Evolutionary design for the behaviour of cellular automaton-based complex systems

Peter Siepmann, Germán Terrazas, Natalio Krasnogor

School of Computer Science & IT
University of Nottingham

Adaptive Computing in Design and Manufacture
25^{th} April 2006
Bristol

Motivation

- automated design and optimisation of complex systems
    - cellular automata models

- how can we verify that our fitness function is robust?

- present a methodology for answering this question
    - supported by experimental illustration

Introduction

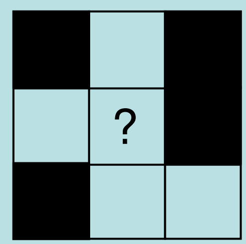
- Self-organising processes
- Modelled using **cellular automata**

Introduction

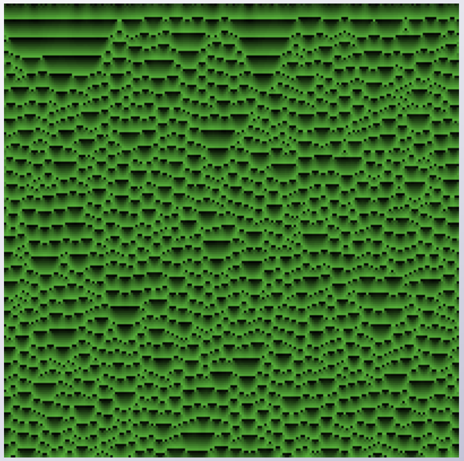
- Self-organising processes
- Modelled using **cellular automata**

- infinite, regular grid of cells
- each cell in one of a finite number of states
- at a given time, *t*, the state of a cell is a function of the states of its *neighbourhood* at time *t-1*.

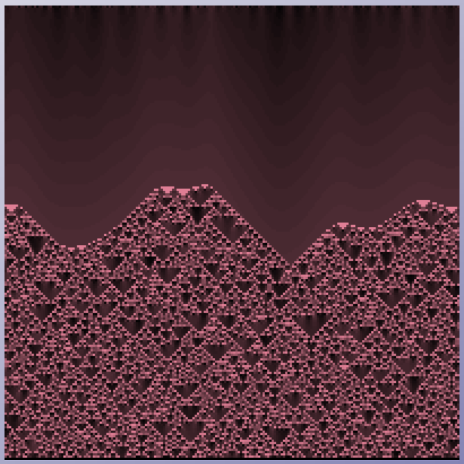
Example
- infinite sheet of graph paper
- each square is either black or white
- in this case, neighbours of a cell are the eight squares touching it
- for each of the 2^8 possible patterns, a rules table would state whether the center cell will be black or white on the next time step.



