Evolution-in-materio:
evolving computation using physical processes

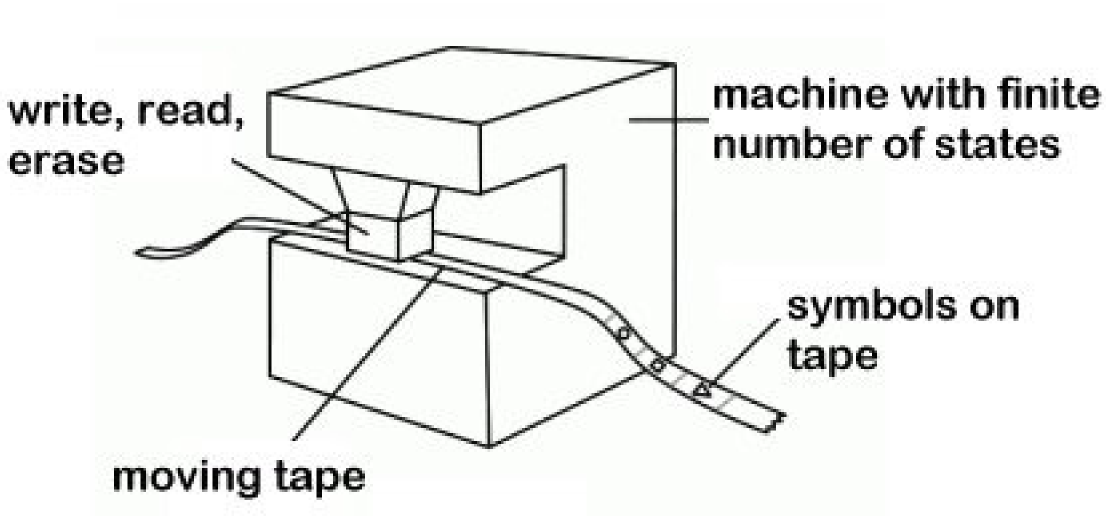
Julian F Miller
Department of Electronics

UNIVERSITY *of York*

NASCENCE

Turing Computation

- The dominant model of computation is that proposed by Turing and Church
- “a problem can be solved by an algorithm iff it can be solved by a Turing Machine”



Who or what writes the program?

Where is the physics?

Evolved machines versus Turing machines

- *Natural evolution* has created biological “machines” that can invent Turing machines and solve many computational problems
- *Artificial evolution* is a type of program running on a Turing machine
  - But Turing machines do not use physics, they are symbolic machines
  - This means artificial evolution does not have access to the physics of the real world. Does it have to be like this?

Some dangers of conventional programming…

- “In conventional design the vast majority of interactions that could possibly contribute to the problem are deliberately excluded” (Michael Conrad 1988)
- “Get a computer to do what needs to be done, without telling it how to do it” (Arthur Samuel 1983)
- “Nothing makes sense in computing except in the light of evolution” (Toffoli 2005)

What is the computational power of matter?

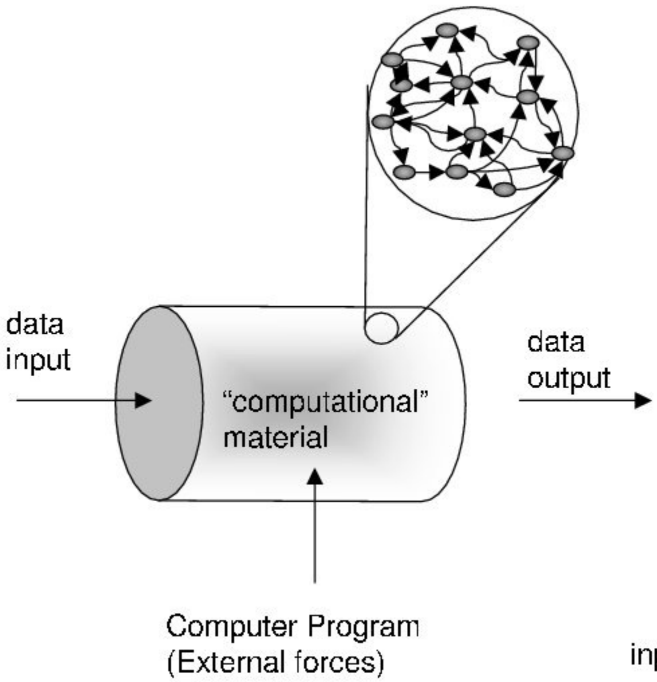
- Seth Lloyd has calculated the potential amount of computation possible in matter. He calculated:
  - 1Kg of matter should be able to carry out about 5.5×10^{50} operations per second and store 10^{31} bits.
- Shouldn't we be trying to directly exploit matter for computation?

