path is given as some runs have outliers.

Another intriguing finding is that despite the factorial increase in solution space between 9, 10 and 11 city TSPs, the average number of generations for the SWCNT-PMMA material to find the shortest path remains almost constant in the best cases. This might suggests that the performance of the hybrid system with the SWCNT-PMMA material could have a linear time complexity (rather than exponential) on NP complete problems such as the TSP. However until we are able to build and test larger SWCNT-PMMA electrode arrays, we are cannot confirm that this is the case and can only say that the potential for the hybrid SWCNT-PMMA system to solve large instances of the TSP efficiently appears promising.

5 Conclusion

Currently our 16 electrode array and low number of analogue outputs restrict the maximum size of TSP we can tackle. However, our most recent experiments suggest it is possible to split the genome of TSP configuration so that the problem is effectively halved. This allows us to sequentially process much larger TSP instances using the same array. While this appears to work, the evolutionary runs are more than doubled in duration making it difficult to acquire meaningful performance data. However, as a proof of concept it lays the way towards trying to tackle scalability by making two 4x4 arrays and using them to solve a single larger instance of the TSP. Such an approach would demonstrate parallel computation that could massively speed up search times, as all inputs to the problem are read simultaneously. While we are at an early stage in this research, we believe that it may be possible in the future to configure many thousands of small SWCNT-based electrode arrays to perform a computation in the way we currently manufacture and program millions of logic gates in silicon.

References

1. Walker, J., Trefzer, M., Bale, S., Tyrrell, A.: Panda: A reconfigurable architecture that adapts to physical substrate variations. IEEE Transactions on Computers 62(8), 1584–1596 (2013)
2. Walker, J.A., Trefzer, M.A., Tyrrell, A.M.: Designing function configuration decoders for the PAnDA architecture using multi-objective cartesian genetic programming. In: Suganthan, P.N. (ed.) 2013 IEEE Symposium Series on Computational Intelligence, Singapore, April 16-19, pp. 96–103 (2013)
3. Broersma, H., Gomez, F., Miller, J.F., Petty, M., Tufte, G.: Nascence project: Nanoscale engineering for novel computation using evolution. International Journal of Unconventional Computing 8(4), 313–317 (2012)
4. Graham, A.P., Duesberg, G.S., Hoenlein, W., Kreupl, F., Liebau, M., Martin, R., Rajasekharan, B., Pamler, W., Seidel, R., Steinhoegl, W., Unger, E.: How do carbon nanotubes fit into the semiconductor roadmap? Applied Physics A 80(6), 1141–1151 (2005)
5. Harding, S., Miller, J.F.: Evolution in materio: Evolving logic gates in liquid crystal. International Journal of Unconventional Computing 3(4), 243–257 (2007)
6. Thompson, A.: Evolving electronic robot controllers that exploit hardware resources. In: Morán, F., Merelo, J.J., Moreno, A., Chacon, P. (eds.) ECAL 1995. LNCS, vol. 929, pp. 640–656. Springer, Heidelberg (1995)
7. Miller, J.F., Harding, S.L., Tufte, G.: Evolution-in-materio: evolving computation in materials. Evolutionary Intelligence 7, 49–67 (2014)
8. Mills, J.W.: Polymer processors, tech. rep. tr580. Technical report, Department of Computer Science, University of Indiana (1995)
9. Harding, S., Miller, J.F.: Evolution in materio: Investigating the stability of robot controllers evolved in liquid crystal. In: Moreno, J.M., Madrenas, J., Cosp, J. (eds.) ICES 2005. LNCS, vol. 3637, pp. 155–164. Springer, Heidelberg (2005)
10. Thompson, A.: Evolving fault tolerant systems. In: Proc. 1st IEE/IEEE Int. Conf. on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA 1995), IEE Conf. Publication No. 414, pp. 524–529 (1995)
11. Harding, S., Miller, J.: Evolution in materio: a tone discriminator in liquid crystal. In: Congress on Evolutionary Computation, CEC 2004, vol. 2, pp. 1800–1807 (June 2004)

12. Adleman, L.M.: Molecular computation of solutions to combinatorial problems. Science 266 (5187), 1021–1024 (1994)
13. Adamatzky, A.: Reaction-Diffusion Automata: Phenomenology, Localisations, Computation. Springer, Heidelberg (2013)
14. Thompson, A.: Exploring beyond the scope of human design: Automatic generation of FPGA configurations through artificial evolution (Keynote). In: Proc. 8th Annual Advanced PLD & FPGA Conference, Miller Freeman, pp. 5–8 (1998)
15. Harding, S.L., Miller, J.F., Rietman, E.A.: Evolution in materio: Exploiting the physics of materials for computation. International Journal of Unconventional Computing 4(2), 155–194 (2008)
16. Reinelt, G.: The Traveling Salesman. LNCS, vol. 840. Springer, Heidelberg (1994)
17. Fatih Tasgetiren, M., Sevkli, M., Yun-Chia, L., Gencyilmaz, G.: Particle swarm optimization algorithm for single machine total weighted tardiness problem. In: Congress on Evolutionary Computation, CEC 2004, vol. 2, pp. 1412–1419 (2004)
18. Tasgetiren, F., Chen, A., Gencyilmaz, G., Gattoufi, S.: Smallest position value approach. In: Onwubolu, G.C., Davendra, D. (eds.) Differential Evolution: A Handbook for Global Permutation-Based Combinatorial Optimization. SCI, vol. 175, pp. 121–138. Springer, Heidelberg (2009)
19. Rechenberg, I.: Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Frommann-Holzbog (1973)
20. Schwefel, H.P.: Numerical optimization of Computer models. John Wiley & Sons, Ltd. (1981)
21. Beyer, H.G., Schwefel, H.P., Wegener, I.: How to analyse evolutionary algorithms. Theoretical Computer Science 287(1), 101–130 (2002)
22. Miller, J.F.: Cartesian Genetic Programming. Springer (2011)
23. Miller, J.F., Mohid, M.: Function optimization using Cartesian Genetic Programming. In: Proc. Conf. on Genetic and Evolutionary Computation (Companion), pp. 147–148 (2013)
24. Walker, J., Miller, J.F.: The automatic acquisition, evolution and re-use of modules in cartesian genetic programming. IEEE Transactions on Evolutionary Computation 12, 397–417 (2008)
25. Thompson, A.: An Evolved Circuit, Intrinsic in Silicon, Entwined with Physics. In: Higuchi, T., Iwata, M., Weixin, L. (eds.) ICES 1996. LNCS, vol. 1259, pp. 390–405. Springer, Heidelberg (1997)