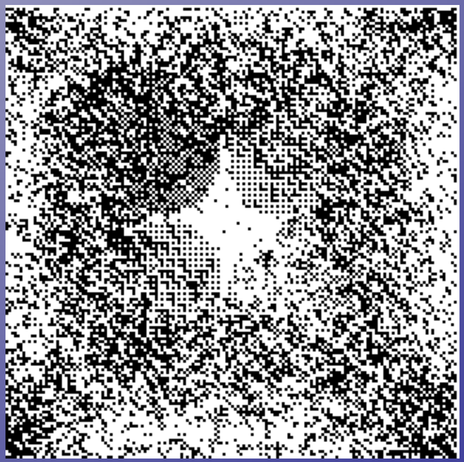
Introduction



CA continuous



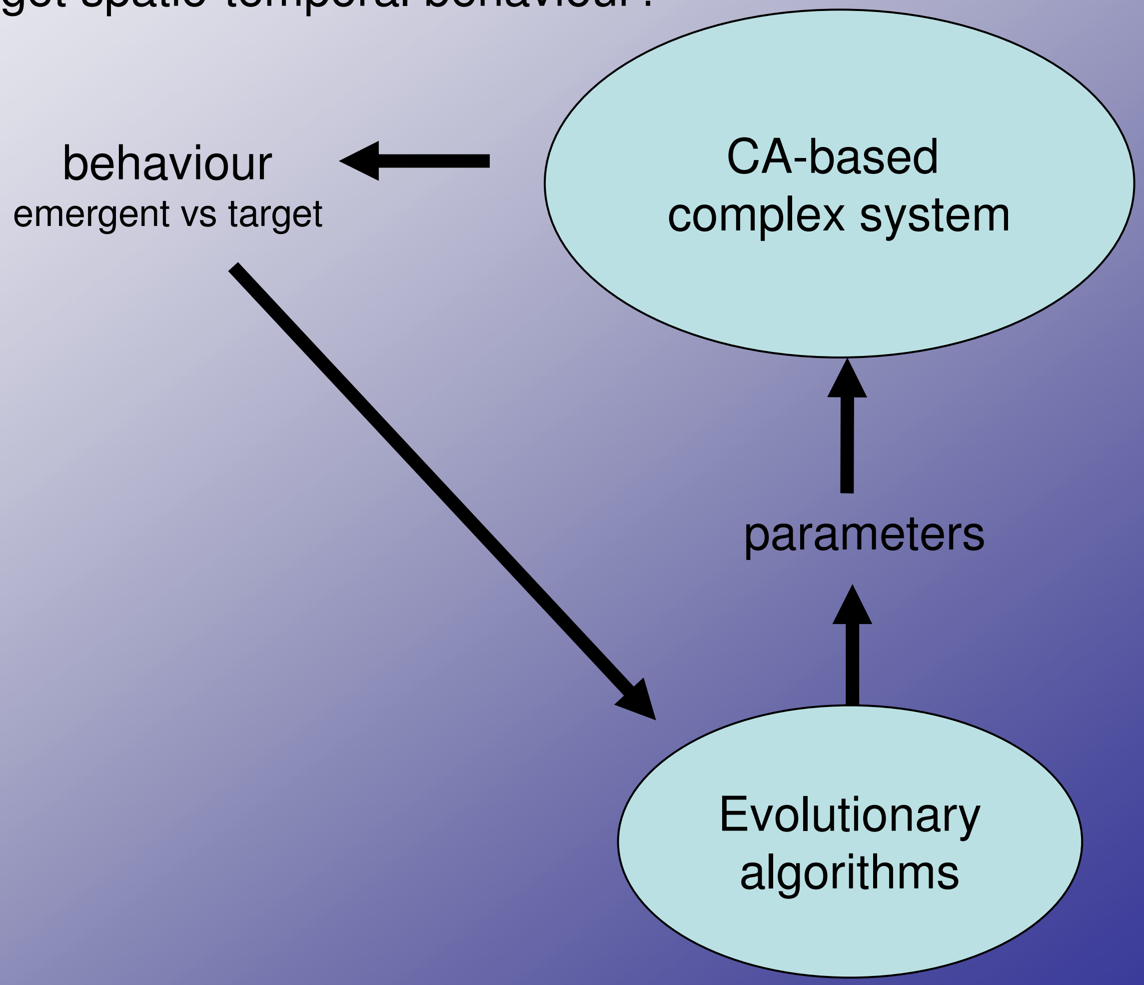
Turbulence



Gas Lattice

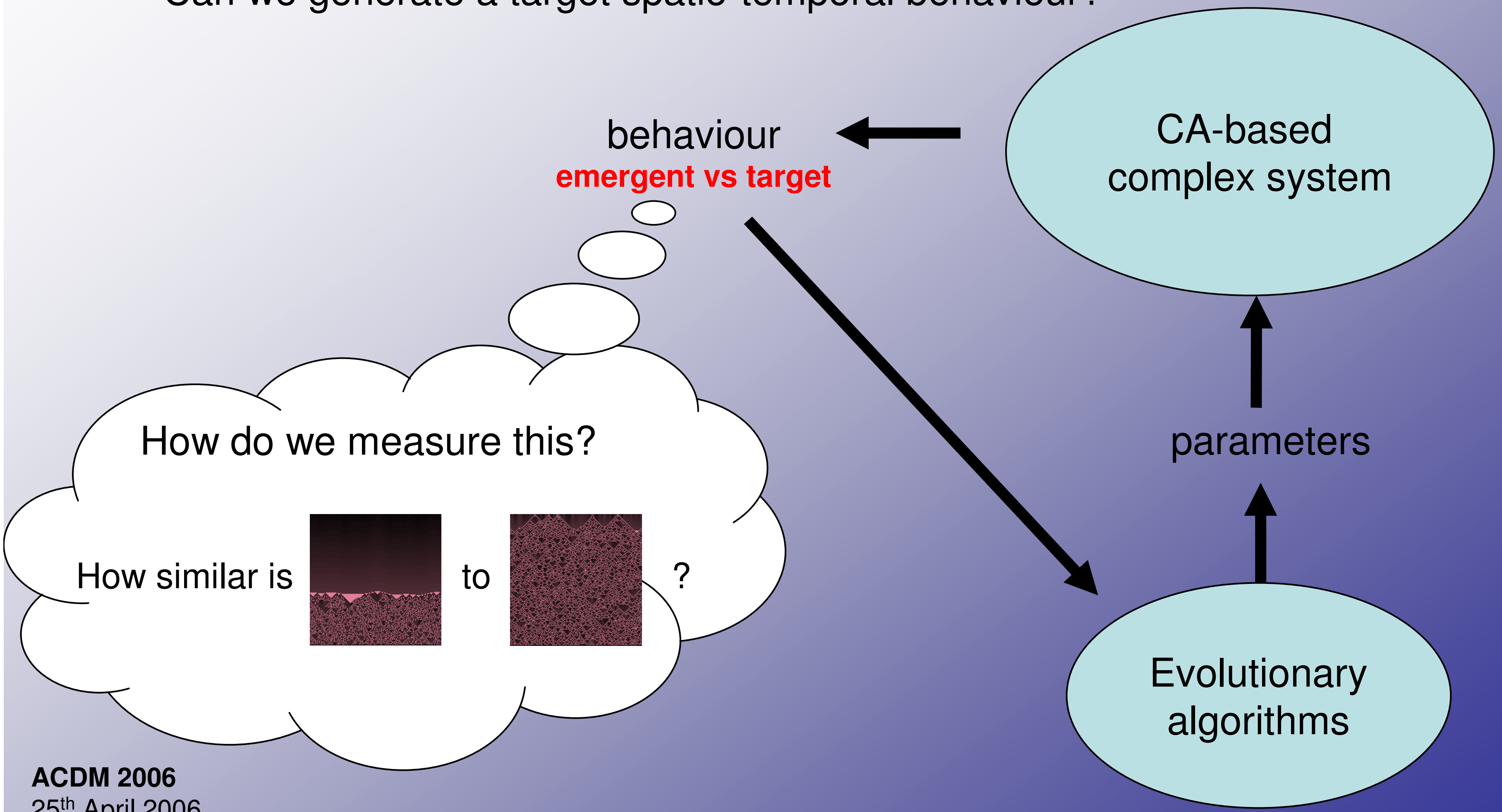
Introduction

- Can we generate a target spatio-temporal behaviour?



Introduction

- Can we generate a target spatio-temporal behaviour?



The Universal Similarity Metric (USM)

is a measure of similarity between two given objects in terms of information distance:

$$d(o_1, o_2) = \frac{\max\{K(o_1 \mid o_2), K(o_2 \mid o_1)\}}{\max\{K(o_1), K(o_2)\}}$$