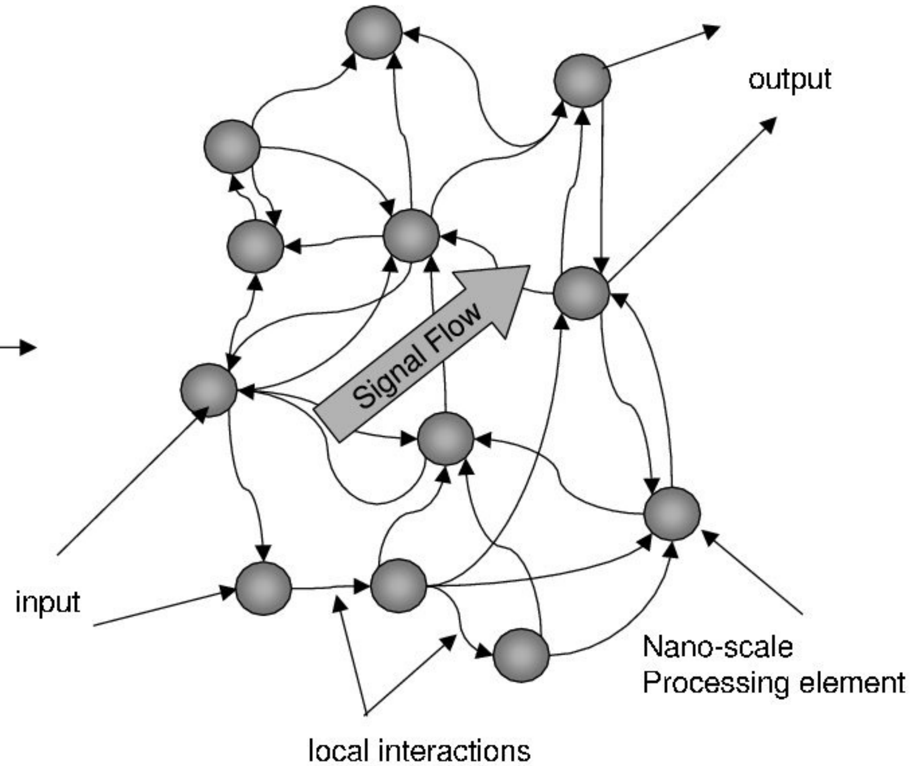
- Lloyd S (2000) Ultimate physical limits to computation. Nature 406:1047–1054

Evolution-in-materio

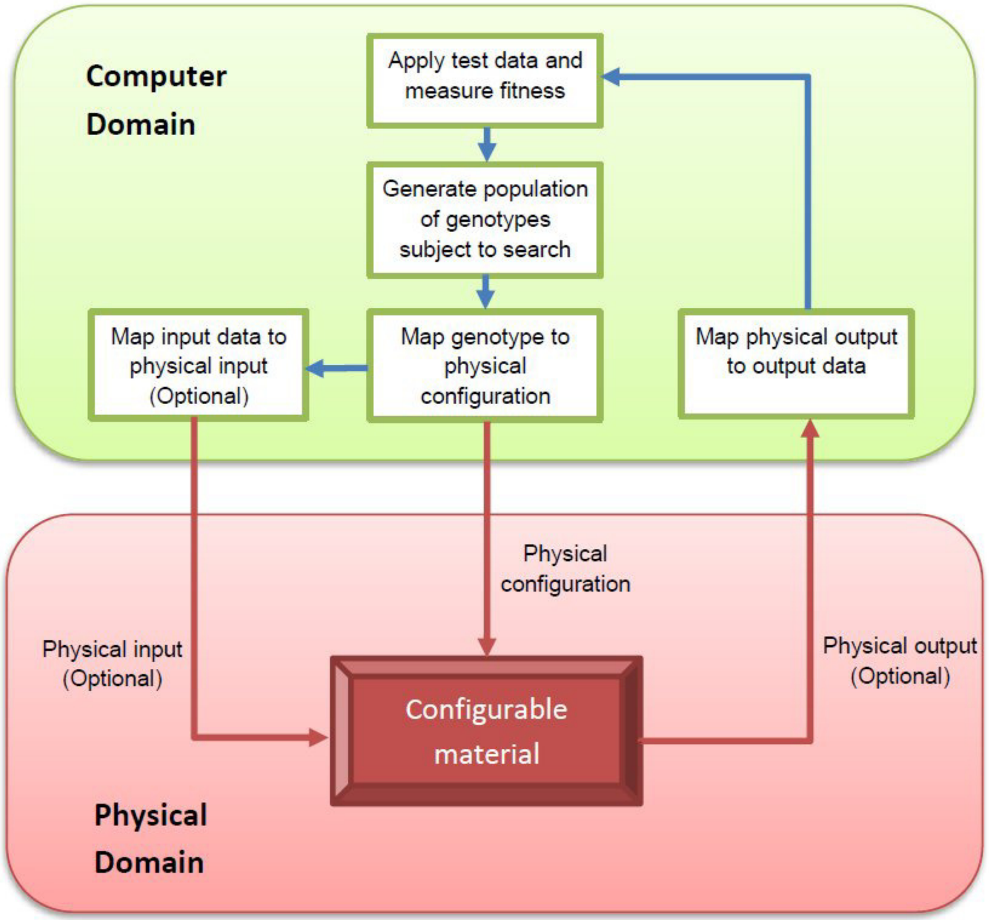
- Natural evolution has been exploiting the physical properties of materials (i.e. proteins) for billions of year
- Evolution-in-materio aims to allow artificial evolution to exploit the properties of materials to solve problems (particularly computational)
  - One of the potential advantages of this is that artificial evolution can potentially exploit physical effects that are either too complex to understand or hitherto unknown.
  - Exploiting the richness of the physical world ought to make it easier to evolve solutions than in simulation

