A Genetic Algorithm Approach to Guiding the Evolution of Self-Organised Nanostructured Systems

Peter Siepmann\textsuperscript{1}, Christopher Martin\textsuperscript{2}, Natalio Krasnogor\textsuperscript{1}, Philip Moriarty\textsuperscript{2}

Condensed Matter and Materials Physics
20\textsuperscript{th} April, University of Exeter
Overview

- Physical background
- Nanoparticle simulation details
- A brief overview of Genetic Algorithms
- Results from initial trials
- Conclusions & further work
A Genetic Algorithm Approach to Guiding the Evolution of Self-Organised Nanostructured Systems

Thiol-passivated Au nanoparticles

Gold core
Thiol groups
Sulphur ‘head’
Alkane ‘tail’, e.g. octane

Dispersed in toluene, and spin cast onto native-oxide-terminated silicon

~3nm

Au core
Thiol groups
Sulphur ‘head’
Alkane ‘tail’, e.g. octane
Au nanoparticles: Morphology

AFM images taken by Matthew O. Blunt, Nottingham
Nanoparticle Simulations

Solvent is represented as a two-dimensional lattice gas.

Each lattice site represents 1nm$^2$.

Nanoparticles are square, and occupy nine lattice sites.

Based on the simulations of Rabani et al. (Nature 2003, 426, 271-274). Includes modifications to include next-nearest neighbours to remove anisotropy.
Nanoparticle Simulations

• The simulation proceeds by the Metropolis algorithm:
  – Each solvent cell is examined and an attempt is made to convert from liquid to vapour (or vice-versa) with an acceptance probability $p_{acc} = \min[1, \exp(-\Delta H / k_B T)]$.
  – Similarly, the particles perform a random walk on wet areas of the substrate, but cannot move into dry areas.
  – The Hamiltonian from which $\Delta H$ is obtained is as follows:

$$H = -\varepsilon_l \sum_{\langle ij \rangle} l_i l_j - \varepsilon_n \sum_{\langle ij \rangle} n_i n_j - \varepsilon_{nl} \sum_{\langle ij \rangle} n_i l_j - \mu \sum_i l_i$$
Nanoparticle Simulations
Nanoparticle Simulations
Nanoparticle Simulations
A brief overview of Genetic Algorithms

Motivation
- optimisation problems
- large search space
- inspired by Darwinian evolution

- area covered?
- degree of order?
- similarity to target pattern?

Motivation
- optimisation problems
- large search space
- inspired by Darwinian evolution
A brief overview of Genetic Algorithms

Evolution
- Recombination (mating)
  e.g. exchanging parameters
  ‘combine the best bits of each parent’
- Mutation
  e.g. altering the value of a parameter at random with some small probability
A brief overview of Genetic Algorithms

*Evolution*

- Recombination (mating)
  e.g. exchanging parameters
  ‘combine the best bits of each parent’

- Mutation
  e.g. altering the value of a parameter at random with some small probability
A brief overview of Genetic Algorithms

**Evolution**
- Recombination (mating)
  e.g. exchanging parameters
  ‘combine the best bits of each parent’
- Mutation
  e.g. altering the value of a parameter at random with some small probability
A brief overview of Genetic Algorithms

Evolution

- Recombination (mating)
  e.g. exchanging parameters
  ‘combine the best bits of each parent’

- Mutation
  e.g. altering the value of a parameter at random with some small probability
A brief overview of Genetic Algorithms

**Evolution**

- Recombination (mating)
  e.g. exchanging parameters
  ‘combine the best bits of each parent’

- Mutation
  e.g. altering the value of a parameter at random with some small probability
A brief overview of Genetic Algorithms

**Evolution**
- Recombination (mating)
  e.g. exchanging parameters
  ‘combine the best bits of each parent’
- Mutation
  e.g. altering the value of a parameter at random with some small probability

converges to optimum solution
Evolving towards a target pattern

*Fitness function:*

“How similar is this pattern to the target pattern we are trying to recreate?”

How do we measure this?

**The Universal Similarity Metric**

is a measure of similarity between two given objects, \( o_1 \) and \( o_2 \), in terms of information distance:

\[
d(o_1, o_2) = \frac{\max\{K(o_1 | o_2), K(o_2 | o_1)\}}{\max\{K(o_1), K(o_2)\}}
\]

where \( K(o) \) is the Kolmogorov complexity:

- \( K(o) \): The length of the shortest program for computing \( o \) by a Turing machine
- \( K(o_1 | o_2) \): How much (more) information is needed to produce object \( o_1 \) if one already knows object \( o_2 \)
Evolving towards a target pattern (simulated)

- Selected a target image from simulated data set
- Initialised GA
  - Roulette Wheel selection
  - Uniform crossover (probability 1)
  - Random reset mutation (probability 0.3)
  - Population size: 10
  - Offspring: 5
  - $\mu + \lambda$ replacement
- Ran the GA for 200 iterations
  - on a single processor server, run time $\approx 5$ days
  - using Nottingham’s cluster (up to 1024 nodes), run time $\approx 12$ hours
A Genetic Algorithm Approach to Guiding the Evolution of Self-Organised Nanostructured Systems

Evolving towards a target pattern (simulated)

Evolving to a simulated target

Target:
A Genetic Algorithm Approach to Guiding the Evolution of Self-Organised Nanostructured Systems

Evolving towards a target pattern (experimental)

Target:

Evolving to a experimental target

Generations

Average
Best

0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180 190 200
Conclusions and further work

• we can evolve target simulated behaviour using a GA with the USM
• work continues to improve the evolution of experimental behaviour
• use of more introspective fitness functions
e.g. Minkowski functionals
  - open ended (multiobjective) evolution
    e.g. “evolve a pattern with as many large spots as possible in as ordered a fashion as possible”
• parameter investigations
  - larger populations
• full fitness landscape analysis
A Genetic Algorithm Approach to Guiding the Evolution of Self-Organised Nanostructured Systems

Peter Siepmann\(^1\), Christopher Martin\(^2\), Natalio Krasnogor\(^1\), Philip Moriarty\(^2\)

Condensed Matter and Materials Physics
20th April, University of Exeter