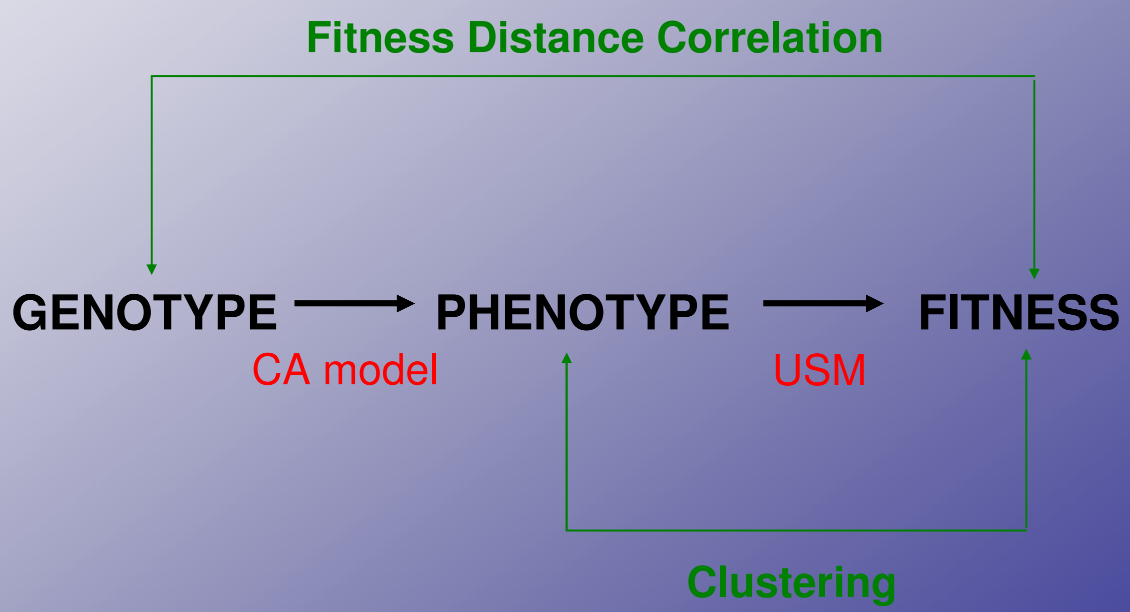
where K(o) is the Kolmogorov complexity

Prior Kolmogorov complexity K(o): The length of the shortest program for computing o by a Turing machine

Conditional Kolmogorov complexity K(o1|o2):
How much (more) information is needed to produce object o1 if one already knows object o2 (as input)

The Universal Similarity Metric (USM)

- Is the USM a good objective function for evolving target spacio-temporal behaviour in a CA system?
- methodology for answering this question
- experimental results



Data set

For each CA system:

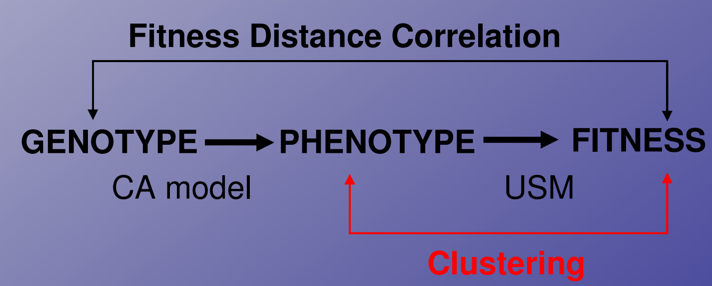
- Keep all but one parameter the same
- Produce 10 behaviour patterns through the variable parameter
- Repeat for other parameters

EXAMPLE

`turb_c4` refers to the spacio-temporal pattern produced by the fourth variation in parameter c of a *Turbulence* CA system

Clustering

- does the USM detect similarity of phenotype?
- calculate a similarity matrix filled with the results of the application of the USM to a set of objects
- during the clustering process, similar objects should be grouped together