Getting matter to compute





One way to do evolution-in-materio



Evolution-in-materio: a brief history

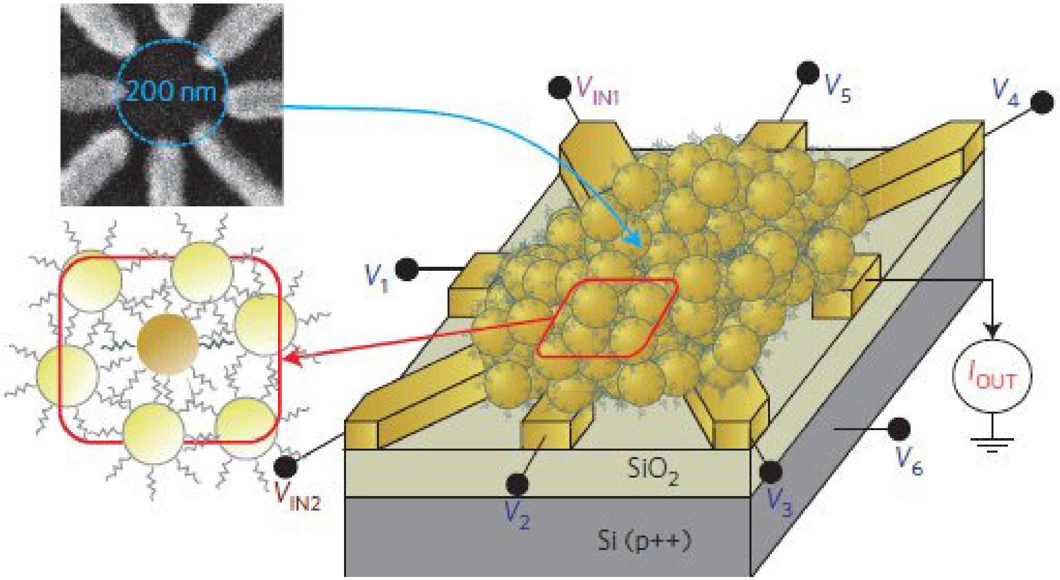
Name	Year	Material
Pask	1958	Ferrous sulphate
Mills	1995	Conducting polymer
Thompson	1996	Silicon (FPGA)
Huelsbergen et al.	1998	Silicon (FPGA)
Layzell	1998	Silicon (Switch array)
Stoica et al.	2000	Silicon (Transistor array)
Langeheine et al.	2000	Silicon (custom FPGA)
Linden	2001	Metal (Reed-switch array)
Harding & Miller	2004	Liquid Crystal (with Switch array)
NASCENCE	2013	SWNT-Polymer (with Switch array), Gold nanoparticles

For review: see Miller, Harding, Tufte, Evolutionary Intelligence 2014

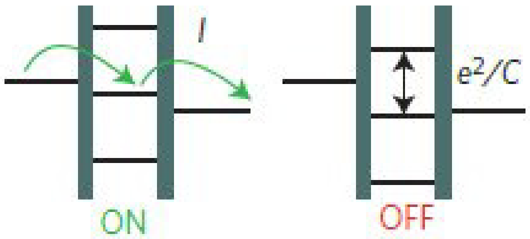
Recently attempted computational problems

Problem	Number of inputs	Number of outputs	Comments	Status
Travelling Salesman	None	Possibly many	Classic NP complete problem	11 city solved PPSN 2014
Tone Discriminator	Few, time dependent	Few	Standard problem	Frequency classifier ICES 2014, SOC2015
Bin Packing	None	Possibly many	Classic NP complete problem	ICES 2014
Robot control	Medium	Medium	Needs simulated or real robot	ECAL 2015
Classification	Variable	Variable	Classic machine learning benchmark	UCI standard problems Lenses/IRIS PPSN 2014, UC journal
Function optimization	None	Many	Classic EC problem	UKCI 2014, SOC2015
Logic gates	Variable	Variable	Commonly studied	UCNC 2014 UC journal (threshold logic gates) Nature Nanotech 2015
3 and 4 parity	3 or 4	one	Genetic programming "benchmark"	UKCI 2015

Nanoparticle device (University of Twente)