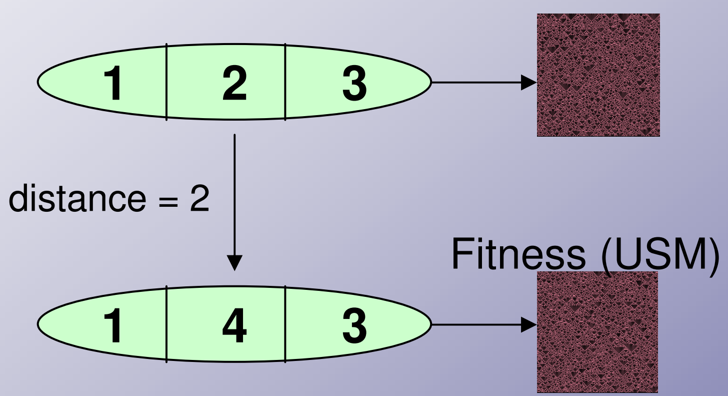


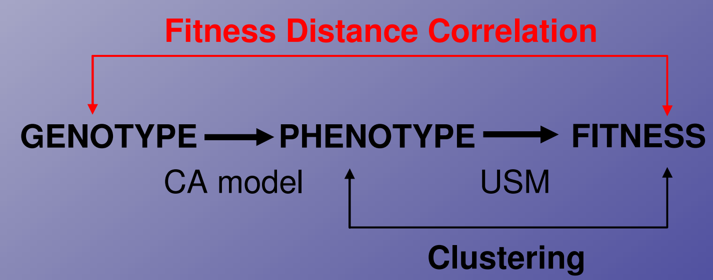

cac_h1
cac_h5
cac_h3
cac_h4
cac_h2
cac_b5
cac_b2
cac_b3
cac_b4
cac_b1
cac_i1
cac_i5
cac_i3
cac_i4
cac_i2
cac_j1
cac_j2
cac_j4
cac_j5
cac_j3
cac_d5
cac_d4
cac_d2
cac_d3
cac_d1
cac_c5
cac_c4
cac_c3
cac_c2
cac_c1
cac_g3
cac_g5
cac_g2
cac_g4
cac_g1
cac_e4
cac_e2
cac_e1
cac_e5
cac_e3
cac_k4
cac_k1
cac_k2
cac_k3
cac_a1
cac_a2
cac_a4
cac_a3
cac_a5
cac_f4
cac_f1
cac_f2
cac_f3
cac_f5