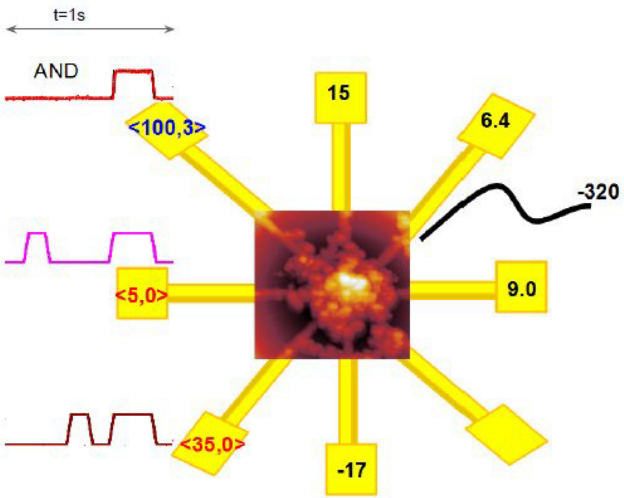
- Operates with gold nanoparticles
- Low temperature (< 1K)
- Nanoparticles act as single-electron transistors
- Applied voltages enable single electron tunneling



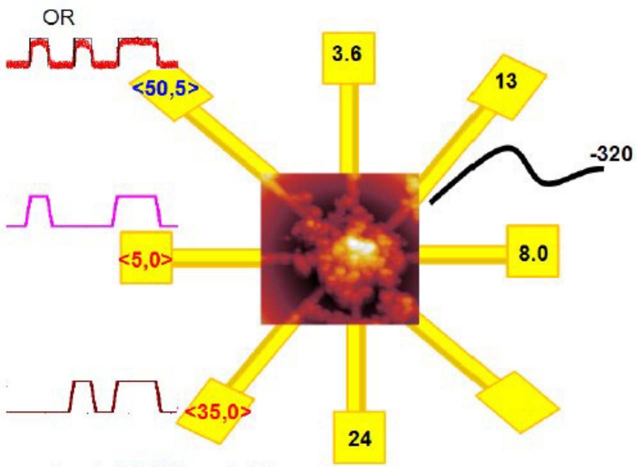
"One electron at a time can tunnel when sufficient energy is available (ON state), either by applying a voltage across the SET or by electrostatically shifting its potential. Otherwise, the transport is blocked because of the Coulomb blockade (OFF state)." Bose at al. Nat. Nanotech. 2015

Two-input logic gates: Nanoparticles

- Genotype data [millivolts]
  - 6 configuration (inc. backgate)
  - 2 electrodes used as inputs
  - 1 electrode is output
- Inputs are applied as pulse sequences
- All two-input Boolean functions have been obtained

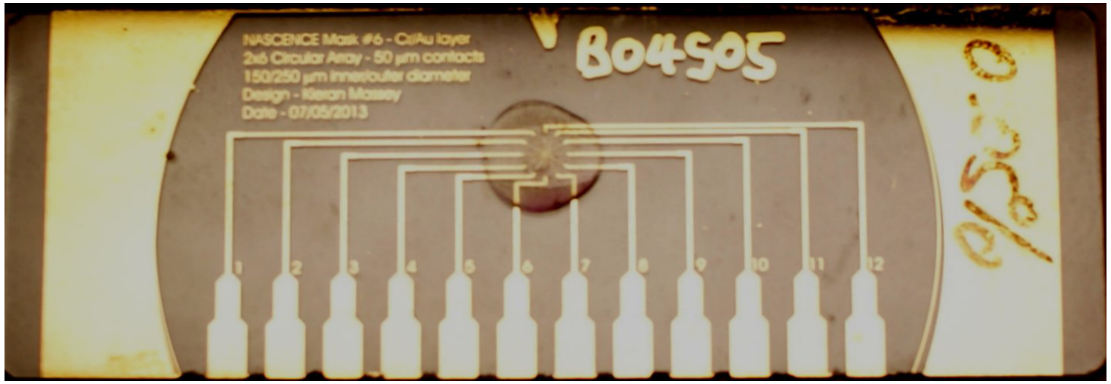


Input voltage signal **<high,low>** (mV)
Configuration pins are **DC voltage** (mV)
Output current signal **<high,low>** (pA)



Input voltage signal **<high,low>** (mV)
Configuration pins are **DC voltage** (mV)
Output current signal **<high,low>** (pA)

Another computational device

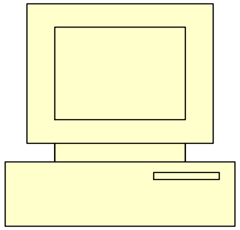


Device designed and fabricated by M. K Massey and M. C. Petty (Univ. of Durham)

- Twelve gold electrodes
- Single walled carbon nanotubes mixed with Polymethyl Methacralate (PMMA) in Anisole surfactant
  - Mixed using ultrasonic homogeniser
  - 20 μL is dropped onto a gold electrode array
  - Sample is then baked to evaporate Anisole

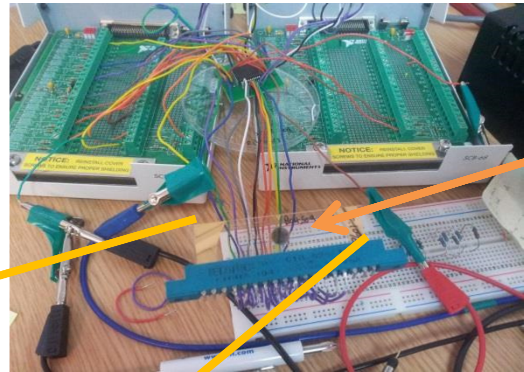
NI DAQ Experimental Setup (Univ. of York)

Matlab running on PC configures switch array and signals



DAQ card handles data acquisition and signal outputs

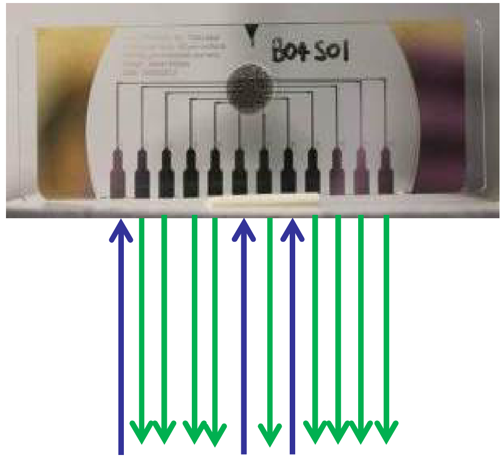
SCB-68 connection boards with 16x16 analogue switch to route connections



Sample being tested

Travelling Salesman: NI DAQ

- Genotype defines:
  - Configuration analogue voltages
  - Which electrodes will receive configuration voltages
  - Which electrodes are used as output
- Number of outputs equals number of cities
- Output vector sorted to read off permutation (SPV representation)

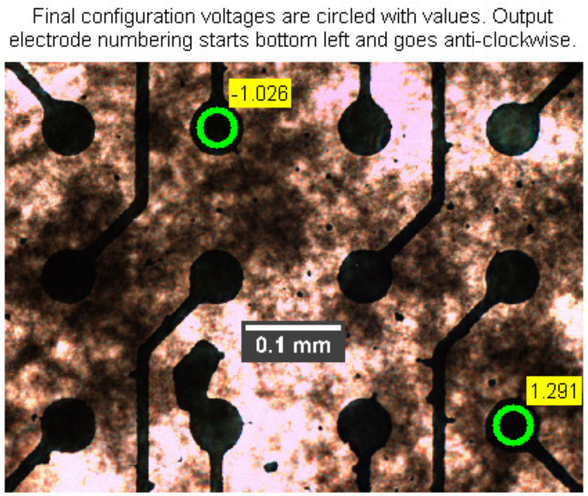


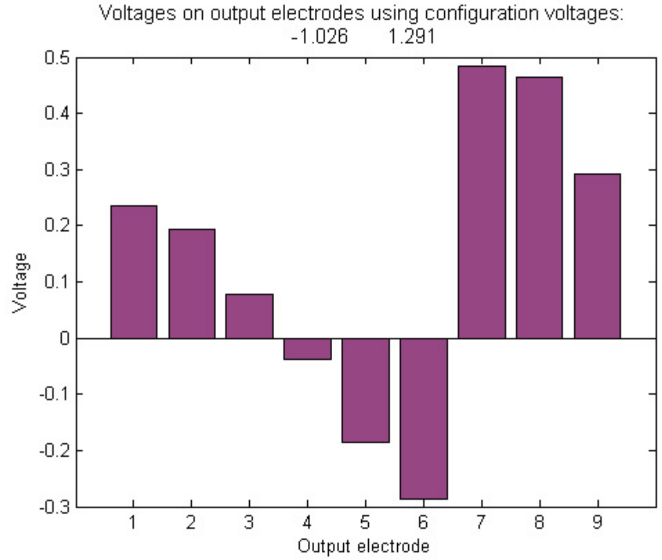
0.2	**-1.2**	**1.5**	**0.4**	**-2.3**	**0.7**	**1.6**	**-0.8**	**1.7**	**1.3**
1	2	3	4	5	6	7	8	9	10