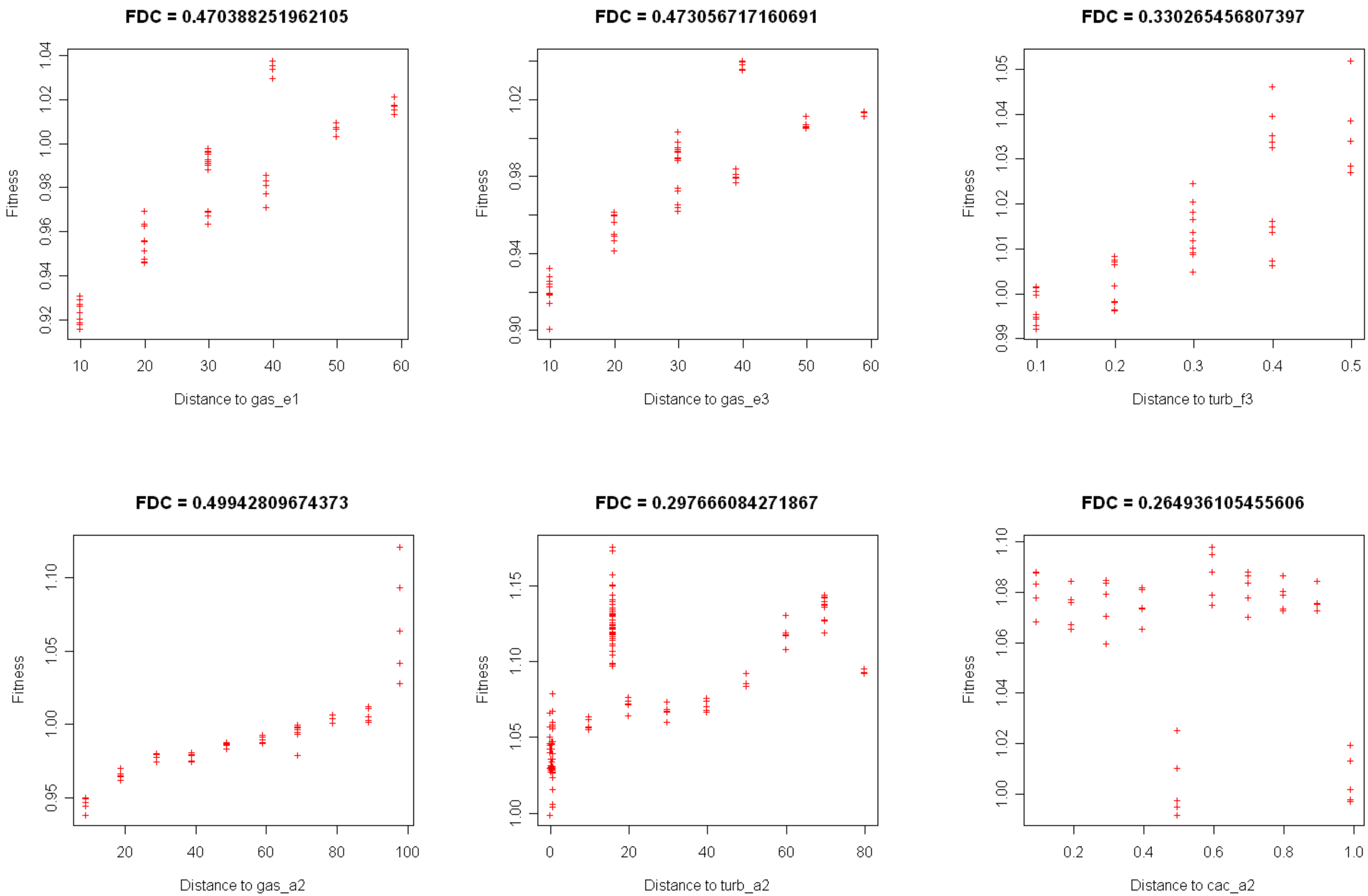
turb_a3
turb_a2
turb_a5
turb_b3
turb_d1
turb_d2
turb_d5
turb_h5
turb_h3
turb_h4
turb_i3
turb_i2
turb_i4
turb_c5
turb_c4
cac_i4
cac_i3
cac_f2
cac_h3
cac_h4
cac_h2
cac_g3
cac_a1
cac_a3
cac_a4
cac_f4
cac_f1
cac_f2
cac_e2
cac_e4
cac_e1
cac_d2
cac_d1
cac_b1
cac_b4
cac_b3
cac_c5
cac_c4
cac_c1
lat_j4
lat_h2
lat_i2
lat_h5
lat_g1
lat_g3
lat_g2
lat_f3
lat_a3
lat_a5
lat_a1
lat_e1
lat_e3
lat_e4
lat_d5
lat_d4
lat_d3
lat_d1
lat_d2
lat_b2
lat_b1
lat_b4

Fitness Distance Correlation

- correlation analyses of a given fitness function versus parametric (genotype) distance.
- larger numbers indicate the problem could be optimised by a GA
- numbers around zero [-0.15, 0.15] indicate bad correlation
- scatter plots are helpful






FDC = 0.470388251962105
Fitness
Distance to gas_e1
FDC = 0.473056717160691
Fitness
Distance to gas_e3
FDC = 0.330265456807397
Fitness
Distance to turb_f3
FDC = 0.49942809674373
Fitness
Distance to gas_a2
FDC = 0.297666084271867
Fitness
Distance to turb_a2
FDC = 0.264936105455606
Fitness
Distance to cac_a2

The Genetic Algorithm

Aim: to evolve a behaviour as similar as possible to some given target behaviour

Parameters

Chromosome: **i | c | r**

$i \in \Re$. $0 \le i \le 100$ *initial turbulence*

$c \in \Re$. $0 \le i \le 1$ *coupling strength*

$r \in \Re$. $0 \le i \le 0.025$ *roughness*

Population size: 10

Offspring: 4

Selection: roulette

Recombination: uniform crossover (probability 1)

Mutation: BCG (probability 0.6) — selects a value in a fixed distribution either side of current value

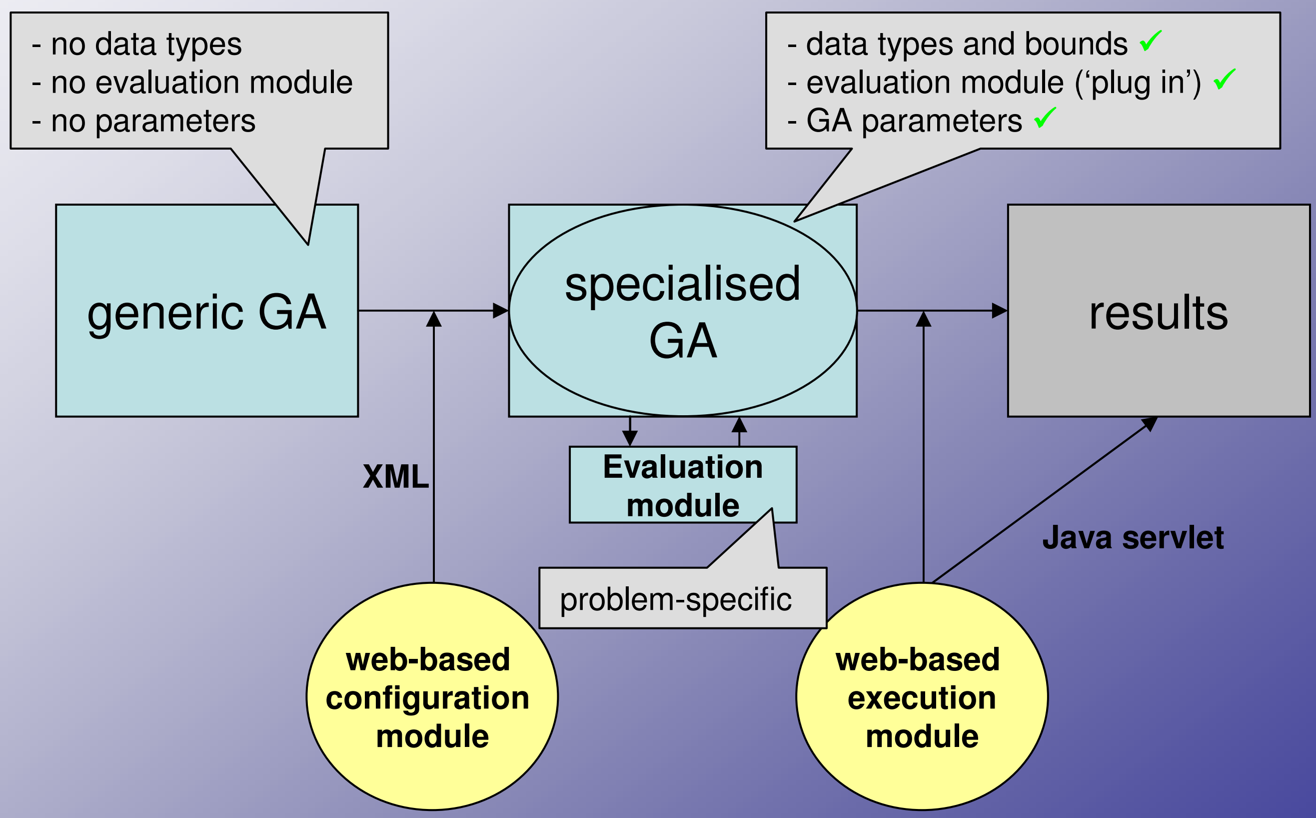
Stopping condition: 200 generations

The Genetic Algorithm

Our GA platform

"we will implement an object-oriented, platform-independent, evolutionary engine (EE). The EE will have a user-friendly interface that will allow the various platform users to specify the platform with which the EE will interact"

Evolvable CHELLware grant application



The Genetic Algorithm

Measuring success

- Visual comparison
- USM
- Genotypic error

We define the error for each gene, $e(g) = \frac{abs(g_T - g_F)}{g_T}$

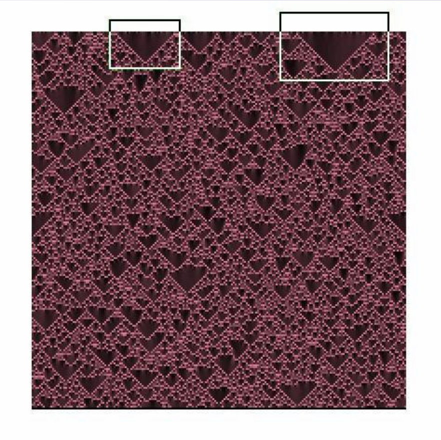
and the *average error* for a given individual, $E(F) = \frac{e(i_F) + e(c_F) + e(r_F)}{3}$