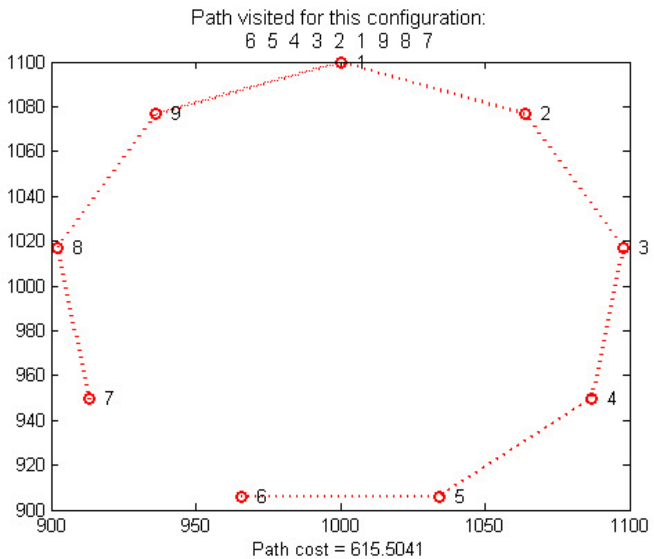
Sort by first field

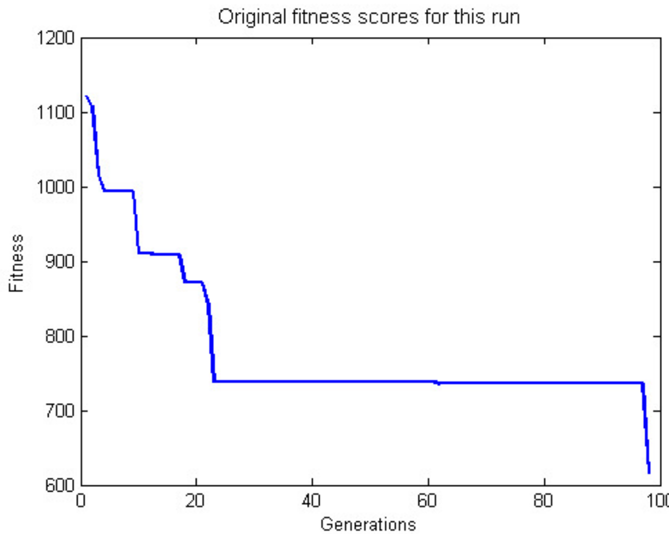
-2.3	**-1.2**	**-0.8**	**0.2**	**0.4**	**0.7**	**1.3**	**1.5**	**1.6**	**1.7**
5	2	8	1	4	6	10	3	7	9

Solving 10 city TSP

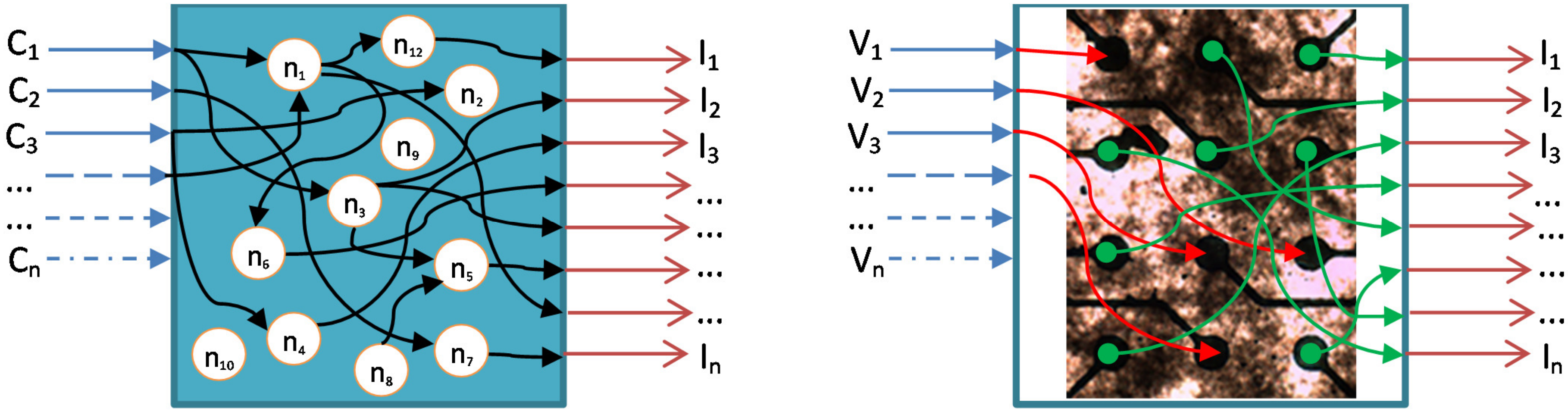








CGP vs SWCNT-PMMA



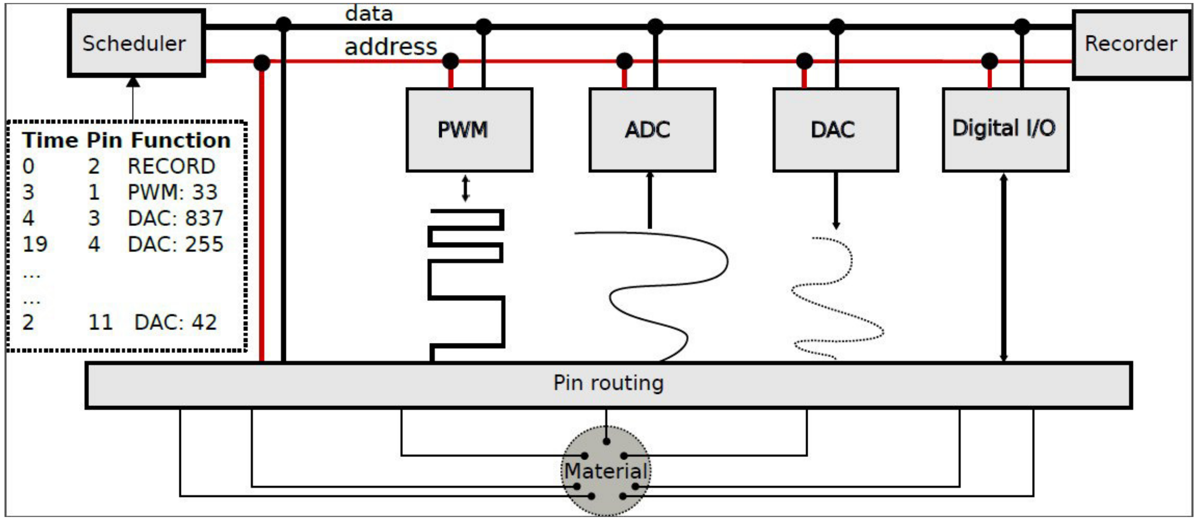
- The two evolutionary search algorithms are not exact correlates…
  - LHS: CGP evolved network of nodes and mathematical operators.
  - RHS: the SWCNT-PMMA material over an early 4x3 electrode.
- Each method uses a complex network of nodes that transforms the inputs to solve a problem.

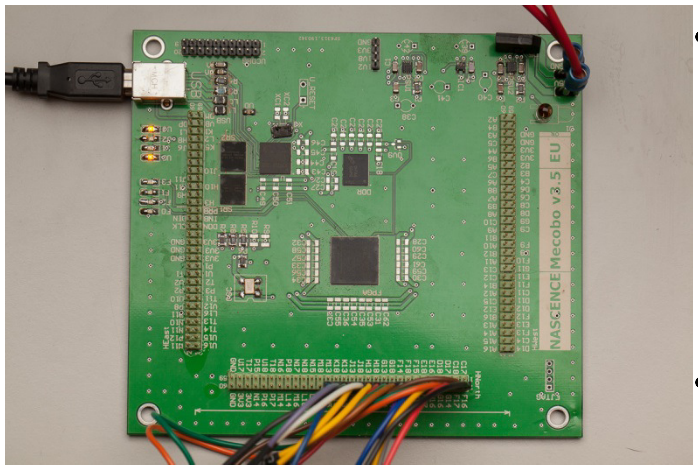
TSP results: in material compared with software evolutionary technique CGP: PPSN paper

Computation type (30 runs)	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	219.9	109	0.0022
	11	3	88.9	58.5	0.0008
	11	4	148.6	133.5	0.0015
Software (CGP encoding)	9	n/a	34.04	29.5	0.0378
	10	n/a	48.96	40.5	0.0054
	11	n/a	91.96	65.5	0.0009

Mecobo EIM platform (NTNU Norway)

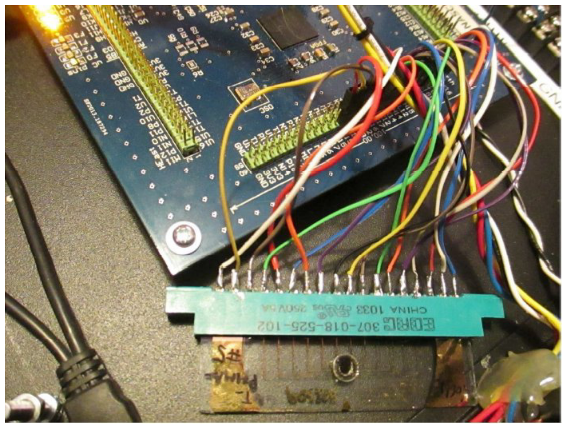
- A genome defines pin 2 to be the output terminal, pin 1 to be the data input and pin 3 - 12 to be configuration signals.





- Custom hardware designed and built by Gunnar Tufte and Odd Rune Lykebbø at NTNU

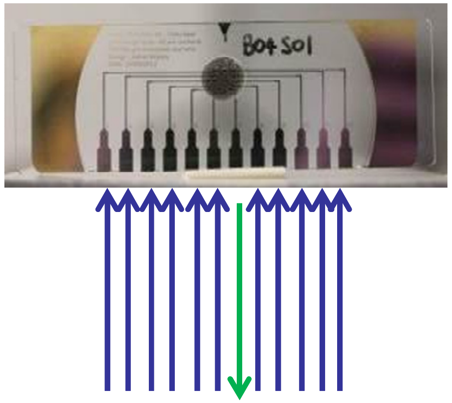
- Connected material sample (right)



Function Optimization: Mecobo

- Function optimisation consists of trying to find the minima of complex multi-modal functions