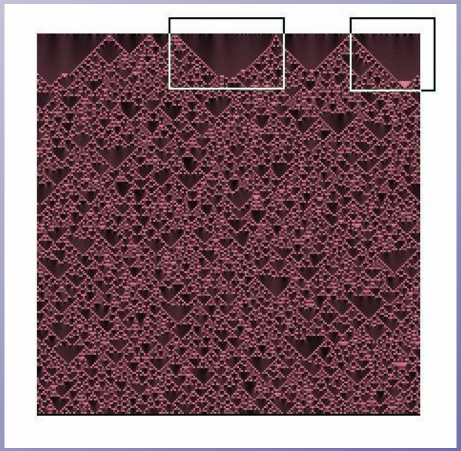
The Genetic Algorithm

Results

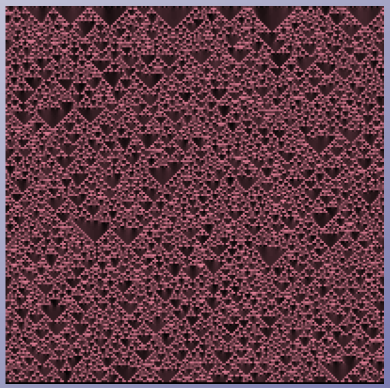
Target | *Designoid*

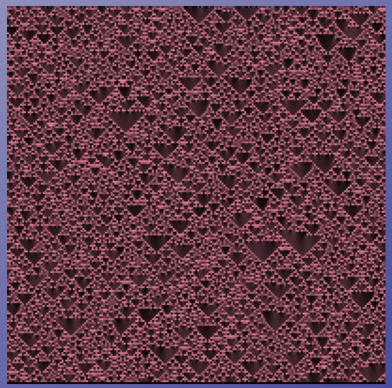
e5





f3





The Genetic Algorithm

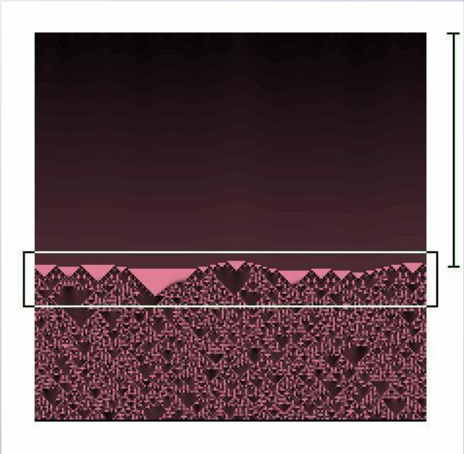
Results

Target		Genotypic error			
	***usm*(*F*,*T*)**	***e*(*i*)**	***e*(*c*)**	***e*(*r*)**	***E***
f2	**0.95657**	0.30659	0.01885	**12.77470**	4.36672
f3	**0.95608**	0.35404	**0.00426**	10.40107	3.58646

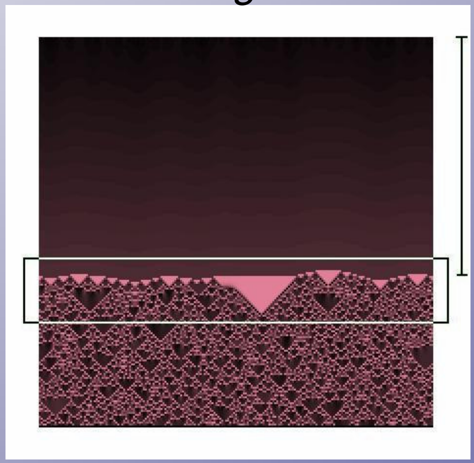
The Genetic Algorithm

Results

Target



Designoid



Target	***usm(F,T)***	***e(i)***	***e(c)***	***e(r)***	***E***
p	0.91980	0.26843	0.35314	0.05552	0.22569

Conclusions

- using Fitness Distance Correlation and Clustering, we can show the USM to be an appropriate objective function in this domain.

- can we generate a target spatio-temporal behaviour in a CA system?

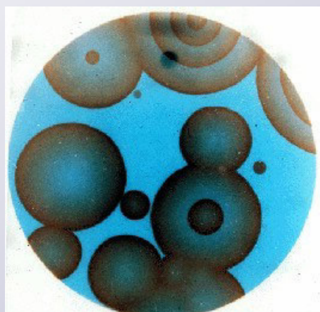
  YES

  - GA generates very convincing designoid patterns

Applications (in design and manufacture) and further work

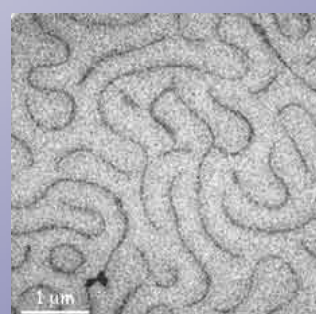
- Many, many systems can be modelled using CAs

- Research into chemical 'design'



e.g. designoid patterns in the BZ reaction

and self-organising nanostructured systems



We are actively working towards these practical goals in the context of the EPSRC grant *CHELLnet* (EP/D023343/1), which comprises
Evolvable CHELLware (EP/D021847/1),
vesiCHELL (EP/D022304/1),
brainCHELL (EP/D023645/1) and
wellCHELL (EP/D023807/1).

CHELLNet
http://www.chellnet.org

Evolutionary design for the behaviour of cellular automaton-based complex systems

Peter Siepmann, Germán Terrazas, Natalio Krasnogor

School of Computer Science & IT
University of Nottingham

Adaptive Computing in Design and Manufacture
25th April 2006
Bristol