- Multiple chromosome (sequential)
  - 11 config and one output
  - Repeated until obtained number of outputs required by function optimization problem
  - Very slow
- Genotype defines for each iteration:
  - Which electrodes are used as output
  - Whether an input will receive a constant input or square wave, amplitude of input (0 or 1), frequency, phase, and duty cycle
- Output is read from digital buffer from sample
  - Mapping function written to convert to real-number
  - Proportional to percentage ones in output buffer
  - Linearly mapped to allowed ranges of domain variables

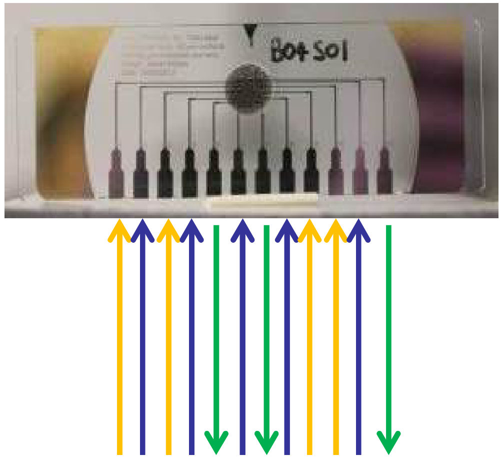


Function optimization results: in materio compared with CGP (Maktuba et al. UKCI 2014)

- Suite of 23 multi-modal complex optimization functions
  - 500 generations of 1+4-ES evolutionary algorithm
  - 12/23 functions EIM gave average results very close to optimum
  - 10/23 case average EIM results equal or are better than evolutionary software technique (CGP)

Classification: Mecobo

- Genotype
  - 5 configuration
  - 4 electrodes used as inputs
  - 3 electrodes are outputs (defining class)
- Genotype defines:
  - Which electrodes are used as output
  - Whether an input will receive a constant input or square wave, amplitude of input (0 or 1), frequency, phase, and duty cycle
- Output is read from digital buffer from sample
  - Average transition gap between 0 and 1 is computed
  - Class decided by whichever output is largest



Classification results: in-materio compared with software search (CGP) – Maktuba et al. PPSN 2014

Dataset Mecobo 3.0 (digital) Mecobo 3.5 (analogue)	Av. Training accuracy (material)	Av. Testing accuracy (material)	Av. Training accuracy (CGP)	Av. Testing accuracy (CGP)
Lenses • 24 instances • 4 attributes • 3 classes • Training set 16 • Testing set 8 • Unbalanced	92.7%	65.8%	93.8%	68.3%
Iris • 150 instances • 4 attributes • 3 classes • Training set 75 • Testing set 75 • Balanced	84.7% 91.33%	77.1% 86.6%	97.7% 87.2%	98.0% 84.4%

What is next?

- What materials *should* we use?
- How do the devices work?
- Scalability
  - How does the EIM scale on harder problems?
- Standalone computational devices
  - Standalone algorithms?
  - What speed can they operate at?
  - What power do they consume?
- Can room temperature devices be built using gold nanoparticle arrays?

Want to learn more about EIM?

Miller, J. F., Harding, S. L., Tufte, G. Evolution-in-materio: evolving computation in materials, Evolutionary Intelligence, Vol. 7 (2014) pp. 49–67

S. K. Bose, C. P. Lawrence, Z. Liu, K. S.Makarenko, R. M. J. van Damme, H. J. Broersma, W. G. van der Wiel. Evolution of a designless nanoparticle network into reconfigurable Boolean logic. Nature Nanotechnology DOI: 10.1038/NNANO.2015.207

M.K. Massey, A. Kotsialos, F. Qaiser, D. A. Zeze, C. Pearson, D. Volpati, L. Bowen, M.C. Petty Computing with carbon nanotubes: Optimization of threshold logic gates using disordered nanotube/polymer composites
J. Appl. Phys. 117, 134903 (2015); http://dx.doi.org/10.1063/1.4915343

Mohid, M. Miller, J. F., Harding, S. L, G. Tufte, M. K. Massey, M. C. Petty. Evolution-In-Materio: Solving Computational Problems Using Carbon Nanotube-Polymer Composites, Soft Computing 2015 (accepted)

Mohid, M. Miller, J. F. Evolving solution to computational problems using carbon nanotubes. International Journal of Unconventional Computing, 2015 (accepted)

Clegg, K. D., Miller, J. F., Massey, M. K., Petty, M. C. Travelling Salesman Problem solved ‘in materio’ by evolved carbon nanotube device. Proceedings of the 13th International Conference on Parallel Problem Solving from Nature (PPSN). Springer LNCS, Vol. 8672, pp. 692-701, 2014.

References

M. Conrad, "The price of programmability", in R. Herken (ed.), The Universal Turing Machine: A Fifty Year Survey, pp 285-307, Oxford University Press, 1988.

Lloyd S. Ultimate physical limits to computation. Nature 406:1047–1054, 2000.

A. Samuel. “AI: Where It has been and Where it is Going”, Int. Joint Conf. on AI, pp 1152-1157, 1983

Toffoli T (2005) Nothing makes sense in computing except in the light of evolution. Int J Unconv Comput 1(1):3–29